Documentation 8.0

ZABBIX

24.10.2025

Contents

Manual do Zabbix **6**

Aviso de direitos autorais . . . 6

1 Introdução . . . 6

1 Estrutura do manual . . . 6

2 O que é o Zabbix . . . 7

3 Funcionalidades do Zabbix . . . 7

4 Visão geral do Zabbix . . . 8

5 Novidades do Zabbix 8.0.0 . . . 9

6 Novidades no Zabbix 8.0.x . . . 11

2 Definições . . . 11

3 Processos do Zabbix . . . 14

1 Server . . . 14

2 Agent . . . 22

3 Agent 2 . . . 26

4 Proxy . . . 29

5 Java gateway . . . 33

6 Sender . . . 37

7 Obter . . . 37

8 JS . . . 38

9 Serviço web . . . 39

4 Instalação . . . 39

1 Obtendo o Zabbix . . . 39

2 Requisitos . . . 40

3 Instalação a partir do código-fonte . . . 49

4 Instalação a partir de pacotes . . . 64

5 Instalação a partir de containers . . . 77

6 Instalação da interface web . . . 86

7 Procedimento de atualização . . . 90

8 Problemas conhecidos . . . 103

9 Alterações em templates . . . 114

10 Notas de atualização para 8.0.0 . . . 114

11 Notas de atualização para 8.0.x . . . 114

5 Início rápido . . . 115

1 Login e configuração do usuário . . . 115

2 Novo host . . . 116

3 Novo item . . . 118

4 Novo trigger . . . 119

5 Recebendo notificações de problemas . . . 121

6 Novo template . . . 125

6 Zabbix appliance . . . 127

7 Configuração . . . 130

1 Hosts e grupos de hosts . . . 139

2 Items . . . 156

3 Triggers . . . 357

4 Eventos . . . 375

5 Correlação de eventos . . . 379

6 Tagging . . . 385

7 Visualização . . . 387

8 Templates e grupos de templates . . . 411

9 Templates prontos para uso . . . 411

10 Notificações em eventos . . . 424
11 Macros . . . 463
12 Usuários e grupos de usuários . . . 477
13 Armazenamento de segredos . . . 485
14 Relatórios agendados . . . 492
15 Exportação de dados . . . 495
8 Monitoramento de serviços . . . 501
1 Árvore de serviços . . . 502
2 SLA . . . 505
3 Exemplo de configuração . . . 507
9 Monitoramento web . . . 511
1 Itens de monitoramento web . . . 519
2 Cenário da vida real . . . 521
10 Monitoramento de máquinas virtuais . . . 526
1 Chaves de item de monitoramento do VMware . . . 527
2 Campos de chave de descoberta de máquina virtual . . . 545
3 exemplos de JSON para items VMware . . . 549
4 Exemplo de configuração de monitoramento VMware . . . 554
11 Manutenção . . . 557
12 Expressões regulares . . . 561
13 Reconhecimento de problema . . . 567
1 Supressão de problema . . . 569
14 Exportação/importação de configuração . . . 570
1 Grupos de templates . . . 572
2 Grupos de hosts . . . 572
3 Templates . . . 572
4 Hosts . . . 589
5 Mapas de rede . . . 605
6 Tipos de mídia . . . 613
15 Descoberta . . . 620
1 Descoberta de rede . . . 620
2 Autoregistro de agent ativo . . . 628
3 Descoberta de baixo nível . . . 631
16 Monitoramento distribuído . . . 680
1 Proxies . . . 681
17 Criptografia . . . 689
1 Usando certificados . . . 697
2 Usando chaves pré-compartilhadas . . . 703
3 Solução de problemas . . . 706
18 Interface web . . . 709
1 Menu . . . 709
2 Seções do frontend . . . 715
3 Configurações do usuário . . . 942
4 Pesquisa global . . . 947
5 Modo de manutenção do frontend . . . 949
6 Parâmetros de página . . . 950
7 Definições . . . 953
8 Criando seu próprio tema . . . 953
9 Modo de depuração . . . 954
10 Cookies usados pelo Zabbix . . . 955
11 Fusos horários . . . 956
12 Redefinindo a senha . . . 957
13 Seletor de período de tempo e host . . . 957
19 Melhores práticas . . . 959
1 Melhores práticas de segurança . . . 959
2 Melhores práticas de configuração . . . 967
20 API . . . 968
Referência de métodos . . . 973
Apêndice 1. Comentário de referência . . . 1778
Apêndice 2. Alterações da 7.4 para a 8.0 . . . 1784
Apêndice 3. Alterações na 8.0 . . . 1785
21 Extensões . . . 1785
1 Módulos carregáveis . . . 1787

2 Plugins . . . 1796
3 Módulos do frontend . . . 1805
22 Apêndices . . . 1806
1 Instalação e configuração . . . 1806
2 Configuração de processos . . . 1864
3 Protocolos . . . 1947
4 Items . . . 1972
5 Funções suportadas . . . 2006
6 Macros . . . 2037
7 Símbolos de unidade . . . 2074
8 Sintaxe do período de tempo . . . 2076
9 Execução de comandos . . . 2077
10 Compatibilidade de versões . . . 2078
12 Biblioteca de vínculo dinâmico do Zabbix sender para Windows . . . 2079
13 Biblioteca Python para Zabbix API . . . 2079
14 Atualização do monitoramento de serviços . . . 2080
15 Outros problemas . . . 2080
16 Comparação entre agent e agent 2 . . . 2081
17 Exemplos de escape . . . 2082
23 Guias de referência rápida . . . 2084
Visão geral . . . 2084
1 Monitorar Linux com o agent do Zabbix . . . 2084
2 Monitorar Windows com o Zabbix agent . . . 2088
3 Monitorar o Apache via HTTP . . . 2091
4 Monitorar o MySQL com o Zabbix agent 2 . . . 2095
5 Monitorar VMware com o Zabbix . . . 2102
6 Monitorar o tráfego de rede com o Zabbix . . . 2106
7 Monitorar o tráfego de rede usando verificações ativas . . . 2109
8 Monitorar sites com itens do Browser . . . 2111
9 Monitorar certificados de sites com o Zabbix agent 2 (passivo) . . . 2115
10 Monitore um switch ou roteador de rede com o Zabbix . . . 2120
11 Monitorar o log de eventos do Windows usando verificações ativas . . . 2127

Zabbix Cloud **2131**
Aviso de direitos autorais . . . 2131
1 Implantar o Zabbix com o Zabbix Cloud . . . 2131
Introdução . . . 2131
Gerenciamento de organizações . . . 2132
Criando um node . . . 2132
Acessando o frontend do Zabbix . . . 2134
2 Configuração de nó . . . 2136
Visão geral . . . 2136
Aba Visão geral . . . 2137
Aba de filtros de acesso . . . 2138
Aba Criptografia . . . 2139
Aba Backups . . . 2140
Aba Histórico . . . 2140
Aba de manutenção . . . 2141
Guia de atualização . . . 2142
3 Adicionando usuários . . . 2142
Introdução . . . 2142
Adicionando usuários às organizações . . . 2142
Adicionando usuários aos nodes . . . 2144
4 Principais diferenças entre Zabbix Cloud e on-premises . . . 2144
Introdução . . . 2144
Diferenças de recursos . . . 2145

Central do Desenvolvedor **2146**
Aviso de direitos autorais . . . 2146
Módulos . . . 2147
Estrutura de arquivos do módulo . . . 2147
Widgets . . . 2155
Tutoriais . . . 2163
Exemplos . . . 2189

Plugins . . . 2189
Exemplos . . . 2192
Criar um plugin (tutorial) . . . 2192
Interfaces de plugin . . . 2197
Alterações no desenvolvimento de extensões . . . 2197

Páginas de manual do Zabbix **2197**
zabbix_agent2 . . . 2197
NOME . . . 2197
SINOPSE . . . 2198
DESCRIÇÃO . . . 2198
OPÇÕES . . . 2198
ARQUIVOS . . . 2199
VEJA TAMBÉM . . . 2199
Índice . . . 2199
zabbix_agentd . . . 2199
NOME . . . 2199
SINOPSE . . . 2199
DESCRIÇÃO . . . 2200
OPÇÕES . . . 2200
ARQUIVOS . . . 2201
VEJA TAMBÉM . . . 2201
Índice . . . 2201
zabbix_get . . . 2201
NOME . . . 2201
SINOPSE . . . 2201
DESCRIÇÃO . . . 2201
OPÇÕES . . . 2202
EXEMPLOS . . . 2203
VEJA TAMBÉM . . . 2203
Índice . . . 2203
zabbix_js . . . 2203
NOME . . . 2203
SINOPSE . . . 2203
DESCRIÇÃO . . . 2203
OPÇÕES . . . 2204
EXEMPLOS . . . 2204
VEJA TAMBÉM . . . 2204
Índice . . . 2204
zabbix_proxy . . . 2204
NOME . . . 2205
SINOPSE . . . 2205
DESCRIÇÃO . . . 2205
OPÇÕES . . . 2205
ARQUIVOS . . . 2206
VEJA TAMBÉM . . . 2206
Índice . . . 2206
zabbix_sender . . . 2206
NOME . . . 2206
SINOPSE . . . 2207
DESCRIÇÃO . . . 2207
OPÇÕES . . . 2207
STATUS DE SAÍDA . . . 2209
EXEMPLOS . . . 2209
VEJA TAMBÉM . . . 2210
Índice . . . 2210
zabbix_server . . . 2210
NOME . . . 2211
SINOPSE . . . 2211
DESCRIÇÃO . . . 2211
OPÇÕES . . . 2211
ARQUIVOS . . . 2212
VEJA TAMBÉM . . . 2213

Índice 2213
zabbix_web_service 2213
NOME 2213
SINOPSE 2213
DESCRIÇÃO 2213
OPÇÕES 2213
ARQUIVOS 2214
VEJA TAMBÉM 2214
Índice 2214

Manual do Zabbix

Bem-vindo ao manual do usuário do software Zabbix. Estas páginas foram criadas para ajudar os usuários a gerenciar com sucesso suas tarefas de monitoramento com o Zabbix, desde as mais simples até as mais complexas.

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 nenhuma forma. Você não deve publicar ou distribuir esta documentação em qualquer forma ou em qualquer 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 distribuída juntamente com o software na mesma mídia. Qualquer outro uso, como qualquer disseminação 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 que não sejam expressamente concedidos acima.

1 Introdução

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

1 Estrutura do manual

Estrutura

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

Ao navegar para as respectivas seções, certifique-se de expandir as pastas das seções para revelar todo o conteúdo incluído nas subseções e páginas individuais.

A ligação cruzada entre páginas de conteúdo relacionado é fornecida tanto quanto possível para garantir que as informações relevantes não sejam perdidas pelos usuários.

Seções

A introdução fornece informações gerais sobre o software Zabbix atual. A leitura desta seção deve lhe 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 de 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 obtenção de dados essenciais, visualização de dados, 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 passo a passo 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.

A 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 por meio 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.

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

Saiba mais sobre as funcionalidades do Zabbix.

Usuários do Zabbix

Muitas organizações de diferentes tamanhos ao redor do mundo confiam no Zabbix como sua principal plataforma 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 pelo server/proxy e pelos agents

Definições flexíveis de limiares

- 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 integrada

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 integrado

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 em vários componentes principais de software. Suas responsabilidades estão descritas abaixo.

Server

O Zabbix server é o componente central para o qual os agents reportam informações de disponibilidade, integridade e estatísticas. O server é o repositório central no qual todos os dados de configuração, estatísticos e operacionais são armazenados.

Armazenamento em banco de dados

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

Interface web

Para um acesso fácil ao Zabbix de qualquer lugar e de qualquer plataforma, a interface baseada na web é fornecida. A interface faz parte do Zabbix server e geralmente (mas não necessariamente) é executada na mesma máquina física que executa o server.

Proxy

O proxy Zabbix pode coletar dados de desempenho e disponibilidade em nome do servidor Zabbix. Um proxy é uma parte opcional da implantação do Zabbix; no entanto, pode ser muito benéfico para distribuir a carga de um único servidor Zabbix.

Agent

Os agents Zabbix são implantados nos alvos de monitoramento para monitorar ativamente recursos e aplicações locais e reportar os dados coletados ao Zabbix server. Desde o Zabbix 4.4, existem dois tipos de agents disponíveis: o Zabbix agent (leve, suportado em muitas plataformas, escrito em C) e o Zabbix agent 2 (extra-flexível, facilmente extensível com plugins, escrito em Go).

Fluxo de dados

Além disso, é importante dar um passo atrás e analisar o fluxo geral de dados dentro do Zabbix. Para criar um item que colete dados, você deve primeiro criar um host. Indo para o outro extremo do espectro do Zabbix, você deve primeiro ter um item para criar um trigger. Você deve ter um trigger para criar uma ação. Assim, se você quiser receber um alerta de que a carga da CPU está muito alta no *Servidor X*, você deve primeiro criar uma entrada de host para o *Servidor X*, seguida de um item para monitorar sua CPU, depois um trigger que é ativado se a CPU estiver muito alta, seguido de uma ação que envia um e-mail para você. Embora isso possa parecer muitos passos, com o uso de templates realmente não é. No entanto, devido a esse design, é possível criar uma configuração muito flexível.

5 Novidades do Zabbix 8.0.0

Veja as mudanças incompatíveis para esta versão.

Templates Novos templates

- Aruba CX 8300s por SNMP, um template que fornece monitoramento baseado em SNMP para a série de switches Aruba CX 8300.
- O conjunto de templates AWS por HTTP foi complementado com o template *AWS Backup Vault por HTTP*.
- Ciena 3906 por SNMP, um template para monitorar dispositivos Ciena 3906.
- Cisco Secure Firewall Threat Defense por HTTP, um template que fornece recursos de monitoramento para dispositivos Cisco Secure Firewall Threat Defense usando a API REST.
- OpenAI Platform por HTTP, um template para monitorar a plataforma de desenvolvedores da OpenAI.
- Stormshield SNS por SNMP, um template para monitorar dispositivos Stormshield Network Security (SNS) via SNMP.
- Vyatta Virtual Router por SNMP, um template para monitorar o roteador virtual Vyatta 1908e.

Templates atualizados

- O template Proxmox VE por HTTP foi atualizado com a funcionalidade LLD aninhada. Além disso, o formato da unidade para items que exibem porcentagens foi ajustado para maior clareza.

Frontend Validação inline no formulário de autorregistro

O formulário de configuração de ação de autorregistro no frontend agora suporta validação inline. Os erros de entrada são exibidos imediatamente após o preenchimento dos campos, melhorando a usabilidade e reduzindo erros de configuração.

Certificados SAML importáveis para single sign-on

Usuários super admin agora podem importar certificados e chaves privadas diretamente no frontend para a configuração do SAML. Três novos campos foram adicionados em *Administração* > *Autenticação* > *SAML*:

- *Certificado IdP* - certificado X.509 apresentado pelo provedor de identidade
- *Certificado SP* - certificado do provedor de serviço usado para trocas SAML
- *Chave privada SP* - chave privada correspondente ao certificado SP

Esses controles permitem adicionar novos valores ou modificar os existentes na aba de configurações SAML. Certificados e chaves privadas são validados antes de serem salvos no backend de armazenamento escolhido; valores inválidos ou malformados são rejeitados com um erro explicativo.

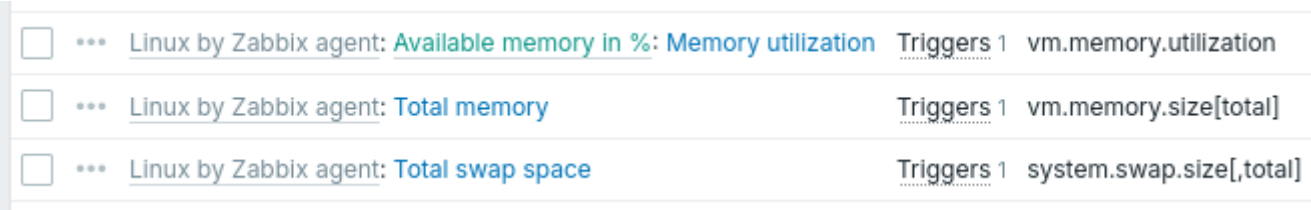
Caixa de seleção Converter para JSON nos formulários de descoberta

Uma nova caixa de seleção *Converter para JSON* foi adicionada ao formulário de Regra de descoberta e ao formulário de Protótipo de descoberta, que é exibida se "HTTP agent" for selecionado no menu suspenso *Tipo*. Esta opção permite marcar os dados recuperados para conversão automática para JSON antes do processamento posterior.

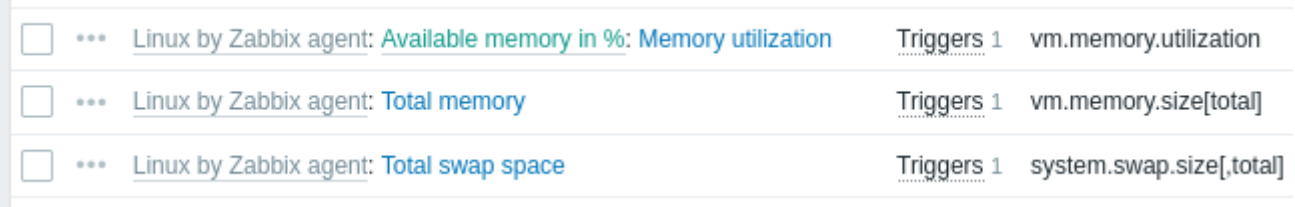
New and embedded fonts

Zabbix now includes new and embedded fonts that improve readability, load faster, and render additional frontend languages with minimal layout impact. Being embedded, these fonts also ensure a consistent appearance across systems.

New fonts:



Legacy fonts:



The new fonts are used in almost all themes. If needed, the newly added *Blue (classic)* and *Dark (classic)* themes are available with legacy fonts.

The font family for monospace content and graphs remains unchanged.

Agrupamento de marcadores de host no Geomap

O widget Geomap agora suporta um parâmetro *Clustering* para controlar como marcadores de host próximos são combinados em um único marcador com uma contagem. Ao definir o nível de zoom do mapa para agrupamento, você pode manter mapas grandes legíveis e manter a visibilidade precisa ao dar zoom.

Documentação Páginas de documentação consolidadas para versões secundárias

A documentação de lançamento para versões secundárias de uma versão principal do Zabbix agora será coletada em páginas únicas de documentação para novos recursos e notas de atualização, respectivamente.

Processos 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 continua disponível para solução de problemas e para dispositivos que não estejam totalmente em conformidade com as especificações SNMPv3.

Recarregamento automático de credenciais SNMPv3

O Zabbix agora armazena em cache os mapeamentos SNMPv3 EngineID → IP e tentará reutilizar os EngineIDs em cache para verificações SNMPv3 subsequentes, reduzindo o tráfego de sondagem e melhorando o desempenho do poller. Se um EngineID reutilizado não responder, o poller retorna a uma sondagem de EngineID e pode remover entradas obsoletas após alterações de interface ou falhas persistentes.

Refinamento do controle de proxy durante a recuperação do cache de histórico

A lógica de controle do proxy foi aprimorada para melhorar a estabilidade do server durante a recuperação do cache de histórico. Quando o uso do cache de histórico atinge o limite de controle, o server continua a parar de aceitar dados do proxy como antes. Quando o uso do cache cai para 60%, o server começa a processar a lista de controle, mas ainda pode rejeitar uploads do proxy que contenham lotes muito grandes (aproximadamente mais de 10 mil registros) até que a pressão do cache diminua ainda mais. Essa alteração reduz o risco de sobrecargas repetidas do cache enquanto o server se recupera.

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.

Items 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 opcional **type** para especificar um valor a ser pesquisado nos discos.

Plugins Plugin Ceph

Este plugin agora opera em dois modos:

- **nativo** - Este modo usa a biblioteca go-ceph para se comunicar diretamente com o cluster Ceph usando a API nativa do Ceph (protocolo msgr2). Este é o modo recomendado para instalações modernas do Ceph, mas só é suportado **em Linux** e a partir do Ceph 16.
- **restful** (obsoleto) - Este modo usa a API RESTful do Ceph para comunicação. É o modo padrão para compatibilidade retroativa, mas não funcionará com o Ceph versão 20 (Tentacle) ou mais recente devido à remoção do módulo mgr/restful.

O modo a ser usado é determinado pelo valor do parâmetro mode (native/restful):

- `Plugins.Ceph.Default.Mode=native` - define o modo nativo para o plugin
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - define o modo nativo para a sessão nomeada

Observe que o conjunto de credenciais de usuário difere para cada modo e eles não são compatíveis entre si. O parâmetro `Plugins.Ceph.InsecureSkipVerify` é ignorado no modo nativo porque a segurança da conexão é definida no lado do cluster Ceph pelo protocolo msgr2 (seguro por padrão).

Observe que o plugin Ceph para o Zabbix agent 2 agora é um plugin carregável e requer etapas adicionais de instalação. Isso se deve ao requisito do pacote librados (para o modo nativo). Veja o readme do plugin Ceph para mais detalhes.

Modo de execução de teste para plugins carregáveis

Plugins carregáveis agora podem ser iniciados em modo de teste usando a flag `-t` (`--test`), passando uma chave de item como argumento. Neste modo, o plugin é executado para fins de depuração e desenvolvimento, e os arquivos de configuração do plugin são ignorados.

Redis plugin — TLS support and startup-time validation

TLS support has been added to the Redis plugin for Zabbix agent 2.

Startup-time validation of the plugin TLS configuration was implemented and validation/error messages improved. An invalid configuration logic (for example: using connection type `verify_full` without specifying `TLSCAFile`) can prevent Zabbix agent 2 from starting.

6 Novidades no Zabbix 8.0.x

Esta página fornece informações coletivas sobre os novos recursos incluídos nas versões secundárias da versão principal do Zabbix.

Veja também Novidades da versão principal.

Para alterações em templates existentes e informações sobre novos templates que abrangem vários dispositivos, serviços e soluções de parceiros, consulte Alterações em templates.

O que há de novo no Zabbix 8.0.1 Esta versão ainda não foi lançada.

2 Definições

Visão 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-lo no banco de dados.

trigger

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

Quando os dados recebidos estão acima do limiar, os triggers passam de 'Ok' para o estado 'Problema'. Quando os dados recebidos estão abaixo do limiar, 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 única ocorrência 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 diferente de coleta de dados.

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 severidade 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.

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 fonte (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, aliviando parte da carga de processamento do server.

Zabbix agent

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

Zabbix agent 2

- uma nova geração do 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 refere-se 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 difere.

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 do 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 determinado dispositivo (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

Use a barra lateral para acessar o conteúdo da seção de processos do Zabbix.

1 Server

Visão geral

O servidor Zabbix é o processo central do software Zabbix.

O servidor 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 servidor pode, ele mesmo, verificar remotamente serviços em rede (como servidores web e servidores de e-mail) usando verificações simples de serviço.

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

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

Todas as informações de configuração do Zabbix são armazenadas no banco de dados, com o qual tanto o servidor 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 itens no banco de dados. Então, cerca de uma vez por minuto, o servidor Zabbix consulta a tabela de itens para obter uma lista dos itens que estão ativos, que é então armazenada em um cache dentro do servidor Zabbix. 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 deve 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 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.	
history_cache_clear=**destino**	Limpa o cache de histórico para o item especificado pelo seu ID. Afeta todos os valores do item, exceto o primeiro e o último valor.	**destino** - ID do item
diaginfo[=**seção**]	Coleta informações de diagnóstico no arquivo de log do servidor.	**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=**destino**	Remove o nó de alta disponibilidade (HA) especificado pelo seu nome ou ID. Observe que nós ativos/standby não podem ser removidos.	**destino** - nome ou ID do nó (pode ser obtido executando ha_status)
ha_set_failover_delay=**atraso**	Define o atraso de failover de alta disponibilidade (HA). Sufixos de tempo são suportados, por exemplo, 10s, 1m.	
proxy_config_cache_reload[=**destino**]	Recarrega a configuração do proxy.	**destino** - lista de nomes de proxies separados por vírgula Se nenhum destino 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 SNMP.	
housekeeper_execute	Inicia o procedimento de housekeeping. Ignorado se o procedimento de housekeeping estiver em andamento.	

Opção	Descrição	Destino
trigger_housekeeper_execute	Inicie o procedimento de housekeeping de trigger para serviços para remover problemas causados por triggers que foram excluídos desde então, incluindo problemas de serviço gerados por tais problemas (considerados como resolvidos no momento do housekeeping). Observe que, até que o procedimento de housekeeping seja iniciado, problemas causados por triggers agora 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 de trigger ajustando o parâmetro de configuração do servidor ProblemHousekeepingFrequency. Ignorado se o procedimento de housekeeping de trigger estiver em andamento.	
log_level_increase[=<**destino**>]	Aumente 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 servidor. **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[=<**destino**>]	Diminua o nível de log, afeta todos os processos se o destino não for especificado. Não suportado em sistemas *BSD*.	
prof_enable[=<**destino**>]	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) Tipos de processos suportados como destinos 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 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[=<**destino**>]	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) Tipos de processos suportados como destinos 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 destino 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
```

Exemplo de uso do controle em tempo de execução para limpar o cache de histórico de um item:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

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 as 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 o iniciar. 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', 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, com relativa facilidade, recuperar, por exemplo, a senha do banco de dados.

Arquivo de configuração

Veja as opções do arquivo de configuração para obter detalhes sobre como configurar o 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 do servidor e threads

- `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 hosts
- `browser poller` - poller para verificações de itens 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 itens
- `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 requerem conexão com o banco de dados
- `history syncer` - gravador de histórico no banco de dados
- `housekeeper` - processo para remoção de dados históricos antigos
- `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 proxies
- `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 gerados por triggers que foram excluídos
- `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.

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

Estatísticas de transação do history syncer

O título do processo history syncer exibe estatísticas detalhadas sobre as transações do history syncer:

```
205182 ?        S      0:00  zabbix_server: history syncer #2 [processed 0 values, 0+0 triggers in 0.00002
205183 ?        S      0:00  zabbix_server: history syncer #3 [processed 18 values, 7+0 triggers in 0.0026
205184 ?        S      0:00  zabbix_server: history syncer #4 [processed 0 values, 0+0 triggers in 0.00002
```

Em "A+B triggers":

- A - triggers processados devido a valores de histórico;
- B - triggers processados devido a timers.

Os tempos, em "processed...in N (<timings>) sec", são:

- Tempo gasto gravando valores de item no banco de dados;
- Tempo gasto atualizando dados do item (estado, erros, inventário do host, etc);
- Tempo gasto gravando tendências no banco de dados;
- Tempo gasto calculando triggers;
- Tempo gasto processando eventos e ações.

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, tolerância a falhas e 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 pode funcionar em outros sistemas operacionais do tipo 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 possui 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 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 standalone a qualquer momento.

Veja também: Detalhes da implementação

Ativando alta disponibilidade

Iniciando o Zabbix server como nó de cluster

Dois parâmetros são necessários na configuração do server para iniciar o 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 conectar-se 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 node ativo lendo as configurações da tabela de nodes no banco de dados do Zabbix. O endereço do node 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**.

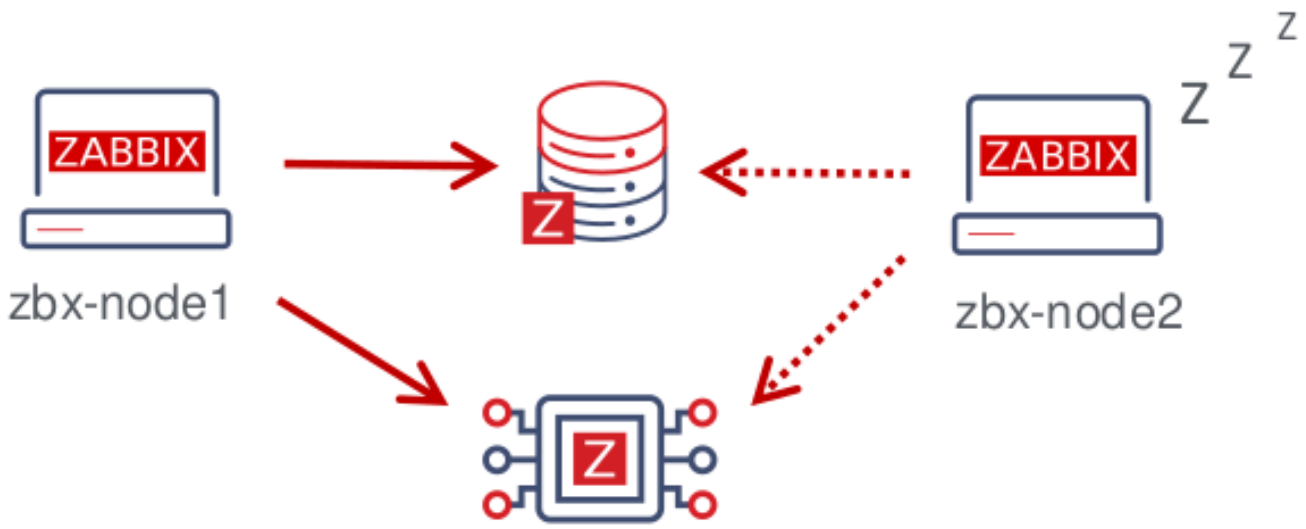
```
Server=zabbix-node-01,zabbix-node-02
```

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 verificações passivas, os nomes dos nós devem ser listados no parâmetro Server parameter, separados por uma **vírgula**.

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

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

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

Failover para o nó em espera

O Zabbix fará o failover para outro nó automaticamente se o nó ativo parar. Deve haver pelo menos um nó em status de espera 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 em espera aguardarão o **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 (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/em espera 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 dos nós 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 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 fará 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, funciona entre sites e não possui requisitos específicos para os bancos de dados reconhecidos pelo Zabbix. Os usuários são livres para usar a solução nativa de HA do Zabbix ou uma solução de HA de terceiros, dependendo do que melhor atenda 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 valor de 'failover delay' em segundos, o nó standby se torna o nó ativo e atribui o status 'indisponível' ao nó anteriormente ativo.

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 'failover delay' segundos, 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 do Zabbix é implantado em um alvo de monitoramento para monitorar ativamente recursos e aplicações locais (discos rígidos, memória, estatísticas do processador, etc.).

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

Os agents do 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 aguardar 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 Requisitos.

Agent em sistemas UNIX-like

O agent 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 a versão do Zabbix, a distribuição do SO e a 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 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 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 agent do Zabbix 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 de instalação MSI do agent Zabbix 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 no Windows.

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

Opções

É possível executar múltiplas 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 em 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 controle em tempo de execução.
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 o início do serviço após os serviços iniciados automaticamente terem concluído a inicialização (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 dos 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 ao estendê-lo 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	Destino
log_level_increase[=<alvo>]	Aumenta o nível de log. Se o destino não for especificado, todos os processos serão afetados.	O destino 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 destino como 'tipo-de-processo,N'.
log_level_decrease[=<alvo>]	Diminui o nível de log. Se o destino 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 execução de 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.

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', 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 do Zabbix, consulte as opções do arquivo de configuração para zabbix_agentd ou agent do 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, há alguns sistemas em que 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.

Checagens passivas e ativas

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

Note:
Por padrão, após uma reinicialização, o agent Zabbix 2 agendará a primeira coleta de dados para checagens ativas em um tempo condicionalmente aleatório dentro do intervalo de atualização do item para evitar picos no uso de recursos. Para executar 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 plugin específicas) no arquivo de configuração. O parâmetro em nível de 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 Zabbix 2 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 fornecidas.
- Fontes do Zabbix - baixe os arquivos fonte e compile o agent Zabbix configurando as fontes com a opção `--enable-agent2`. Observe que um ambiente Go configurado com uma versão Go suportada é necessário para compilar o agent Zabbix 2.

Se instalado como pacote

O Zabbix agent 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 Zabbix agent 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 Zabbix agent 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 Zabbix 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. Observe que é necessário um ambiente Go configurado com uma versão Go suportada para compilar o agent Zabbix 2.

Note:
As capacidades de monitoramento do agent Zabbix 2 podem ser estendidas com plugins. Enquanto os plugins embutidos estão disponíveis por padrão, plugins carregáveis no Windows devem ser instalados separadamente. Para mais informações, consulte Plugins carregáveis.

Consulte a página Agent Zabbix no Microsoft Windows para detalhes adicionais sobre como instalar o 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 (caso não esteja 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 (caso não esteja 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.
-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 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.

Parâmetro	Descrição
-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
```

Arquivo de configuração

Os parâmetros de configuração do agent 2 são em sua maioria compatíveis com o agent do Zabbix, com algumas exceções.

Novos parâmetros	Descrição
ControlSocket	O caminho do socket de controle em tempo de execução. O 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 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	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 agent 2 ficará ouvindo para solicitações de status HTTP e exibição de uma lista de plugins configurados e alguns parâmetros internos
Parâmetros removidos	**Descrição**
AllowRoot, *User*	Não suportado porque a execução como daemon não é suportada.
LoadModule, *LoadModulePath*	Módulos carregáveis não são suportados.

Novos parâmetros	Descrição
StartAgents	Este parâmetro era usado no agent do Zabbix para aumentar a concorrência de verificações passivas ou desativá-las. No Agent 2, a concorrência é configurada no nível do plugin e pode ser limitada por uma configuração de capacidade. Já a desativação de verificações passivas não é suportada atualmente.
HostInterface, *HostInterfaceItem*	Ainda não suportado.

Para mais detalhes, consulte as opções do arquivo de configuração para zabbix_agent2.

Códigos de saída

O agent Zabbix 2 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 de mutex ou desbloqueio falhar, uma mensagem de erro é impressa no fluxo de erro padrão (STDERR) e o Agent 2 sai com o código de retorno 2 ou 3, respectivamente.

4 Proxy

Visão geral

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

Implantar um proxy é opcional, mas pode ser muito benéfico para distribuir a carga de um único server Zabbix. Se apenas proxies coletarem dados, o processamento no server 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.

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 iniciar 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	Alvo
config_cache_reload	Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O proxy ativo do Zabbix irá se conectar ao servidor Zabbix e solicitar os dados de configuração. O proxy passivo do Zabbix solicitará os dados de configuração ao servidor Zabbix na próxima vez que o servidor se conectar ao proxy.	
history_cache_clear[=**alvo**]	Limpa o cache de histórico para o item especificado pelo seu ID. Afeta todos os valores do item, exceto o primeiro e o último valor.	**alvo** - ID do item
diaginfo[=**seção**]	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 vazio 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 de SNMP.	
housekeeper_execute	Inicia o procedimento de housekeeping. Ignorado se o procedimento de housekeeping estiver em andamento.	
log_level_increase[=**alvo**]	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 (ex: poller) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (ex: poller,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, afeta todos os processos se o alvo não for especificado. Não suportado em sistemas *BSD*.	
prof_enable[=**alvo**]	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 (ex: history syncer) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (ex: history syncer,1) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o alvo como 'tipo de processo,N'. **escopo** - `rwlock`, `mutex`, `processing` podem ser usados com o tipo de processo e número (ex: history syncer,1,processing) ou todos os processos do tipo (ex: history syncer,rwlock)

Opção	Descrição	Alvo
prof_disable[=<**alvo**>]	Desabilita o profiling. Afeta todos os processos se o alvo não for especificado.	**tipo de processo** - Todos os processos do tipo especificado (ex: history syncer) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (ex: 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 proxy:

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

Exemplo de uso do controle em tempo de execução para limpar o cache de histórico de um item:

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

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 irá recarregar 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 nas credenciais (por exemplo, devido a inconsistências de engineBoots/engineID ou dispositivos não-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 em seu sistema. Você só pode executar o proxy como 'root' se modificar o parâmetro 'AllowRoot' no arquivo de configuração do proxy de acordo.

Arquivo de configuração

Veja as opções do arquivo de configuração para detalhes sobre a configuração do 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 de 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 remoção de dados históricos antigos
- 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 coleta de estatísticas internas do server
- 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 inacessíveis
- vmware collector - coletor de dados VMware responsável pela coleta de dados de serviços VMware

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

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

Estatísticas de transações do history syncer

O título do processo history syncer exibe estatísticas detalhadas sobre as transações do history syncer.

```
205276 ?        S      0:00  zabbix_proxy: history syncer #1 [processed 1 values in 0.001179 (0.001167,0.0
205277 ?        S      0:00  zabbix_proxy: history syncer #2 [processed 0 values in 0.000022 (0.000000,0.0
```

Os tempos, em "processed...in N (<timings>) sec", são:

- Tempo gasto gravando valores de item no banco de dados;
- Tempo gasto atualizando dados do item (estado, erros).

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, auto-registro 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 ao 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 utilizado, 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, com configurações conflitantes, 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 das filas durante a manutenção

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

5 Java gateway

Visão geral

O Java gateway do Zabbix 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 do Zabbix chamado ”Zabbix Java gateway”. O Zabbix Java gateway é um daemon escrito em Java. Para descobrir o valor de um determinado contador JMX em um host, o Zabbix server consulta o Zabbix Java gateway, que utiliza a API de gerenciamento JMX para consultar remotamente a aplicação de interesse. A aplicação não precisa de nenhum software adicional instalado, apenas deve 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 levar 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 timeout para operações de rede JMX.

O servidor ou proxy 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.

É recomendado que **StartJavaPollers** seja menor ou igual a **START_POLLERS**, caso contrário, podem ocorrer situações em que não há threads disponíveis no Java gateway para atender às solicitações recebidas; nesse caso, o Java gateway usa ThreadPoolExecutor.CallerRunsPolicy, o que significa que a thread principal irá atender à solicitação recebida e não aceitará novas solicitações temporariamente.

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

1 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 biblioteca Android JSON.

```
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 incluído pelos scripts de inicialização e desligamento 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 em execução 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 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 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 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 um arquivo JAR apropriado no diretório lib. Veja o Manual do SLF4J para mais detalhes.

Monitoramento JMX

Consulte a página Monitoramento JMX para mais detalhes.

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 utilizá-lo, deve especificar o número de instâncias pré-inicializadas 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 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 investigação 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.

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 em execução 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 utilizá-lo, deve especificar o número de instâncias pré-iniciadas dos 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 o proxy após terminar de configurá-los.

Depurando o Java gateway

O arquivo de log do Java gateway do Zabbix 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 detalhada):

```
<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.

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 geralmente é 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 tipo de item trapper deve ser configurado.

Veja também zabbix_utils - uma biblioteca Python 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. Veja 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 são 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]
```

Os cenários de envio em tempo real do zabbix_sender irão agrupar múltiplos valores passados para ele em rápida sucessão e enviá-los para o 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 finalizado se houver uma entrada de parâmetro inválida (que não siga a notação *parâmetro=valor*) 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 formatado em 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 Obter

Visão geral

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

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

Veja também zabbix_utils - uma biblioteca Python que possui funcionalidade integrada 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 agent do Zabbix.

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 para um arquivo contendo o(s) certificado(s) da(s
--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
--tls-key-file <key file>               Caminho completo para 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 para um arquivo contendo a chave pré-compartilhad
--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 ou superior) ou strin
```

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 do parâmetro de entrada. Se '-' for especif
-p, --param input-param           Especifica o parâmetro de entrada.
-l, --loglevel log-level          Especifica o nível de log.
-t, --timeout timeout             Especifica o tempo limite em segundos. Faixa válida: 1-600 segundos (pad
-h, --help                        Exibe informações de ajuda.
-V, --version                     Exibe o número da versão.
-w <webdriver url>                Habilita o monitoramento de navegador.
```

Exemplo:

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

9 Serviço 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 (veja 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 a acessos não autorizados.

4 Instalação

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

1 Obtendo o Zabbix

Visão geral

Existem quatro maneiras de obter o Zabbix:

- Instale a partir dos pacotes de distribuição
- Baixe o último arquivo fonte e compile você mesmo
- Instale a partir dos containers
- Baixe o appliance virtual

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

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 as versões noturnas 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 releases suportados também são espelhados no Github em https://github.com/zabbix/zabbix

Um cliente Git deve estar 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 por 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 completamente 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

Tamanho da instalação	Métricas monitoradas[1]	Núcleos de CPU/vCPU	Memória (GiB)	Banco de dados	Amazon EC2[2]
Grande	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Muito grande	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	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 itens 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, tolerância a falhas e 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	
Windows	-	x	x	O agent do Zabbix é suportado em todas as versões desktop e server desde o Windows XP (64-bit)/Server 2003. O agent 2 do Zabbix é suportado em todas as versões desktop e server desde o Windows 10 (32-bit)/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 desabilitação 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	8.4.0-9.0.X	Necessário se MySQL (ou Percona) for usado como banco de dados backend do Zabbix. O mecanismo InnoDB é obrigatório. Recomendamos usar a biblioteca C API (libmysqlclient) para compilar o server/proxy.
MariaDB		10.11.00-12.0.X	O mecanismo InnoDB é obrigatório. A versão recomendada é a 11.4. Recomendamos usar a biblioteca MariaDB Connector/C para compilar o server/proxy. Veja também: Possíveis deadlocks com MariaDB e Acesso a elementos da interface com MariaDB 10.5.1–10.5.9.
PostgreSQL		15.0-17.X	Necessário se 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 determinada operação seja suficiente e a execução da consulta não demore muito tempo.
TimescaleDB para PostgreSQL		2.20.X-2.21.X	Necessário se 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.
SQLite	Opcional	3.3.5-3.34.X	SQLite é suportado apenas com proxies Zabbix. Necessário se SQLite for usado como banco de dados do proxy Zabbix.
smartmontools		7.1 ou superior	Necessário para o Zabbix agent 2.
who			Necessário para o plugin user count.
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 usar 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 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 - 8.4.X	
Apache	Um dos	2.4 ou posterior	
Nginx		1.20 ou posterior	
MySQL	Um dos	Veja Software externo de terceiros	
PostgreSQL			
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.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); necessário para suporte a script de usuário.
gd		2.0.28 ou posterior	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 posterior pode ser necessária para evitar possíveis sobreposições de texto em gráficos para alguns idiomas do frontend.
libxml		2.6.15 ou posterior	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.
cURL		7.19.4 ou posterior	php-curl; necessário para Duo Universal Prompt MFA e autenticação SMTP.

Bibliotecas de frontend de terceiros fornecidas com o Zabbix:

Biblioteca	Status obrigatório	Versão mínima	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		4.0.0	Um toolkit PHP que adiciona suporte à autenticação SAML 2.0 para poder fazer login no Zabbix.
Symfony Yaml Component		5.1.0	Adiciona suporte para exportar e importar elementos de configuração do Zabbix no formato YAML.

Note:
O Zabbix pode funcionar em versões anteriores do Apache, MySQL e PostgreSQL também.

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 aplicação 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-processamento 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 dashboards em `http://secure-zabbix.com/zabbix/`, terá acesso total ao JS do Zabbix.

Server/proxy

Se declarado como obrigatório, o software/biblioteca é estritamente necessário. Os opcionais são necessários para suportar alguma função específica.

Requisito	Status obrigatório	Descrição
libpcre2	Sim	A biblioteca PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). PCRE2 v10.x é suportado.
libevent		Necessário para comunicação entre processos. Versão 2.0.10 ou superior.
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.
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 1.8.0 ou superior (libssh2); 0.9.0 ou superior (libssh).

Requisito	Status obrigatório	Descrição
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 (Básica: versão 7.20.0 ou superior; OAuth: versão 7.33 ou superior; veja também os requisitos do frontend); - Elasticsearch (versão 7.28.0 ou superior). É recomendada a versão 7.28.0 ou superior para todos os recursos. Para usar recursos aprimorados 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 posterior.
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. Versão mínima: 1.16.0

Agent

Requisito	Status obrigatório	Descrição
libpcre2	Sim	A biblioteca PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). PCRE2 v10.x é suportada. Necessária para monitoramento de logs. Também é necessária no Windows.
libpthread		Necessária para suporte a mutex e bloqueio de leitura-escrita (pode fazer parte da libc). Não é necessária no Windows.
libresolv		Necessária para resolução de DNS (pode fazer parte da libc). Não é necessária no Windows.
libiconv		Necessária 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 da libc). Não é necessária no Windows.
libgnutls ou *libopenssl*	Não	Necessária se estiver usando criptografia. Versões mínimas: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 No Microsoft Windows, é necessário OpenSSL 1.1.1 ou superior.
libldap		Necessária se LDAP for usado. Não é suportada no Windows.
libcurl		Necessária para suporte estendido aos itens web.page.* do Zabbix agent. Sem libcurl, a funcionalidade básica está disponível (por exemplo, web.page.get[http://example.com]). Com libcurl, o agent suporta recursos adicionais, como URLs HTTP com credenciais (por exemplo, http://user:password@example.com) e URLs HTTPS. Versão 7.19.1 ou superior é necessária (7.28.0 ou superior é recomendada). Para usar recursos aprimorados do cURL, reinicie o Zabbix agent.
libmodbus		Necessária 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 1.23 ou superior 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 (bibliotecas marcadas como `indirect` no repositório são dependências de outras bibliotecas necessárias): - Zabbix agent 2 - Suporte a plugin - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Plugin de exemplo
libpcre2	Sim	A biblioteca PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). PCRE2 v10.x é suportado. Necessário para monitoramento de logs. Também é necessário no Windows.
libopenssl	Não	Necessário ao usar criptografia. OpenSSL 1.0.1 ou superior 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 superior 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 de código-fonte ou de um arquivo compactado, então as dependências necessárias já estão incluídas na árvore de código-fonte.

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 também com outras versões dessas bibliotecas.

A tabela a seguir lista os arquivos JAR que atualmente são fornecidos junto com o Java gateway no código original:

Biblioteca	Status obrigatório	Versão mínima	Comentários
android-json	Sim	4.3r1	JSON (JavaScript Object Notation) é um formato leve de intercâmbio de dados. Esta é a implementação 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 mais recente). Os pacotes fornecidos pelo Zabbix são compilados usando o OpenJDK. A tabela a seguir lista os pacotes OpenJDK usados para compilar os pacotes do 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)

Distribuição	Pacote OpenJDK
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
Zabbix trapper	10051	TCP	sob demanda

Note:
Os números das portas devem estar 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 destas 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 servidor Zabbix 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 utilizado, 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** por 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 exigirá 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 de 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 de 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 uma única tendência, depende do mecanismo de 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 utilizado imediatamente após a instalação do Zabbix. O tamanho do banco de dados crescerá e depois parará de crescer em algum ponto, 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.

1 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-fonte executando:

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

Note:
Insira 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 não privilegiados. Se um daemon do Zabbix for iniciado a partir de uma conta de usuário não privilegiada, 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 há necessidade de 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 banco de dados Zabbix

Para os processos 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 schema do banco de dados e inserir o conjunto de dados. O banco de dados do proxy Zabbix precisa apenas do schema, enquanto o banco de dados do server Zabbix requer também o conjunto de dados além do schema.

Tendo criado um banco de dados Zabbix, prossiga para as próximas etapas 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:
Um ambiente Go configurado com uma versão Go suportada é necessário para compilar o Zabbix agent 2.

Notas sobre 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-libpcre2[=DIR]` - o Zabbix sempre compila com a biblioteca PCRE2; esta opção apenas permite especificar um caminho de instalação personalizado para o PCRE2.
- `--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.
- `--enable-static` - vincula as bibliotecas estaticamente (não suportado no Solaris). Use isso se você planeja distribuir binários compilados para sistemas sem as bibliotecas necessárias. Não recomendado ao compilar o Zabbix server. Para compilar o server estaticamente, é necessária uma versão estática de cada biblioteca externa. 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 faltando, 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 do 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 do Zabbix no arquivo. Conexões de outros hosts serão negadas.

- edite o arquivo de configuração do server do 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) do Zabbix.

- se você instalou um proxy do 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
```

2 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.

Locais comuns de 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 recomendado 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.

3 Instalando o Java gateway

É necessário instalar o Java gateway apenas se você deseja 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 código-fonte.

Para compilar o Java gateway, execute o script `./configure` com a opção `--enable-java`. Recomenda-se especificar a opção `--prefix` para solicitar um caminho de instalação diferente do padrão /usr/local, porque 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 você 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 o setup para mais detalhes sobre configuração e execução do Java gateway.

4 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:**
> Um ambiente Go configurado com uma versão Go suportada é necessário 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 x64 Native Tools Command Prompt (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 Zabbix com vcpkg

Esta seção contém instruções para compilar o agent 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 de área de trabalho 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 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 de origem do Zabbix e extraia-o para `C:\Zabbix\zabbix-8.0.0`.

4. Navegue até o diretório de compilação do Zabbix (`C:\Zabbix\zabbix-8.0.0\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 compilação:
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 Zabbix executando o script:

```
build.bat
```

Após a compilação, os binários dos componentes do Zabbix estarão localizados em `C:\Zabbix\zabbix-8.0.0\bin\win64`. O arquivo de configuração do agent Zabbix está localizado em `C:\Zabbix\zabbix-8.0.0\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-8.0.0\bin\win64\zabbix_agent.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agent.win.conf C:\Zabbix\agent\

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

Compilando o agent do Zabbix manualmente

> **Attention:**
> Este método de compilação do agent do 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 do 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 Zabbix sem suporte a TLS, prossiga para a seção Instalando o PCRE2.

1. Baixe e instale o Strawberry Perl (disponível como um 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 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 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.

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-8.0.0`.

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 build 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-8.0.0\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-8.0.0\bin\win64`. O arquivo de configuração do agent do Zabbix está localizado em `C:\Zabbix\zabbix-8.0.0\conf`.

Para executar o agent, copie `zabbix_agent.exe` e seu arquivo de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent`) e então execute o agent:

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

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

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).

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-8.0.0`.

6. Navegue até o diretório de build do Zabbix (`C:\Zabbix\zabbix-8.0.0\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 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 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-8.0.0\bin\win64` (para compilações 64-bit) ou `C:\Zabbix\zabbix-8.0.0\bin\win32` (para compilações 32-bit). Os arquivos de configuração do Zabbix agent 2 estão localizados em `C:\Zabbix\zabbix-8.0.0\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-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### Para compilações 32-bit:
copy C:\Zabbix\zabbix-8.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

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

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

Se 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-embe` e extraia-o para `C:\Zabbix`.

2. Navegue até o diretório do plugin extraído e compile o plugin:

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

#### 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 ember.conf para o plugin Ember+:

```
Plugins.EmberPlus.System.Path=/path/to/executable/zabbix-agent2-plugin-ember-plus
```

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=/path/to/plugin/configuration/file/ember.conf
```

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, incluindo a instalação das ferramentas de compilação e dependências necessárias e, em seguida, a compilação do agent.

Configurando as ferramentas de compilação

1. Baixe e instale o MSYS2 (disponível como 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 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 64-bit: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Para builds 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 builds 32-bit).

Instalando o OpenSSL

> **Note:**
> Para compilar o agent do Zabbix sem suporte a TLS, prossiga 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 possível 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 do 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.

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 de origem do PCRE2 e extraia-o 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 de 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-o para `C:\Zabbix\zabbix-8.0.0`.

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-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw64\bin;%PATH%
mklink /D C:\Zabbix\x64\OpenSSL-Win64-350-static\lib C:\Zabbix\x64\OpenSSL-Win64-350-static\lib64

#### 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-8.0.0\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-8.0.0\bin\win64` (ou `C:\Zabbix\zabbix-8.0` para builds 32-bit). Os arquivos de configuração do Zabbix agent 2 estão localizados em `C:\Zabbix\zabbix-8.0.0\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-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

Se necessário, continue com a compilação dos plugins carregáveis do Zabbix agent 2.

3 Compilando o agent Zabbix no macOS

Visão geral

Esta seção demonstra como compilar os 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, disponível em 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, são máquinas macOS que usam o Homebrew para compilar os binários do agent 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 do código-fonte e vincular o agent 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 o mini-gmp, que está incluído 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 8.0 PRÉ-LANÇAMENTO 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.

Os pacotes estão disponíveis para as seguintes distribuições Linux:

- Red Hat Enterprise Linux e seus derivados: AlmaLinux, Amazon Linux 2023, CentOS Stream, CentOS 7, Oracle Linux, Rocky Linux
- Debian, Ubuntu, Raspberry Pi OS, Raspbian
- SUSE Linux Enterprise Server, openSUSE Leap

> **Attention:**
> 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 o Zabbix anteriormente a partir do repositório do seu sistema operacional, veja 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.

O Zabbix também fornece binários pré-compilados do Zabbix agent para sistemas operacionais não-Linux; veja:

- Instalação do agent do Windows a partir do MSI
- Instalação do agent do 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 em Configuração do TimescaleDB.
- Se estiver instalando o Java gateway do Zabbix (para monitorar aplicações JMX), veja instruções adicionais para sistemas baseados em RHEL e 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 habilitou o repositório EPEL para EL9, que também fornece pacotes 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 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), também permita 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 básica do SELinux, permitindo que o Zabbix crie e use 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 7, edite `/etc/yum.repos.d/zabbix.repo` e defina `enabled=1` para a seção `zabbix-debuginfo`:

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

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

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 de SO 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.

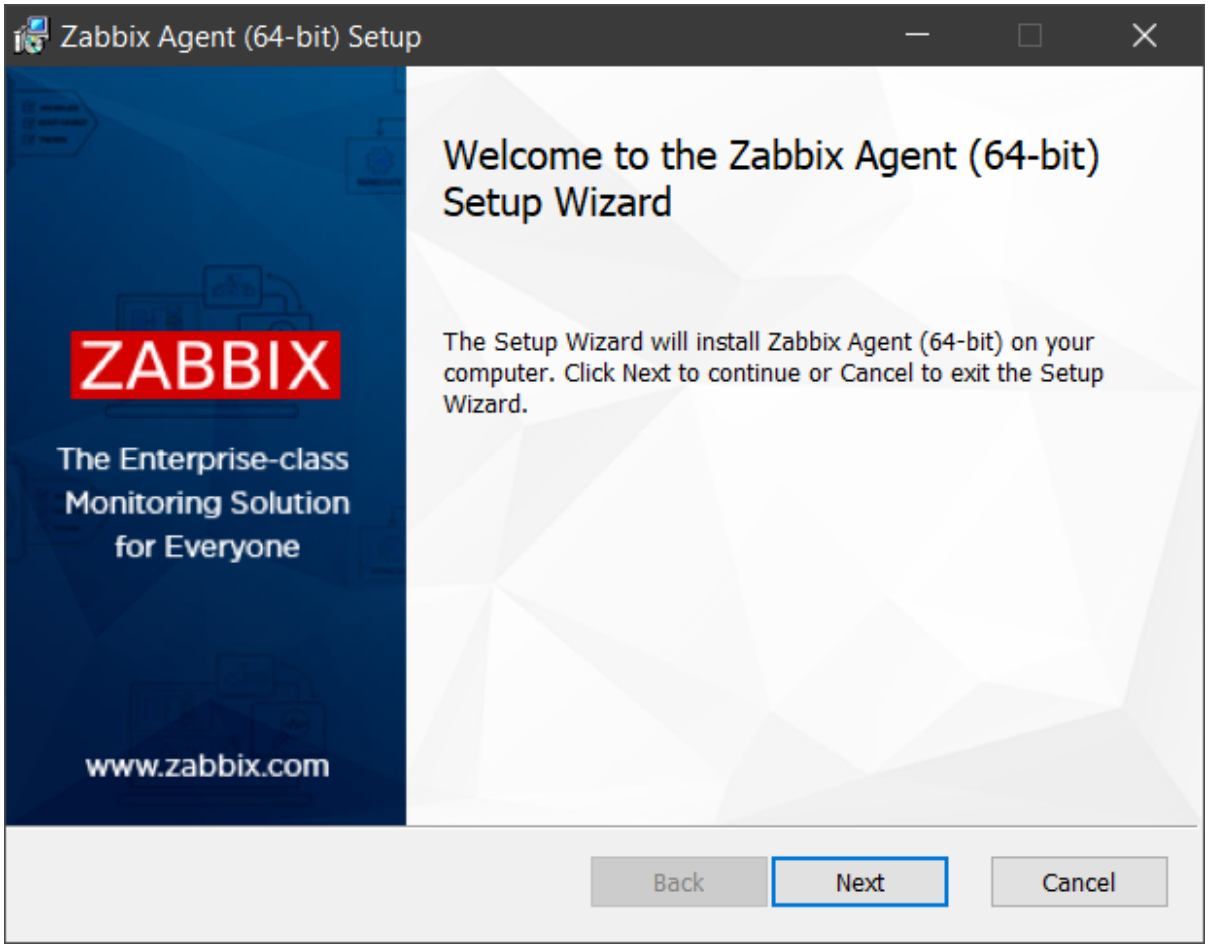
Embora a instalação usando pacotes MSI seja totalmente suportada, recomenda-se instalar pelo menos o Microsoft .NET Framework 2 para um tratamento adequado de erros.

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:

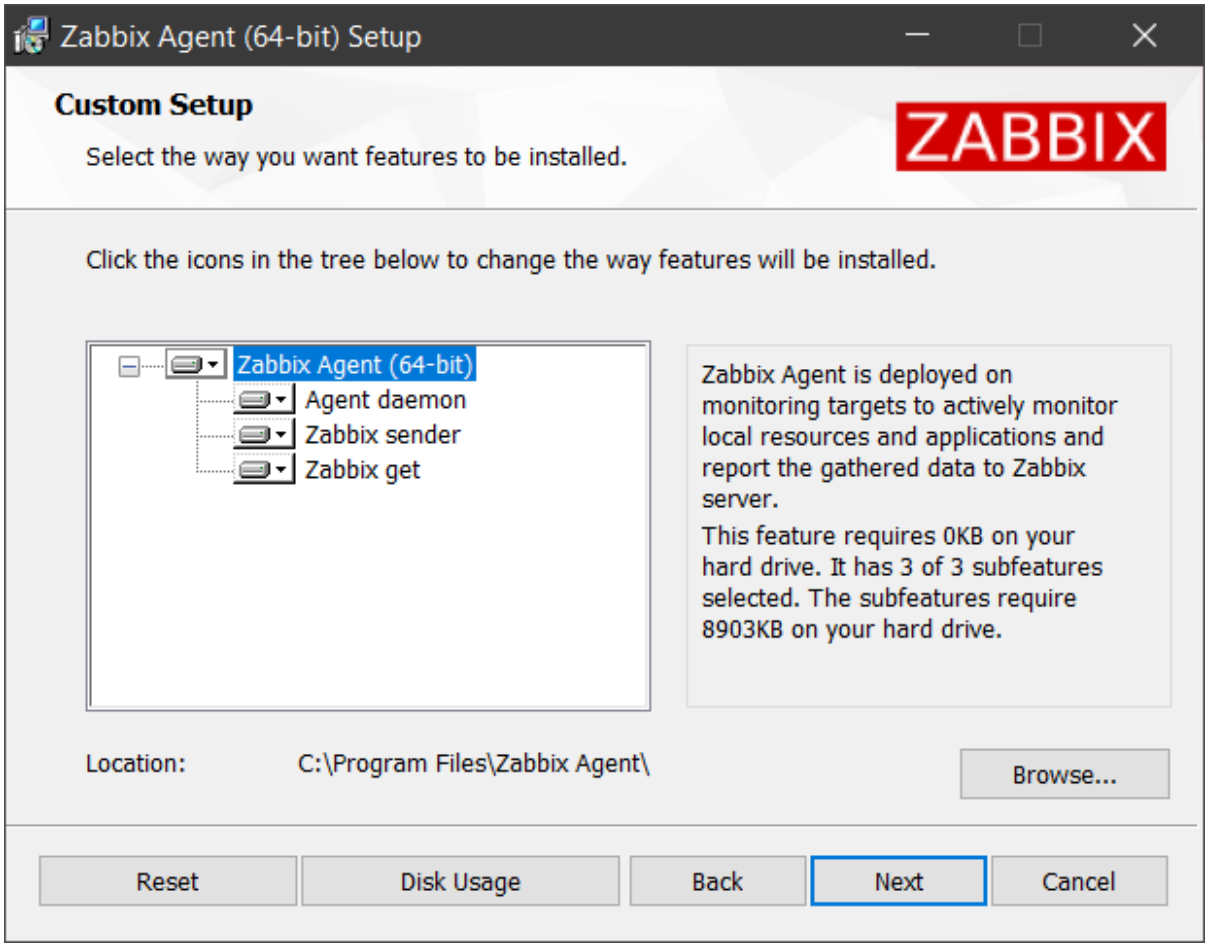


2. Aceite o Contrato de Licença do Usuário Final:



3. Selecione os componentes do Zabbix (Agent daemon, Zabbix sender, Zabbix get) a serem instalados:

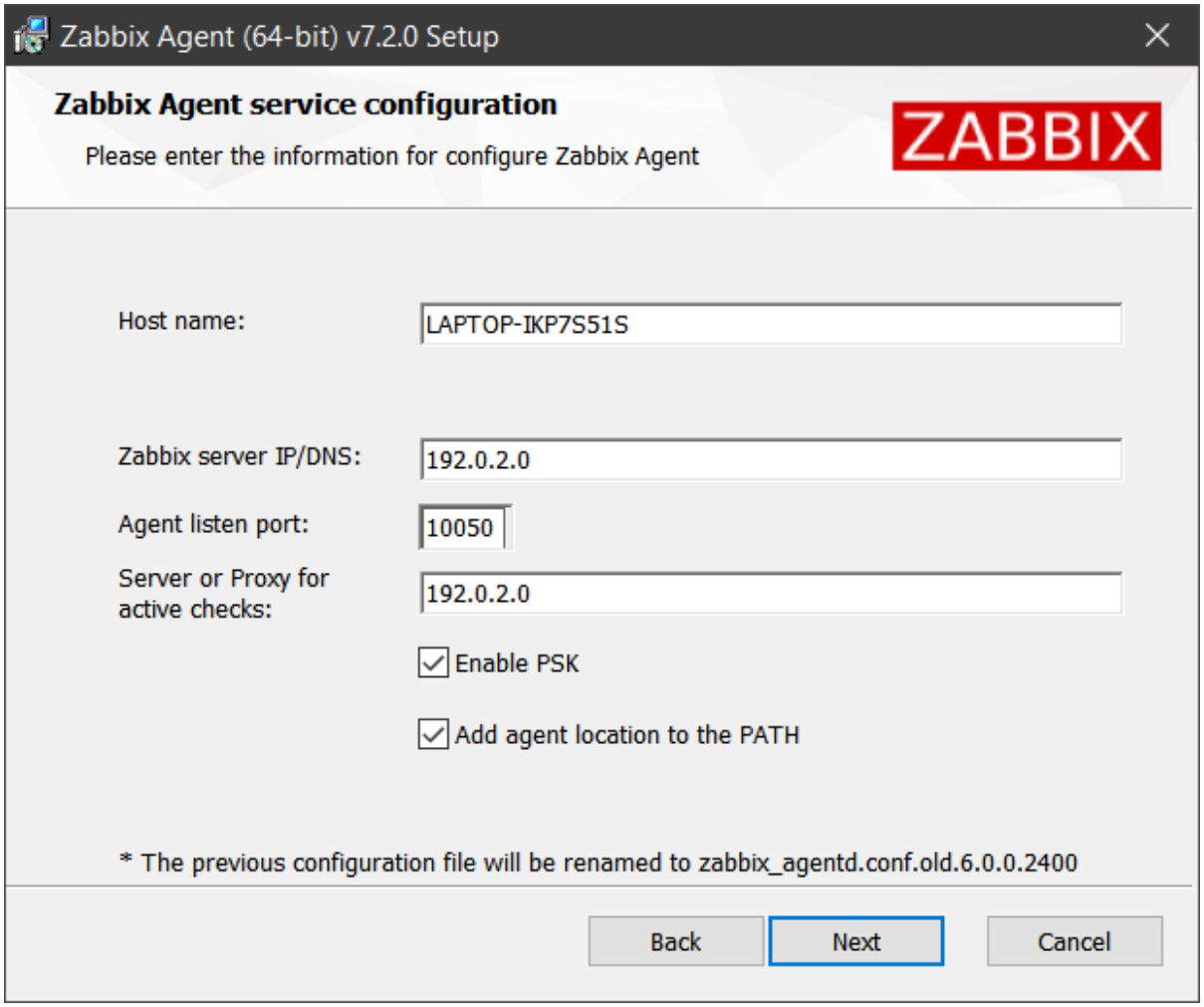
Attention:
Recomenda-se usar o local de instalação padrão fornecido pelo instalador. Usar um local personalizado sem as permissões necessárias pode comprometer a segurança da instalação.



4. Configure os seguintes parâmetros. Seus valores serão definidos no arquivo de configuração do Zabbix agent:

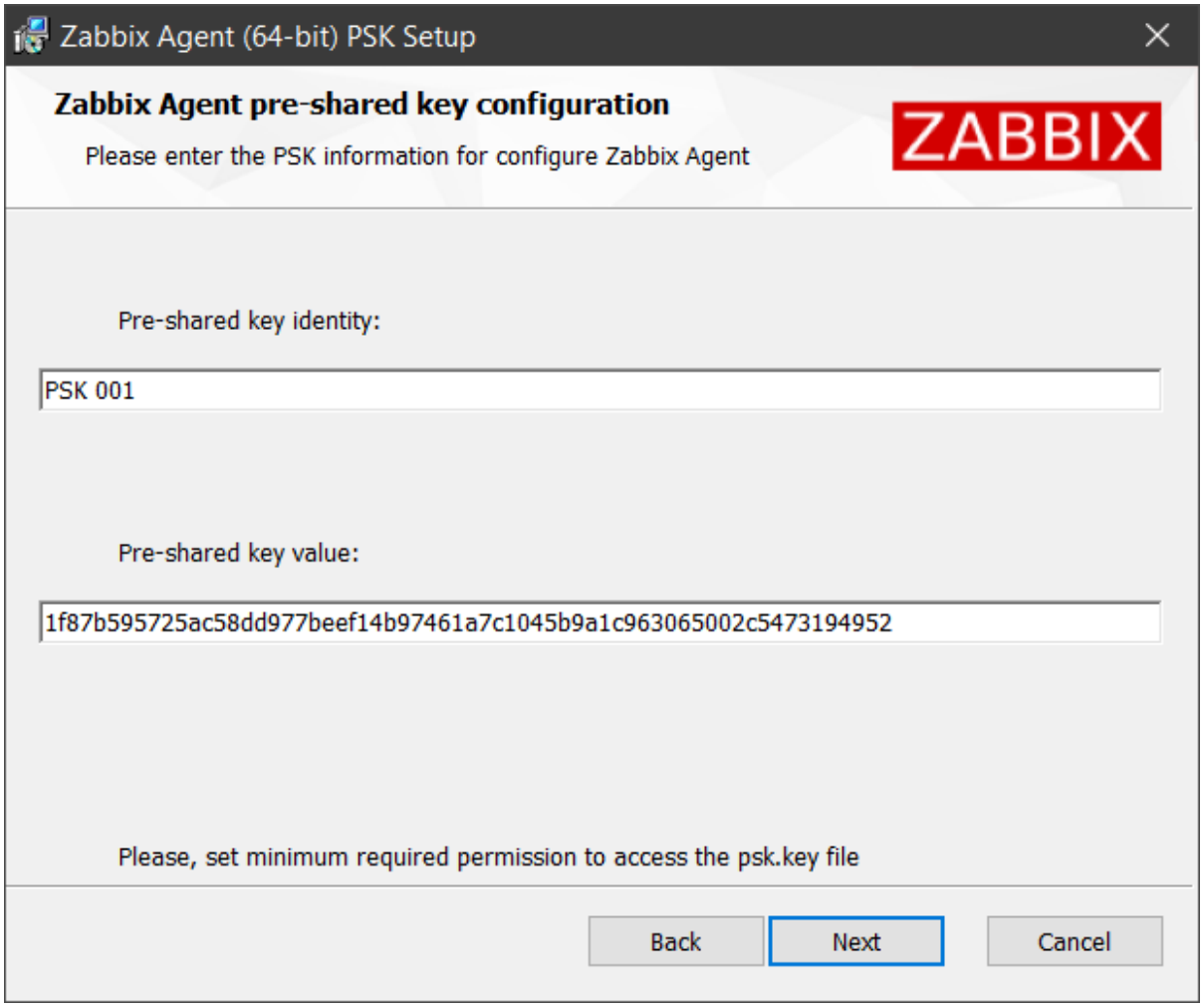
Parâmetro	Descrição
Host name	O nome do host da máquina onde o Zabbix agent está sendo instalado. Define o parâmetro Hostname.
Zabbix server IP/DNS	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de Zabbix servers ou Zabbix proxies. Este parâmetro é **obrigatório**. 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.

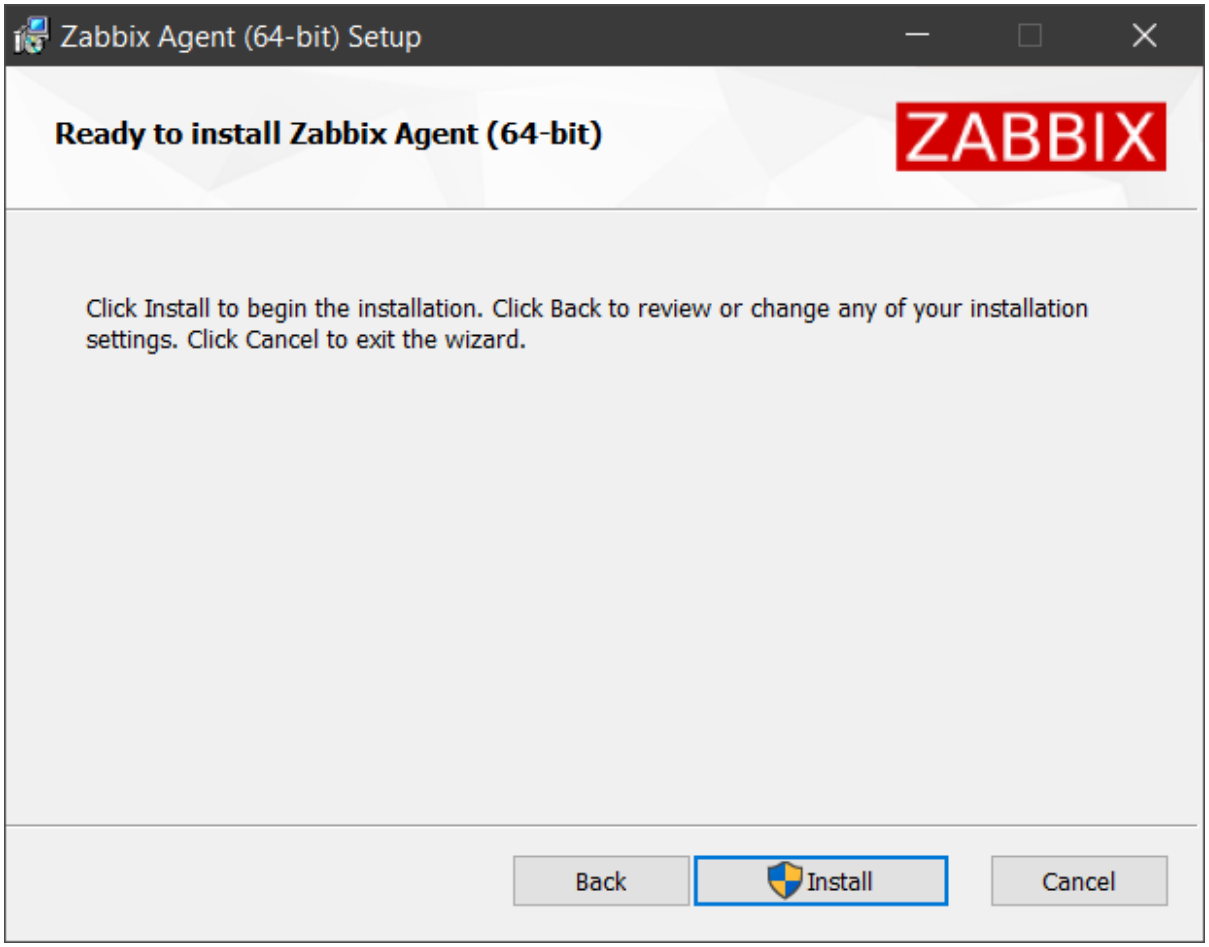


6. Clique em *Install* para iniciar a instalação.

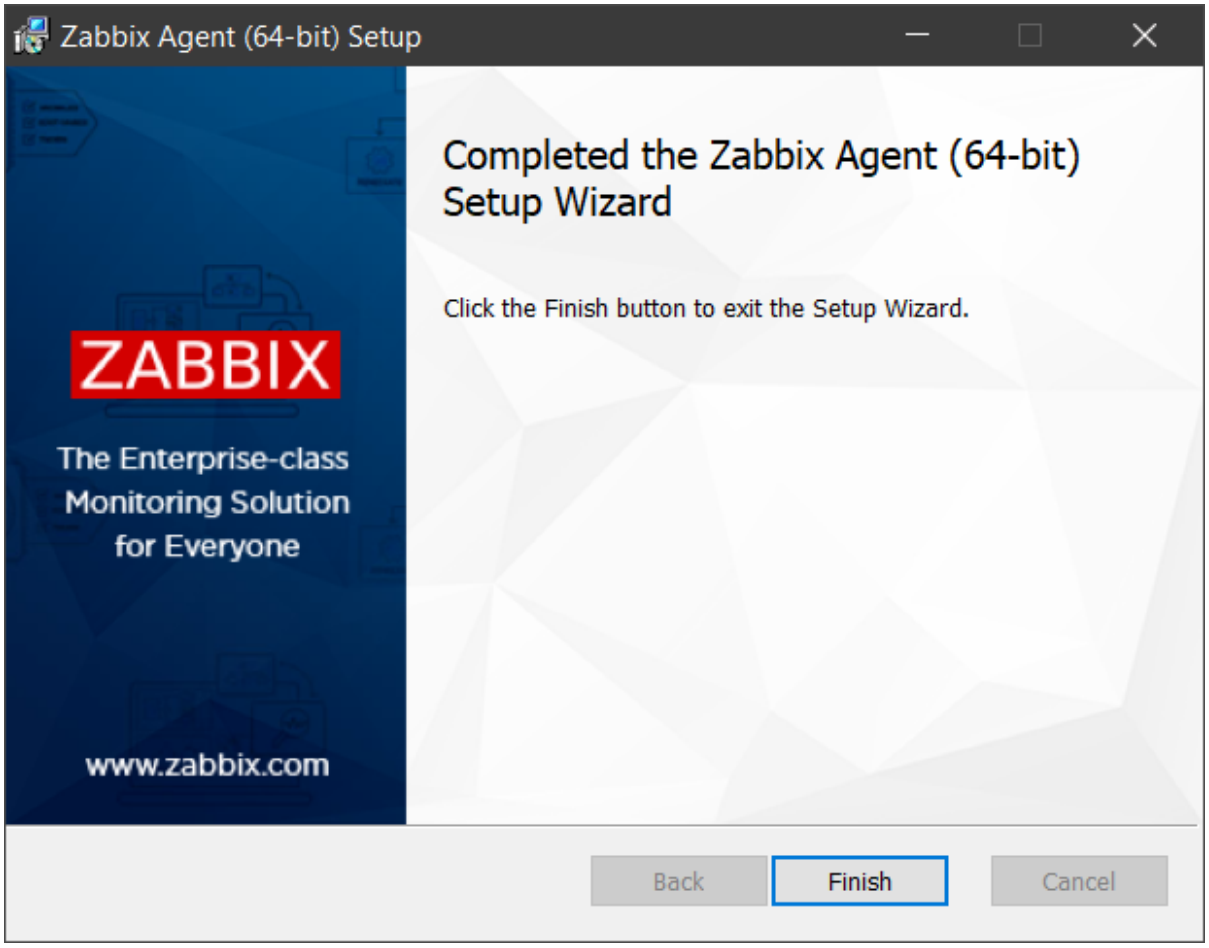
Todos os componentes Zabbix selecionados e o arquivo de configuração do Zabbix agent serão instalados no local especificado (padrão: `C:\Program Files\Zabbix Agent`). O mesmo se aplica ao Zabbix agent 2, exceto que arquivos de configuração adicionais para seus plugins embutidos serão instalados na subpasta `zabbix_agent2.d\plugins.d`.

Além disso, 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 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-8.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Este método permite instalações não assistidas e configurações personalizadas usando parâmetros.

Parâmetros suportados

Os pacotes do instalador 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, veja 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, veja 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[type C:\Windows\System32\drivers\etc\hosts];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"`
ENABLEPATH	Use `ENABLEPATH=1` para adicionar a localização do Zabbix agent à variável PATH do sistema.
ENABLEPERSISTENTBUFFER	Somente para o Zabbix agent 2. Habilita o uso de armazenamento local persistente para itens ativos.
HOSTINTERFACE	Parâmetro opcional que define a interface do host.
HOSTMETADATA	Parâmetro opcional que define os metadados do host.
HOSTMETADATAITEM	Parâmetro opcional que define um item usado para obter os metadados do host.
HOSTNAME	Parâmetro opcional que define o nome do host.
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 escritos 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 para o Zabbix agent 2. O arquivo onde o Zabbix agent 2 deve manter o banco de dados SQLite.
PERSISTENTBUFFERPERIOD	Somente para o 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**.
SERVERACTIVE	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas.

Parâmetro	Descrição
SKIP	Use `SKIP=fw` para impedir que o instalador MSI adicione uma regra de exceção no Firewall do Windows para o Zabbix agent.
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 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** - desativa o serviço para que não possa ser iniciado por um usuário ou aplicativo. Exemplo: `STARTUPTYPE=disabled`
STATUSPORT	Somente para o 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 recebidas 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 o(s) certificado(s) da(s) CA(s) de nível superior para verificação do certificado do par.
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 forem 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 forem 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.

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

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-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### 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-8.0.1-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

Os plugins carregáveis do Zabbix agent 2 podem ser instalados no Windows usando pacotes de instalador MSI de 64 bits, disponíveis para download.

Os requisitos mínimos do sistema operacional para instalação via MSI são Windows 10 (64 bits) ou Windows Server 2016.

Os pacotes incluem:

- Plugin MongoDB
- Plugin PostgreSQL
- Plugin MSSQL

Os pacotes não incluem o plugin Ember+, que atualmente só está disponível para ser compilado a partir do código-fonte (tanto para Unix quanto para Windows).

> **Attention:**
> Antes de instalar um plugin, verifique seu arquivo README. Ele pode conter requisitos adicionais e instruções de instalação específicas para o plugin.

Assim como o Zabbix agent/agent2, os plugins carregáveis podem ser instalados usando o Assistente de Configuração ou a linha de comando.

Instalação pelo Assistente de Instalação

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 (plugin MongoDB, plugin PostgreSQL, plugin MSSQL) 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 Instalação.

Instalação pela linha de comando

Os plugins carregáveis do Zabbix agent 2 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-8.0.0-windows-amd64.msi" /qn+
```

Os pacotes do instalador MSI dos plugins carregáveis do Zabbix agent 2 suportam os seguintes parâmetros.

Parâmetro	Descrição
ADDDEFAULT	Uma lista de componentes separados por vírgula para instalar em sua configuração padrão. Para mais informações, consulte propriedade ADDDEFAULT. Valores possíveis: `MongodbPlugin`, `PostgresqlPlugin`, `MssqlPlugin`, `ALL` Exemplo: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Uma lista de componentes separados por vírgula para instalar localmente. Para mais informações, consulte propriedade ADDLOCAL. Valores possíveis: `MongodbPlugin`, `PostgresqlPlugin`, `MssqlPlugin`, `ALL` 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 no 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 pela linha de comando, por exemplo:

```
sudo installer -pkg zabbix_agent-8.0.0-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.

Solução de problemas e remoção do agent

Esta seção lista alguns comandos úteis que podem ser usados para solucionar problemas e remover a instalação do agent Zabbix.

Veja se o agent Zabbix está em execução:

```
ps aux | grep zabbix_agentd
```

Veja se o agent 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 a / inicial não é exibida 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 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 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 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) - $basearch
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/rhel/8/$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 "Zabbix unstable repository".

```
#### Zabbix unstable repository
deb https://repo.zabbix.com/zabbix/8.0/unstable/debian bullseye main
deb-src https://repo.zabbix.com/zabbix/8.0/unstable/debian bullseye main
```

SUSE

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

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 multi-container 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'.
Como alternativa, 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 base. 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 mais recente e estável 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
Interface web do Zabbix	
baseada no servidor web Apache2 com suporte a MySQL	zabbix/zabbix-web-apache-mysql
baseada no servidor web Apache2 com suporte a PostgreSQL	zabbix/zabbix-web-apache-pgsql
baseada no servidor web Nginx com suporte a MySQL	zabbix/zabbix-web-nginx-mysql
baseada 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 gateway	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	A versão de desenvolvimento mais recente (nightly build) 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 minor mais recente de um componente Zabbix de uma versão major específica e sistema operacional. **<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 major do Zabbix (ex.: *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	A versão minor mais recente de um componente Zabbix de uma versão major específica e sistema operacional. **<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 major do Zabbix (ex.: *6.0*, *7.4*, *8.0*). * - a versão minor do Zabbix	zabbix-agent:alpine-8.0.1

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.

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.

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.
`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 visível da instalação do Zabbix no canto superior direito 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 a 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 verificações externas. 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 SSL do servidor. 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 traps SNMP 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 proxy	
/usr/lib/zabbix/externalscripts	O volume é usado por verificações externas. 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 SSL do servidor. 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 traps SNMP 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
Interface web do Zabbix baseada no servidor web Apache2	
/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
Interface web do Zabbix baseada no servidor web Nginx	
/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 `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 traps SNMP 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 servidor MySQL:

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

3. 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-8.0-latest
```

4. Inicie a instância do Zabbix server e vincule a instância ao servidor MySQL criado:

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

5. Inicie a interface web do Zabbix e vincule a instância ao servidor MySQL e ao Zabbix server criados:

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

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 servidor PostgreSQL:

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

3. Inicie a instância do Zabbix snmptraps:

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

4. Inicie a instância do Zabbix server e vincule a instância ao servidor PostgreSQL criado:

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

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

5. Inicie a interface web do Zabbix e vincule a instância ao servidor PostgreSQL e ao Zabbix server criados:

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

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 8.4:

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

4. 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 Alternativamente, 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 8.0
```

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`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Alpine Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_alpine_mysql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Alpine Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_alpine_pgsql_latest.yaml`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Alpine Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_alpine_pgsql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Alpine Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ol_mysql_latest.yaml`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Oracle Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_ol_mysql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Oracle Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_ol_pgsql_latest.yaml`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Oracle Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ol_pgsql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Oracle Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Ubuntu 22.04 com suporte ao banco de dados MySQL.
`docker-compose_v3_ubuntu_mysql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Ubuntu 22.04 com suporte ao banco de dados MySQL.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	Os arquivos baixam e executam a versão mais recente dos componentes do Zabbix 8.0 no Ubuntu 22.04 com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	Os arquivos constroem localmente a versão mais recente do Zabbix 8.0 e executam os componentes do Zabbix no Ubuntu 22.04 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 em 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. Consulte variáveis de ambiente para obter detalhes sobre a nomenclatura das variáveis e a seleção disponível.

Exemplos

Exemplo 1

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

O comando fará o download das imagens mais recentes do Zabbix 8.0 para cada componente do Zabbix e as executará 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 8.0
docker compose -f ./docker-compose_v3_ubuntu_mysql_local.yaml up -d
```

O comando fará o download da imagem base Ubuntu 24.04 (noble), depois irá construir localmente os componentes do Zabbix 8.0 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.

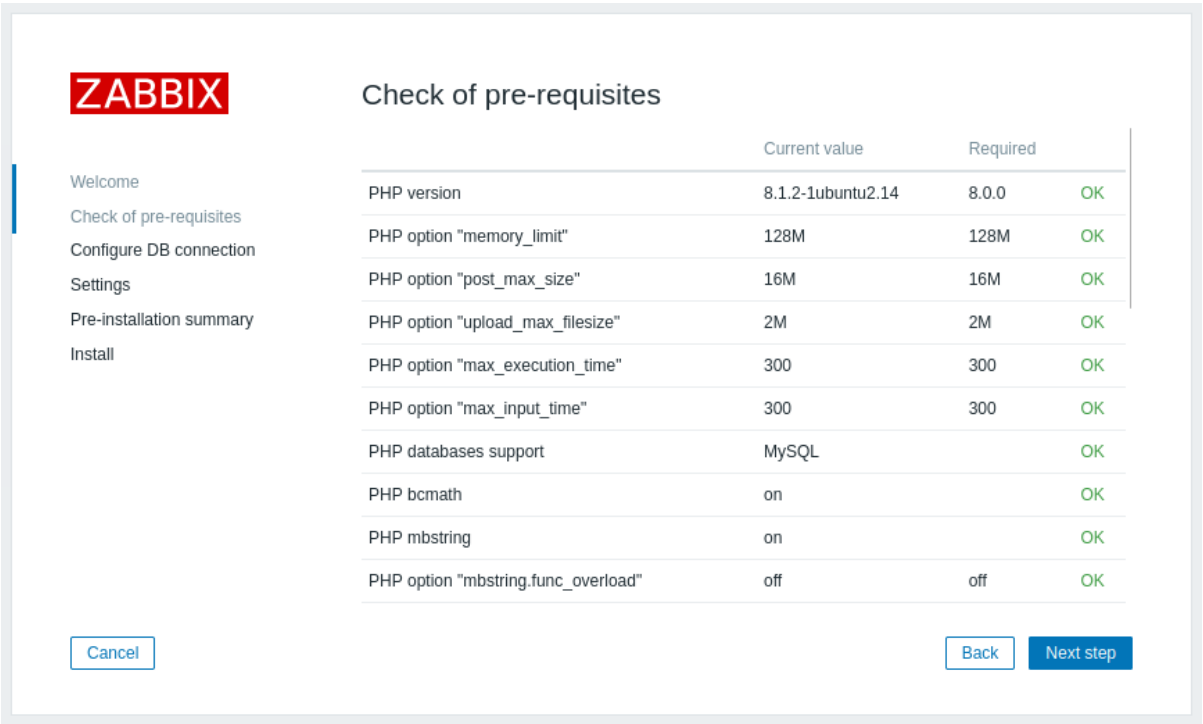
Use o menu suspenso *Idioma padrão* para alterar o idioma padrão do sistema e continuar o processo de instalação no idioma selecionado (opcional). Para mais informações, consulte Instalação de idiomas adicionais no frontend.

Observe que definir o idioma como *English (en_US)* também habilitará 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 sejam atendidos.



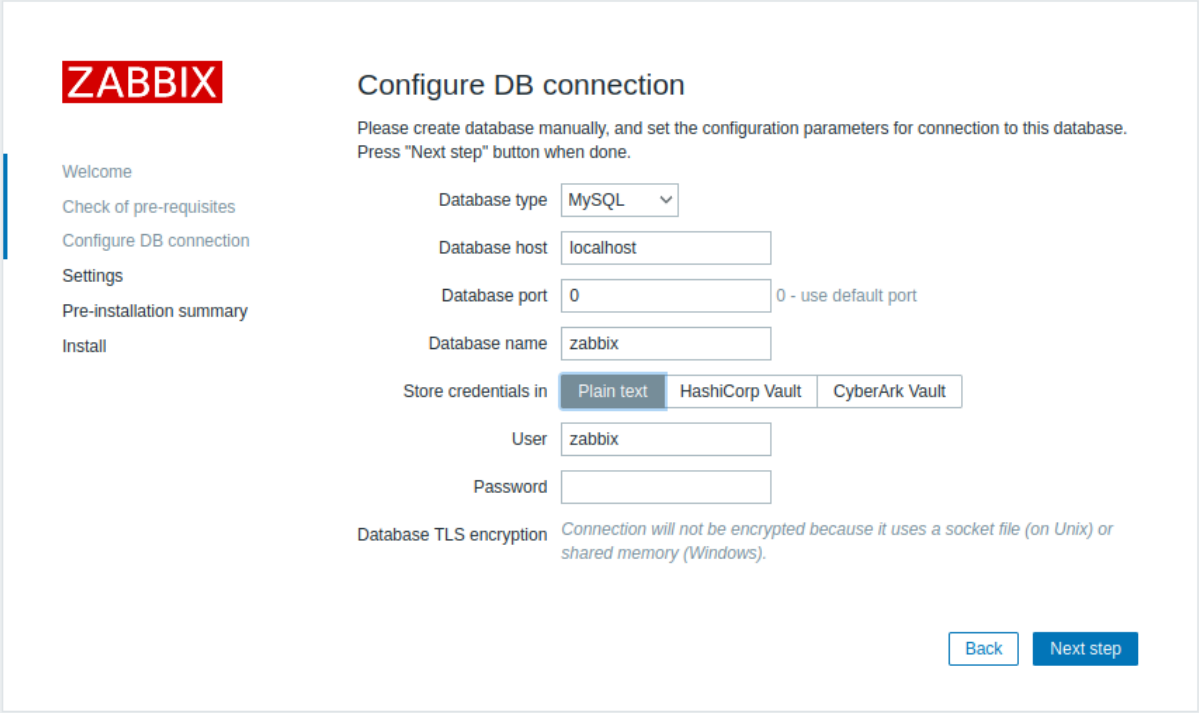
Pré-requisito	Valor mínimo	Descrição
Versão do PHP	8.0.0	
Opção PHP "memory_limit"	128MB	No php.ini: memory_limit = 128M
Opção PHP "post_max_size"	16MB	No php.ini: post_max_size = 16M
Opção PHP "upload_max_filesize"	2MB	No php.ini: upload_max_filesize = 2M
Opção PHP "max_execution_time"	300 segundos	No php.ini: max_execution_time = 300 (os valores 0 e -1 também são permitidos)
Opção PHP "max_input_time"	300 segundos	No php.ini: max_input_time = 300 (os valores 0 e -1 também são permitidos)

Pré-requisito	Valor mínimo	Descrição
Suporte a bancos de dados PHP	Um dos: MySQL, PostgreSQL	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 *Warning* em vermelho, e o processo de instalação pode prosseguir mesmo que eles não sejam atendidos.
PHP bcmath	deve estar habilitado	
PHP mbstring	deve estar habilitado	
Opção PHP "mbstring.func_overload"	deve estar desabilitado	No php.ini: mbstring.func_overload = 0
Opção PHP "session.auto_start"	deve estar desabilitado	No php.ini: session.auto_start = 0
Opção 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 instalação do Zabbix pode não conseguir continuar.

Configurar conexão com o banco de dados

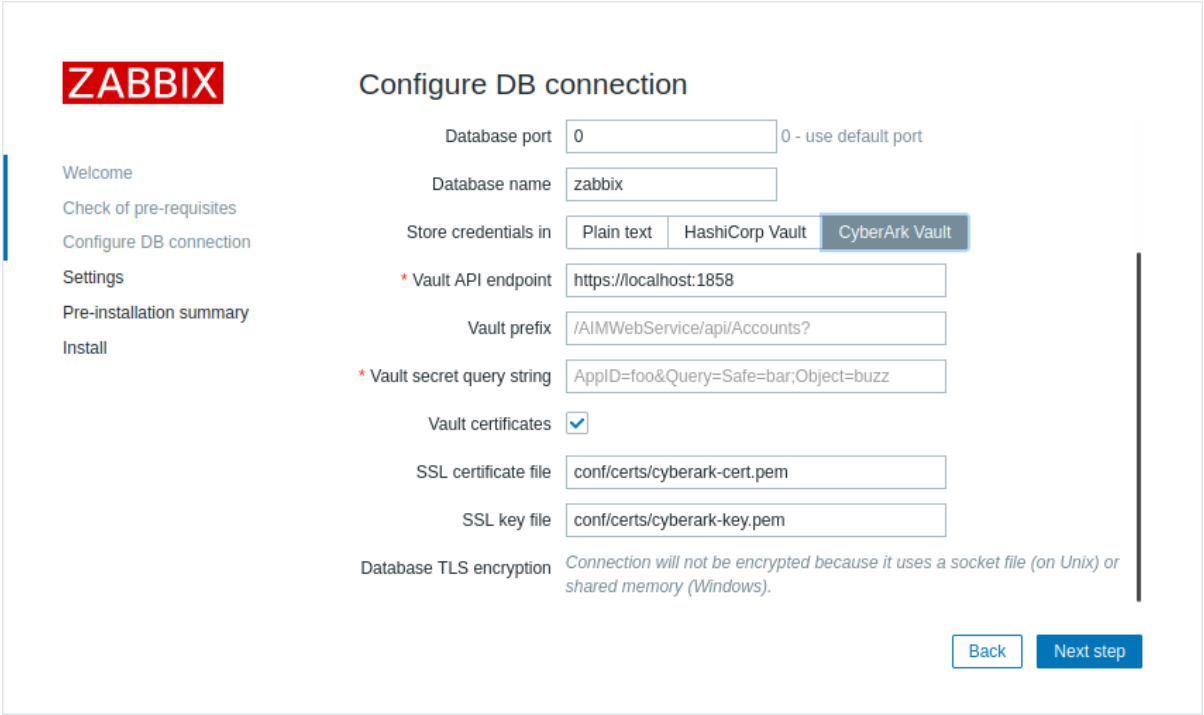
Insira os detalhes para conectar ao banco de dados. O banco de dados do Zabbix deve já estar criado.



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.

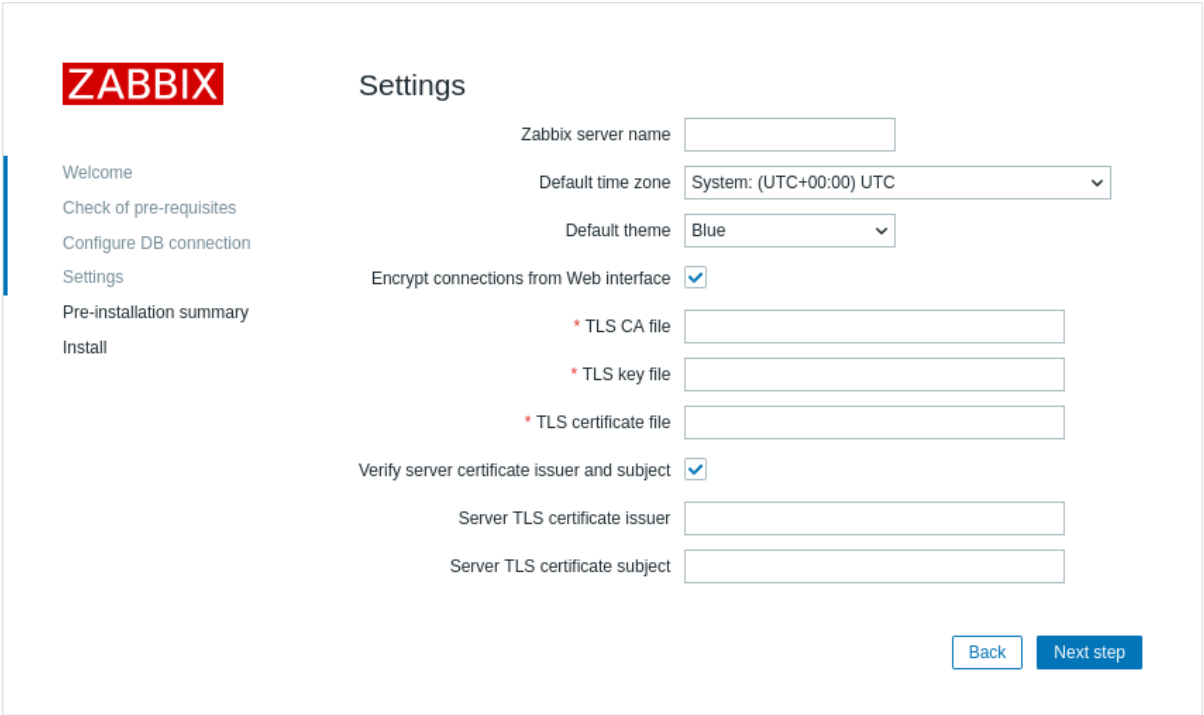


Configurações

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.

Se a opção *Criptografar conexões da interface Web* estiver marcada, então campos adicionais para configurar a conexão TLS entre o Zabbix server e o frontend aparecerão no formulário.



Resumo da pré-instalação

Revise um resumo das configurações.

A subpágina mostrará os dados se a configuração TLS tiver sido adicionada.

Instalar

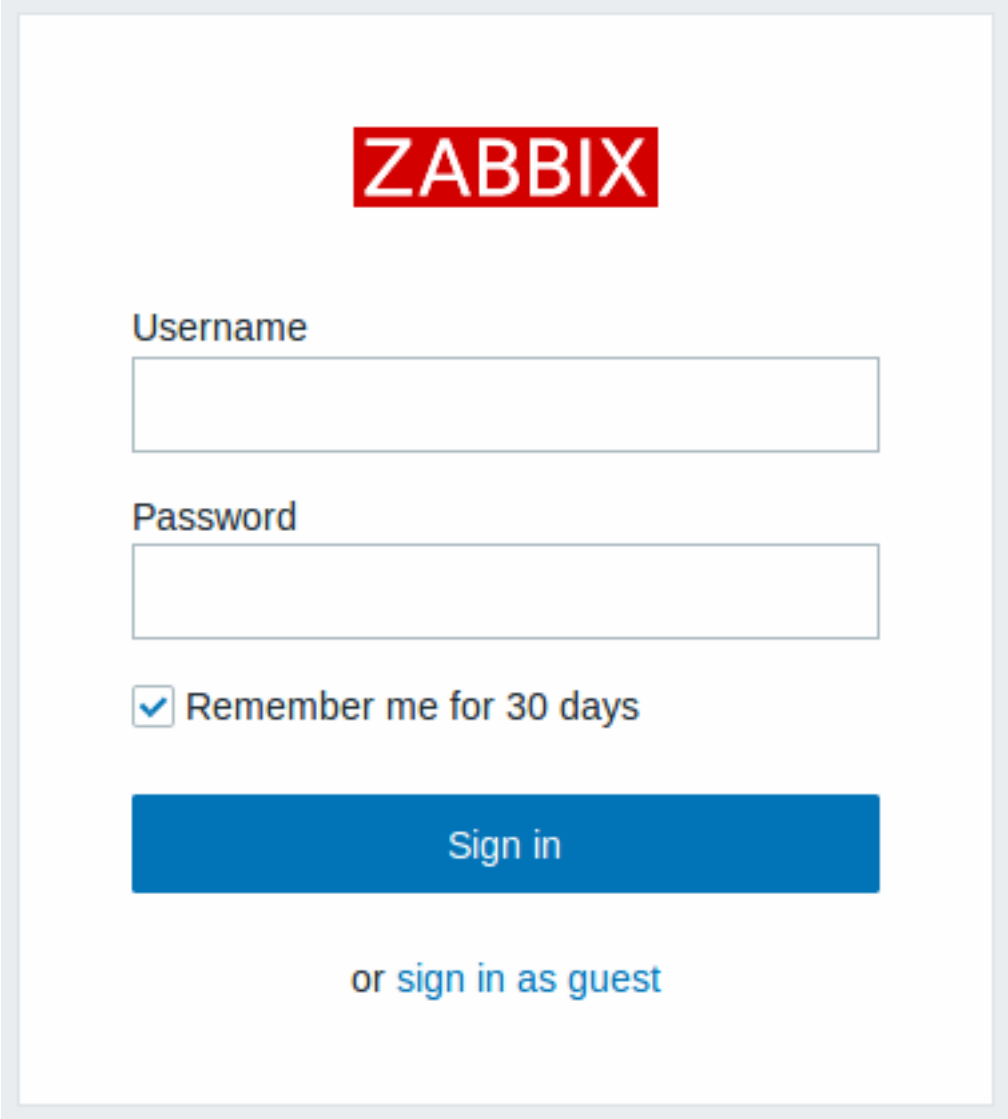
Se estiver instalando o Zabbix a partir do código-fonte, baixe o arquivo de configuração e coloque-o em conf/ no subdiretório de documentos HTML do servidor web onde você copiou os arquivos PHP do Zabbix.

Note:
Se o usuário do servidor web tiver permissão de gravação no diretório conf/, o arquivo de configuração será salvo automaticamente e será possível prosseguir para a próxima etapa imediatamente.

Finalize a instalação.

Login

O frontend do Zabbix está pronto! O nome de usuário padrão é **Admin**, senha **zabbix**.



Prossiga para primeiros passos com o Zabbix.

7 Procedimento de atualização

Visão geral

Esta seção fornece informações de atualização para o Zabbix **8.0**:

- 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 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 desativada, 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 novos 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 Zabbix proxy for iniciado pela primeira vez e o arquivo de banco de dados SQLite estiver ausente, o proxy o criará automaticamente.

Observe que se o Zabbix proxy usar SQLite3 e, ao iniciar, 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 históricos armazenados no arquivo de banco de dados SQLite serão perdidos. Se a versão do Zabbix proxy 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 8.0 pode levar muito tempo.

A atualização direta para o Zabbix 8.0.x é suportada a partir das versões **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 atualizada do Zabbix.

As seguintes notas de atualização estão disponíveis:

Atualizar de	Leia as notas de atualização completas	Alterações mais importantes entre as versões
7.4.x	Para: Zabbix 8.0	Versões mínimas exigidas do banco de dados aumentadas.
7.2.x	Para: Zabbix 7.4 Zabbix 8.0	Suporte à biblioteca PCRE removido.
7.0.x	Para: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Suporte ao Oracle DB removido.
6.4.x	Para: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida 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. Oracle DB obsoleto. Tipo de valor numérico antigo (float) removido.
6.2.x	Para: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida 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.

Atualizar de	Leia as notas de atualização completas	Alterações mais importantes entre as versões
6.0.x LTS	Para: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versões mínimas exigidas do banco de dados aumentadas. Server/proxy não iniciará se o banco de dados estiver desatualizado. Registros de 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versões mínimas exigidas do banco 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida do PHP aumentada de 7.2.0 para 7.2.5. Algoritmo de hash de senha alterado de MD5 para bcrypt.
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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Suporte ao IBM DB2 removido. Versão mínima exigida do PHP aumentada de 5.4.0 para 7.2.0. Versões mínimas exigidas do banco 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Tipos de mídia Jabber, Ez Texting removidos.

Atualizar de	Leia as notas de atualização completas	Alterações mais importantes entre as versões
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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Bibliotecas 'libpthread' e 'zlib' agora obrigatórias. Suporte ao protocolo de texto simples 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	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 sinalizador 'nowait', bem como scripts executados pelo Zabbix server. O Zabbix Java gateway deve ser atualizado para suportar a nova funcionalidade.
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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	A atualização do banco de dados pode ser lenta, dependendo do tamanho da tabela de histórico.

Atualizar de	Leia as notas de atualização completas	Alterações mais importantes entre as versões
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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida 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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	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 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Versão mínima exigida do PHP aumentada de 5.1.6 para 5.3.0. Banco de dados MySQL sensível a maiúsculas e minúsculas 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' exigida 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 **7.4**.x para o Zabbix **8.0**.x usando fontes

oficiais do Zabbix.

Warning:
Antes da atualização, certifique-se de ler as **notas de atualização** relevantes!

Você também pode querer revisar os requisitos para o 8.0.

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 servidor

Pare o servidor Zabbix 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 de energia, 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 servidor

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 Zabbix server irá atualizar o banco de dados automaticamente. Ao iniciar, o Zabbix server 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 Zabbix server 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 Zabbix server. 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 Zabbix server não será iniciado. O Zabbix server também não será iniciado se a versão obrigatória atual do banco de dados for mais recente que a exigida. O Zabbix server 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 server:

- 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

Não há alterações obrigatórias nesta versão nos parâmetros do proxy.

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 irá atualizar o banco de dados automaticamente. A atualização do banco de dados ocorre de forma semelhante ao iniciar o 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 novos binários do 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

Não há alterações obrigatórias nesta versão nos parâmetros do agent nem do agent 2.

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 8.0.x (por exemplo, de 8.0.1 para 8.0.3), 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 é feita no banco de dados.

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 SO

Algumas distribuições de SO (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 SO (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 **7.4.x** para a versão mais recente do Zabbix **8.0.x** usando os pacotes oficiais do Zabbix para Red Hat Enterprise Linux ou seus derivados - AlmaLinux, CentOS Stream, Oracle Linux e Rocky Linux.

Warning:
Antes de atualizar, revise as notas de atualização relevantes e certifique-se de que seu sistema atende aos requisitos para o Zabbix 8.0.

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 8.0.x (por exemplo, de 8.0.1 para 8.0.3), consulte Atualização entre versões menores.

Procedimento de atualização

1 Pare os processos do Zabbix

Pare o Zabbix server para garantir que nenhum novo dado seja inserido no banco de dados:

```
systemctl stop zabbix-server
```

Se estiver atualizando o Zabbix proxy, agent ou agent 2, 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 na 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 10**, execute:

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

No **RHEL 9**, execute:

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/9/noarch/zabbix-release-latest.el9.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 Atualize os 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 do Zabbix atualizados:

```
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 8.0.x (por exemplo, de 8.0.1 para 8.0.3).

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 **7.4.x** para a versão mais recente do Zabbix **8.0.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 atende aos requisitos para o Zabbix 8.0.

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 8.0.x (por exemplo, de 8.0.1 para 8.0.3), consulte Atualização entre versões menores.

Procedimento de atualização

1 Pare 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 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 novos e incluir quaisquer correções de segurança ou bugs recentes.

No **Debian 12**, execute:

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian12_all.deb
```

Note:
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 de pacotes, considere atualizar seu sistema operacional. Para uma lista de componentes incluídos, consulte Pacotes Zabbix.

No **Ubuntu 24.04**, execute:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
```

No **Ubuntu 22.04**, execute:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu22.04_all.deb
```

Note:
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 de pacotes, considere atualizar seu sistema operacional. Para uma lista de componentes incluídos, consulte Pacotes Zabbix.

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 levar a 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 do Zabbix atualizados:

```
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 8.0.x (por exemplo, de 8.0.1 para 8.0.3).

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 **8.0**.x.

Conjuntos separados de instruções estão disponíveis para atualizar imagens de componentes individuais do Zabbix e arquivos Docker compose.

> **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 determinísticas precisarão ser criadas durante a atualização. Se o log binário estiver ativado 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. Nesse 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 8.0 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-8.0-latest
```

`zabbix/zabbix-server-mysql:alpine-8.0-latest` irá baixar a última versão menor lançada do Zabbix server 8.0 com suporte ao 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-8.0-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-8.0-latest
```

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 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 altere para o branch necessário:

```
git pull
git checkout 8.0
```

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.

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 e anteriores, porque nessas versões o formato de linha padrão é 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 ambos os 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 no 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 o 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 ambos os 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 explicitamente `localhost` 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 ambos os 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 do 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.conf`:

```
[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, certifique-se de ter 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 contêiner.

Para mais informações, consulte ZBX-24291.

Chave de assinatura expirada para pacotes RHEL

Ao atualizar o Zabbix no Red Hat Enterprise Linux ou 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 assinaturas de 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 esse tipo de problema, 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 10**, execute:

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

Em seguida, atualize as informações do repositório:

```
dnf update
```

Para mais informações, consulte ZBX-24761.

Timescale DB: alto uso de memória com grande número de partições

As versões do PostgreSQL 9.6-12 usam muita memória ao atualizar tabelas com um grande número de partições. Esse problema se manifesta quando o Zabbix atualiza tendências em sistemas com TimescaleDB se as tendências forem divididas em partes relativamente pequenas (por exemplo, 1 dia). Isso leva a centenas de partes presentes nas tabelas de tendências com as configurações padrão de housekeeping - a condição em que o PostgreSQL provavelmente ficará sem memória.

O problema foi resolvido desde o Zabbix 5.0.1 para novas instalações com TimescaleDB, mas se o TimescaleDB foi configurado com o Zabbix antes disso, consulte ZBX-16347 para as notas de migração.

Timescale DB 2.5.0: a política de compressão pode falhar em tabelas que contêm inteiros

Esse problema se manifesta quando o TimescaleDB 2.5.0/2.5.1 é usado. Ele foi resolvido desde o TimescaleDB 2.5.2.

Para mais informações, consulte TimescaleDB Issue #3773.

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 utilizado.

Possíveis deadlocks com MySQL/MariaDB

Ao executar sob alta carga e com mais de um worker de LLD envolvido, é possível encontrar um deadlock causado por um erro do InnoDB relacionado à estratégia de bloqueio de linhas (veja upstream bug). 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 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 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 de expressões regulares do Go.

Verificações IPMI

As verificações IPMI não funcionarão com o pacote padrão da biblioteca OpenIPMI no Debian anterior ao 9 (stretch) e Ubuntu anterior ao 16.04 (xenial). Para corrigir isso, recompile a biblioteca OpenIPMI com o OpenSSL habilitado, conforme discutido em ZBX-6139.

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 compilados 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 recomendada:

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 de uma consulta SQL (por exemplo, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Caso contrário, o item de monitoramento de 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 detalhes sobre como corrigir essa situação, consulte ZBX-19308.

Monitoramento web e agente HTTP

O servidor Zabbix vaza memória em algumas distribuições Linux devido a um bug upstream quando "SSL verify peer" está habilitado em cenários web ou 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 com a 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, veja 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, 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

Picos em dados SNMP foram observados e 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 alerter no RHEL 7

Foram encontrados casos de falha do processo alerter do Zabbix server no RHEL 7. Consulte ZBX-10461 para 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 um idioma diferente. 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 multi-thread, quando há vários projetos executados pelo mesmo processo do Apache, é possível que o idioma seja alterado em outra thread e isso mude 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

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), as 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 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.

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 reduzido 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, portanto, 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 alternativas 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 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 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 seu servidor web.

Para mais detalhes, veja:

- ZBX-22952
- Apache 2.4 + PHP-FPM e cabeçalhos Authorization
- Diretivas 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 Zabbix server ou proxy. 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 ser carregados corretamente se você atualizou a partir 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 requisição, 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 sobrescrever objetos globais internos a partir do pré-processamento.

Exemplo seguro:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

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 colações 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, 𝓀, 𝔞). Esse problema ocorre devido à forma como a colação 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 colação 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 colação 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 colações, consulte a documentação do MySQL ou a documentação do MariaDB. Para detalhes sobre diferenças de colação, consulte Unicode Character Sets na documentação do MySQL.

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, contate o administrador do Zabbix.". Esse problema afeta instalações usando versões do MariaDB de 10.5.1 até 10.5.9.

Para evitar esse problema, atualize o MariaDB para uma versão posterior à 10.5.9. Para mais detalhes, veja 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 alvo. 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.
- GCC Option Summary sobre opções de compilação (`-std=gnu99`, `-g`, `-O0`, etc.).
- Gperftools Heap Profiler documentação sobre variáveis de ambiente para profiling do tcmalloc.

Replicação de Grupo MySQL 8.0 no modo multi-primário

Ao usar a Replicação de Grupo MySQL 8.0 no modo multi-primário, 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 acionado 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.

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 da versão 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 de funções 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:

```
\\\
```

```
'$,1m:now,"regexp","\\\
```

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$232_BACKSLASHES_A})
```

9 Alterações em templates

Esta página lista todas as alterações nos templates padrão que são fornecidos com o Zabbix.

Atualizar para a versão mais recente do Zabbix não atualizará automaticamente os templates utilizados. Recomenda-se modificar os templates em instalações existentes baixando os templates mais recentes do repositório Git do Zabbix e importando manualmente para o Zabbix.

Se já existirem templates com os mesmos nomes, as opções *Excluir ausentes* devem ser marcadas ao importar para obter uma importação limpa. Dessa forma, os itens antigos que não estão mais no template atualizado serão removidos (observe que isso significará perder o histórico desses itens antigos).

Note:
Informamos que, desde o Zabbix 6.0, todos os templates seguem um formato atualizado, o que pode impactar a importação de templates anteriores à 6.0. Para mais informações, consulte Alterações em templates no 6.0.

Alterações na 8.0.0

10 Notas de atualização para 8.0.0

Estas notas são para atualização do Zabbix 7.4.x para o Zabbix 8.0.0.

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 8.0.0;
- Atualizando cluster HA para instruções sobre atualização de servers em um cluster de **alta disponibilidade** (HA).

Mudanças incompatíveis Versões do banco de dados

As versões mínimas de banco de dados necessárias foram atualizadas:

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

Plugins

O plugin Ceph para o Zabbix agent 2 agora é um plugin carregável e requer etapas adicionais de instalação. Consulte o readme do plugin Ceph para obter detalhes.

Outros

11 Notas de atualização para 8.0.x

Esta página fornece notas de atualização coletivas para versões secundárias da versão principal do Zabbix.

Veja também as notas de atualização da versão principal.

Notas de atualização para 8.0.1 Esta versão ainda não foi lançada.

5 Início rápido

Use a barra lateral para acessar o conteúdo da seção Início rápido.

1 Login e configuração do usuário

Visão geral

Nesta seção, você aprenderá como fazer login e configurar um usuário do 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 sobrescrito, 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 irá expirar de acordo com o intervalo de *Logout automático* configurado.

Proteção contra ataques de força bruta

No caso de cinco tentativas consecutivas de login sem sucesso, a interface do Zabbix irá pausar por 30 segundos para evitar ataques de força bruta e de dicionário.

O endereço IP de uma tentativa de login sem sucesso será exibido após um login bem-sucedido.

Adicionando usuário

Para visualizar informações sobre usuários, vá em *Usuários* > *Usuários* no menu vertical da barra lateral.

Para adicionar um novo usuário, selecione *Criar usuário* no canto superior direito.

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. Para detalhes sobre os campos deste formulário de configuração, consulte a página Configurações do usuário.

Por padrão, novos usuários não têm mídias (métodos de entrega de notificação) definidas. Para criar uma, vá até a aba 'Mídia' e clique em *Adicionar*.

Na janela pop-up, insira o endereço de e-mail do 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 habilitados por enquanto.

Clique em *Adicionar* para salvar a mídia e, em seguida, vá para a aba Permissões.

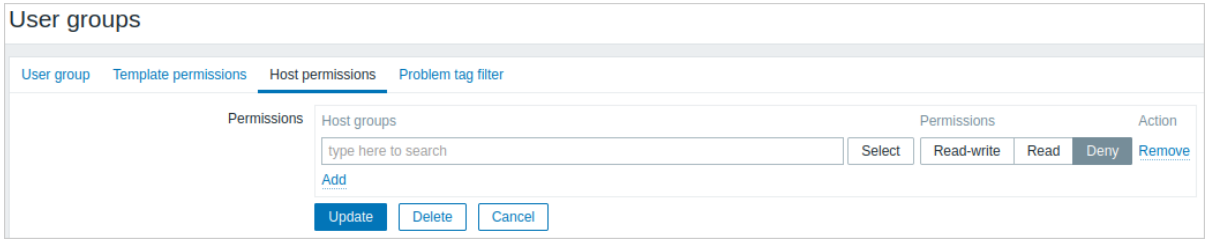
A aba Permissões possui um campo obrigatório *Função*. A função determina quais elementos do frontend o usuário pode visualizar e quais ações ele pode executar. Pressione Selecionar e escolha uma das funções da lista. Por exemplo, selecione *Função de administrador* para permitir acesso a todas as seções 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:

user_permissions.png

Clique em *Adicionar* no formulário de propriedades do usuário para salvar o usuário. O novo usuário aparecerá 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 *Grupos de usuários*, 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 host seja exibido:



Em seguida, clique em *Selecionar* ao lado do campo para ver a lista dos 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 grupos de templates. As etapas são idênticas à atribuição de permissões a grupos de hosts. Uma visão geral dos templates está disponível na seção Novo template deste Guia Rápido.

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

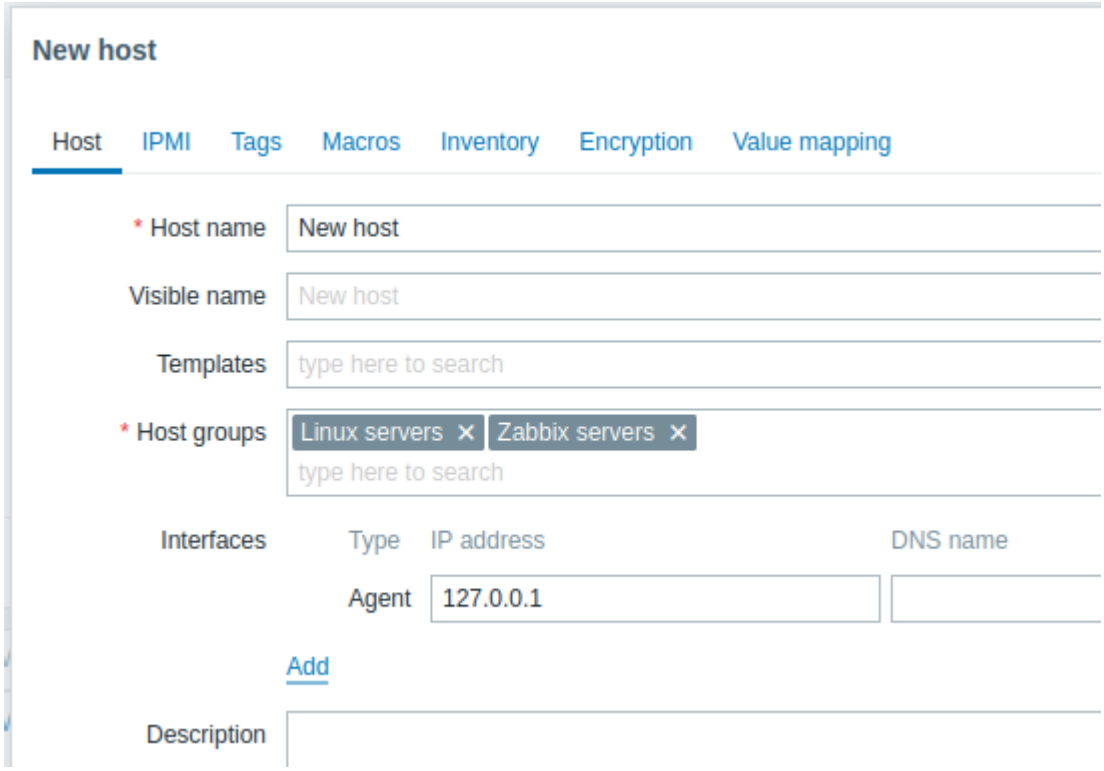
Nesta seção, você aprenderá como configurar um novo host.

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. Pode ser um servidor físico, um switch de rede, uma máquina virtual ou algum aplicativo.

Adicionando host

As informações sobre hosts configurados no Zabbix estão disponíveis nas seções de menu *Coleta de dados* > *Hosts* e *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.

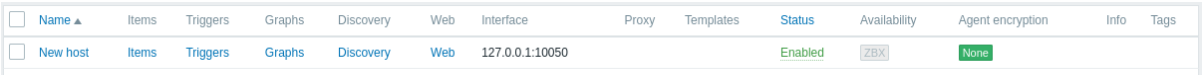
Note:
Todas as permissões de acesso são atribuídas a grupos de hosts, não a hosts individuais. Por isso, um host deve pertencer a pelo menos um grupo.

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 nos dois locais.

Outras opções podem ser mantidas 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:

- ZBX - o status do host não foi estabelecido; nenhuma verificação de métrica foi realizada ainda
- ZBX - o host está disponível, uma verificação de métrica foi bem-sucedida
- ZBX - 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

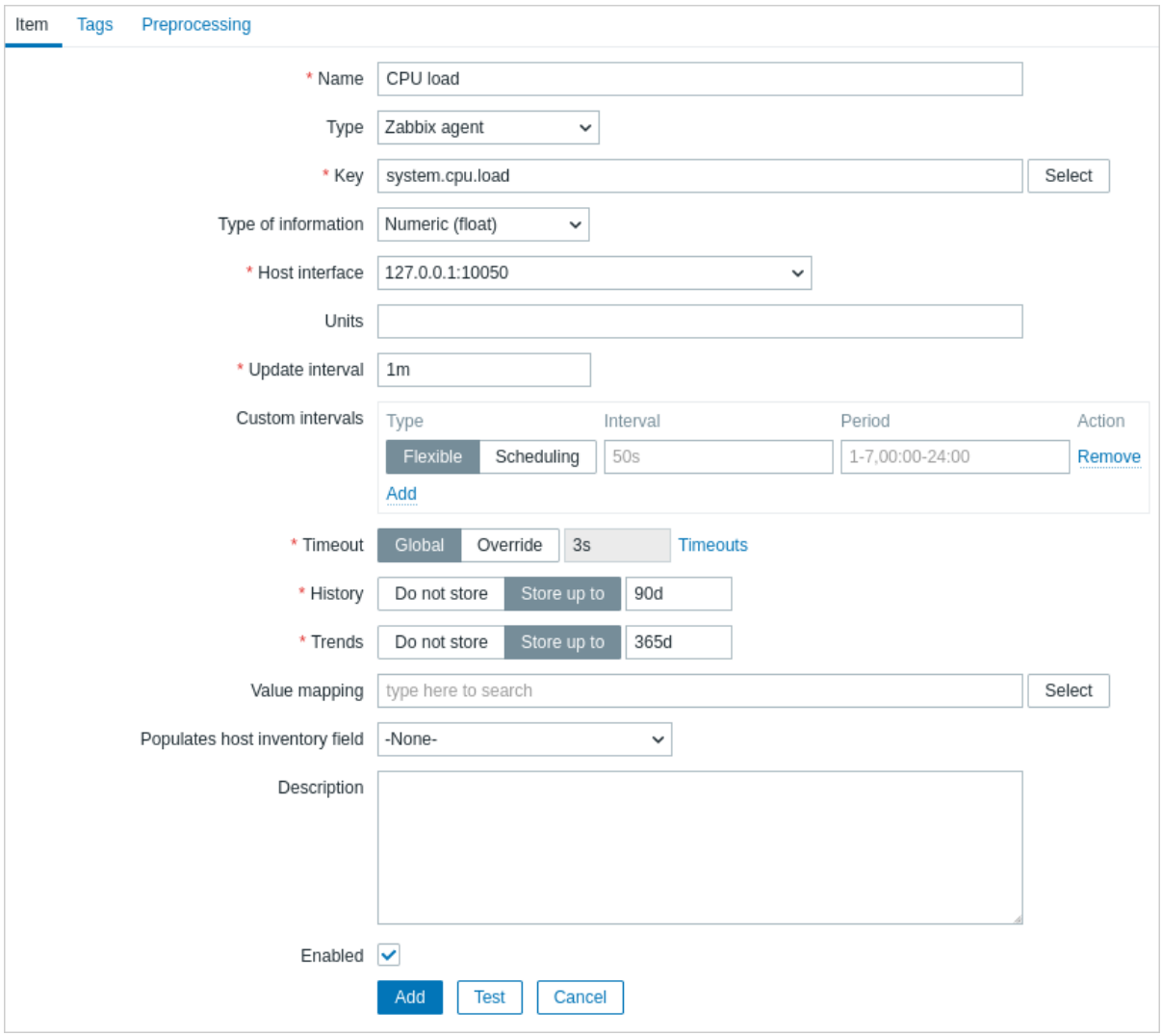
Nesta seção, você aprenderá como configurar um item.

Itens são a base da coleta de dados no Zabbix. Sem itens, não há dados - porque apenas um item define uma métrica única ou o tipo de dado a ser coletado de um host.

Adicionando item

Todos os items 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 *Items* na linha do "Novo host" e, em seguida, clique em *Criar item*. Isso apresentará um formulário de definição 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

- Insira *Carga da CPU* como valor. Este será o nome do item exibido em listas e outros locais.

Chave

- Insira 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 pré-definidas que acompanham o agent Zabbix.

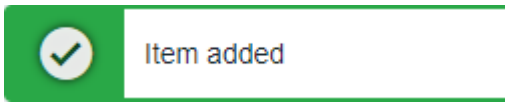
Tipo de informação

- 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.

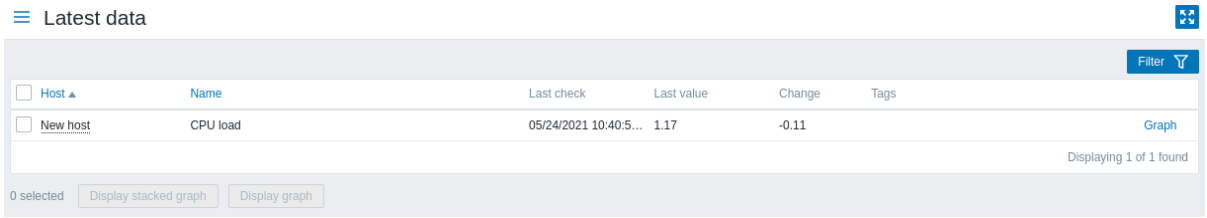
Outras opções podem ser mantidas com seus valores padrão por enquanto.

Quando terminar, clique em *Adicionar*. O novo item deve aparecer na lista de items 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 items 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.

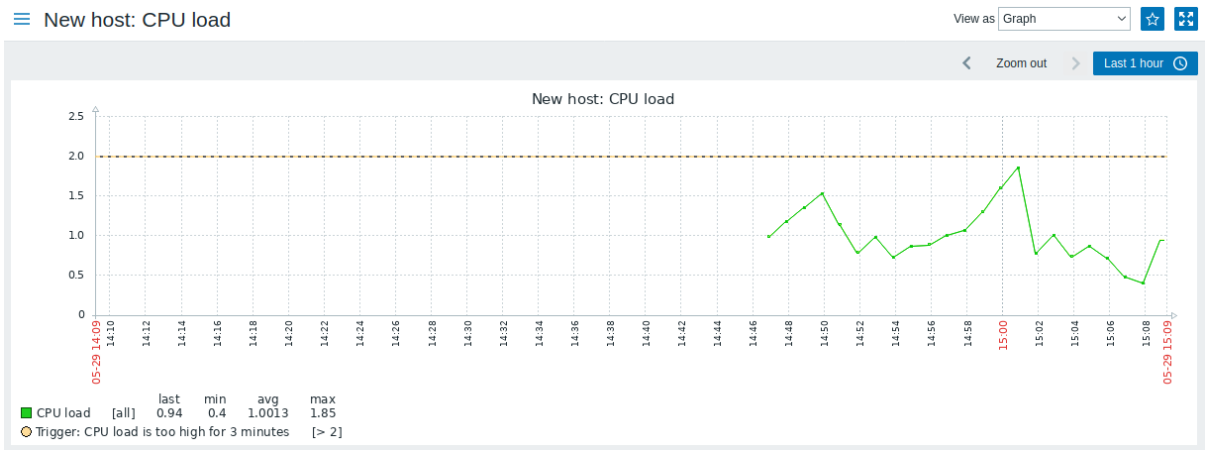
Se você não vir informações sobre o item como na captura de tela, certifique-se de que:

- 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 é '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, pode ser hora de ver algo visual. Gráficos simples estão disponíveis para qualquer item numérico monitorado sem nenhuma configuração adicional. Esses gráficos são gerados em tempo de execução.

Para visualizar o gráfico, vá em *Monitoramento* > *Últimos dados* e clique no link 'Gráfico' ao lado do item.



4 Novo trigger

Visão geral

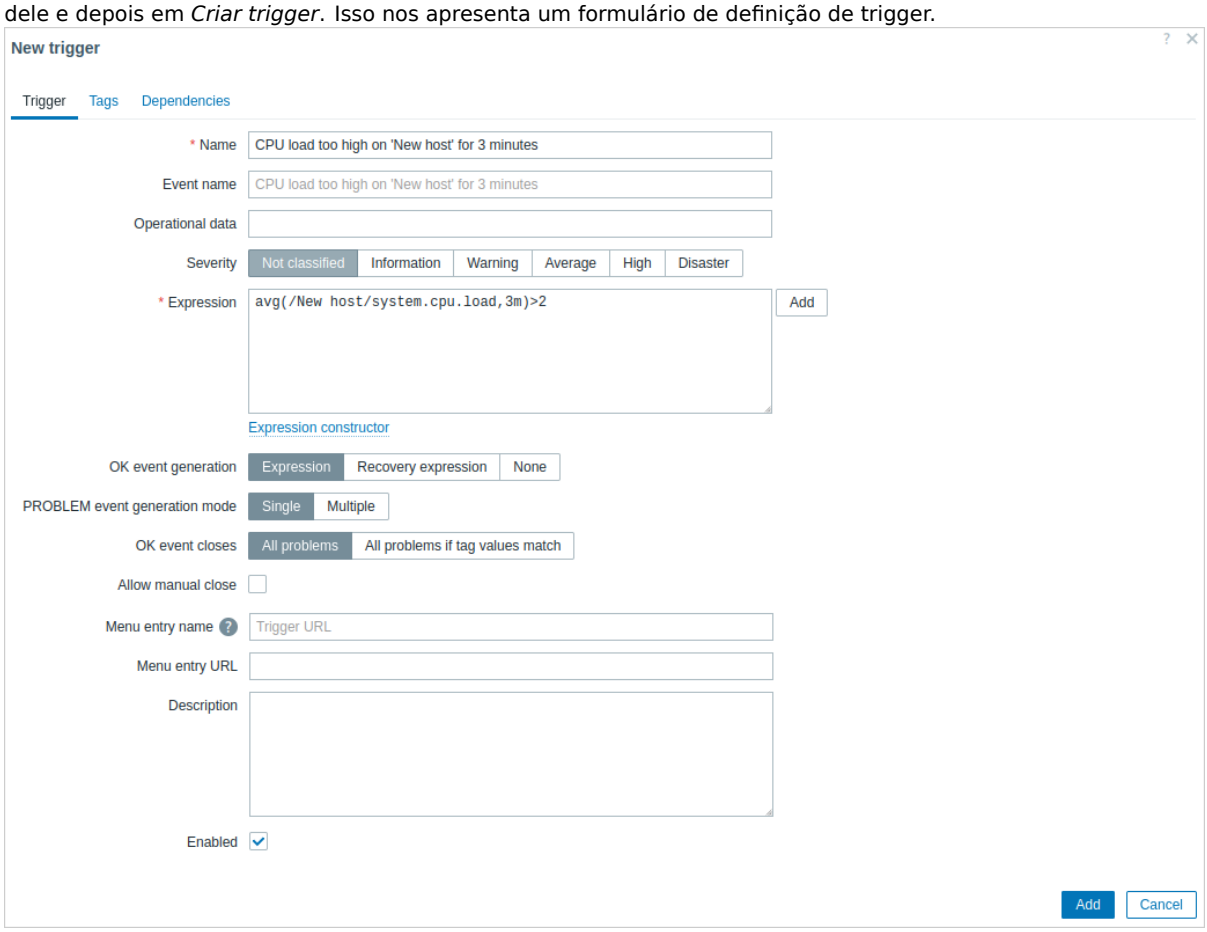
Nesta seção você aprenderá como configurar um trigger.

Os items apenas coletam dados. Para avaliar automaticamente os dados recebidos, precisamos definir triggers. Um trigger contém uma expressão que define um limite do que é um nível aceitável para os dados.

Se esse nível for ultrapassado pelos dados recebidos, um trigger será "acionado" ou entrará em um estado de 'Problema' - nos informando que algo aconteceu que pode exigir atenção. Se o nível voltar a ser aceitável, o trigger retorna ao estado 'Ok'.

Adicionando trigger

Para configurar um trigger para nosso item, vá para *Coleta de dados* > *Hosts*, encontre 'Novo host' e clique em *Triggers* ao lado dele e depois em *Criar trigger*. Isso nos apresenta um formulário de definição de trigger.



Para nosso trigger, as informações essenciais a serem inseridas aqui são:

Nome

- Insira *Carga de CPU muito alta em '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

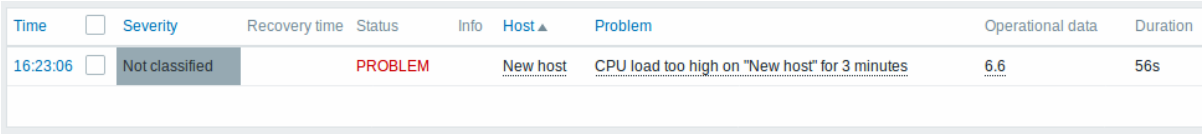
Esta é a expressão do trigger. Certifique-se de que a expressão seja inserida corretamente, até o último símbolo. A chave do item aqui (system.cpu.load) é usada para se referir ao item. Esta expressão específica basicamente diz que o limite do problema é excedido quando o valor médio da carga da CPU por 3 minutos é superior a 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 o status do trigger

Com um trigger definido, você pode estar interessado em ver seu status.

Se a carga da CPU excedeu o nível de limite que você definiu no trigger, o problema será exibido em *Monitoramento* > *Problemas*.



O piscar na coluna de status indica uma alteração recente no status do trigger, que ocorreu nos últimos 30 minutos.

5 Recebendo notificações de problemas

Visão geral

Nesta seção, você aprenderá como configurar alertas na forma de notificações no Zabbix.

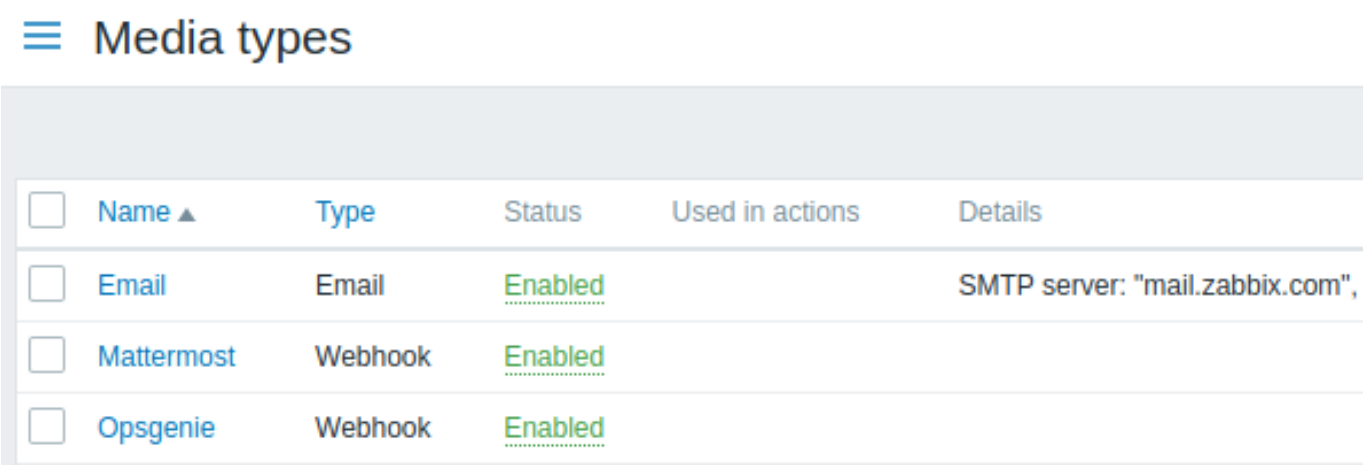
Com os items coletando dados e triggers projetados para ”disparar” em situações de problema, também seria útil ter algum mecanismo de alerta que nos notificasse sobre eventos importantes, mesmo quando não estivermos olhando diretamente para o frontend do Zabbix.

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.

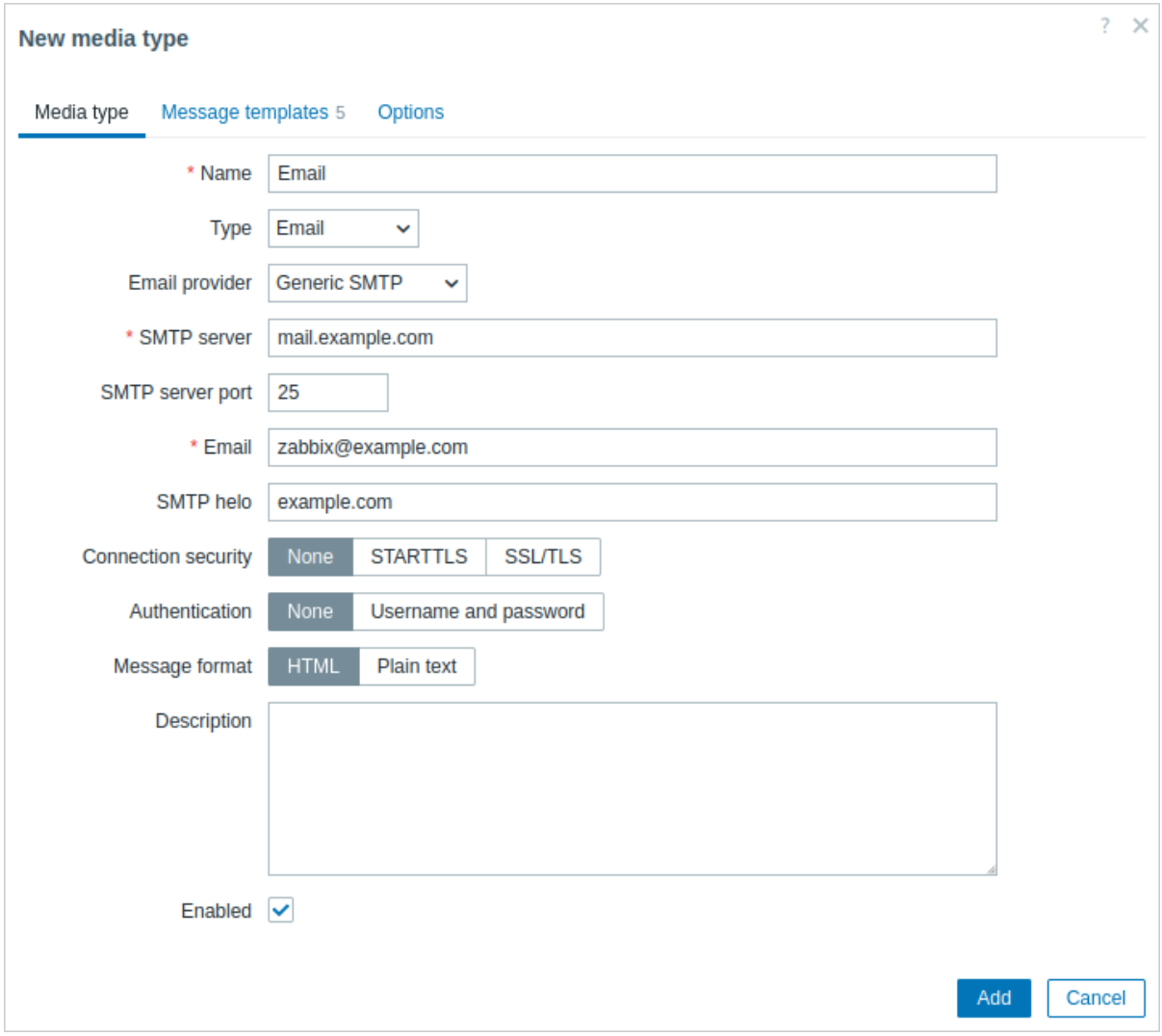
Configurações de e-mail

Inicialmente, existem vários métodos de entrega de notificação predefinidos no Zabbix. Email é um deles.

Para configurar as definições de e-mail, vá em *Alertas* > *Tipos de mídia* e clique em *Email* na lista de tipos de mídia predefinidos.



Isso apresentará o formulário de definição das configurações de e-mail.



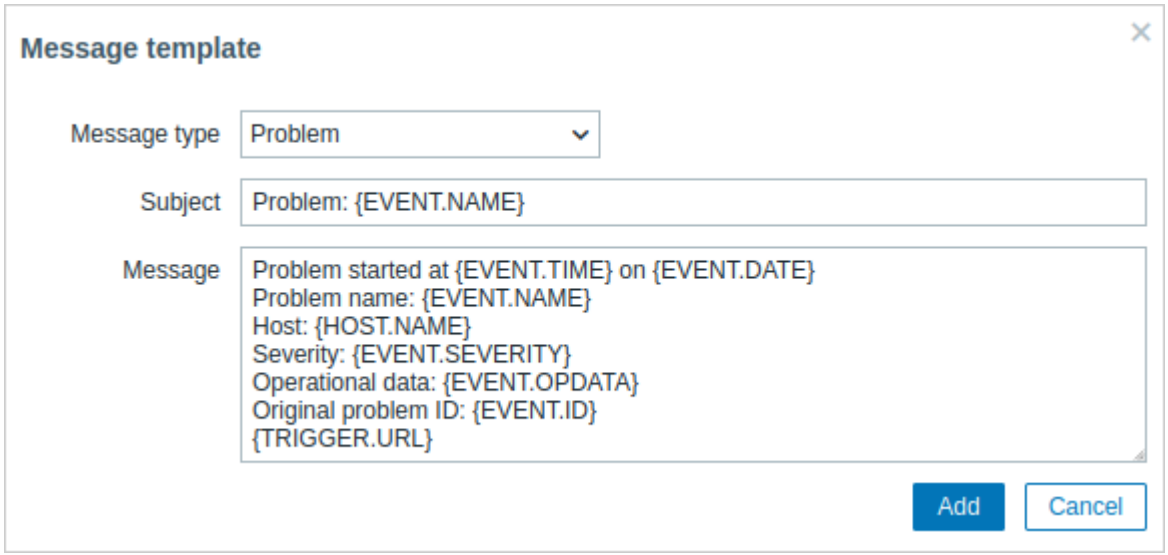
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

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.

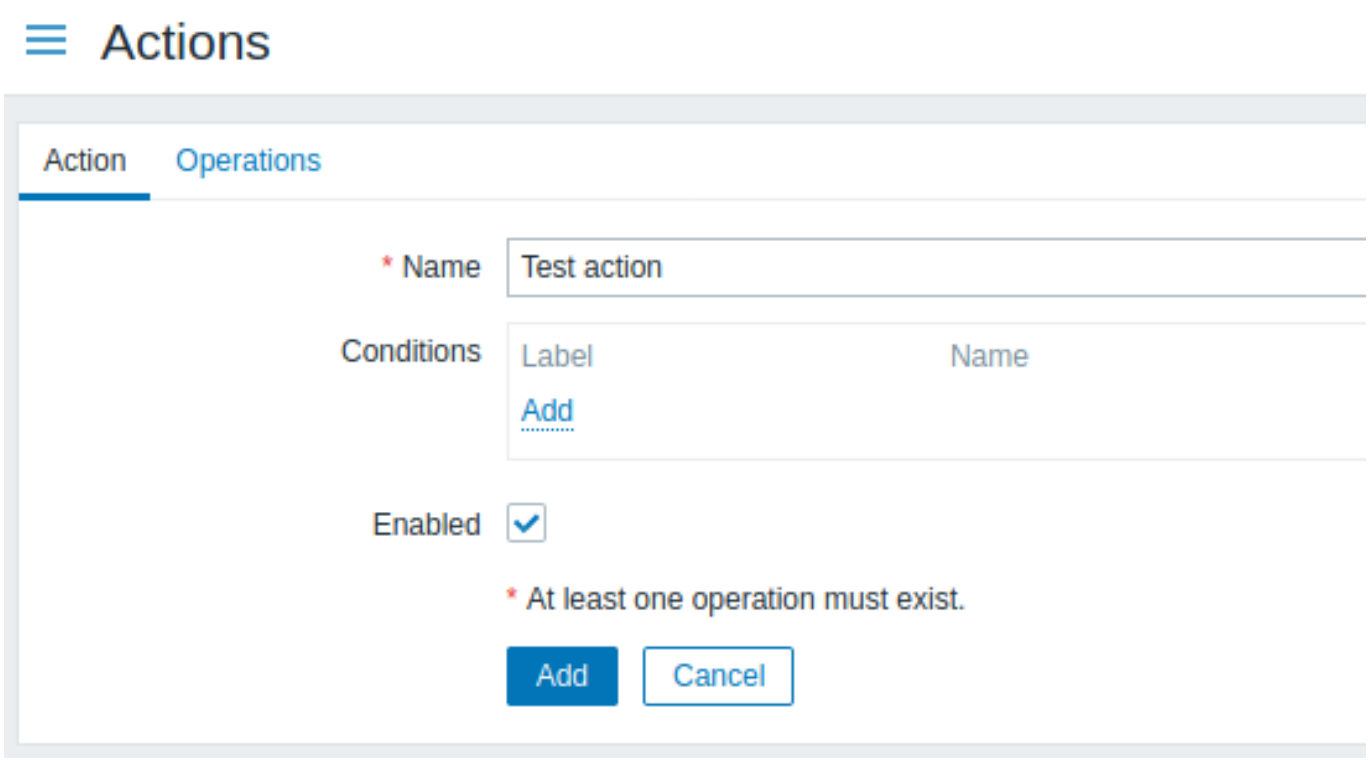


Clique em *Adicionar* quando estiver pronto e salve o formulário.

Agora você configurou o *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*.

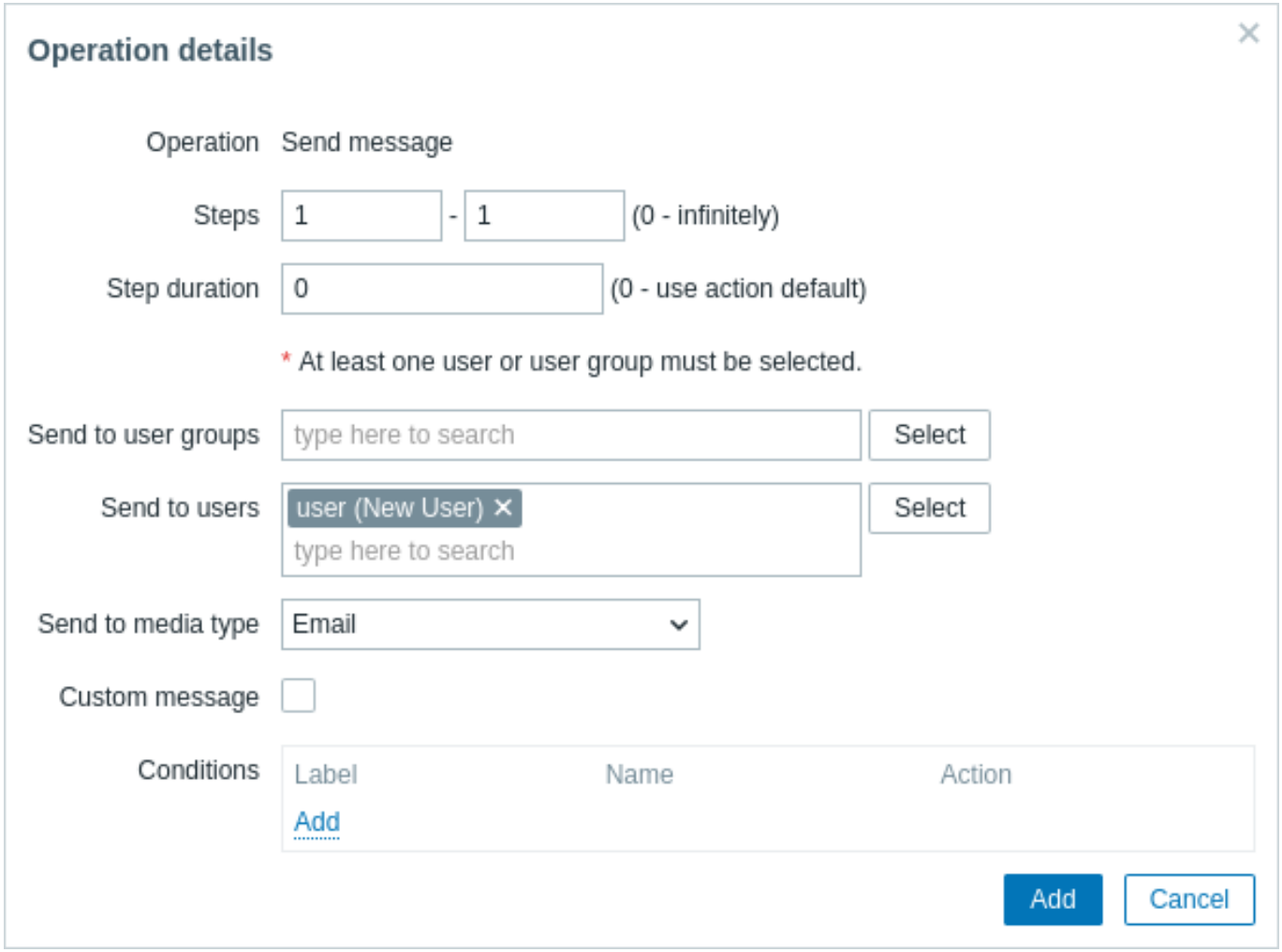


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Neste formulário, insira um nome para a ação.

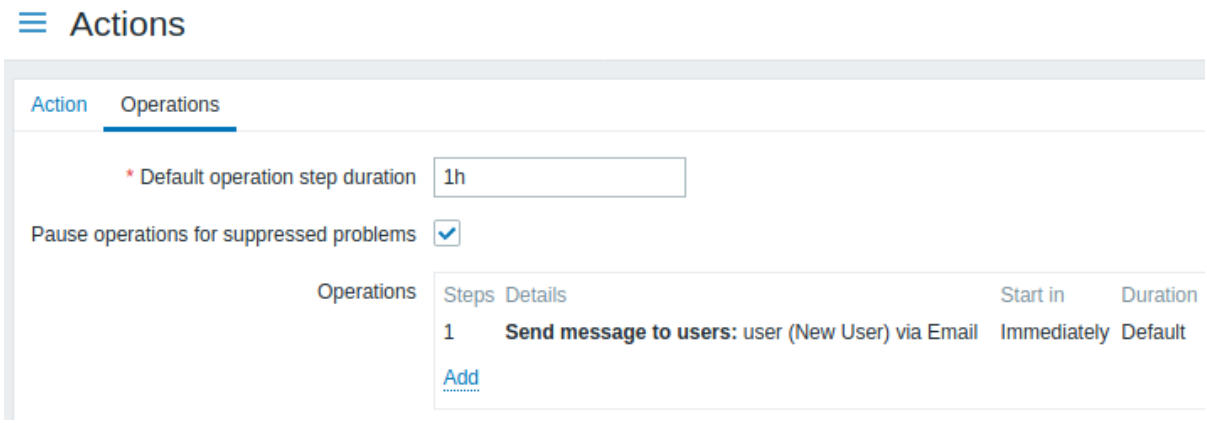
No caso mais simples, se não adicionarmos nenhuma condição mais específica, a ação será executada em qualquer alteração de trigger de 'Ok' para 'Problema'.

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 o valor de *Enviar para tipo de mídia*. Quando terminar, clique em *Adicionar* e a operação deverá ser 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, de propósito, aumentar 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 template

Visão geral

Nesta seção, você aprenderá como configurar um template.

Anteriormente, aprendemos como configurar um item, um trigger e como receber uma notificação de problema para o host.

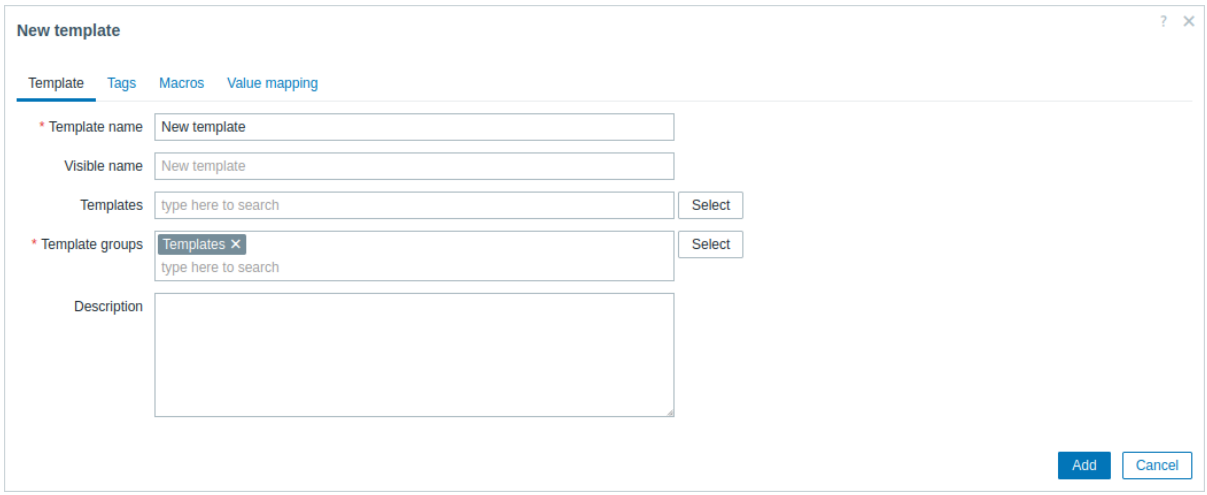
Embora todas essas etapas ofereçam um grande grau de flexibilidade por si só, pode parecer um grande número de etapas a serem realizadas se necessário, por exemplo, para mil hosts. Alguma automação seria útil.

É aí que os templates ajudam. Os templates permitem agrupar itens, triggers e outras entidades úteis para que possam ser reutilizados repetidamente, aplicando-os aos hosts em uma única etapa.

Quando um template é vinculado a um host, o host herda todas as entidades do template. Assim, basicamente, um conjunto pré-preparado de verificações pode ser aplicado 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 abrirá 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

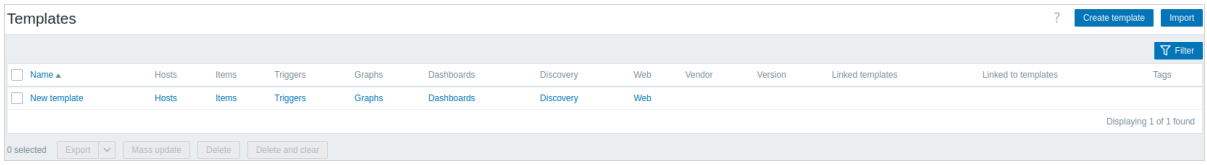
- Insira um nome para o template. São permitidos caracteres alfanuméricos, espaços e sublinhados.

Grupos de templates

- Selecione um ou mais grupos clicando no botão *Selecionar*. O template deve pertencer a um grupo.

Note:
As permissões de acesso aos grupos de templates 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.



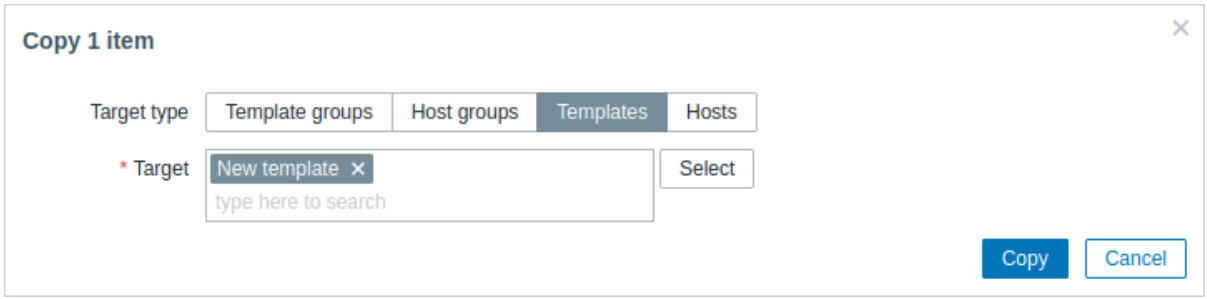
Como você pode ver, o template está lá, mas não contém nada - nenhum item, trigger ou outras entidades.

Adicionando item ao template

Para adicionar um item ao template, abra a lista de itens do 'Novo host' navegando até *Coleta de dados* → *Hosts* e clicando 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 a guia *Templates*.
- Selecione o template para o qual deseja copiar o item.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

- Clique em *Copiar*.

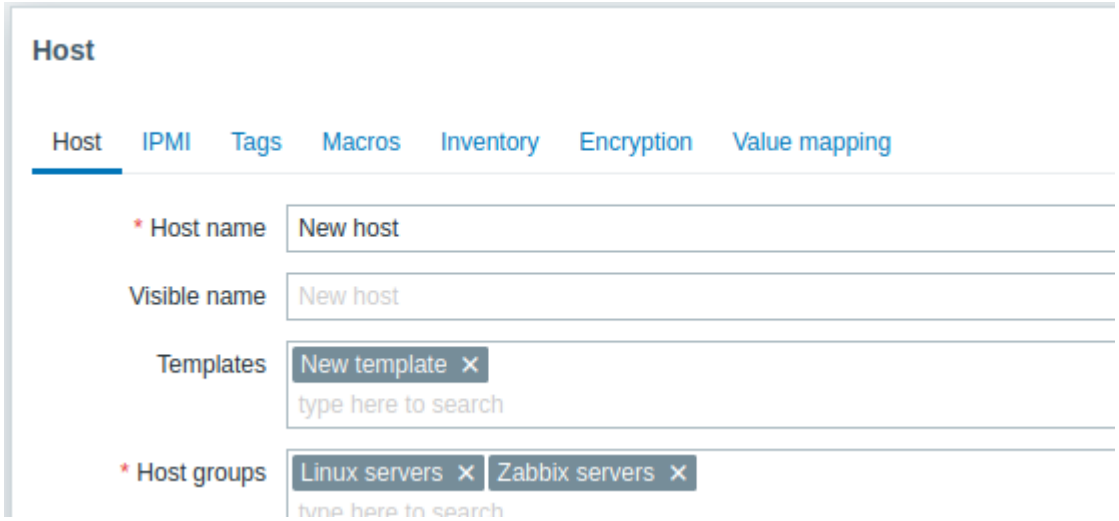
Se você for agora em *Coleta de dados* > *Templates*, o 'Novo template' deverá ter um novo item.

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 determinado propósito (monitoramento de SO, monitoramento de uma única aplicação).

Vinculando template ao host

Com um template pronto, só falta adicioná-lo a um host. Para isso, vá em *Coleta de dados* > *Hosts*, clique em 'Novo host' para abrir o formulário de configuração 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.

Vinculando templates predefinidos a hosts

Como você deve ter notado, o Zabbix vem com um conjunto de templates predefinidos para vários sistemas operacionais, dispositivos e aplicações. Para começar a monitorar rapidamente, você pode vincular o template apropriado a um host, mas esteja ciente de que esses templates precisam ser ajustados para o seu ambiente. Algumas verificações podem não ser necessárias e os intervalos de coleta podem ser muito frequentes.

Mais informações sobre templates estão disponíveis.

6 Zabbix appliance

Visão geral O appliance Zabbix fornece uma maneira de implantar instantaneamente o Zabbix server e o frontend, em vez de configurá-los manualmente ou reutilizar um server existente para o Zabbix.

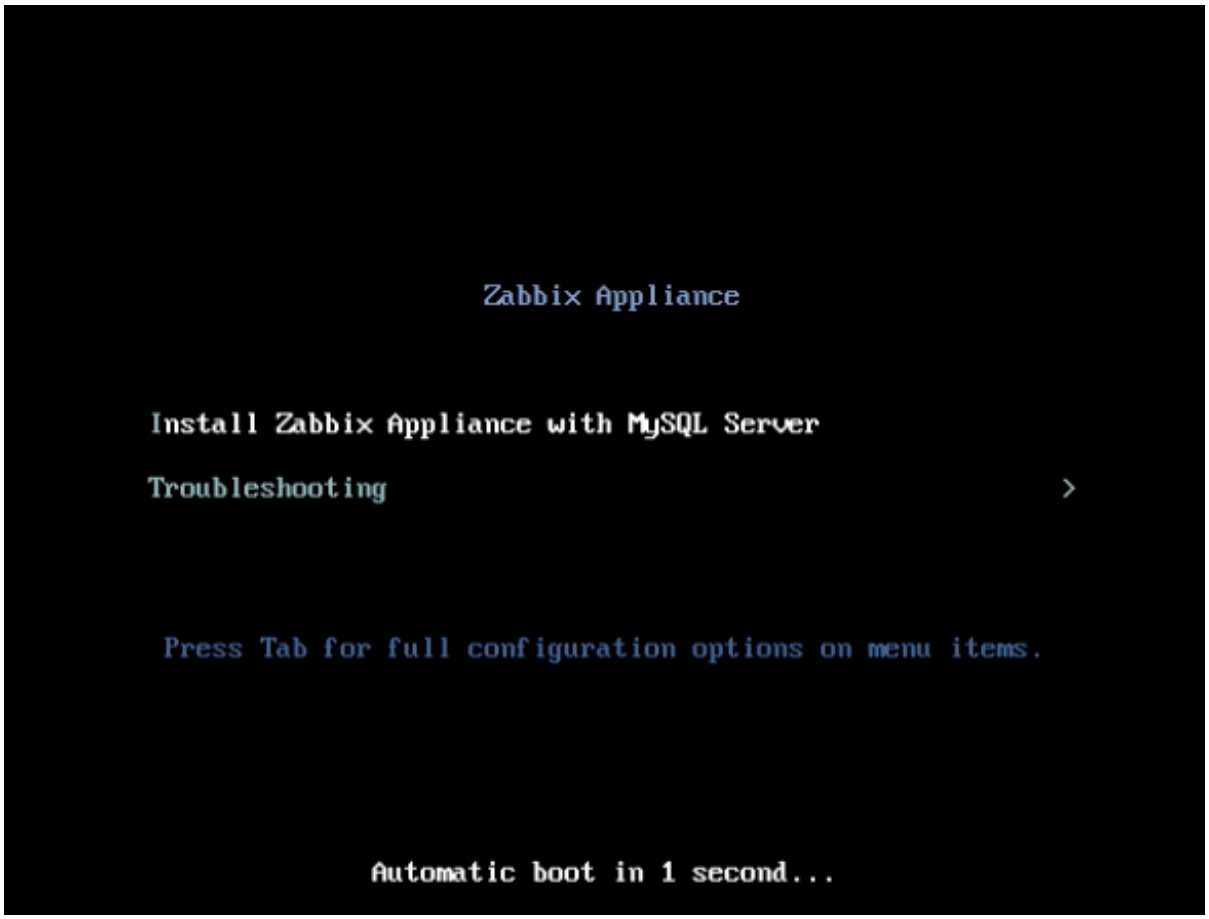
O appliance é baseado no AlmaLinux 8 (x86_64) e contém um Zabbix server pré-configurado rodando no MySQL e o frontend rodando no servidor web Nginx.

> **Attention:**
> Este appliance foi projetado para avaliação do Zabbix. O uso em ambientes de produção sérios é desencorajado.

As imagens do appliance estão disponíveis para download nos seguintes formatos:

- CD/DVD de instalação (.iso)
- VMware (.vmx) - veja notas
- Formato de virtualização aberto (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - veja notas
- KVM, Parallels, QEMU, USB stick, VirtualBox, Xen (.raw) - veja notas
- KVM, QEMU (.qcow2)

Menu de inicialização do CD/DVD de instalação do Zabbix:



Início rápido Pré-requisitos

1. Certifique-se de que a máquina host possui recursos suficientes para atender aos requisitos de sistema da máquina virtual:

- *RAM*: 4 GB
- *Espaço em disco*: pelo menos 8 GB devem ser alocados para a máquina virtual
- *CPU*: mínimo de 2 núcleos

2. Caso ainda não tenha sido instalado, instale o software de virtualização para inicializar a imagem do appliance (por exemplo, VirtualBox).

3. Baixe o appliance no formato suportado pelo seu software de virtualização.

4. Verifique as configurações de rede para garantir que o DHCP esteja habilitado na máquina host.

Instalação

1. Inicialize a máquina virtual do appliance a partir da imagem baixada.

2. Configure as configurações de rede da máquina virtual para permitir o acesso a partir de um navegador na máquina host. Isso pode ser feito ativando o *modo Bridge*.

3. Faça login na máquina virtual usando as credenciais padrão do sistema.

4. Para recuperar o endereço IP, execute o seguinte comando na máquina virtual:

```
ip addr show
```

5. Abra um navegador na máquina host e aponte para o endereço IP que o appliance recebeu via DHCP.

6. Faça login no Zabbix usando as credenciais padrão e comece a monitorar.

Configuração Esta seção descreve as configurações padrão frequentemente necessárias, juntamente com as opções de personalização disponíveis.

Credenciais

Sistema

- Nome de usuário: root
- Senha: zabbix

Frontend do Zabbix

- Nome de usuário: Admin
- Senha: zabbix

Após o login, você pode alterar a senha padrão nas configurações do perfil do usuário ou criar novos usuários e excluir o usuário padrão.

Banco de dados

As senhas para todos os usuários do banco de dados são geradas aleatoriamente durante o processo de instalação. Os seguintes usuários são definidos para o banco de dados:

Root:

- Nome de usuário: root
- Senha: a senha é armazenada no arquivo `/root/.my.cnf`. Não é necessário informar uma senha na conta root.

Servidor Zabbix:

- Nome de usuário: zabbix_srv
- Senha: a senha é armazenada no arquivo `/etc/zabbix/zabbix\_server.conf`

Frontend do Zabbix:

- Nome de usuário: zabbix_web
- Senha: a senha é armazenada no arquivo `/etc/zabbix/web/zabbix.conf.php`

Para alterar a senha de um usuário do banco de dados, modifique-a no MySQL e no arquivo de configuração correspondente.

Acesso ao frontend

O frontend do Zabbix pode ser acessado em *http://<IP da máquina virtual>*.

Por padrão, o acesso é permitido de qualquer lugar. Para limitar o acesso, modifique `/etc/nginx/conf.d/zabbix.conf`. Após salvar o arquivo editado, reinicie o Nginx conectando-se via SSH como usuário root e executando:

```
systemctl restart nginx
```

Endereço IP estático

Por padrão, o appliance usa DHCP para obter o endereço IP. Para definir um endereço IP estático:

- Faça login como usuário root.
- Execute os seguintes comandos, substituindo os valores pelos seus endereços IP personalizados:

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # Endereço IP do appliance/prefixo CIDR
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # Endereço IP do gateway
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # Endereço IP do servidor DNS
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

Como alternativa, você pode abrir o arquivo `/etc/NetworkManager/system-connections/eth0.nmconnection` e adicionar as seguintes linhas:

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

Após salvar o arquivo editado, execute o comando `systemctl restart network`.

Configuração do firewall

Para gerenciar as configurações do firewall, o appliance utiliza o iptables com regras predefinidas:

- Abrir a porta SSH (22 TCP)
- Abrir as portas do agent Zabbix (10050 TCP) e do trapper Zabbix (10051 TCP)
- Abrir as portas HTTP (80 TCP) e HTTPS (443 TCP)
- Abrir a porta SNMP trap (162 UDP)
- Abrir conexões de saída para a porta NTP (123 UDP)
- Limitar pacotes ICMP a 5 pacotes por segundo
- Bloquear todas as outras conexões de entrada

Para abrir portas adicionais, modifique o arquivo `/etc/sysconfig/iptables` e recarregue as regras do firewall:

```
systemctl reload iptables
```

Repositórios

O appliance Zabbix usa o pacote *zabbix-release* do repositório Zabbix. Os repositórios são configurados no diretório `/etc/yum.repos.d/*`.

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
```

> **Note:**
> O fuso horário do frontend do Zabbix é definido separadamente e pode ser alterado nas configurações do frontend. O fuso horário padrão para o frontend do Zabbix é Europe/Riga.

Localização dos arquivos

- Os arquivos de configuração estão localizados em `/etc/zabbix`
- Os arquivos de log do Zabbix server, proxy e agent 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`

Serviços do sistema

Os serviços do Systemd estão disponíveis. Para ver a lista de serviços do Zabbix, execute o seguinte comando na máquina virtual:

```
systemctl list-units zabbix*
```

Observações específicas do formato de imagem 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. Como alternativa, após a conclusão da conversão, você pode excluir o adaptador existente e adicionar um adaptador E1000.

Imagem HDD/flash (raw)

Para inicializar a imagem, execute:

```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Substitua `/dev/sdc` pelo caminho do seu dispositivo de disco Flash/HDD.

Hyper-V

Se o appliance não iniciar no Hyper-V, tente pressionar **Ctrl+Alt+F2** para alternar para uma sessão TTY.

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 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 template

Visão geral

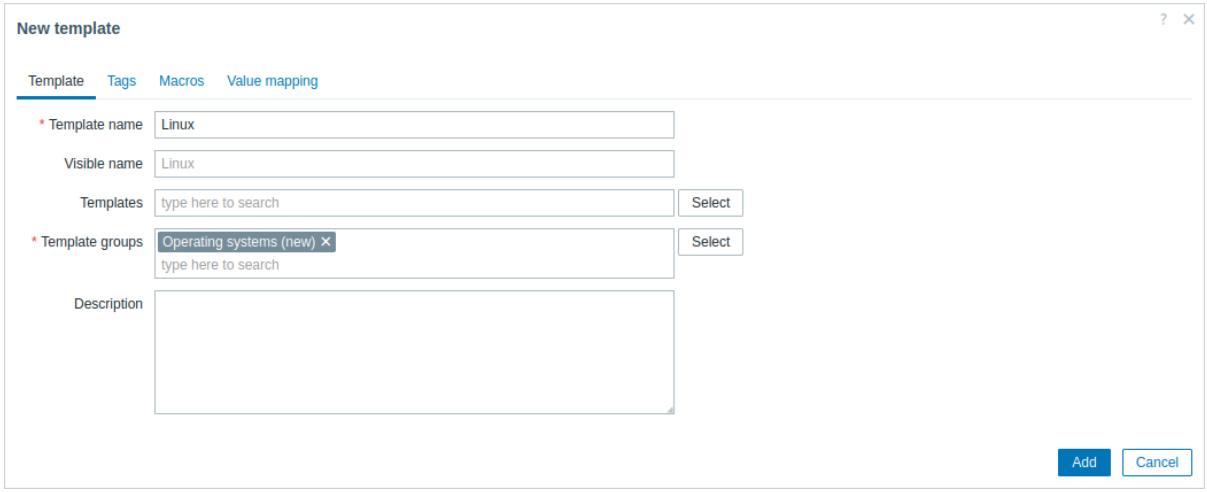
Configurar um template requer que você primeiro crie um template definindo seus parâmetros gerais e, em seguida, adicione entidades (itens, triggers, gráficos, etc.) a ele.

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.

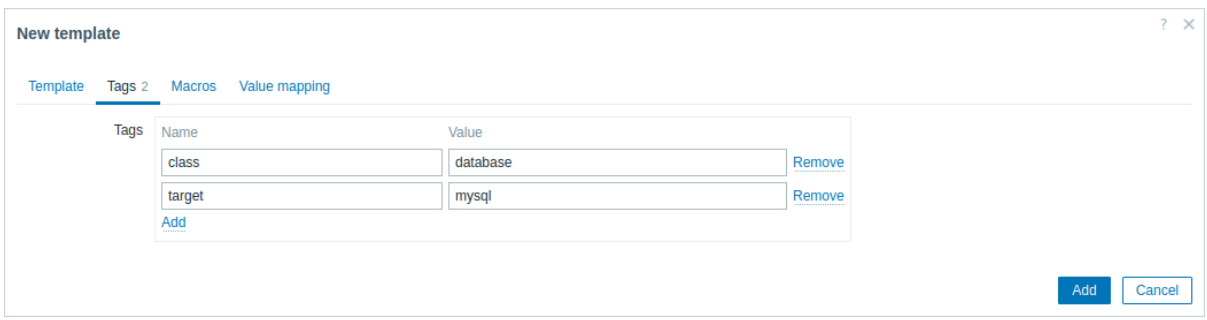


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Atributos do template:

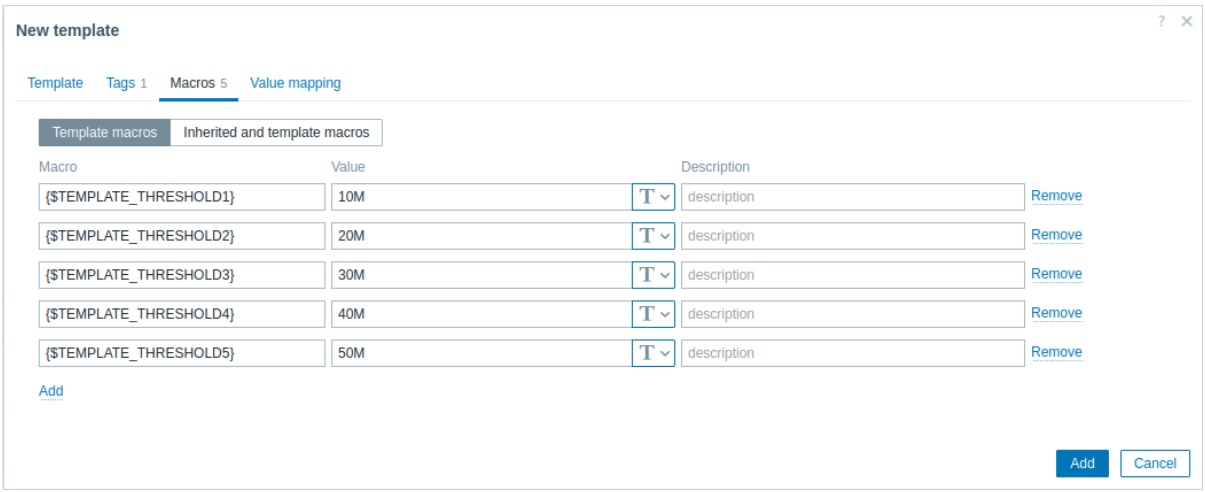
Parâmetro	Descrição
Nome do template	Nome único do template. Alfanuméricos, espaços, pontos, traços e sublinhados são permitidos. Espaços à esquerda e à direita não são permitidos.
Nome visível	Se você definir este nome, ele será o visível em listas, mapas, etc.
Templates	Vincule um ou mais templates 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, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

A aba **Tags** permite definir tags no nível do template. Todos os problemas dos hosts vinculados a este template serão marcados com os valores inseridos aqui.

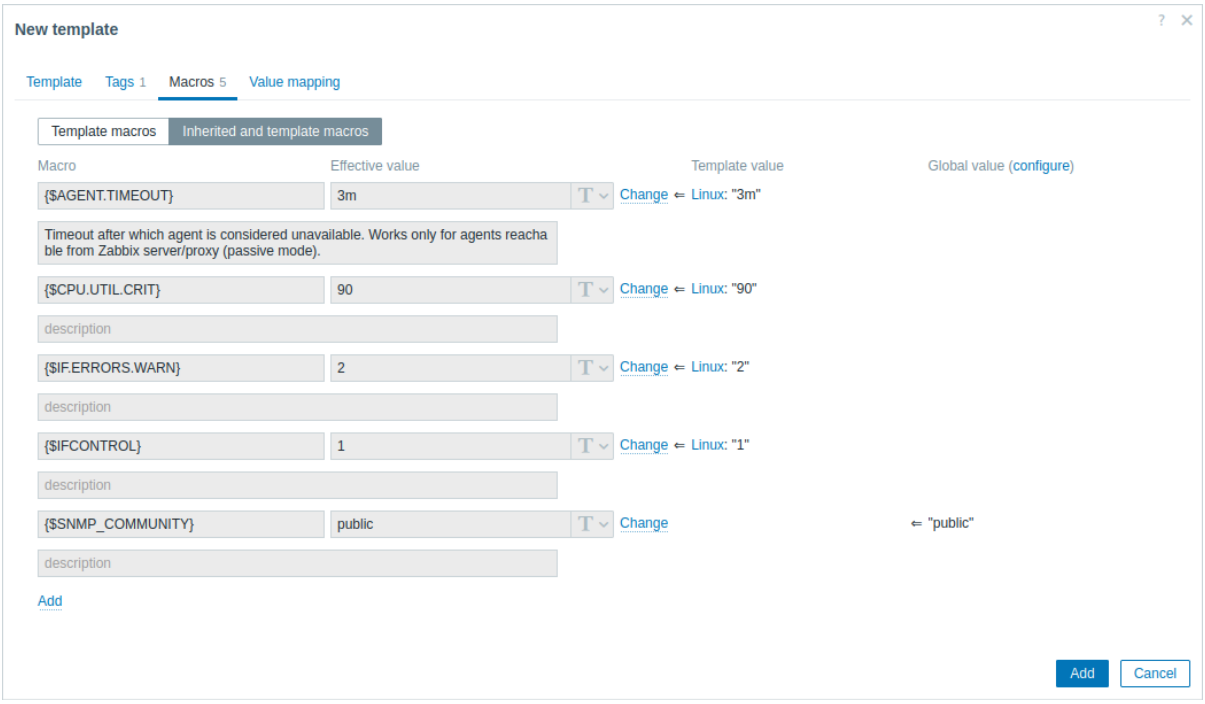


Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags.

A aba **Macros** permite definir macros de usuário no nível do 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 dos 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 uma template. Triggers e gráficos não podem ser adicionados sem o respectivo item.

Existem duas maneiras de adicionar items à template:

1. Para criar novos items, siga as orientações em Criando um item.

2. Para adicionar items existentes à 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 à template.
- Clique em *Copiar* abaixo da lista de items.
- Selecione a template (ou grupo de templates) para a qual os items devem ser copiados e clique em *Copiar*. Todos os items selecionados devem ser copiados para a 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 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 seção descoberta de baixo nível 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 atribuição de 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:**
> Somente usuários Super admin podem criar grupos de templates.

Existem duas opções para criar um grupo de templates no frontend do Zabbix.

Opção um:

- Vá para: *Coleta de dados* → *Grupos de templates*
- Clique em *Criar grupo de templates* 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 templates*.

Depois que o grupo de templates for criado, você pode clicar no nome do grupo na lista em *Coleta de dados* → *Grupos de templates* para editar o nome do grupo, clonar o grupo ou excluir o grupo.

Excluir um grupo de templates exclui apenas o grupo lógico, não os templates do grupo. Não é possível excluir um grupo de templates 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`.

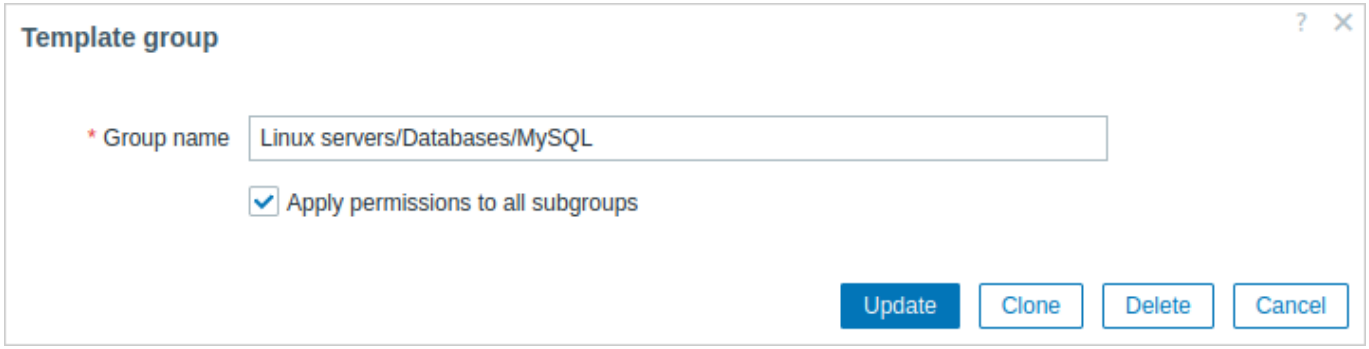
Ao criar um subgrupo, não é permitido usar barras no início ou no final, nem várias barras seguidas. Não há suporte para escape de '/'.

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 já 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 já 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*.



Marcando esta caixa de seleção e clicando 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 unlink 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.

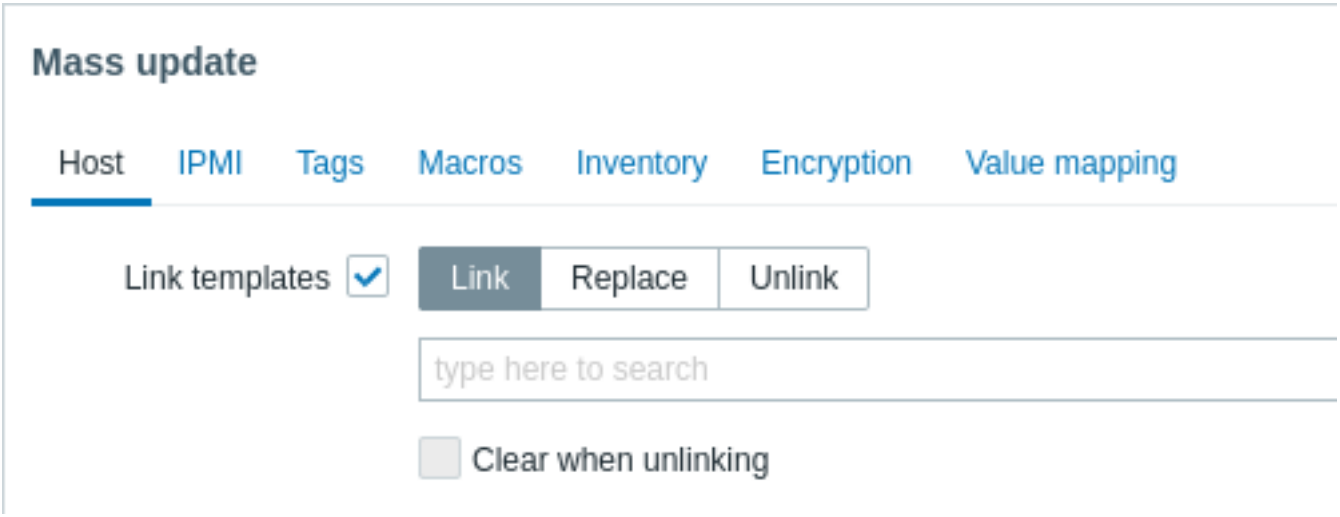
Critérios de unicidade de entidades

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 do trigger e expressão;
- para gráficos personalizados - nome do gráfico e seus items.

Vinculando templates a vários hosts

Para atualizar o vínculo 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 *Vincular templates*:



Para vincular templates adicionais, 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 o template a ser vinculado.

A opção *Substituir* permitirá vincular um novo template enquanto desvincula qualquer template que estava vinculado aos hosts anteriormente. A opção *Desvincular* permitirá especificar quais templates desvincular. A opção *Limpar ao desvincular* permitirá não apenas desvincular quaisquer templates vinculados anteriormente, 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 um 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, você ainda pode, por exemplo, habilitar/desabilitar um item em hosts individuais e definir o intervalo de atualização, o tempo de retenção do 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 nas entidades implementadas no nível do template substituirão as personalizações anteriores das entidades no nível do 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 localize 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* irá simplesmente remover 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* irá remover 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, dashboards de 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 alterar algum atributo para vários templates de uma só vez. Em vez de abrir cada template individualmente para edição, você pode usar a função de atualização em massa para isso.

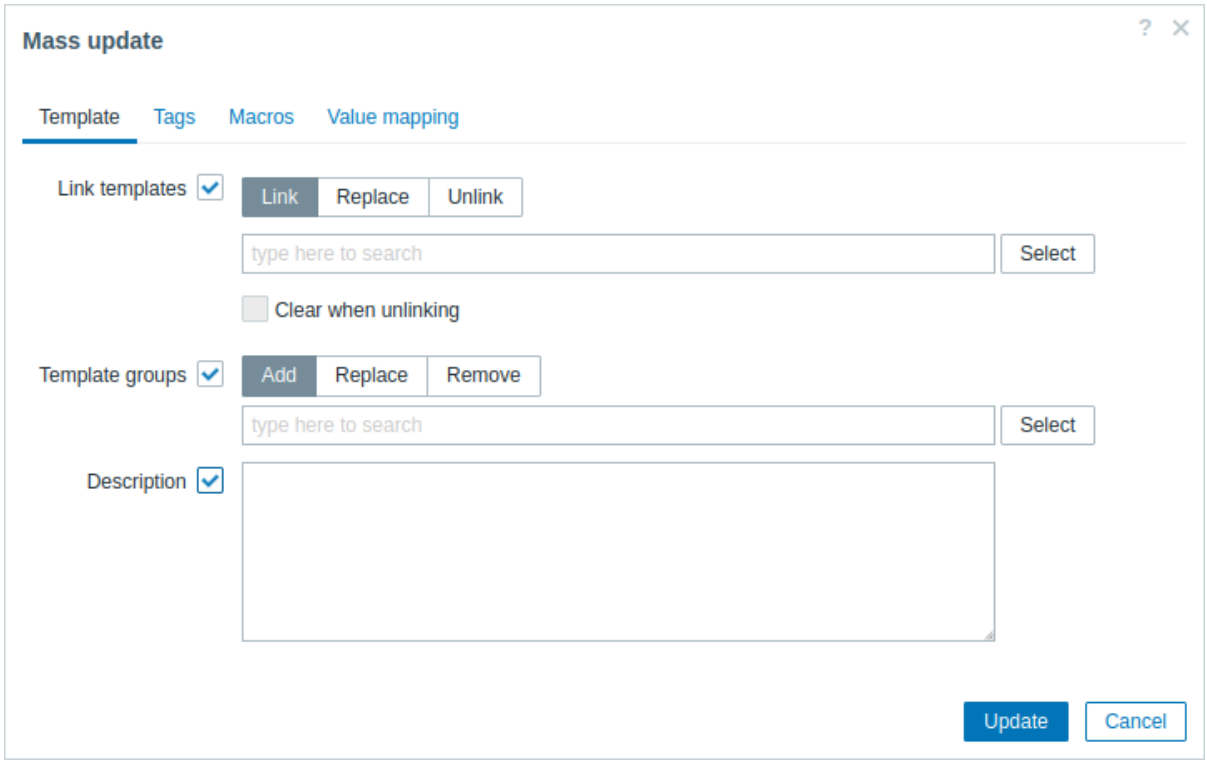
Usando atualização em massa

Para atualizar em massa alguns templates, faça o seguinte:

1. Marque as caixas de seleção antes dos templates que 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 aba **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.

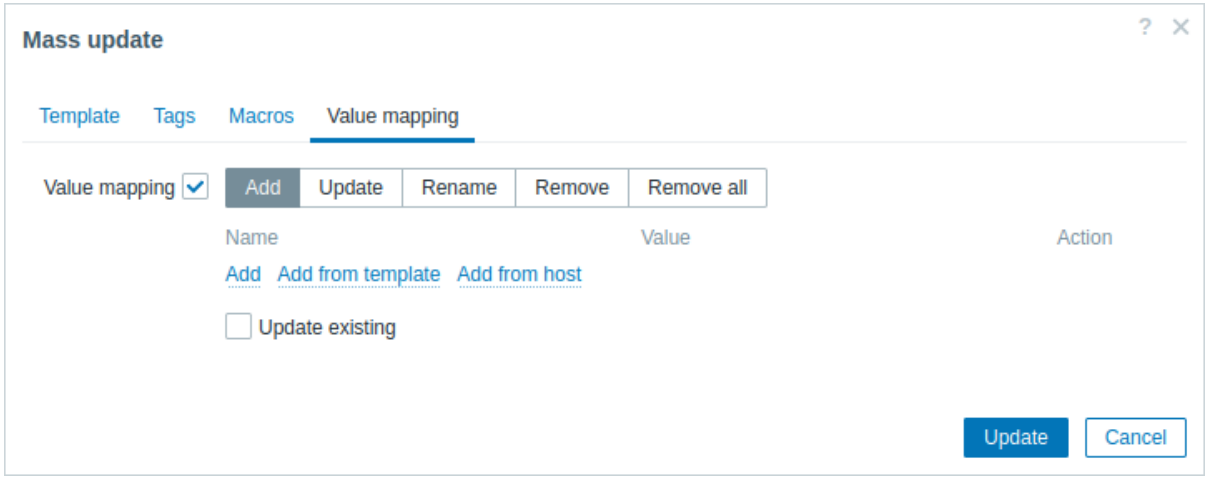
A aba **Macros** permite atualizar em massa macros no nível do template.

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 especificada 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 as 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 aba **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 de template* e *Adicionar de 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, aplicação, 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 poderá adicionar itens de monitoramento a ele.

Hosts são organizados em grupos de hosts.

Prossiga para:

- Assistente de Host para uma interface guiada, passo a passo, para criar e configurar um host novo ou existente.
- Configurando um host para uma abordagem clássica de criação e configuração de um host.

1 Assistente de host

Visão geral

O Assistente de Host é uma interface guiada, passo a passo, para configurar seu alvo de monitoramento (dispositivo, aplicação, serviço, etc.) no Zabbix. Ele orienta você pelas seguintes etapas:

- Seleção de um template
- Criação ou seleção de um host
- Instalação do Zabbix agent ou agent 2
- Adição de uma interface de host
- Aplicação de configuração adicional ao seu alvo de monitoramento ou ao host Zabbix (se exigido pelo template)

Para acessar o Assistente de Host no frontend do Zabbix, faça o seguinte:

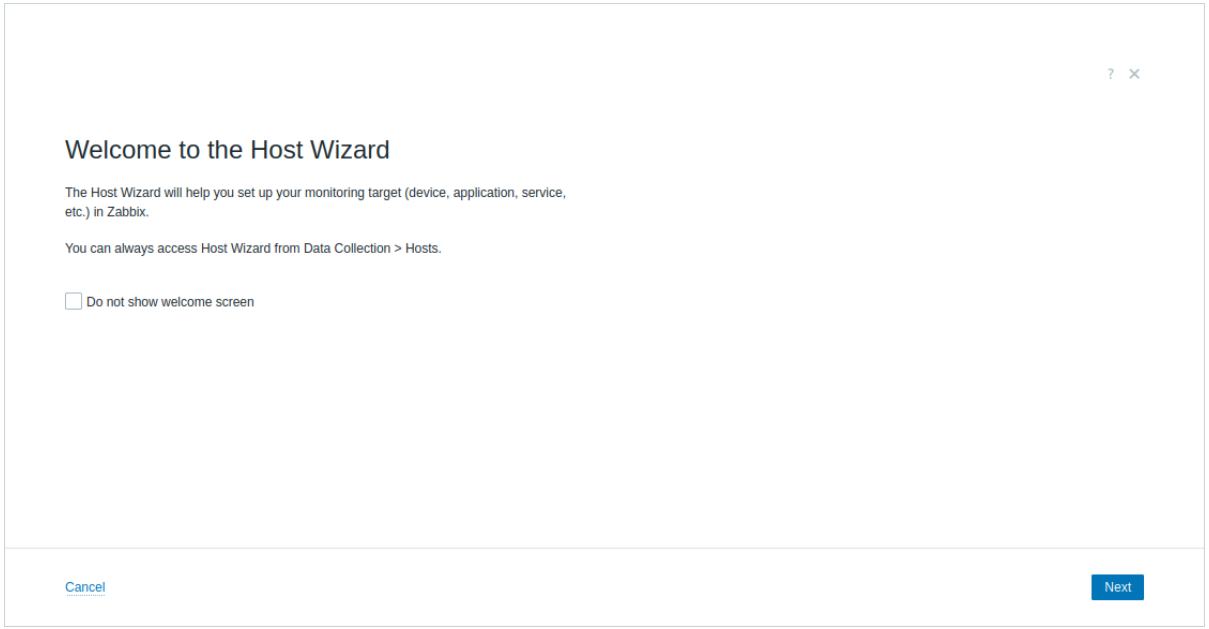
- Vá para: *Coleta de dados* > *Hosts*
- Clique em *Assistente de Host* no canto superior direito da tela
- Siga as instruções do assistente

Você também pode editar um host existente clicando em seu nome (ou no ícone de três pontos ao lado dele) em várias seções do frontend para acessar o Assistente de Host a partir do menu de host. Observe que algumas etapas são puladas ao editar um host existente se ele já incluir a configuração exigida pelo template selecionado.

Alternativamente, você pode usar a abordagem clássica para criar e configurar um host.

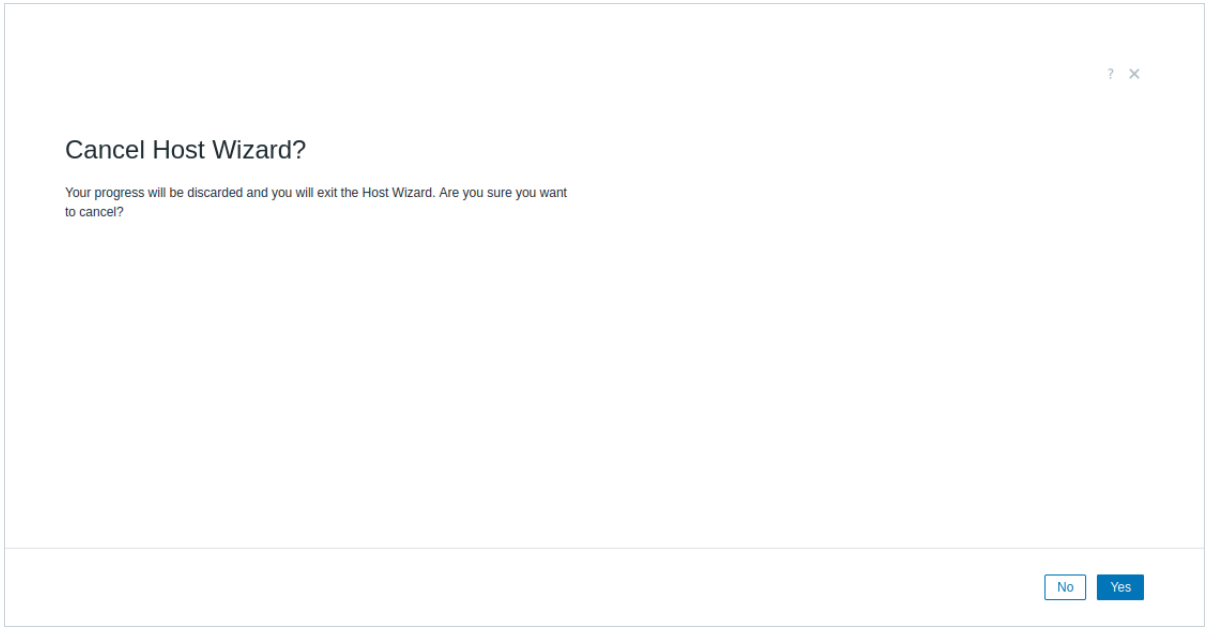
Bem-vindo ao Assistente de Host

Ao iniciar o Assistente de Host, uma tela de boas-vindas é exibida:



Para pular esta tela em sessões futuras, você pode marcar a caixa de seleção *Não mostrar tela de boas-vindas* e clicar em *Próximo*.

Se você começar a configurar as opções nas próximas etapas do Assistente de Host e tentar fechar o assistente clicando no ícone de fechar ou pressionando ESC, uma tela de confirmação será exibida:



Clicar em *Sim* sairá do Assistente de Host sem salvar seu progresso. Clicar em *Não*, ESC ou no ícone de fechar retornará à última etapa.

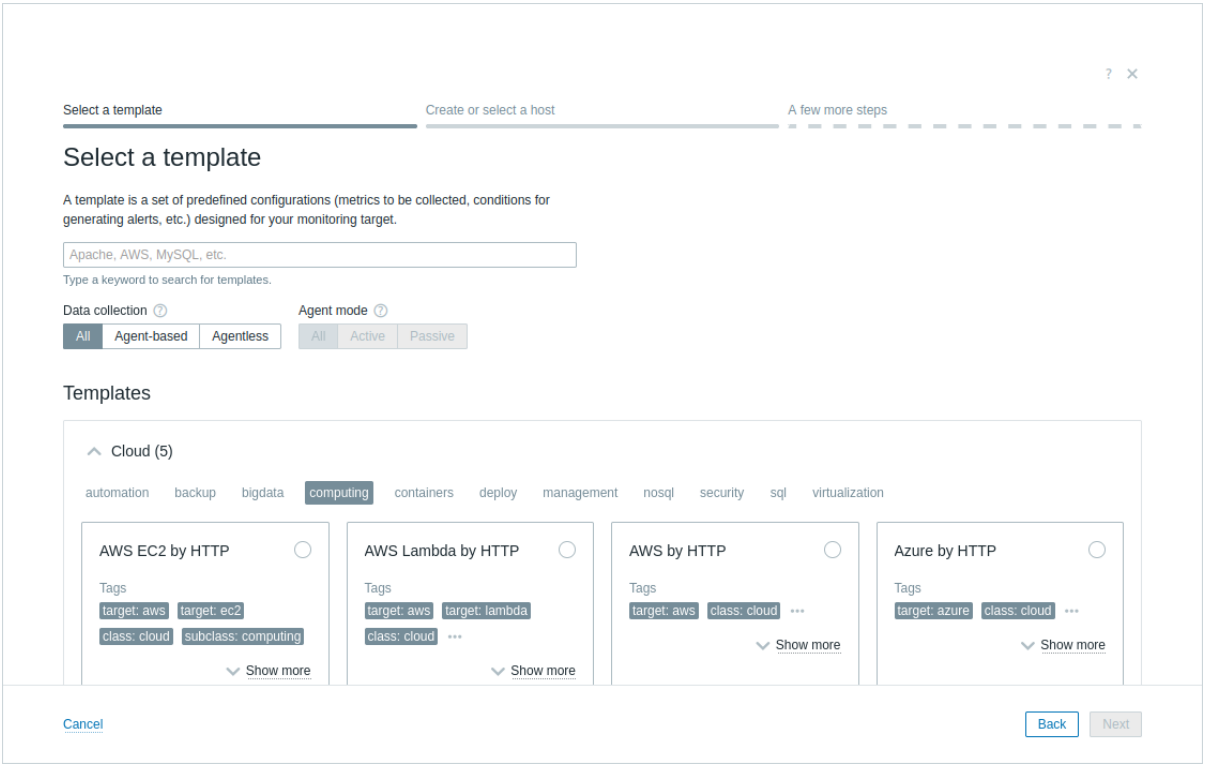
Selecione um template

O primeiro passo para configurar seu alvo de monitoramento é selecionar um template—um conjunto de configurações predefinidas (métricas a serem coletadas, condições para geração de alertas, etc.) projetadas para seu alvo de monitoramento.

Os templates prontos para uso do Zabbix oferecem uma variedade de configurações de monitoramento predefinidas para sistemas operacionais, aplicativos, bancos de dados, dispositivos de rede, serviços e muito mais.

Nesta etapa, você pode:

- Navegar pelos templates por classe (Cloud, Database, Network, etc.) e subclasse (por exemplo, automation, backup), que são baseadas em tags de template.
- Pesquisar templates por palavras-chave no nome do template ou nos nomes e valores das tags do template.
- Filtrar templates pelo método de coleta de dados (baseado em agent ou sem agent; templates baseados em agent incluem pelo menos um item Zabbix agent).
- Filtrar templates pelo modo do agent (ativo ou passivo; templates ativos incluem pelo menos um item Zabbix agent (ativo), enquanto templates passivos incluem pelo menos um item passivo; para detalhes, consulte Verificações passivas e ativas do agent).
- Ao usar o Assistente de Host para configurar um host existente, você também pode filtrar templates já vinculados ao host.



Attention:
Ao selecionar um template, você verá a seguinte mensagem: *Alguns templates (n) são incompatíveis com o Assistente de Host. Veja como atualizá-los. Templates personalizados não são suportados.* Isso indica que alguns templates prontos para uso ainda não são compatíveis com o Assistente de Host e precisam ser atualizados; veja as instruções de atualização.

O template selecionado determinará as próximas etapas no Assistente de Host. Por exemplo, se você selecionar o template *MySQL por Zabbix agent*, que usa o Zabbix agent para coleta de dados, o assistente irá guiá-lo pelo processo de instalação do agent.

Crie ou selecione um host

Um template deve ser vinculado a um host—uma entidade no Zabbix que representa seu alvo de monitoramento. Quando vinculado, o host recebe todas as entidades do template, como items (métricas a serem coletadas) e triggers (condições para geração de alertas), que são necessárias para o monitoramento.

Cada host também deve pertencer a pelo menos um grupo de hosts, que é usado para organizar hosts e atribuir permissões de usuário a eles.

Nesta etapa, você pode:

- Criar um novo host e grupo de hosts inserindo nomes para ambos.
- Criar um novo host inserindo seu nome e selecionar um grupo de hosts existente ao qual o novo host será atribuído.
- Selecionar um host existente; você também pode atribuir este host a grupos de hosts adicionais sem removê-lo de nenhum grupo atual.

Note:
Selecionar um host existente não remove ou sobrescreve sua configuração atual, a menos que seja declarado explicitamente em etapas posteriores. Hosts descobertos não podem ser selecionados.

Por exemplo, você pode criar um novo host chamado *MySQL server* para representar o servidor MySQL instalado localmente que deseja monitorar. O template *MySQL by Zabbix agent* selecionado anteriormente será vinculado a este host. Além disso, você pode selecionar o grupo de hosts existente *Databases* (ou criar um novo, por exemplo, *MySQL servers*) para organizar o host com outros hosts relacionados a banco de dados e, posteriormente, simplificar o gerenciamento de permissões.

Instalar o agent do Zabbix

Templates baseados em agent (como *MySQL por Zabbix agent*) requerem a instalação do agent do Zabbix ou do agent 2—um processo implantado no seu alvo de monitoramento para monitorar ativamente recursos e aplicações locais (se necessário pelo template).

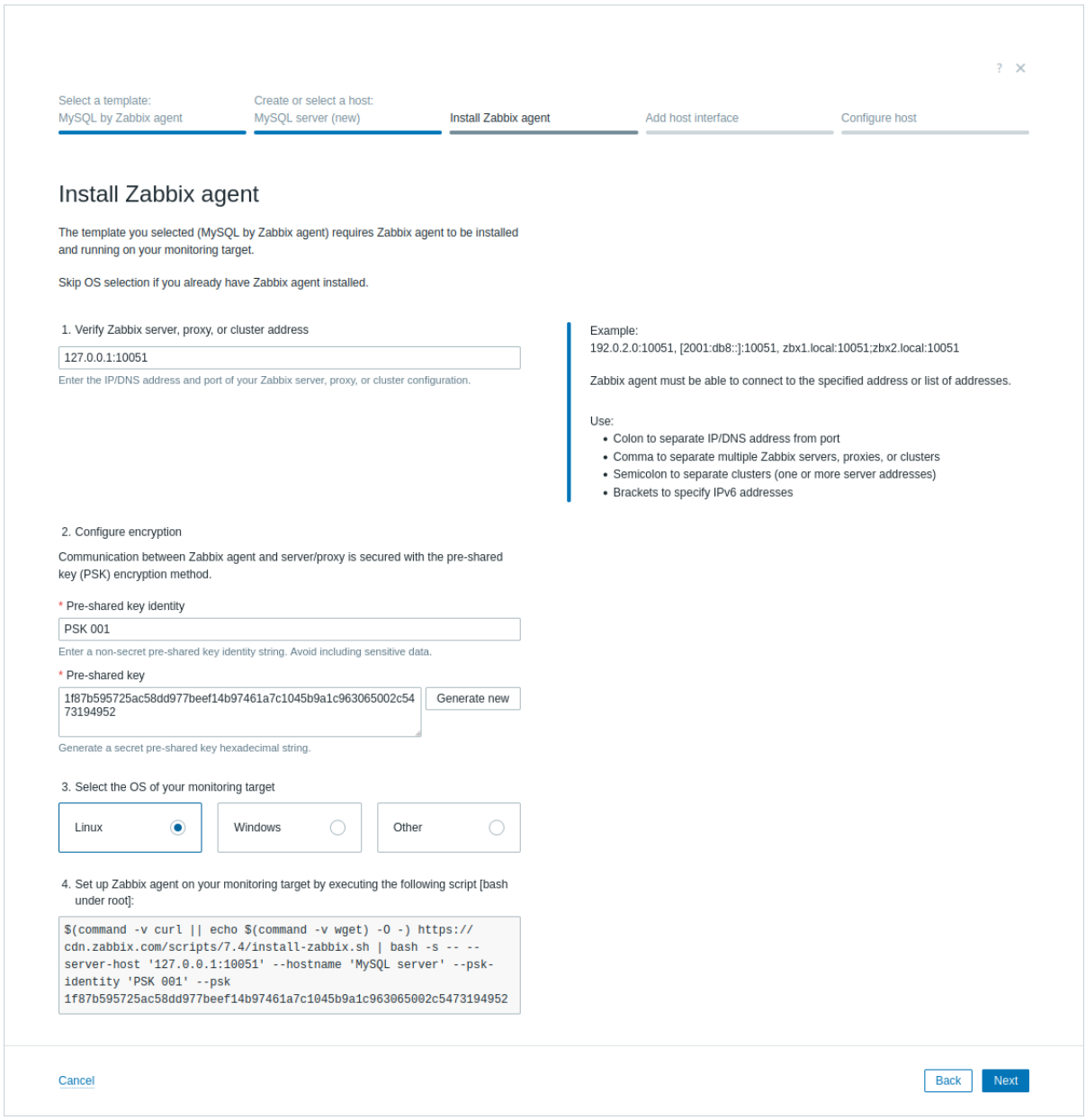
Neste passo:

- Verifique o endereço do server, proxy ou cluster do Zabbix (por exemplo, 192.0.2.0:10051). Este valor será usado para configurar o parâmetro Server e ServerActive para o agent do Zabbix.
- Insira uma nova identidade de chave pré-compartilhada exclusiva e não secreta (por exemplo, *PSK 001* ou *mysql-agent-psk1*) e gere a chave pré-compartilhada. Esses valores serão usados para configurar a criptografia de chave pré-compartilhada (PSK) para o host do Zabbix (por exemplo, *Servidor MySQL*) e o agent do Zabbix.

Attention:
Ao editar um host existente, a configuração de PSK substituirá todas as configurações de criptografia existentes no host.

- Selecione o sistema operacional da máquina que hospeda seu alvo de monitoramento.
- Instale o agent do Zabbix nesse sistema executando o script fornecido ou seguindo as instruções de instalação.

Por exemplo, você pode definir *Identidade da chave pré-compartilhada* como *PSK 001* e gerar uma nova *Chave pré-compartilhada*. Em seguida, você pode selecionar *Linux* como o sistema operacional e executar o script fornecido nesse sistema. Após instalar o agent do Zabbix, retorne ao Assistente de Host.



Adicionar interface do host

A interface do host define como o Zabbix server se conecta ao seu alvo de monitoramento pela rede. Ela define parâmetros de conexão como endereço IP, nome DNS, porta e tipo de interface (Agent, SNMP, JMX ou IPMI), dependendo do método de coleta de dados exigido pelo template selecionado.

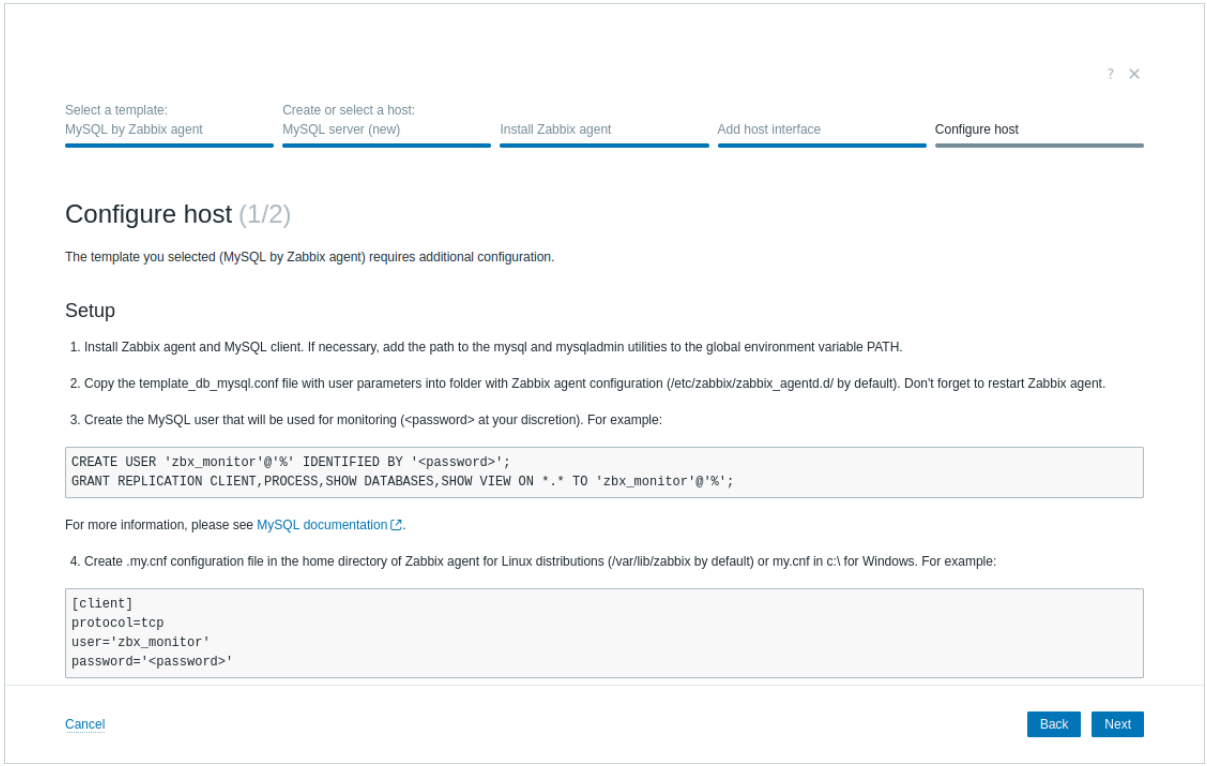
Nesta etapa, insira os detalhes da interface do host exigidos pelo template selecionado.

Por exemplo, você pode usar o endereço padrão do agent (127.0.0.1) e a porta (10050) se o Zabbix server, o Zabbix agent e o servidor MySQL estiverem todos rodando localmente na mesma máquina.

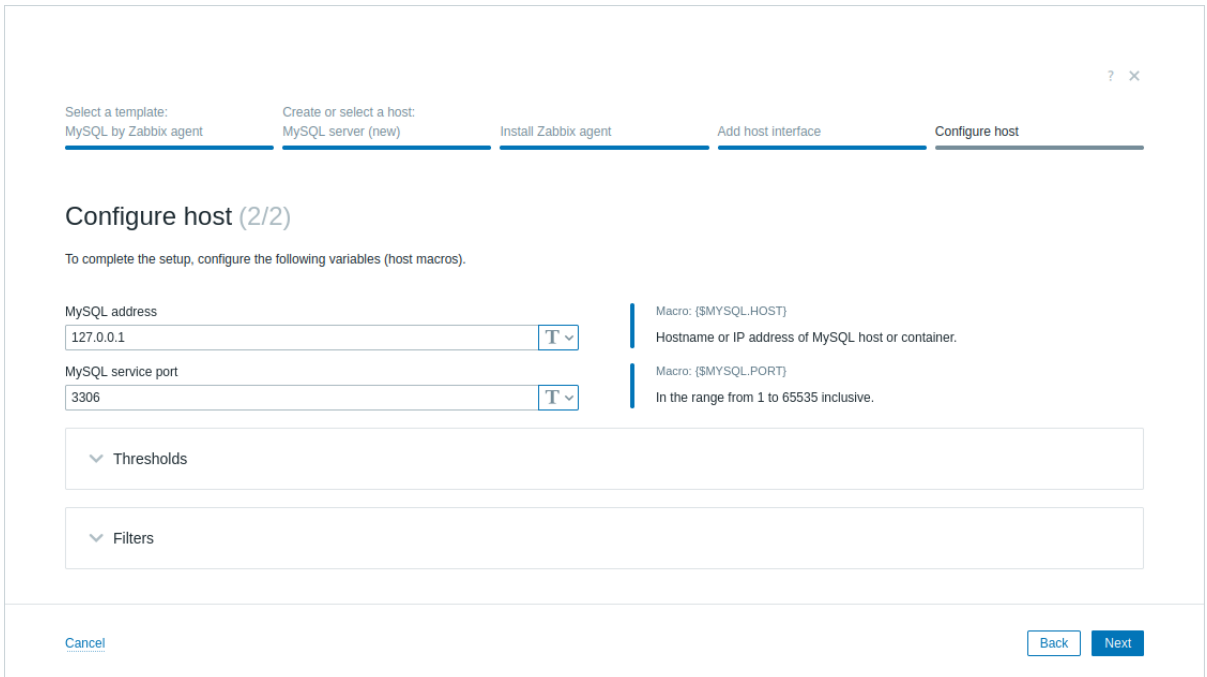
Configurar host

Alguns templates exigem configuração adicional antes que o host possa ser criado. Isso pode incluir:

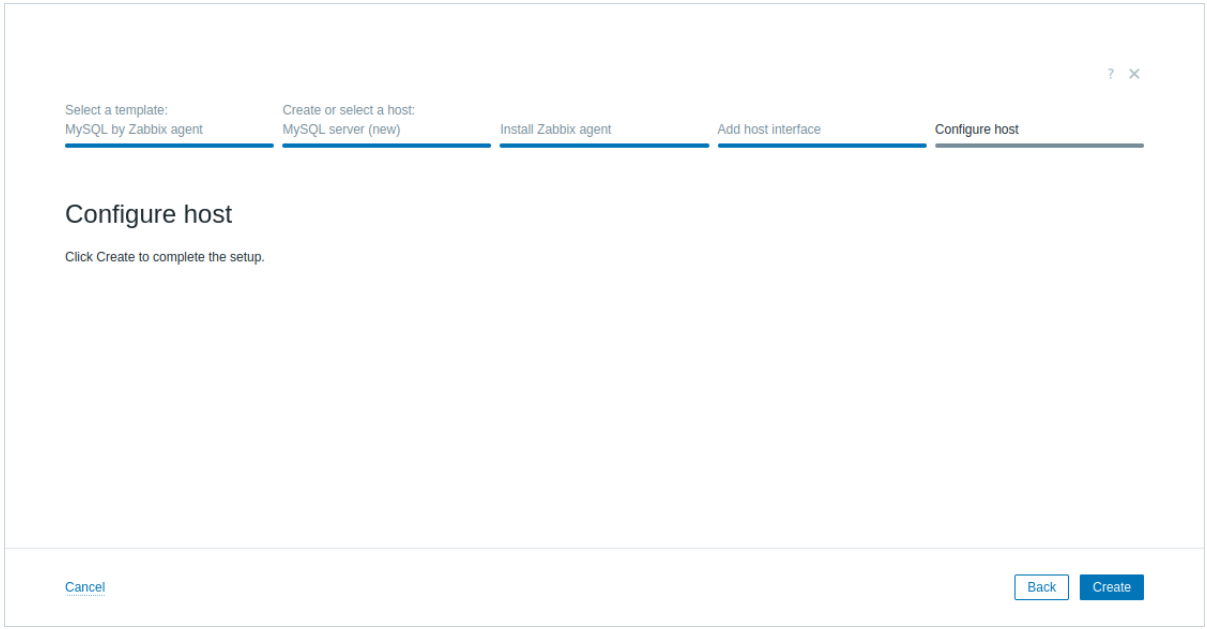
- Configuração manual do seu alvo de monitoramento (por exemplo, habilitar serviços específicos, criar usuários de serviço ou conceder permissões):



- Definição de macros de host—variáveis que controlam o comportamento do item, configurações de conexão, credenciais de autenticação e outros parâmetros:



Após concluir todas as etapas de configuração necessárias, clique em *Criar* para adicionar o host ao Zabbix (ou *Atualizar* para atualizar a configuração de um host existente):



Configuração concluída

Neste ponto, o Zabbix já está monitorando seu destino.

Para novos hosts, clique em *Concluir* para navegar até a seção Últimos dados e visualizar os dados mais recentes do seu host. Para hosts existentes, clique em *Concluir* para fechar o Assistente de Host e retornar à tela onde ele foi aberto.

2 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

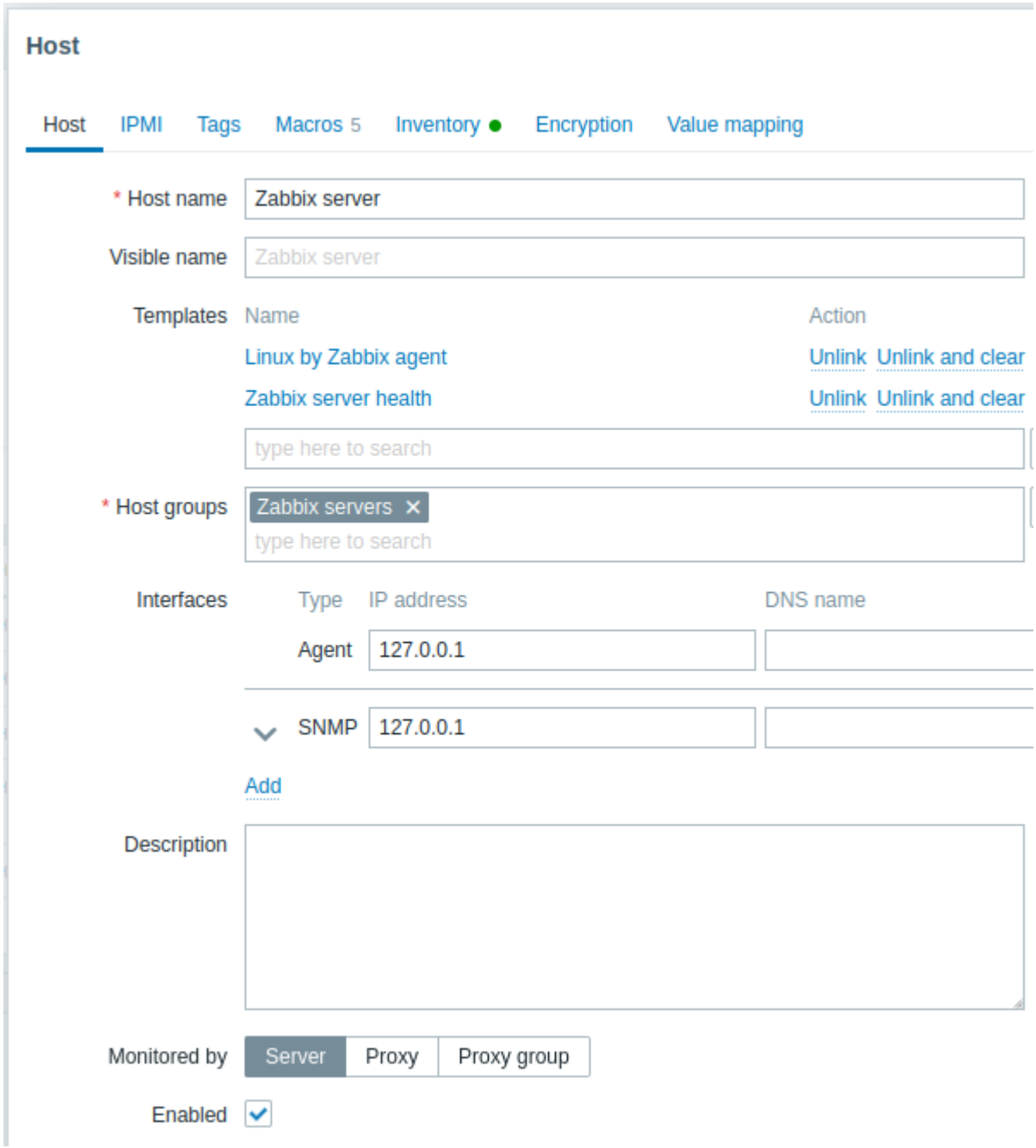
Você também pode usar o botão *Clonar* no formulário de configuração de um host existente para criar um novo host. Este host terá todas as propriedades do host existente, incluindo templates vinculados, entidades (items, triggers, etc.) desses templates, bem como as entidades diretamente anexadas ao host existente.

Observe que, quando um host é clonado, ele manterá todas as entidades do template como elas 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.

Como alternativa, você pode usar o Assistente de Host para configurar um host por meio de uma interface guiada, passo a passo.

Configuração

A aba **Host** contém atributos gerais do host:



Todos os campos obrigatórios são marcados com um asterisco vermelho.

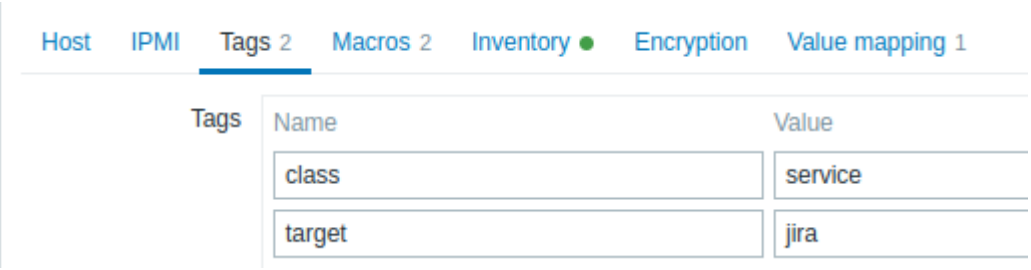
Parâmetro	Descrição
Nome do host	Insira um nome de host exclusivo. Alfanuméricos, espaços, pontos, traços e sublinhados são permitidos. No entanto, espaços à esquerda e à direita não são permitidos. *Nota:* Com o agent Zabbix em execução no host que você está configurando, o parâmetro *Hostname* no arquivo 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	Insira um nome visível exclusivo para o host. Se você definir este nome, ele será o visível em listas, mapas, etc., em vez do nome técnico do host. Este atributo possui 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 interfaces 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 o 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 o agent Zabbix, 161 para o agent SNMP, 12345 para JMX e 623 para IPMI.
Padrão	Marque o botão de opção para definir a interface padrão.
Descrição	Insira 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 prazo (aprox. 5 segundos) em que o agent Zabbix recebe a informação de que o host foi desabilitado. Durante esse breve intervalo, o host continuará coletando localmente dados 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. Ao desabilitar o host, seus items são imediatamente removidos do cache de histórico (exceto seus últimos valores, que são mantidos para logs).

A aba **IPMI** contém atributos de gerenciamento IPMI.

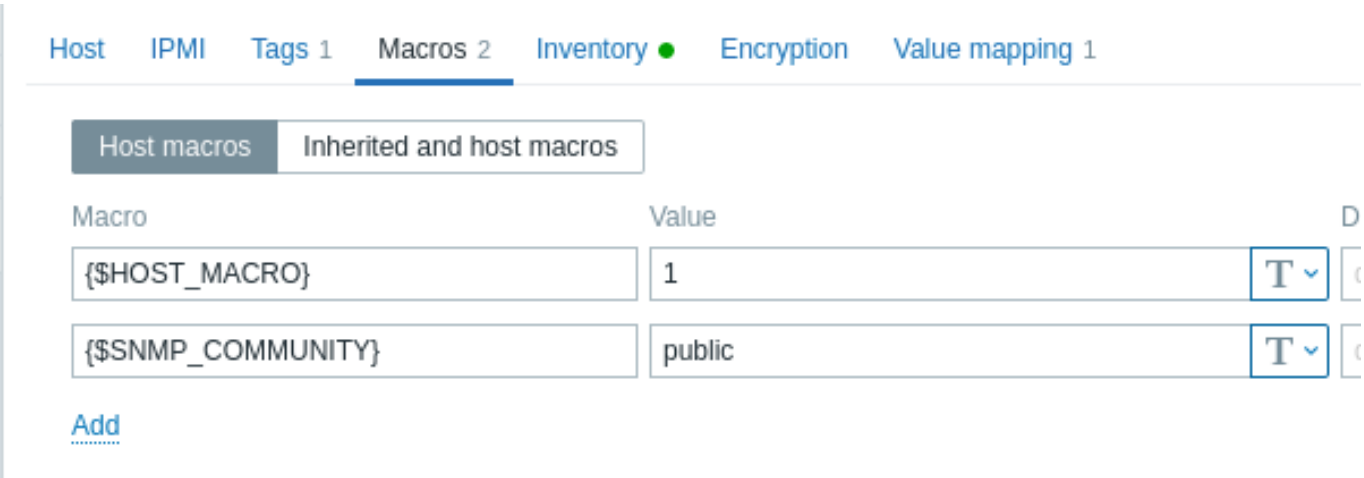
Parâmetro	Descrição
Algoritmo de autenticação	Selecione o algoritmo de autenticação.
Nível de privilégio	Selecione o nível de privilégio.
Nome de usuário	Nome de usuário para autenticação. Macros de usuário podem ser usadas.
Senha	Senha para autenticação. Macros de usuário podem ser usadas.

A aba **Tags** permite definir tags no nível do host. Todos os problemas deste host serão marcados com os valores inseridos aqui.

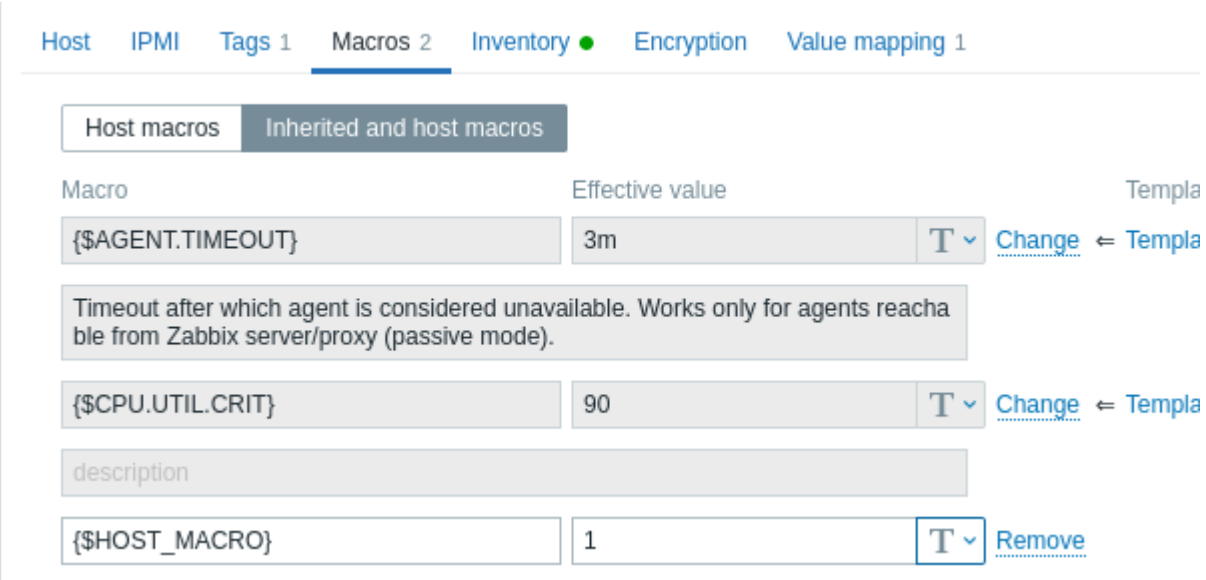


Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags.

A aba **Macros** permite definir macros de usuário no nível do host 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.

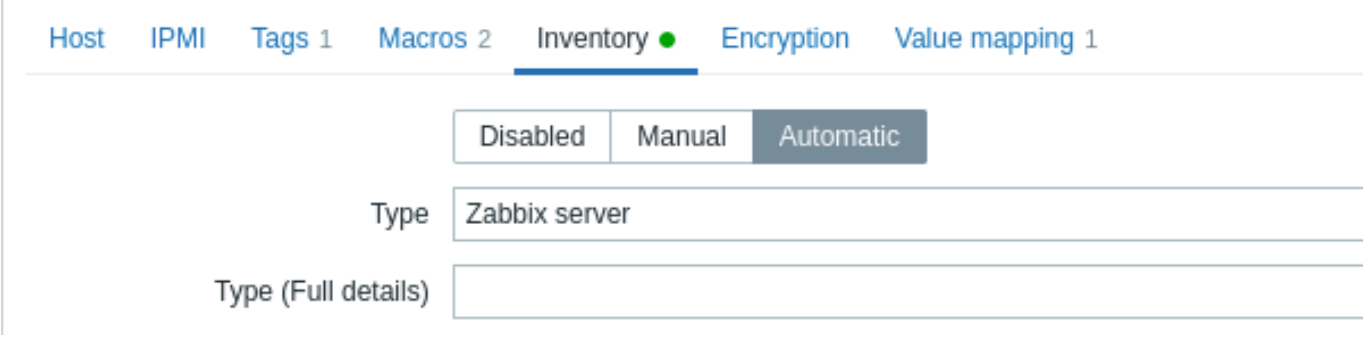


Você também pode visualizar aqui macros de usuário em nível de template e globais se selecionar a opção *Macros herdadas e do host*. É aqui que todas as macros de usuário definidas para o host são exibidas com o valor ao qual elas se resolvem, bem como sua origem.



Para conveniência, links para os respectivos templates e configuração de macro global são fornecidos. 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 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 do 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 múltiplas CAs. Se este campo estiver vazio, qualquer CA é 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 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

Mapeamento de valores

A aba **Mapeamento de valores** permite configurar uma representação amigável dos dados do item em mapeamentos de valores.

3 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.

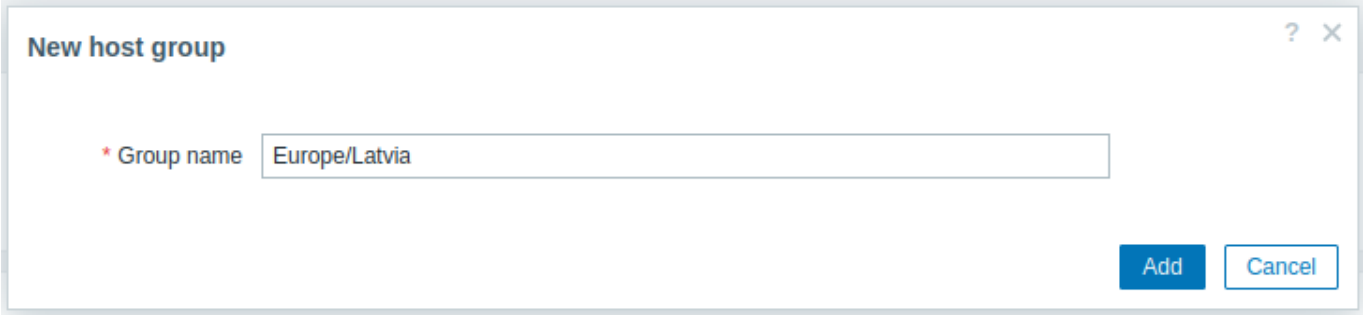
Configuração

Attention:
Apenas usuários Super admin podem criar grupos de host.

Existem duas opções para criar um grupo de host 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 host 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 host exclui apenas o grupo lógico, não os hosts do grupo. Não é possível excluir um grupo de host 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 pais aninhados `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

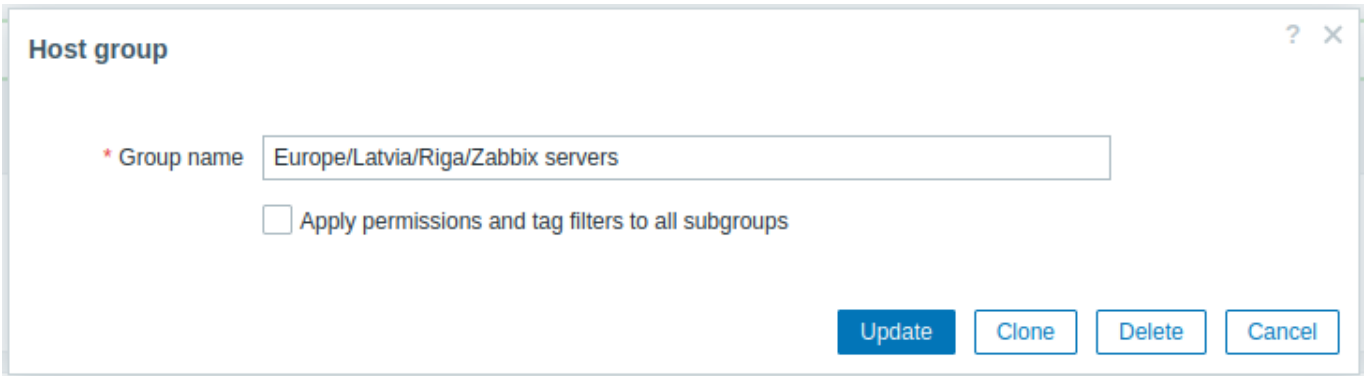
Ao criar um subgrupo, não é permitido usar barras à esquerda ou à direita, ou várias barras seguidas. O escape de '/' não é suportado.

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, criando `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, criando `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*.



Marcando esta caixa de seleção e clicando 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.

4 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 *Item irá preencher o campo do inventário do host*) na configuração do item.

Itens que são especialmente úteis para 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 por padrão para novos hosts é selecionado com base na configuração *Modo de inventário de host padrão* em *Administração → Geral → Outros*.

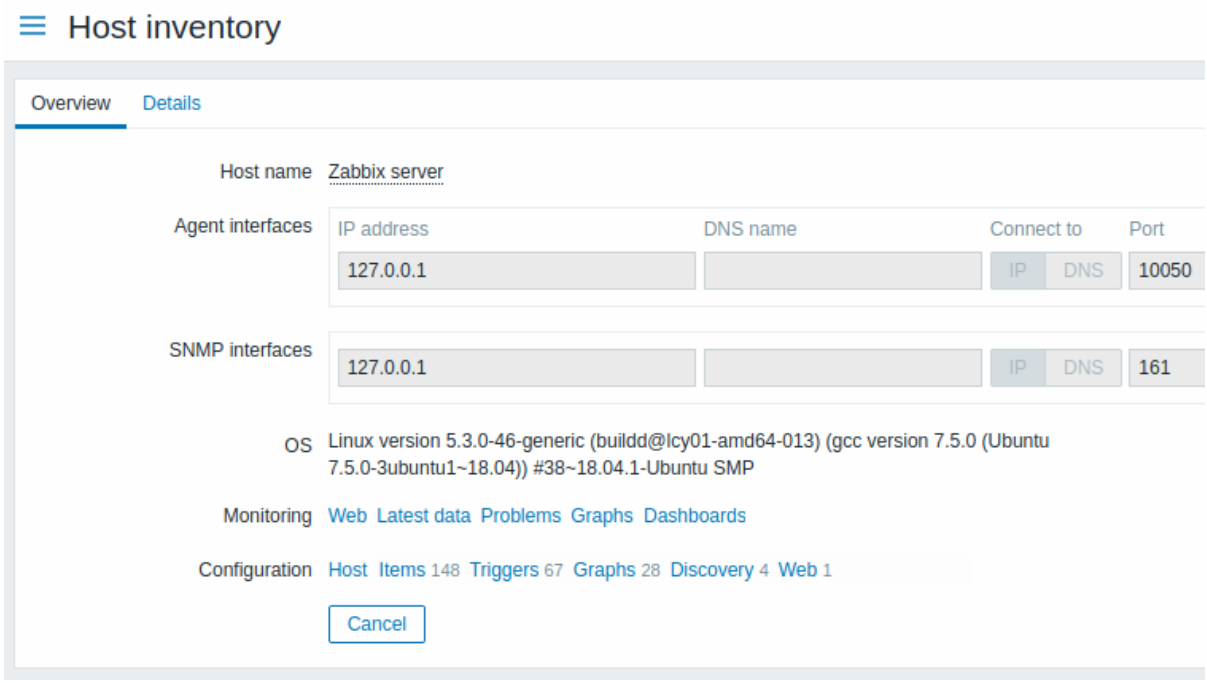
Para hosts adicionados por descoberta de rede ou ações de autorregistro, é possível definir uma operação *Definir modo de inventário do host* selecionando o modo manual ou automático. Esta operação substitui a configuração *Modo de inventário de host padrão*.

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*, *Items*, *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.

5 Atualização em massa

Visão geral

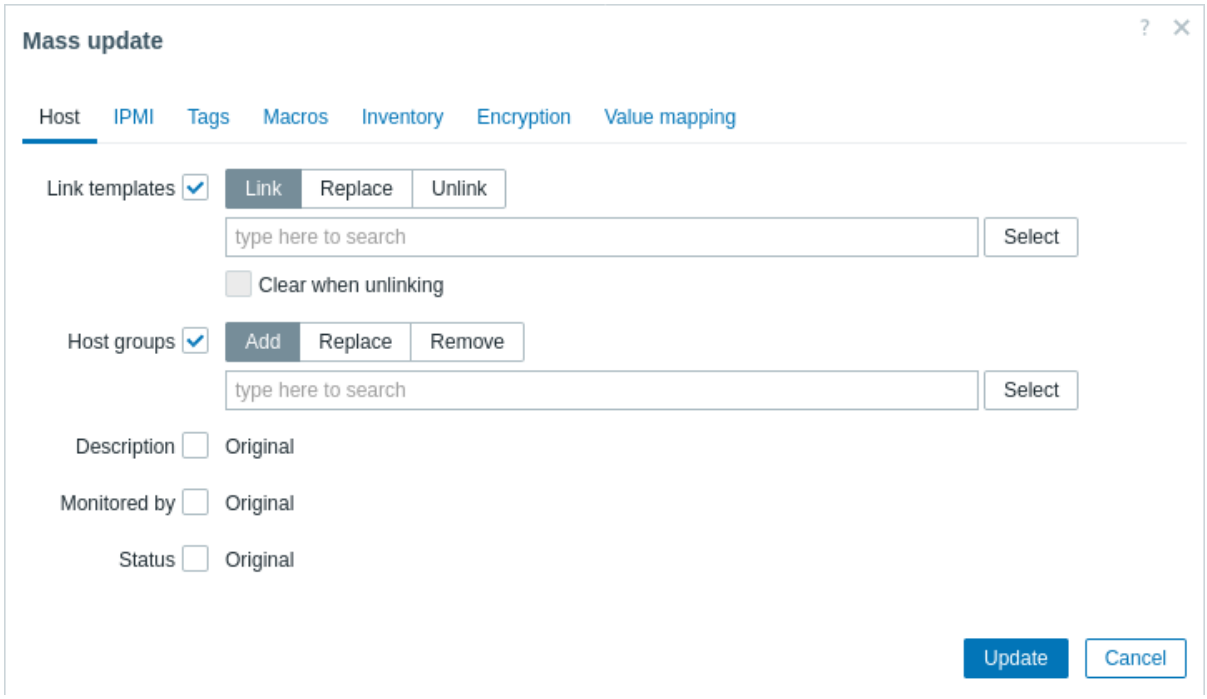
Às vezes, você pode querer alterar algum atributo para vários hosts de uma só vez. Em vez de abrir cada host individualmente para edição, você pode usar a função de atualização em massa para isso.

Usando atualização em massa

Para atualizar vários hosts em massa, faça o seguinte:

- Marque as caixas de seleção antes dos hosts que você deseja atualizar na lista de hosts
- Clique em *Atualizar em massa* abaixo da lista

- Navegue até a guia 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 vinculação de **template**:

- *Vincular* - especifique quais templates adicionais vincular
- *Substituir* - especifique quais templates vincular enquanto desvincula qualquer template que estava vinculado aos hosts anteriormente
- *Desvincular* - especifique quais templates desvincular

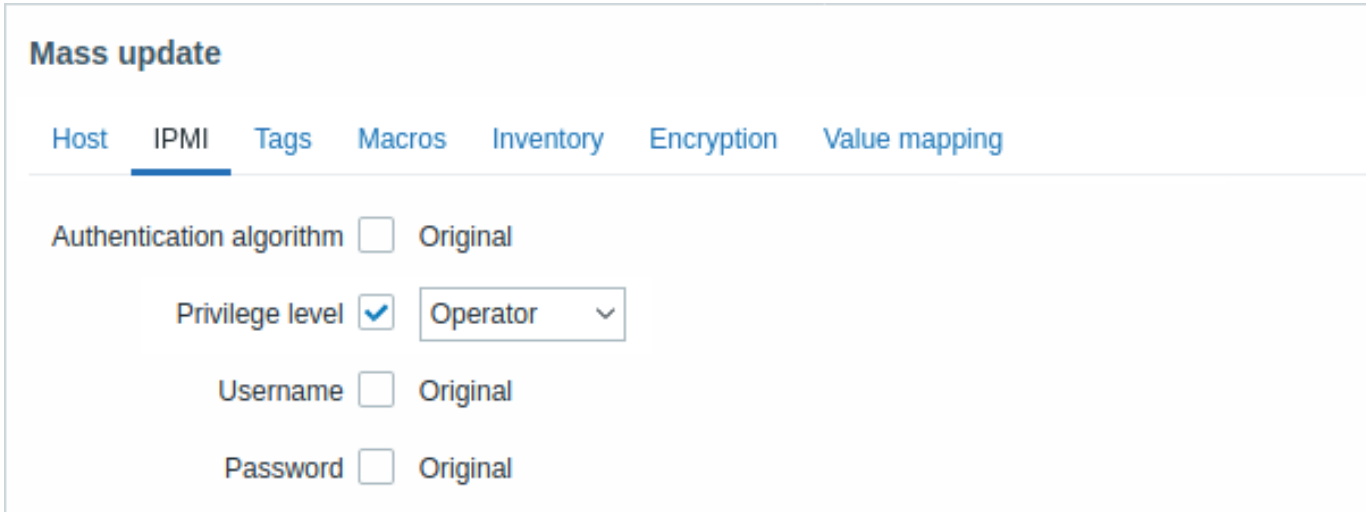
Para especificar os templates a serem vinculados/desvinculados, comece a digitar o nome do template no campo de preenchimento automático até que um menu suspenso apareça oferecendo os templates correspondentes. Basta rolar para baixo para selecionar o template necessário.

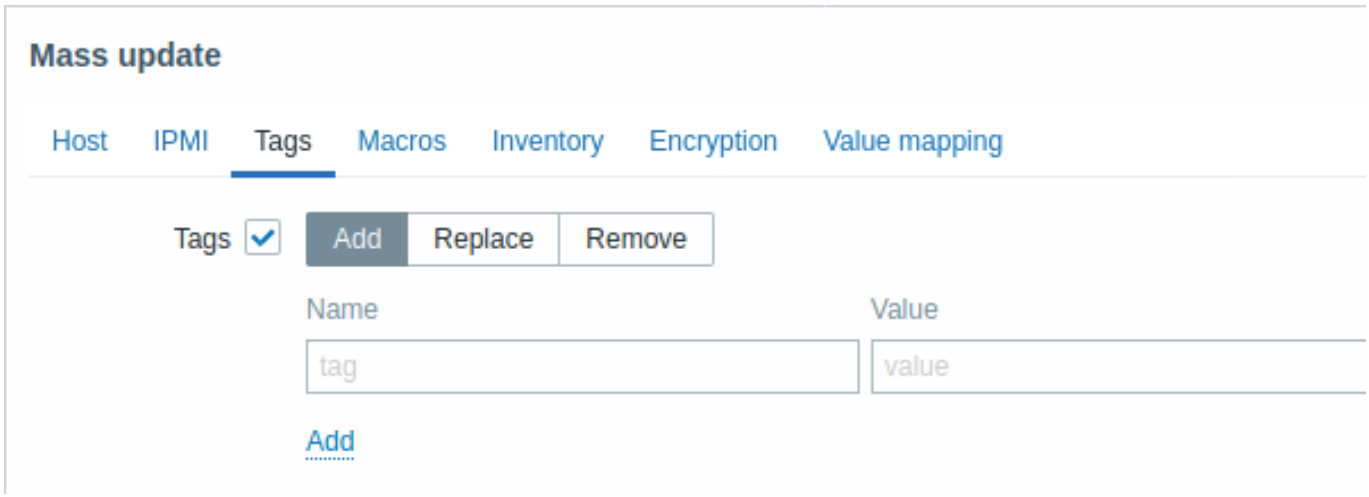
A opção *Limpar ao desvincular* permitirá não apenas desvincular quaisquer templates vinculados anteriormente, mas também remover todos os elementos herdados deles (itens, triggers, etc.).

As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de **grupo de hosts**:

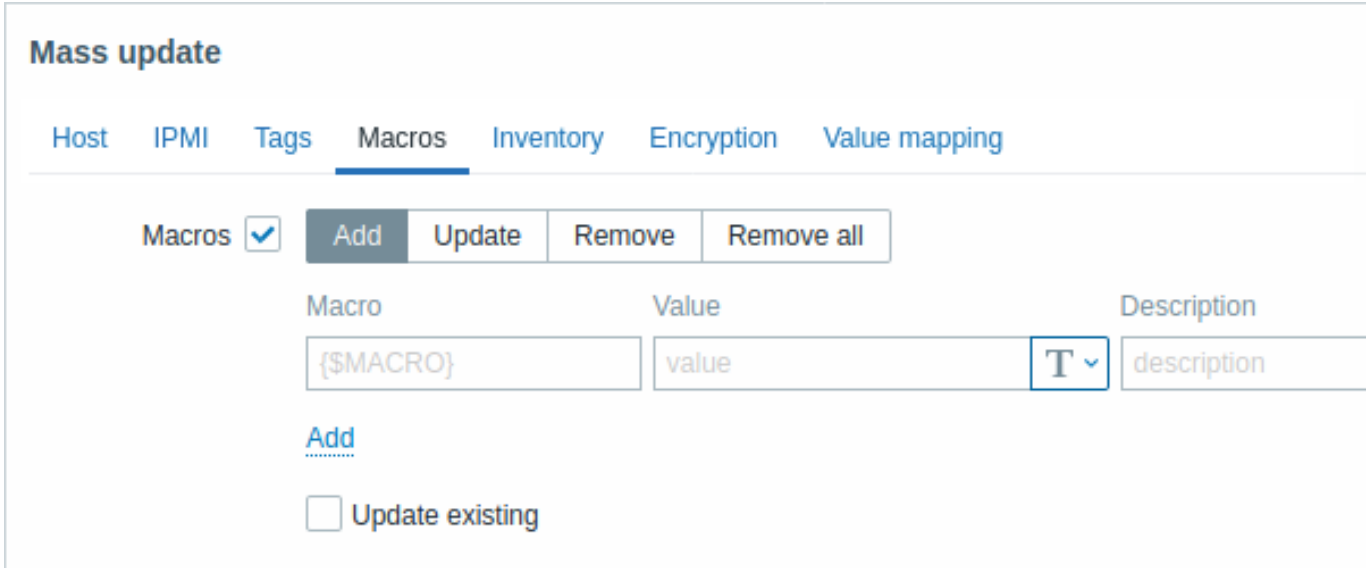
- *Adicionar* - permite especificar grupos de hosts adicionais dos existentes ou inserir grupos de hosts completamente novos para os hosts
- *Substituir* - removerá o host de quaisquer grupos de hosts existentes e os substituirá pelos especificados neste campo (grupos de hosts existentes ou novos)
- *Remover* - removerá grupos de hosts específicos dos hosts

Esses campos são de preenchimento automático - 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.

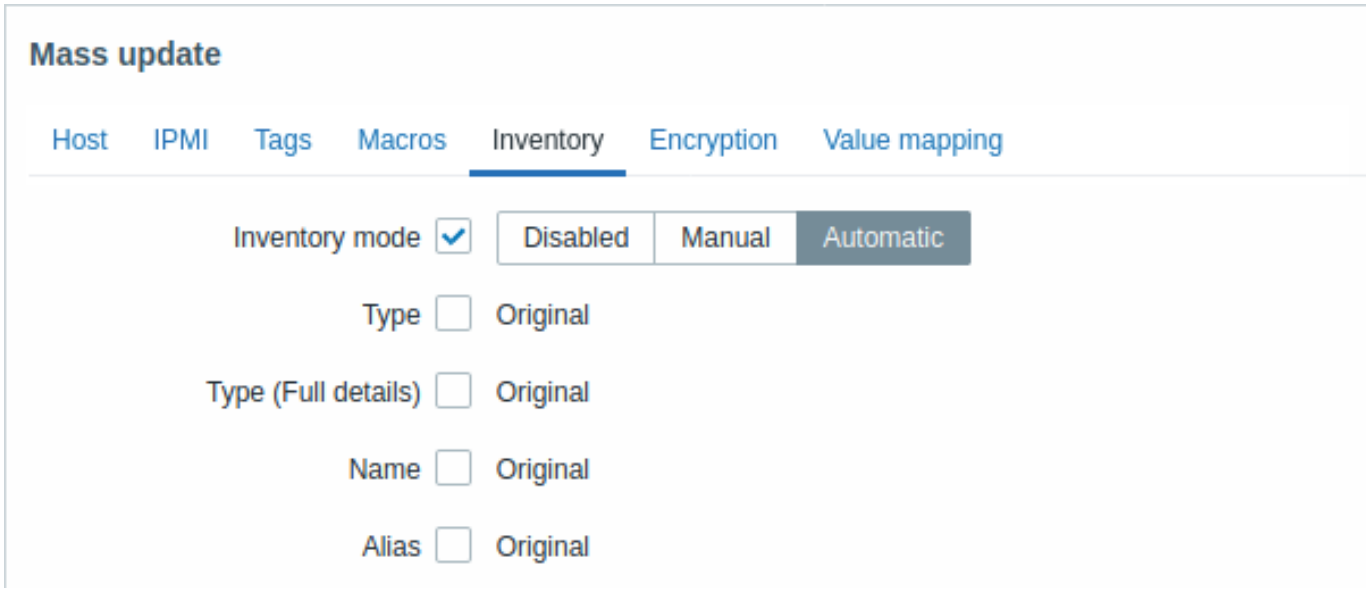


As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de macros:

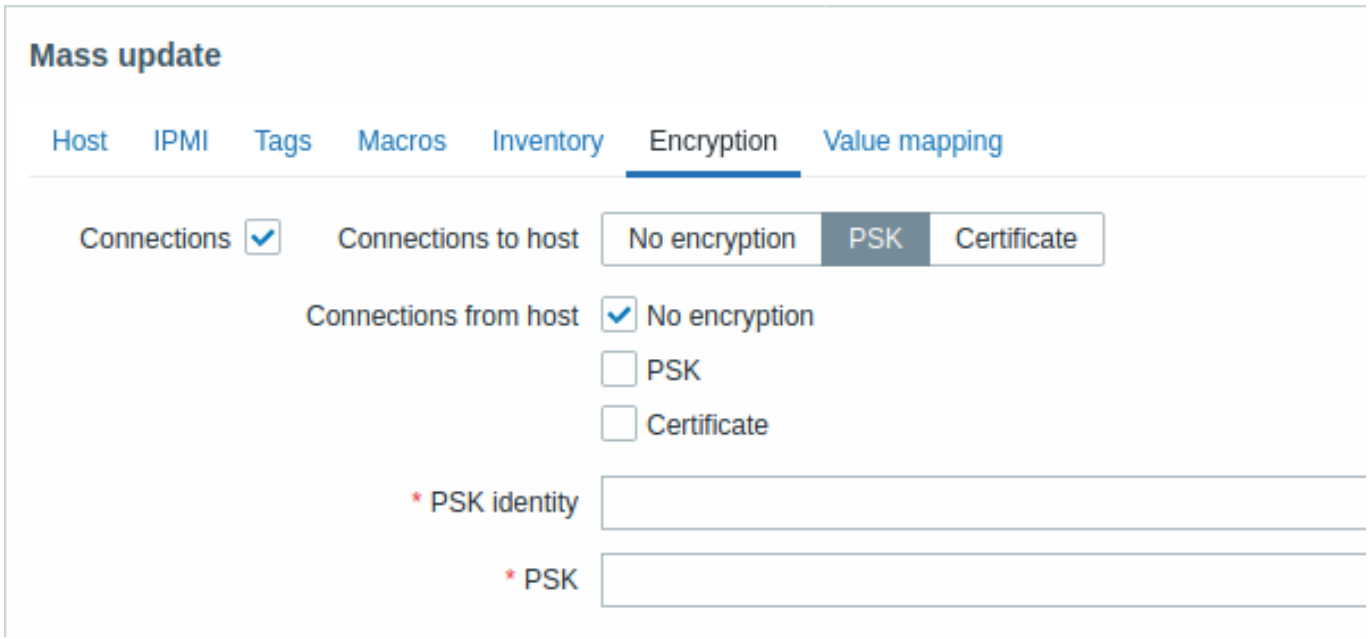
- *Adicionar* - permite especificar macros de usuário adicionais para os hosts. Se a caixa *Atualizar existentes* estiver marcada, o valor, tipo e descrição para o nome da macro especificada serão atualizados. Se desmarcada, se uma macro com esse nome já existir no(s) host(s), ela não será atualizada.
- *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 as 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,

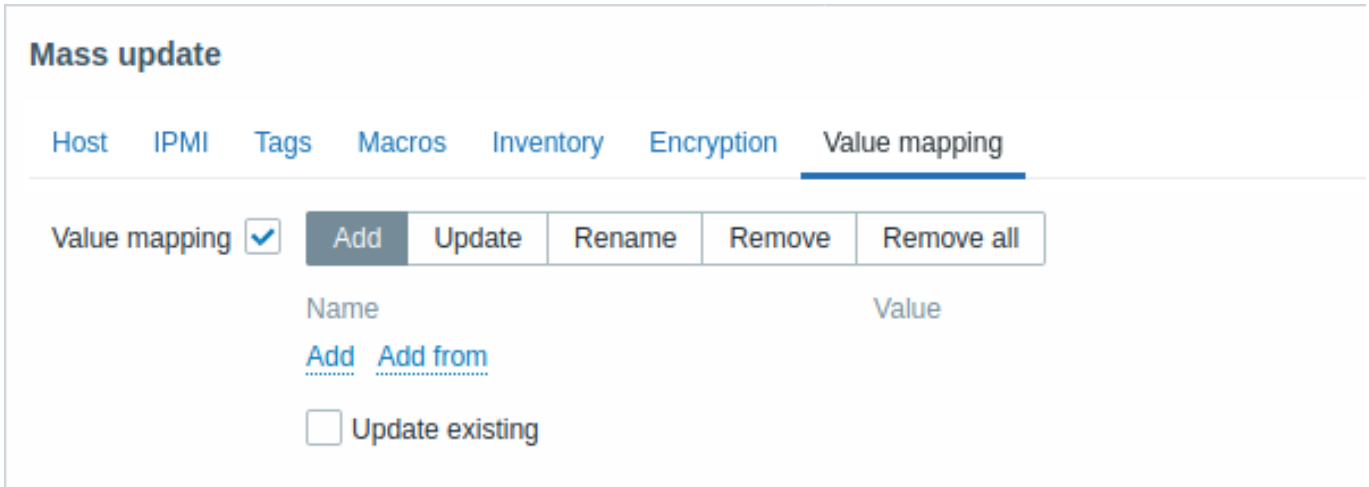
exceto as especificadas na lista, serão removidas. Se desmarcada, apenas as macros especificadas na lista serão removidas.

- *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 campos de inventário em massa, 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 um 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 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.

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 dado 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 dos últimos 5 minutos, enquanto net.if.in**[eth0]** mostrará o tráfego de entrada na interface ”eth0”.

> **Note:**
> Veja as seções individuais dos tipos de item para todos os tipos de item e chaves de item suportados.

Prossiga para criar e configurar um item.

1 Criando um item

Visão geral

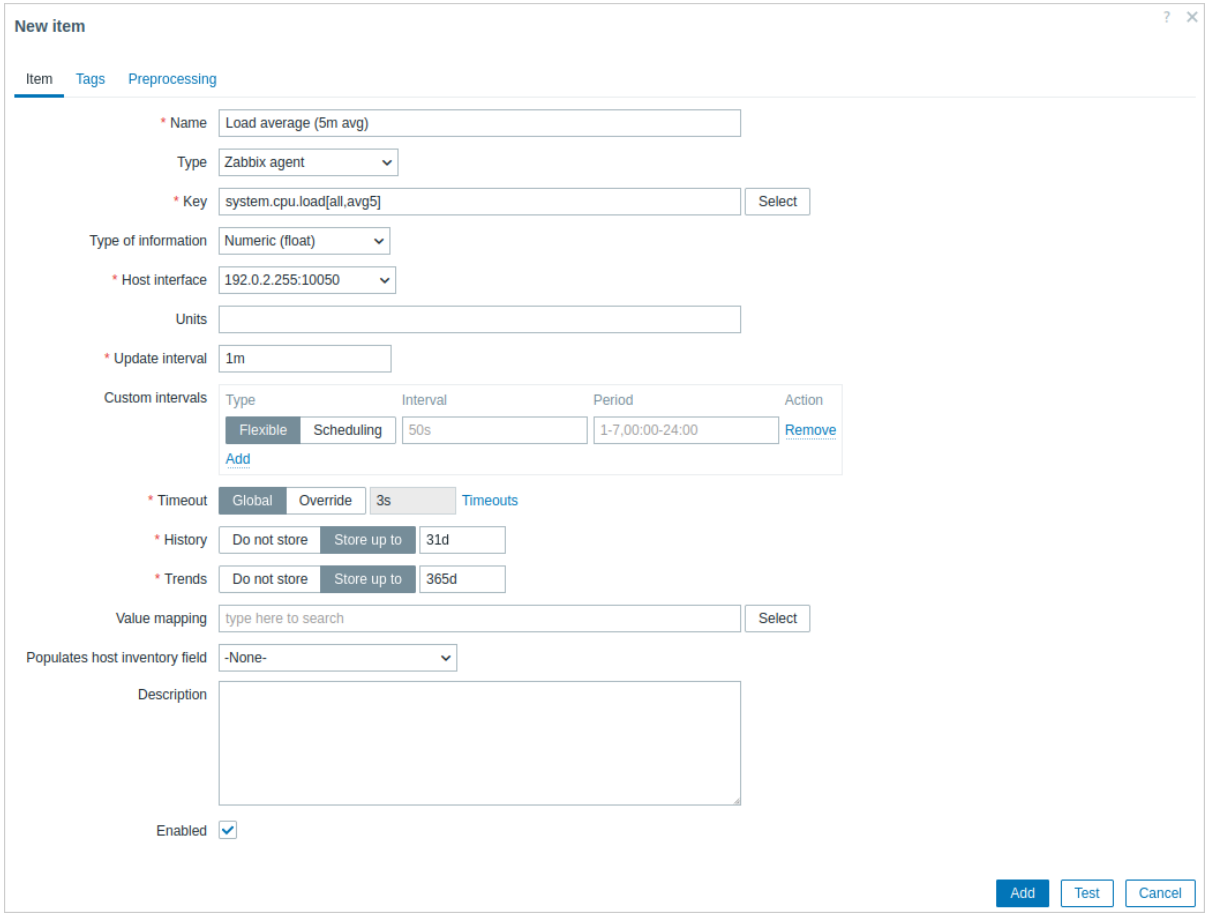
Para criar um item no frontend do Zabbix, faça o seguinte:

- 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 são 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 realizar 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). Receber valores em notação científica também é suportado. Ex: 1.23E+7, 1e308, 1.1E-4. **Caracter** - 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). Um número binário será resolvido para uma string estática "binary value" em *Últimos dados*; {ITEM.VALUE}, {ITEM.LASTVALUE} e macros de expressão serão resolvidos como UNKNOWN. 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.

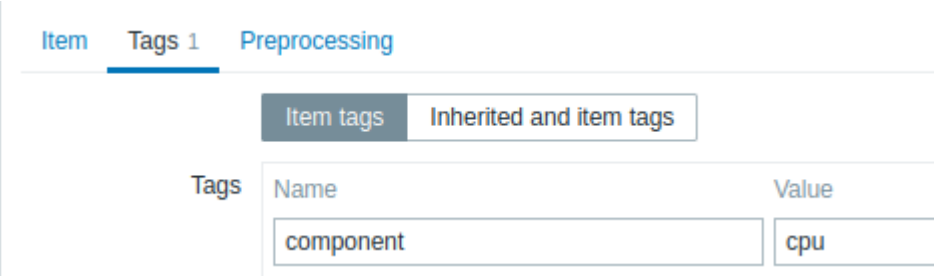
Parâmetro	Descrição
Unidades	Se um símbolo de unidade for definido, o Zabbix aplica pós-processamento ao valor do item recebido e o exibe com o sufixo de unidade especificado. Unidades suportadas com formatação especial (e exemplos de valor recebido → valor exibido): **B** - bytes (1024 → 1 KB) **Bps** - 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 deve ser *Numérico (sem sinal)*. Para outras unidades (como Hz, W, etc.), se o valor recebido exceder 1000, ele é 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 e conversão de unidades, 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 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 um intervalo personalizado (flexível ou agendado) existir com valor diferente de zero, o item será coletado durante a duração do intervalo personalizado. *Nota* que a primeira coleta do item após ele se tornar ativo ou após alteração do intervalo 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 coletado imediatamente pressionando o botão *Executar agora* botão.
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. 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** - 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 permissão 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 mestres 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 de *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 é exibida, ex: *Sobrescrito pelas configurações globais do housekeeper (1d)*. Recomenda-se manter os valores gravados 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 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 de *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 é exibida, ex: *Sobrescrito pelas configurações globais do housekeeper (7d)*. *Nota:* Manter tendências não está disponível para dados não numéricos - caracter, 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. Funciona com itens *Numérico (sem sinal)*, *Numérico (float)* e *Caracter*. Por exemplo, ”Estados de serviços 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 1 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	Insira uma descrição do item. Macros de usuário são suportadas.
Habilitado	Marque a caixa para habilitar o item para que ele seja processado. Quando você desabilita o item, ele é imediatamente removido do cache de histórico (exceto pelo seu último valor, que é mantido para logs).
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 do 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 e ir para o 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 de item

A aba **Pré-processamento** permite definir regras de transformação para os valores recebidos.

Testando

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 do 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.

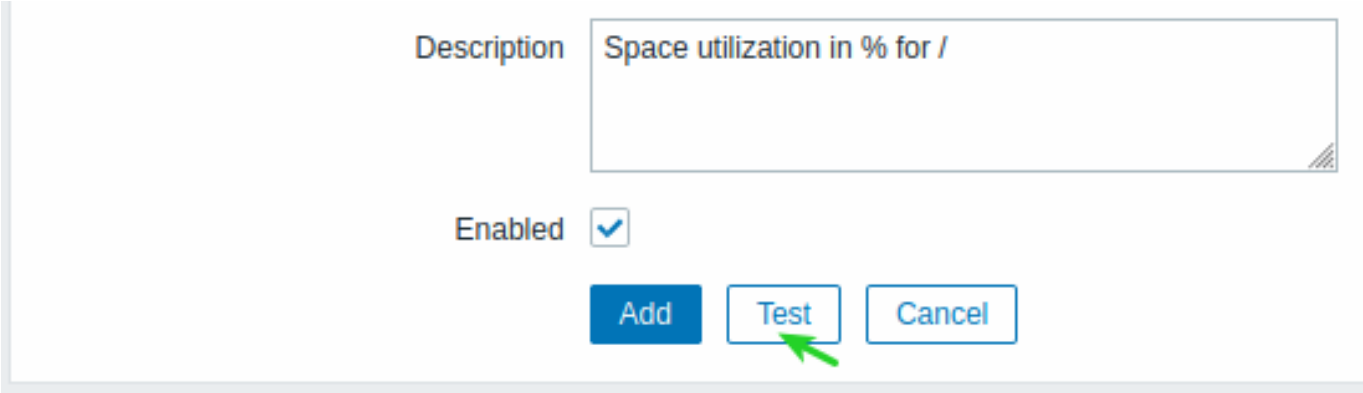
É possível testar um item e, se configurado corretamente, obter um valor real como retorno. O teste pode ocorrer mesmo antes de um item ser salvo.

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).

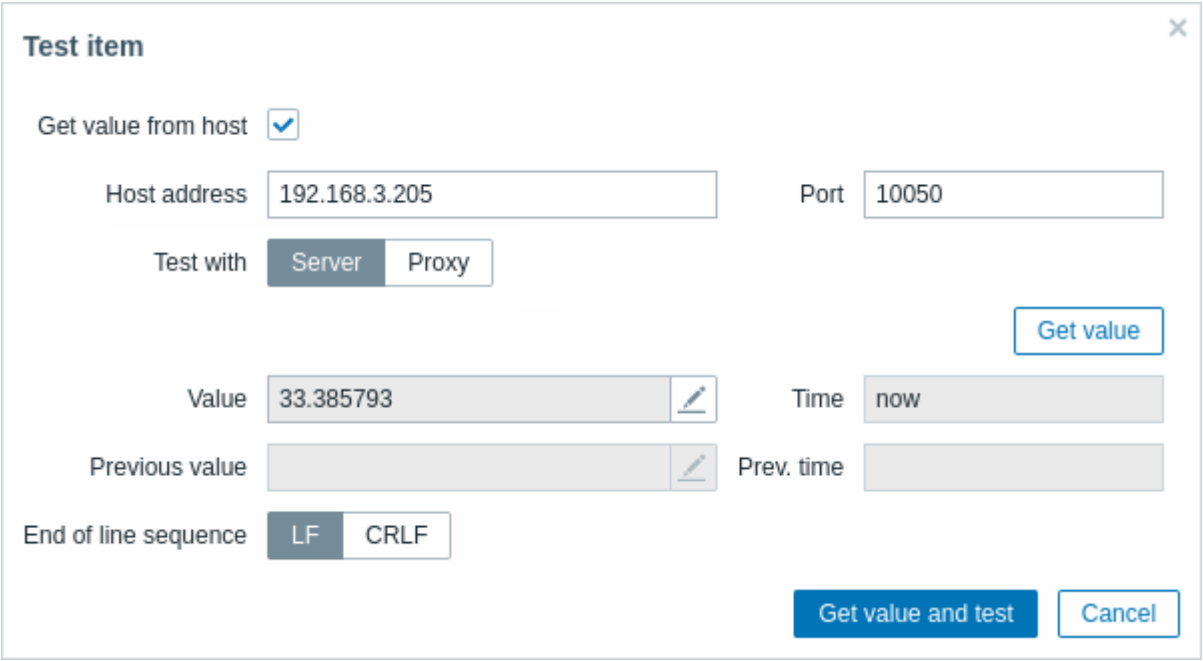


O formulário de teste de item possui campos para os parâmetros necessários do host (endereço do host, porta, testar com server/proxy (nome do proxy)) e detalhes específicos do item (como comunidade SNMPv2 ou credenciais de segurança SNMPv3). Esses campos são sensíveis ao contexto:

- Os valores são preenchidos automaticamente quando possível, ou seja, para itens que exigem um agent, obtendo as informações da interface 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 qualquer 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 proxy são desabilitados para itens calculados)

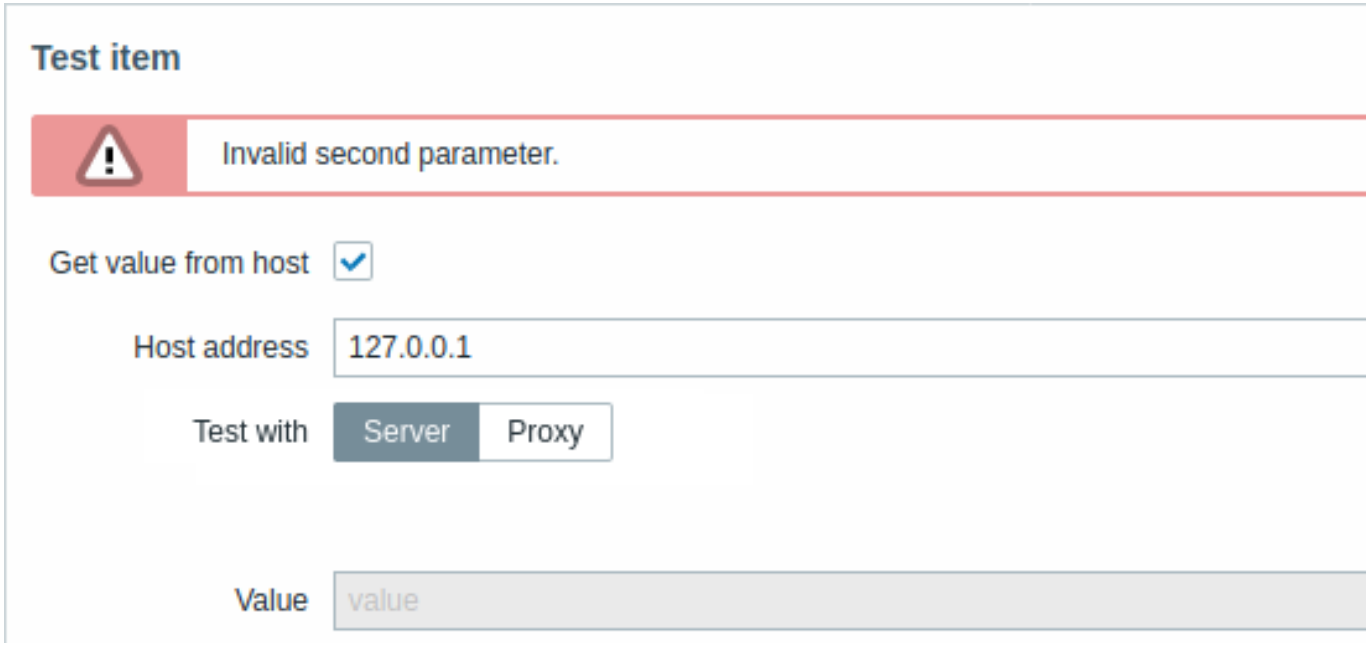
Para testar o item, clique em *Obter valor*. Se o valor for recuperado com sucesso, ele preencherá o campo *Valor*, movendo o valor atual (se houver) para o campo *Valor anterior* enquanto também calcula o campo *Tempo ant.*, ou seja, a diferença de tempo entre os dois valores (cliques) e tentando detectar 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 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 obter um novo valor do 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		
	Caractere	**Log**	**Texto**
MySQL	255 caracteres	65536 bytes	65536 bytes
PostgreSQL	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
SQLite (apenas Zabbix proxy)	65535	sem limite

Items não suportados

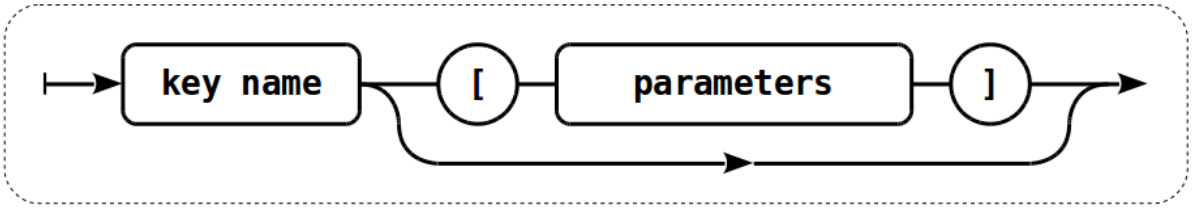
Itens não suportados são relatados com o status *Não suportado* e ainda são verificados novamente no seu *Intervalo de atualização* padrão.

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*. Esse é o comportamento esperado—um item muda para o status *Normal* somente quando recebe novos dados históricos válidos.

Assim como problemas, 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 somente ao receber novos dados. No entanto, se uma expressão de trigger incluir uma função de data e hora e/ou nodata(), a trigger será recalculada a cada 30 segundos (veja tempo de cálculo das 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 ou não parâmetros - conforme mostrado pelas duas linhas que podem ser seguidas.

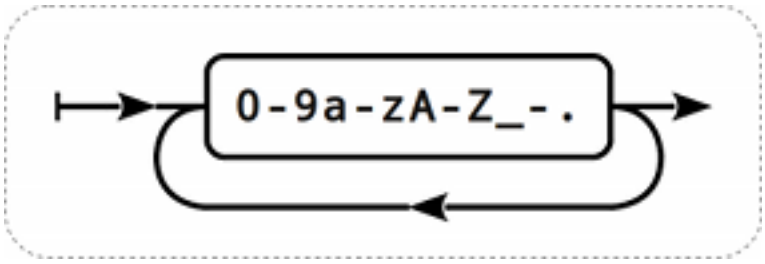
Nome da chave

O próprio nome da chave possui 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.

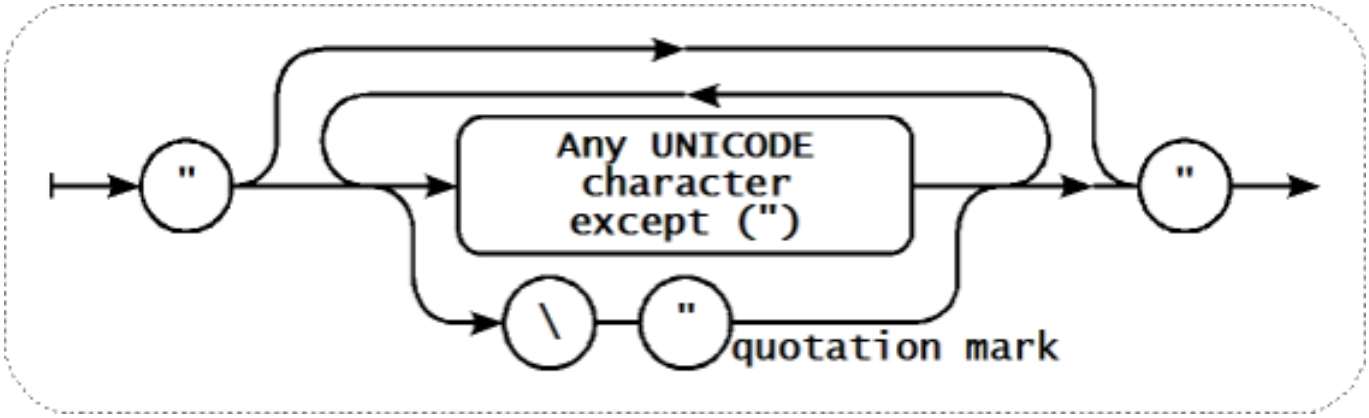
Cada parâmetro de chave pode ser uma string entre aspas, uma string sem aspas ou um array.

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.

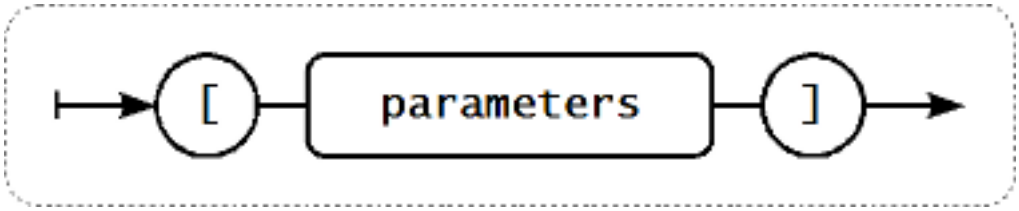
Attention:
Arrays de parâmetros multinível, por exemplo, `[a,[b,[c,d]],e]`, não são permitidos.

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 uma programação de verificação independente, que é executada 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' corresponderá, 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>`.

Os 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 corresponder a 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 em 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
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 os 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 do Zabbix ou items ativos de agent, é recomendável definir o fuso horário dos respectivos proxies ou agent igual ao do server do Zabbix, caso contrário a fila pode relatar atrasos de items incorretamente.

O fuso horário para proxy ou agent do Zabbix 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 de item recebidos antes de salvá-los no banco de dados. Essas transformações/etapas de pré-processamento são realizadas pelo Zabbix server ou proxy (se os itens 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 torna 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 itens 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 receber 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*. Consulte Criando um item para obter uma descrição detalhada dos parâmetros.

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	Compara o valor com a expressão regular e substitui pela saída desejada.	Texto
Substituir	Encontra a string de pesquisa e a substitui por outra (ou nada).	
Remover espaços	Remove caracteres especificados do início e do fim do valor.	
Remover espaços à direita	Remove caracteres especificados do final do valor.	
Remover espaços à esquerda	Remove 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 do 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 que um valor deve corresponder.	

Nome	Descrição	Tipo
Não corresponde à expressão regular	Especifica uma expressão regular que um 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 mudou.	Limitação
Descartar inalterado com heartbeat	Descarta um valor se ele não mudou 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 correspondente. 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 de expressões regulares para alguns exemplos existentes.

Substituir

Encontre a string de pesquisa e substitua 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.

Remover à esquerda

Remove os caracteres especificados do início do valor.

XML XPath

Extrai 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 específico ou definir uma mensagem de erro específica. 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 específico 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 a string hexadecimal como string hexadecimal não escapada (*note* que dicas de exibição ainda são aplicadas);
- **UTF-8 de hex-STRING** - converte a string hexadecimal para string UTF-8;
- **MAC de hex-STRING** - valida a string hexadecimal 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 maiores que 8 bytes, os bytes subsequentes serão ignorados.

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.

SNMP walk para JSON

Converte valores SNMP para JSON.

Especifique um nome de campo no JSON e o caminho SNMP OID correspondente. Os valores dos campos serão preenchidos pelos valores no caminho SNMP OID 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 SNMP OID;
- 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 maiores 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 de ponto flutuante, 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 uma etapa de pré-processamento com falha, 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 anterior.

Comentários:

- Esta etapa pode ser útil para medir um valor que cresce constantemente;
- 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 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.

Alteração por segundo

Calcula a velocidade de alteração do 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 cresce constantemente;
- Como esse cálculo pode produzir números de ponto flutuante, é recomendável 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, é recomendável 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 alteração por item ("Alteração simples" ou "Alteração 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, por exemplo, com o estouro (overflow) 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.

Booleano 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 TRUE e zero é considerado 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 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.

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 em 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 *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.

Corresponde à expressão regular

Especifique uma expressão regular que um valor deve corresponder.

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.

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

Verifica uma mensagem de erro em nível de aplicação usando uma expressão regular. 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.

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 do 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 *Personalizar 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 todas as etapas de pré-processamento serem executadas. 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. As 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.

Veja Verificações 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 sendo 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 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 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 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 *Change* e *Throttling*.
Hora anterior	Insira a hora do valor de entrada anterior para comparar. Somente para as etapas de pré-processamento *Change* 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 usadas, elas são listadas junto com seus valores. Os valores são editáveis para fins de teste, mas as alterações só serão salvas no contexto do 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 o botão *Testar* ser clicado. Os resultados dos testes são truncados para um tamanho máximo de 512KB quando enviados para o frontend. Os resultados dos testes podem ser copiados (não mais do que os 512KB truncados). 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. O resultado final pode ser copiado (não mais do que os 512KB truncados). 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. Os valores são perdidos ao atualizar a página.

O teste é feito 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 *Change* 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 secreta, 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 mapa de valor aplicado'.

Parâmetros específicos para obter um valor real de um host:

Parâmetro	Descrição
Obter valor do host	Marque esta caixa 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.

Parâmetro	Descrição
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.)	Consulte Configurando monitoramento SNMP para obter 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 do 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

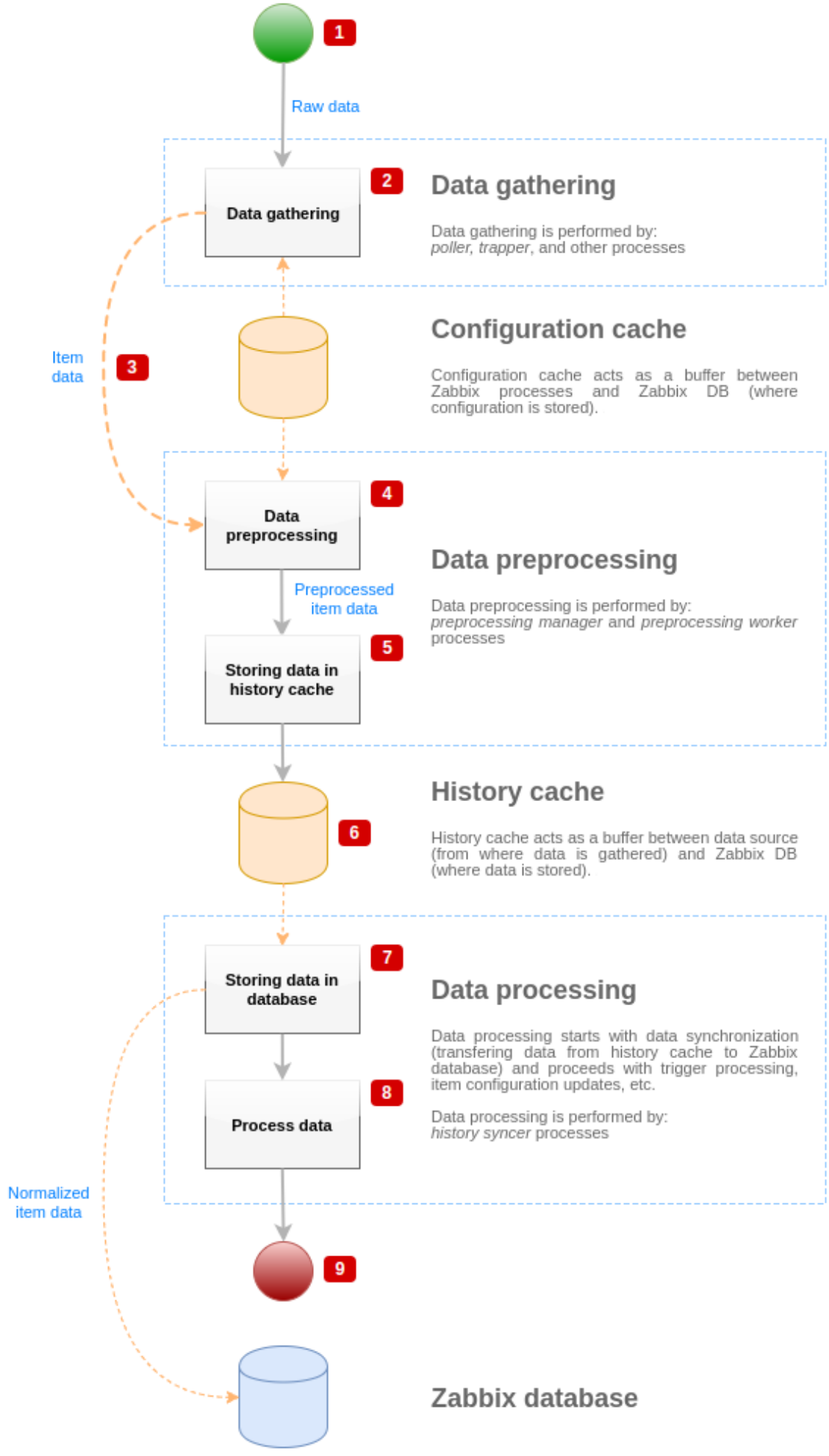
Visão geral

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 gerenciador de pré-processamento juntamente com os workers de pré-processamento que executam as etapas de pré-processamento. Todos os valores com pré-processamento, 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 do valor do item

Para visualizar o fluxo de dados desde a fonte 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 local de dados 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 itens 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 itens 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 de histórico do cache local de dados do gerenciador de pré-processamento estão sendo gravados no 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.

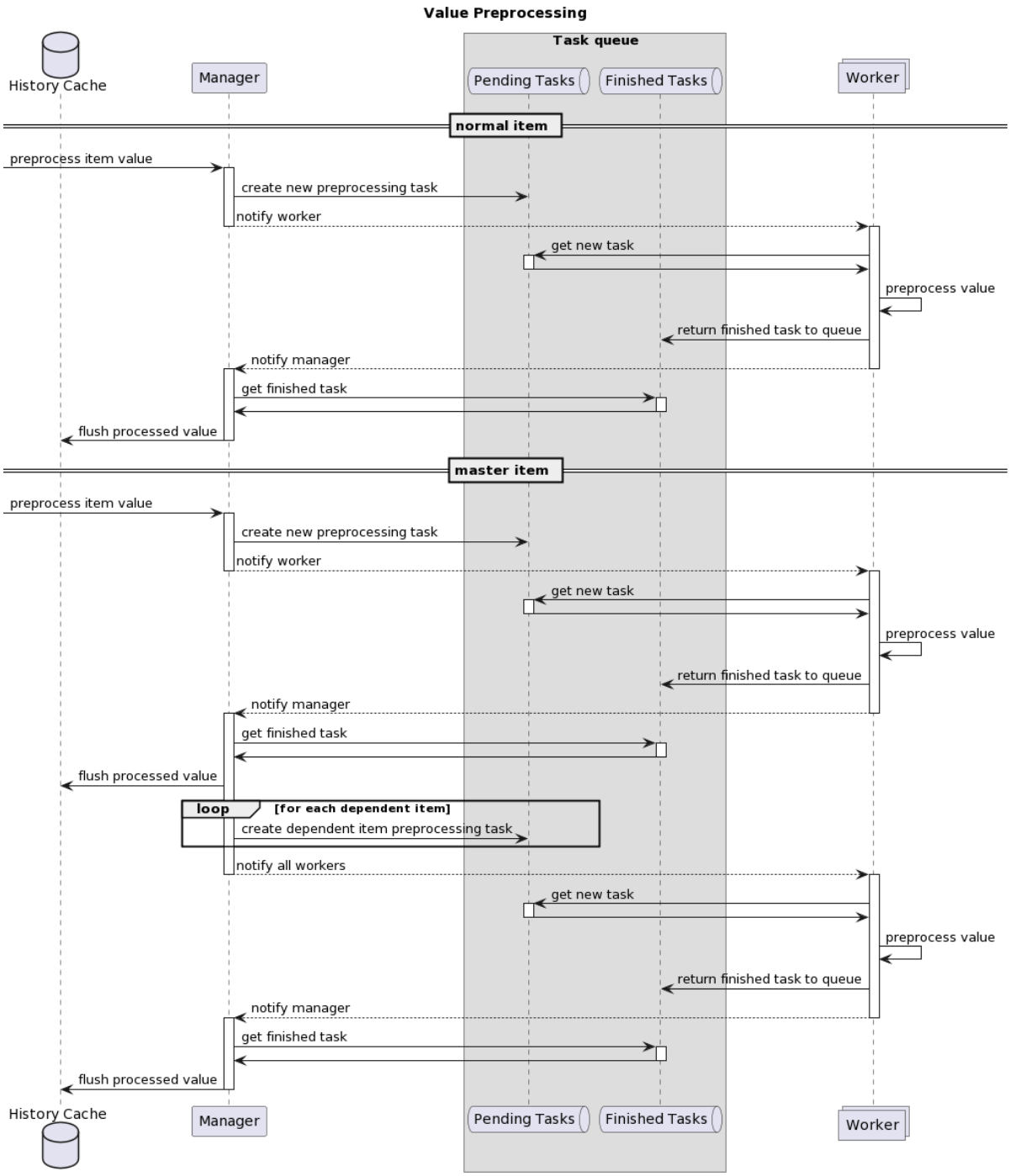
Note:
Um item pode mudar seu estado para NÃO SUPORTADO se a normalização dos dados falhar (por exemplo, quando um valor textual não pode ser convertido para número).

- Os dados coletados estão sendo processados - triggers são verificados, 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 do valor do item.

Pré-processamento de valor de item

O pré-processamento de 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 de LLD. Caso contrário, o valor do item é passado para o gerenciador de pré-processamento usando um mecanismo IPC baseado em socket UNIX.
- 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 execução bem-sucedida 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 de LLD.
- Se houver itens dependentes para o item processado, os itens dependentes são adicionados à fila de pré-processamento com o valor do item mestre 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 é ligeiramente simplificado ao ignorar o cache de pré-processamento.

Fila de pré-processamento

A fila de pré-processamento é organizada como:

- 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 múltiplos 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 contagem média de etapas para o pré-processamento de items, etc. Trabalhadores insuficientes podem levar ao alto uso de memória. Para solucionar o uso excessivo de memória em sua instalação do Zabbix, consulte Perfilando uso excessivo de memória com tcmalloc.

Mas assumindo que não há operações pesadas de pré-processamento, como análise de grandes blocos 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 lote) 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, atingem a 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 em 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 a partir 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 utiliza 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 tiver sido 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 aba *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 do item recuperado 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 obtido.

Quando um processo 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 aba *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 (inclusive); 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 valor mínimo 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 que suporta 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 agregadas do 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 produzem 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 preservar a ordem original das propriedades for essencial, 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": "Next day delivery in local town",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Printing and assembling book in A5 format",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Various restoration methods",
      "active": false,
      "methods": [
        {
          "description": "Chemical cleaning",
          "price": 46
        },
        {
          "description": "Pressing pages damaged by moisture",
          "price": 24.5
        },
        {
          "description": "Rebinding torn book",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Store is closed",
  "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”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinido	[”Printing and assembling book in A5 format”, ”Rebinding torn book”]
`$..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	1002

1 Escape de caracteres especiais de valores de macros LLD em JSONPath

Quando macros de descoberta de baixo nível são usadas no pré-processamento JSONPath e seus valores são resolvidos, as seguintes regras de escape de caracteres especiais são aplicadas:

- apenas barra invertida (\) e aspas duplas (") são consideradas para escape;
- se o valor da macro resolvida contiver esses caracteres, cada um deles será 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 o caractere especial seguinte 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 em 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 string), com algumas exceções:

- retornar um valor undefined resultará em erro;
- retornar um 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 em código JavaScript. Se um script contiver macros de usuário, 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 precisam 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. 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 agente 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 workers 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 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 for definido, HttpRequest definirá Content-Type como application/json se os dados enviados estiverem em 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 invertidas 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 Duktape:

- btoa(data) - codifica os dados em uma string base64;
- atob(base64_string) - decodifica a string base64 como um buffer Uint8Array.

```
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 o hash HMAC como uma string formatada em hexadecimal; os tipos de hash MD5 e SHA256 são suportados; 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 é gerado;
 key - a chave privada. Ela deve corresponder ao 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 quebras 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).
 OpenSSL ou GnuTLS é usado para calcular as assinaturas. Se o Zabbix foi compilado sem nenhuma dessas bibliotecas de criptografia, um erro será gerado ('missing OpenSSL or GnuTLS library').

2 Objetos JavaScript de item de navegador

Visão geral

Esta seção descreve as adições do Zabbix à linguagem JavaScript implementadas 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 na criação e encerrando-a na destruição. 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, CSS selector/link text/partial link text/tag name/Xpath) Estratégia de localização; `selector` - (string) Seletor do elemento usando a estratégia de localização especificada.
`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, CSS selector/link text/partial link text/tag name/Xpath) Estratégia de localização; `target` - (string) Seletor do elemento usando a estratégia de localização especificada.
`getCookies()`	Retorna um array de objetos `Cookie`.

Método	Descrição
`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) Altura em pixels; `y` - (string) Largura em pixels.
`setScriptTimeout(timeout)`	Define o tempo limite de carregamento do script. Parâmetros: `timeout` - (integer) Valor do tempo limite em milissegundos.
`setSessionTimeout(timeout)`	Define o tempo limite da sessão (carregamento da página). Parâmetros: `timeout` - (integer) Valor do tempo limite em milissegundos.
`setElementWaitTimeout(timeout)`	Define o tempo limite da estratégia de localização do elemento (implícito). Parâmetros: `timeout` - (integer) 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. 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, alterna 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 `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 métodos, 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.

Alerta

O objeto `Alert` representa um alerta de página da 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 pelo 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 obtenção 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 of objects	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 of objects	Array de estatísticas do tipo mark/measure.
marks	array of objects	Índices de snapshot de desempenho marcados.
name	string	Nome da marca de snapshot de desempenho.
index	integer	Índice do snapshot de desempenho no array details.

6 Pré-processamento de CSV para JSON

Visão geral

Nesta etapa de pré-processamento, é possível converter dados de arquivos CSV para o formato JSON. É suportado em:

- items (prototipagem de item)
- regras de descoberta de baixo nível

Configuração

Para configurar uma etapa de pré-processamento de CSV para JSON:

- Vá para a aba Pré-processamento na configuração do item/regra de descoberta
- Clique em *Adicionar*
- Selecione a opção *CSV para JSON*

O primeiro parâmetro permite definir um delimitador personalizado. Observe que, se a primeira linha da entrada CSV começar com "Sep=" e for seguida por um único caractere UTF-8, esse caractere será usado como delimitador caso o primeiro parâmetro não esteja definido. Se o primeiro parâmetro não estiver definido e um delimitador não for obtido da linha "Sep=", então uma vírgula será usada como separador.

O segundo parâmetro opcional permite definir um símbolo de aspas.

Se a caixa de seleção *Com linha de cabeçalho* estiver marcada, os valores da linha de cabeçalho serão interpretados como nomes de coluna (consulte Processamento de cabeçalho para mais informações).

Se a caixa de seleção *Personalizar em caso de falha* estiver marcada, o item não ficará como não suportado em caso de falha na etapa de pré-processamento. Além disso, opções personalizadas de tratamento de erro podem ser definidas: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

Processamento do cabeçalho

A linha de cabeçalho do arquivo CSV pode ser processada de duas maneiras diferentes:

- Se a caixa de seleção *Com linha de cabeçalho* estiver marcada - os valores da linha de cabeçalho são interpretados como nomes de coluna. Neste caso, os nomes das colunas devem ser únicos e a linha de dados não deve conter mais colunas do que a linha de cabeçalho.

- Se a caixa de seleção *Com linha de cabeçalho* não estiver marcada - a linha de cabeçalho é interpretada como dados. Os nomes das colunas 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 caractere de aspas dentro de um campo entre aspas na entrada deve ser escapado precedendo-o com outro caractere de aspas.

Processando a 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 da 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"
    }
]
```

3 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 necessários.

Os seguintes tipos de item são atualmente 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 através de parâmetros de usuário ou módulos carregáveis.

Algumas verificações são realizadas apenas pelo server Zabbix (como monitoramento sem agent), enquanto outras requerem 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 Agent Zabbix

Visão geral

Esta seção fornece detalhes sobre as chaves de item que usam 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 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.

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 atualmente abertos.	
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.	
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 de 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.	

Chave do item	Descrição	Grupo de item
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 a porcentagem de inodes.	
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 items monitorados na fila que estão atrasados no server ou proxy Zabbix remotamente.	

Plataformas suportadas

Exceto onde especificado de forma diferente nos detalhes do item, os items do agent (e todos os parâmetros) são suportados em:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Muitos items do agent também são suportados no **Windows**. Veja a página item do agent do Windows para 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: *Integer*.
 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: *Inteiro*.
 Plataformas suportadas: Linux (o item pode funcionar em outras plataformas do tipo 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 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;
- 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 de um registro de log:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #este item irá
```

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 antigas para obter as linhas mais recentes 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>]

 O 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 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** - 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, relê se o horário de modificação ou tamanho mudar (padrão),
mtime-noreread - registros não exclusivos, relê apenas se o tamanho mudar (ignora a alteração do horário 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 no último horário de modificação dos arquivos;
- Observe que logrt foi projetado para funcionar com um arquivo de log ativo atualmente, 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 pegar muitos arquivos, isso pode levar ao esgotamento da memória e a uma falha no monitoramento.
- Se `output` for deixado vazio - 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 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["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 do 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: *Inteiro*.
 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 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, relê se o tempo de modificação ou tamanho mudar (padrão),
mtime-noreread - registros não exclusivos, relê apenas se o tamanho mudar (ignora 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 cheque ativo;
- 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}$",,,100] #este item contará o número de correspondências
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","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: *string JSON*.
 Plataformas suportadas: Linux.

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.

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 DNS falhou (o servidor DNS não respondeu ou retornou um erro); 1 - resolução DNS bem-sucedida.
 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 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.

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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 utilize 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 utilize o Zabbix agent 2) - o tempo limite para a solicitação em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que utilize 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 Zabbix agent no Windows, Zabbix agent 2 em todos os SO), *HINFO*, *MINFO*, *TXT*, *SRV*

- Para consultas DNS reversas (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, por favor, use nomes codificados em IDNA.
- 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 DNS. Valor de retorno: uma string de caracteres com o tipo de informação requerida.
 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 DNS reversas (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: *string JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

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. 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. 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. 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 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.

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 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 daemon 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 aguardando uma resposta do serviço).
 Veja 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.

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 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.

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), caso o agent tenha sido compilado em um Solaris sem suporte a zonas, mas esteja rodando em um Solaris mais novo onde 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.

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: *string 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 para Linux).

Exemplos:

```
proc.get[zabbix_server,zabbix,,process] #lista de todos os processos zabbix_server em execução sob o usuár
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 em execução sob o usuário zabbix,
```

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 (padrão é *todos os processos*);
- **user** - o nome do usuário (padrão é *todos os usuários*);
- **mode** - valores possíveis: *avg*, *max*, *min* ou *sum* (padrão);
- **cmdline** - filtro por 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 para 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.

Parâmetros:

- **name** - o nome do processo (padrão: *todos os processos*);
- **user** - o nome do usuário (padrão: *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;
- Quando este item é invocado pela 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) caso o agent tenha sido compilado em um Solaris sem suporte a zonas, mas esteja rodando em um Solaris mais novo onde 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] #número de processos rodando sob o usuário mysql
proc.num[apache2,www-data] #número de processos apache2 rodando sob o usuário www-data
proc.num[,oracle,sleep,oracleZABBIX] #número de processos em estado sleep rodando sob Oracle e contendo or
```

sensor[device,sensor,<mode>]

 Leitura de sensor de hardware.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, OpenBSD.

Parâmetros:

- **device** - o nome do dispositivo, pode ser uma expressão regular se o modo for omitido;
- **sensor** - o nome do sensor, pode ser uma expressão regular se o modo for omitido;
- **mode** - valores possíveis: *avg*, *max* ou *min* (se este parâmetro for omitido, device e sensor são tratados literalmente).

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.

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: *string 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.

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.

Parâmetros:

- **cpu** - *<número da CPU>* ou *all* (padrão);
- **type** - valores possíveis: *user* (padrão), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (no OpenBSD), *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>]

 As informações 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 nenhum parâmetro).

Exemplo:

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 A listagem de 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 plataformas suportadas.

Parâmetros:

- **type** - valores possíveis: *utc* - (padrão) o tempo desde a Época (00:00:00 UTC, 1 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 apenas como uma verificação passiva.

Exemplo:

```
system.localtime[local] #crie um item usando esta chave e, em seguida, use-o para exibir a hora do host no
```

system.run[command,<mode>]

 Executa o comando especificado no host.
 Valor de retorno: resultado do *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 juntamente 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 do usuário (float)
sy - tempo do sistema (float)
id - tempo ocioso (float)
wa - tempo ocioso durante o qual o sistema tinha solicitações de E/S de disco/NFS pendentes (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 nos níveis 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 items:
 `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

 Informação sobre a arquitetura do software.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

A informação é adquirida da função `uname()`.

Exemplo:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Informações do sistema operacional.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux, Windows.

Parâmetro:

- **info** - valores possíveis: *full* (padrão), *short* ou *name*

A informação é obtida 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 menor e maior, etc).
 Valor de retorno: *string 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 gerenciador 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: *string 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 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 não-Linux), *sectors* (setores trocados), ou *pages* (páginas trocadas, padrão no Linux).

Comentários:

- A fonte dessas informações 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.

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 os 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 de `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:

- Em 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.

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 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 seu tipo. Usado para descoberta de baixo nível.
 Valor de retorno: *string 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 no diretório.
 Valor de retorno: *Inteiro*.
 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 tanto em `types incl` quanto em `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 que o arquivo seja contado. Arquivos menores não serão contados. Sufixos de memória podem ser usados.
- **max size** - o tamanho máximo (em bytes) para que o arquivo seja contado. Arquivos maiores não serão contados. Sufixos de memória podem ser usados.
- **min age** - a idade mínima (em segundos) da entrada de 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 de diretório a ser contada. Entradas tão antigas ou mais 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:

- 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 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 (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>, recomenda-se 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: *string 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`) a NÃO listar. Se algum tipo de entrada estiver em ambos `types incl` e `types excl`, as 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:

- 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 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. 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`, recomenda-se 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. O item pode fun-

cionar 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 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);
- **mode** - valores possíveis: *apparent* (padrão) - obtém tamanhos aparentes de arquivos em vez do uso em disco (atua como `du -sb dir`), *disk* - obtém o uso em 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 percorrer: **-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:

- 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 exceder 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 de arquivo depende do suporte a arquivos grandes.

Exemplos:

```
vfs.dir.size[/tmp,log] #calcula o tamanho de todos os arquivos em /tmp contendo 'log' em seus nomes
vfs.dir.size[/tmp,log,^.+\.old$] #calcula o tamanho de todos os arquivos em /tmp contendo 'log' em seus no
```

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*.

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:

- 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** - caminho completo para o arquivo;
- **types incl** - lista de tipos de arquivos 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** - lista de tipos de arquivos a serem excluídos, veja types_incl para valores possíveis (por padrão, nenhum tipo é excluído)

Comentários:

- 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 são 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: *string JSON*.
 Veja plataformas suportadas.

Parâmetro:

- **file** - caminho completo para o arquivo

Comentários:

- Tipos de arquivos suportados em sistemas UNIX-like: arquivo regular, diretório, link simbólico, socket, dispositivo de bloco, 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: Cadeia de caracteres (hash MD5 do arquivo).
 Veja plataformas suportadas.

Parâmetro:

- **file** - o caminho completo para o arquivo

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* (apenas Unix);
- **resulttype** - *name* (padrão) ou *id*; para id - retorna uid/gid no Unix, SID no Windows.

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 semelhantes ao 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 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:

- 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:

- 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 vazias também são contadas).

Comentários:

- 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 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

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: *string JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD.

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: *string JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD.

Comentários:

- Sistemas de arquivos com contagem de inodes igual a zero, o que pode ser o caso de 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 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 relatados 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:
1) *total* - quantidade total de memória
2) tipos de memória específicos da plataforma: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*
3) estimativas em nível de usuário sobre quanta 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 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.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 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 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 do Zabbix (Zabbix agent ou Zabbix agent 2).
 Valor de retorno: 1 - Zabbix agent; 2 - Zabbix agent 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 remotamente.
 Valor de retorno: *string 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)

Comentários:

- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, veja Monitoramento remoto de estatísticas do Zabbix;
- Note 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 remotamente.
 Valor de retorno: *string 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)
- **queue** - constante (usar como está)
- **from** - atrasado por pelo menos (padrão é 6 segundos)
- **to** - atrasado por no máximo (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 de 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 *all* padrão 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 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 do libiconv (Projeto GNU) ou na documentação do Microsoft Windows SDK para "Code Page Identifiers".

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 os 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) é aplicada.

Solução de problemas de items de 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 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 que o valor de `Timeout` no arquivo de configuração do agent.

1 Zabbix agent 2

O agent Zabbix 2 suporta todas as chaves de item suportadas pelo 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 os detalhes completos.

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.	
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.	

Chave do item	Descrição	Plugin
mongodb.collection.stats	Retorna uma variedade de estatísticas de armazenamento para uma coleção específica.	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 de banco de dados atual para outros membros do cluster fragmentado ou conjunto de réplicas.	
mongodb.db.stats	Retorna as estatísticas que refletem o estado do sistema de banco de dados fornecido.	
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 específico ou tópicos (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 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
nvml.device.count	O número de dispositivos GPU.	NVIDIA GPU
nvml.device.decoder.utilization	Utilização do decodificador do dispositivo GPU.	
nvml.device.ecc.mode	Modo ECC do dispositivo GPU.	
nvml.device.encoder.stats.get	Estatísticas do codificador do dispositivo GPU.	
nvml.device.encoder.utilization	Utilização do codificador do dispositivo GPU.	
nvml.device.energy.consumption	Consumo de energia do dispositivo GPU.	
nvml.device.errors.memory	Informações de erro de memória ECC do dispositivo GPU.	
nvml.device.errors.register	Informações de erro de registro ECC do dispositivo GPU.	
nvml.device.fan.speed.avg	Velocidade média do ventilador do dispositivo GPU.	
nvml.device.get	GPUs descobertas com UUID e nome.	
nvml.device.graphics.frequency	Velocidade do clock gráfico do dispositivo GPU.	
nvml.device.memory.bar1.get	Informações de memória BAR1 do dispositivo GPU.	

Chave do item	Descrição	Plugin
nvml.device.memory.fb.get	Informações de memória framebuffer do dispositivo GPU.	
nvml.device.memory.frequency	Velocidade do clock de memória do dispositivo GPU.	
nvml.device.pci.utilization	Utilização do PCI do dispositivo GPU.	
nvml.device.performance.state	Estado de desempenho do dispositivo GPU.	
nvml.device.power.limit	Limite de energia do dispositivo GPU.	
nvml.device.power.usage	Uso de energia do dispositivo GPU.	
nvml.device.serial	Número de série do dispositivo GPU.	
nvml.device.sm.frequency	Velocidade do clock do multiprocessador de streaming do dispositivo GPU.	
nvml.device.temperature	Temperatura do dispositivo GPU.	
nvml.device.utilization	Estatísticas de utilização do dispositivo GPU.	
nvml.device.video.frequency	Velocidade do clock de vídeo do dispositivo GPU.	
nvml.system.driver.version	A versão do driver NVIDIA.	
nvml.version	A versão da biblioteca NVML.	
oracle.diskgroups.stats	Retorna as estatísticas dos grupos de disco Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Retorna a lista de grupos de disco ASM.	
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 cache do buffer 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.	

Chave do item	Descrição	Plugin
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 dispositivo 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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, password** - as credenciais de login do 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 de 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 replica set.
 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.db.stats[connString,<user>,<password>,<database>]

 Retorna as estatísticas que refletem o estado do sistema de um determinado banco de dados.
 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/` no 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. É possível especificar um nome de instância como parte da URI, por exemplo: `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. 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. Testa 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 de quorum.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.quorum.member.get

 Retorna os membros do quórum.
 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.

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" no 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" 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" 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 são 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 a serem passados para a 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 consultas DNS reversas (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, por favor, 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
```

nvml.device.count

 O número de dispositivos GPU.
 Valor de retorno: *Inteiro*.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.decoder.utilization[<deviceUUID>]

 Utilização do decodificador do dispositivo GPU como uma porcentagem.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.ecc.mode[<deviceUUID>]

 Informações do modo ECC do dispositivo GPU (atual, pendente).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.encoder.stats.get[<deviceUUID>]

 Estatísticas do codificador do dispositivo GPU.
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.encoder.utilization[<deviceUUID>]

 Utilização do codificador do dispositivo GPU como uma porcentagem.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.energy.consumption[<deviceUUID>]

 Consumo total de energia do dispositivo GPU em milijoules (mJ) desde que o driver foi recarregado pela última vez.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.errors.memory[<deviceUUID>]

 Informações de erro de memória ECC do dispositivo GPU (corrigido, não corrigido).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.errors.register[<deviceUUID>]

 Informações de erro de registro ECC do dispositivo GPU (corrigido, não corrigido).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.fan.speed.avg[<deviceUUID>]

 Velocidade média do ventilador do dispositivo GPU como uma porcentagem da velocidade máxima.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.get

 GPUs descobertas com UUID e nome.
 Valor de retorno: *JSON*.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.graphics.frequency[<deviceUUID>]

 Velocidade do clock gráfico do dispositivo GPU em MHz.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.memory.fb.get[<deviceUUID>]

 Estatísticas de memória framebuffer do dispositivo GPU (total, reservada, livre, usada).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.memory.bar1.get[<deviceUUID>]

 Estatísticas de memória BAR1 do dispositivo GPU (total, livre, usada).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.memory.frequency[<deviceUUID>]

 Velocidade do clock da memória do dispositivo GPU em MHz.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.pci.utilization[<deviceUUID>]

 Utilização do PCI do dispositivo GPU (taxa de transmissão/recepção em KBps).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.performance.state[<deviceUUID>]

 Estado de desempenho do dispositivo GPU.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.power.limit[<deviceUUID>]

 Limite de energia do dispositivo GPU em miliwatts.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.power.usage[<deviceUUID>]

 Uso atual de energia do dispositivo GPU em miliwatts.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.serial[<deviceUUID>]

 Número de série do dispositivo GPU.
 Valor de retorno: *String*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.sm.frequency[<deviceUUID>]

 Velocidade do clock do multiprocessador de streaming do dispositivo GPU em MHz.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.temperature[<deviceUUID>]

 Temperatura do dispositivo GPU em Celsius.
 Valor de retorno: *Inteiro*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.utilization[<deviceUUID>]

 Estatísticas de utilização do dispositivo GPU (utilização da GPU/memória como uma porcentagem).
 Valor de retorno: *JSON*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.device.video.frequency[<deviceUUID>]

 Velocidade do clock de vídeo do dispositivo GPU em MHz.
 Valor de retorno: *Integer*.

Parâmetro:

- **deviceUUID** - UUID do dispositivo GPU.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.system.driver.version

 A versão do driver NVIDIA.
 Valor de retorno: *String*.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

nvml.version

 A versão da biblioteca NVML.
 Valor de retorno: *String*.

Para mais informações, consulte o readme do plugin NVIDIA GPU.

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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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, não deve conter um espaço à direita).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **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>]

 As 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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` (a opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **destination** - 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **diskgroup** - o nome do grupo de disco ASM a ser consultado.

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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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, não deve conter um espaço à direita).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **destination** - 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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` (a opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita).
- **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, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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` (a opção de login não diferencia maiúsculas de minúsculas e não deve conter espaço à direita).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.redolog.info[connString,<user>,<password>,<service>]

 As informações do arquivo de log a partir 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;

    - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Retorna as estatísticas dos 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **tablespace** - nome do tablespace a ser consultado. Padrão (se deixado em branco e `type` estiver definido):
  - "TEMP" (se `type` for definido como "TEMPORARY");
  - "USERS" (se `type` for definido como "PERMANENT").
- **type** - o tipo de tablespace a ser consultado. 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 dos 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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário 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, não deve conter um espaço à direita).
- **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 - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS - 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).
- **password** - a senha do Oracle;
- **service** - o nome do serviço 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.

pgsql.archive[uri,<username>,<password>,<database name>]

 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.

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 O número combinado de checkpoints para o cluster do 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.

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.

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.

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 uma 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.

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 O número de tabelas inchadas por 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.

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 A lista de bancos de dados 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.

pgsql.db.size[uri,<username>,<password>,<database name>]

 O tamanho do banco de dados em bytes.
 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.

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.

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.

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.

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.

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.

pgsql.queries[uri,<username>,<password>,<database name>,<time period>]

 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;
- **timePeriod** - 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>]

 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.

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.

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.

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 o padrão não continha nenhum caractere curinga.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis;
- **pattern** - um padrão de estilo glob (* por padrão).

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: *Inteiro*.

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 escanear 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};
- Tipos de unidade HDD, SSD e NVME são suportados. Se uma unidade não pertencer a um RAID, o {#RAIDTYPE} ficará vazio. {#NAME} terá um complemento em 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 únicas 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[nome da unidade,<interface>]

 Retorna todas as propriedades de uma unidade systemd.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **nome da unidade** - 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}"] #coletar informações do estado ativo (active, reloading, inactive, faile
systemd.unit.info["{#UNIT.NAME}",LoadState] #coletar informações do estado de carregamento nas unidades sy
systemd.unit.info[mysqld.service,Id] #recuperar o nome técnico do serviço (mysqld.service)
systemd.unit.info[mysqld.service,Description] #recuperar a descrição do serviço (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #recuperar a última vez que o serviço entrou no est
systemd.unit.info[dbus.socket,NConnections,Socket] #coletar 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*.

Parâmetro:

- **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*: o hash SHA-1 do certificado.
  - *sha256_fingerprint*: o hash SHA-256 do certificado.
- O campo `$.result.value` indica o resultado da validação do certificado. Os 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 pelo novo agent Zabbix de segunda geração, o agent 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 item no 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.	

Chave do item	Descrição	Grupo de itens
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 é suportada apenas pelo agent Zabbix 2.	
net.tcp.service.perf	Verifica o desempenho de um serviço TCP. A verificação de LDAP e HTTPS no Windows é suportada apenas 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 em 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.	
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 em Windows 64 bits.	
proc.num	O número de processos. No Windows, apenas os parâmetros `name` e `user` são suportados.	Processos
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 obtido 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 em 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 obtido 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.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 no diretório. No Windows, symlinks de diretório são ignorados e hard links são contados apenas uma vez.	Sistemas de arquivos virtuais

Chave do item	Descrição	Grupo de itens
vfs.dir.get	A lista de entradas do diretório. No Windows, symlinks de diretório são ignorados e hard links são contados apenas uma vez.	
vfs.dir.size	O tamanho do diretório. No Windows, qualquer symlink 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 a ferramenta 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).	
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 similar, 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.	

Chave do item	Descrição	Grupo de item
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 processo(s) específico(s).	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
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 de 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 um cheque ativo;
- O agent não consegue enviar eventos do log ”Forwarded events”;
- 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 log.

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 que descreve o padrão necessário (diferencia maiúsculas de minúsculas);
- **severity** - uma expressão regular que descreve 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 que descreve o identificador de origem (não diferencia maiúsculas de minúsculas);
- **eventid** - uma expressão regular que descreve 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 itens recém-criados).

Comentários:

- O item deve ser configurado como um active check;
- 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 origem 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;
- Habilitar/desabilitar 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 em nomes de interface.

perf_counter[counter,<interval>]

 O valor de qualquer contador de desempenho do Windows.
 Valor de retorno: *Inteiro*, *float*, *string* ou *texto* (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:

- `interval` é usado para contadores que exigem mais de uma amostra (como utilização de CPU), então 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:

- `interval` é usado para contadores que exigem mais de uma amostra (como utilização de CPU), portanto, 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 dos objetos 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 falhas de página
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-bit, é necessário um agent Zabbix 64-bit 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 falhas de página para processos do Internet Exp
```

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[chave,<modo>,<regexp do nome>]

 A lista de valores ou chaves do Registro do Windows localizados na chave fornecida.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **chave** - 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);
- **modo** - valores possíveis:
values (padrão) ou *keys*;
- **regexp do nome** - descobre apenas valores com nomes que correspondam à expressão regular (padrão - descobre todos os valores). Permitido apenas com *values* como `modo`.

Chaves com espaços devem ser colocadas entre aspas duplas.

Exemplos:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #retorna os dados de todos o
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #retorna todas as subchaves de
```

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 snap-in Serviços do MMC;
- **param** - *state* (padrão), *displayname*, *path*, *user*, *startup* ou *description*.

Comentários:

- Items como `service.info[service,state]` e `service.info[service]` retornarão as mesmas informações;
- 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] - 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 uma 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 de memória atualmente comprometido 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 abrindo as propriedades do serviço. Na guia *Geral* você deve ver um campo chamado "Nome do serviço". O valor que aparece 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**.

Etapa 2

Configure um item para monitorar o serviço.

O item `service.info[service,<param>]` recupera informações sobre um determinado serviço. Dependendo das informações 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 no qual 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, 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 de 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 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 inicialmente configurado com suporte a SNMP, especificando o parâmetro `--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 do server e do proxy Zabbix registram linhas semelhantes à seguinte 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 sempre tentará novamente 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 o RFC 2571 (seção 3.1.1.1), ele deve ser exclusivo para cada dispositivo.

Warning:
O 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.

Note:
O Zabbix armazena em cache os mapeamentos SNMPv3 EngineID → IP e reutiliza os EngineIDs em cache para verificações subsequentes, em vez de enviar uma sonda a cada vez, reduzindo o tráfego de rede. Se um EngineID não puder ser reutilizado, uma nova tentativa com uma sonda será realizada para descobrir o novo EngineID.

Configurando o monitoramento SNMP

Para começar a monitorar um dispositivo via SNMP, as seguintes etapas devem ser executadas:

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 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 seu 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
```

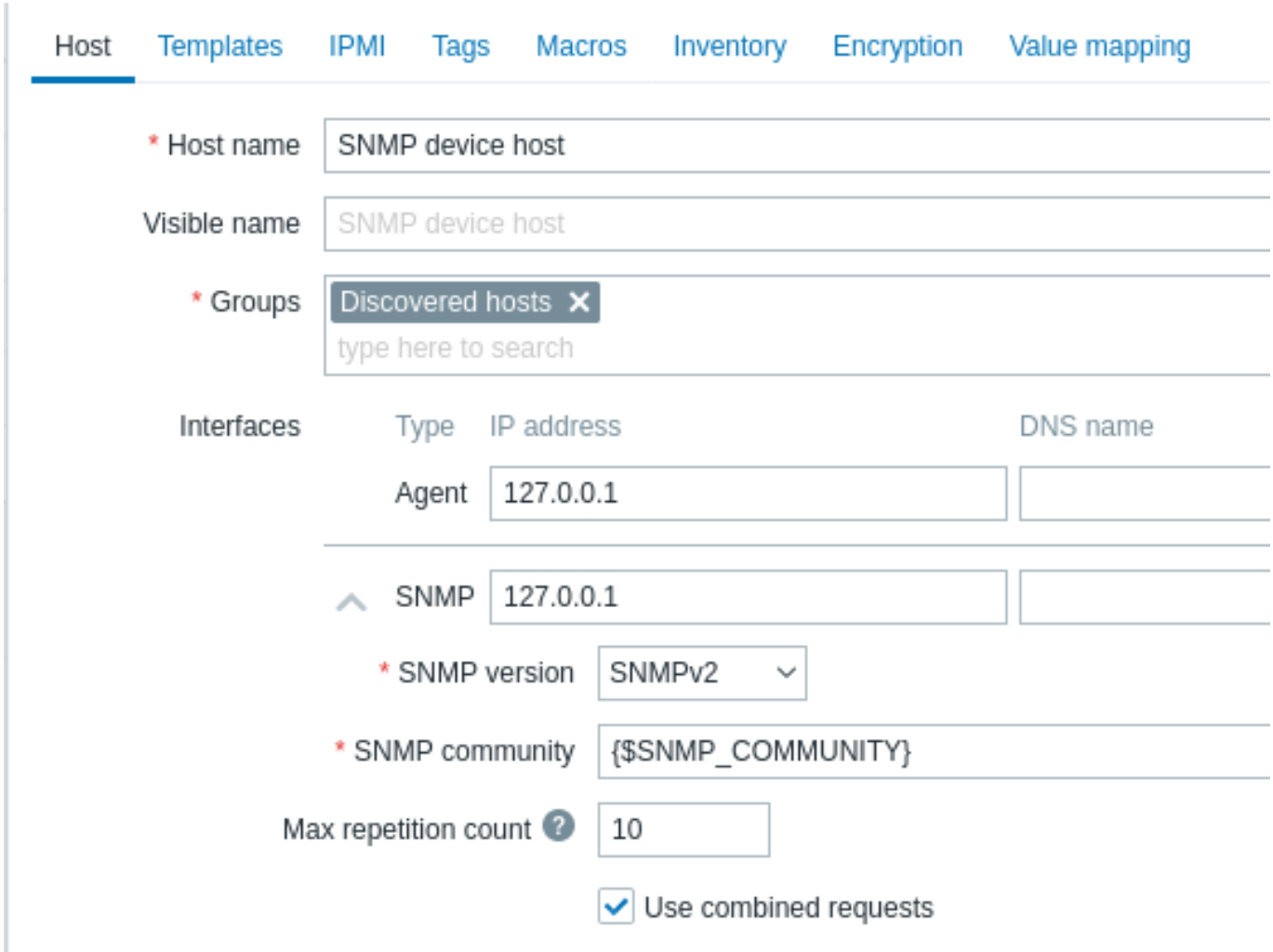
Novamente, o último número no OID é o número da porta.

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 exigem apenas a comunidade (geralmente 'public').
  - SNMPv3 requer opções mais específicas (veja abaixo).
- Especifique o valor de repetição máxima (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 o tempo limite da 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 recentes *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 opções a seguir:

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.

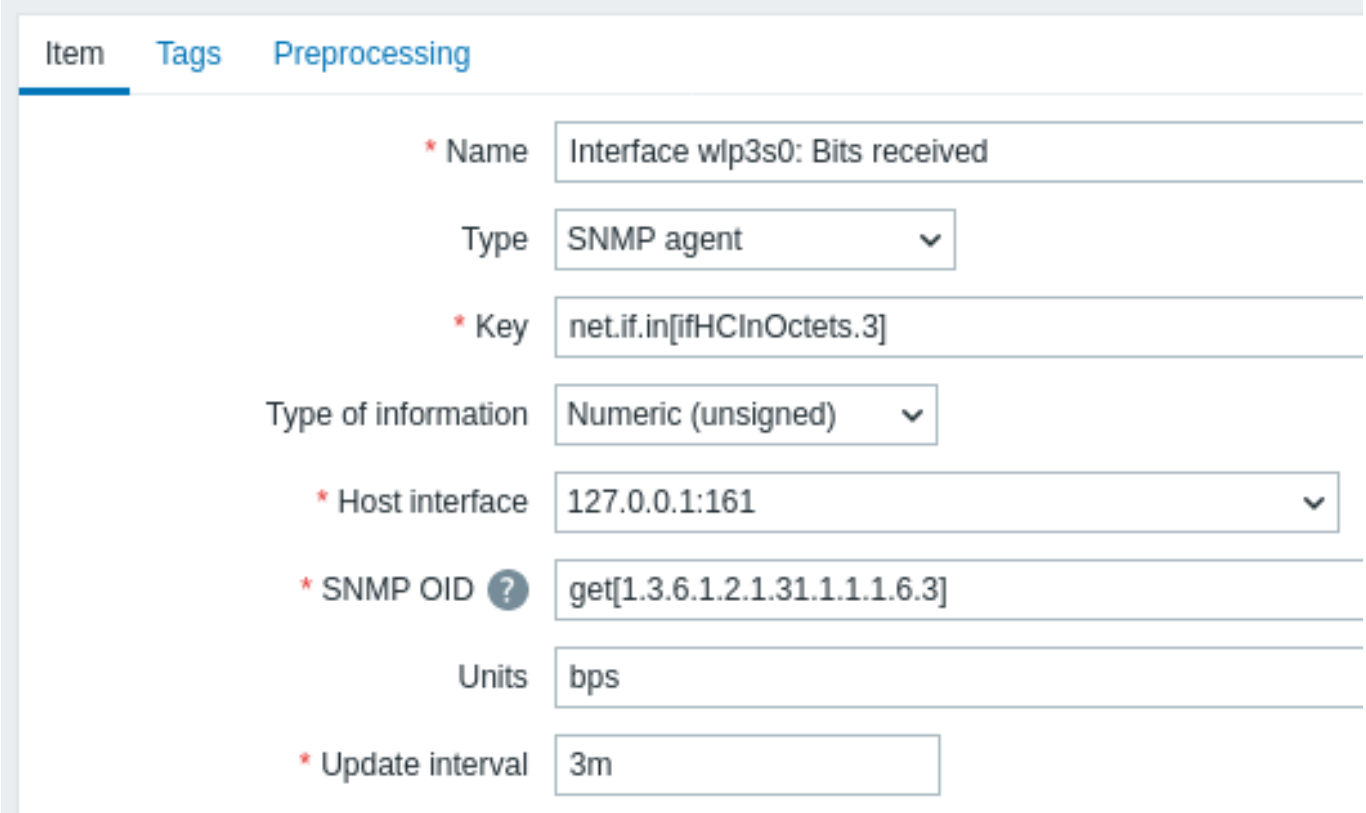
Se sua biblioteca net-snmp não suportar os protocolos AES192 e superiores, recompile o net-snmp com a opção `--enable-blumenthal-aes` e, em seguida, recompile o Zabbix server especificando a opção `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-`

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 manualmente 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 nativas SNMP bulk (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 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). Você pode usar este item como item mestre, com itens dependentes que extraem dados do mestre usando pré-processamento. É possível especificar múltiplos 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 feita uma tentativa de recuperar um único registro sem requisição bulk. Nomes de MIB 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 será 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 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). **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, um passo *Alteração por segundo* deve ser adicionado na aba *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)
Chave	<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 a representação numérica. A ferramenta snmpget pode ser usada para este propósito:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Exemplo 2

Monitoramento do tempo de atividade:

Parâmetro	Descrição
OID	MIB::sysUpTime.0
Key	router.uptime
Tipo de valor	Float
Unidades	uptime
Etapa de pré-processamento: Multiplicador personalizado	0.01

Requisições nativas SNMP bulk

O **walk[OID1,OID2,...]** item permite usar a funcionalidade nativa SNMP para requisições bulk (GetBulkRequest-PDUs), disponível nas versões SNMP 2/3.

Uma requisição GetBulk no SNMP executa múltiplas requisições GetNext e retorna o resultado em uma única resposta. Isso pode ser usado para itens SNMP regulares, bem como para descoberta SNMP, para 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 nativas SNMP bulk 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).

Uma nova tentativa ocorrerá para itens SNMP bulk para evitar falha se um dos pacotes for perdido. O timeout para itens SNMP com `get` e `walk` (definido no formulário de configuração do item) é definido para toda a sessão. Se o timeout for atingido, uma nova tentativa ocorrerá, o timeout será redefinido e a última requisição será reenviada, permitindo continuar a sessão a partir da última requisição se um único pacote for perdido ou chegar tarde demais. Considere definir um valor de timeout baixo para evitar longos atrasos se o dispositivo estiver inacessível, pois até 5 novas 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 ligações de 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 ”percorrimento”: ”obtenção” é usada para verificação de índice e ”percorrimento” para construir o cache;
- regras de descoberta de baixo nível SNMP se beneficiam das melhorias de ”percorrimento”.

No entanto, há uma questão técnica de 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 o 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, assumindo que o maior tamanho de resposta seja 32 variáveis, as requisições subsequentes terão os 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 requisições 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. Assim, 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, em vez do 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.

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 percorrimento de 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 com o 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 coleta de dados no item.

Parâmetro	Descrição
index	Método de processamento. Atualmente, um método é suportado: **index** – procura pelo índice e o adiciona ao OID dos 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 consulta. Sensível a maiúsculas e minúsculas.

Exemplo

Obtendo o uso de memória do processo *apache*.

Se estiver usando 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 caso outro item se refira ao mesmo OID base posteriormente - o Zabbix pesquisaria 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 de 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 a 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.
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.

OID especial	Identificador	Descrição
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. 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, será necessário 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 as MIBs, pode reabilitar
# o carregamento delas 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 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 entre 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, porque 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 que correspondem 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 com a expressão regular 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 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 server/proxy do Zabbix

Para ler os traps, o server ou proxy do Zabbix deve ser configurado para iniciar o processo SNMP trapper e apontar para o arquivo de trap que está sendo gravado 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 server do Zabbix a partir do snmptrapd usando o arquivo trapper. Para configurá-lo, adicione a opção `traphandle` ao arquivo de configuração do snmptrapd (`snmptrapd.conf`), veja 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 PERL RECEIVER]";
```

- 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 trap 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 declaração FORMAT deve começar com "ZBXTRAP [address]", onde [address] será comparado aos endereços IP e DNS das interfaces SNMP no Zabbix. Por 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 trap SNMP

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 poderá interpretar 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 de 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 volta para o passo 1.
3. O Zabbix lê os dados do arquivo atualmente aberto e define a nova localização.
4. Os novos dados são analisados. Se este era o arquivo rotacionado, o arquivo é fechado e volta para o passo 2.
5. Se não houver novos dados, o Zabbix dorme por 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 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/trunk/Dock
```

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 um 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 TESTE (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 em texto simples e, portanto, essas versões do protocolo só devem ser usadas em ambientes seguros, como redes privadas, e nunca devem ser usadas em redes públicas ou de terceiros.

A versão 1 do SNMP praticamente não é mais utilizada, pois não suporta contadores de 64 bits e é considerada um protocolo legado.

Para habilitar o recebimento 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 uma 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 o recebimento 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 desejar usar métodos de criptografia fortes, como AES192 ou AES256, use 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 disponibilidade de dispositivos IPMI (Intelligent Platform Management Interface) 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 das chamadas placas 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 para os 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 de 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 mais detalhes.

Configuração do server

Por padrão, o server Zabbix não está configurado para iniciar nenhum poller IPMI, portanto, quaisquer items IPMI adicionados não funcionarão. Para alterar isso, abra o arquivo de configuração do server 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 distinguidos 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 deseja 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 os 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 mais updates para a especificação IPMI).

O primeiro parâmetro para começar 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 de ”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 de 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 os 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 assim:

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

para alertar 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

Nomes de 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 `ipmitool` e OpenIPMI-2.0.19 exibem nomes de 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 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 usa OpenIPMI-2.0.19 (ou posterior), os itens com esses sensores discretos IPMI se tornarão ”NÃO SUPORTADO”. Você deve alterar os nomes desses *IPMI sensor* (remover o ’0’ no final) e aguardar por algum tempo até que eles se tornem ”Habilitados” novamente.

Notas 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.

Notas sobre o término 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 (fazendo 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.
icmppingretry	Acessibilidade do host por ICMP ping com tentativas.
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.

icmppingretry[<target>,<retries>,<backoff>,<size>,<timeout>,<options>]

 Acessibilidade do host por ICMP ping com tentativas. Se o primeiro pacote for bem-sucedido, pare; se o pacote falhar, tente novamente. Este item é útil para reduzir o número de pacotes enviados pela rede.
 Valor de retorno: *0* - ICMP ping falhou; *1* - ICMP ping bem-sucedido.

Parâmetros:

- **target** - o IP do host ou nome DNS;
- **retries** - o número de vezes que uma tentativa de ping em um destino será feita, não incluindo a primeira tentativa (0 ou mais; padrão 1);
- **backoff** - o número pelo qual o tempo de espera é multiplicado a cada solicitação sucessiva (intervalo de 1.0-5.0; padrão 1.0);
- **size** - o tamanho do pacote em bytes;
- **timeout** - o tempo limite em milissegundos;
- **options** - usado para permitir redirecionamento: se vazio (valor padrão), as respostas redirecionadas são tratadas como host de destino inativo; se definido como *allow_redirect*, as 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 ICMP ping (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* a indicação da porta é obrigatória;
- 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 TCP/UDP personalizadas 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.

Pings ICMP

O Zabbix usa o utilitário externo **fping** para processar pings ICMP (**icmpping**, **icmppingloss**, **icmppingretry**, **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 e seus derivados. 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 esse usuário deve ter permissões 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
retries	número	Número de tentativas de ping em um destino, não incluindo a primeira tentativa	-r	3	1	0	ilimitado
backoff factor	número	Número pelo qual o tempo de espera é multiplicado a cada requisição sucessiva	-B	1.5	1.0	1.0	5.0

Os valores padrão podem diferir ligeiramente dependendo da plataforma e versão.

Além disso, o Zabbix utiliza as opções do fping *-i intervalo ms* (não confunda com o parâmetro *interval* do item mencionado na tabela acima, que corresponde à opção *-p* do fping) 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 aceito 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 checados por qualquer uma das chaves *icmpping** em um arquivo temporário, que é então passado para o fping. Se os itens tiverem parâmetros de chave diferentes, apenas os que possuem parâmetros de chave idênticos

são gravados em um único arquivo. Todos os endereços IP gravados no mesmo arquivo serão checados pelo fping em paralelo, portanto, 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 Zabbix em execução no host
- O 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 preencher:

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. Veja a seção de chaves de item do Zabbix agent suportadas para 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 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. 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 data/hora do log para esta linha seria "pppppp:yyyyMMdd:hhmmss". Observe que os caracteres "p" e ":" são placeholders 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 utiliza 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 arquivo de 64 bits.
    * Em sistemas de arquivos ReFS (apenas a partir do Microsoft Windows Server 2012), IDs de arquivo de 128 bits.
    * Em sistemas de arquivos onde os índices de arquivo 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 acompanhando 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 dos arquivos de log não será respeitada, o que pode levar ao relatório de registros de arquivos de log correspondentes em ordem alterada (o problema não ocorre se os arquivos de log tiverem horários de 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 o 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 arquivos 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 arquivos de log analisados e 1000 registros correspondentes enviados ao server do Zabbix em uma verificação. Se o *Intervalo de atualização* for definido como 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 **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 realizado 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 o limite de 50% designado.
- Para registros de arquivos 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** se refere 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 perder alertas**. Use com cuidado, por sua conta e risco, apenas 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 gravar 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 ao 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 prossegue com a análise do 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 sequer 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, "pulos" 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" para manter 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 valores MD5 dos blocos iniciais (até os primeiros 512 bytes) serão diferentes. Dois arquivos com os mesmos valores 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.

Observações 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 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 do log 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 a reinicialização do agent Zabbix.

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. Mais tarde, o sistema operacional e o agent Zabbix são reinicializados 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

Ao iniciar, 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 suportados 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, sem tratamento especial para isso. A gravação 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: 1) o item de log correspondente não for mais monitorado, 2) 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.

Nome e localização dos 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 do item `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 ficam 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 `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 todas tivessem sido encontradas em uma única verificação.
- As verificações `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), então 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 Itens calculados

Visão geral

Um item calculado permite criar um cálculo baseado 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 valores de histórico quanto de tendências são armazenados e gráficos podem ser gerados.

Note:
Se o resultado do cálculo for um valor float, ele será arredondado para um 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 na fórmula de nenhuma maneira.

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.
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 referenciar 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.

Diferente das expressões de trigger, o Zabbix processa os itens calculados de acordo com o intervalo de atualização do item, 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 servidor Zabbix e, em seguida, calcular um agregado, dependendo da função agregada utilizada.

Os cálculos agregados não exigem que nenhum agent esteja 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 as 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"]))
```

Exemplos de sintaxe correta/incorreta

Expressões (incluindo chamadas de função) não podem ser usadas como parâmetros de funções de histórico, tendência ou foreach function. 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.

Verificações suportadas

As chaves de item são listadas sem parâmetros personalizáveis e informações adicionais. Clique na chave do item para ver todos os detalhes.

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 quantidade de valores enfileirados na fila do conector.
zabbix[discovery_queue]	A quantidade de verificações 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 atual de manutenção do host.
zabbix[host,active_agent,available]	A disponibilidade das verificações de agent ativo no host.
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 verificação 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 quantidade 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 quantidade 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>) gastou 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 proxies.
zabbix[proxy group,,pavailable]	A porcentagem de proxies online em um grupo de proxies.

Chave do item	Descrição
zabbix[proxy group„proxies]	A lista de proxies do Zabbix em um grupo de proxies.
zabbix[proxy group„state]	O estado de um grupo de proxies.
zabbix[proxy group,discovery]	Retorna uma lista de grupos de proxies 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 de disco/memória desde o início.
zabbix[proxy_buffer,state,current]	Retorna o estado de funcionamento 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„]	As métricas internas de um Zabbix server ou proxy remoto.
zabbix[stats„,queue„]	As métricas internas da fila de um Zabbix server ou proxy remoto.
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 items e parâmetros de item rotulados como ”não suportado em proxy” só podem ser coletados se o host for monitorado pelo server. Da mesma forma, valores ”não suportados no server” só podem ser coletados se o host for monitorado pelo proxy.

zabbix[boottime]

 O tempo de inicialização do processo do Zabbix server ou do Zabbix proxy em segundos.
 Valor de retorno: *Integer*.

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.

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: *Integer*.

zabbix[host„items_unsupported]

 O número de items habilitados e não suportados no host.
 Valor de retorno: *Inteiro*.

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 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* - não disponível; *1* - disponível; *2* - desconhecido.

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>]

 Informações sobre o Zabbix Java gateway.
 Valores de retorno: *1* - se <param> for *ping*; *versão do Java gateway* - se <param> for *version* (por exemplo: ”8.0.0”).

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: *Inteiro*.

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*.

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 determinado processo do Zabbix ou um grupo de processos (identificados por <type> e <mode>) passou no <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*, *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 requisição (padrão);
idle - o processo está em estado ocioso, sem fazer nada.

Comentários:

- Se <mode> for um número de processo do Zabbix que não está em execução (por exemplo, com 5 pollers em execução, o <mode> for especificado como 6), tal item ficará como 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 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>]

 Informações 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 estão sem envio; 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 do Zabbix com nome, modo, criptografia, compressão, versão, último visto, contagem de hosts, contagem de items, valores necessários por segundo (vps), status da versão (atual/desatualizada/não suportada), 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, compressão, versão, última visualização, contagem de hosts, contagem de items, valores por segundo (vps) necessários, 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 reportado 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 reportado 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: *Integer*; *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 server Zabbix.

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 das filas durante a manutenção para mais 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 aproximadamente com "Desempenho necessário do server, novos valores por segundo" em *Relatórios* → *Informações do sistema*.

zabbix[stats,<ip>,<port>]

 As métricas internas de um Zabbix server ou proxy remoto.
 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, veja Monitoramento remoto das estatísticas do Zabbix.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 As métricas internas da fila (veja `zabbix[queue,<from>,<to>]`) de um Zabbix server ou proxy remoto.
 Valores de retorno: *objeto JSON*.

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);
- **from** - atrasado por pelo menos (padrão é 6 segundos);
- **to** - atrasado por no máximo (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 de 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 da função de tendência do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **cache** - *buffer*;

- **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 - número de items em cache;
requests - 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: *Integer*.

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 efetividade do cache de valores do Zabbix.
 Valores de retorno: *Integer*. Com o parâmetro *mode* retorna: 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 neste estado por 24 horas, mesmo que o problema que acionou este modo seja resolvido antes.
- Você pode usar esta 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.

zabbixversion

 A versão do Zabbix server ou proxy.
 Valor de retorno: *String*. Por exemplo: `8.0.0`.

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 escrita 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 items não suportados (contador);
float - o número de valores float processados (contador);
uint - o número de valores unsigned integer processados (contador);
str - o número de valores character/string processados (contador);
log - o número de valores log processados (contador);
text - o número de valores text processados (contador);
not supported - o número de vezes que o processamento do item resultou em item se tornando não suportado ou mantendo esse estado (contador);
(com cache *history*, *index*, *trend*) *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:

- Especificar <cache> é obrigatório. O parâmetro de cache `trend` não é suportado com o Zabbix proxy.
- O cache de history é usado para armazenar valores de items. Um número baixo indica problemas de desempenho no lado do banco de dados.
- O cache de index de history é usado para indexar os valores armazenados no cache de history.
- Após o cache de history ser preenchido e então limpo, o cache de index de history 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 trend armazena o agregado para a hora atual para todos os items 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 recomendado em vez de libssh2.

Configuração

Autenticação por frase secreta

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 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 ficarão 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 a 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.
```

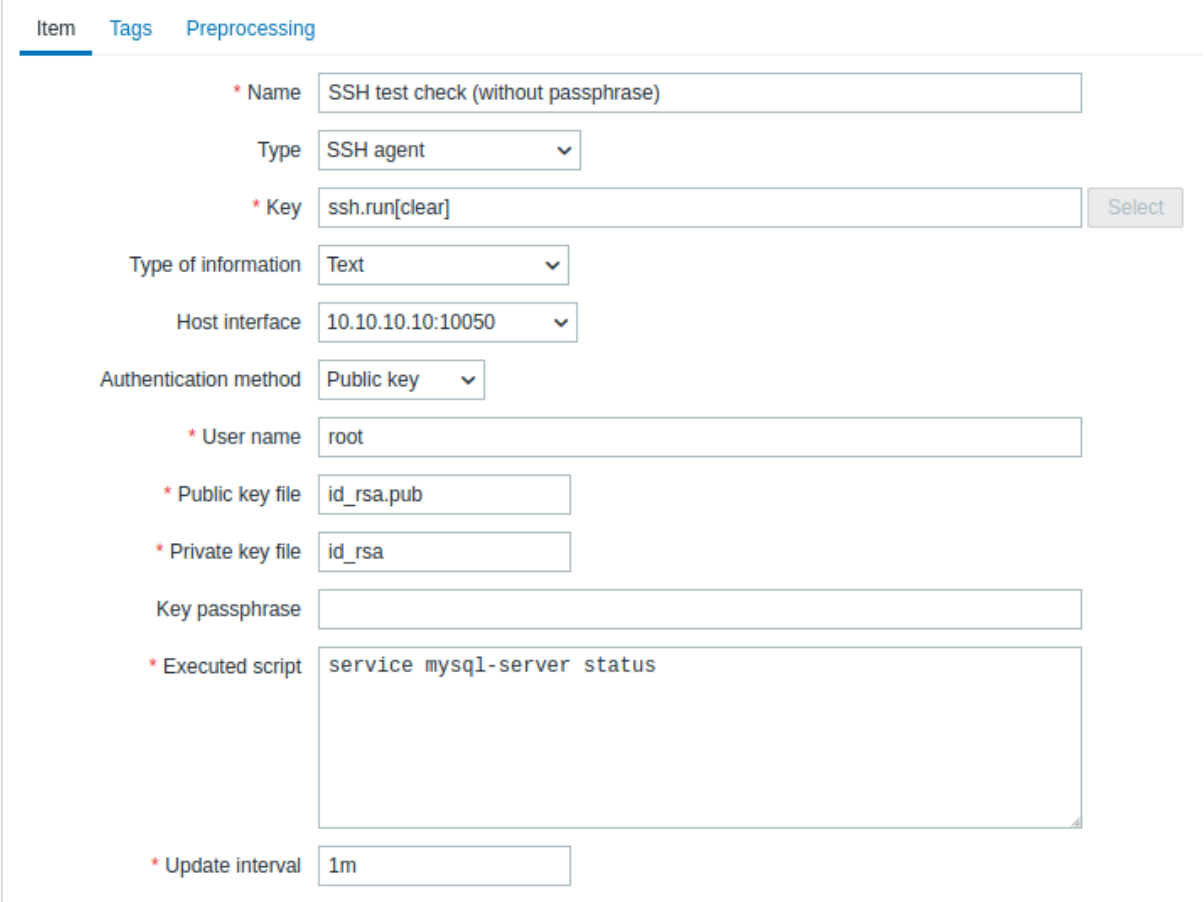
Agora é possível verificar o login SSH usando a chave privada padrão (*/home/zabbix/.ssh/id_rsa*) para a conta de usuário *zabbix*:

```
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 pode ser encerrada.

Configuração do item

O(s) comando(s) real(is) a ser(em) executado(s) deve(m) ser colocado(s) 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 requerem 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>,<subsystem>]**	**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`, `PubkeyAcceptedKeyTypes`. O suporte à chave e ao valor da opção depende da biblioteca SSH (por exemplo, `PubkeyAcceptedKeyTypes` é suportada apenas com libssh); se uma opção não for suportada, um erro será retornado e o item se tornará não suportado.
		
		Observe que o sinal "+" para adicionar 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-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		subsystem permite passar um subsistema SSH, limitando a conexão SSH a operações específicas permitidas pelo subsistema (por exemplo, transferências de arquivos usando SFTP ou gerenciamento de dispositivos de rede usando NETCONF). Observe que o uso de um subsistema também pode exigir o uso de uma sintaxe de script específica no parâmetro *Script executado*.
		
		Exemplos:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Método de autenticação	Um dos "Senha" ou "Chave pública".	
Nome de usuário	Nome de usuário (até 255 caracteres) para autenticação 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 pelo 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.

Parâmetro	Descrição	Comentários
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`
		
		Exemplo (para subsistema NETCONF):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

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) deve(m) ser colocado(s) no campo **Executed script** 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 *<encoding>* da chave deve ser especificado corretamente. Veja a página codificação de 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, pode 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 sob 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 server/proxy do Zabbix 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 (caracter, log ou tipo de informação 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 criado para o script será encerrado.

12 Itens trapper

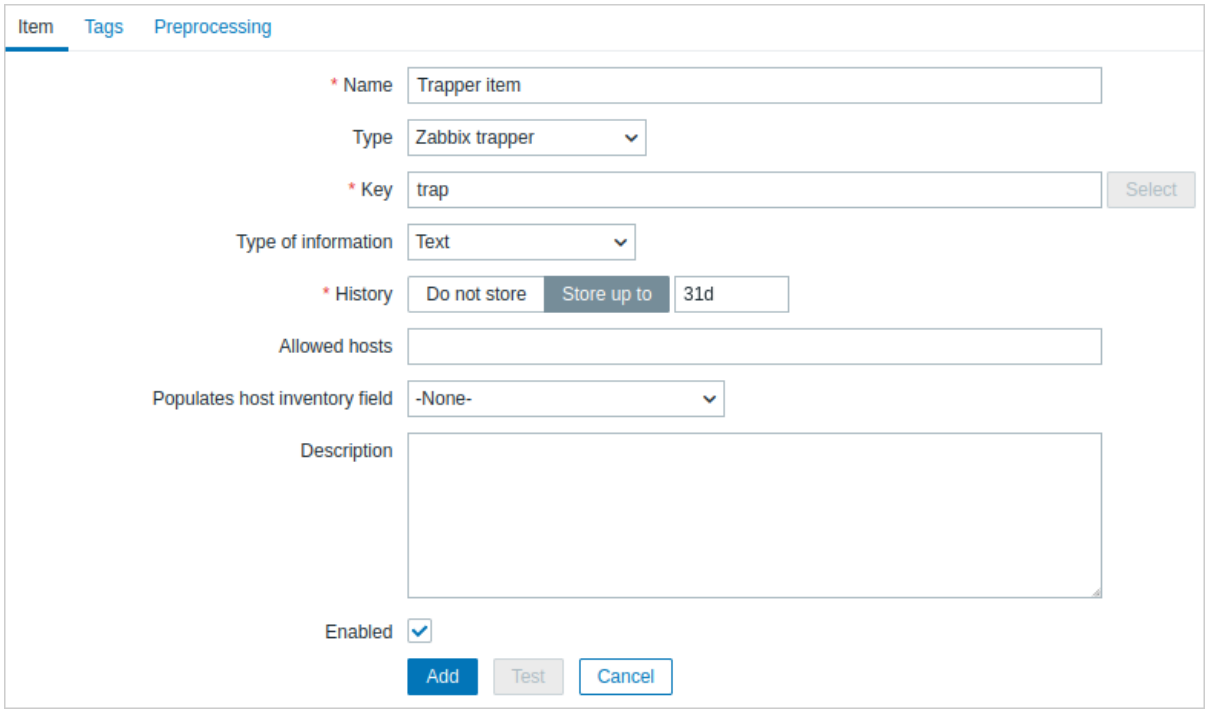
Visão geral

Itens 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 exigem 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 ao 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:

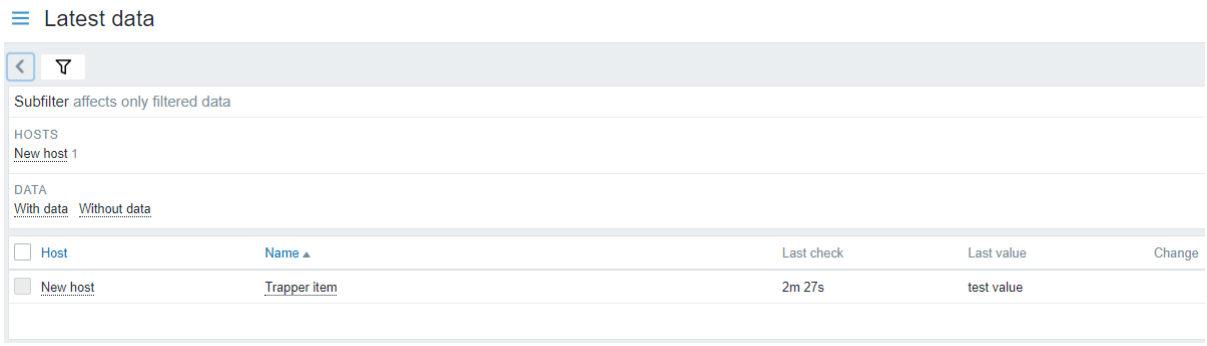
- o usuário que está enviando os dados não tem permissão de *leitura* para o host do item;
- o host está desabilitado ou em manutenção sem coleta de dados;
- o item não existe ou ainda não está incluído no cache de configuração do server;
- o item está desabilitado ou seu tipo é diferente de Zabbix trapper 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 via 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 informa que não é necessário 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, existem 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 Monitoramento e Gerenciamento usando 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 uma aplicação Java iniciada 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.

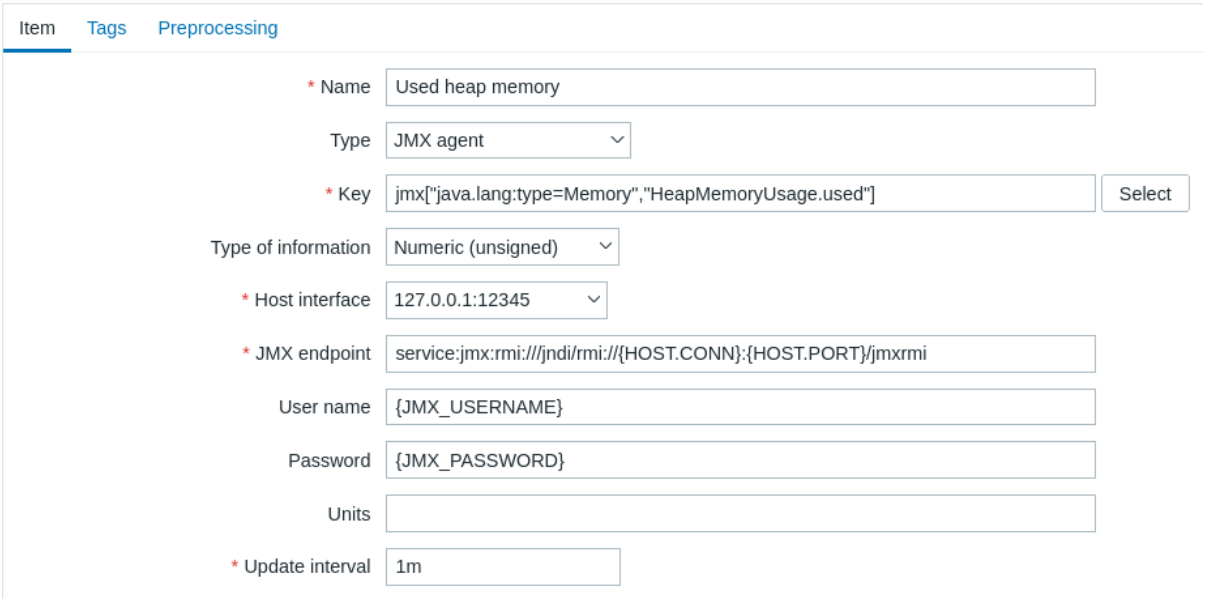


Todos os campos obrigatórios são marcados com um asterisco vermelho.

Adicionando um item de agent JMX

Para cada contador JMX do seu interesse, adicione um item **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 requerem 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 única** - uma descrição única que permite vários itens JMX com o mesmo nome de objeto e nome de atributo no host (opcional) Veja abaixo para 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 sua aplicação Java. Macros de usuário são suportadas.
Senha	Especifique a senha (até 255 caracteres), se você configurou autenticação em sua aplicação 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 ficará 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 um nome de atributo e uma chave de 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 do nome do atributo, então este valor de item retornará a estrutura completa do atributo em 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”, não apenas ”all”? Como distinguir um ponto que faz parte do nome do 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.

Na verdade, é só isso. Feliz monitoramento JMX!

Tipos de dados não primitivos

É possível trabalhar com MBeans personalizados retornando tipos de dados não primitivos, que sobrescrevem 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, pode fazê-lo de acordo com 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 desabilitados (ou configurados adequadamente)

Vamos fazer algumas configurações simples no zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

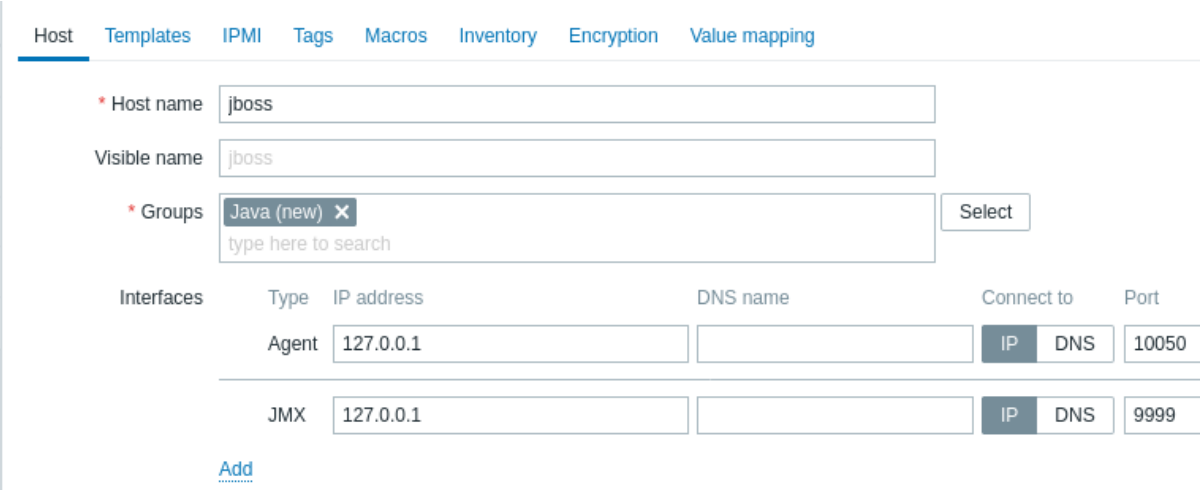
E no arquivo de configuração `zabbix_java/settings.sh` (ou `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Verifique se o JBoss está ouvindo em sua porta de gerenciamento padrão:

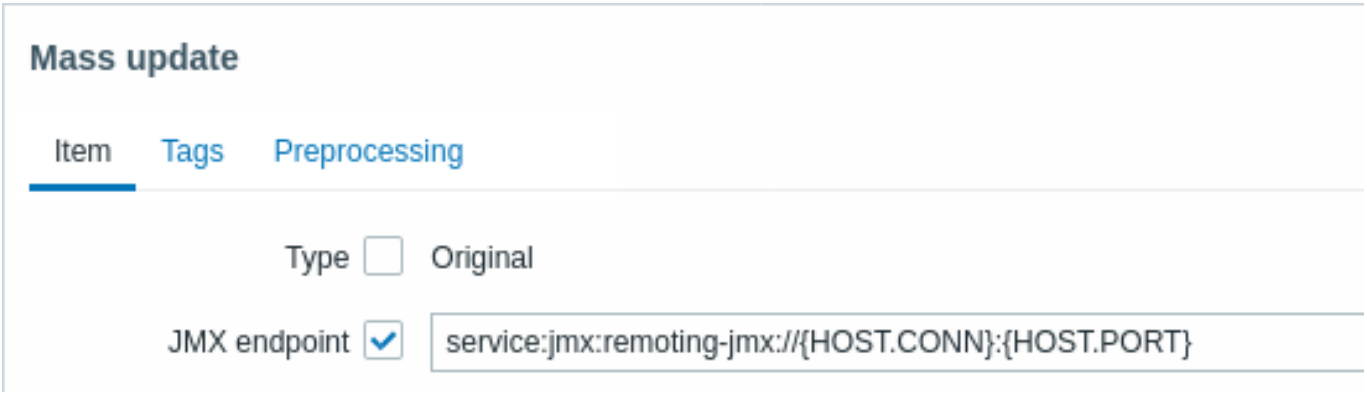
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Agora vamos criar um host com 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 de endpoint JMX para os items em nosso template JMX da seguinte forma:

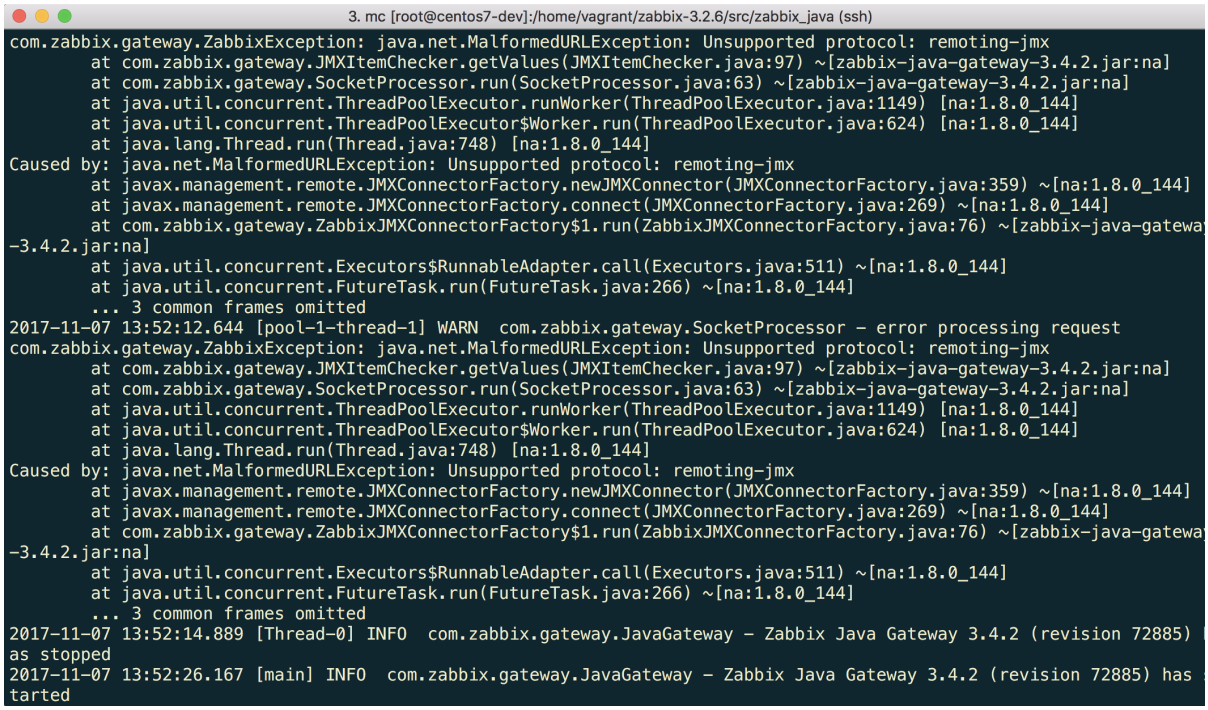
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Vamos atualizar o cache de configuração:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Observe que você pode encontrar um erro inicialmente.



”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 recomendada 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 do 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 o driver ODBC com o pacote unixODBC.`

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 PostgreSQL unixODBC, 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 em sistemas Ubuntu/Debian, use o gerenciador de pacotes 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 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 conexão a um servidor de banco de dados sem suporte para acesso a 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 a 1.6, se o gerenciador de drivers 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.

Parâmetro	Descrição
Driver	Nome do driver do banco de dados (conforme especificado em `odbcinst.ini`).
Server	IP/DNS do servidor de 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=8.0+
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 único 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 todas as colunas na cláusula `WHERE`.
ShowSystemTables	Especifica se o driver do banco de dados deve tratar tabelas de sistema como tabelas regulares em SQLTables; útil para acessibilidade, permitindo visibilidade nas tabelas de sistema.
Fetch	Especifica se o driver deve usar automaticamente declare cursor/fetch para lidar com instruções `SELECT` e manter um cache de 100 linhas.
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

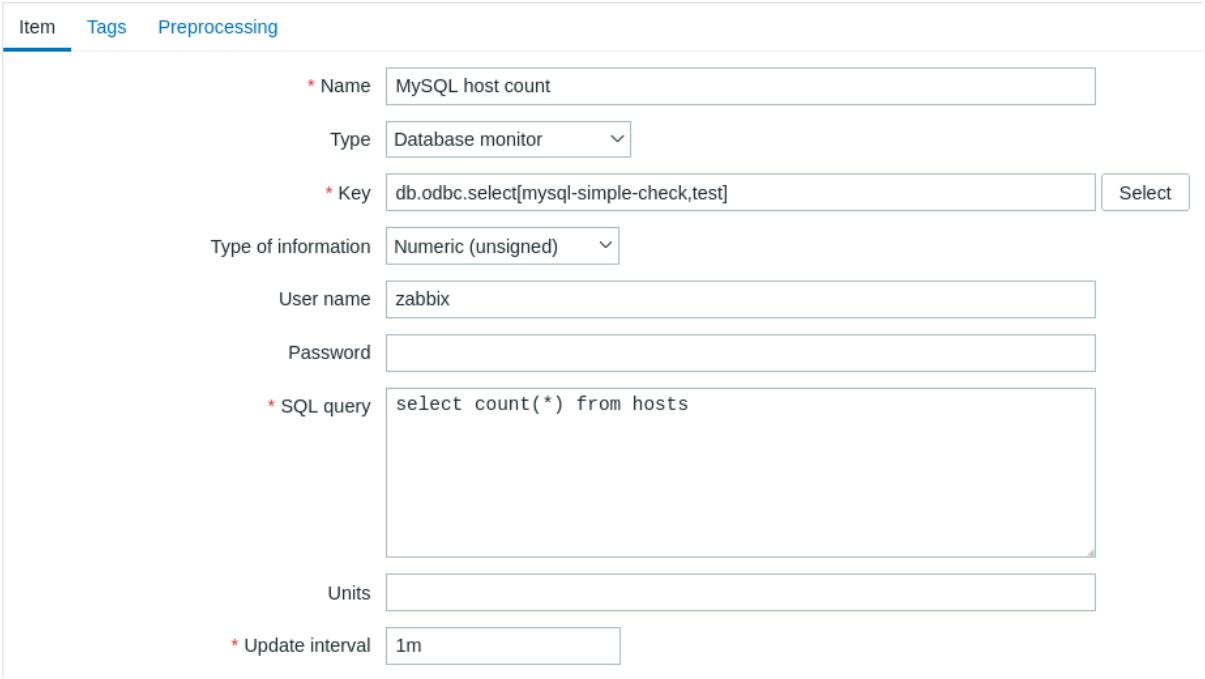
Para testar se a conexão ODBC está funcionando corretamente, você pode usar o utilitário `isql` (incluído no pacote `unixODBC`):

```
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, ela será adicionada à string de conexão como `PWD=<password>`. Caracteres especiais são suportados neste campo. A senha será adicionada à string de conexão após o nome de usuário como, por exemplo, `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 retornado pela consulta. Se o tipo de informação for selecionado incorretamente, o item se tornará não suportado.

Notas importantes

- Os items de monitoramento de banco de dados ficarão sem suporte 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 SGBD que irá processá-la. A sintaxe da requisição para uma procedure de armazenamento 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 única>,<dsn>,<string de conexão>]

 Transforma o resultado da consulta SQL em um array JSON.
 Valor de retorno: *objeto JSON*.

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.
- 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[Exemplo MySQL,,"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 mesmo 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, 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.

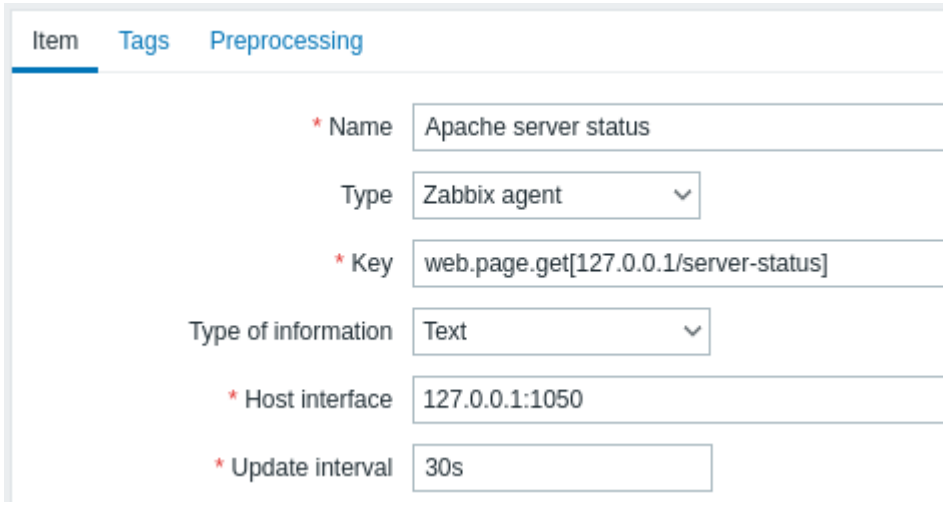
Limitações

- 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
- 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. Por isso, o **item mestre** deve ser configurado (ou existir) primeiro:

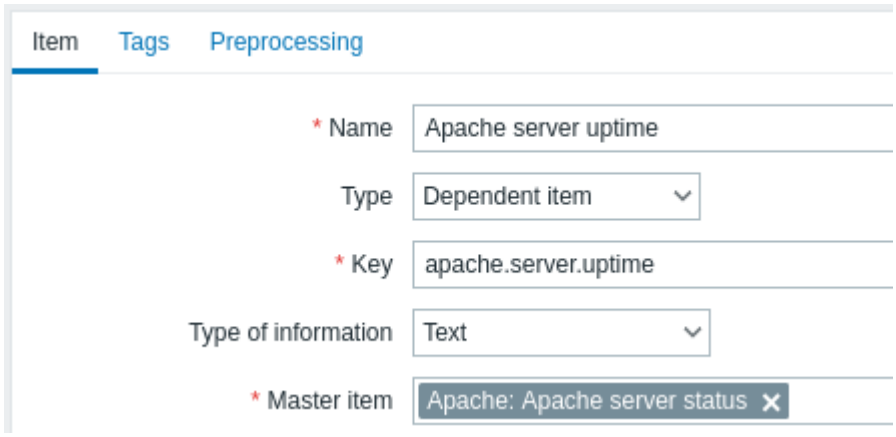
- Vá para: *Coleta de dados* → *Hosts*
- Clique em *Itens* na linha do host
- Clique em *Criar item*
- Insira os parâmetros do item no formulário



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**.

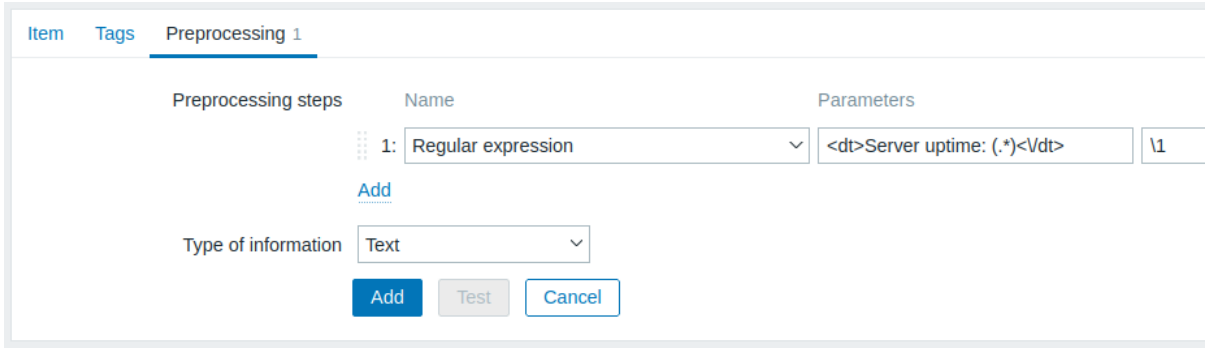


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

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.

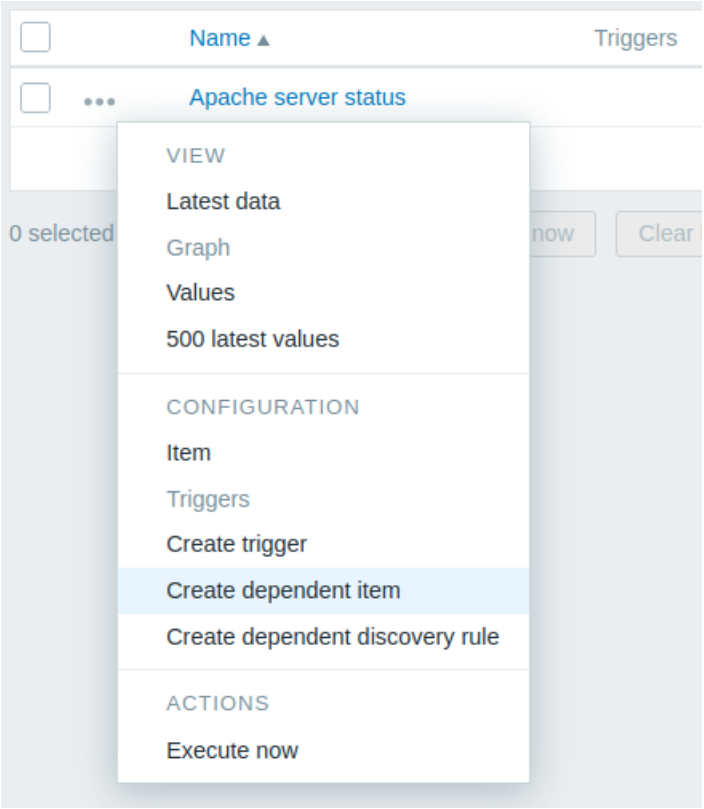
Você pode usar o pré-processamento do valor do item para extrair a parte necessária do valor do item mestre.



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.

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 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, opcionalmente, redirecionamentos (veja a opção *Seguir redirecionamentos* abaixo). O número máximo de redirecionamentos é fixado em 10 (usando a opção cURL CURLOPT_MAXREDIRS).

> **Attention:**
> O Zabbix server/proxy deve ser inicialmente configurado com suporte ao 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

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}, {HOST.PORT}, {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}, {HOST.PORT}, {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 da 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 da 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" *Observe* 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}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível.
Cabeçalhos	Cabeçalhos HTTP personalizados que serão enviados ao realizar uma requisição. Especificados como pares de atributo e valor. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {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 é salvo como um objeto, caso contrário, é 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. *Observe* que apenas autenticação simples é suportada com proxy HTTP. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {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 certificado SSL	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 SSLCALocation do Zabbix server ou proxy. 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 SSLCertLocation do Zabbix server ou proxy. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. 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 SSLKeyLocation do Zabbix server ou proxy. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {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.

Parâmetro	Descrição
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*, consulte atributos gerais do item.
Habilitar trapping	Com esta caixa 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 Zabbix sender, ou aceitará dados enviados ao Zabbix server usando o método de API `history.push`. Para mais informações sobre o envio de dados, consulte: Itens trapper.
Hosts permitidos	Visível apenas se a caixa *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 forma 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 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 (geralmente 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:

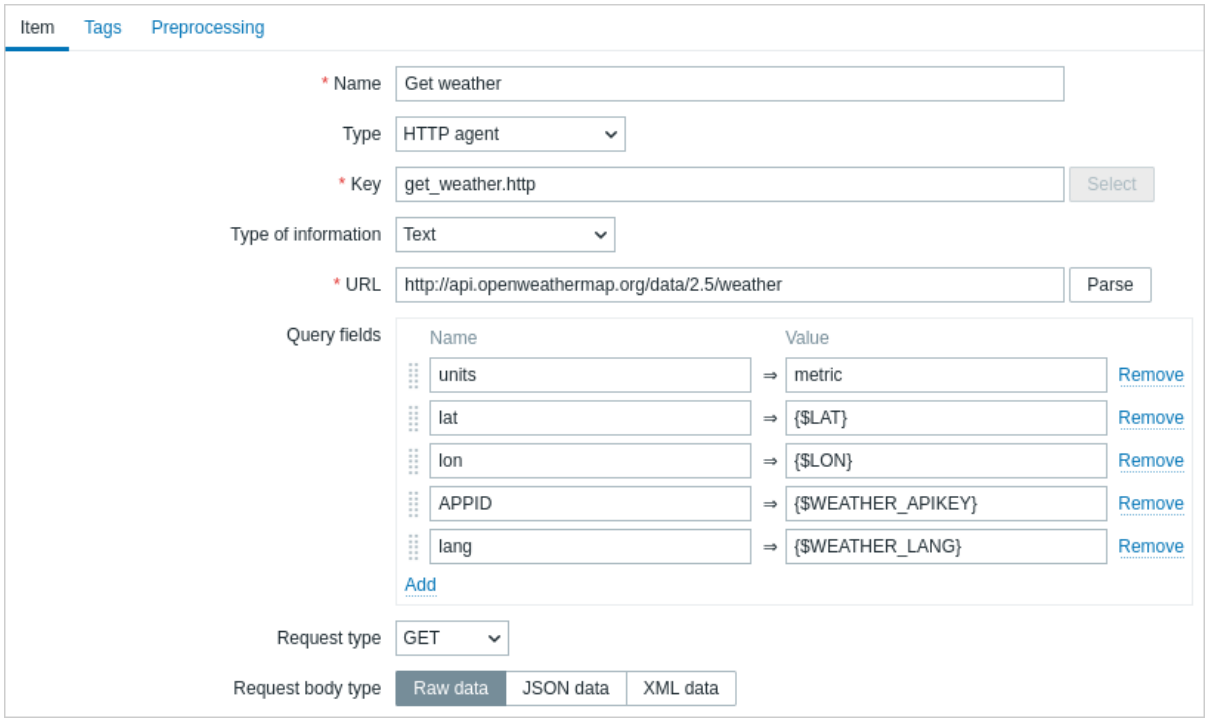
Observe o uso do método POST com dados JSON, definindo os cabeçalhos da requisição e solicitando o retorno apenas dos cabeçalhos:

- 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.

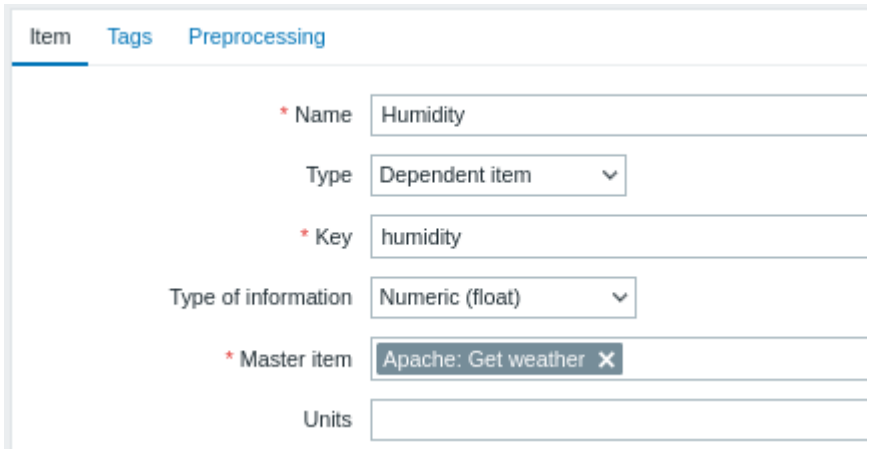
Exemplo de JSON retornado em resposta ao HTTP agent:


```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
```

```
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
    }
}
```

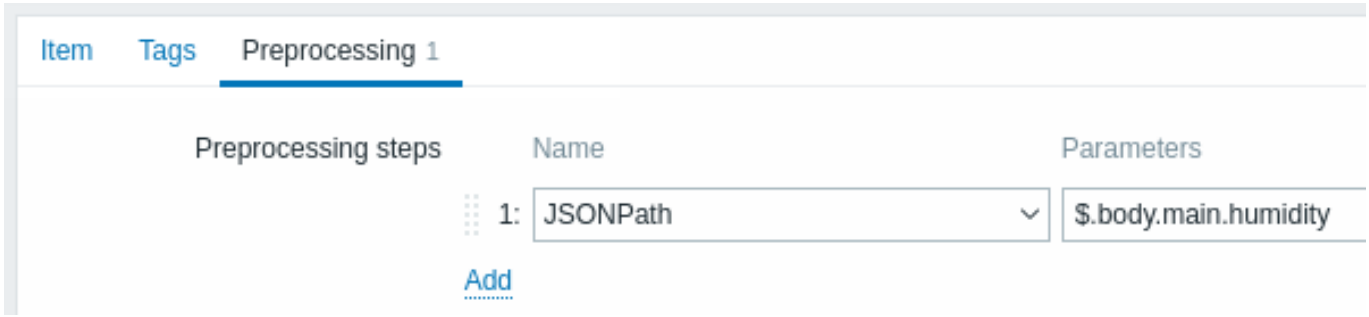

A próxima tarefa é configurar itens dependentes que extraem dados do JSON.

- Configure um item dependente de exemplo para umidade:

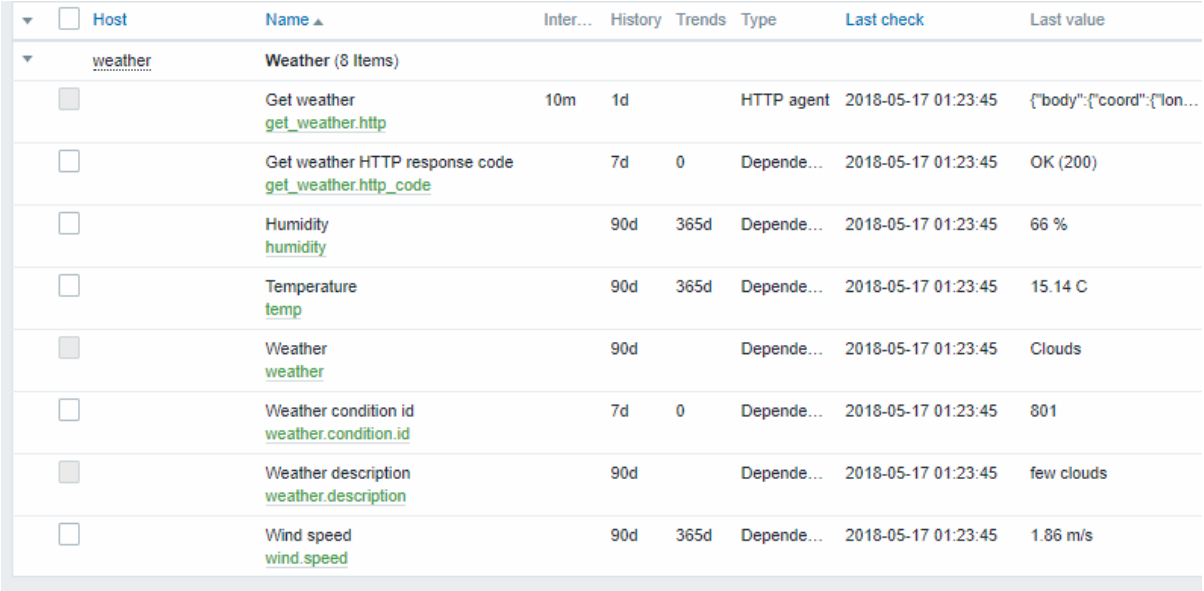


Outras métricas meteorológicas, como 'Temperatura', são adicionadas da mesma forma.

- 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 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 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 itens 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 itens 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 itens 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> - seleciona por nome da métrica {__name__=”<nome da métrica>”} - seleciona por nome da métrica {__name__=~”<regex>”} - seleciona por nome da métrica usando expressão regular {<nome do rótulo>=”<valor do rótulo>”,...} - seleciona por nome do rótulo {<nome do rótulo>=~”<regex>”,...} - seleciona por nome do rótulo usando expressão regular {__name__=~”.*”}==<valor> - seleciona por valor da métrica Ou uma combinação dos acima: <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 a barra invertida, aspas duplas e quebras de linha devem ser escapadas 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*
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 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 saúde (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 Prometheus do Zabbix.

PromQL instant vector selector		Pré-processamento Prometheus do Zabbix
Diferenças		
Destino da consulta	Servidor Prometheus	Texto simples no formato de exposição Prometheus
Retorna	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	<metric name> ou {__name__="<metric name>"}	<metric name> ou {__name__="<metric name>"}
Selecionando por nome de métrica que corresponde à expressão regular	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Selecionando por valor de <label name> igual à string	{<label name>="<label value>",...}	{<label name>="<label value>",...}

PromQL instant vector selector		Pré-processamento Prometheus do Zabbix
Selecionando por valor de <label name> que corresponde à expressão regular	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
Selecionando por valor igual à string	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

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 timeout 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, depois 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.

script_item.png

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 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, pesquise por "Item do tipo Script" 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

Coletar 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 do 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);
```

Múltiplas solicitações HTTP

Coletar o conteúdo de *https://www.example.com* e *https://www.example.com/release_notes*:

- Crie um item do 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");
```

Logando

Adicione a entrada "Log test" ao log do Zabbix server e receba o valor do item "1" em 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 items de navegador permitem monitorar sites e aplicações web complexas usando um navegador.

> **Attention:**
> O suporte a items de navegador é atualmente experimental.

Os items de 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 aplicações web.

Além do script, uma lista opcional de parâmetros (pares de nome e valor) e timeout pode ser especificada.

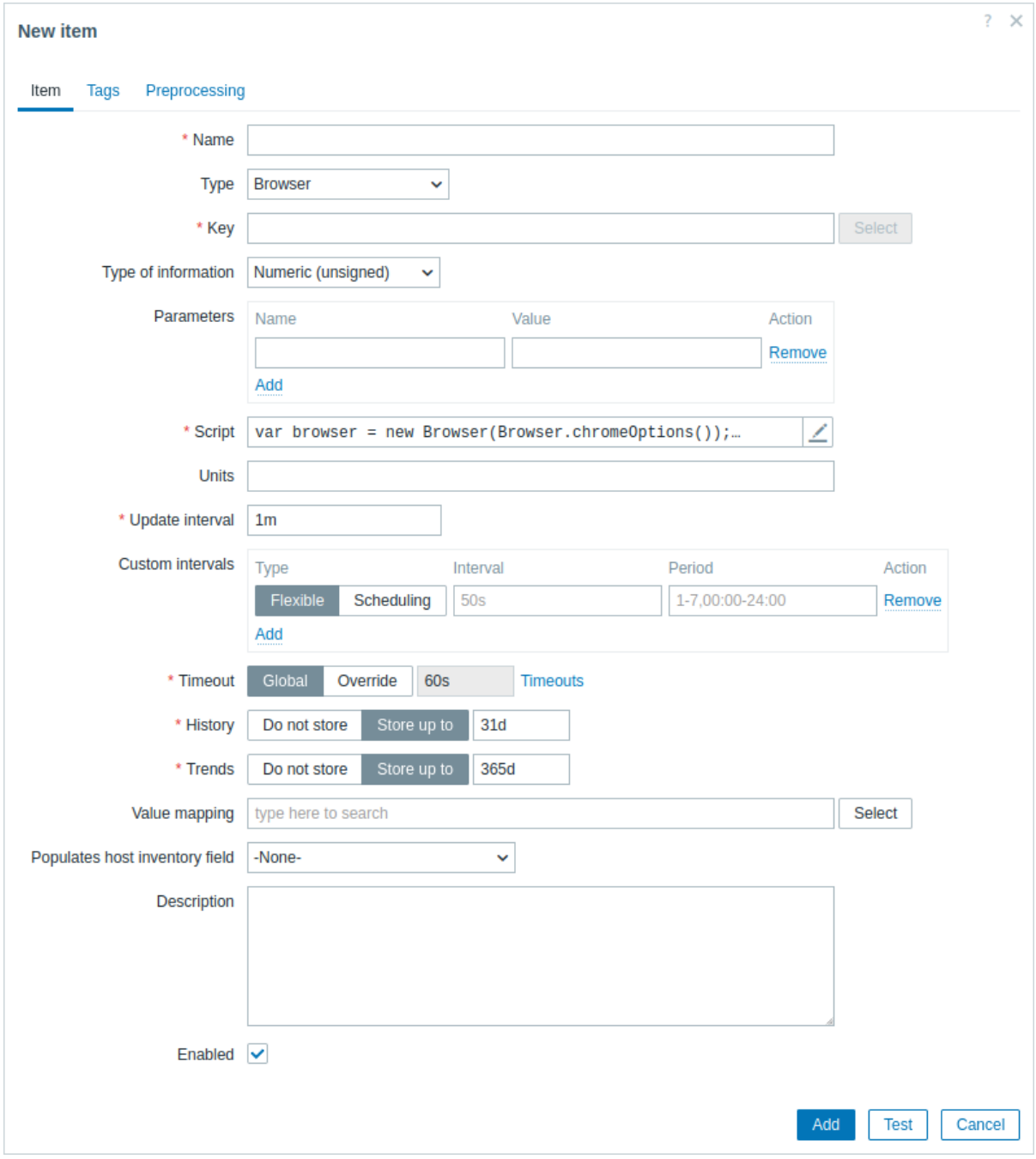
> **Attention:**
> O item implementa parcialmente o padrão W3C WebDriver com Selenium Server ou um WebDriver puro (por exemplo, ChromeDriver) como endpoint de teste web. Para que o item funcione, defina o endpoint no parâmetro `WebDriverURL` do arquivo de configuração do Zabbix server/proxy (se estiver usando o ChromeDriver, veja Considerações de Segurança). Para melhor desempenho, considere usar um servidor dedicado para o ambiente de teste web.

As verificações de items de navegador são executadas e processadas pelos browser pollers do Zabbix server ou proxy. Se necessário, você pode ajustar o número de instâncias pré-inicializadas de browser pollers no parâmetro `StartBrowserPollers` do arquivo de configuração do Zabbix server/proxy.

Para monitorar sites e aplicações web complexas, 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, a todos os objetos JavaScript adicionais e objetos JavaScript do item 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 container, você pode usar um cliente Virtual Network Computing (VNC) para se conectar ao servidor VNC da máquina. Isso permite visualizar e interagir 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 utiliza 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 até uma página (especificada como 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 utiliza 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);
}
```

4 Histórico 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 e forte 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 itens dependentes e inventário. Nenhuma função de trigger será avaliada porque a avaliação de trigger é baseada apenas em dados de histórico.

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 formulário
- 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 Zabbix server acumula dados de tendências em tempo de execução no cache de tendências, à medida que os dados chegam. O server grava as tendências da **hora anterior** de cada item no banco de dados (onde o frontend pode encontrá-las) nas seguintes situações:

- o server 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 server 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 server grava o cache de tendências e já existem tendências no banco de dados para essa hora (por exemplo, o server foi reiniciado no meio da hora), o server precisa usar comandos de atualização em vez de simples inserções. Portanto, em uma instalação maior, se for necessário reiniciar, é desejável parar o server 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 forma alguma.

Attention:
Se as tendências forem definidas como '0', o Zabbix server 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 server pode resultar em perda de precisão nos cálculos de valor médio do tipo de dado sem sinal para a hora atual.

5 Parâmetros de 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 entram em cena.

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. Insira 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 da 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 agent sempre retornará '1' para um item com a chave 'ping'.

Um exemplo mais complexo:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

O agent 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 dois sinais de cifrão - 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, tal processamento de referência é ignorado e, portanto, qualquer citação de sinal $ não é necessária.

Attention:
Certos símbolos não são permitidos em parâmetros de usuário por padrão. Veja a documentação UnsafeUserParameters para uma lista completa.

Exemplo 1

Algo muito simples:

```
UserParameter=ping[*],echo $1
```

Podemos definir um número ilimitado de items para monitoramento, todos tendo o 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 alive
```

Este parâmetro pode ser usado para monitorar a disponibilidade do banco de dados MySQL. Podemos passar o nome de usuário e a 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 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.

Parâmetros de usuário que retornam texto (caractere, log ou tipo de informação texto) também podem retornar um espaço em branco. Em caso de resultado inválido, o item ficará não suportado.

1 Estendendo os agents do Zabbix

Este tutorial fornece instruções passo a passo sobre como estender a funcionalidade do agent do Zabbix com o uso de um parâmetro de usuário.

Passo 1

Escreva um script ou linha de comando para recuperar o parâmetro necessário.

Por exemplo, podemos escrever o seguinte comando para obter o número total de consultas executadas por um servidor MySQL:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -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 | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions é um identificador único. Pode ser qualquer identificador de chave válido, por exemplo, *queries*.

Teste este parâmetro usando o agent do Zabbix com a opção "-t" (se estiver executando como root, no entanto, observe que o agent pode ter permissões diferentes quando iniciado como daemon):

```
zabbix_agentd -t mysql.questions
```

Passo 3

Recarregue os parâmetros do usuário a partir do arquivo de configuração executando:

```
zabbix_agentd -R userparameter_reload
```

Você também pode reiniciar o agent em vez do comando de controle em tempo de execução.

Teste o parâmetro usando o utilitário zabbix_get.

Passo 4

Adicione um novo item com a Chave=mysql.questions ao host monitorado. O tipo do item deve ser Zabbix Agent ou Zabbix Agent (active).

Esteja ciente de que o tipo de valores retornados deve ser definido corretamente no Zabbix server. 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 a chave *perf_counter[]*.

Por exemplo:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

ou

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Para mais informações sobre o uso desta chave ou seu equivalente apenas em inglês `perf_counter_en`, consulte Chaves de item específicas do Windows.

Para obter uma lista completa dos contadores de desempenho disponíveis para monitoramento, você pode executar:

```
typeperf -qx
```

Você também pode usar a descoberta de baixo nível para descobrir várias instâncias de objeto de contadores de desempenho do Windows e automatizar a criação de itens perf_counter para objetos de múltiplas instâncias.

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 do local do sistema.

Para descobrir os equivalentes numéricos, execute o **regedit** e localize o *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 de contador de desempenho

Você pode implantar alguns parâmetros PerfCounter para o monitoramento de contadores de desempenho do Windows.

Por exemplo, você pode adicioná-los ao arquivo de configuração do agent Zabbix:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
ou
PerfCounter=UserPerfCounter2,"\4\24",30
```

Com esses parâmetros em vigor, você pode simplesmente usar *UserPerfCounter1* ou *UserPerfCounter2* como as chaves para criar os respectivos items.

Lembre-se de reiniciar o agent Zabbix após fazer alterações no arquivo de configuração.

7 Atualização em massa

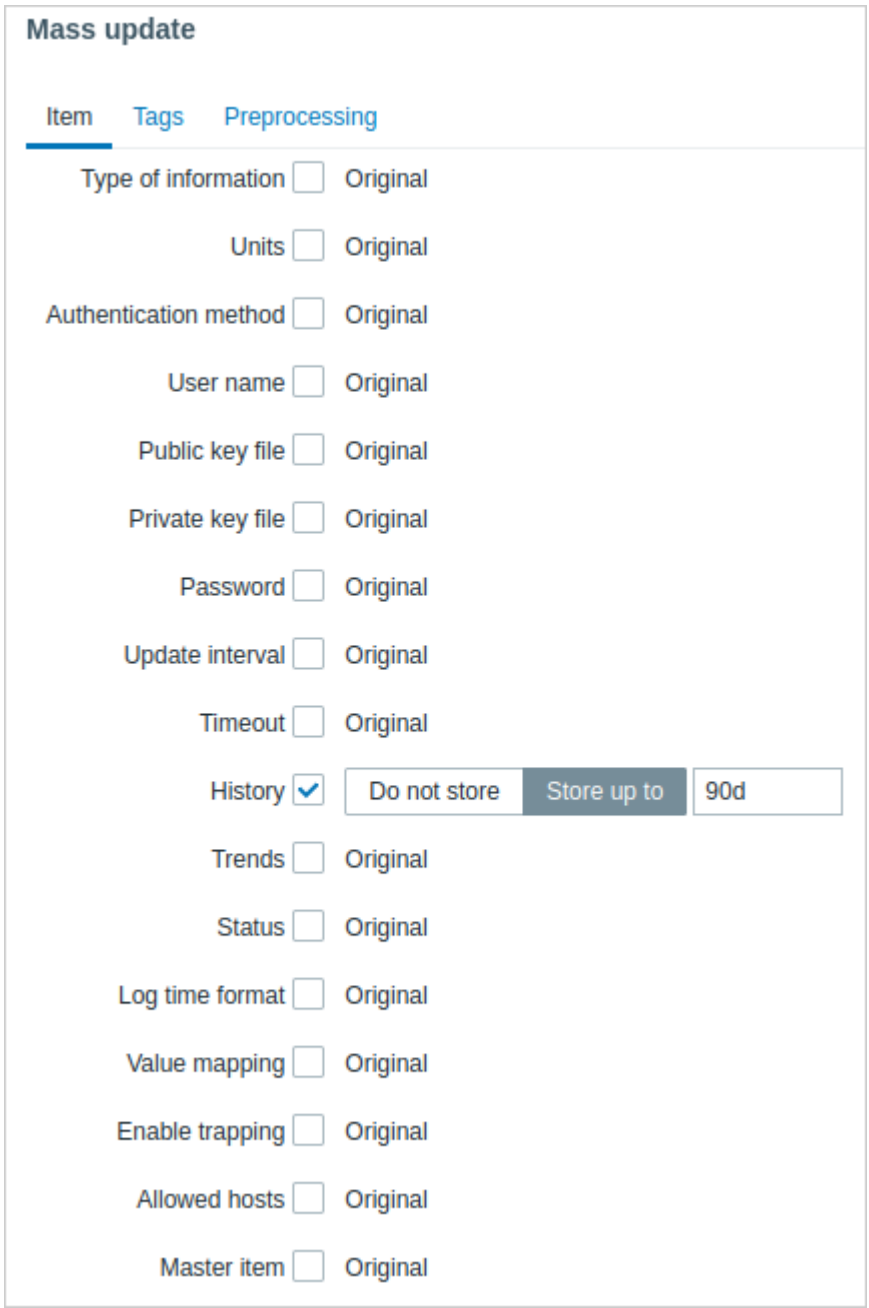
Visão geral

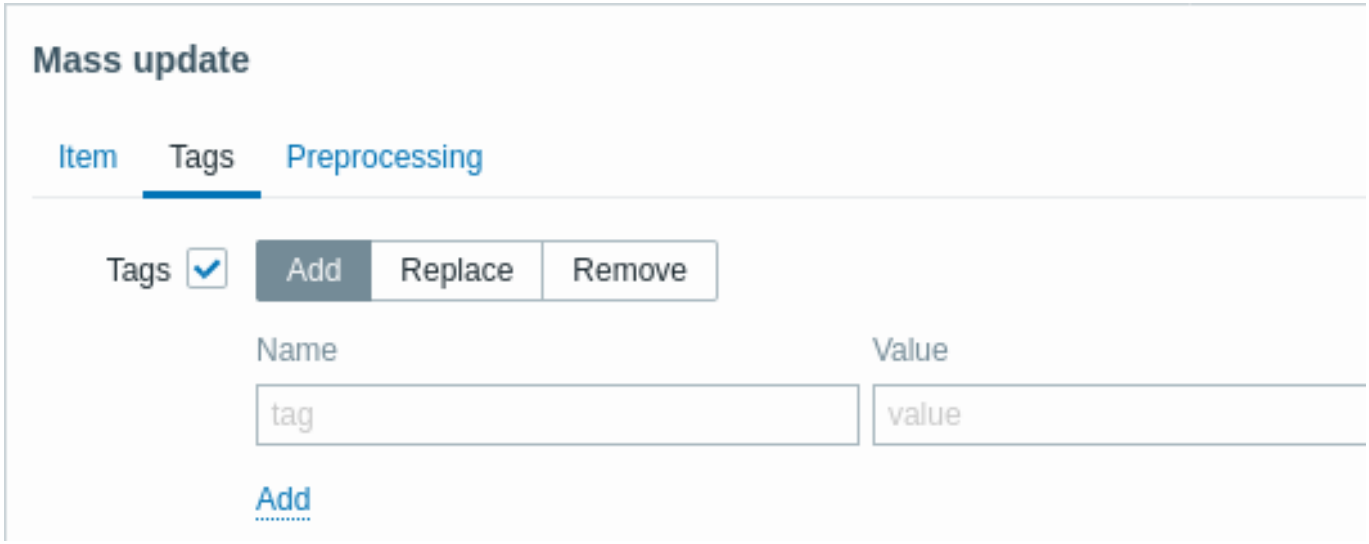
Às vezes, você pode querer alterar algum atributo para vários items de uma vez. Em vez de abrir cada item individualmente para edição, você pode usar a função de atualização em massa para isso.

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

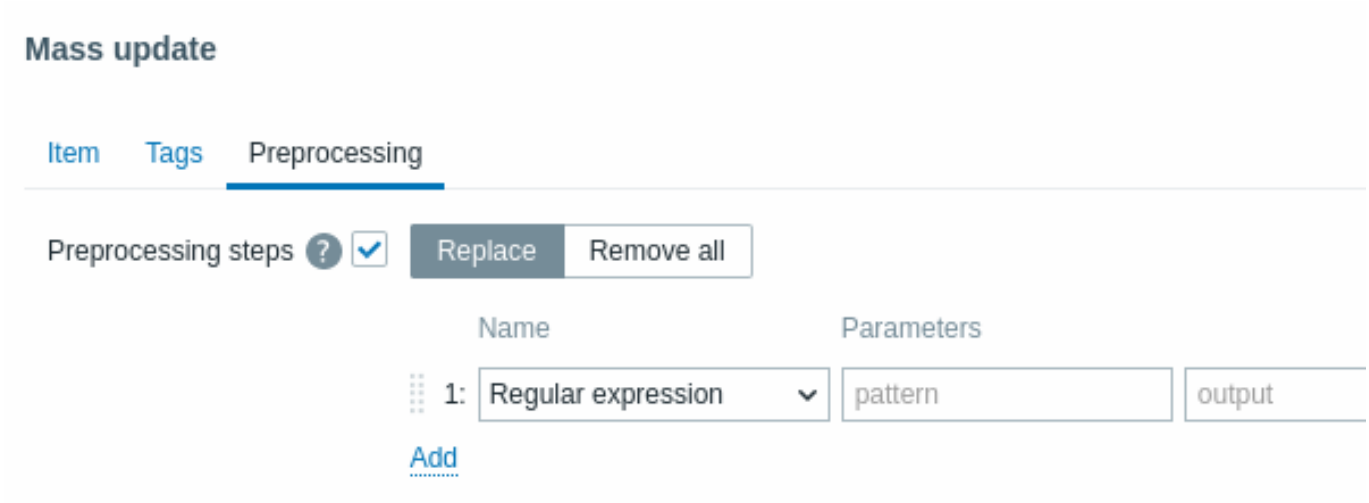




A opção *Tags* permite:

- *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 forma mais amigável:

- 0 → Não disponível
- 1 → Disponível

Os mapeamentos de valores para tipos de backup de dados podem ser configurados da seguinte forma:

- F → Completo
- D → Diferencial
- I → Incremental

Os intervalos de valores para tensão podem ser configurados da seguinte forma:

- <=209 → Baixo
- 210-230 → OK
- >=231 → Alto

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 mapeamento de valor - 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*.

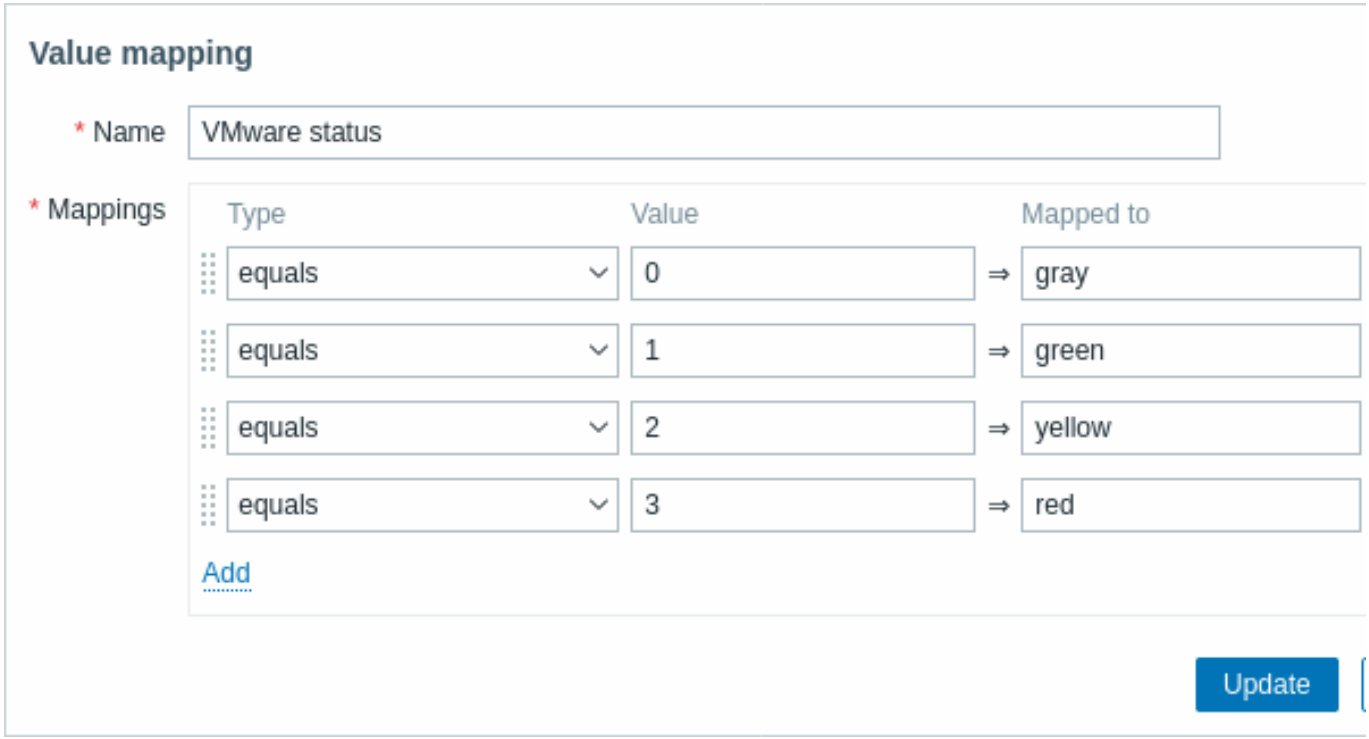
Os mapeamentos de valores são exportados/importados com o respectivo template ou host. Eles também podem ser atualizados em massa usando os formulários de atualização em massa de host e template.

Configuração

Para configurar um mapeamento de valor, siga estas etapas:

1. Abra o formulário de configuração do host ou template.

2. Na aba *Mapeamento de valor*, clique em *Adicionar* para adicionar um novo mapeamento de valor, ou clique no nome de um mapeamento existente para editá-lo.

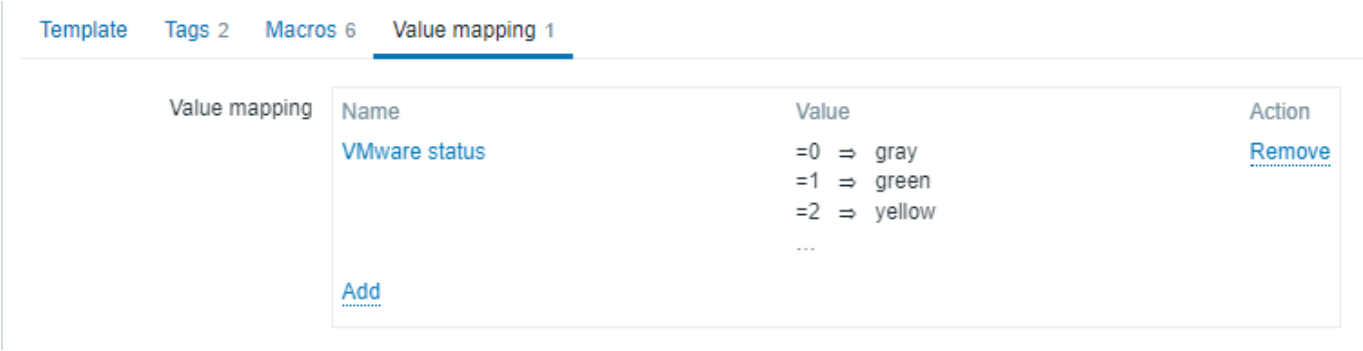


Parâmetros de um mapeamento de valor:

Parâmetro	Descrição
Nome	Nome exclusivo para o conjunto de mapeamentos de valor.
Mapeamentos	Regras individuais para mapear valores numéricos/strings para representações em string. O mapeamento é aplicado na ordem das regras, que podem ser reordenadas por arrastar.
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 no 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.

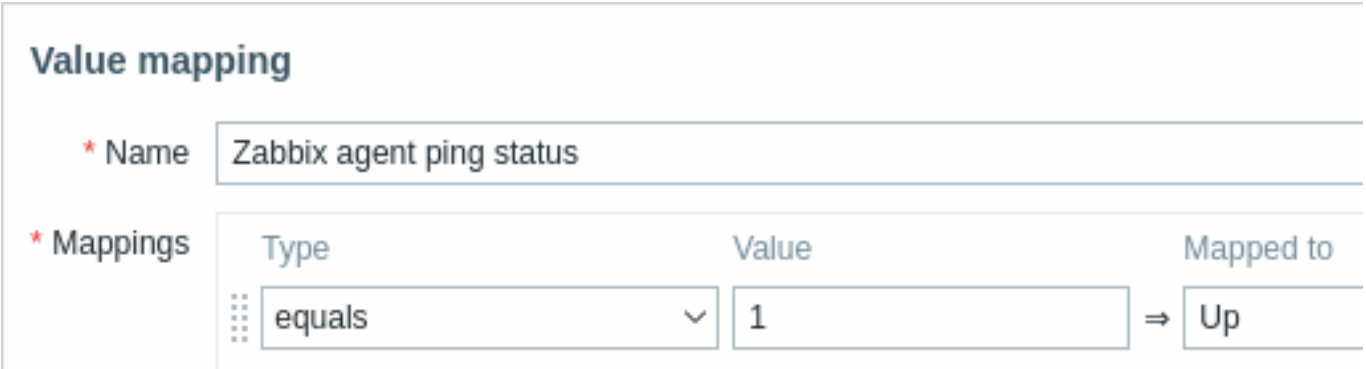
Todos os campos obrigatórios são marcados com um asterisco vermelho.

Ao visualizar o mapeamento de valor na lista, apenas os três primeiros mapeamentos são visíveis, com três pontos indicando que existem mais mapeamentos.



Exemplo de mapeamento de valores

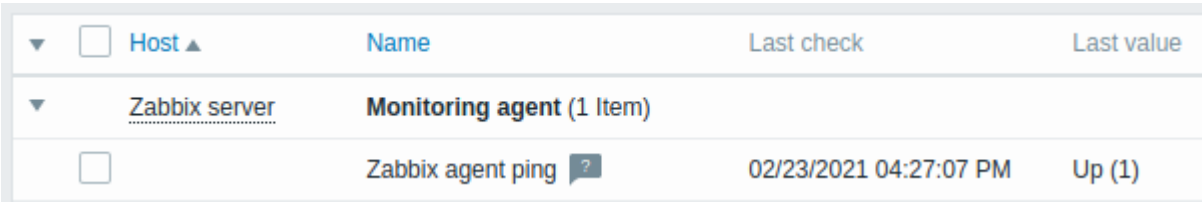
Um dos itens de agent predefinidos, *Zabbix agent ping*, utiliza um mapeamento de valores em nível de template "Zabbix agent ping status" para exibir seus valores.



No formulário de configuração do item, você pode encontrar uma referência a este mapeamento de valores no campo *Mapeamento de valores*:

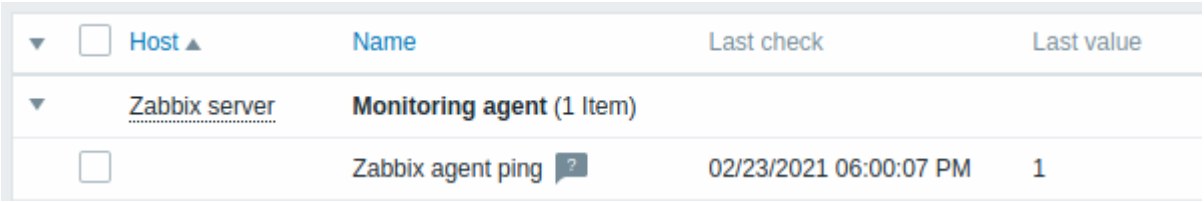


Este mapeamento é utilizado na seção *Monitoramento* → *Últimos dados* para exibir "Up" (com o valor bruto entre parênteses).



Note:
Na seção *Últimos dados*, os valores exibidos são reduzidos para 20 símbolos. Se o mapeamento de valores for utilizado, essa redução não é aplicada ao valor mapeado, mas apenas ao valor bruto (exibido entre parênteses).

Sem um mapeamento de valores 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 enfileirados pelo período de atualização dos dados históricos do proxy.

Apenas items com horários 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 log de eventos do agent Zabbix
- items de SNMP trap
- items de trapper
- items de monitoramento web
- items dependentes

As estatísticas mostradas pela fila são um bom indicador do desempenho do server Zabbix.

A fila é recuperada diretamente do server Zabbix usando o protocolo JSON. A informação está disponível apenas se o server Zabbix 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 assumir que o server está indo bem.

A fila mostra alguns items aguardando 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.

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.

Item de fila

Um item interno especial **zabbix[queue,<from>,<to>]** pode ser usado para monitorar a saúde da fila no Zabbix. Ele retornará o número de items atrasados pela quantidade de tempo definida. Para mais informações, consulte 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.

Esse 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 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 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 for do tipo ”Item dependente”, seu item mestre também deve ser um dos tipos acima.

Configuração

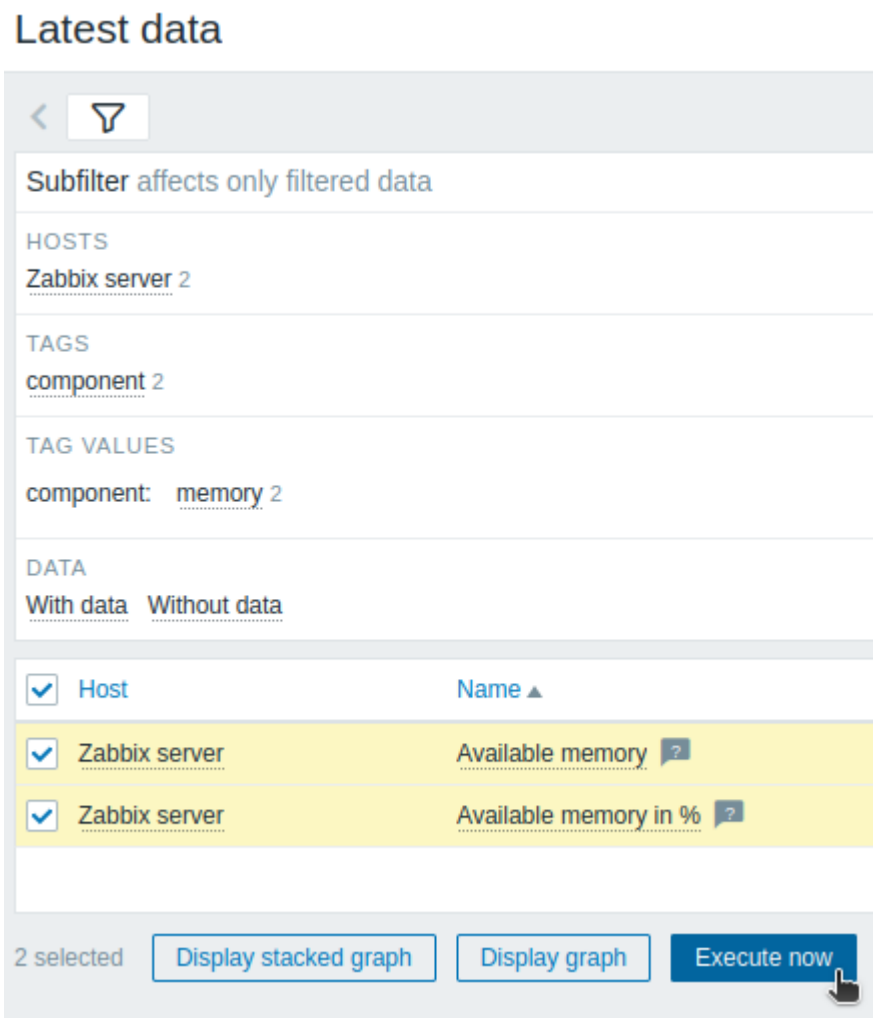
Você pode executar verificações passivas a partir de:

- *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

É possível restringir verificações no lado do agent criando uma blacklist de item, uma whitelist ou uma combinação de whitelist/blacklist.

Para isso, use uma combinação de dois parâmetros de configuração do agent:

- `AllowKey=<pattern>` - quais verificações são permitidas; <pattern> é especificado usando uma expressão curinga (*)
- `DenyKey=<pattern>` - quais verificações são negadas; <pattern> é especificado usando uma expressão curinga (*)

Observe que:

- Todos os itens system.run[*] (comandos remotos, scripts) são desabilitados por padrão, mesmo quando nenhuma chave de negação é especificada, deve-se assumir que DenyKey=system.run[*] é implicitamente adicionado.
- Desde o Zabbix 5.0.2, o parâmetro EnableRemoteCommands do agent está:
  - obsoleto para o Zabbix agent
  - não suportado pelo Zabbix agent2

Portanto, para permitir comandos remotos, especifique um AllowKey=system.run[<comando>,*] para cada comando permitido, * representa os modos wait e nowait. Também é possível especificar AllowKey=system.run[*] para permitir todos os comandos com os modos wait e nowait. Para negar comandos remotos específicos, adicione parâmetros DenyKey com comandos system.run[] antes do parâmetro AllowKey=system.run[*].

Regras importantes

- Uma whitelist sem uma regra de negação só é permitida para itens system.run[*]. Para todos os outros itens, os parâmetros AllowKey não são permitidos sem um parâmetro DenyKey; neste caso, o agent Zabbix **não iniciará** apenas com parâmetros AllowKey.
- A ordem importa. Os parâmetros especificados são verificados um a um de acordo com sua ordem de aparição no arquivo de configuração:
  - Assim que uma chave de item corresponder a uma regra de permissão/negação, o item será permitido ou negado; e a verificação de regras é interrompida. Portanto, se um item corresponder tanto a uma regra de permissão quanto a uma regra de negação, o resultado dependerá de qual regra vem primeiro.
  - A ordem também afeta o parâmetro EnableRemoteCommands (se usado).
- É suportado um número ilimitado de parâmetros AllowKey/DenyKey.
- As regras AllowKey, DenyKey não afetam os parâmetros de configuração HostnameItem, HostMetadataItem, HostInterfaceItem.
- O padrão de chave é uma expressão curinga onde o caractere curinga (*) corresponde a qualquer número de quaisquer caracteres em determinada posição. Pode ser usado tanto no nome da chave quanto nos parâmetros.
- Se uma chave de item específica for negada na configuração do agent, o item será relatado como não suportado (nenhuma dica é fornecida quanto ao motivo);
- O agent Zabbix com a opção de linha de comando --print (-p) não mostrará chaves que não são permitidas pela configuração;
- O agent Zabbix com a opção de linha de comando --test (-t) retornará o status "Unsupported item key." para chaves que não são permitidas pela configuração;
- Comandos remotos negados não serão registrados no log do agent (se LogRemoteCommands=1).

Casos de uso

Negar uma verificação específica

- Coloque uma verificação específica na lista negra com o parâmetro DenyKey. As keys correspondentes serão proibidas. Todas as keys que não corresponderem serão permitidas, exceto os items system.run[].

Por exemplo:

```
# Negar acesso a dados sensíveis
DenyKey=vfs.file.contents[/etc/passwd,*]
```

Attention:
Uma lista negra pode não ser uma boa escolha, pois uma nova versão do Zabbix pode ter novas keys que não estão explicitamente restritas pela configuração existente. Isso pode causar uma falha de segurança.

Negar comando específico, permitir outros

- Coloque um comando específico na lista negra com o parâmetro DenyKey. Coloque todos os outros comandos na lista branca, com o parâmetro AllowKey.

```
# Negar comando específico
DenyKey=system.run[ls -l /]

# Permitir outros scripts
AllowKey=system.run[*]
```

Permitir verificação específica, negar outras

- Permitir verificações específicas com os parâmetros AllowKey, negar outras com DenyKey=*

Por exemplo:

```
# Permitir leitura de logs:
AllowKey=vfs.file.*[/var/log/*]

# Permitir verificações de localtime
AllowKey=system.localtime[*]

# Negar todas as outras keys
DenyKey=*
```

Permitir script com espaços no caminho e argumentos

- Coloque na lista de permissões um script localizado em um caminho que contenha espaços e permita a passagem de argumentos para ele. No Windows, escape os espaços usando ^ ao especificar o caminho.

Por exemplo:

```
# Permitir a execução de test.bat com ou sem argumentos
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
```

Para testar:

```
PS C:\Program Files\Zabbix Agent 2> .\zabbix_get.exe -s 127.0.0.1 -k system.run["C:\Program^ Files\Zabbix^
```

Esta configuração permite a execução do script independentemente de argumentos serem passados (como `caret`) e garante que `system.run` seja permitido mesmo que o caminho do script contenha espaços.

Exemplos de padrões

Padrão	Descrição	Corresponde	Não corresponde
*	Corresponde a todas as possíveis keys com ou sem parâmetros.	Qualquer	Nenhuma
vfs.file.contents	Corresponde a `vfs.file.contents` sem parâmetros.	vfs.file.contents	vfs.file.contents[/etc/passwd]
vfs.file.contents[]	Corresponde a `vfs.file.contents` com parâmetros vazios.	vfs.file.contents[]	vfs.file.contents
vfs.file.contents[*]	Corresponde a `vfs.file.contents` com quaisquer parâmetros; não corresponderá a `vfs.file.contents` sem colchetes.	vfs.file.contents[] vfs.file.contents[/path/to/file]	vfs.file.contents
vfs.file.contents[/etc/passwd,]*	Corresponde a `vfs.file.contents` com o primeiro parâmetro igual a /etc/passwd e todos os outros parâmetros com qualquer valor (inclusive vazio).	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]	vfs.file.contents[/etc/passwd] vfs.file.contents[/var/log/zabbix_s vfs.file.contents[]
*vfs.file.contents[*passwd*]*	Corresponde a `vfs.file.contents` com o primeiro parâmetro igual a *passwd* e nenhum outro parâmetro.	vfs.file.contents[/etc/passwd]	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd, utf8]
*vfs.file.contents[*passwd*,*]*	Corresponde a `vfs.file.contents` com apenas o primeiro parâmetro igual a *passwd* e todos os parâmetros seguintes com qualquer valor (inclusive vazio).	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd, utf8]	vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/var/log/zabbix_server.log,,abc]*	Corresponde a `vfs.file.contents` com o primeiro parâmetro igual a /var/log/zabbix_server.log, o terceiro parâmetro igual a 'abc' e qualquer (inclusive vazio) segundo parâmetro.	vfs.file.contents[/var/log/zabbix_server.log,,abc] vfs.file.contents[/var/log/zabbix_server.log,utf8,abc]	vfs.file.contents[/var/log/zabbix_s
vfs.file.contents[/etc/passwd,utf8]	Corresponde a `vfs.file.contents` com o primeiro parâmetro igual a /etc/passwd, o segundo parâmetro igual a 'utf8' e nenhum outro argumento.	vfs.file.contents[/etc/passwd,utf8]	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf1
*vfs.file.**	Corresponde a qualquer key começando com `vfs.file.` sem quaisquer parâmetros.	vfs.file.contents vfs.file.size	vfs.file.contents[] vfs.file.size[/var/log/zabbix_serve
vfs.file.[*]*	Corresponde a qualquer key começando com `vfs.file.` com quaisquer parâmetros.	vfs.file.size.bytes[] vfs.file.size[/var/log/zabbix_server.log, utf8]	vfs.file.size.bytes
vfs..contents*	Corresponde a qualquer key começando com `vfs.` e terminando com `.contents` sem quaisquer parâmetros.	vfs.mount.point.file.contents vfs..contents	vfs.contents

system.run e AllowKey

Um script hipotético como 'myscript.sh' pode ser executado em um host via Zabbix agent de várias maneiras:

1. Como uma chave de item em uma verificação passiva ou ativa, por exemplo:

- system.run[myscript.sh]
- system.run[myscript.sh,wait]
- system.run[myscript.sh,nowait]

Aqui o usuário pode adicionar "wait", "nowait" ou omitir o 2º argumento para usar seu valor padrão em system.run[].

2. Como um script global (iniciado pelo usuário no frontend ou API).

Um usuário configura este script em *Alertas → Scripts*, define "Executar em: Zabbix agent" e coloca "myscript.sh" no campo de entrada "Comandos" do script. Quando invocado pelo frontend ou API, o Zabbix server envia para o agent:

- system.run[myscript.sh,wait] - até o Zabbix 5.0.4
- system.run[myscript.sh] - a partir do 5.0.5

Aqui o usuário não controla os parâmetros "wait"/"nowait".

3. Como um comando remoto de uma ação. O Zabbix server envia para o agent:

- system.run[myscript.sh,nowait]

Aqui novamente o usuário não controla os parâmetros ”wait”/”nowait”.

O que isso significa é que se definirmos AllowKey como:

```
AllowKey=system.run[myscript.sh]
```

então

- system.run[myscript.sh] - será permitido
- system.run[myscript.sh,wait], system.run[myscript.sh,nowait] não serão permitidos - o script não será executado se invocado como uma etapa de ação

Para permitir todas as variantes descritas, você pode adicionar:

```
AllowKey=system.run[myscript.sh,*]
DenyKey=system.run[*]
```

aos parâmetros do agent/agent2.

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 que mereça atenção. O trabalho de ”avaliar” os dados pode ser deixado para as 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 algum dado, 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 com valores binários.

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().

A maioria das funções de trigger são avaliadas com base nos dados de valor do item history, enquanto algumas funções de trigger para análise de longo prazo, por exemplo, **trendavg()**, **trendcount()**, etc, usam dados de tendência.

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 que está incluída na expressão é recalculada (não apenas a 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 data e hora e/ou nodata() forem usadas na expressão.

As funções 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 horário mais recente de recálculo do trigger (veja Tempo de cálculo acima); ”agora” não é o horário atual 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 num do passado, até ”agora”
  - Se houver 0 valores disponíveis para o período de tempo ou quantidade num especificados - então 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 recentemente 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 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 existam algumas exceções). Para mais detalhes, consulte Expressões com operandos desconhecidos.

É possível ser notificado sobre triggers desconhecidas.

1 Configurando um trigger

Visão geral

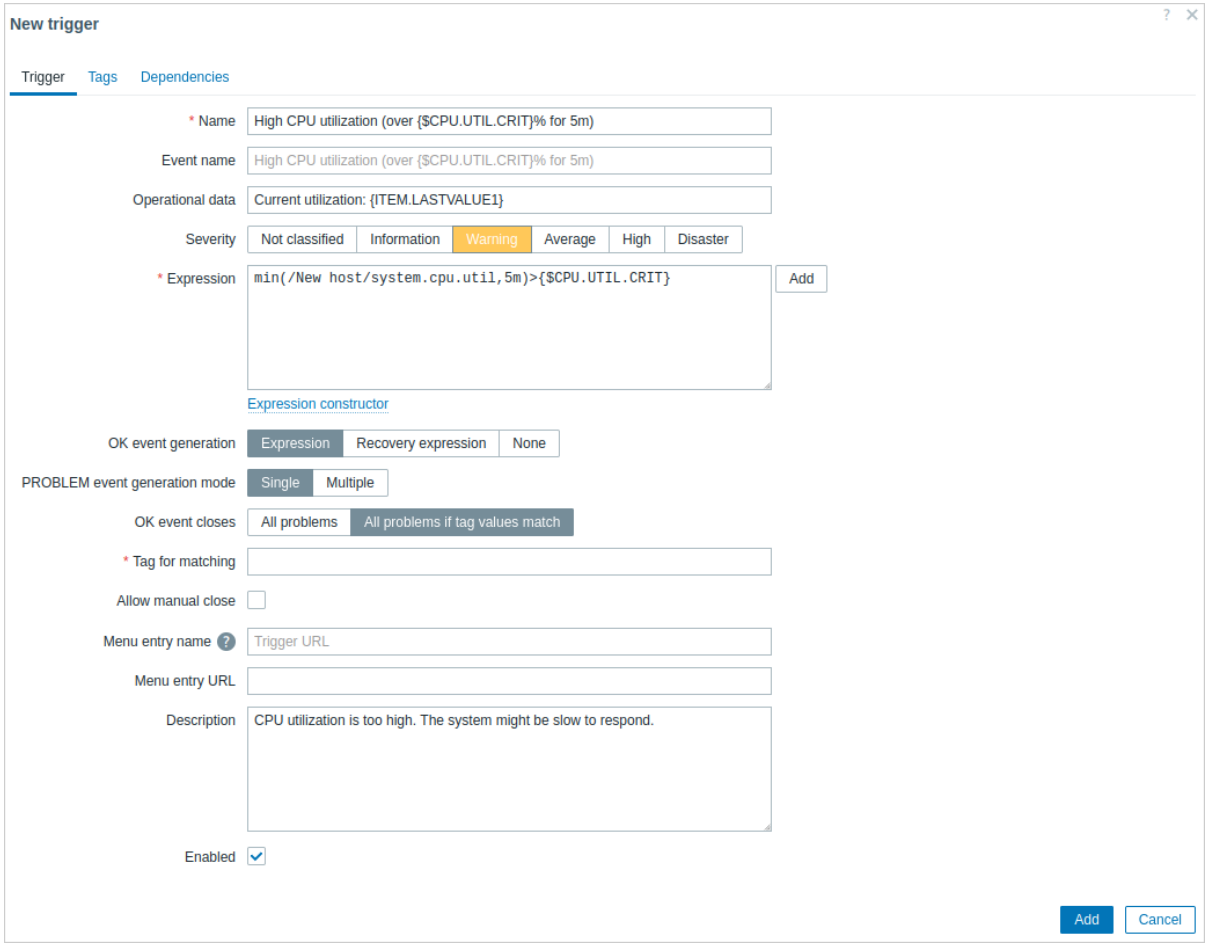
Para configurar um trigger, faça o seguinte:

- 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.

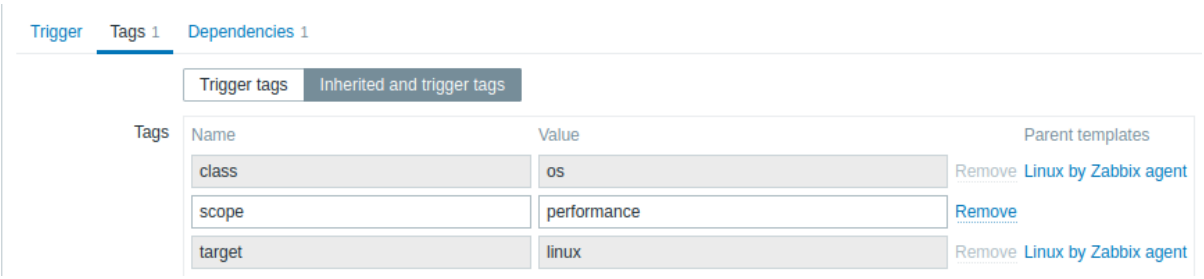


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome do trigger. As macros suportadas são: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*} e 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	Se definido, este nome será usado para criar o nome do evento de problema, em vez do 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}, {TIMESTAMP} 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.

Parâmetro	Descrição
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; **Nenhuma** - 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, depois que a expressão de problema original já tenha 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 de 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 vazio, o nome padrão *Trigger URL* é usado. O mesmo conjunto de macros é suportado como na URL do trigger.
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 do 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}}` e macros de descoberta de baixo nível 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 nos valores das tags do 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.

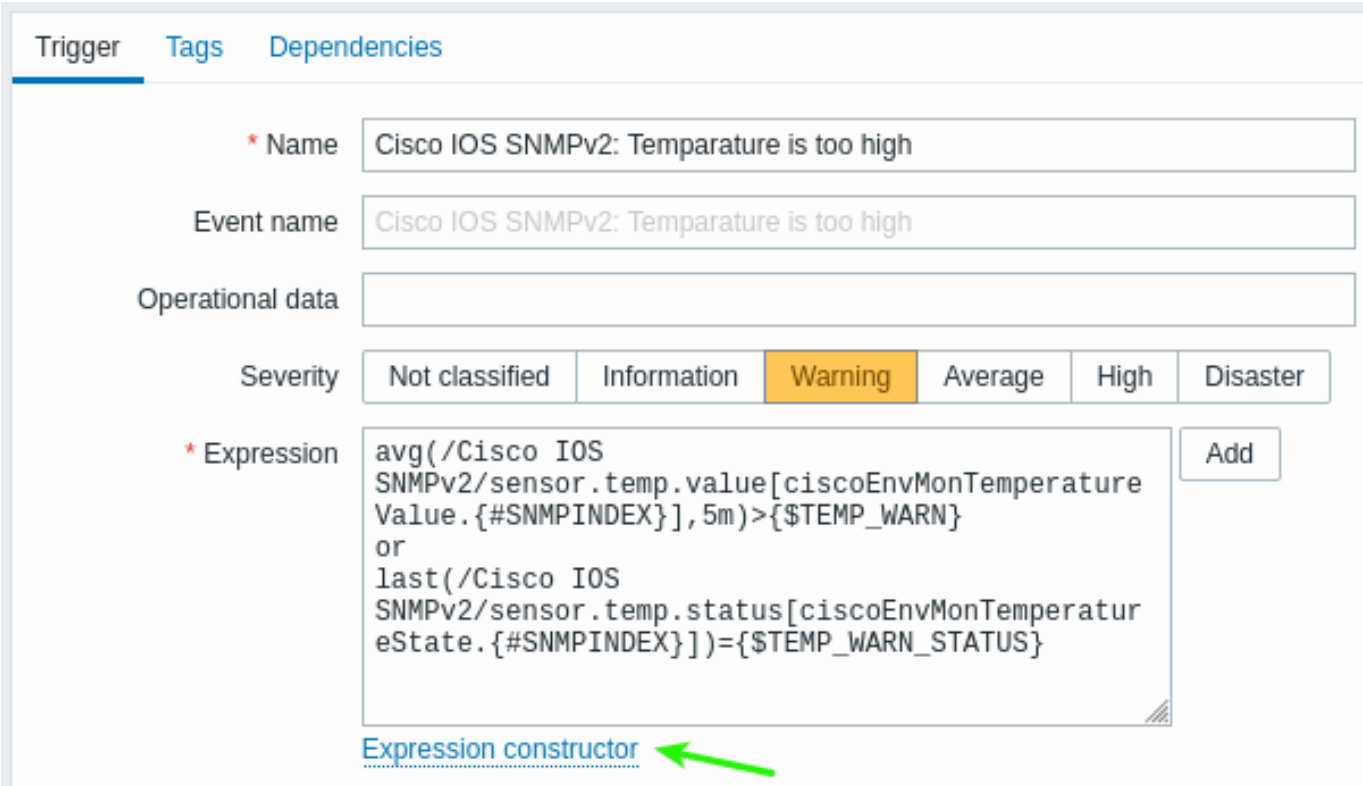
Testando expressões

É possível testar a expressão de trigger configurada para verificar qual seria o resultado da expressão dependendo do valor recebido.

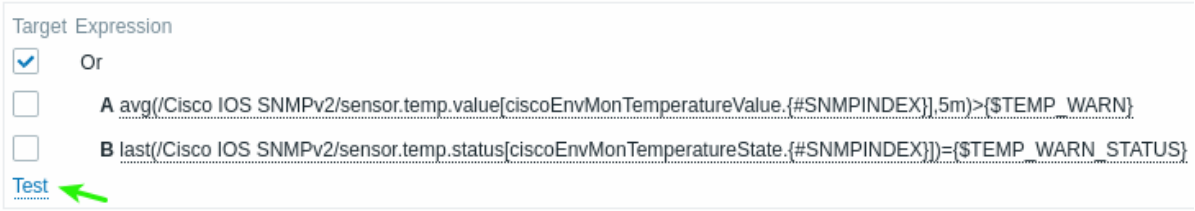
A seguinte expressão de um template oficial é usada como exemplo:

```
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}
```

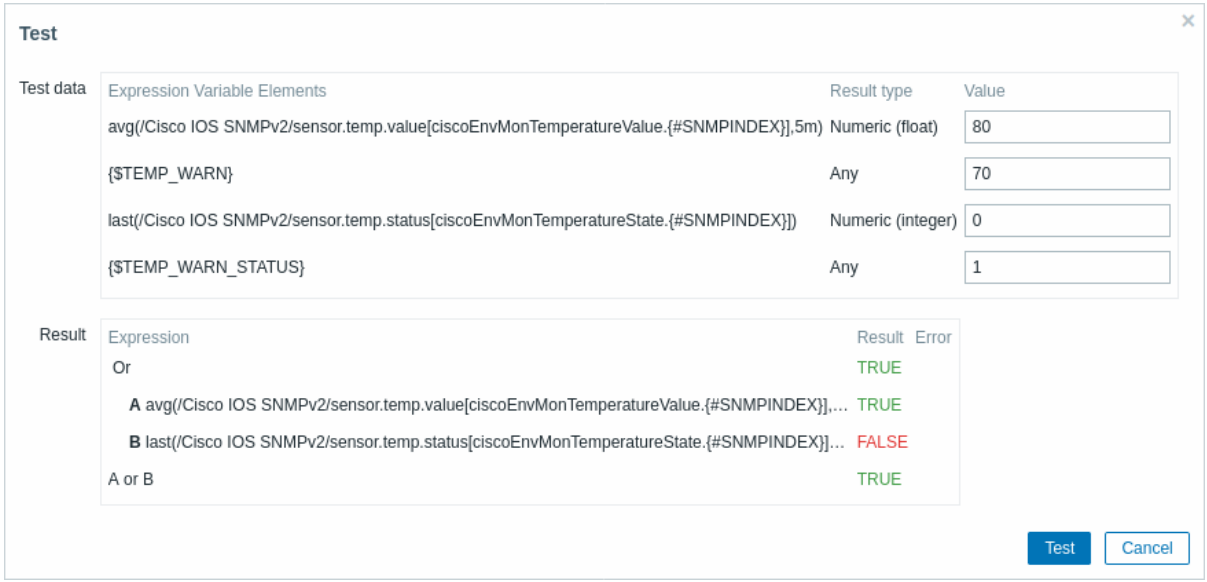
Para testar a expressão, clique em *Construtor de expressão* abaixo do campo de expressão.



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 visualizado.

”TRUE” significa que a expressão especificada está correta. Neste caso particular A, ”80” é maior que o valor especificado em {$TEMP_WARN}, ”70” neste exemplo. Como esperado, um resultado ”TRUE” aparece.

”FALSE” significa que a expressão especificada está incorreta. Neste caso particular B, {$TEMP_WARN_STATUS} ”1” precisa ser igual ao valor especificado, ”0” neste exemplo. Como esperado, um resultado ”FALSE” aparece.

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 acionado 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 o limite do problema quanto como o 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 oscilações do trigger.

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), localizar 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/chave`.
A omissão do nome do host no primeiro parâmetro (ou seja, como em `function(//chave,parâmetro,...)`) só é suportada em certos contextos:

- Na fórmula de itens 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 usada 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 da 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 que não usam histórico em triggers, essa limitação não se aplica em itens 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. Esse 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 de uma hora há 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**	NÃO 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	+	Mais aritmético	1.2 + Desconhecido → Desconhecido	Sim
	-	Menos aritmético	1.2 - Desconhecido → Desconhecido	Sim

Prioridade	Operador	Definição	**Notas para valores desconhecidos**	Forçar conversão do operando para float [1]
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]
	<>	Diferente. O operador é definido como: A<>B ⇔ (A<B-0.000001) ou (A>B+0.000001)		Não [1]
7	**and**	E lógico	0 **and** Desconhecido → 0 1 **and** Desconhecido → Desconhecido Desconhecido **and** Desconhecido → Desconhecido	Sim
8	**or**	OU 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

(Se a conversão falhar - o operando numérico é convertido para string e ambos os operandos são comparados como strings.)

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 retornar '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 retornar '1' se ambos os operandos forem diferentes de '0'; caso contrário, retorna '0'; para operandos desconhecidos, **and** retorna '0' apenas se um operando for igual a '0'; caso contrário, retorna 'Desconhecido';
- **or** para operandos conhecidos deve retornar '1' se qualquer um dos operandos for diferente de '0'; caso contrário, retorna '0'; para operandos desconhecidos, **or** retorna '1' apenas se um operando for diferente de '0'; caso contrário, retorna '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** retorna '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 do 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 PROBLEMA 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 maior que 5 ou a carga do processador foi maior que 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 batimentos cardíacos nos últimos 3 minutos.

Uso da função nodata():

```
nodata(/example.example.com/tick,3m)=1
```

Para usar 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 troca da semana (domingo, 23:00 - segunda-feira, 01:00).

Exemplo 11

Verifique se a hora local do cliente está sincronizada com a hora do Zabbix server.

Uso da função fuzzytime():

```
fuzzytime(/MySQL_DB/system.localtime,10s)=0
```

O trigger mudará para o estado de problema caso a hora local no server MySQL_DB e no Zabbix server difira em mais de 10 segundos. Observe que 'system.localtime' deve ser configurado como uma verificação passiva.

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 disparado 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 divergiu 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) escrever 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) usar 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.

Histerese

Às vezes, é necessário um intervalo entre os estados de problema e recuperação, em vez de um simples limite. 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 limite 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 ser 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 dos operandos desconhecidos:
  - **Caso 1**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" pode ser avaliado para um resultado conhecido ('1' ou "Problema"),
  - **Caso 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) 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 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), permanecerá 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 aí que alguma dependência entre hosts pode ser útil. Com a dependência definida, as notificações dos dependentes podem ser retidas e apenas a notificação sobre o 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.

Então, 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 houver dependência, 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, 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 date and time 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 depende 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 protótipo de trigger. Um protótipo de trigger de host não pode depender de um trigger de um template.

Configuração

Para definir uma dependência, abra a aba Dependências no formulário de configuração do trigger. Clique em *Adicionar* no bloco 'Dependências' e selecione um ou mais triggers dos quais o trigger irá depender.

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 fora do ar, 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 fora do ar é excessivo.

Portanto, neste caso, definimos duas dependências:

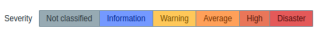
```
o trigger 'Host está fora do ar' depende do trigger 'Router2 está fora do ar'
o trigger 'Router2 está fora do ar' depende do trigger 'Router1 está fora do ar'
```

Antes de alterar o status do trigger 'Host está fora do ar', o Zabbix verificará 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 recursivamente. Se o Router1 ou o Router2 estiverem inacessíveis, o trigger do Host não será atualizado.

4 Severidade do trigger

A severidade do trigger representa o nível de importância de um trigger.



O Zabbix suporta as seguintes severidades de trigger por padrão.

Severidade	Cor	Descrição
Não classificado	Cinza	Pode ser usado quando o nível de severidade 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 usado para eventos informativos que não exigem atenção imediata, mas que ainda podem fornecer informações valiosas.
Aviso	Amarelo	Pode ser usado para indicar um possível problema que pode exigir investigação ou ação, mas que não é crítico.
Média	Laranja	Pode ser usado para indicar um problema significativo que deve ser resolvido relativamente em breve para evitar problemas adicionais.
Alta	Vermelho claro	Pode ser usado para indicar problemas críticos que precisam de atenção imediata para evitar interrupções significativas.
Desastre	Vermelho	Pode ser usado para indicar um incidente grave que requer ação imediata para evitar, por exemplo, paralisação do sistema ou perda de dados.

Note:
Os nomes e cores das severidades dos triggers podem ser personalizados.

As severidades dos triggers são usadas para:

- representação visual dos triggers - cores diferentes para diferentes severidades;
- áudio em alarmes globais - diferentes áudios para diferentes severidades;
- mídia do usuário - diferentes mídias (canal de notificação) para diferentes severidades (por exemplo, SMS para triggers de severidade *Alta* e *Desastre*, e E-mail para triggers de outras severidades);
- limitar ações por condições baseadas nas severidades dos triggers.

5 Personalizando severidades de trigger

Os nomes e cores das severidades de trigger para elementos relacionados à severidade na interface gráfica podem ser configurados em *Administração → Geral → Opções de exibição de trigger*. As cores são compartilhadas entre todos os temas da interface.

Traduzindo nomes personalizados de severidade

> **Attention:**
> Se traduções do frontend do Zabbix forem usadas, 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 desejado, 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 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

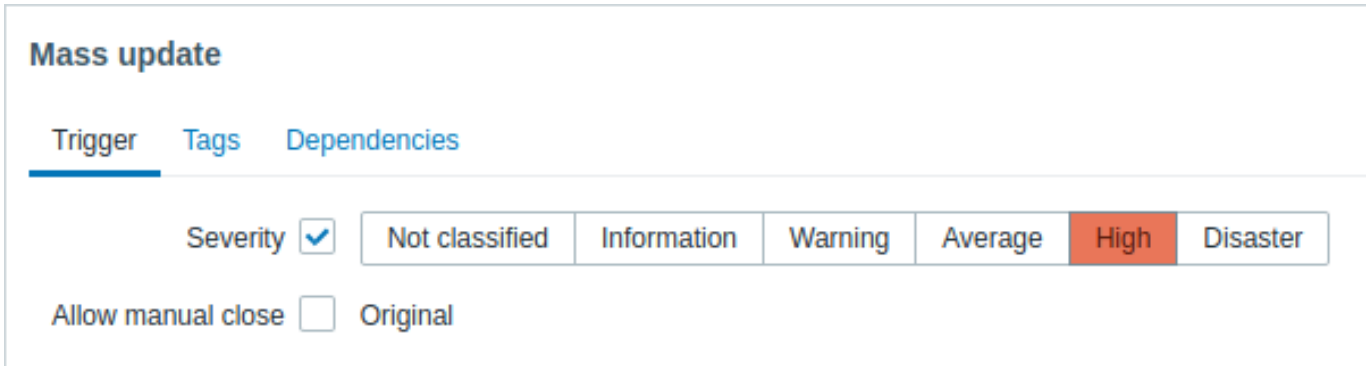
Visão geral

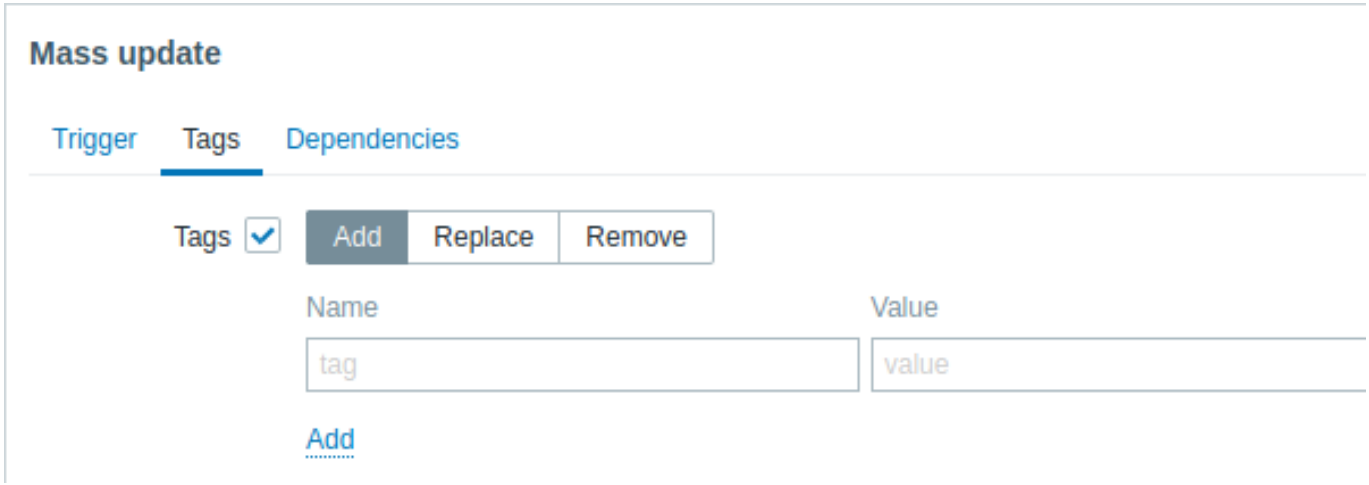
Com a atualização em massa, você pode alterar alguns atributos de vários triggers de uma só vez, poupando o trabalho de abrir cada trigger individualmente para edição.

Usando atualização em massa

Para atualizar em massa alguns triggers, faça o seguinte:

- Marque as caixas de seleção dos triggers que deseja atualizar na lista
- Clique em *Atualizar em massa* abaixo da lista
- Navegue até a aba com os atributos necessários (*Trigger*, *Tags* ou *Dependências*)
- Marque as caixas de seleção de qualquer atributo a ser atualizado

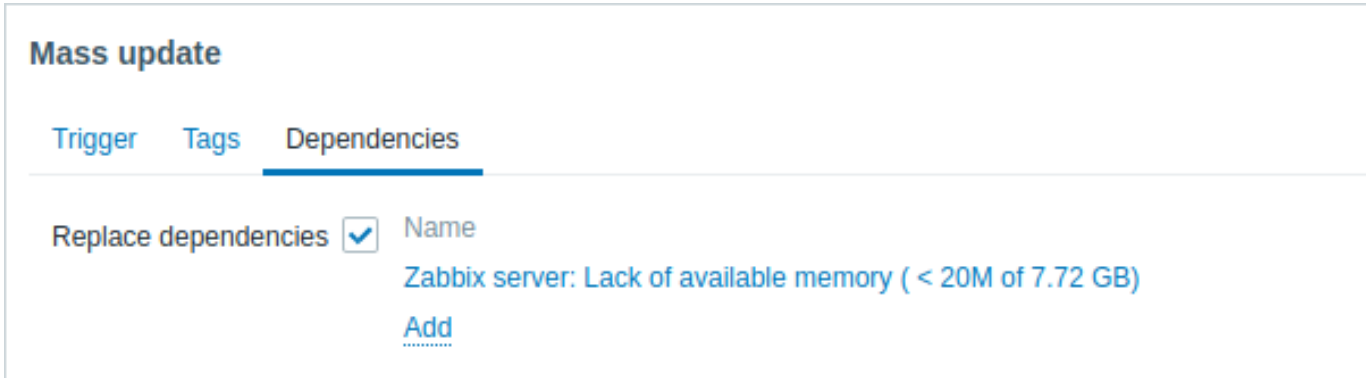




As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de tags:

- *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 de trigger preditivas

Visão geral

Às vezes, há sinais de um problema iminente. Esses sinais podem ser detectados 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. Então, 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 de qualquer 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

Antes de tudo, você deve especificar o período histórico que o Zabbix deve analisar para gerar a previsão. Você faz isso de uma maneira já conhecida, usando o parâmetro `time period` e 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 importando 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 convencionais, e 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`	**forecast** resultado
value	f(agora + `time`)
max	$\max_{\text{agora} <= t <= \text{agora} + \texttt{time}}$ f(t)
min	$\min_{\text{agora} <= t <= \text{agora} + \texttt{time}}$ 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 está na ordem de 10^9, a época ao quadrado é 10^{18}, a precisão double 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 sua trigger em um estado de problema. Geralmente isso é bom porque você recebe um aviso de que suas previsões não estão funcionando corretamente e 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 complexa [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 a 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, uma 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, de repente, se recuperar 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, nesse caso, -1 pode fazer com que sua 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 agents ativos são auto-registrados 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 carimbados com data/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 de estado, um evento é gerado. O evento contém detalhes da alteração do estado do trigger - quando aconteceu 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 manualmente fechado

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 o critério de recuperação do trigger for diferente do critério 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 de '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 do trigger 'Tag para correspondência'. 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 pareados 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 caso, 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 opção 'Permitir fechamento manual' estiver habilitada 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 será 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 periodicamente escaneia 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 está ATIVO para o IP.
Host Inativo	Se todos os serviços não estão respondendo.
Serviço Descoberto	Se o serviço voltou após uma indisponibilidade ou foi descoberto pela primeira vez.
Serviço Perdido	Se o serviço foi perdido após estar ativo.
Host Descoberto	Se o host voltou após uma indisponibilidade ou foi descoberto pela primeira vez.
Host Perdido	Se o host foi perdido após estar ativo.

Eventos de autoregistro de agent ativo

O autoregistro de agent ativo cria eventos no Zabbix.

Se configurado, um evento de autoregistro 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 autoregistrado, 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 acontecem 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 torna não suportado e para 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 Fechamento 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 desempenho ideal. Nesse caso, o problema é relatado aos administradores Linux, eles o corrigem e depois 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 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 funções de data e hora e/ou nodata() são usadas na expressão.

Configuração

São necessárias duas etapas para fechar um problema manualmente.

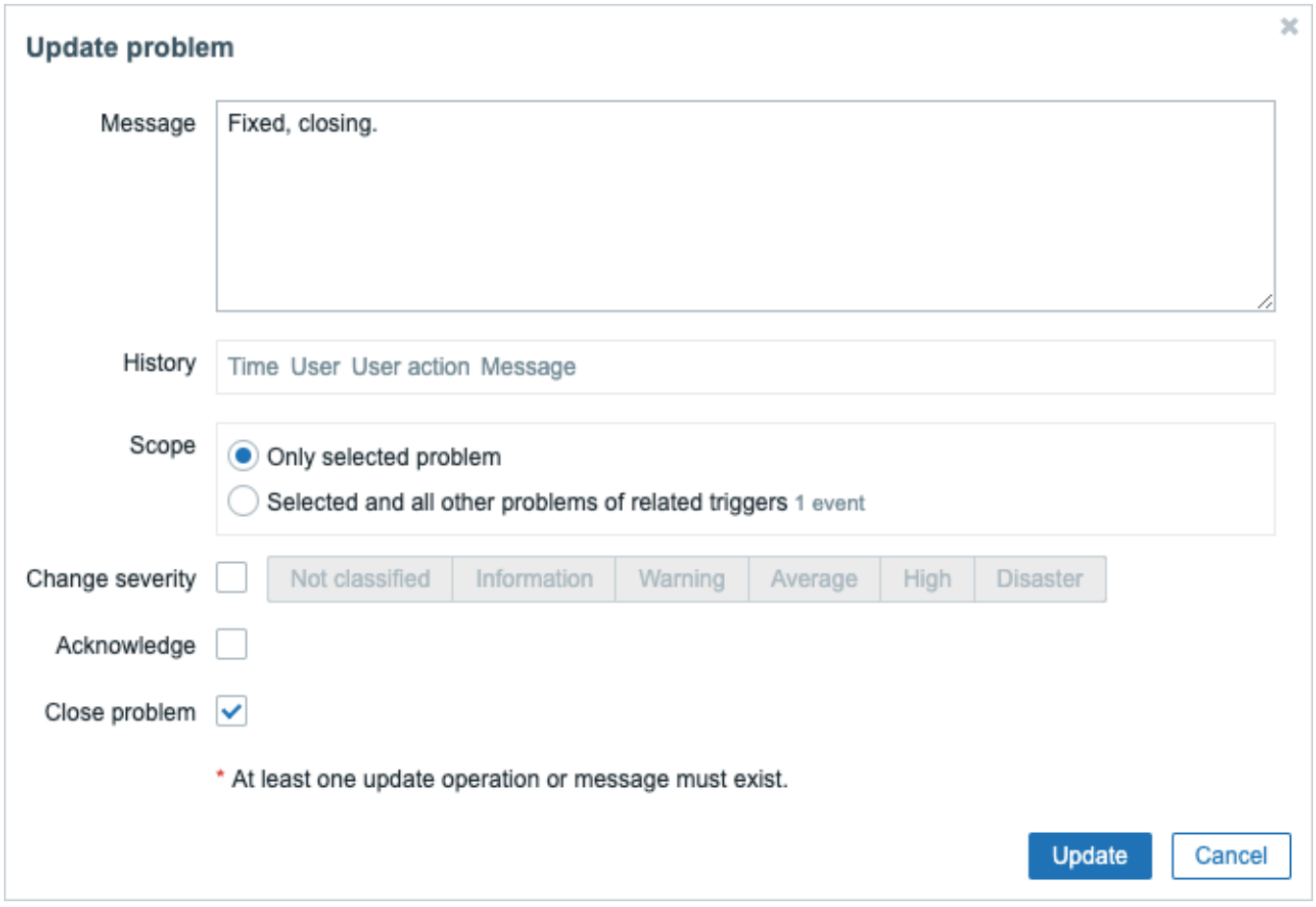
Configuração de trigger

Na configuração de trigger, habilite a opção *Permitir fechamento manual*.

Janela de atualização de problema

Se um problema surgir para um trigger com o parâmetro *Fechamento manual* ativado, você pode abrir a janela popup 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 maneira 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 globais

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 PROBLEM* 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 a 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
```

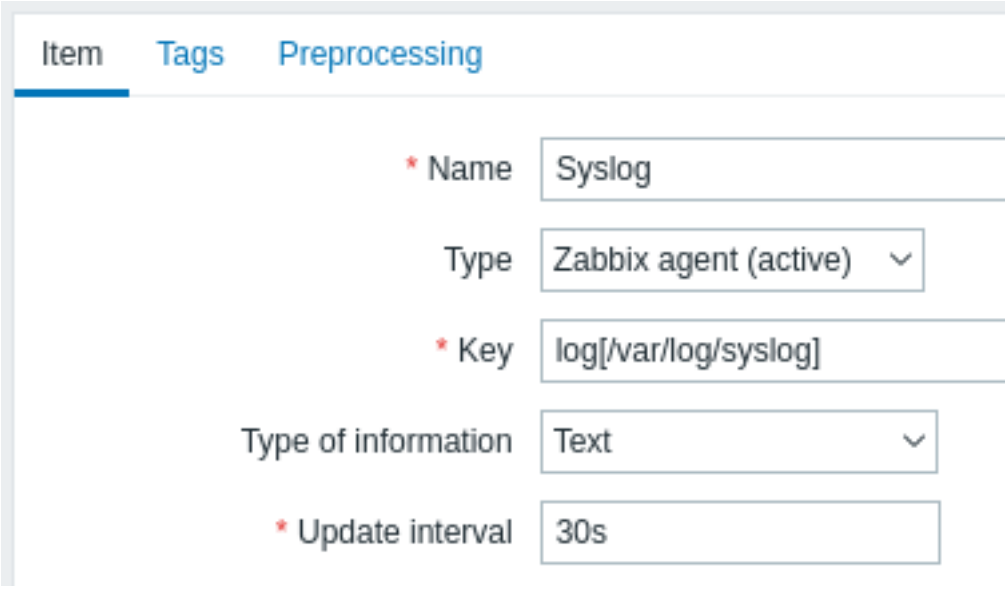
Para fazer isso, você precisa marcar esses eventos relacionados como, por exemplo, ”Serviço 1” e ”Serviço 2”. Isso pode ser feito aplicando uma expressão regular à linha do log para extrair o valor da tag. Então, quando os eventos são criados, eles são marcados como ”Serviço 1” e ”Serviço 2” respectivamente 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 valem a pena prestar 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'.

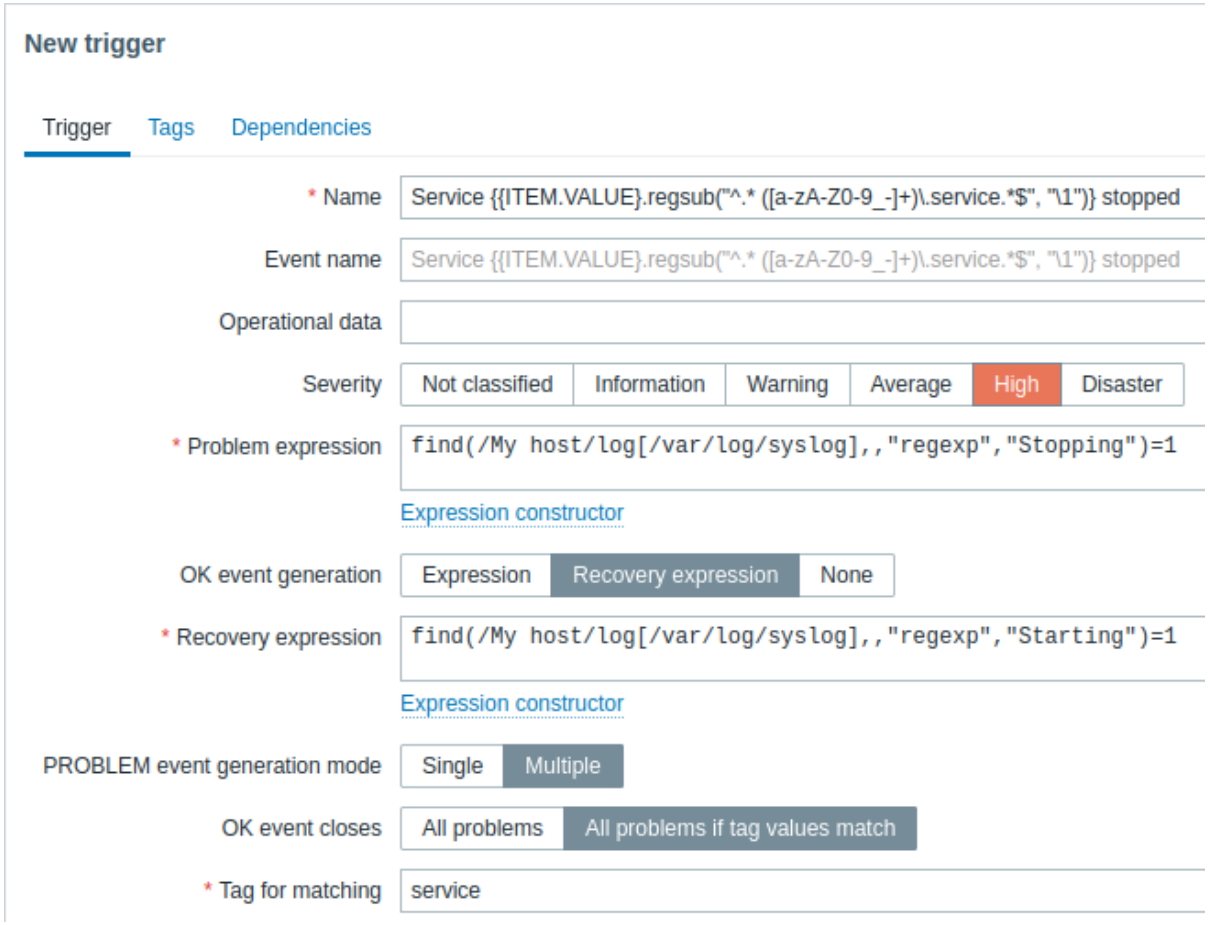
Em seguida, defina uma expressão de recuperação. A seguinte expressão de recuperação resolverá todos os problemas se uma linha de log for encontrada contendo a string "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

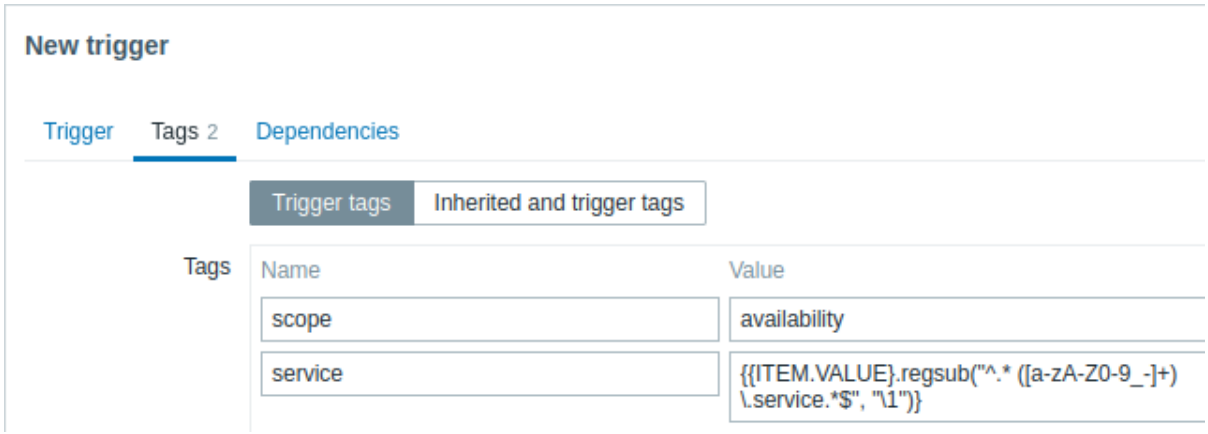
Como não queremos isso, é importante garantir de alguma forma que os problemas raiz correspondentes sejam fechados, não apenas todos os problemas. É aí que a marcação pode ajudar.

Problemas e resoluções podem ser correspondidos especificando uma tag na configuração do trigger. As seguintes configurações devem ser feitas:

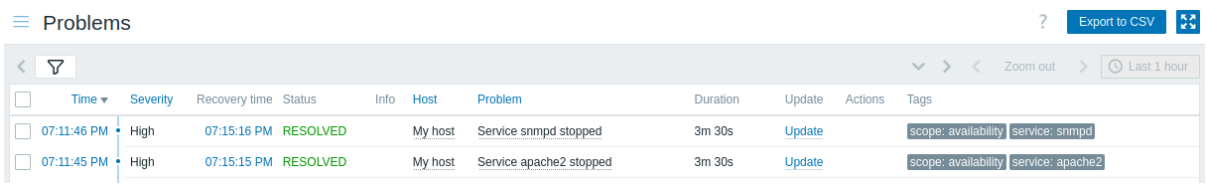
- *Modo de geração de evento de problema*: Múltiplos
- *Evento OK fecha*: Todos os problemas se os valores das tags corresponderem
- Insira o nome da tag para correspondência de eventos



- 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 evento semelhantes podem ser criadas para problemas **não relacionados**, 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 fechar uma mensagem de erro do serviço B.
- Tags e valores de tags reais 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 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.

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:

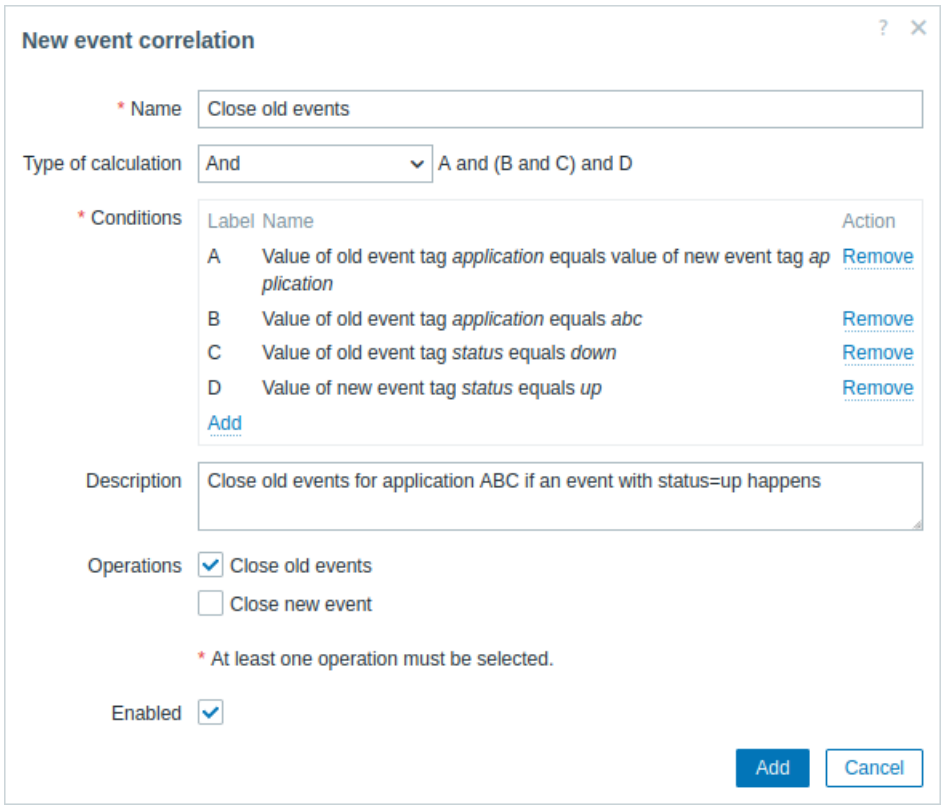
- Reduza o escopo da correlação. Sempre defina uma tag exclusiva para o novo evento que será pareado com eventos antigos e use a condição de correlação *Nome da tag do novo evento*;
- Adicione uma condição baseada no evento antigo ao usar a operação *Fechar evento antigo* (caso contrário, todos os problemas existentes podem ser fechados);
- 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:

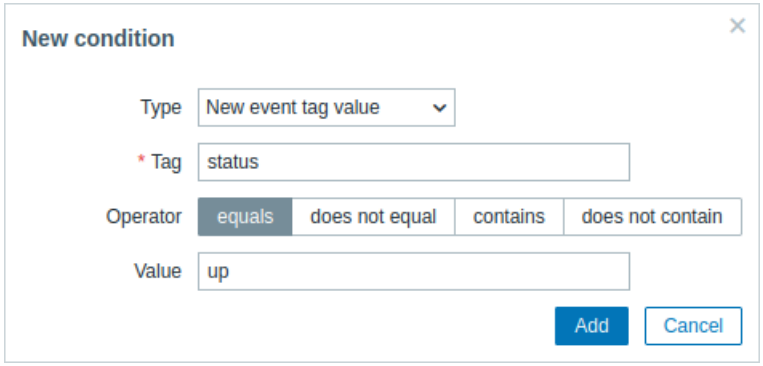
- Vá para *Coleta de dados* > *Correlação de eventos*
- Clique em *Criar correlação de eventos* à direita (ou no nome da correlação para editar uma regra existente)
- Insira os parâmetros da regra de correlação no formulário



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/minúsculas), **or** (diferencia maiúsculas/minúsculas), **not** (diferencia maiúsculas/minúsculas).
Condições	Lista de condições. Veja abaixo para detalhes sobre como configurar uma condição.
Descrição	Descrição da regra de correlação.
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
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 da tag do evento antigo e o valor para correspondência, usando os seguintes operadores: *igual* - possui o valor da tag do evento antigo *diferente* - 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 da tag do novo evento e o valor para correspondência, usando os seguintes operadores: *igual* - possui o valor da tag do novo evento *diferente* - 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 eventos semelhantes podem ser criadas para problemas **não relacionados**, revise os casos descritos abaixo!

- As tags e valores de tags reais 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 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 eventos 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 Tagging

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 emparelhá-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 macros resolvidas) 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 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 a utilização de tags são os seguintes:

1. Marcar eventos de trigger:
   - Defina uma tag de trigger (por exemplo, `scope:performance`).
   - Os problemas criados por essa 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 desse 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 desse host terão a tag de host.
4. Filtrar items relacionados:
   - Defina uma tag de item (por exemplo, `component:cpu`).
   - Em *Monitoramento* → *Últimos dados*, os items podem ser filtrados pela tag `component:cpu`.
5. Usar informações extraídas do valor do item como valor da tag:
   - Defina uma tag com um 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 um 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 coincidirem";
        - 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 coincidirem com os dados da tag.
8. Identificar problemas nas notificações:
    - Defina tags de trigger.
    - Use o 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.
    - Triggers criadas a partir desse trigger de template terão sua tag.
10. Criar triggers com tags a partir de low-level discovery (LLD):
    - Defina uma tag de protótipo de trigger com um macro LLD no nome ou valor da tag (por exemplo, `scope:{#FSNAME}`).
    - Triggers criadas 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`).
    - 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.
    - Problemas com as tags definidas serão suprimidos.
14. Conceder acesso a grupos de usuários:
    - Defina tags na configuração do grupo de usuários.
    - Usuários no 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.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} 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 protótipo de host e 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 macro função. 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.

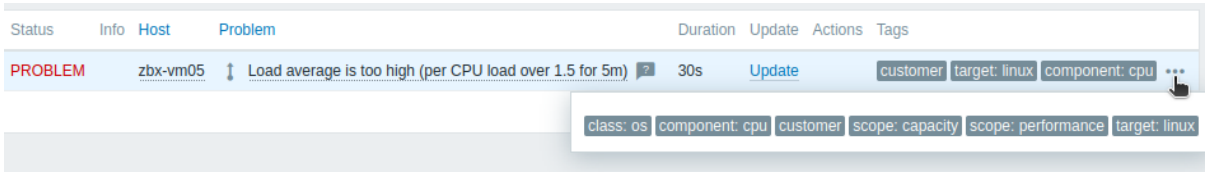
Veja também: Usando funções de macro em macros de descoberta de baixo nível para marcação de eventos.

Visualizando tags de eventos

As tags, se definidas, podem ser visualizadas com novos eventos em:

- *Monitoramento* → *Problemas*
- *Monitoramento* → *Problemas* → *Detalhes do evento*
- *Dashboards* → widget *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 oferece 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.

Nenhuma configuração é necessária 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 de 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 e host.

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 valores máximos e mínimos é preenchido com fundo amarelo.

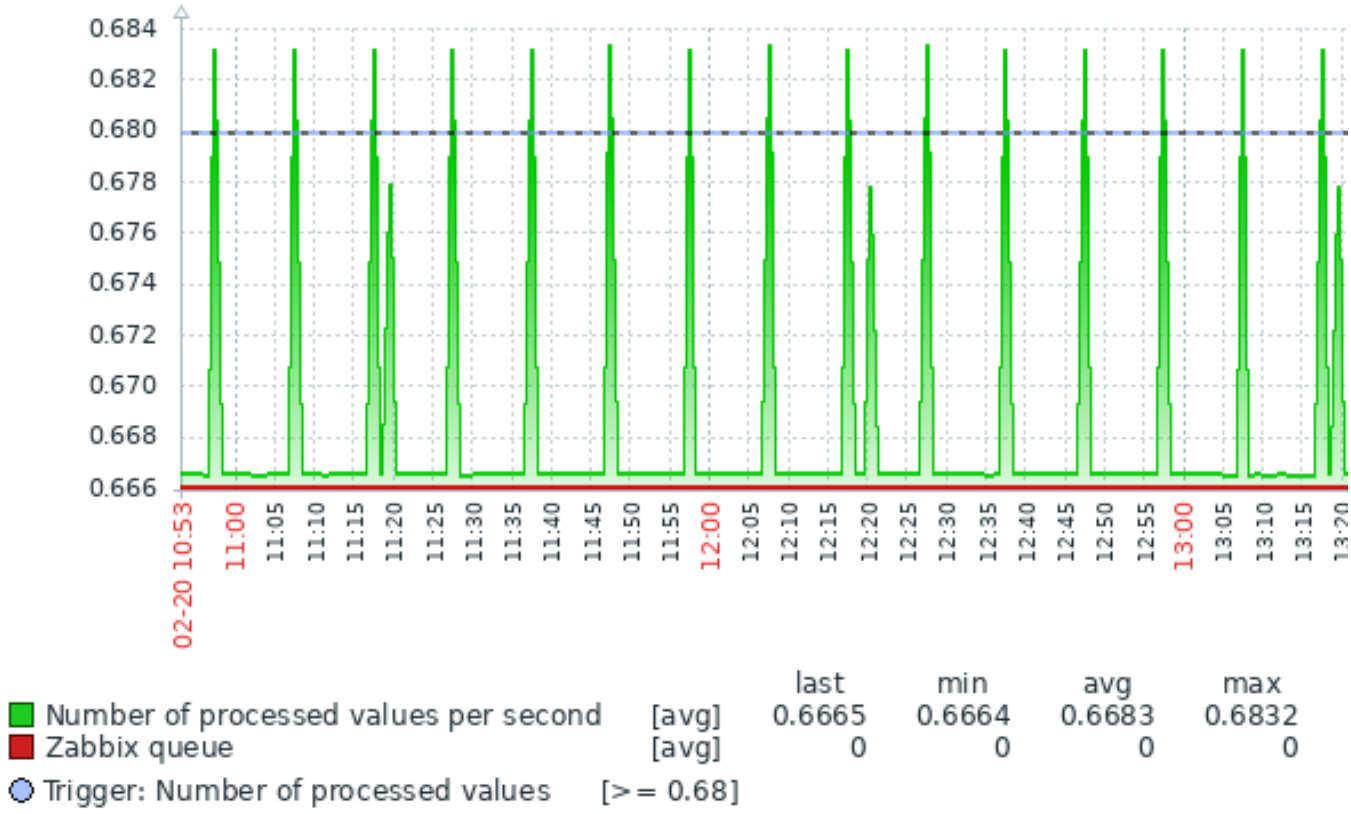
O tempo de trabalho (dias úteis) é exibido nos gráficos como fundo branco, enquanto o tempo não útil é exibido em cinza (com o tema padrão *Original blue* do frontend).

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 são priorizados. 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 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 que 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 tendência 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 últimos valores*. 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 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 mudar 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.

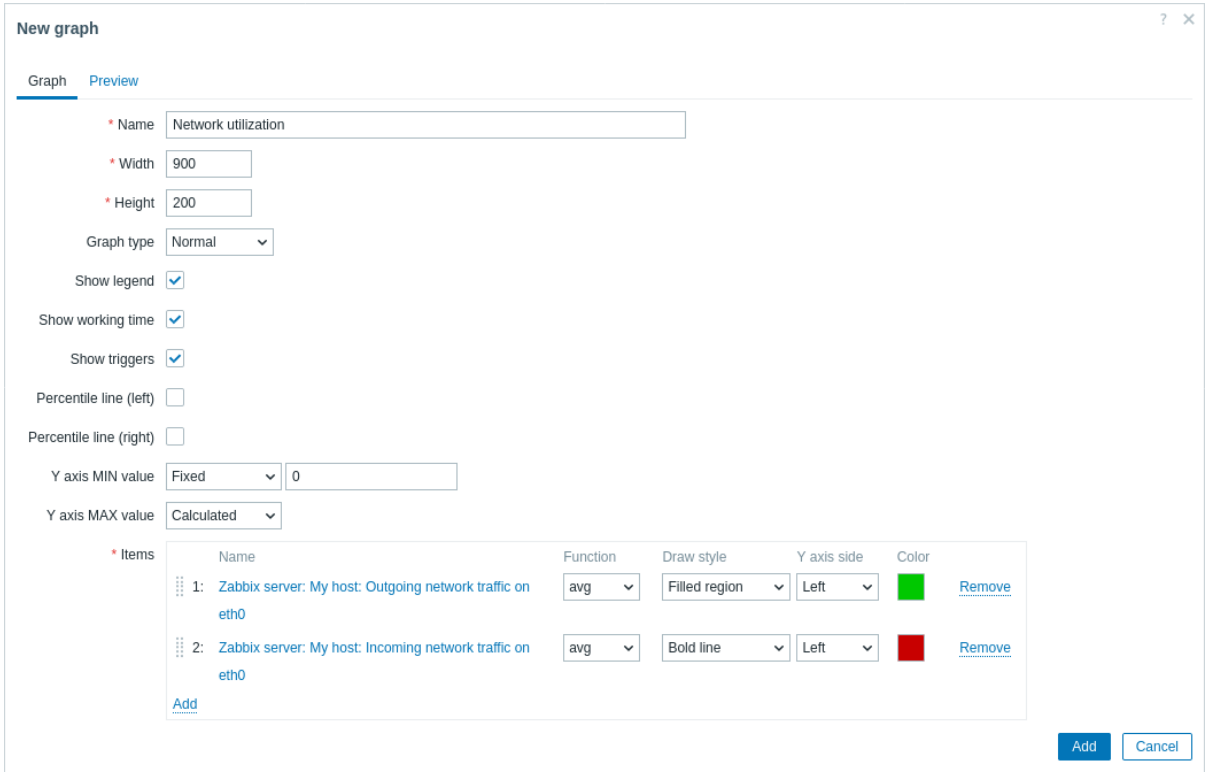
Gráficos personalizados são configurados manualmente.

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 exibidas 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 de percentil (esquerda)	Exibir percentil para o eixo Y esquerdo. Se, por exemplo, o percentil de 95% for definido, a linha de percentil estará no nível onde 95% dos valores ficam abaixo. Exibido como uma linha verde brilhante. Disponível apenas para gráficos normais.

Parâmetro	Descrição
Linha de percentil (direita)	Exibir percentil para o eixo Y direito. Se, por exemplo, o percentil de 95% for definido, a linha de percentil estará no nível onde 95% dos valores ficam abaixo. Exibido como uma linha vermelha brilhante. Disponível apenas para gráficos normais.
Valor MIN 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 MAX 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	Habilitar 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 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 em 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 às informações do trigger exibidas na legenda.

Note:
Não mais do que 3 linhas de trigger podem ser exibidas. Se houver mais triggers, então os triggers com menor severidade são priorizados 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

Enquanto um gráfico simples é ótimo para acessar dados de um item e gráficos personalizados oferecem opções de personalização, nenhum dos dois permite criar rapidamente um gráfico de comparação para vários items com pouco esforço e sem manutenção.

Para resolver esse problema, é possível criar gráficos ad-hoc para vários items 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 desejados
- Marque as caixas dos items que você 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

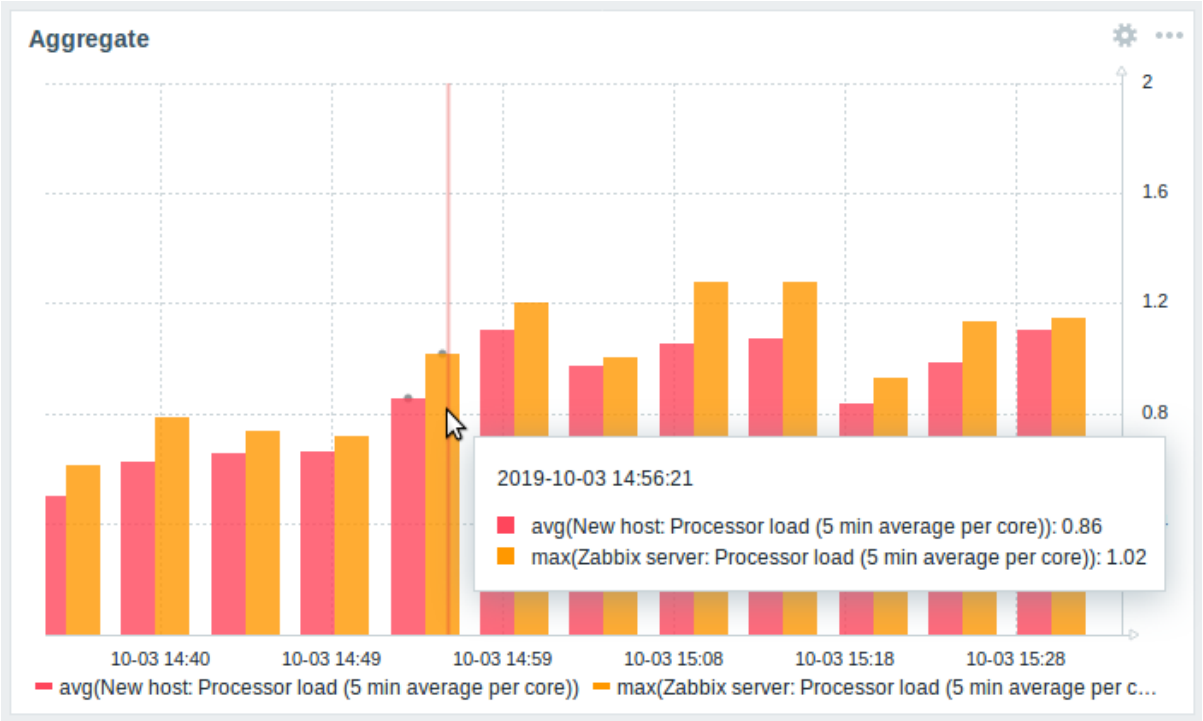
As funções de agregação, disponíveis nos widgets de gráfico e gráfico de pizza do dashboard, permitem exibir um valor agregado para o intervalo escolhido (5 minutos, uma hora, um dia), em vez de todos os valores.

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)

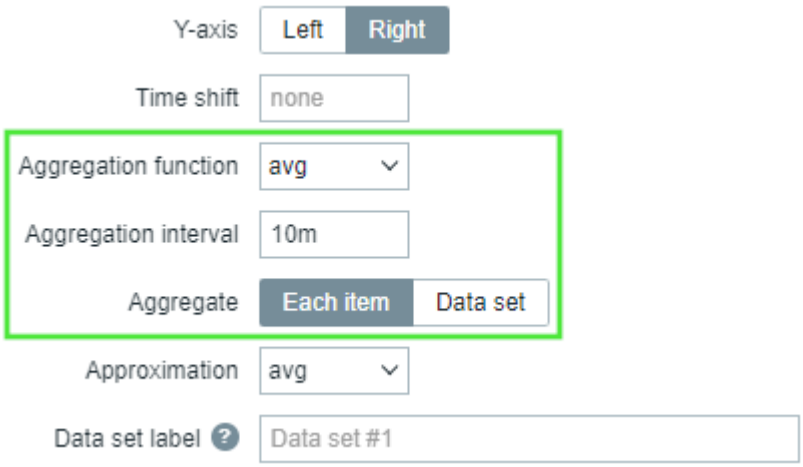
O uso mais interessante da agregação de dados é a possibilidade de criar belas comparações lado a lado de dados para algum período:



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, precedidos pela função de agregação utilizada. Se o widget de gráfico tiver um *Rótulo do conjunto de dados* configurado, o rótulo é exibido entre parênteses, precedido 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`
- um gráfico de barras é exibido, onde cada barra representa o número médio de solicitações por segundo por dia

Espaço mínimo semanal em disco entre clusters

Veja o menor espaço em disco entre clusters durante uma semana.

- adicione ao conjunto de dados: hosts `cluster*`, chave "Free disk space on /data"
- selecione a função agregada `min` e especifique o intervalo `1w`

- um gráfico de barras é exibido, onde cada barra representa o espaço mínimo em disco 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 acessíveis 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 possuem 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 permissão de leitura e escrita para este mapa e pelo menos permissão 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 quem o mapa é compartilhado pelo proprietário. Usuários regulares (não Super administradores) só podem compartilhar com os grupos e usuários dos quais fazem parte. Usuários com nível de administrador podem ver mapas privados independentemente de serem o proprietário ou de pertencerem à lista de usuários compartilhados. Mapas privados podem ser editados pelo proprietário do mapa e caso um usuário/grupo de usuários tenha permissão de leitura e escrita para este mapa e pelo menos permissão 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.

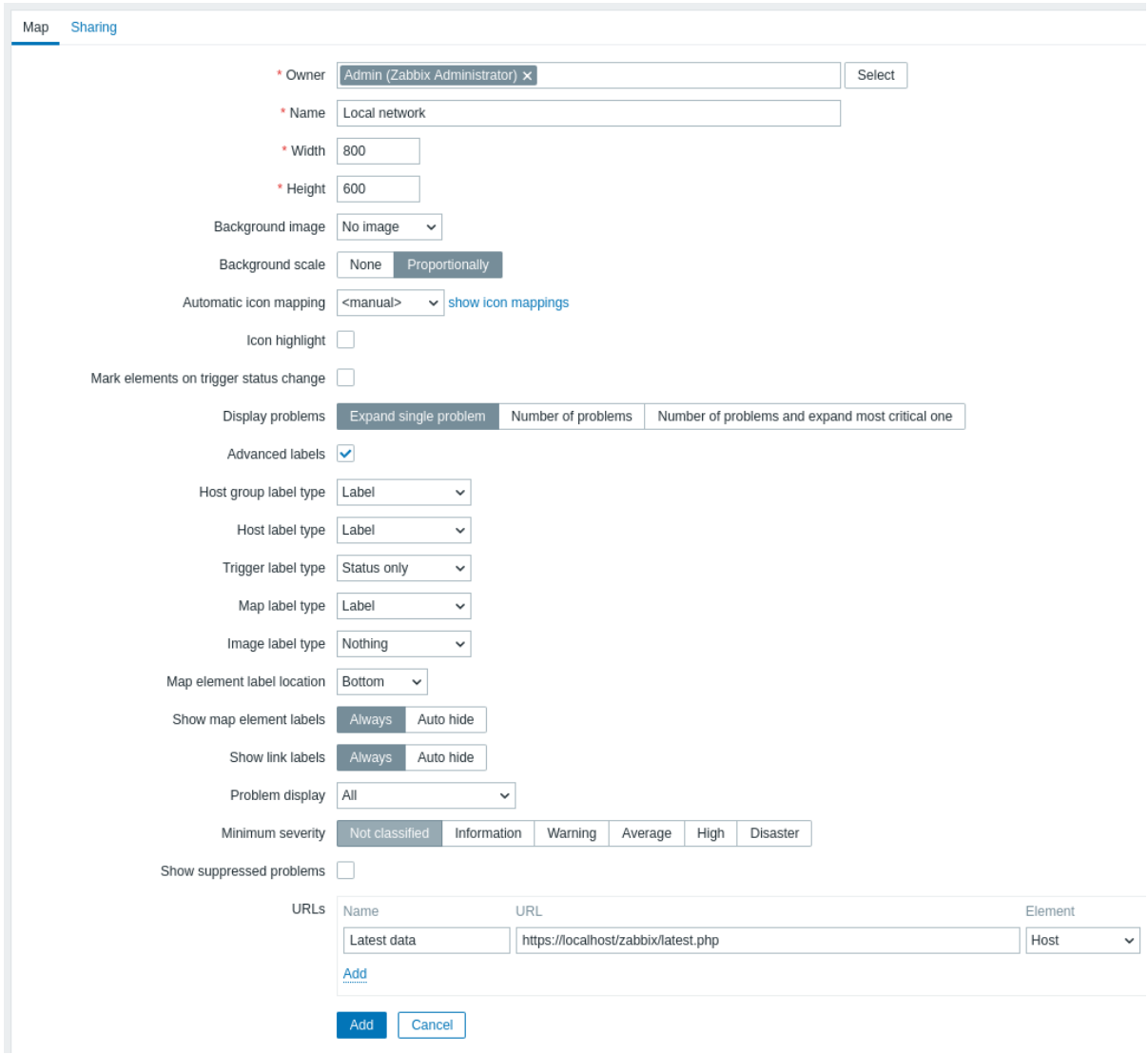
Criando um mapa

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.

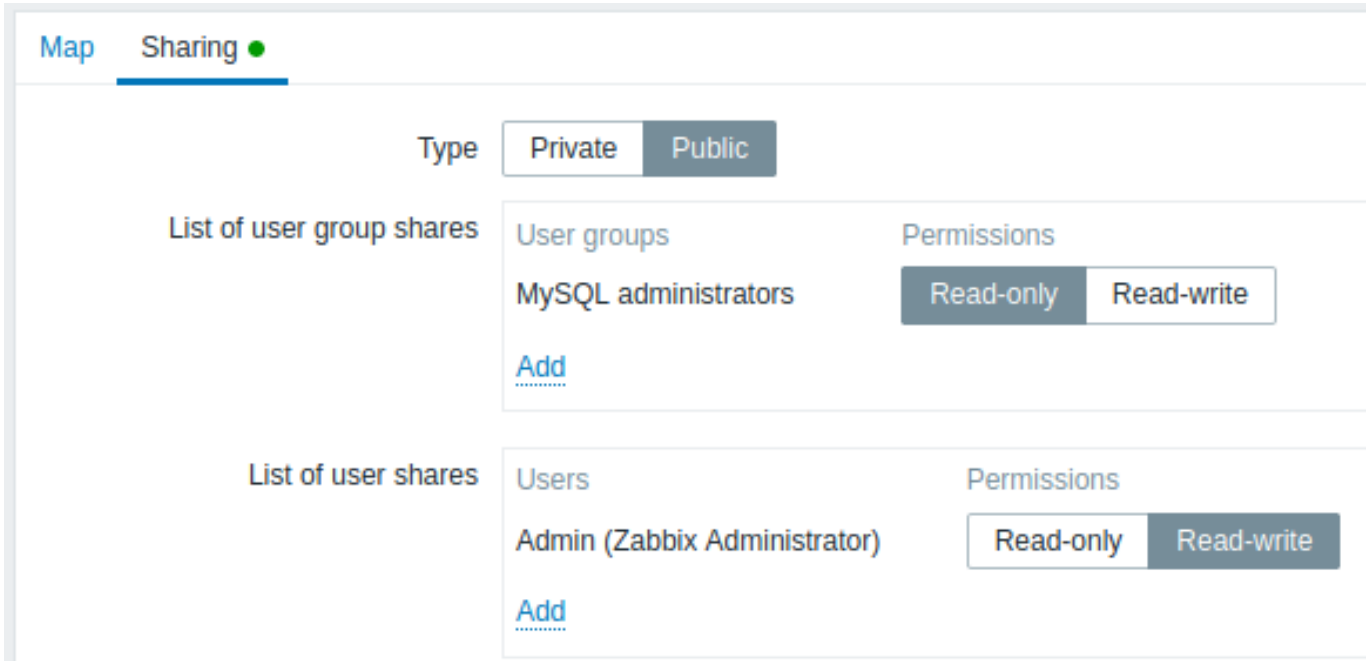
Atributos gerais do mapa:

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. Você pode usar um mapa geográfico ou qualquer outra imagem para aprimorar seu mapa.
Escala de fundo	Usar escala de fundo: **Nenhuma** - sem escala; **Proporcionalmente** - cobre todo o fundo do mapa, mas mantém as proporções da imagem (padrão).
Mapeamento automático de ícones	Você pode definir para usar 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.

Parâmetro	Descrição
Destaque de ícone	Se você marcar esta caixa, os elementos do mapa serão destacados. Elementos com um trigger ativo receberão um fundo redondo, 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 alteração 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 estão 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 gravidade 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 gravidade do problema e, se igual, na posição do trigger na lista de triggers. Em caso de múltiplos 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
Exibir rótulos dos elementos do mapa	Selecione como exibir os rótulos dos elementos do mapa: **Sempre** - sempre exibido (padrão); **Ocultar automaticamente** - oculta o rótulo se não estiver selecionado ou com o mouse sobre ele.
Exibir rótulos dos links	Selecione como exibir os rótulos dos links: **Sempre** - sempre exibido (padrão); **Ocultar automaticamente** - oculta o rótulo se não estiver selecionado ou com o mouse sobre ele.
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
Gravidade mínima do trigger	Problemas abaixo do nível mínimo de gravidade 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.
Exibir 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 URL de mapa’.

Compartilhamento

A aba **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.

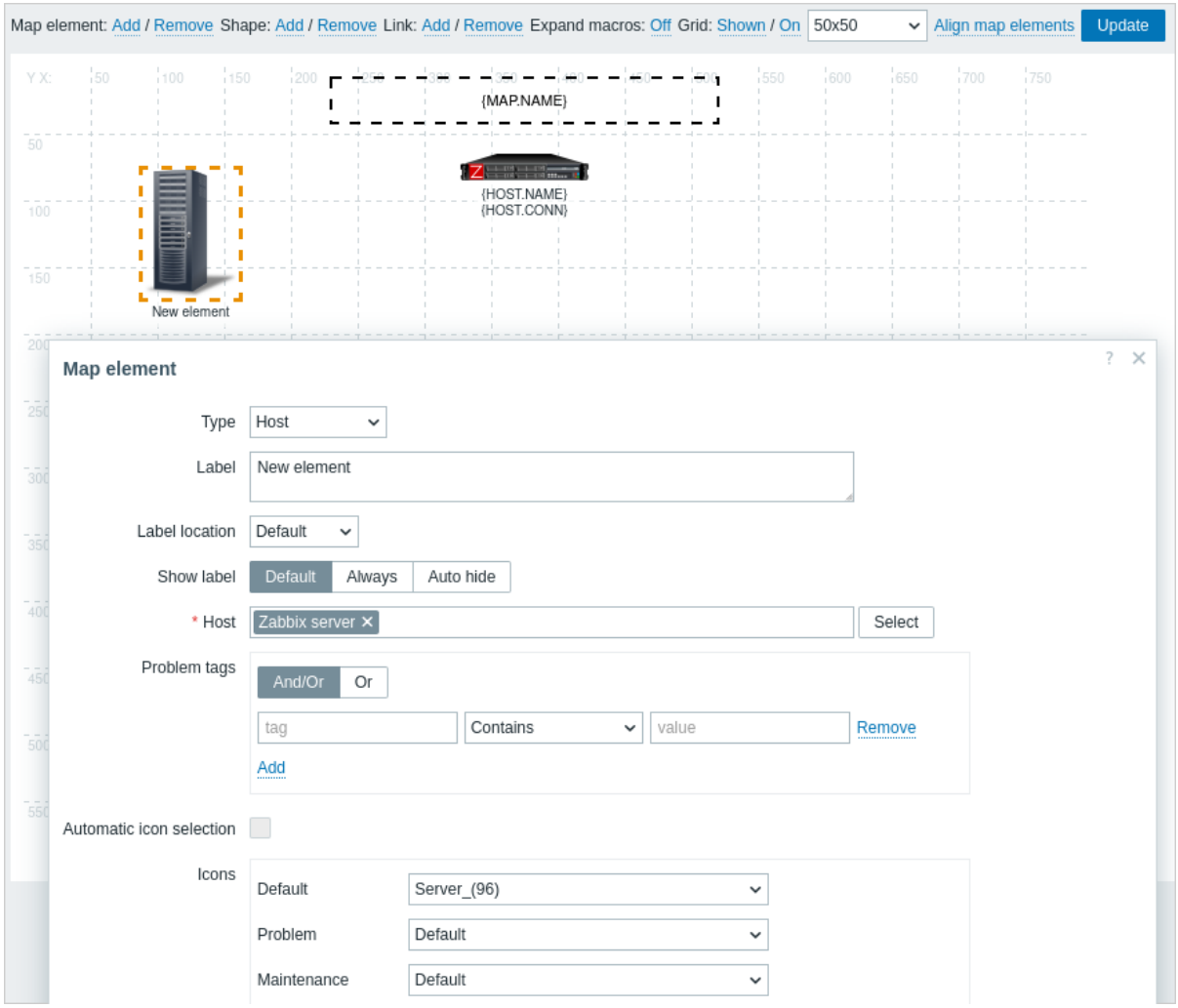
Ao clicar em *Adicionar* para salvar este mapa, você criou um mapa vazio com um nome, dimensões e certas preferências. Agora você precisa adicionar alguns elementos. Para isso, clique em *Editar* na lista de mapas para abrir a área editável.

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 onde desejar.

Observe que, com a opção Grade ativada, 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 os elementos em qualquer lugar sem alinhamento, desative a opção. (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 do 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 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'.
Exibir rótulo	Selecione como exibir o rótulo do elemento: **Padrão** - usa as configurações *Exibir rótulos de elementos do mapa* da configuração do mapa (padrão); **Sempre** - sempre exibido (padrão); **Ocultar automaticamente** - oculta o rótulo se não estiver selecionado ou com o mouse sobre ele.
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

Parâmetro	Descrição
Host	Informe o host se o tipo do 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 do 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.
Triggers	Se o tipo do 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 dos triggers. A seleção de múltiplos triggers também afeta a resolução das macros {HOST.*} tanto nos modos de edição quanto de visualização. // 1 No modo de edição// as primeiras macros {HOST.*} exibidas serão resolvidas de acordo com o primeiro trigger da lista (com base na severidade do trigger). // 2 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 de acordo com o último trigger de problema detectado (independentemente da severidade) ou o primeiro trigger da 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 de acordo com a severidade do trigger.
Grupo de hosts	Informe o grupo de hosts se o tipo do 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 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 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 definidas para seu tipo, 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 poderão ser perdidas.
Portanto, é uma boa ideia clicar no botão **Atualizar** no canto superior direito. Depois de clicar, 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 pressionada a tecla *Ctrl* para selecionar os outros.

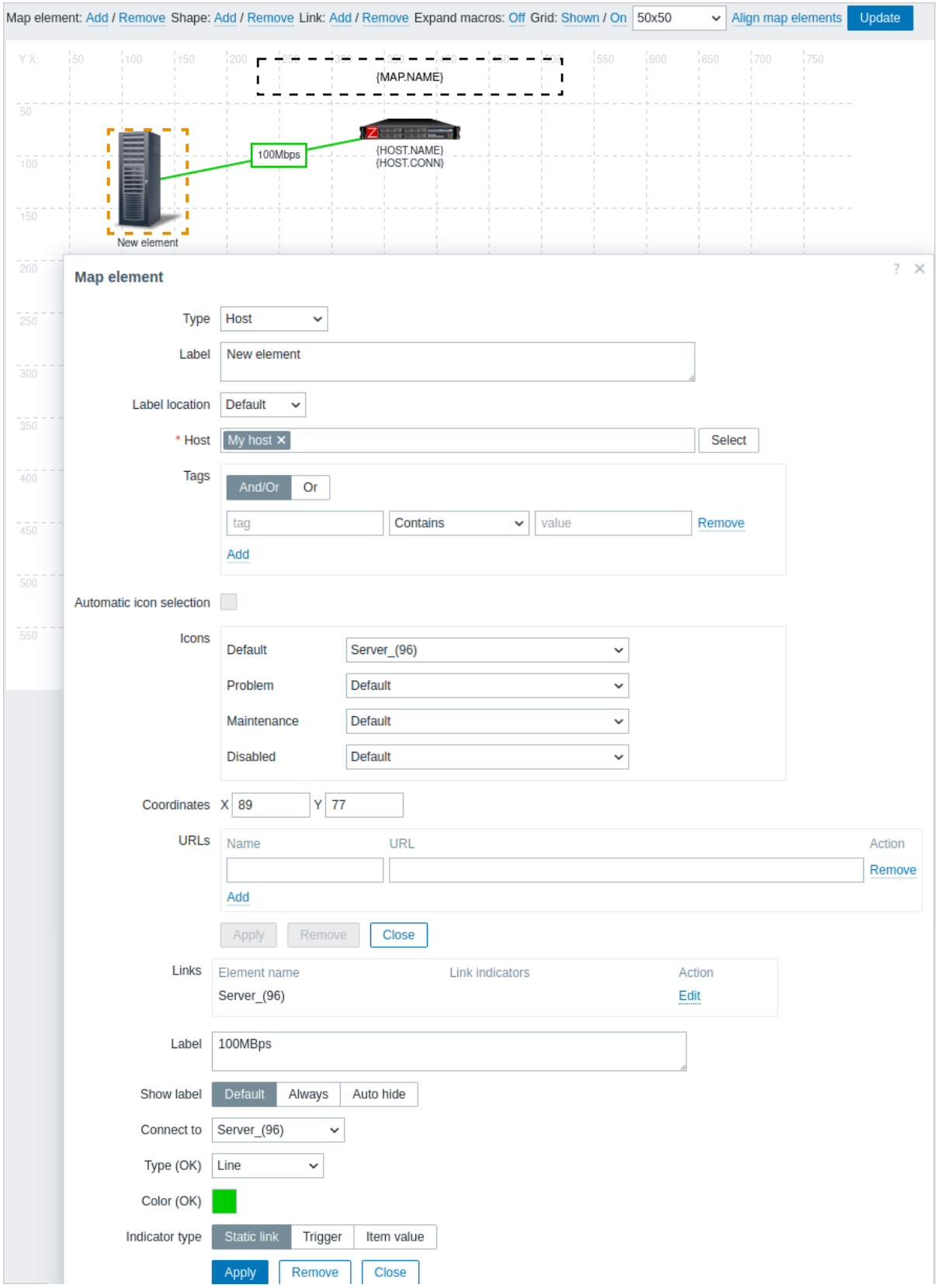
Você também pode selecionar vários elementos arrastando um retângulo na área editável e selecionando todos os elementos dentro dele.

Assim que você selecionar mais de um elemento, o formulário de propriedades do elemento muda para o modo de atualização em massa, para que você possa alterar os atributos dos elementos selecionados de uma só vez. Para 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:

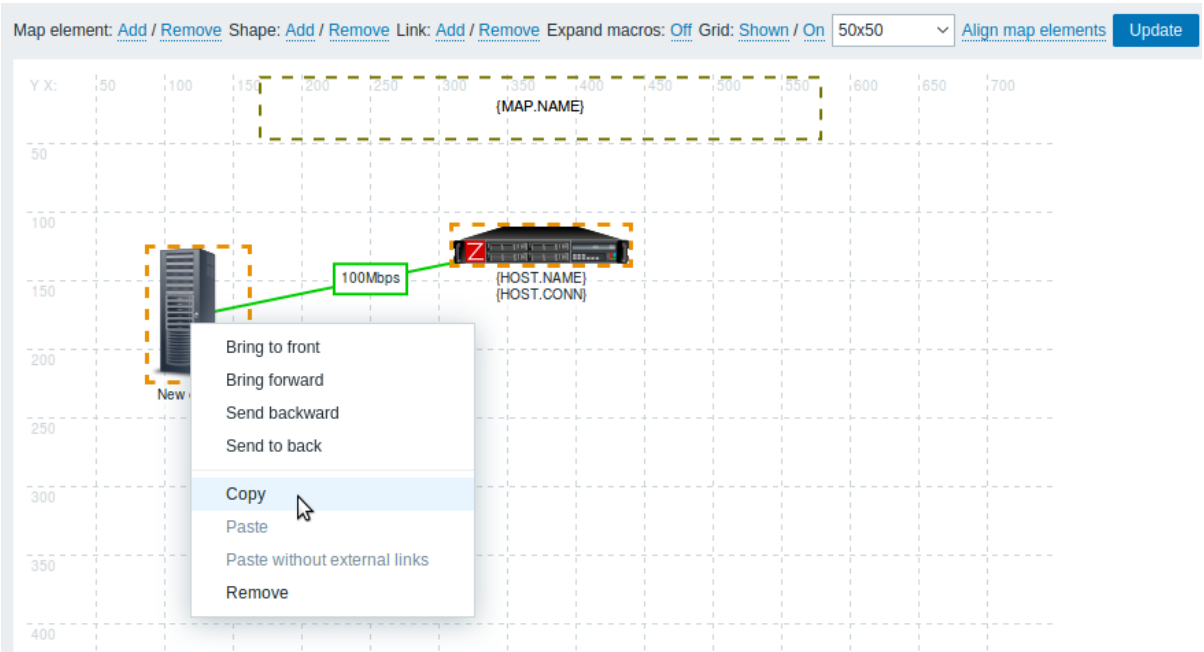
Parâmetro	Descrição
Rótulo	Rótulo que será exibido sobre o link. Expressões macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com o tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`).

Parâmetro	Descrição
Mostrar rótulo	Selecione como exibir o rótulo do link: **Padrão** - usa as configurações *Mostrar rótulos dos elementos do mapa* da configuração do mapa (padrão); **Sempre** - sempre exibido (padrão); **Ocultar automaticamente** - oculta o rótulo se não estiver selecionado ou com o mouse sobre ele.
Conectar a	O elemento ao qual o link se conecta.
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.
Tipo de indicador	Selecione o tipo de indicador do link: **Link estático** - sem indicadores; **Trigger** - permite indicadores baseados em trigger; **Valor do item** - permite indicadores baseados no valor do item.
Item	Selecione o item. O valor deste item afetará o estilo do link. Este campo só está disponível se "Valor do item" for selecionado no campo *Tipo de indicador*.
Indicadores	Lista de indicadores do link. **Triggers** Se "Trigger" for selecionado no campo *Tipo de indicador*, clique em *Adicionar* para adicionar triggers. Selecione o estilo do trigger (cor, tipo de linha). Caso um trigger tenha status PROBLEM, seu estilo será aplicado ao link. **Limiares/padrões de valor do item** Se "Valor do item" for selecionado no campo *Tipo de indicador*, adicione limiares de valor do item (para um item numérico) ou padrões (para um item de texto). Selecione o estilo do limiar/padrão (cor, tipo de linha). Se o valor do item atingir o limiar especificado (ou corresponder ao padrão), seu estilo será aplicado ao link.

Movendo e copiando/colando elementos

Vários elementos selecionados podem ser **movidos** para outro local no mapa clicando em um dos elementos selecionados, mantendo o botão do mouse pressionado e movendo o cursor para o local desejado.

Um ou mais elementos podem ser **copiados** selecionando os elementos, clicando em um elemento selecionado com o botão direito do mouse e selecionando *Copiar* no menu.



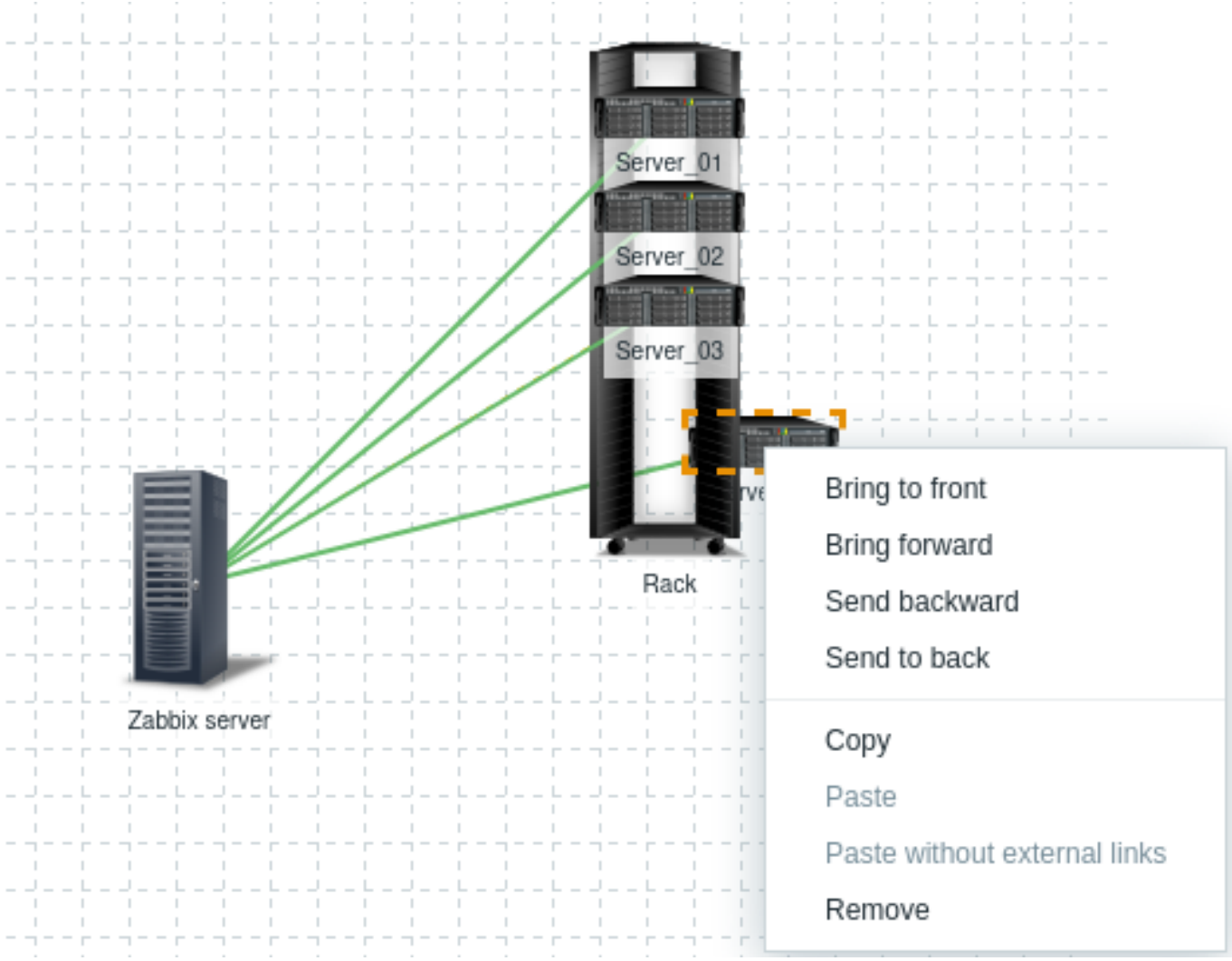
Para colar os elementos, clique em uma área do mapa com o botão direito do mouse e selecione *Colar* no menu. A opção *Colar*

sem links externos irá colar os elementos mantendo apenas os links que estão entre os elementos selecionados.

Copiar/colar funciona dentro da mesma janela do navegador. Atalhos de teclado não são suportados.

Ordenando elementos

Para trazer um elemento para frente de outro (ou vice-versa), clique no elemento com o botão direito do mouse e selecione *Trazer para frente/Trazer para o topo* ou *Enviar para trás/Enviar para o fundo*.

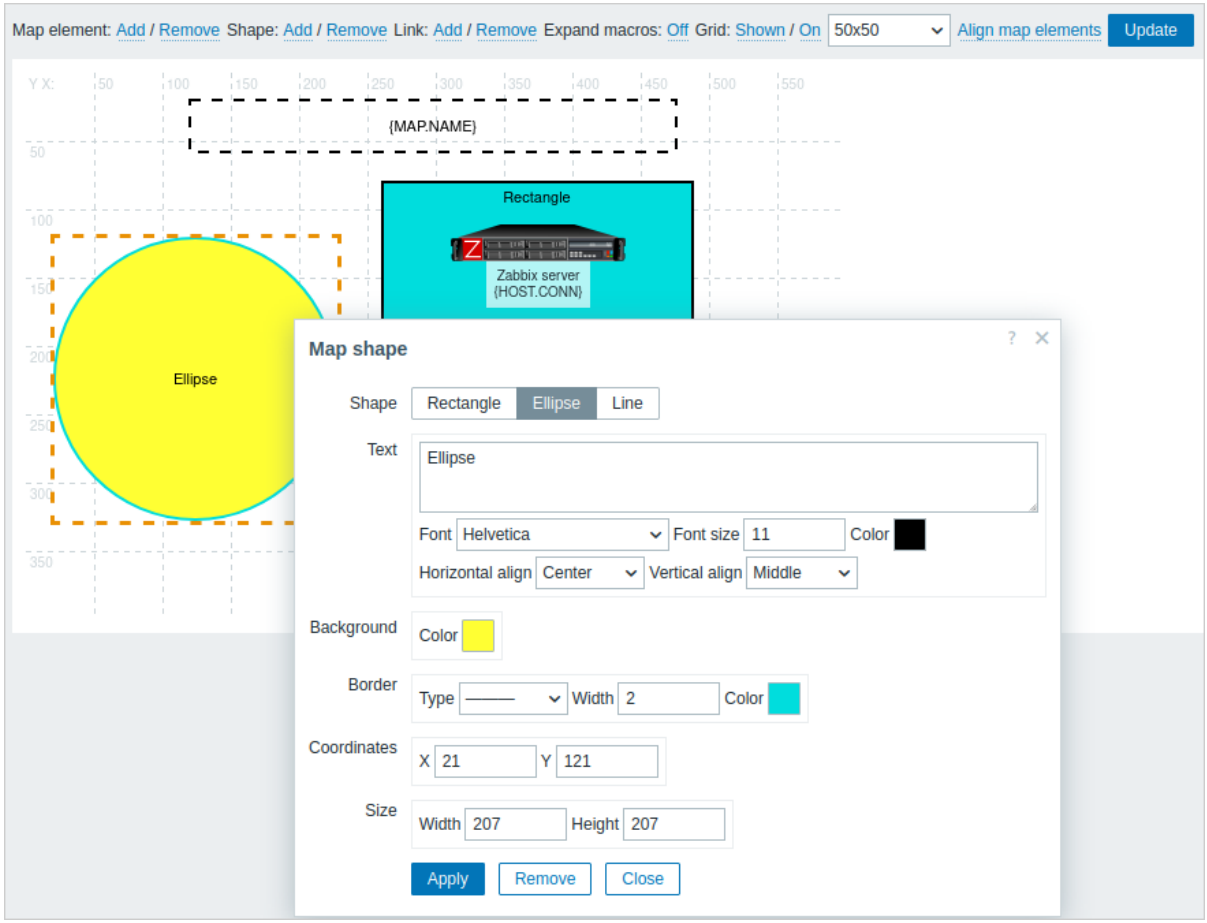


Adicionando formas

Além dos elementos do mapa, também é possível adicionar algumas formas. As formas não são elementos do mapa; elas 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 desejar.

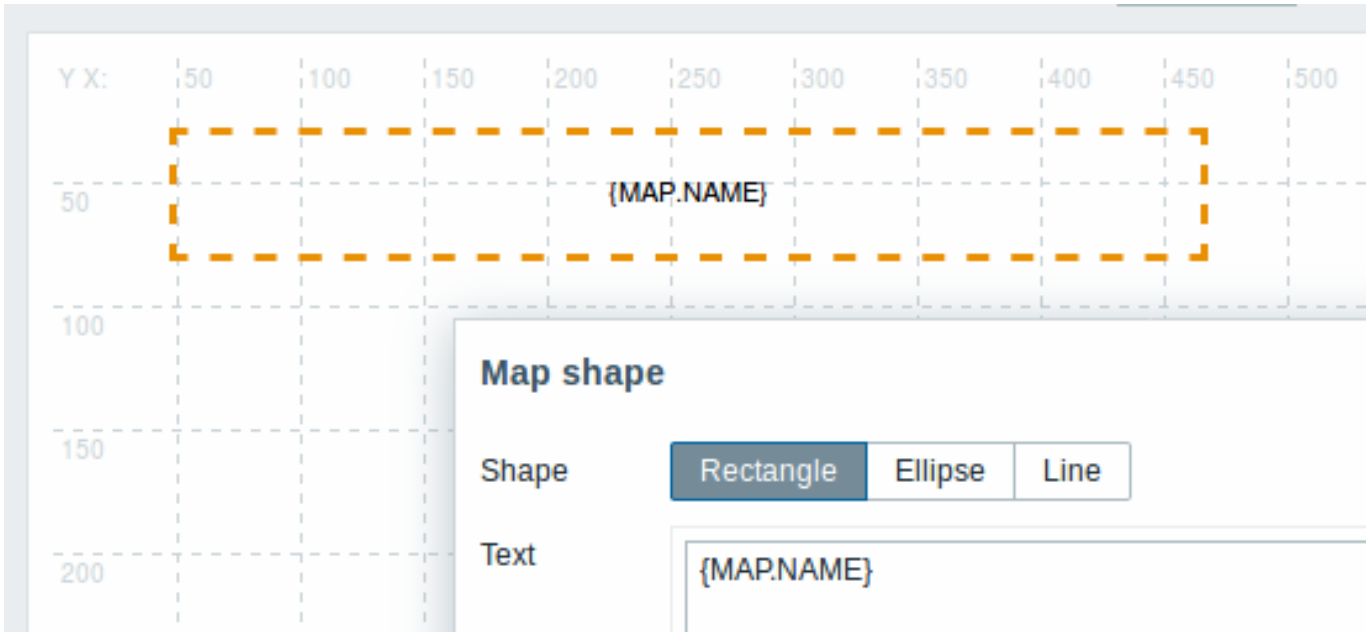
Uma nova forma é adicionada com cores padrão. Ao clicar na forma, um formulário é exibido e você pode personalizar a aparência da forma, adicionar texto, etc.



Para selecionar formas, selecione uma e depois mantenha pressionada a tecla *Ctrl* para selecionar as outras. Com várias formas selecionadas, as propriedades comuns podem ser atualizadas em massa, da mesma forma que com os elementos.

O texto pode ser adicionado nas formas. Expressões macros são suportadas no texto, mas apenas com as funções `avg`, `last`, `min` e `max`, com o 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 'Nenhum' 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 (outras páginas com mapas).

Adicionando linhas

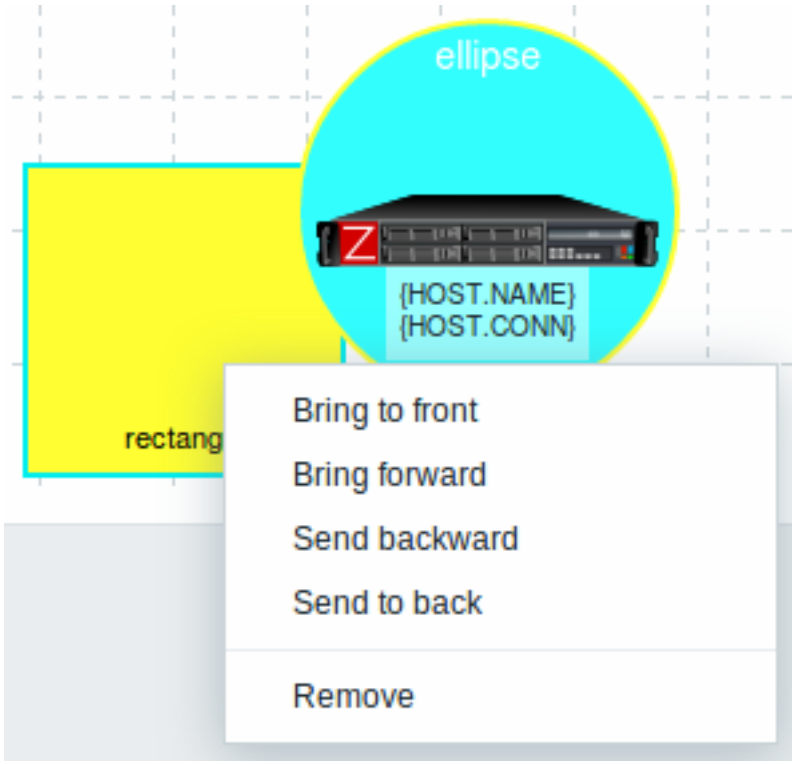
Além das formas, também é possível adicionar algumas linhas. As linhas podem ser usadas para vincular elementos ou formas em um mapa.

Para adicionar uma linha, clique em *Adicionar* ao lado de Forma. Uma nova forma aparecerá no canto superior esquerdo do mapa. Selecione-a e clique em *Linha* no formulário de edição para alterar a forma para uma linha. Em seguida, ajuste as propriedades da linha, como tipo de linha, largura, cor, etc.

map_line.png

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.

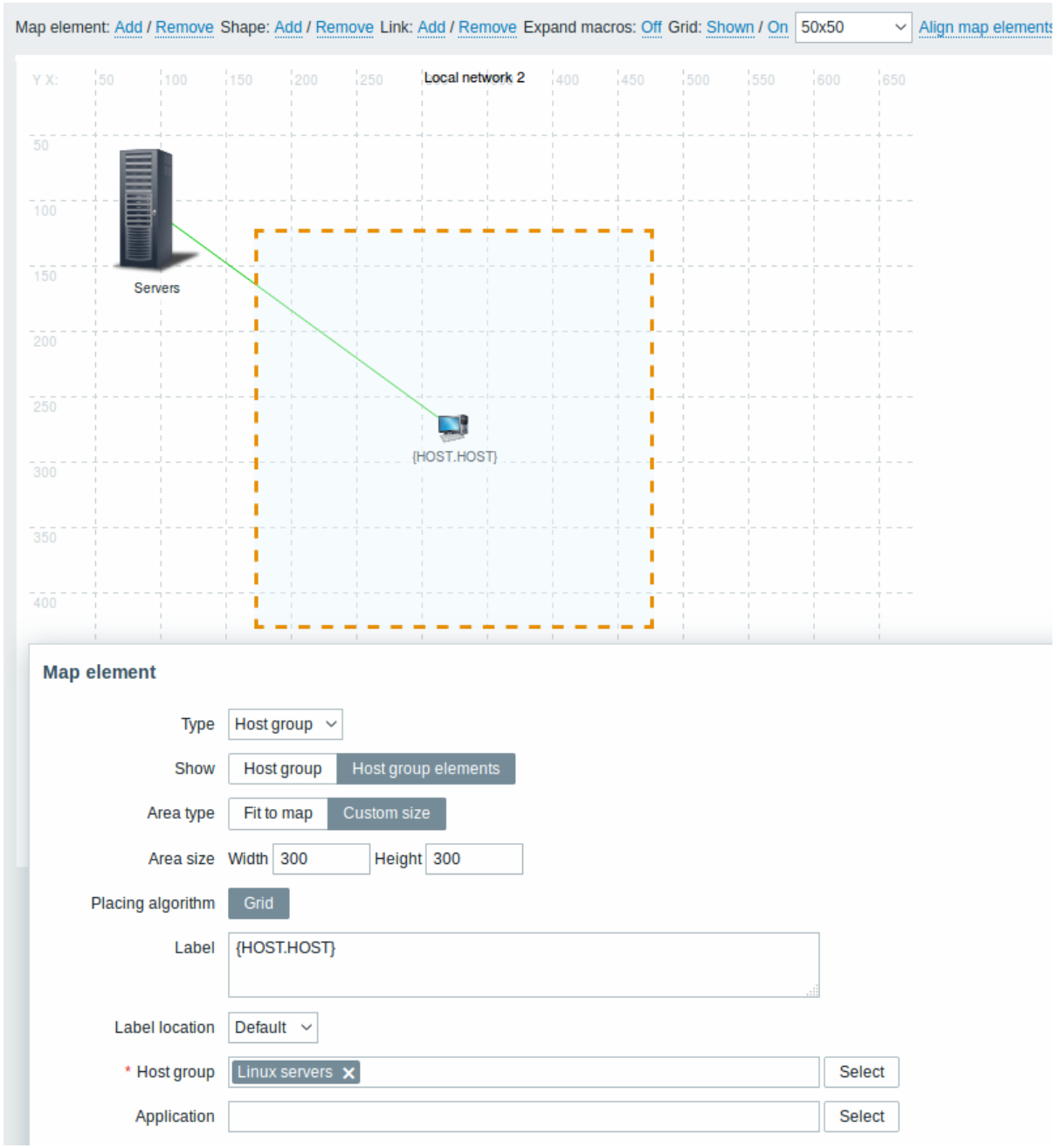


2 Elementos de grupo de hosts

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.

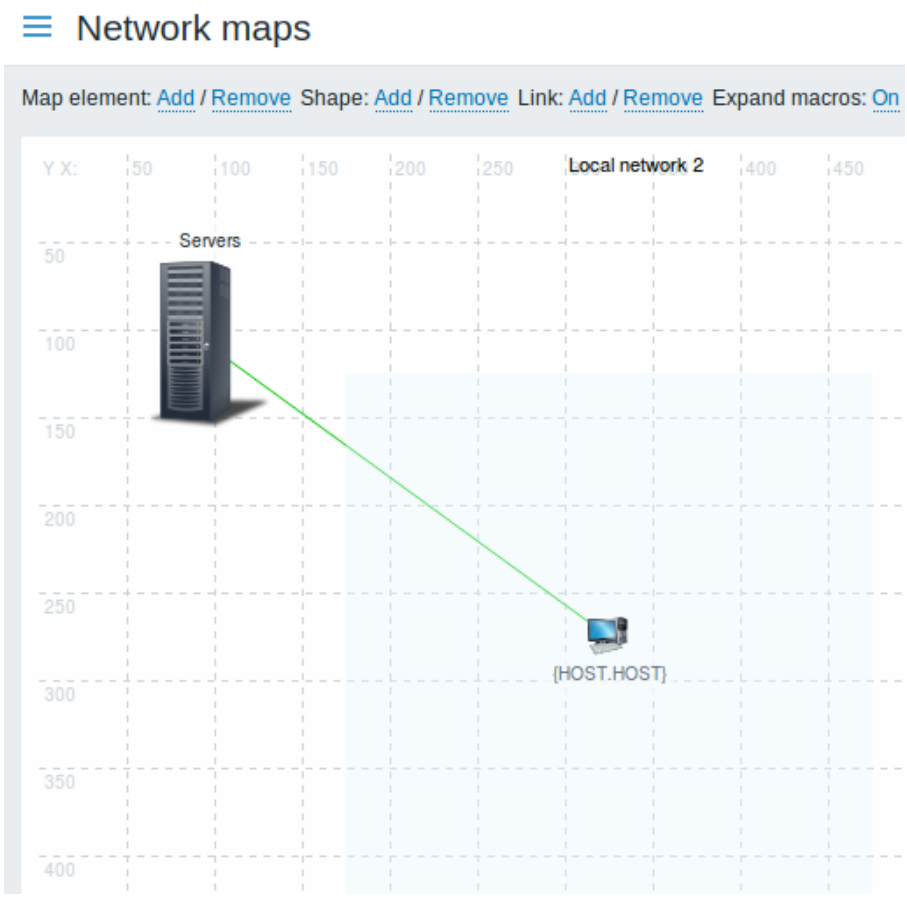
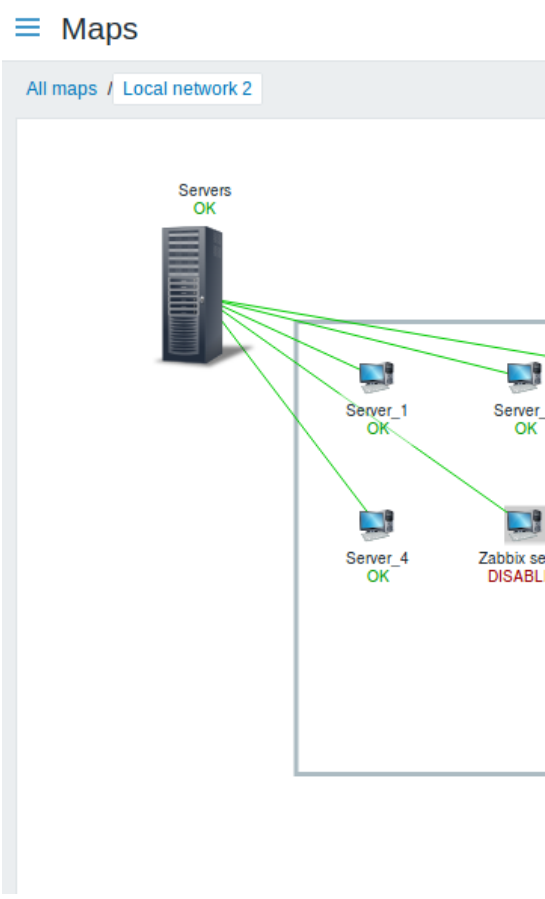
Esta tabela consiste em parâmetros típicos para o tipo de elemento *Grupo de hosts*:

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** - ao selecionar esta opção, será exibido um único ícone mostrando as informações correspondentes sobre o grupo de hosts **Elementos do grupo de hosts** - ao selecionar esta opção, serão exibidos vários ícones mostrando informações correspondentes sobre cada elemento (host) do grupo de hosts
Tipo de área	Esta configuração está disponível se o parâmetro “Elementos do grupo de hosts” for selecionado: **Ajustar ao mapa** - todos os elementos do grupo de hosts são igualmente posicionados dentro do mapa **Tamanho personalizado** - configuração manual da área do mapa para exibir todos os elementos do grupo de hosts

Parâmetro	Descrição
Tamanho da área	Esta configuração está disponível se o parâmetro “Elementos do grupo de hosts” e o parâmetro “Tipo de área” forem selecionados: **Largura** - valor numérico a ser inserido para especificar a largura da área do mapa **Altura** - valor numérico a ser inserido 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 nos 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. 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 do grupo de hosts

Esta opção está disponível se a opção ”Elementos do grupo de hosts” for escolhida para exibição. Ao selecionar ”Elementos do grupo de hosts” como opção de *exibição*, 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 link

Visão geral

Você pode atribuir indicadores a um link entre elementos em um mapa de rede.

Os indicadores podem ser baseados em triggers ou valores de item. É possível exibir diferentes estilos e cores de link:

- quando triggers entram em estado de problema;
- quando o valor do item:
  - atinge um limite (para itens numéricos);
  - corresponde a uma expressão regular (para itens de texto).

Ao configurar um link, você define o tipo e a cor padrão do link. Ao atribuir indicadores a um link, torna-se possível fazer com que o estilo e a cor do link dependam do status do trigger ou do valor do item.

Por exemplo, se algum dos triggers atribuídos entrar em estado de problema, o estilo e a cor do link mudarão para refletir isso. Então, talvez seu link padrão fosse uma linha verde. Agora, com o trigger em estado de problema, seu link pode se tornar vermelho em negrito (se você definiu assim).

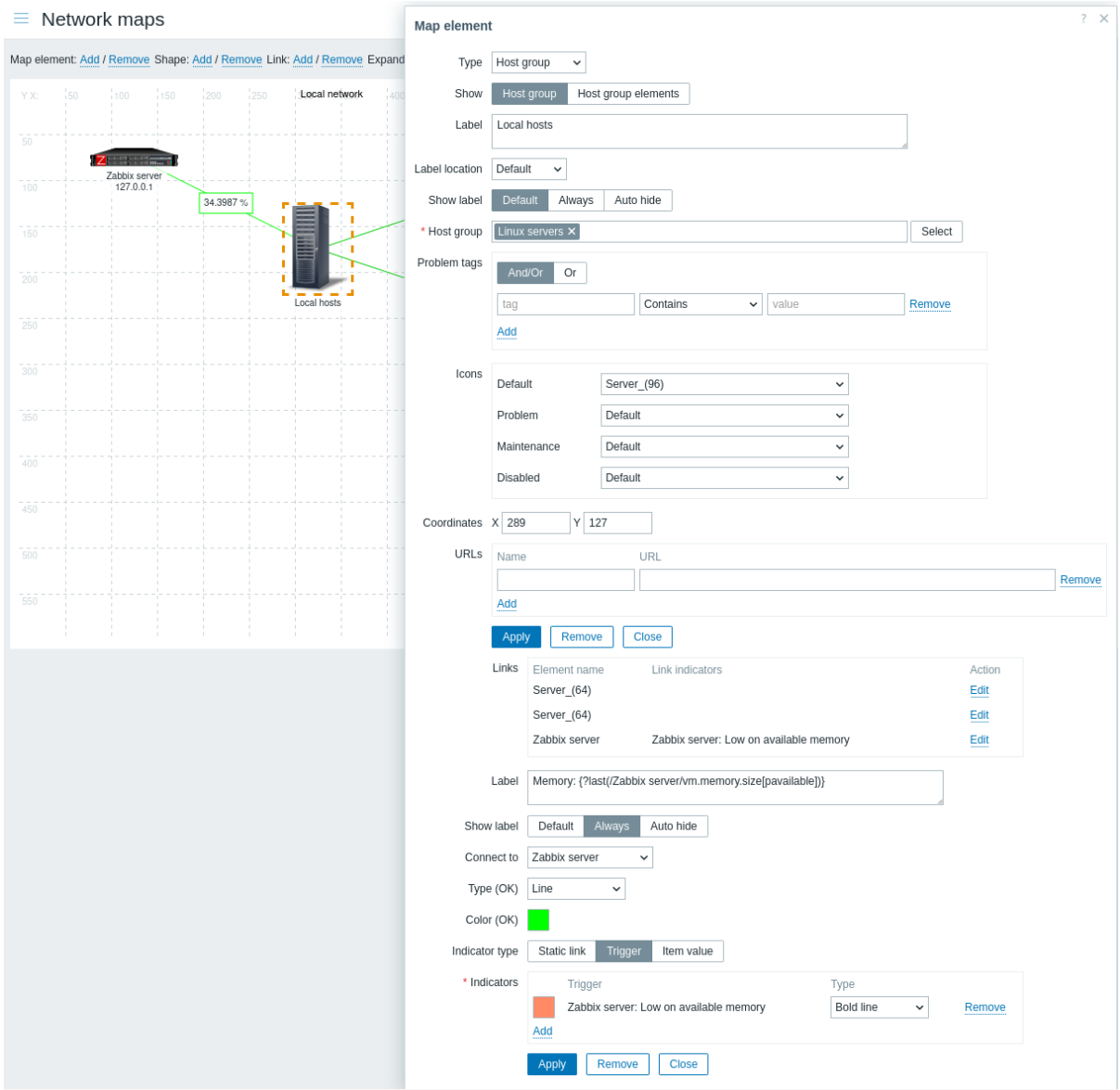
Da mesma forma, se um valor de item atingir um limite especificado ou corresponder a uma expressão regular especificada, o estilo do link pode refletir isso.

Configuração

Triggers

Para atribuir triggers como indicadores de link, faça o seguinte:

- selecione um elemento do mapa
- clique em *Editar* na seção *Links* para o link apropriado
- selecione *Trigger* como o tipo de indicador
- clique em *Adicionar* no bloco *Indicadores* e selecione um ou mais triggers

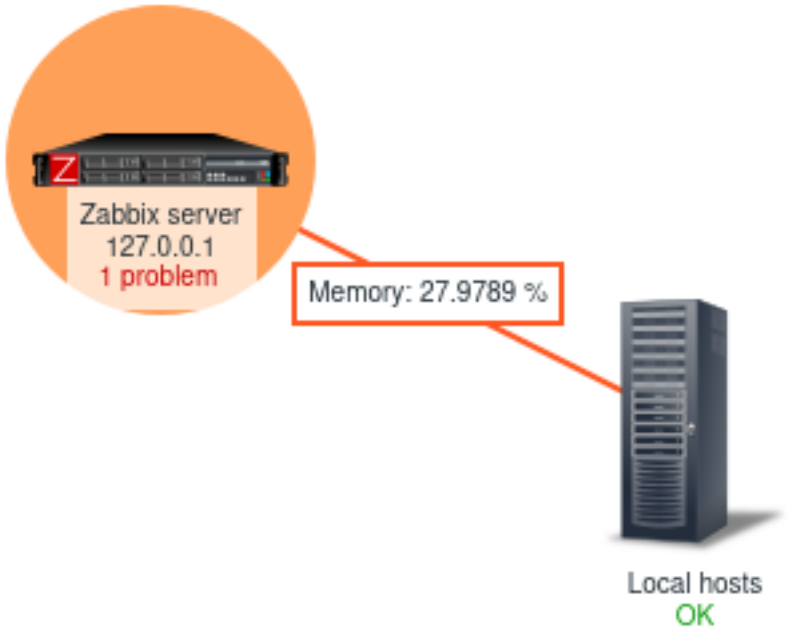


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os triggers adicionados podem ser vistos na lista *Indicadores*.

Você pode definir o tipo e a cor do link para cada indicador diretamente na lista. Quando terminar, clique em *Aplicar*, feche o formulário e clique em *Atualizar* para salvar as alterações do mapa.

Em *Monitoramento* → *Mapas*, o respectivo tipo e cor do link serão exibidos se o **trigger** entrar em estado de problema.



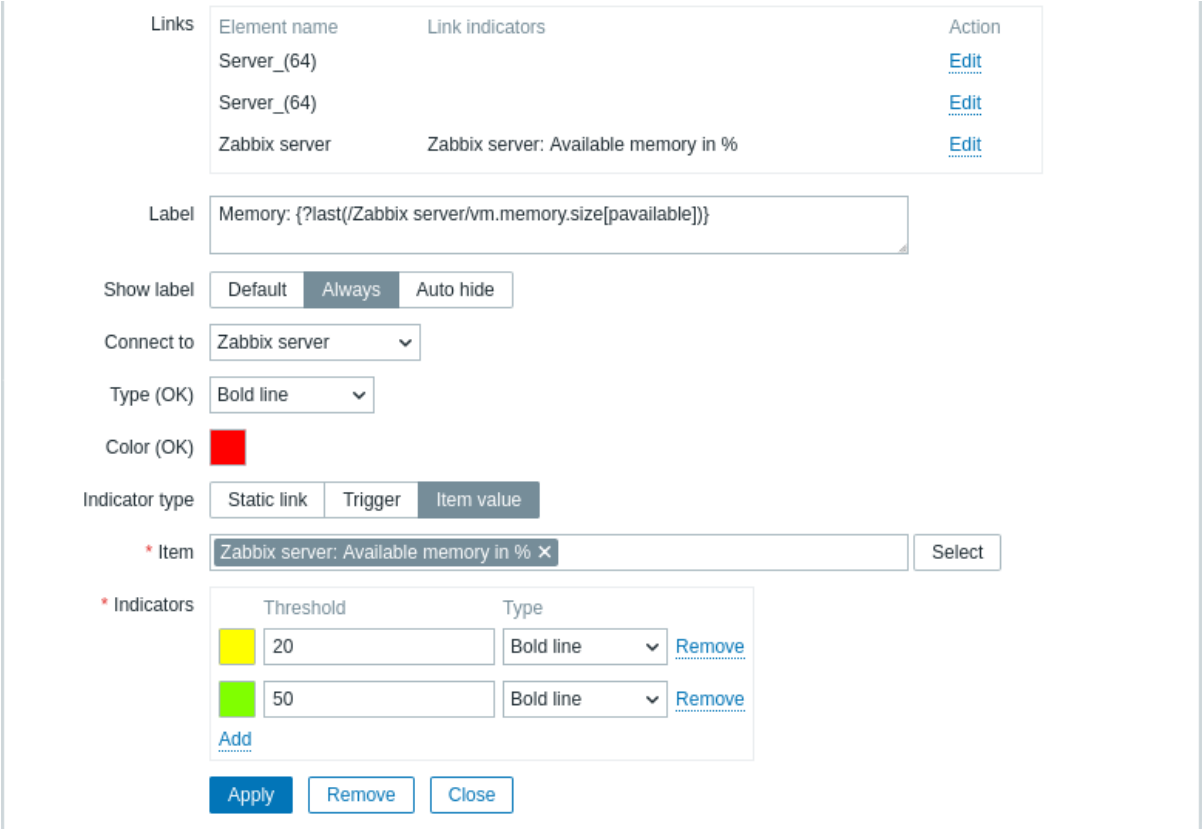
Note:
Se vários triggers entrarem em estado de problema, o problema com a maior severidade determinará o estilo e a cor do link. Se vários triggers com a mesma severidade forem atribuídos ao mesmo link do mapa, aquele com o menor ID terá precedência. Observe também que:

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.

Valores de item

Para atribuir valores de item como indicadores de link, faça o seguinte:

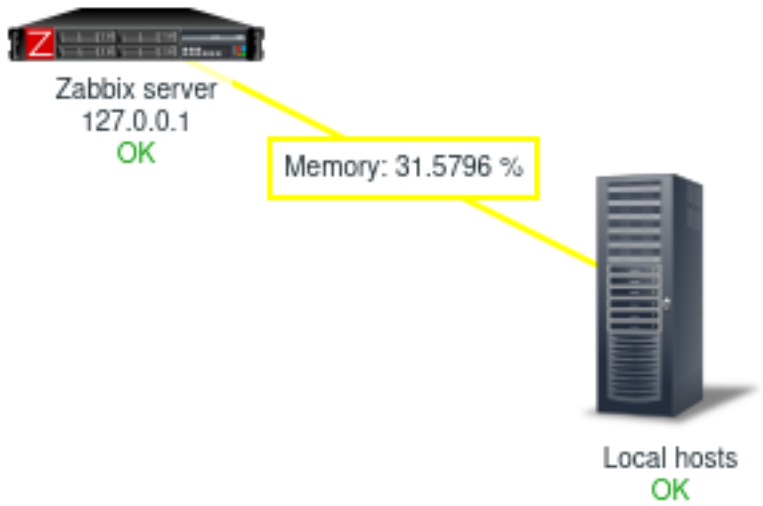
- selecione um elemento do mapa
- clique em *Editar* na seção *Links* para o link apropriado
- selecione *Valor do item* como o tipo de indicador
- selecione o item
- adicione um ou mais limites ou padrões de valor de item no bloco *Indicadores*



Os limites/padrões de item adicionados podem ser vistos na lista *Indicadores*.

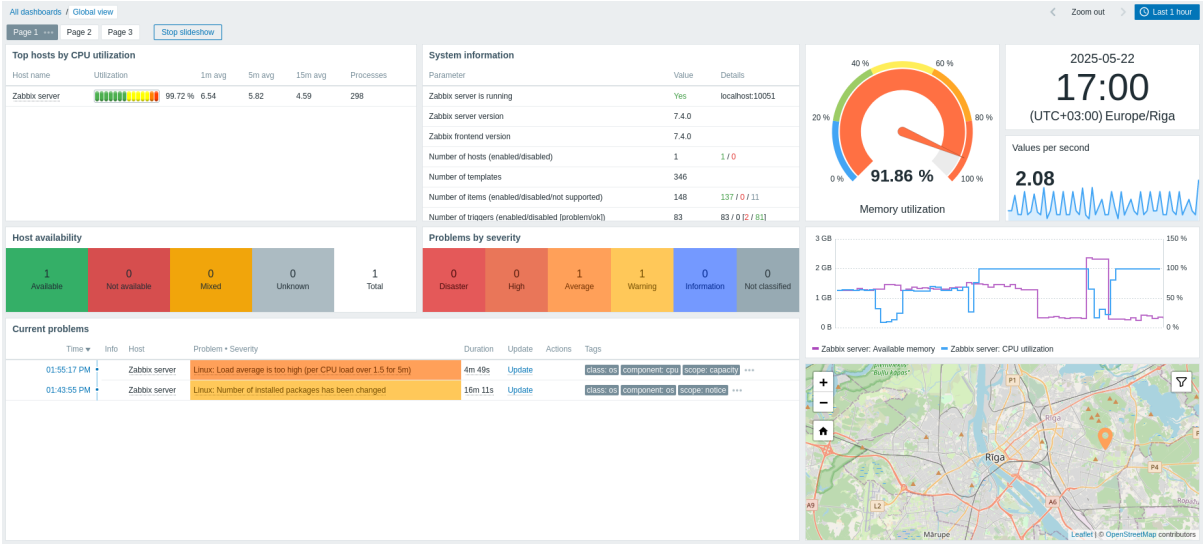
Você pode definir o tipo e a cor do link para cada indicador diretamente na lista. Quando terminar, clique em *Aplicar*, feche o formulário e clique em *Atualizar* para salvar as alterações do mapa.

Em *Monitoramento* → *Mapas*, o respectivo tipo e cor do link serão exibidos se o **valor do item** atingir o limite definido (para tipos de dados numéricos) ou corresponder ao padrão de expressão regular (para tipos de dados de texto).



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. 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 depois 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 Templates prontos para uso

Visão geral

O Zabbix fornece um conjunto crescente de templates pré-configurados para simplificar e acelerar 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.

Utilize 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, *master* para a versão de desenvolvimento mais recente).
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, vá em *Coleta de dados* > *Templates* e importe o arquivo local.

Compatibilidade com o Assistente de Host

Para atualizar vários templates que são compatíveis com o Assistente de Host, você pode importar os seguintes pacotes de templates:

- cloud_aws_azure.yaml
- cloud_gcp_oracle.yaml
- database.yaml
- network_os.yaml
- software.yaml

Para uma lista completa de templates compatíveis, consulte templates_in_bundles.md.

Para tornar um template personalizado compatível com o Assistente de Host:

1. No frontend do Zabbix, vá para *Coleta de dados* > *Templates* e exporte o template personalizado.
2. Edite o arquivo exportado:
    - Adicione o elemento `wizard_ready` template element para indicar a compatibilidade com o Assistente de Host.
    - Opcionalmente, adicione os elementos `readme` ou `config` template element conforme necessário.
3. No frontend do Zabbix, vá para *Coleta de dados* > *Templates* e importe o arquivo de template atualizado.

1 Operação de template de agent Zabbix

Etapas para garantir a operação correta de templates que coletam métricas com o agent Zabbix:

1. Certifique-se de que o agent Zabbix está instalado no host. Para verificações ativas, também certifique-se de que o endereço do servidor/proxy Zabbix 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, pode ser 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 agent Zabbix
- Apache por agent Zabbix ativo
- HAProxy por agent Zabbix
- IIS por agent Zabbix
- IIS por agent Zabbix ativo
- Microsoft Exchange Server 2016 por agent Zabbix
- Microsoft Exchange Server 2016 por agent Zabbix ativo
- MySQL por agent Zabbix
- MySQL por agent Zabbix ativo
- Nginx por agent Zabbix
- Nginx por agent Zabbix ativo
- PHP-FPM por agent Zabbix
- PHP-FPM por agent Zabbix ativo

- PostgreSQL por agent Zabbix
- PostgreSQL por agent Zabbix ativo
- RabbitMQ cluster por agent Zabbix

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 na sua instalação do Zabbix, pode ser 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. Enquanto a configuração básica pode ser feita simplesmente ajustando macros de usuário, a personalização mais profunda pode ser alcançada configurando o plugin em si. Por exemplo, se um plugin suporta 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 site por Zabbix agent 2

3 Operação de template HTTP

Etapas para garantir a operação correta de templates que coletam métricas com o agent HTTP:

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 em nível de 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

- OpenAI Platform 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
- 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
- VMware SD-WAN VeloCloud por HTTP
- YugabyteDB por HTTP
- ZooKeeper por HTTP

4 Operação de template IPMI

Templates IPMI não requerem nenhuma configuração específica. Para começar o monitoramento, 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 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:

- 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 de 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 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:

- 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.

Templates para muitas famílias de dispositivos de rede são fornecidos. Todos os templates 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 do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certificado
Aruba CX 8300s por 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	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Brocade FC SNMP	Brocade	Brocade FC switches	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Desempenho Falha
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certificado
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Desempenho Falha
Check Point Next Generation Firewall por SNMP	Check Point	Next Generation Firewall	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certificado
Ciena 3906 por SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certificado
Cisco Catalyst 3750<modelo do dispositivo> 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	Certificado
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certificado
Cisco IOS versões 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T e 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certificado
Cisco IOS anterior a 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certificado
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	Certificado
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Desempenho Falha Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certificado
FortiGate por SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Desempenho Inventário
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certificado
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	Certificado
Huawei OceanStor 5300 V5 por SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado com software V5		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificado
Huawei OceanStor Dorado por SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300/5500/5600/6800/18500/18800 com software V6/V700		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificado
Huawei VRP por SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certificado
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Dispositivos Intel/QLogic Infiniband	Infiniband 12300		ICS-CHASSIS-MIB	Falha Inventário
Juniper SNMP	Juniper	Modelos MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Juniper MX NETCONF	Juniper	Modelos MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Não aplicável (usa NETCONF sobre SSH)*	Certificado
Juniper MX SNMP	Juniper	Modelos MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certificado
Mellanox SNMP	Mellanox	Dispositivos Mellanox Infini-band	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certificado
MikroTik CCR<modelo do dispositivo> SNMP	MikroTik	MikroTik Cloud Core Routers (série CCR)	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	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
MikroTik CRS<modelo do dispositivo> SNMP	MikroTik	MikroTik Cloud Router Switches (série CRS)	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	Certificado
MikroTik CSS<modelo do dispositivo> SNMP	MikroTik	MikroTik Cloud Smart Switches (série CSS)	Templates dedicados separados estão disponíveis para MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificado
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificado
MikroTik hEX <modelo do dispositivo> 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	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
MikroTik netPower <modelo do dispositivo> 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	Certificado
MikroTik PowerBox <modelo do dispositivo> SNMP	MikroTik	MikroTik PowerBox	Templates dedicados separados estão disponíveis para MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificado
MikroTik RB<modelo do dispositivo> SNMP	MikroTik	Roteadores MikroTik série RB	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	Certificado
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	Certificado

Nome do template	Fabricante	Família do dispositivo	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Falha Inventário
QTech QSW SNMP	QTech	Dispositivos Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Desempenho Inventário
Stormshield SNS por 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	Certificado
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Desempenho Inventário
Ubiquiti AirOS SNMP	Ubiquiti	Dispositivos wireless Ubiquiti AirOS	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Desempenho
Vyatta Virtual Router por SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Desempenho Inventário

Design do template

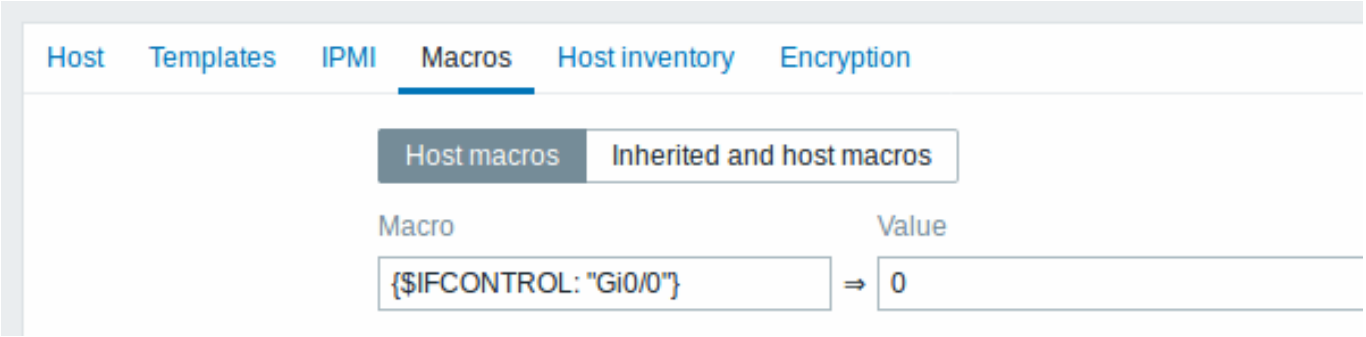
Os templates foram projetados com o seguinte em mente:

- Macros de usuário são usadas tanto quanto possível para que 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 em items e 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 quando possível. Se não for suportado, contadores padrão de 32 bits são usados em vez disso.

Todas as interfaces de rede descobertas têm um trigger que monitora seu status operacional (link), por exemplo:

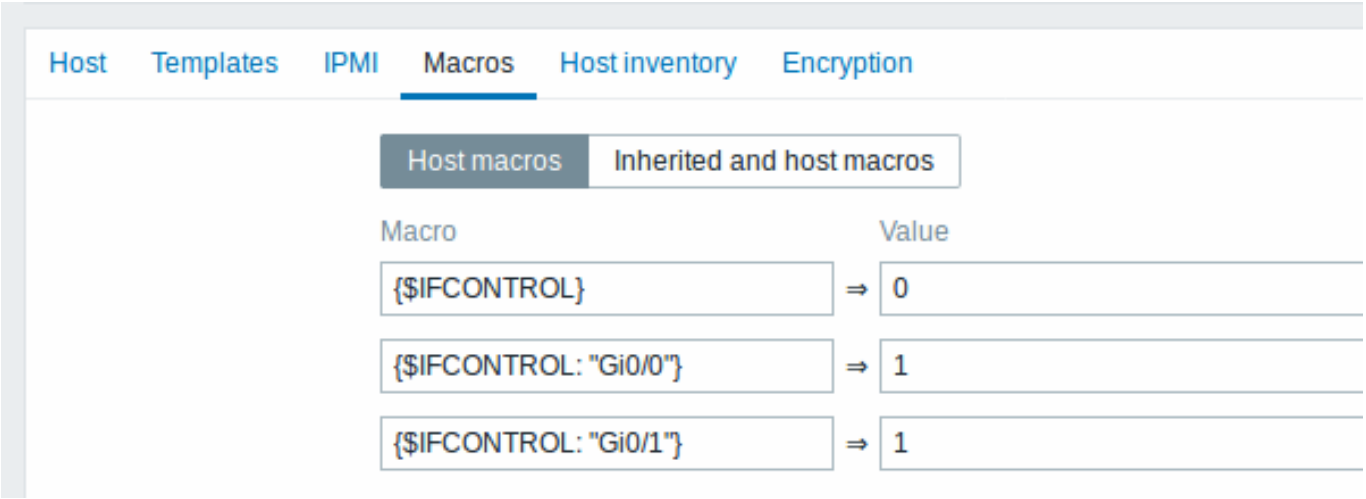
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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:



Tags

- Performance – as MIBs da família de dispositivos fornecem uma maneira de monitorar itens de CPU e memória;
- Fault - as MIBs da família de dispositivos fornecem uma maneira de monitorar pelo menos um sensor de temperatura;
- Inventory – as MIBs da família de dispositivos fornecem uma maneira de coletar pelo menos o número de série do dispositivo e o nome do modelo;
- Certified – todas as três principais categorias 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 o VMware vCenter ou hipervisores 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 estar 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 o autoregistro ativo do agent Zabbix estiver habilitado, o processo de autoregistro lerá os nomes dos hosts conforme foram registrados durante a inicialização (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 em descoberta;
- O template *VMware Guest* não deve ser vinculado manualmente a um host e só pode ser usado em 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 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 ver que todas as pessoas envolvidas foram informadas.

É por isso que o envio de notificações é uma das principais ações oferecidas pelo Zabbix. Quem e quando deve ser notificado em um 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

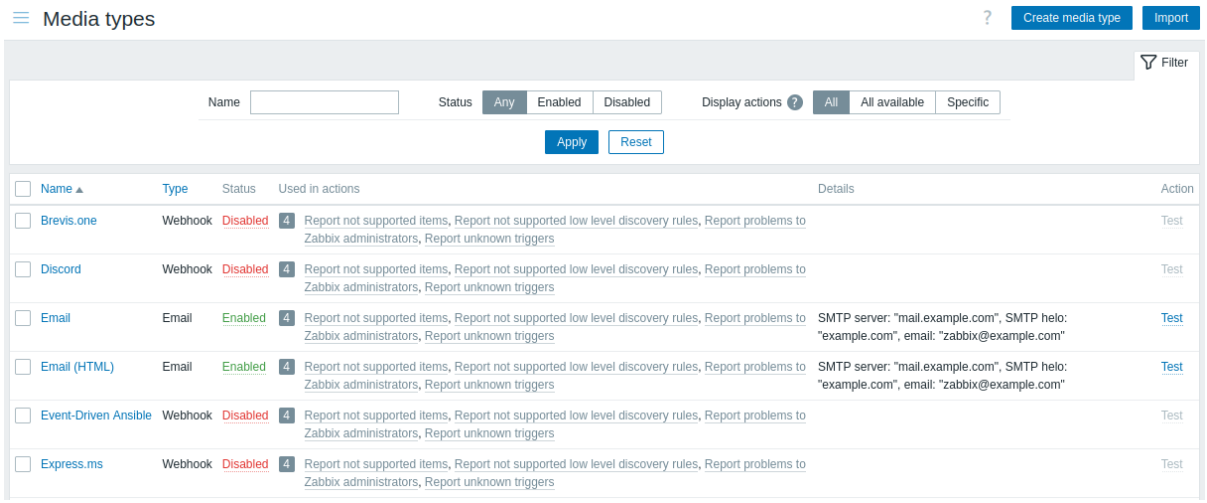
Visão geral

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. Detalhes de entrega em nível de usuário (como endereços de e-mail, números de telefone, etc.) devem ser definidos na 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 ao 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:

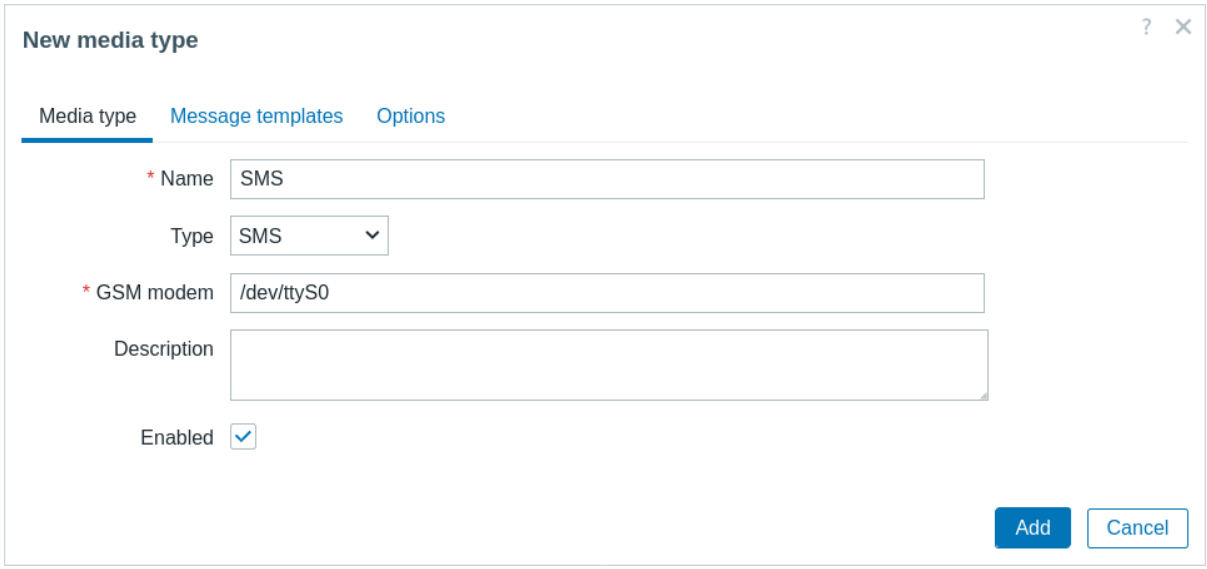
- E-mail
- Webhook
- Script

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 de 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 para *Usuários* > *Usuários* e abra o formulário de propriedades do usuário (ou vá para *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	Forneça as informações de contato necessárias para onde enviar as mensagens. Para um tipo de mídia de e-mail, é possível adicionar vários endereços clicando em Add abaixo do campo de endereço. Nesse caso, a notificação será enviada para todos os endereços de e-mail fornecidos. Também é possível especificar o nome do destinatário no campo *Enviar para* do destinatário do e-mail no formato 'Nome do destinatário <endereco1@empresa.com>'. Observe que, se um nome de destinatário for fornecido, um endereço de e-mail deve ser colocado entre sinais de menor e maior que (<>). Caracteres UTF-8 no nome são suportados, pares entre aspas e comentários não são. Por exemplo: *John Abercroft <manager@nycdatacenter.com>* e *manager@nycdatacenter.com* são ambos formatos válidos. Exemplos incorretos: *John Doe zabbix@company.com*, *%%"Zabbix\@\<H(comment)Q\>" zabbix@company.com %%*.
Quando ativo	Você pode limitar o tempo 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 poderá precisar ser ajustado para não perder mensagens importantes. Consulte a página Especificação do período de tempo para obter a 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. *Observação*: 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 de gravidade correspondentes, enquanto as não selecionadas ficarão esmaecidas.
Status	Status da mídia do usuário. **Habilitado** - está em uso. **Desabilitado** - não está sendo usado.

1 E-mail

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.

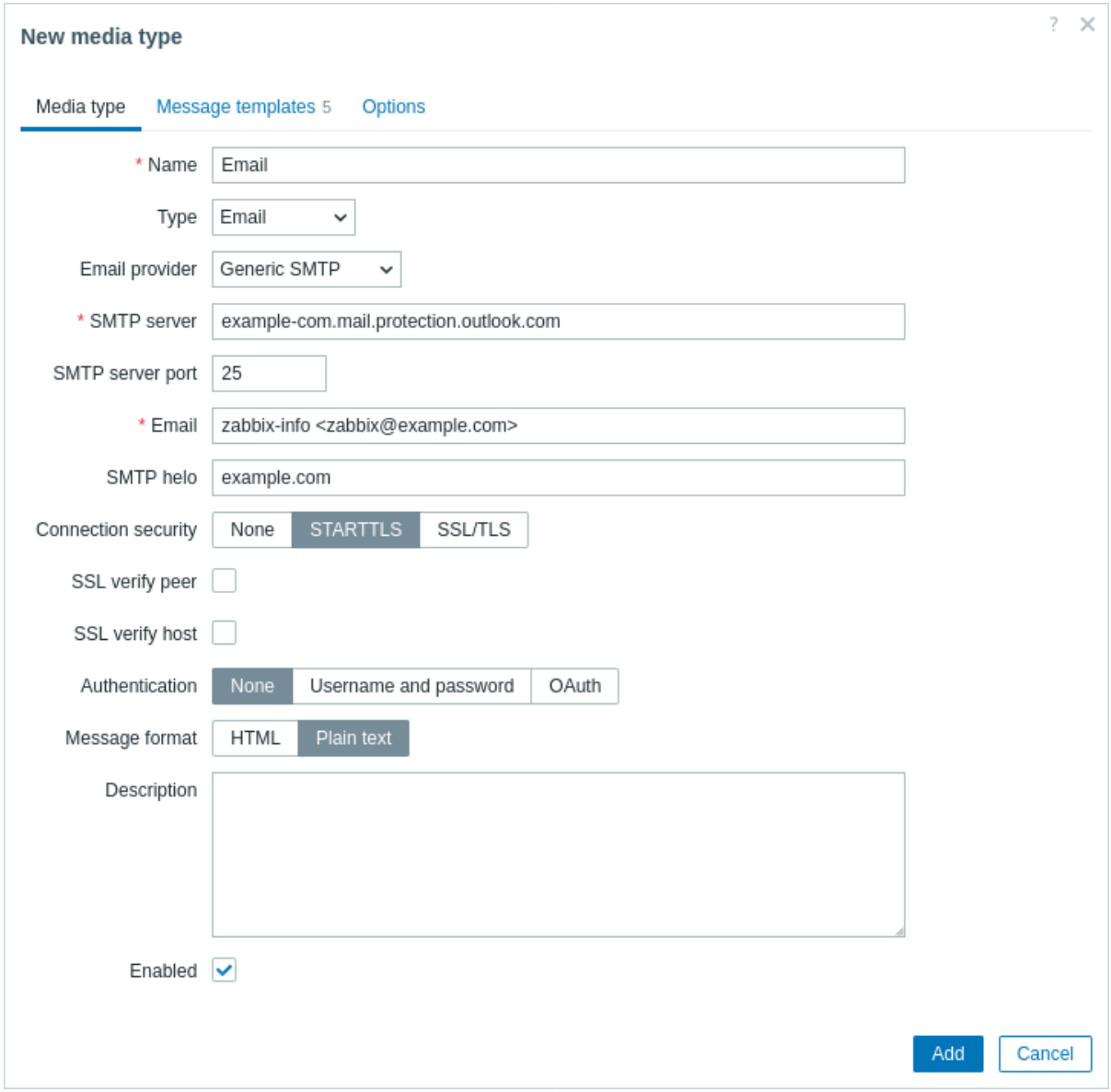
> **Note:**
> Várias notificações para um único evento serão agrupadas no mesmo tópico de e-mail.

Configuração

Para configurar o 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 os 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.

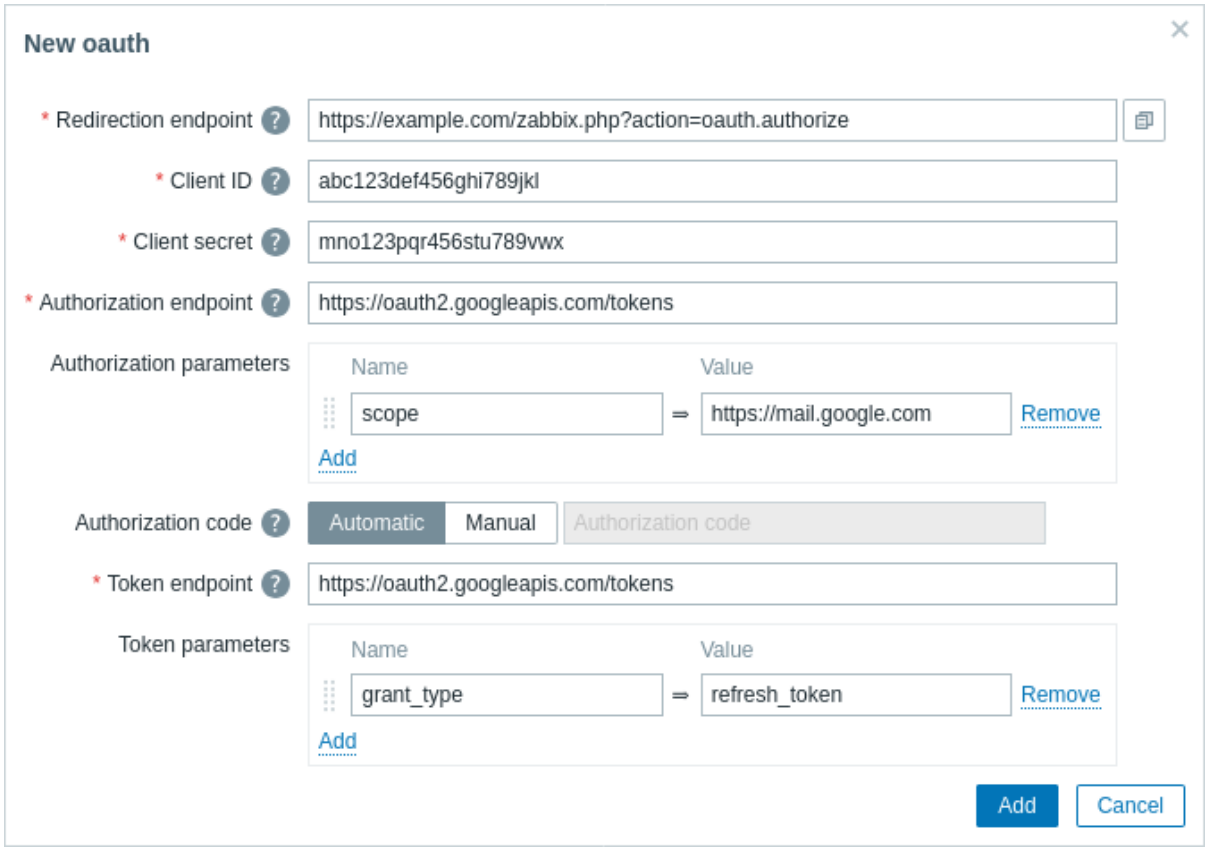
Parâmetro	Descrição
E-mail	O endereço inserido aqui será usado como o endereço **De** para as mensagens enviadas. Adicionar um nome de exibição do remetente (como "Zabbix_info" em *Zabbix_info <zabbix@company.com>* na captura de tela acima) com o endereço de e-mail real é suportado. Existem algumas restrições 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 em torno 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, será enviado o nome de domínio do e-mail (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 **OAuth** - autenticação OAuth A autenticação OAuth não é suportada para o provedor de e-mail *Office365 relay*.
Nome de usuário	Nome de usuário para usar na autenticação. Isso define o valor de CURLOPT_USERNAME. Macros de usuário suportadas.
Senha	Senha para usar 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.
Tokens OAuth	Clique em *Configurar* para definir os parâmetros para recuperar tokens OAuth em uma nova janela. Este campo só está disponível se "OAuth" for selecionado no campo *Autenticação*.
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 de tipo de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

Tokens OAuth

Os seguintes parâmetros são necessários para recuperar tokens OAuth:



Parâmetro	Descrição
Redirection endpoint	Insira a URL do frontend do Zabbix para onde o serviço OAuth irá redirecionar a autorização OAuth (usando o formato `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). É definido automaticamente para um novo tipo de mídia de e-mail quando a URL do frontend está definida.
Client ID	Insira o identificador exclusivo do aplicativo cliente registrado no servidor de autorização OAuth.
Client secret	Insira o segredo privado do aplicativo cliente registrado no servidor de autorização OAuth.
Authorization endpoint	Insira a URL do servidor de autorização OAuth para solicitar a autorização do usuário.
Authorization parameters	Insira os parâmetros para o endpoint de autorização.
Authorization code	Insira o código de autorização: **Automático** - o código será recuperado automaticamente por meio de uma página de redirecionamento **Manual** - insira o código manualmente se a recuperação automática falhar
Token endpoint	Insira a URL do servidor de autorização OAuth para trocar o código de autorização por tokens de acesso e atualização.
Token parameters	Insira os parâmetros para o token de acesso.

Note:
A recuperação de tokens OAuth possui recursos automatizados para os provedores de e-mail **Gmail**, **Gmail relay** e **Office365**. É necessário apenas fornecer os valores dos parâmetros *Redirection endpoint*, *Client ID* e *Client secret*. O Zabbix preencherá automaticamente os outros valores necessários (veja Padrões de URL OAuth por provedor).

Recuperação do token de acesso e atualização do OAuth

Após o envio dos parâmetros OAuth:

1. Uma janela pop-up do navegador é aberta, direcionando o usuário para o *ponto de extremidade de autorização*.
2. O usuário autoriza o Zabbix no serviço OAuth.

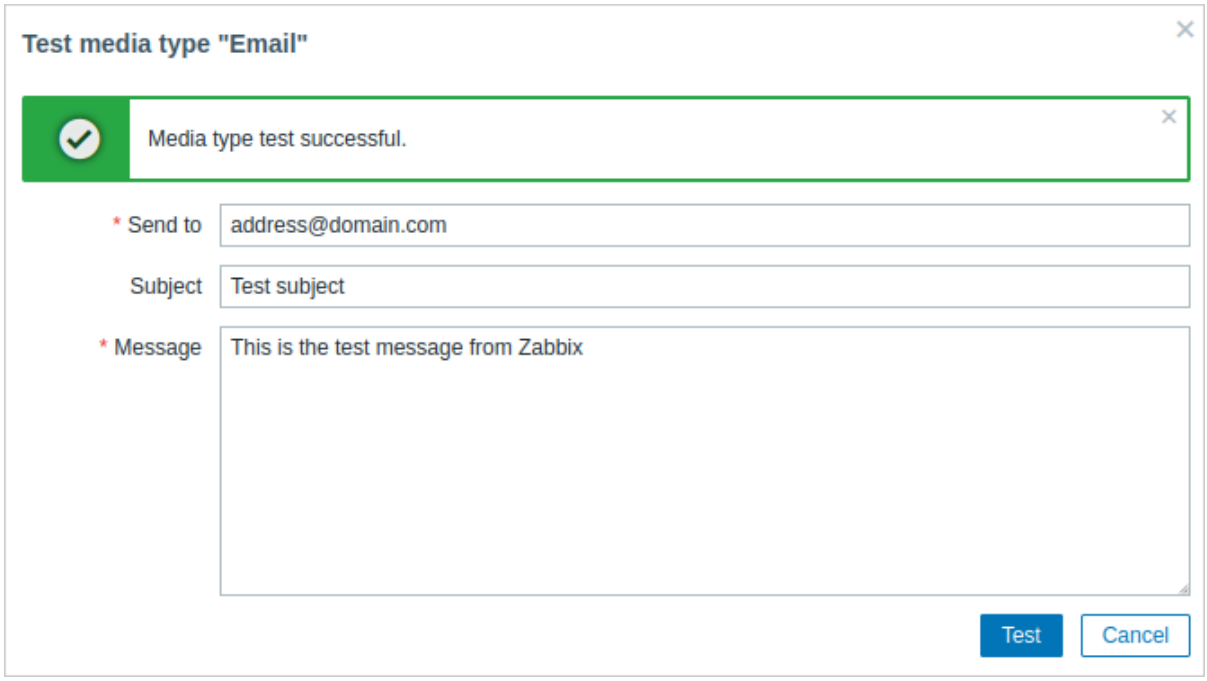
3. O serviço OAuth redireciona o usuário para a ação do frontend do Zabbix `oauth.authorize` com o código de autorização e o valor do escopo.
4. Em resposta, o Zabbix fará uma solicitação ao *ponto de extremidade do token* para trocar o código de autorização por tokens de acesso e atualização.

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. Insira 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 será exibida na mesma janela:



Mídia do usuário

Depois que o tipo de mídia de e-mail estiver 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.

1 Tipos de mídia automatizados do Gmail/Office365

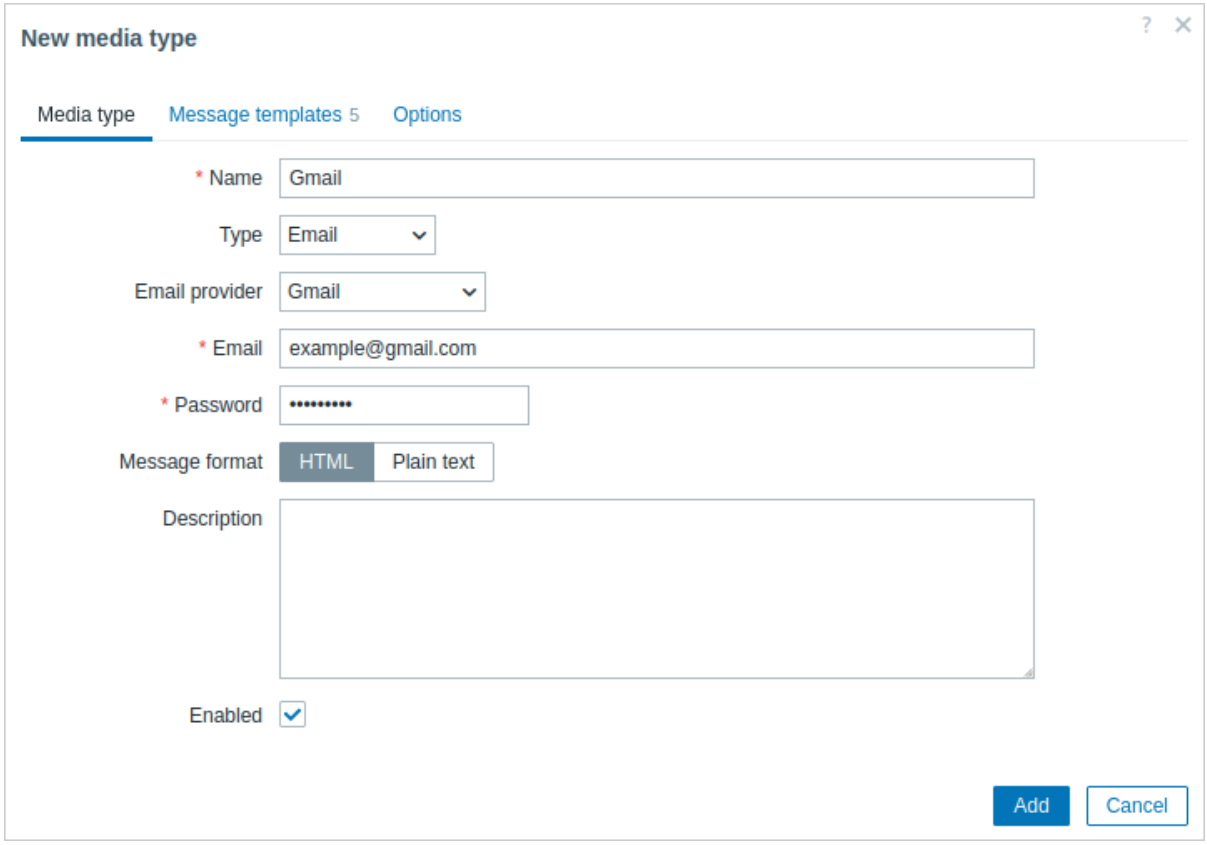
Visão geral

Usuários do Gmail ou Office365 podem se beneficiar de recursos automatizados na configuração do tipo de mídia.

Configuração

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 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 mesmo 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).

Tokens OAuth

A obtenção de tokens OAuth possui recursos automatizados para os provedores de e-mail *Gmail*, *Gmail relay* e *Office365*.

É necessário fornecer apenas um conjunto limitado de parâmetros para a obtenção do token de acesso e de atualização do OAuth - *Redirection endpoint*, *Client ID* e *Client secret*. O Zabbix preencherá automaticamente os outros valores necessários (consulte Padrões de URL OAuth por provedor).

Observe que também é possível usar o formulário SMTP genérico para configurar a autorização OAuth para esses provedores.

Note:
O SmtpClientAuthentication deve ser habilitado por usuário/caixa de correio compartilhada ou para todo o tenant no Office365 (não está habilitado por padrão).

Padrões de URL OAuth por provedor

A tabela a seguir lista os valores padrão de URL OAuth e parâmetros por provedor que são usados pelo Zabbix.

Parâmetro	Gmail	Office365	SMTP genérico
Authorization end-point	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authori`	Sem padrão
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send, offline_access`	Sem padrão
access_type	`offline`	Não usado	Sem padrão
prompt	`consent`	Não usado	Sem padrão

Parâmetro	Gmail	Office365	SMTP genérico
redirect_uri	O valor do campo *Redirection endpoint* é adicionado pelo código automaticamente. Não listado nos parâmetros de *Authorization endpoint*.		
state	Hash exclusivo para identificar a mídia sendo atualizada com o token OAuth é adicionado pelo código automaticamente. Não listado nos parâmetros de *Authorization endpoint*.		
Token endpoint	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Sem padrão
grant_type	`authorization_code`	`authorization_code`	Sem padrão
redirect_uri	O valor do campo *Redirection endpoint* é adicionado pelo código automaticamente. Não listado nos parâmetros de *Token endpoint*.		
state	Hash exclusivo para identificar a mídia sendo atualizada com o token OAuth é adicionado pelo código automaticamente. Não listado nos parâmetros de *Token endpoint*.		

Solução alternativa do Office365 para contas pessoais

As contas pessoais do Office365 não suportam strings de consulta na URL de redirecionamento.

Para contornar esse problema, o administrador do servidor Apache pode adicionar a seguinte regra de reescrita ao arquivo `.htaccess`:

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Isso permite o uso de uma URL de redirecionamento simplificada como:

```
http://server-name/zabbix/oauth.authorize
```

em vez de uma URL com parâmetros de consulta, garantindo a compatibilidade com contas pessoais do Office365.

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 de tipos de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas. Observe que o processamento paralelo do envio de notificações por 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.

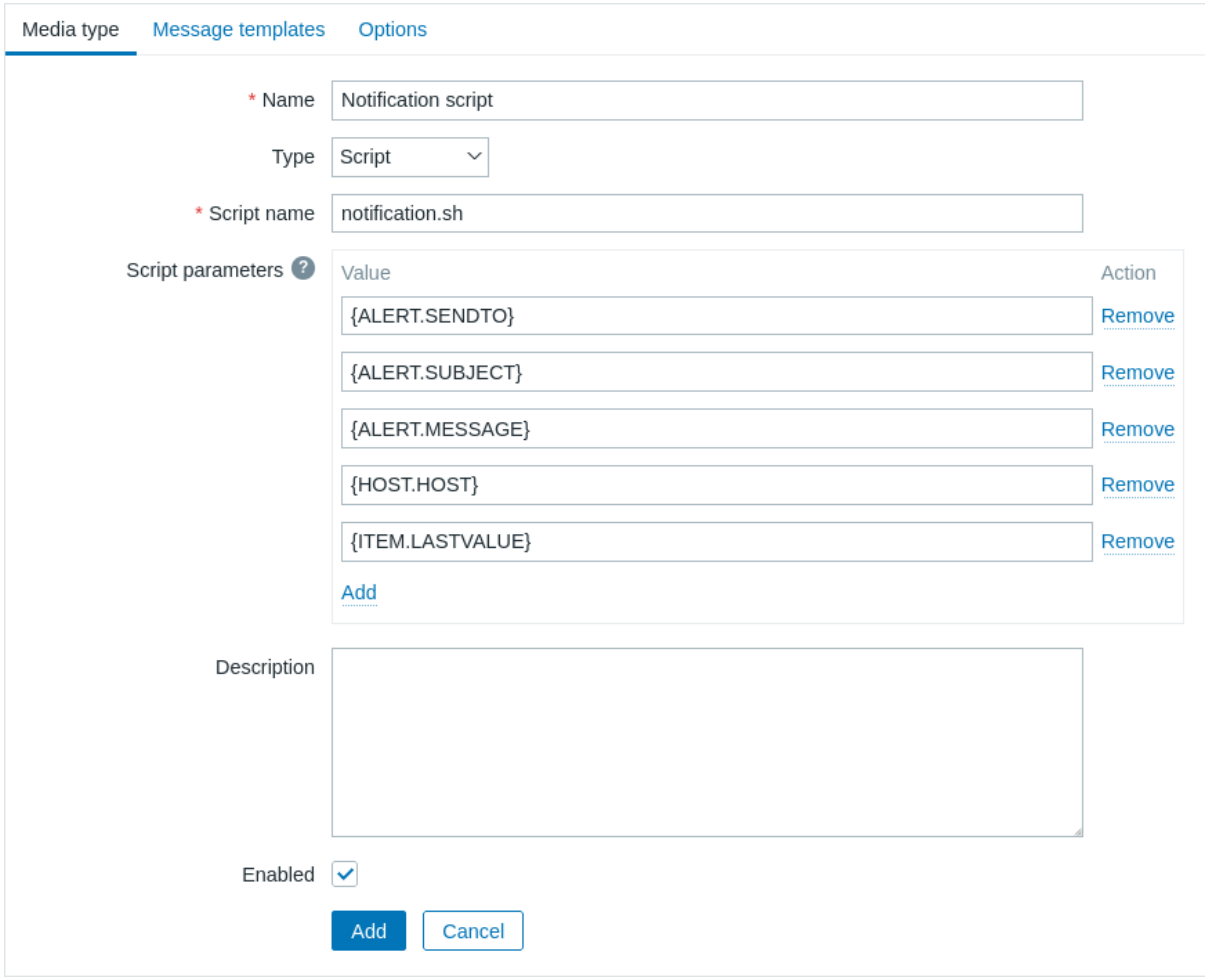
Variáveis de ambiente não são preservadas ou criadas para o script, portanto, devem ser tratadas explicitamente.

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 script:

Parâmetro	Descrição
Nome do script	Informe 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 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.

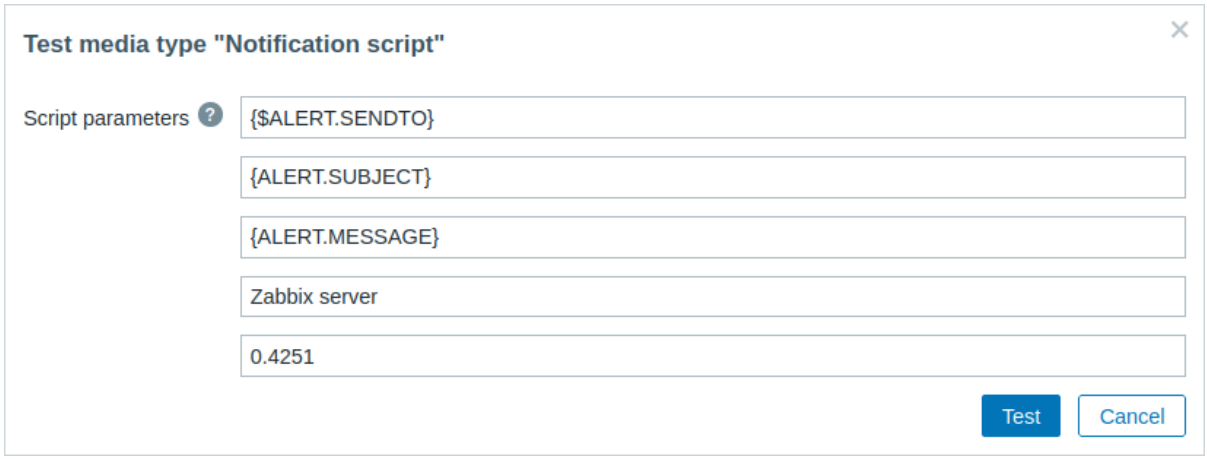
Attention:
Se mais de um tipo de mídia script for configurado, esses scripts podem ser processados em paralelo pelos processos de alerta. O número total de processos de alerta é limitado pelo parâmetro `StartAlerters` no arquivo de configuração do server.

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, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e macros de usuário serão resolvidos 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 softwares externos, 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
- 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 chamados 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 o campo proxy HTTP na configuração do item. 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).

Parâmetro	Descrição
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 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 de 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 marcada, 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.
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 de detalhes adicionais sobre a 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.

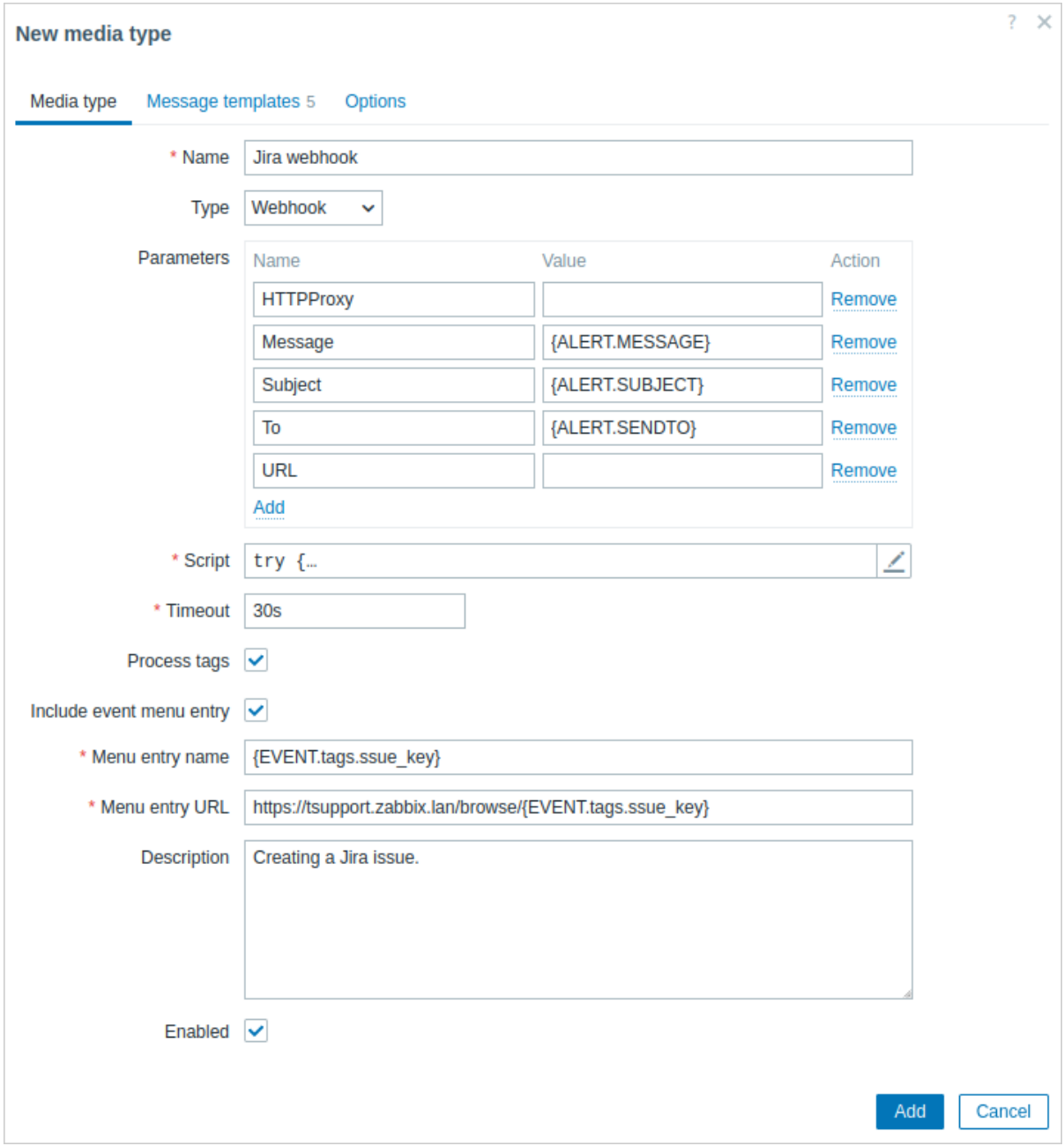
1 Exemplos de scripts de webhook

Visão geral

Embora o Zabbix ofereça um grande número de integrações de webhook disponíveis prontas para uso, você pode querer criar seus próprios webhooks. Esta seção fornece exemplos de scripts de webhook personalizados (usados no parâmetro *Script*). Veja 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 irá criar um chamado no JIRA e retornar algumas informações sobre o chamado 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();
    }

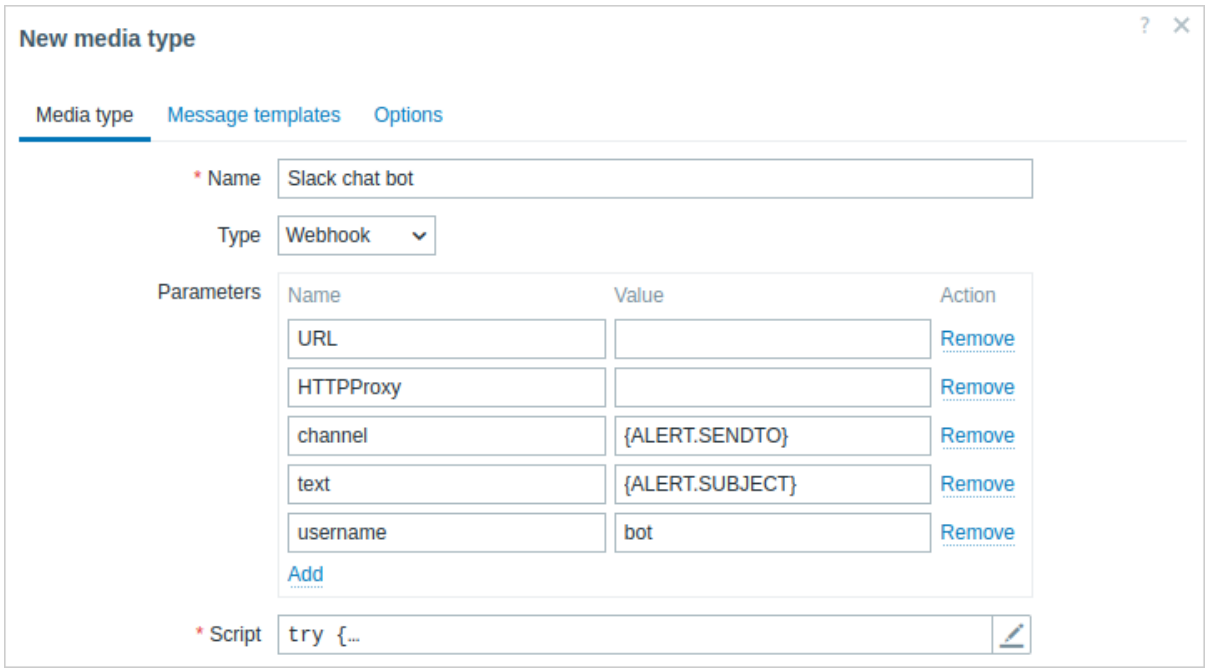
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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 os metadados do host mudam para os já registrados)
- Ações internas - para eventos quando os items se tornam não suportados ou os triggers entram em estado desconhecido

As principais diferenças das ações de serviço são:

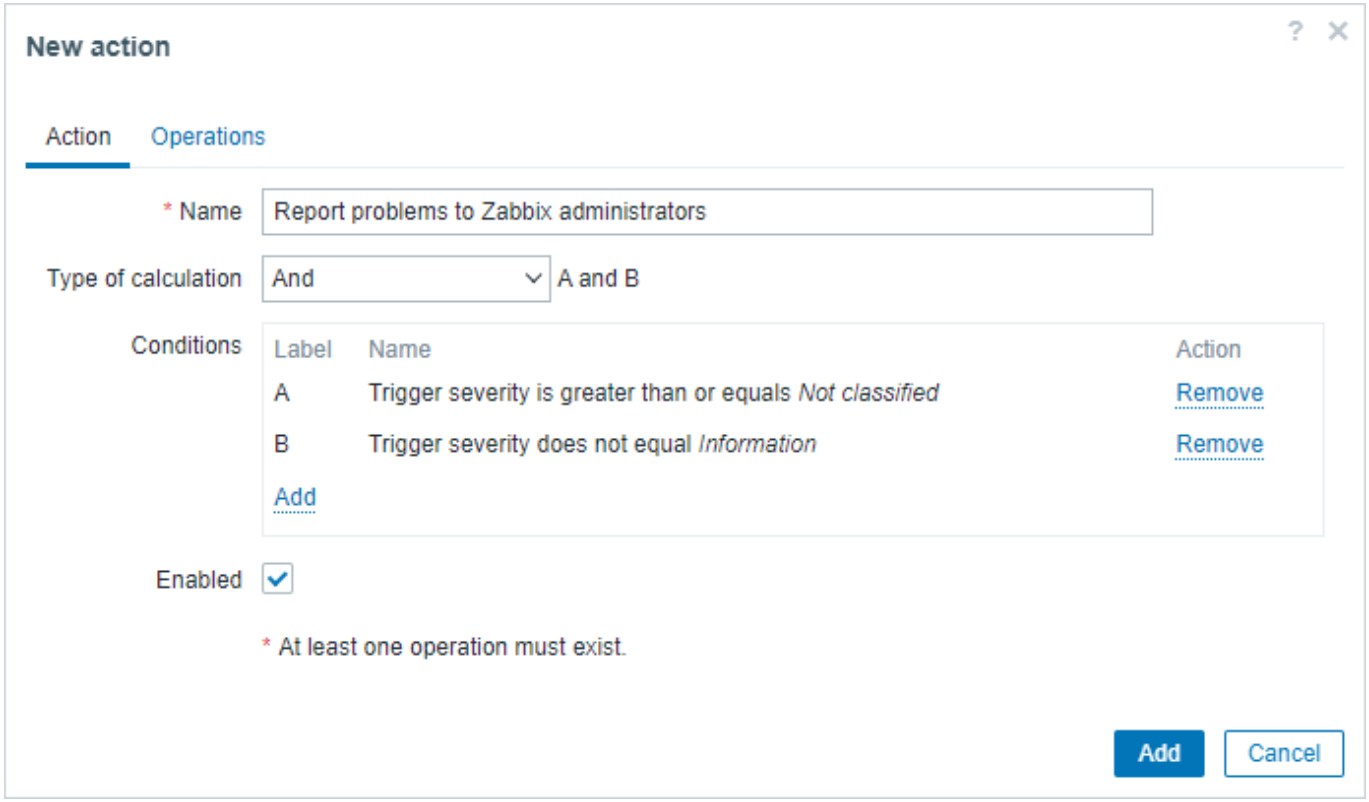
- O acesso do usuário às ações de serviço depende dos direitos de acesso aos serviços concedidos pela função do usuário
- As ações de serviço suportam um conjunto diferente de condições

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 estão 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.
Habilitado	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 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. 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 é igual**.
Template	igual não é igual	Especifique templates ou templates a serem excluídos. **igual** - o evento pertence a uma trigger herdada deste template. **não é igual** - o evento não pertence a uma trigger herdada 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.
Nome da tag	igual não é igual contém não contém	Especifique uma tag de evento ou tag de 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 contendo esta string. **não contém** - o evento não possui uma tag contendo esta string.
Valor da tag	igual não é igual contém não contém	Especifique uma combinação de tag e valor de evento ou 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 contendo estas strings. **não contém** - o evento não possui uma tag e valor contendo estas strings.
Trigger	igual não é igual	Especifique triggers ou triggers a serem excluídas. **igual** - o evento é gerado por esta trigger. **não é 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 é igual é maior ou igual é menor ou igual	Especifique a severidade da trigger. **igual** - igual à severidade da trigger. **não é 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 especificação do período de tempo para descrição do formato. Macros de usuário são suportadas.
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 estas strings. **não contém** - o evento não possui uma tag e valor contendo estas 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
IP do host	igual não é igual	Especifique um intervalo de endereços IP ou um intervalo a ser excluído para um host descoberto. **igual** - o IP do host está no intervalo. **não é igual** - 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.
Tipo de serviço	igual não é igual	Especifique um tipo de serviço de um serviço descoberto ou um tipo de serviço a ser excluído. **igual** - corresponde ao serviço descoberto. **não é igual** - 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
Porta do serviço	igual não é igual	Especifique um intervalo de portas TCP de um serviço descoberto ou um intervalo a ser excluído. **igual** - a porta do serviço está no intervalo. **não é igual** - a porta do serviço não está no intervalo.
Regra de descoberta	igual não é igual	Especifique uma regra de descoberta ou uma regra de descoberta a ser excluída. **igual** - usando esta regra de descoberta. **não é igual** - usando qualquer outra regra de descoberta, exceto esta.
Verificação de descoberta	igual não é igual	Especifique uma verificação de descoberta ou uma verificação de descoberta a ser excluída. **igual** - usando esta verificação de descoberta. **não é igual** - usando qualquer outra verificação de descoberta, exceto esta.
Objeto de descoberta	igual	Especifique o objeto descoberto. **igual** - igual ao objeto descoberto (um dispositivo ou um serviço).
Status de descoberta	igual	**Ativo** - corresponde aos eventos 'Host Ativo' e 'Serviço Ativo'. **Inativo** - corresponde aos eventos 'Host Inativo' e 'Serviço Inativo'. **Descoberto** - corresponde aos eventos 'Host Descoberto' e 'Serviço Descoberto'. **Perdido** - corresponde aos eventos 'Host Perdido' e 'Serviço Perdido'.
Tempo ativo/inativo	é maior ou igual é menor ou igual	Tempo ativo para eventos 'Host Ativo' e 'Serviço Ativo'. Tempo inativo para eventos 'Host Inativo' e 'Serviço Inativo'. **é maior ou igual** - é maior ou igual. O parâmetro é dado em segundos. **é menor ou igual** - é menor ou igual. O parâmetro é dado em segundos.
Valor recebido	igual não é igual é maior ou igual é menor ou igual contém não contém	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). **igual** - igual ao valor. **não é igual** - diferente do valor. **é maior ou igual** - maior ou igual ao valor. **é menor ou igual** - menor ou igual ao valor. **contém** - contém a substring. O parâmetro é dado como uma string. **não contém** - não contém a substring. O parâmetro é dado como uma string.
Proxy	igual não é igual	Especifique um proxy ou um proxy a ser excluído. **igual** - usando este proxy. **não é igual** - 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 **múltiplos** valores forem configurados para as condições `Tipo de serviço`, `Porta do serviço` ou `Valor recebido` na ação, eles serão comparados a um evento de descoberta por vez, mas **não** a vários eventos simultaneamente. Como resultado, ações com múltiplos 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 contendo esta string. **não contém** - o evento não possui uma tag contendo 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 contendo estas strings. **não contém** - o evento não possui uma tag e valor contendo 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.

Exclusão de 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 da 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.

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 auto-registro, 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, passará uma hora 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) utilizado, 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. **Ações** - 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) utilizado, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. **Ações** - 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) utilizado, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. **Ações** - links para editar e remover uma operação são exibidos.
Pausar operações para problemas de sintoma	Marque esta caixa para pausar operações (após a primeira operação) para problemas de sintoma. 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*.
Pausar operações para problemas suprimidos	Marque esta caixa para atrasar o início das operações pela duração de um período de manutenção. Quando as operações forem iniciadas, após a manutenção, todas as operações serão realizadas, incluindo aquelas para os eventos 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, 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 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 no bloco *Operações*. Para editar uma operação existente, clique em 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>** - executa 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).

Parâmetro	Descrição
Passos	Selecione o(s) passo(s) para atribuir a operação em uma programação de escalonamento: **De** - executar a partir deste passo. **Até** - executar até este passo (0=infinito, a execução não será limitada).
Duração do passo	Duração personalizada para esses passos (0=usar a duração padrão do passo). Sufixos de tempo são suportados, por exemplo, 60s, 1m, 2h, 1d. Macros de usuário são suportadas. Várias operações podem ser atribuídas ao mesmo passo. Se essas operações tiverem durações de passo diferentes definidas, a menor será considerada e aplicada ao passo.
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.
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

O envio de uma mensagem é uma das melhores maneiras de notificar as pessoas sobre um problema. É por isso que 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 severidade padrão do trigger (’Não classificado’) **deve ser** marcada na configuração de 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 escalações.

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 seja executado automaticamente no host monitorado mediante alguma condição.

Assim, comandos remotos são um mecanismo poderoso para monitoramento pró-ativo inteligente.

Nos usos mais óbvios do recurso, você pode tentar:

- Reiniciar automaticamente algum aplicativo (servidor web, middleware, CRM) se ele não responder
- Usar o comando IPMI 'reboot' para reiniciar algum servidor remoto se ele não responder às solicitações
- Liberar automaticamente espaço em disco (removendo arquivos antigos, limpando /tmp) se estiver ficando sem espaço em disco
- Migrar uma VM de uma máquina física para outra dependendo da carga da CPU
- Adicionar novos nós a um ambiente de nuvem quando houver recursos insuficientes de CPU (disco, memória, etc.)

Configurar uma ação para comandos remotos é semelhante à configuração para envio de mensagem, sendo a única diferença que o Zabbix executará um comando em vez de enviar uma mensagem.

Comandos remotos podem ser executados pelo Zabbix server, proxy ou agent. Comandos remotos no Zabbix agent podem ser executados diretamente pelo Zabbix server ou através do Zabbix proxy. Tanto no Zabbix agent quanto no Zabbix proxy, comandos remotos estão desabilitados por padrão. Eles podem ser habilitados por:

- adicionando um parâmetro `AllowKey=system.run[*]` na configuração do agent;
- definindo o parâmetro EnableRemoteCommands como '1' na configuração do proxy.

Comandos remotos executados pelo Zabbix server são executados conforme descrito em Execução de comandos, incluindo a verificação do código de saída.

Comandos remotos são executados mesmo que o host de destino esteja 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
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 de 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

Esses comandos remotos que são executados no agent do 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 esse parâmetro.

Então, ao configurar uma nova ação em *Alertas* → *Ações* → *Ações de trigger*:

1. Defina as condições apropriadas, por exemplo, defina que a ação é ativada em qualquer problema de desastre com um dos aplicativos 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 destino* para o script.

Scripts predefinidos

Scripts que estão 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 para 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 ao resultado 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 ao resultado 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 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

- <comando> - um dos comandos IPMI sem espaços
- <valor> - 'on', 'off' ou qualquer número 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 sob determinada condição.

Para reiniciar automaticamente o Windows quando um problema for detectado 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 operações adicionais 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, recomenda-se 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 deve ser usado para um nome de host, então todos os ":" (dois-pontos) são substituídos por "_" (sublinhados), já que dois-pontos não são permitidos em nomes de host.

Attention:
Se estiver realizando a descoberta por um proxy, atualmente a busca pelo nome do host ainda ocorre no Zabbix server.

Attention:
Se um host já existir 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 das mensagens e no texto 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

Os 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, ela 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, a mensagem 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, ela será substituída por algo como:

```
Nome do problema: Processor load is too high on a local host
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 (Processor load (1 min average per core))
2. Valor do item em Myotherhost: 5.125 (Processor load (1 min average per core))
```

Exemplo 6

Recebendo detalhes do evento de problema e 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 um único passo e, portanto, serão executadas simultaneamente.

Casos de uso

Alguns casos de uso para operações de recuperação são os seguintes:

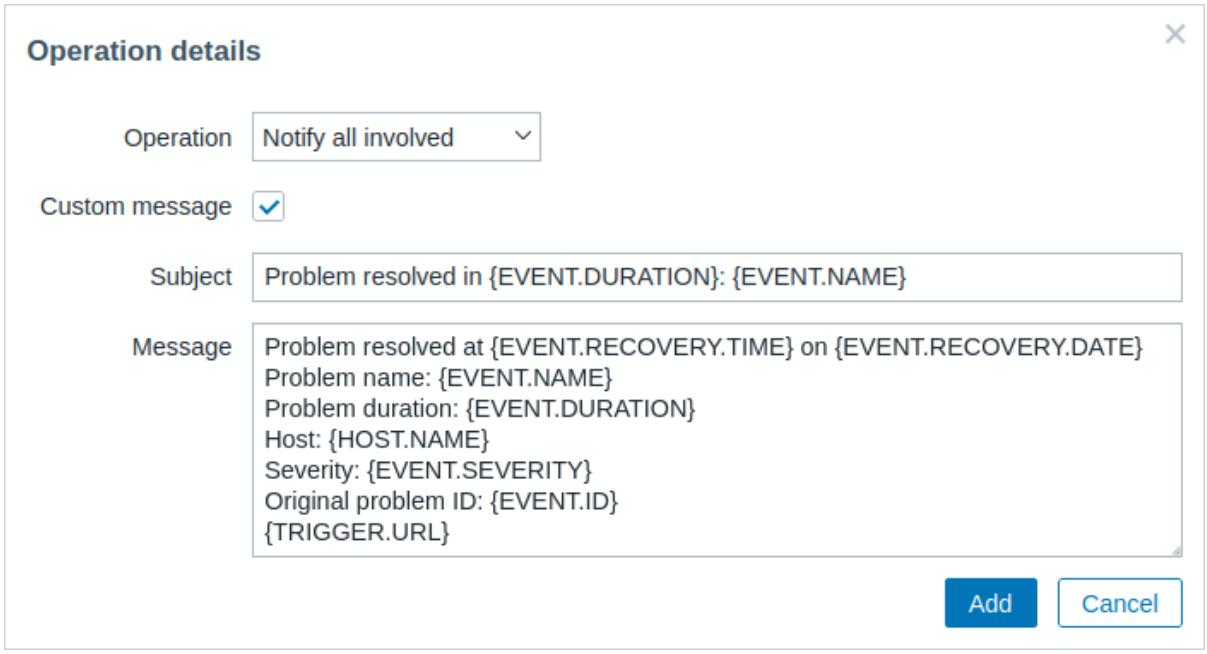
1. Notificar na recuperação todos os usuários que foram notificados sobre o problema:
    - Selecione *Notificar todos os envolvidos* como tipo de operação.
2. Ter várias operações na recuperação: enviar uma notificação e executar um comando remoto:
    - Adicione tipos de operação para enviar uma mensagem 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, vá até a aba *Operações* na configuração da ação.

Para configurar os detalhes de uma nova operação de recuperação, clique em no bloco *Operações de recuperação*. Para editar uma operação existente, clique em ao lado da operação. Uma janela pop-up será aberta onde você pode editar os detalhes da etapa 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 definidos anteriormente 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.

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.

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 Atualizar operações

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, vá até a aba *Operações* na configuração da ação.

Para configurar os detalhes de uma nova operação de atualização, clique em no bloco *Operações de atualização*. Para editar uma operação existente, clique em ao lado da operação. Uma janela pop-up será aberta onde você pode editar os detalhes da etapa da operação.

Detalhes da operação de atualização

As operações de atualização oferecem o mesmo conjunto de parâmetros que as operações de recuperação.

5 Escalonamentos

Visão geral

Com as escalonamentos, você pode criar cenários personalizados para o envio de notificações ou execução de comandos remotos.

Na prática, isso 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 adiado.
- As notificações podem ser escalonadas para outro grupo de usuários ”superior”.
- Comandos remotos podem ser executados imediatamente ou quando um problema não for resolvido por um longo período.

As ações são escalonadas com base no **passo de escalonamento**. Cada passo tem uma duração em tempo.

Você pode definir tanto a duração padrão quanto uma duração personalizada de um passo individual. A duração mínima de um passo de escalonamento é de 60 segundos.

Você pode iniciar ações, como enviar notificações ou executar comandos, a partir de qualquer passo. O passo um é para ações imediatas. Se você quiser atrasar uma ação, pode atribuí-la a um passo posterior. Para cada passo, várias ações podem ser definidas.

O número de passos de escalonamento não é limitado.

Os escalonamentos são definidos ao configurar uma operação. Os escalonamentos são suportados apenas para operações de problema, não para recuperação.

Aspectos diversos do comportamento de escalonamento

Vamos considerar o que acontece em diferentes circunstâncias se uma ação contiver 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 configuração da ação, todas as etapas restantes do escalonamento são executadas com um atraso causado pelo período de manutenção ou sem atraso. Um período de manutenção não cancela operações.
O período de tempo definido na condição de ação ***Período de tempo*** *termina após o envio da notificação inicial*	Todas as etapas restantes do escalonamento são executadas. A condição *Período de tempo* não pode interromper operações; ela tem efeito em relação a quando as ações são iniciadas/não iniciadas, não 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 configuração da ação, todas as etapas do 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	Deve aguardar o disparo do trigger, antes que todas as etapas do escalonamento sejam executadas.
Diferentes escalonamentos seguem em rápida sucessão e se sobrepõem	A execução de cada novo escalonamento substitui o escalonamento anterior, mas pelo menos uma etapa de escalonamento é sempre executada no escalonamento anterior. Esse comportamento é relevante em ações sobre eventos que são criados com CADA avaliação de problema do trigger.

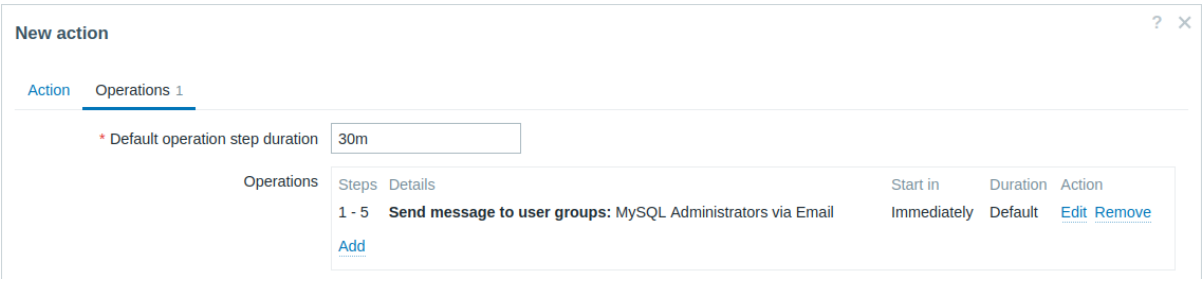
Situação	Comportamento
Durante um escalonamento em andamento (como uma mensagem sendo enviada), com base em qualquer tipo de evento: - a ação é desabilitada Com base em evento de trigger: - o trigger é desabilitado - o host ou item é desabilitado Com base em evento interno sobre triggers: - o trigger é desabilitado Com base em evento interno sobre itens/regras de descoberta de baixo nível: - o item é desabilitado - o host é desabilitado	A mensagem em andamento é enviada e, em seguida, mais uma mensagem no escalonamento é enviada. A mensagem de acompanhamento terá o texto de cancelamento no início do corpo da mensagem (*NOTA: Escalonamento cancelado*) nomeando o motivo (por exemplo, *NOTA: Escalonamento cancelado: ação '<Nome da ação>' desabilitada*). Dessa forma, o destinatário é informado de que o escalonamento foi cancelado e nenhuma outra etapa será executada. Esta mensagem é enviada a todos que receberam as notificações anteriores. O motivo do cancelamento também é registrado no arquivo de log do server (a partir do Nível de Depuração 3=Aviso). Observe que a mensagem *Escalonamento cancelado* também é enviada se as operações forem concluídas, mas as operações de recuperação estiverem configuradas e ainda não forem executadas.
Durante um escalonamento em andamento (como uma mensagem sendo enviada) a ação é excluída	Nenhuma outra mensagem é enviada. A informação é registrada no arquivo de log do server (a partir do Nível de Depuração 3=Aviso), por exemplo: `escalonamento cancelado: ação id:334 excluída`

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 aba *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 seja, 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 do banco de dados. Observe que o gerente só receberá uma mensagem caso o problema ainda não tenha 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 escalada, como notificações enviadas e comandos executados.

Exemplo 4

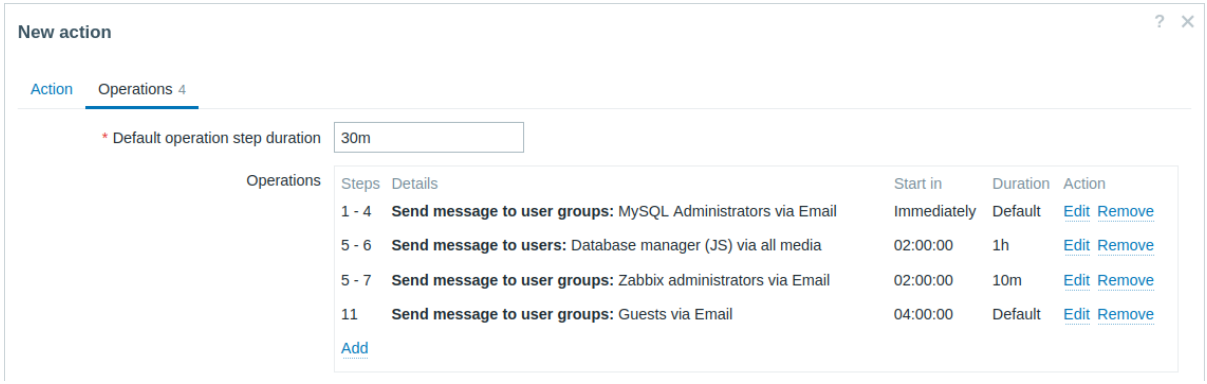
Um cenário mais complexo. Após várias mensagens para os administradores do MySQL e escalonamento para o gerente, o Zabbix tentará reiniciar o banco de dados MySQL. Isso acontecerá se o problema existir por 2:30 horas e não tiver sido reconhecido.

Se o problema ainda existir, após mais 30 minutos o Zabbix enviará uma mensagem para todos os usuários convidados.

Se isso não ajudar, após 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 atribuídas a uma etapa e intervalos personalizados usados. 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, 1:30 após o início do problema.
- Para o gerente do banco de dados às 2:00 e 2:10. (e não às 3:00; considerando que as etapas 5 e 6 se sobrepõem à próxima operação, a duração personalizada mais curta de 10 minutos na próxima operação substitui a duração mais longa de 1 hora tentada aqui).
- Para os administradores do Zabbix às 2:00, 2:10, 2:20 após o início do problema (a duração personalizada da etapa de 10 minutos funcionando).
- Para usuários convidados às 4:00 horas após o início do problema (a duração padrão da etapa de 30 minutos retornando entre as etapas 8 e 11).

3 Recebendo notificação sobre items 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 vice-versa);
- quando triggers passam de ’normal’ para ’desconhecido’ (e vice-versa);
- quando regras de descoberta de baixo nível passam de ’normal’ para ’não suportado’ (e vice-versa).

Esta seção apresenta um passo a passo para **receber notificação** quando um item se tornar não suportado.

Configuração

No geral, o processo de configuração da notificação deve ser familiar para aqueles que já 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 comunicação do usuário se você quiser receber notificações para eventos internos.

Passo 2

Vá para *Alertas* → *Ações* → *Ações internas*.

Clique em *Criar ação* no canto superior direito da página para abrir o formulário de configuração da 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.

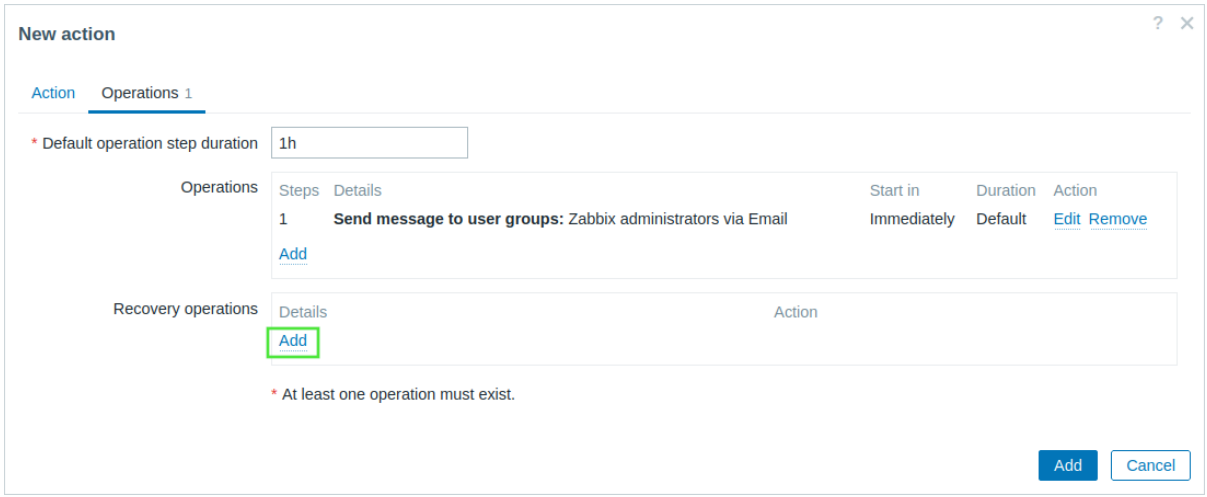
Selecione alguns destinatários da mensagem (grupos de usuários/usuários) e o tipo de mídia (ou ”Todos”) a ser usado para entrega. Marque a caixa de seleção *Mensagem personalizada* se desejar inserir o assunto/conteúdo personalizado da mensagem de problema.

Clique em *Adicionar* para realmente listar a operação no bloco *Operações*.

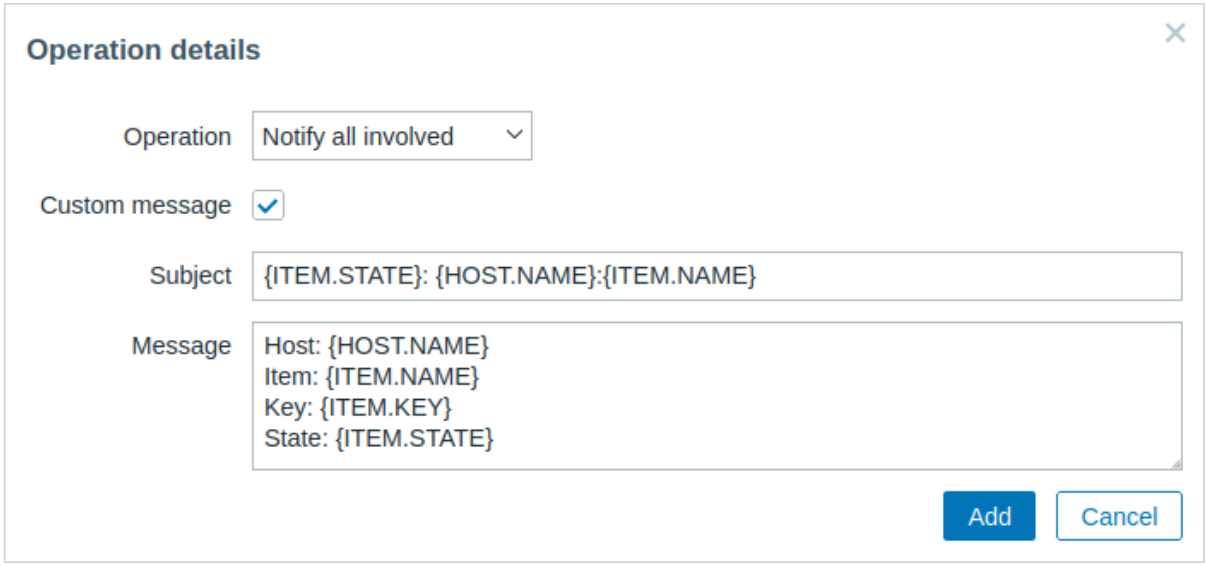
Se você deseja receber mais de uma notificação, defina a duração do passo da operação (intervalo entre as mensagens enviadas) e adicione outro passo.

Passo 5

O bloco *Operações de recuperação* permite configurar uma notificação de recuperação quando um item volta 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 problema.



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 uma série de macros internas que podem ser usadas em várias situações. Essas macros são variáveis, identificadas por uma sintaxe específica:

`{MACRO}`

As macros são resolvidas 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, uma macro pode ser usada 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.

As macros podem ser usadas nos parâmetros da chave do item. Uma macro pode ser usada apenas em 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 na macro resolvida.

Existem outros tipos de macros no Zabbix.

O Zabbix suporta as seguintes macros:

- `{MACRO}` - macro interna (veja a lista completa)
- `{<macro>.<func>(<params>)}` - funções de macro
- `{$MACRO}` - macro definida 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 explicitamente declarado que apenas uma macro é esperada (por exemplo, ao configurar macros de host ou filtros de regras de descoberta de baixo nível).

É suportada uma única função por macro; múltiplas funções de macro em cadeia não são suportadas.

> **Note:**
> Veja 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 em 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 de codificação HTML.
htmlencode	Codificação do valor da macro em 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.
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 de codificação de URL.
urlencode	Codificação do valor da macro em codificação de URL.

Detalhes da função

Os parâmetros opcionais da função são indicados por < >.

btoa

Codifica um valor de macro em Base64. A codificação Base64 é um método para representar dados binários como texto, útil para armazenar e transmitir com segurança conteúdo binário por protocolos baseados em texto.

Exemplo:

```
{{ITEM.VALUE}.btoa()} - codificará em Base64 um valor como "zabbix" em "emFiYml4"
```

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 "outubro" a partir de um valor recebido de "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} - retornará "1 setembro" a partir de um valor recebido de "1633098961"
```

htmldecode

Decodifica um valor de macro de uma codificação HTML.

Os seguintes caracteres são suportados:

Valor	Valor decodificado
`&`	`&`
`<`	`<`
`>`	`>`
`"`	`"`
`'`	`'`
`'`	`'`

Exemplo:

```
{{ITEM.VALUE}.htmldecode()} - irá decodificar HTML de um valor como "&lt;" para "<"
```

htmlencode

Codifica um valor de macro em codificação HTML.

Os seguintes caracteres são suportados:

Valor	Valor codificado
`&`	`&`
`<`	`<`
`>`	`>`

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 como "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.

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.

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 dos caracteres do valor da macro.

- **caracteres** - o conjunto de caracteres a serem substituídos;
- **substituição** - o conjunto de caracteres de substituição correspondentes por posição.

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 ocorrênc
{{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.

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.

Exemplo:

```
{{ITEM.VALUE}.urldecode()} - irá decodificar um valor como "%2F" em "/"
```

urlencode

Codificando um valor de macro em codificação de URL.

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.

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:"{{#M}.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, ela não será resolvida.

Attention:
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:
Recomenda-se 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 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, 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}** seja 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 hash 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 *Mídia* de um usuário;
- use-a para configurar a coleta de item mais frequente durante o horário de trabalho:

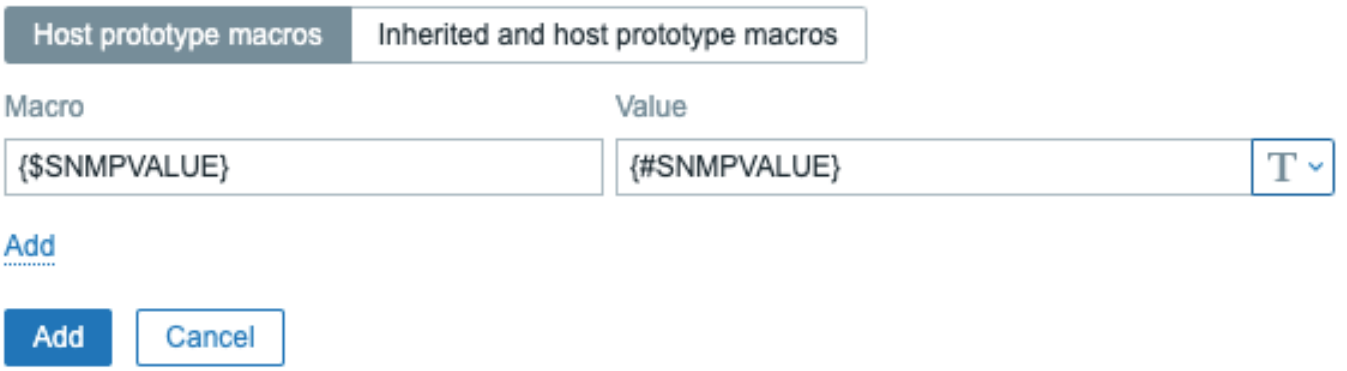


- use-a na condição de ação *Período de tempo*;
- ajuste o horário de trabalho em *Administração → Macros*, se necessário.

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

Consulte macros de usuário com contexto.

3 Macros de usuário com contexto

Visão geral

Um contexto opcional pode ser usado em macros de usuário, permitindo substituir o valor padrão por um valor específico de 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 na configuração de macro de usuário. Se o prefixo `regex:` for usado em outro lugar como contexto de macro de usuário, como em uma expressão de trigger, ele será tratado como contexto estático.

A citação do contexto é opcional (veja também notas importantes).

Exemplos de contexto de macro:

Exemplo	Descrição
{$LOW_SPACE_LIMIT}	Macro de usuário sem contexto.
{$LOW_SPACE_LIMIT:/tmp}	Macro de usuário com contexto (string estática).
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	Macro de usuário com contexto (expressão regular). Igual a {$LOW_SPACE_LIMIT:/tmp}.
{$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

Macros de usuário com contexto podem ser definidas para realizar limites de trigger mais flexíveis em expressões de trigger (com base nos valores recuperados 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 nas 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

Notas importantes

- Se mais de uma macro de usuário com contexto existir, o Zabbix tentará corresponder primeiro as macros de contexto simples e depois as macros de contexto com expressões regulares em uma ordem indefinida.

Warning:
Não crie macros de contexto diferentes que correspondam à mesma string para evitar comportamento indefinido.

- Se uma macro com seu contexto não for encontrada no host, templates vinculados ou globalmente, então a macro sem contexto será pesquisada.

- Apenas macros de descoberta de baixo nível são suportadas no contexto. Qualquer outra macro é ignorada e tratada como texto simples.

Tecnicamente, o contexto da macro é especificado usando regras semelhantes aos parâmetros da chave do item, exceto que o contexto da macro não é analisado como vários parâmetros se houver um caractere `,`:

- O contexto da macro deve ser colocado entre aspas com " se o contexto contiver um caractere } ou começar com um caractere ". Aspas dentro do contexto entre aspas devem ser escapadas com o caractere \.
- O próprio caractere \ não é escapado, o que significa que não é possível ter um contexto entre aspas terminando com o caractere \ - a macro {$MACRO:"a:\b\c\"} é inválida.
- Os espaços à esquerda no contexto são ignorados, os espaços à direita não:
  - Por exemplo, {$MACRO:A} é o mesmo que {$MACRO: A}, mas não {$MACRO:A }.
- Todos os espaços antes das aspas iniciais e após as aspas finais são ignorados, mas todos os espaços dentro das aspas não são:
  - As macros {$MACRO:"A"}, {$MACRO: "A"}, {$MACRO:"A" } e {$MACRO: "A" } são as mesmas, mas as macros {$MACRO:"A"} e {$MACRO:" A "} não são.

As seguintes macros são todas equivalentes, porque têm o mesmo contexto: {$MACRO:A}, {$MACRO: A} e {$MACRO:"A"}. Isso contrasta com as chaves de item, onde 'key[a]', 'key[a]' e 'key["a"]' são semanticamente iguais, mas diferentes 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

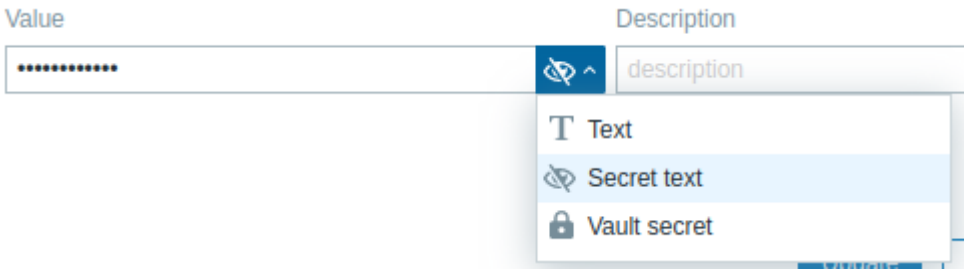
Note:
Embora o valor de uma macro secreta fique oculto, ele pode ser revelado por meio do uso em items. Por exemplo, em um script externo, uma instrução `echo` referenciando uma macro secreta pode ser usada para revelar o valor da macro para o frontend, porque o Zabbix server tem acesso ao valor real da macro. Veja 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 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 no 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 de cada provedor, consulte:

- HashiCorp
- CyberArk

Os valores das macros secretas do vault são recuperados do vault pelo Zabbix server (e pelo Zabbix proxy, se *Resolver macros secretas do vault por* estiver definido como *Zabbix server e proxy*) a cada atualização dos dados de configuração e, em seguida, armazenados no cache de configuração. O Zabbix server e o Zabbix proxy podem usar vaults diferentes.

Se *Resolver macros secretas do vault por* estiver definido como *Zabbix server*, então os segredos do vault são recuperados apenas pelo server e 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. 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` runtime control (apenas no server).

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:
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*
SNMP agent	*Comunidade SNMP* *Nome do contexto* (SNMPv3) *Nome de segurança* (SNMPv3) *Senha de autenticação* (SNMPv3) *Senha de privacidade* (SNMPv3)
HTTP agent	*URL* *Campos de consulta* *Post*

Contexto	Parâmetro
	Cabeçalhos
	Nome de usuário
	Senha
	Senha da chave SSL
Script	*Parâmetros*
	Script
Browser	*Parâmetros*
	Script
Database monitor	*Consulta SQL*
TELNET agent	*Script*
	Nome de usuário
	Senha
SSH agent	*Script*
	Nome de usuário
	Senha
Simple check	*Nome de usuário*
	Senha
JMX agent	*Nome de usuário*
	Senha
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 de 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)
	Senha de autenticação (SNMPv3)
	Senha de privacidade (SNMPv3)
Scripts globais	
Webhook	*Script JavaScript*
	Valor do parâmetro do script JavaScript

Contexto	Parâmetro
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 item, trigger, gráfico e regra LLD. Então, ao descobrir sistemas de arquivos reais, interfaces de rede, etc., essas macros são substituídas por valores reais e são a base para a criação de entidades reais.

Essas macros também são usadas na criação de *protótipos* de host e grupo de hosts na descoberta de máquinas virtuais.

Algumas macros de descoberta de baixo nível vêm ”pré-embaladas” com a função LLD no Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. No entanto, seguir esses nomes não é obrigatório 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 de um dos seguintes tipos primitivos:

- string;
- number;
- boolean.

Arrays, objetos e valores nulos não são suportados. Qualquer macro LLD que se refira a esse tipo de valor permanecerá não expandida e aparecerá literalmente (por exemplo, `'{#MY_MACRO}'`) durante o pré-processamento e a criação do item.

Locais suportados

As macros LLD podem ser usadas:

- no filtro da regra de descoberta de baixo nível
- para protótipos de item e protótipos de descoberta em
  - nome
  - parâmetros da chave
  - unidade
  - intervalo de atualização[1]
  - tempo limite[1]
  - período de armazenamento do histórico[1]
  - período de armazenamento de 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 campos de consulta HTTP do agente HTTP
    - campo de corpo da requisição do agente HTTP
    - campo de códigos de status necessários do agente HTTP
    - chave e valor do campo de 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 personalizados: 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 estático.

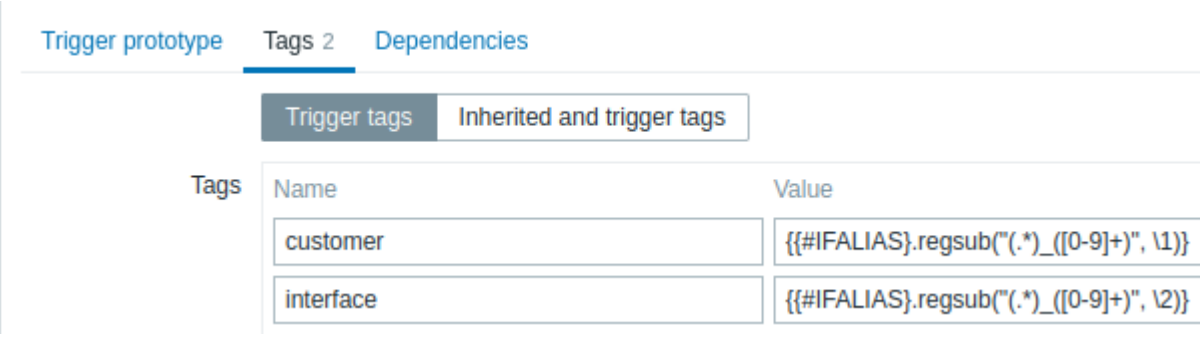
Usando funções de macro

Funções de macro são suportadas com macros de descoberta de baixo nível (exceto na regra de descoberta de baixo nível filtro), 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 de funções de macro, consulte: Funções de macro

Funções de macro são suportadas em macros de descoberta de baixo nível desde o Zabbix 4.0.

Notas de rodapé

[1] Nos campos marcados com [1], um único macro deve preencher todo o campo. Vários macros em um campo ou macros misturados com texto não são suportados.

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 suportados, consulte Macros suportadas.

Observações sobre o uso:

- Use macros {FUNCTION.*} para referenciar valores de função de expressões de trigger/expressões de recuperação.
- Use macros {HOST.HOST<1-9>} e {ITEM.KEY<1-9>} para referenciar hosts e items.
- Em templates, use macros {HOST.HOST<1-9>} ou omita o host completamente para o primeiro host—por exemplo, `{?avg(//item1,1h)}`—em vez de nomes de template, pois os nomes de template não são substituídos por nomes de host durante o vinculamento de template.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

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 no Zabbix acessam o aplicativo Zabbix através do frontend baseado na web. Cada usuário recebe um nome de login exclusivo e uma senha.

Todas as senhas dos usuários são criptografadas e armazenadas no banco de dados do 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 adequadamente no UNIX. A comunicação entre o servidor web e o navegador do usuário pode ser protegida usando SSL.

Com um esquema flexível de permissões de usuário, você pode restringir e diferenciar direitos para:

- acessar funções administrativas do frontend do Zabbix
- executar determinadas ações no frontend
- acessar hosts monitorados em grupos de hosts
- usar métodos específicos da API

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:

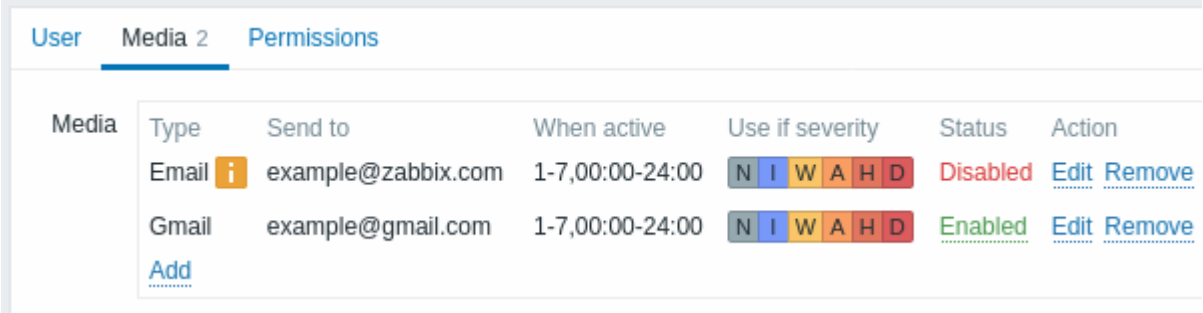
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome de usuário	Nome de usuário exclusivo, 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 de preenchimento automático, portanto, 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. Clicar no botão *Alterar senha* abre dois campos para digitar uma nova senha. Para o usuário com o *Papel de superadministrador* alterando sua própria senha, clicar no botão *Alterar senha* abre 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 usam 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.
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.

Parâmetro	Descrição
Logout automático	Com esta caixa marcada, o usuário será desconectado automaticamente após o tempo definido em 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.

Note:
As permissões do usuário para alterar detalhes de mídia para si mesmo podem ser concedidas/revogadas com base em sua função de usuário (veja a opção *Criar e editar a própria mídia*). As permissões do superadministrador para alterar detalhes de mídia para outros também podem ser concedidas/revogadas com base em sua função de usuário (veja a opção *Criar e editar mídia do usuário*)

Veja a seção Tipos de mídia para detalhes sobre a configuração de 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 *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 tal 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 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 atribuídas explicitamente;
- *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 concedidas explicitamente;
- *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 daquela 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		+	+
	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		+	+

Seção do menu		Usuário	Admin	Super admin
	Correlação de eventos			+
	Descoberta		+	+
Alertas			+	+
	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			+
	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 aba *Permissões*, por usuários Super admin.

Acesso a hosts

O acesso a qualquer host e dados de 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 host). 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.

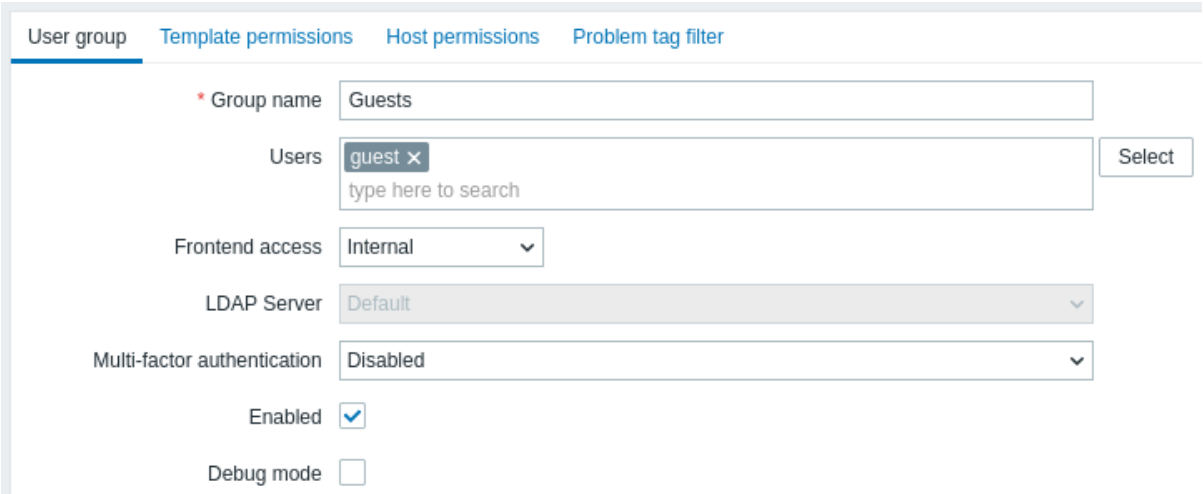
Configuração

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

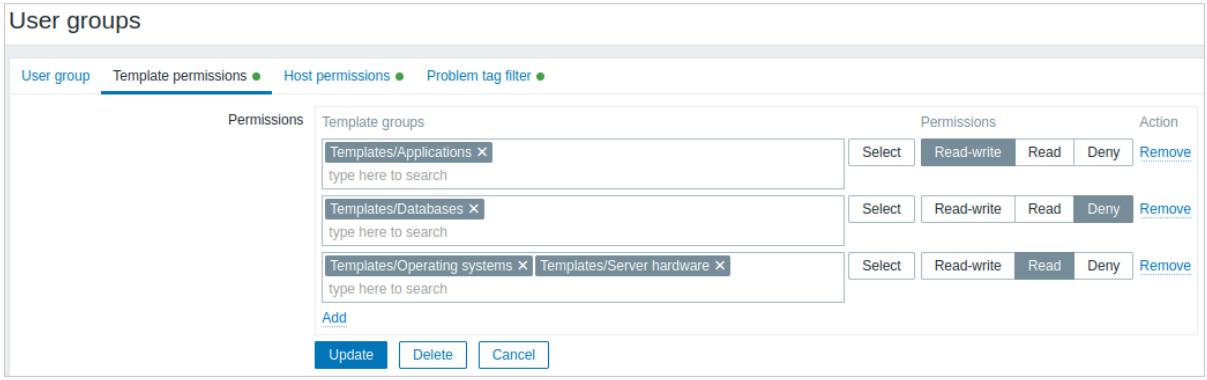
A aba **Grupo de usuários** contém os atributos gerais do grupo:



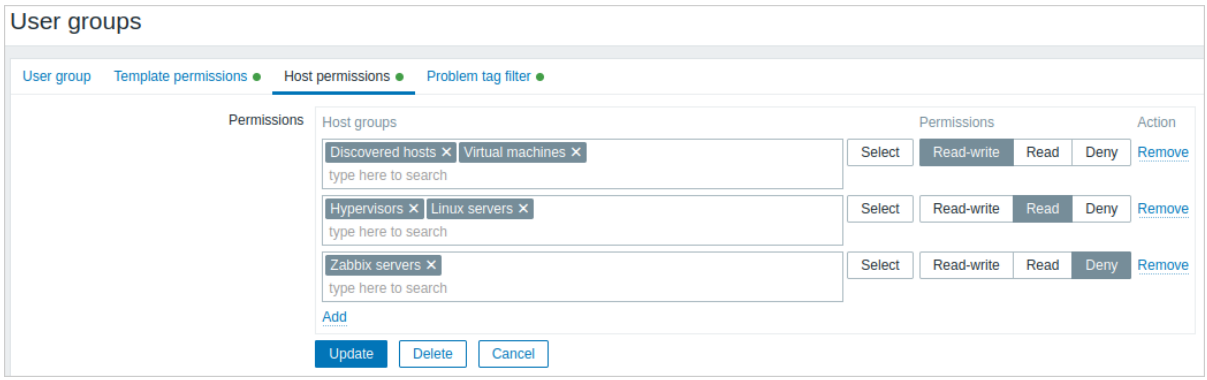
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

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 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 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.

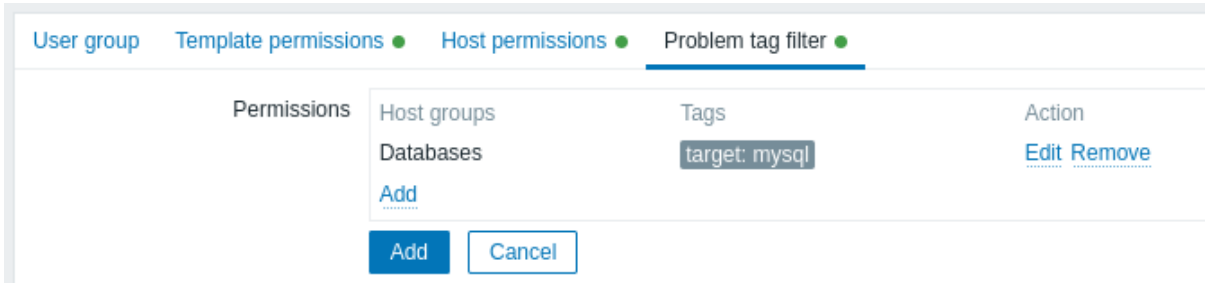
Observe que um usuário *Super admin* pode forçar grupos aninhados a terem o mesmo nível de permissões que o grupo pai; isso pode ser feito no formulário de configuração do grupo de host/template.

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 **Leitura-escrita** para um host e **Negar** ou nenhuma permissão para um template vinculado a esse host, os usuários desse grupo não poderão editar itens de template no host, e o nome do template será exibido como *Template inacessível*.

A aba **Filtro de tag de problema** permite definir permissões baseadas em tag para que grupos de usuários vejam problemas filtrados por nome e valor da tag:



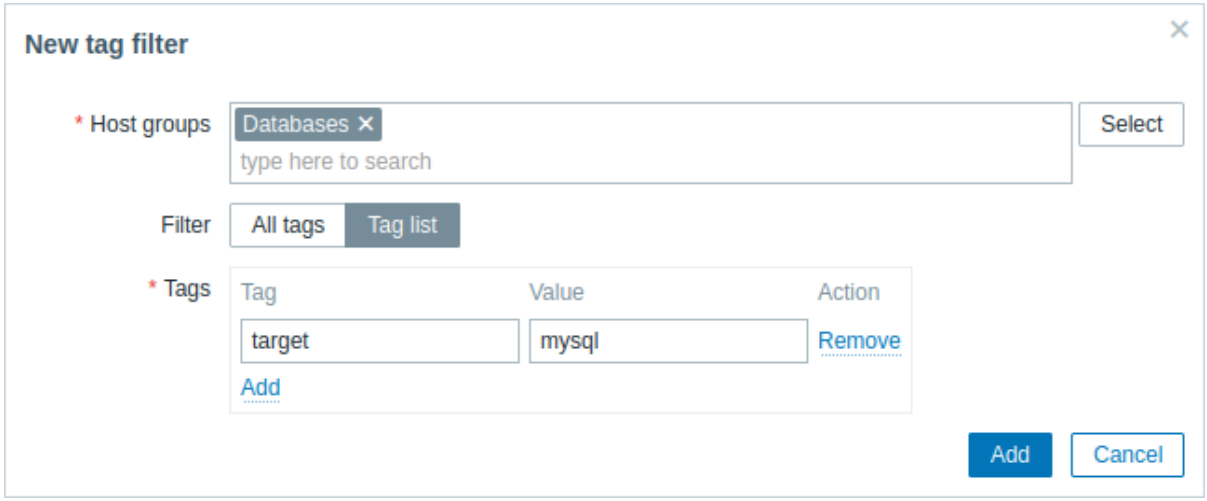
Clique em Add para escolher os grupos de host. Para selecionar um grupo de host para aplicar um filtro de tag, clique em *Selecionar* para obter a lista completa de grupos de host existentes ou comece a digitar o nome de um grupo de host para obter

um menu suspenso de grupos correspondentes. Apenas grupos de host serão exibidos, pois o filtro de tag 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 tag 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 da tag 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 em **Filtro de tag 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, considere 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-escrita* ao Grupo de hosts 1, o usuário terá acesso **Leitura-escrita** a ’X’.

Attention:
As permissões “Leitura-escrita” têm precedência sobre as permissões “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-escrita* ao Grupo de hosts 1.
- Se o Grupo A não tiver permissões definidas e o Grupo B tiver acesso *Leitura-escrita* ao Grupo de hosts 1, o usuário terá acesso **Leitura-escrita** a ’X’.
- Se o Grupo A tiver acesso *Negado* ao Grupo de hosts 1 e o Grupo B tiver acesso *Leitura-escrita* 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 locais.
- 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

Por padrão, 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.

Também é possível configurar para que os valores das macros sejam recuperados pelo Zabbix server e pelo Zabbix proxy de forma independente.

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 o cache das credenciais do banco de dados do Zabbix frontend, o cache é desabilitado por padrão, mas pode ser habilitado definindo a opção `$DB['VAULT_CACHE'] = true` no 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 forem 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:
Os parâmetros de configuração `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` e `VaultPrefix` também são usados para autenticação no vault ao processar macros de vault secreto pelo Zabbix server (e Zabbix proxy, se configurado). O Zabbix server e os proxies não abrirão macros de vault secreto que contenham credenciais de banco de dados do VaultDBPath.

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. Em *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:

- Usuário: <username>
- Senha: <password>

Frontend

Para obter as 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 do segredo 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 do 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:
O Zabbix server (e o Zabbix proxy, se configurado) requerem acesso aos valores de macro *Vault secret* do vault. O Zabbix frontend não precisa 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.

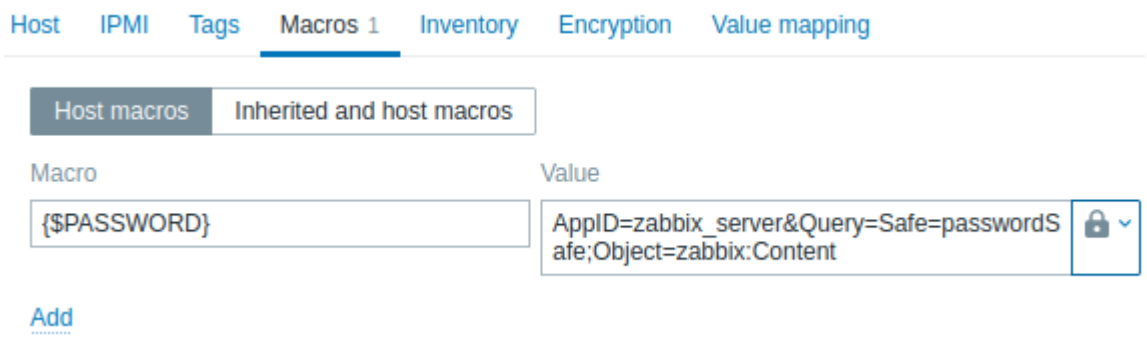
Sintaxe da consulta

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 irá resolver 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 do 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 (consulte 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 forem 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:
Os parâmetros de configuração `Vault`, `VaultToken`, `VaultURL` e `VaultPrefix` também são usados para autenticação no vault ao processar macros de vault secreto pelo Zabbix server (e Zabbix proxy, se configurado). O Zabbix server e os proxies não abrirão macros de segredo do vault que contenham credenciais de banco de dados do VaultDBPath.

É fortemente recomendado usar tokens diferentes para proxies diferentes.

O Zabbix server e o Zabbix proxy leem os parâmetros de configuração relacionados ao vault dos arquivos *zabbix_server.conf* e *zabbix_proxy.conf* durante a inicialização. Além disso, o Zabbix server e o Zabbix proxy lerão a variável de ambiente `VAULT_TOKEN` uma vez durante a inicialização e a removerão para que não fique disponível através de scripts derivados; é um erro se ambos os parâmetros `VaultToken` e `VAULT_TOKEN` contiverem um valor.

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
```

```
#### Por fim, teste com Curl; observe que "data" precisa ser adicionado manualmente após o ponto de montag
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:

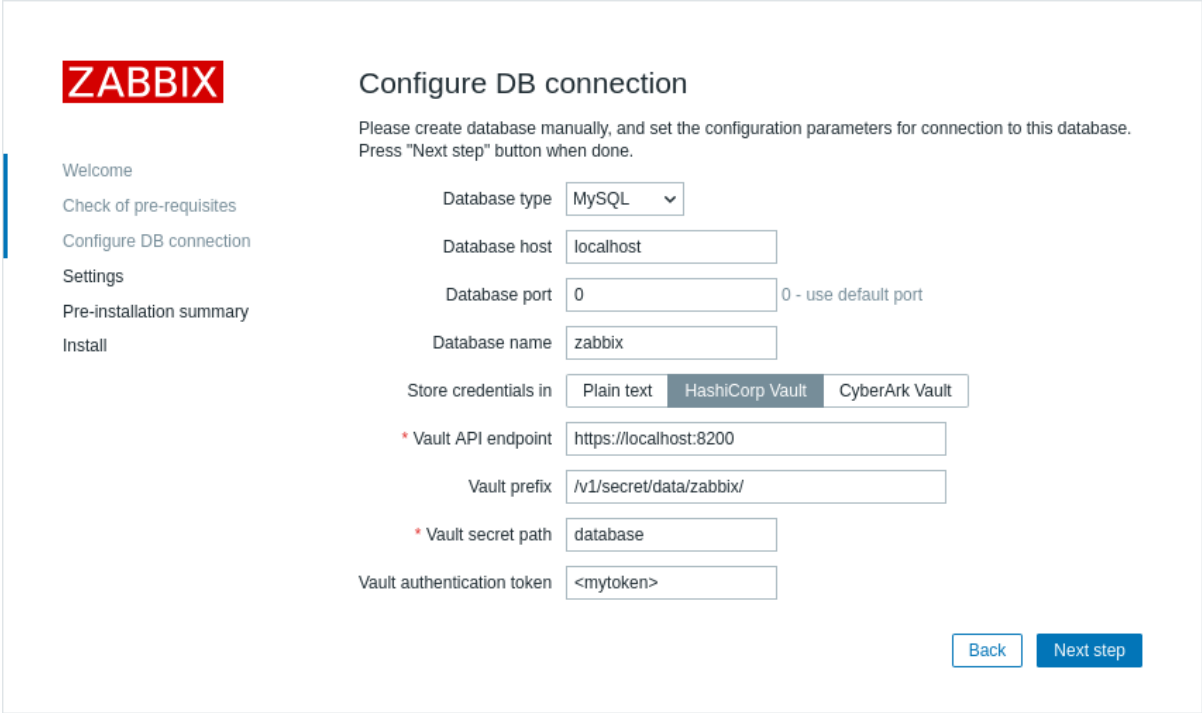
- Username: zabbix
- Password: <password>

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 configuração existente.

Durante a **instalação do frontend**, os parâmetros de configuração devem ser especificados na etapa *Configurar conexão com o BD*:



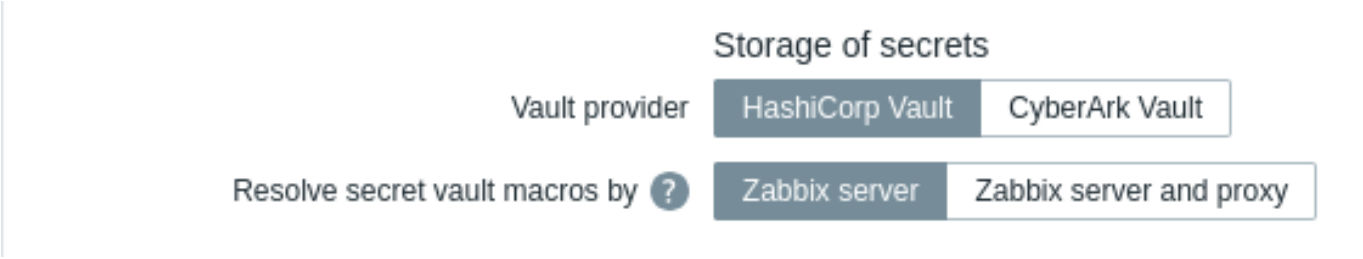
- Defina o parâmetro *Armazenar credenciais em* como ”HashiCorp Vault”.
- Especifique os parâmetros de conexão:

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 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 informações sobre criação de tokens e políticas do vault.

Recuperando valores de macros de usuário

Para usar o HashiCorp Vault para armazenar valores de macro de usuário do tipo *Vault secret*, certifique-se de que:

- O Zabbix server/proxy está configurado para trabalhar com o HashiCorp Vault;
- O parâmetro *Vault provider* em *Administração → Geral → Outros* está definido como "HashiCorp Vault" (padrão);



Note:
O Zabbix server (e o Zabbix proxy, se configurado) requer acesso aos valores de macro *Vault secret* do vault. O Zabbix frontend não precisa 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/proxy (pelo parâmetro `VaultToken`) deve fornecer acesso somente leitura a esse caminho.

Veja *Macros de segredo do Vault* para informações detalhadas sobre o processamento do valor da macro pelo Zabbix.

Sintaxe do caminho

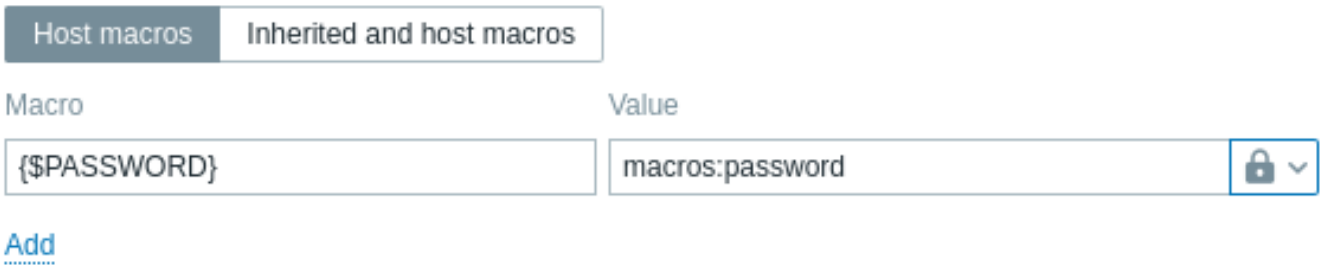
Os símbolos barra ("/") e dois pontos (":") são reservados.

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/';
```

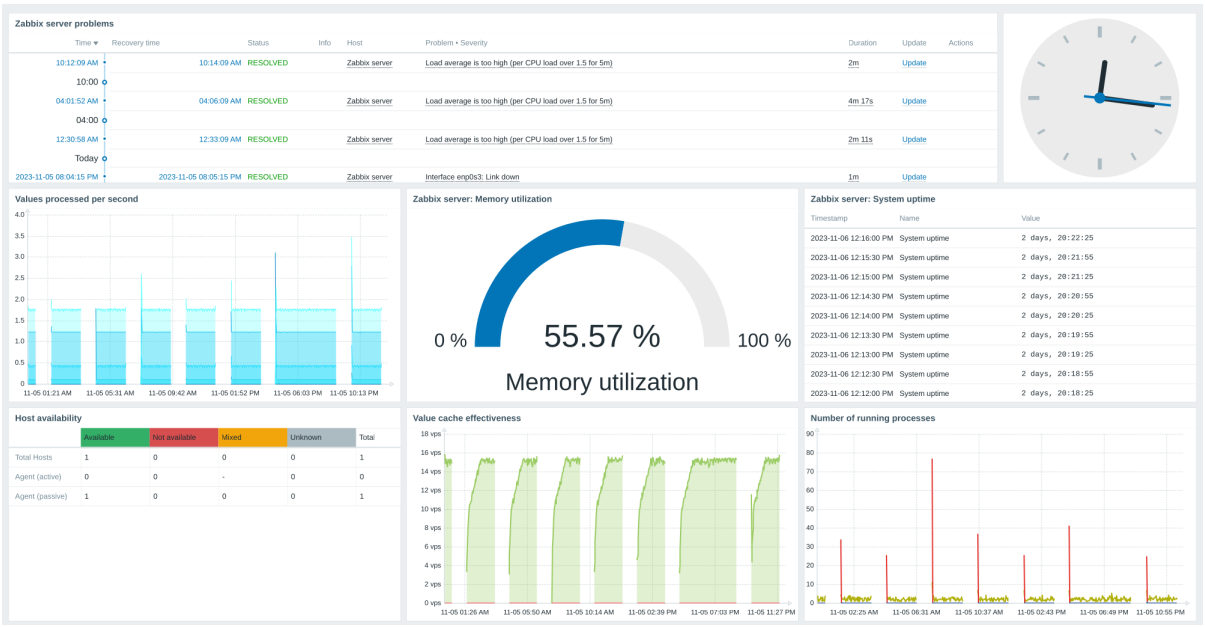
3. Configure macros de usuário conforme descrito na seção *Valores de macro de usuário*, se necessário.

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 - veja 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:

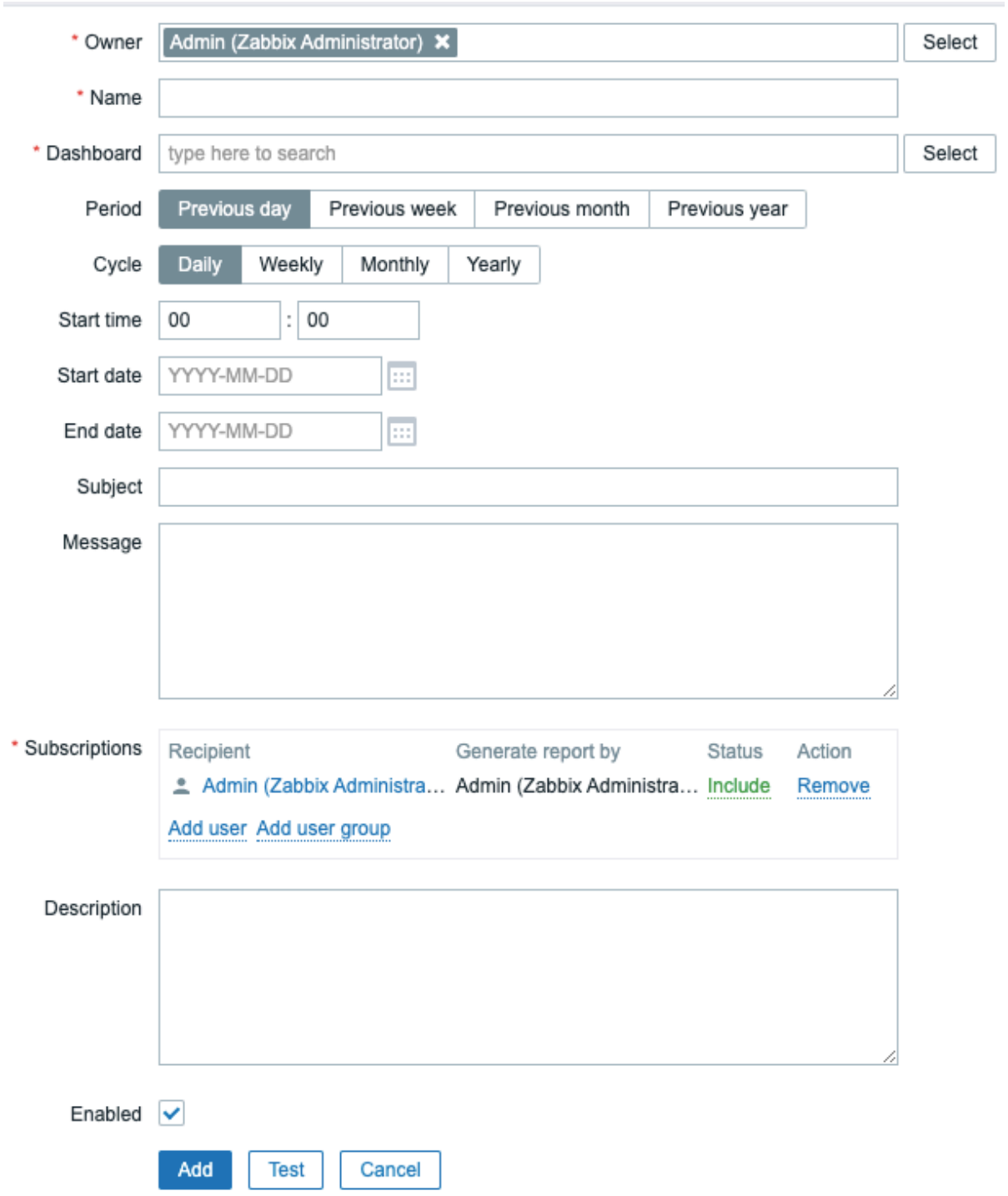
- Vá para: *Relatórios* > *Relatórios agendados*.
- Clique em *Criar relatório* no canto superior direito da tela.

- 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 são 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 será 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.

Parâmetro	Descrição
Período	Período 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.
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. Macros suportadas: {TIME}, {TIMESTAMP}.
Mensagem	Corpo do e-mail do relatório. Macros suportadas: {TIME}, {TIMESTAMP}.
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 para 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.

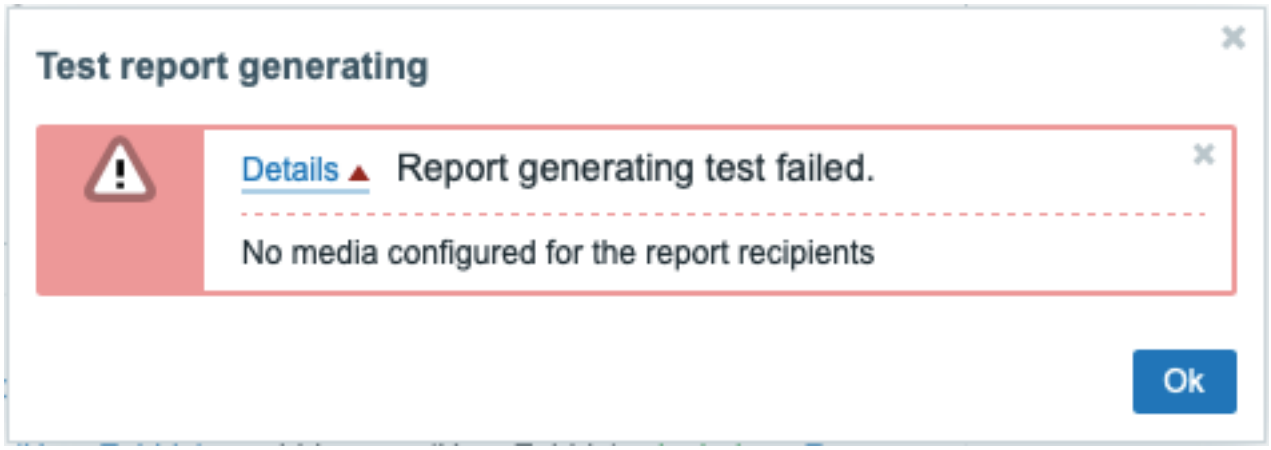
Testando

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.

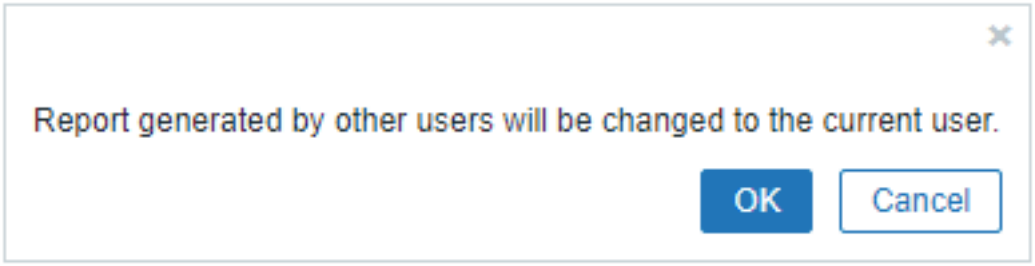
Se a configuração estiver incorreta, uma mensagem de erro é exibida descrevendo a possível causa.



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 de 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 *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 a 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 quebras de 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 dos dados no arquivo de exportação é retomada quando o problema de gravação é resolvido.

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 com a adição de .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 consumir 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 (consulte 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

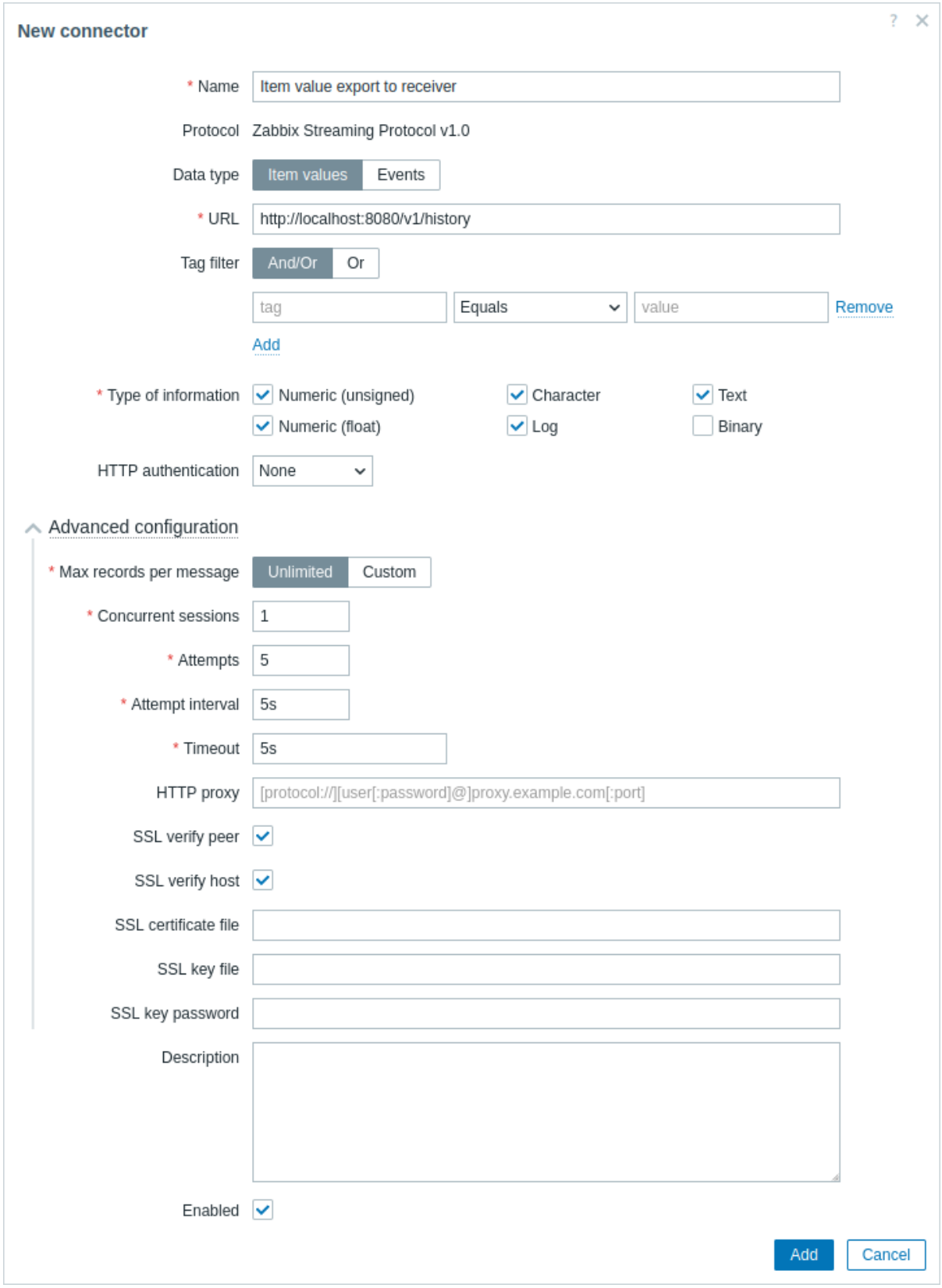
As seguintes etapas são necessárias para configurar o streaming de dados para um sistema externo:

1. Tenha um sistema remoto configurado para receber dados do Zabbix. Para esse 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 items 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` no `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`.



Campos obrigatórios são marcados com um asterisco.

Parâmetro	Descrição
Nome	Informe 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 dos eventos do Zabbix para sistemas externos.
URL	Informe 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 ou excluir tags e valores de tag 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 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; **Não é igual** - 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.
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 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; **Bearer** - autenticação Bearer é usada.
Nome de usuário	Informe o nome de usuário (até 255 caracteres). Macros de usuário são suportadas. Este campo está disponível se a *Autenticação HTTP* estiver definida como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Senha	Informe a senha do usuário (até 255 caracteres). Macros de usuário são suportadas. Este campo está disponível se a *Autenticação HTTP* estiver definida como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Token Bearer	Informe o token Bearer. Macros de usuário são suportadas. Este campo está disponível e é obrigatório se a *Autenticação HTTP* estiver definida 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 quando 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 sobrescreverá as variáveis de ambiente relacionadas ao proxy, como `http_proxy`, `HTTPS_PROXY`. Se não for especificado, o proxy não sobrescreverá as variáveis de ambiente relacionadas ao proxy. O valor inserido é repassado 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 obtido automaticamente do local da autoridade de certificação (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	Informe a descrição do conector.
Habilitado	Marque a caixa de seleção para habilitar o conector.

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 JSON delimitado por nova linha.

Solicitação do servidor

Exemplo de transmissão de valores de item:

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

Exemplo de transmissão de eventos:

```
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 SNMP gateway

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 de grupo de host (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 quaisquer dados 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 de grupo de host.

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 sob `OID=1.3.6.1.4.1.3043.7.55.X.3`. ”X” aqui será o número do campo 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 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 gateway SNMP:

- por severidade (por padrão `ProblemMinSeverity=-1`)
- ocultando problemas reconhecidos (por padrão `ProblemHideAck=false`)

Você pode limitar a contagem de problemas por grupo de hosts na configuração do gateway SNMP:

- por triggers em estado desconhecido (por padrão `CountUnknown=-false`)
- por triggers com problemas reconhecidos/não reconhecidos/todos (por padrão `CountAcknowledgeStatus=all`)

8 Monitoramento de serviços

Visão geral O monitoramento de serviços é um monitoramento em nível de negócio 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 superior. 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 Zabbix server 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 Zabbix server 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

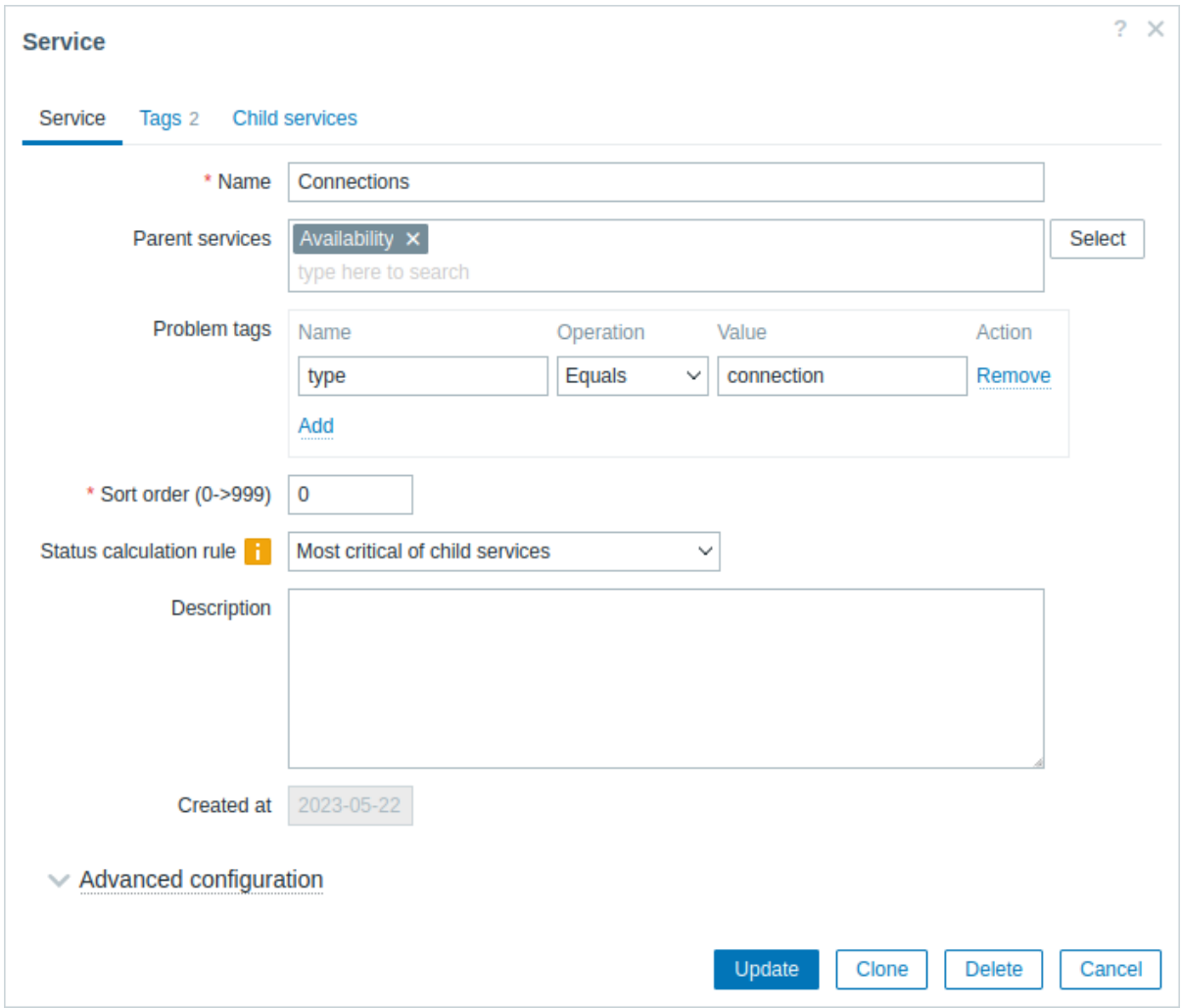
1 Árvore de serviços

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.

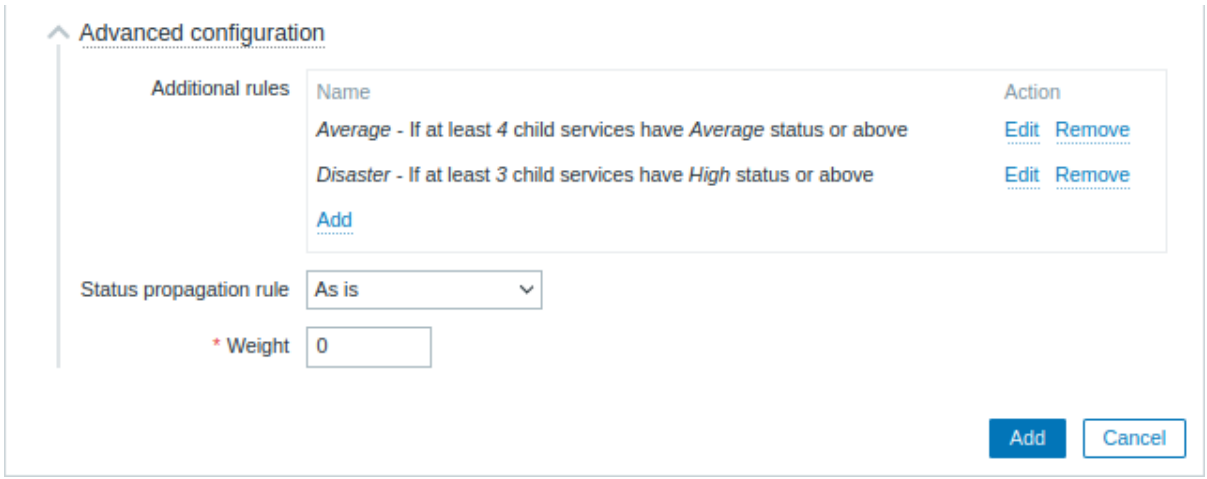
Configuração do serviço Na aba **Serviço**, especifique os parâmetros necessários do serviço:



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. Nesse 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 tag especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tag 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 severidade será definida.

Parâmetro	Descrição
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 severidade **Diminuir em** - você pode diminuir o status propagado em 1 a 5 níveis de severidade **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 severidade 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 deveria 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).

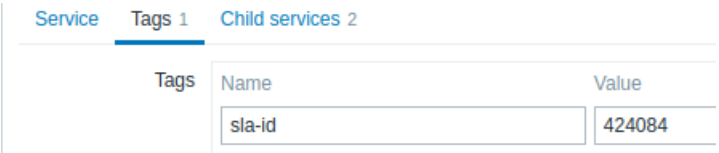
A aba **Serviços filhos** permite especificar serviços dependentes. Clique em *Adicionar* para adicionar um serviço da lista de serviços existentes. Se você quiser adicionar um novo serviço filho, salve este serviço primeiro, depois clique no ícone de mais ao lado do serviço que você acabou de criar.

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 com 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*.

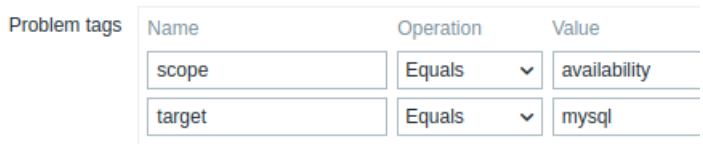


Tags de problema

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 coincidirem, 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 esse 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.

Modificando serviços existentes

> **Note:**
> Os botões descritos abaixo são visíveis apenas quando a seção *Serviços* está no modo de Edição.

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 subirão um nível na árvore de serviços (os filhos de 1º nível terão o mesmo nível que o 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 o desempenho deles está de acordo com o acordo de nível de serviço (SLA).

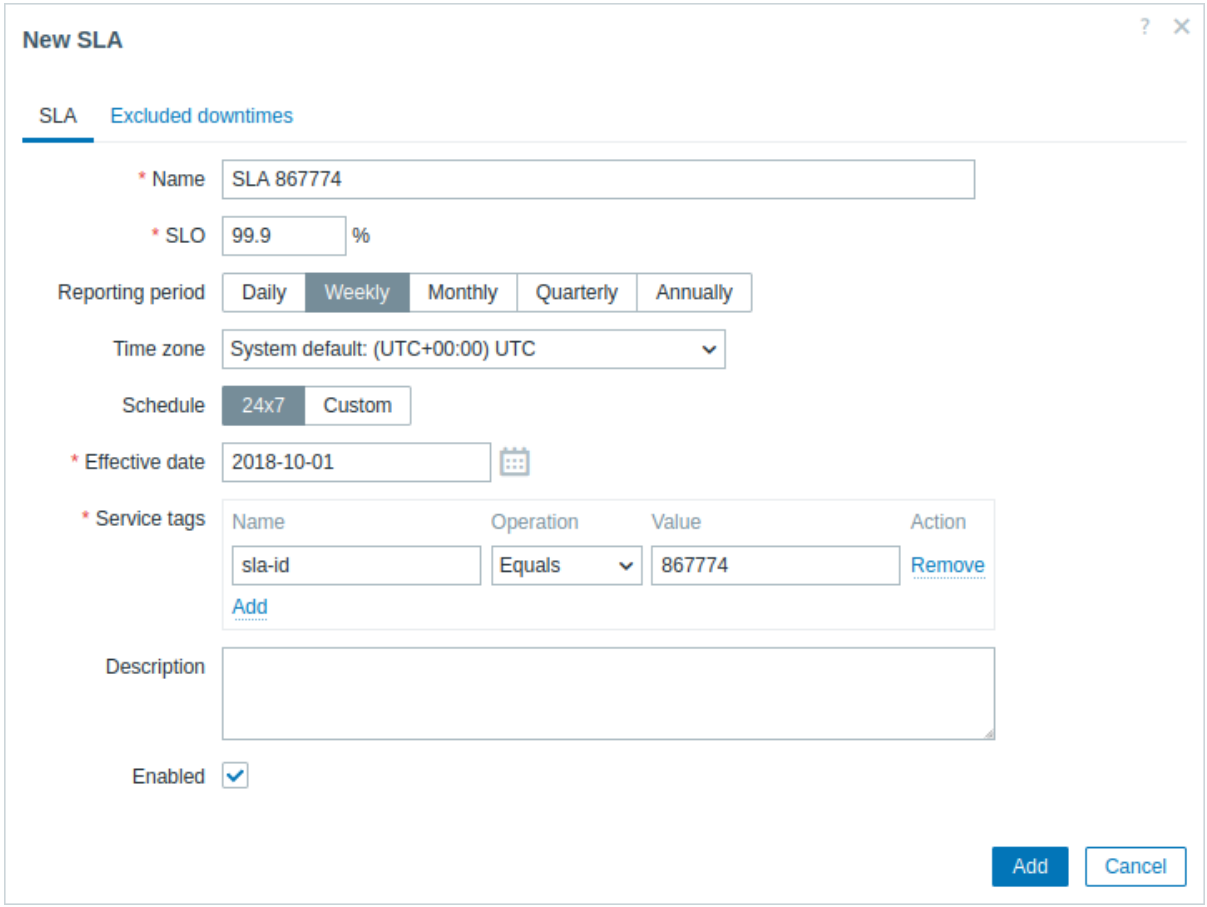
A seção de menu *Serviços->SLA* permite configurar SLAs para vários serviços. Um SLA no Zabbix define o objetivo de nível de serviço (SLO), a programação de tempo de atividade esperada e os períodos de inatividade planejados.

SLAs e serviços são associados 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 SLA 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 %).

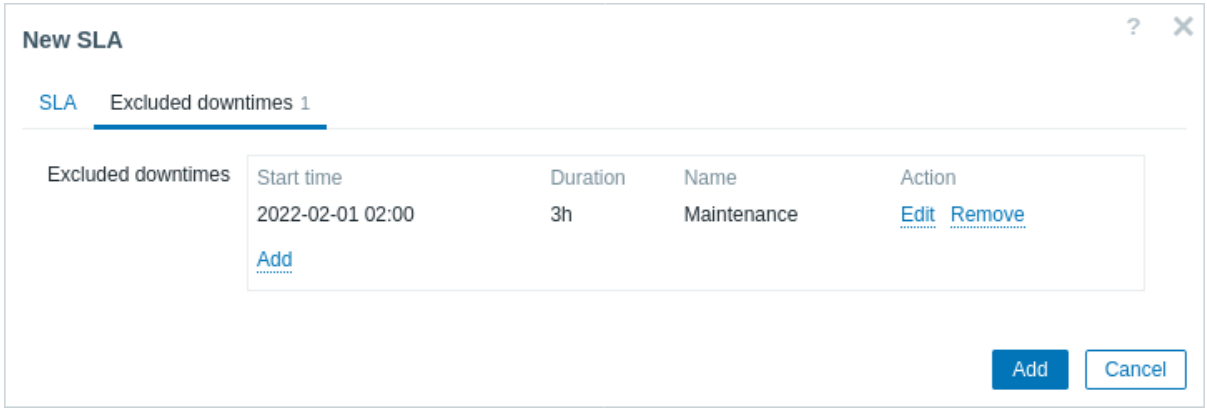
Configuração Para criar um novo SLA, clique no botão *Criar SLA*.

A aba **SLA** permite especificar os parâmetros gerais do SLA.



Parâmetro	Descrição
Nome	Digite o nome do SLA.
SLO	Digite o objetivo de nível de serviço (SLO) como percentual.
Período de relatório	Selecionar o período afetará quais períodos são usados no relatório de SLA - *diário*, *semanal*, *mensal*, *trimestral* ou *anual*.
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.
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 digite o nome do período, data de início e 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 aba de informações;
- no widget do Dashboard *Relatório de SLA*.

Uma vez que um SLA esteja configurado, a aba *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.

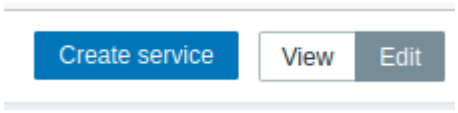
Pré-requisitos Antes de configurar o monitoramento de serviços, você precisa ter os hosts configurados:

- *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`

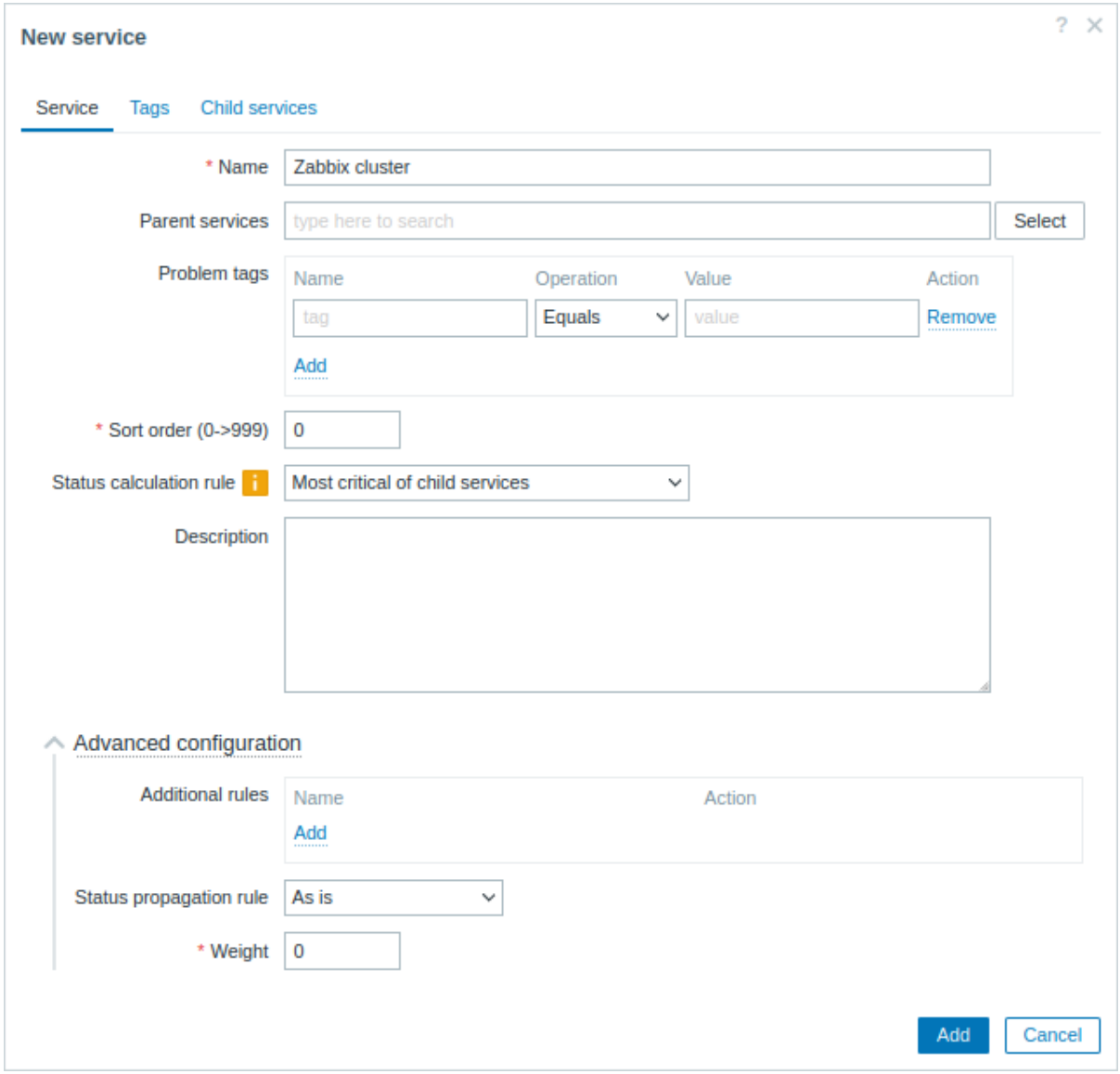
Árvore de serviços O próximo passo é construir a árvore de serviços. Neste exemplo, a infraestrutura é muito básica e consiste em três serviços: *Zabbix cluster* (pai) e dois serviços filhos *Zabbix server node 1* e *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

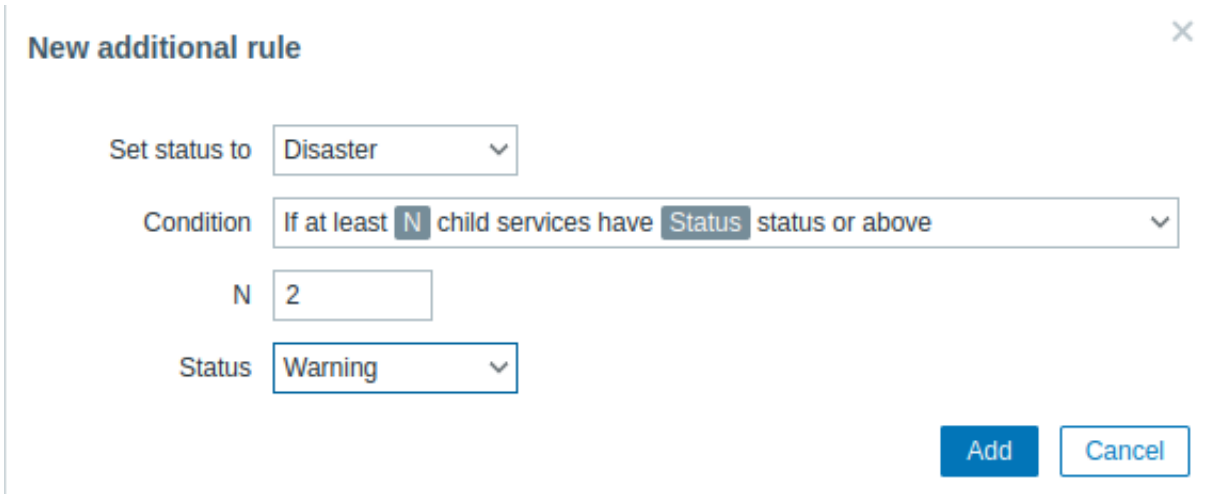
Na página Serviços, ative o modo *Editar* e pressione Criar serviço:



Na janela de configuração do serviço, insira o nome *Zabbix cluster* e clique no rótulo *Configuração avançada* para exibir as opções avançadas de configuração.



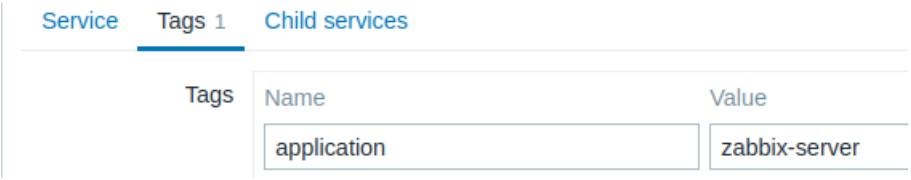
Configure uma regra adicional:



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:

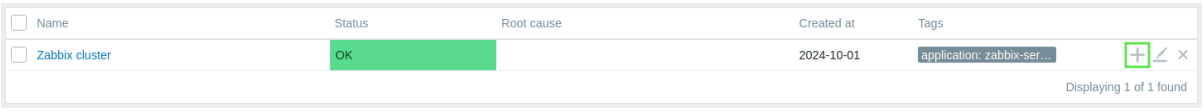
- Definir status como: Desastre
- Condição: Se pelo menos `N` serviços filhos tiverem status `Status` ou superior
- N: 2
- Status: Aviso

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.

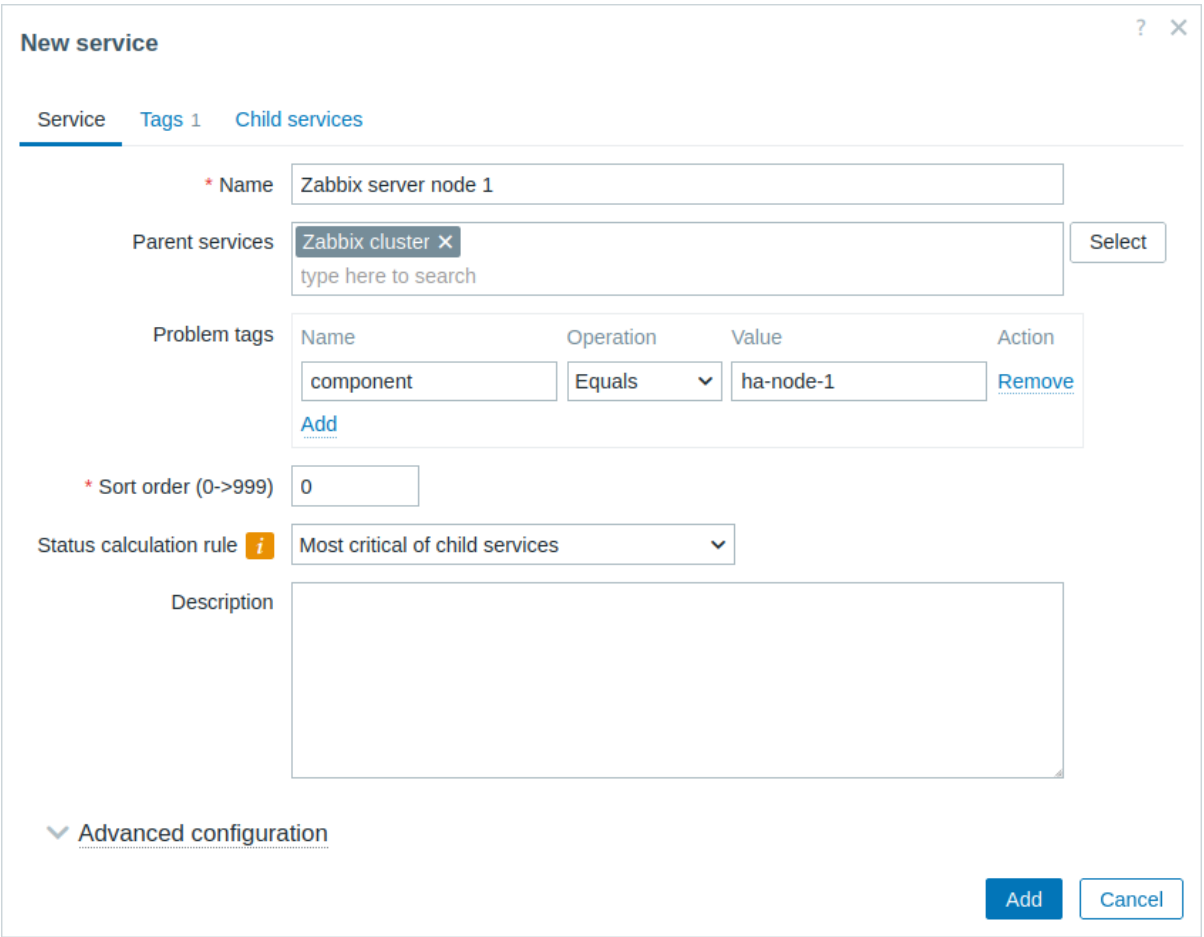
Para adicionar um serviço filho, pressione o ícone de mais ao lado do serviço do cluster Zabbix (o ícone é visível apenas no modo *Editar*).



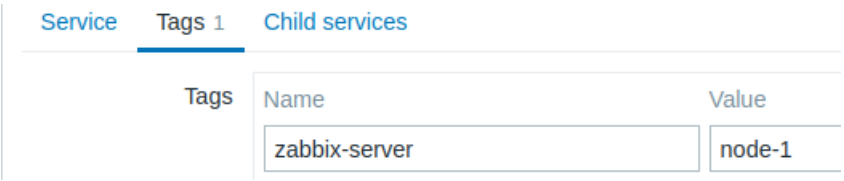
Na janela de configuração do serviço, insira o nome *Zabbix server node 1*. Observe que o parâmetro Serviços pai já está preenchido com *Zabbix cluster*.

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 aba *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 aba *Tags* e adicione uma tag de serviço: `zabbix-server:node-2`.

Salve o novo serviço.

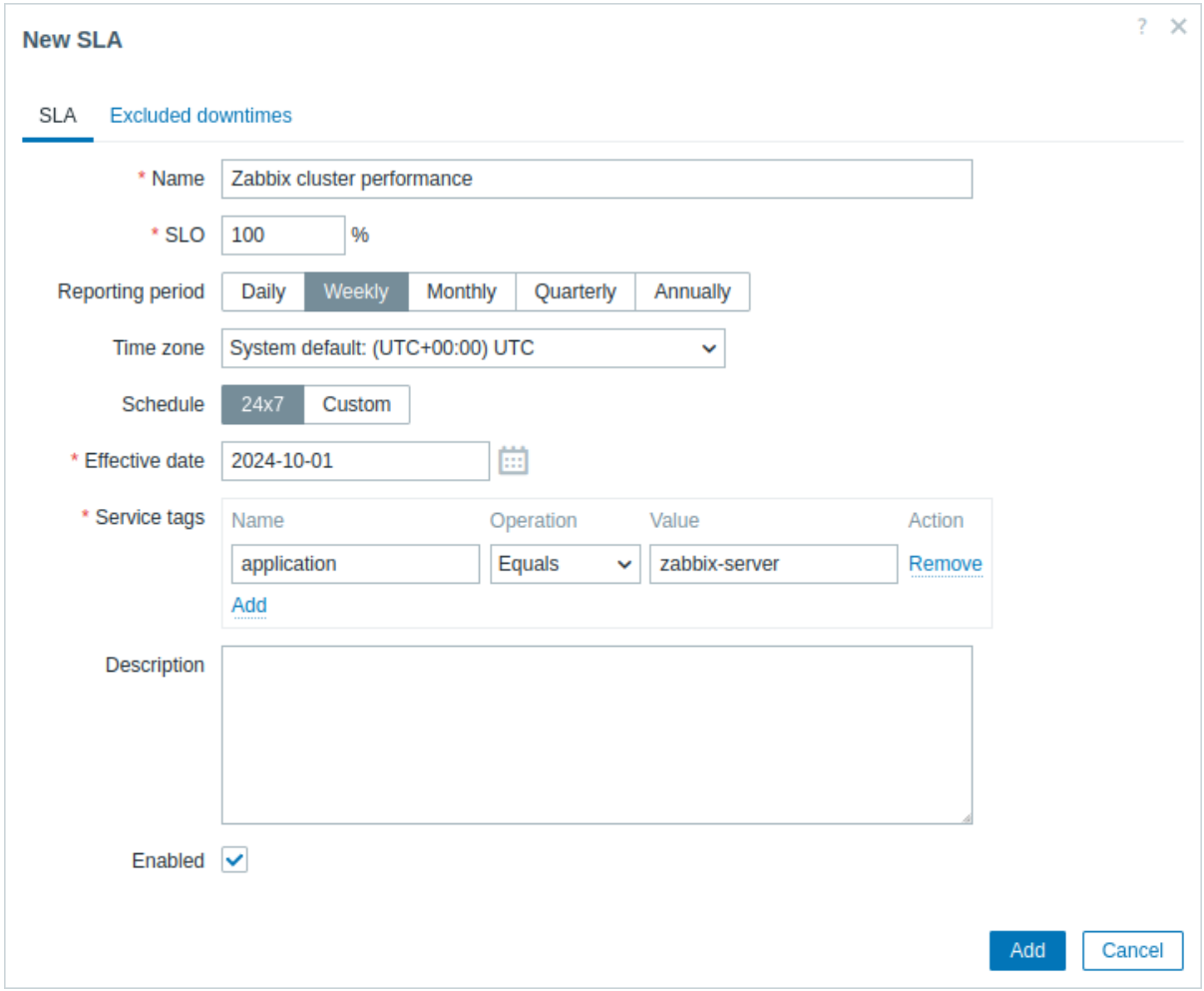
SLA Neste exemplo, o desempenho esperado do cluster Zabbix é de 100%, excluindo o período de manutenção semestral de uma hora.

Primeiro, é necessário 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%.

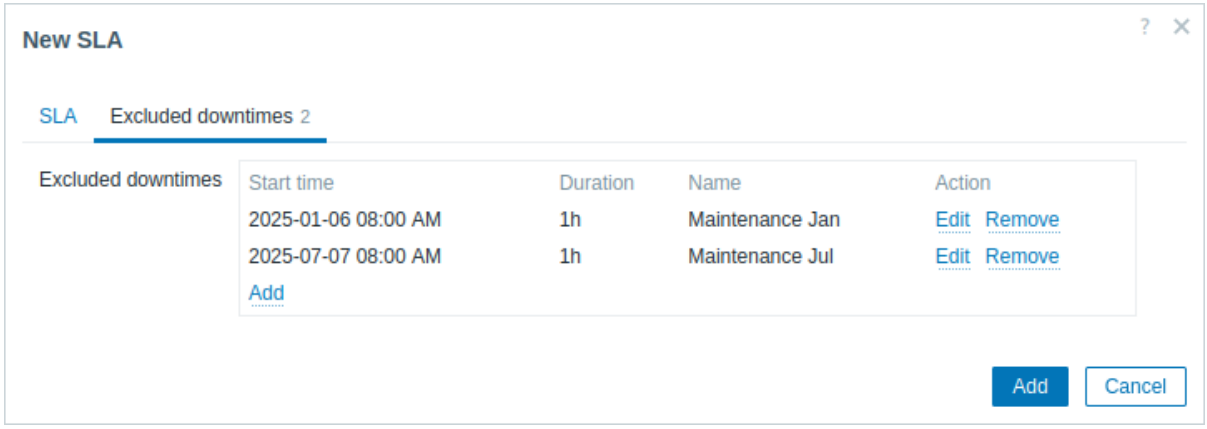
O serviço do cluster Zabbix possui uma tag de serviço `application:zabbix-server`. Para usar este SLA para medir o desempenho do cluster Zabbix, no parâmetro *Tags de serviço*, especifique:

- Nome: application
- Operação: Igual
- Valor: zabbix-server



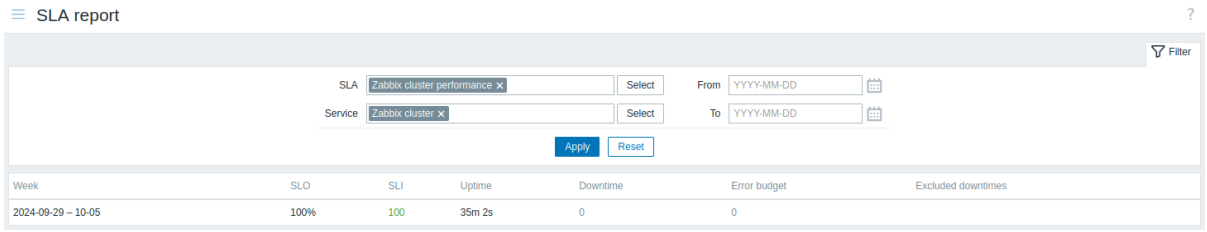
Em um ambiente real, você também pode atualizar o período de relatório desejado, o fuso horário e a data de início ou alterar a programação de 24/7 para personalizada. Para este exemplo, as configurações padrão são suficientes.

Altere para a guia *Períodos de inatividade excluídos* e adicione períodos de inatividade para períodos de manutenção programada 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.

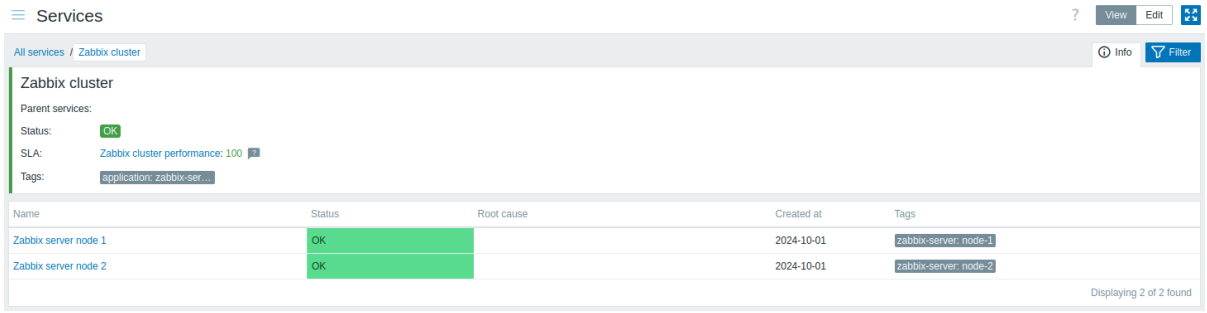


Pressione Adicionar para salvar o novo SLA.

Altere para a seção de relatórios de SLA para visualizar o relatório de SLA do 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 predefinida. 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 de todo o cenário
- 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 predefinida. 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 irá, opcionalmente, seguir redirecionamentos (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. Podem ser especificadas até 10 tentativas, o valor padrão é 1. *Observação*: o Zabbix não repetirá uma etapa devido a um código de resposta incorreto 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 for 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) 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 quando usadas em post bruto ou diretamente na URL.
Cabeçalhos	Cabeçalhos HTTP são usados ao executar 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 no nível da etapa. **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, mesmo o cabeçalho padrão 'User-Agent' pode ser sobrescrito ao especificá-lo no nível da 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.

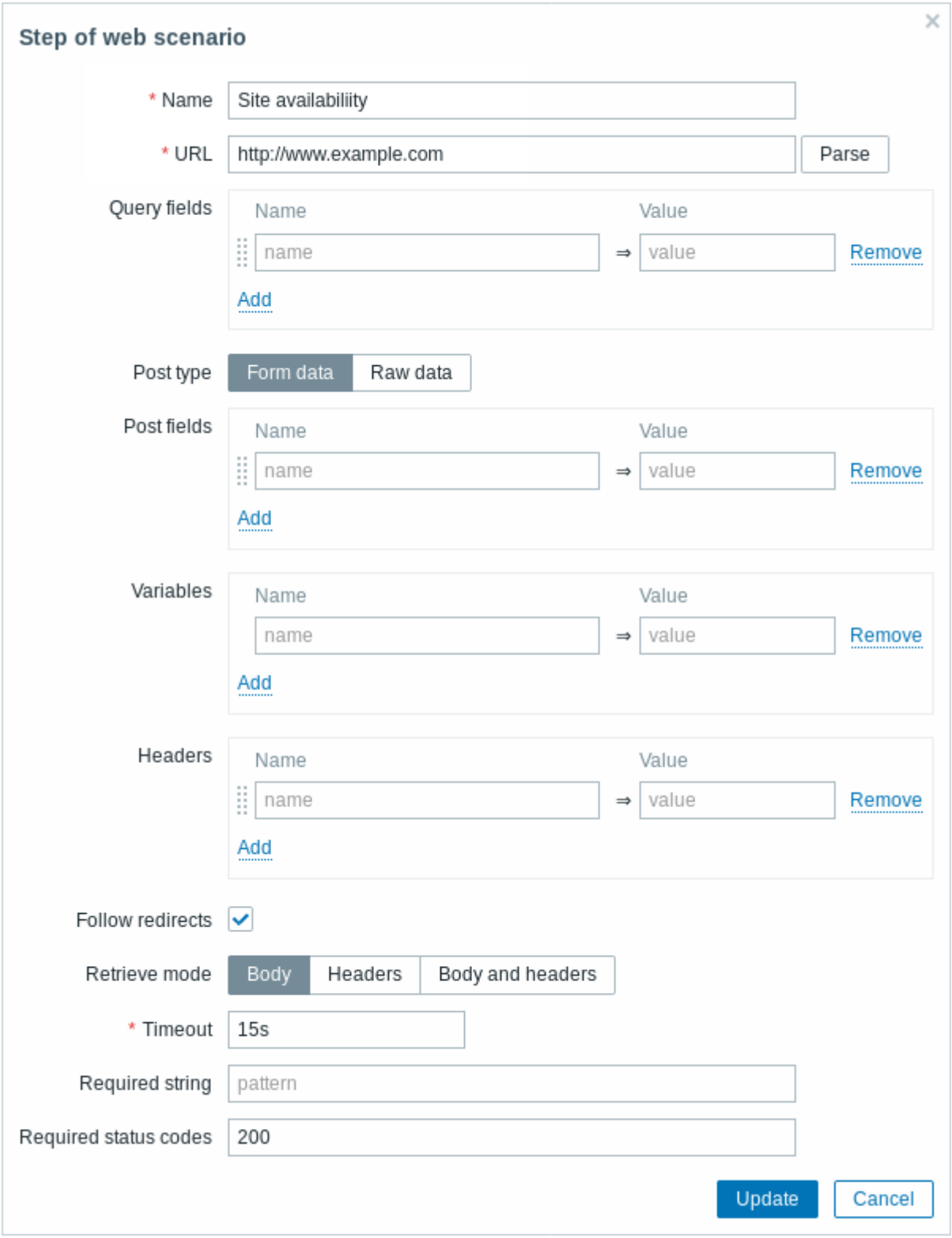
Observe que ao editar um cenário existente, dois botões extras estão disponíveis no formulário:

Clone	Cria outro cenário com base nas propriedades do existente.
Clear history and trends	Exclui os dados de histórico e 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 as variáveis de ambiente relacionadas ao proxy.
Para verificações HTTP - defina a variável de ambiente **http_proxy** para o usuário do Zabbix server. 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 exclusivo 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 (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 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. Valores de variáveis do cenário, macros de usuário ou macros {HOST.*} são resolvidos e então codificados automaticamente. 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. Valores de variáveis do cenário, macros de usuário ou macros {HOST.*} são resolvidos e então codificados automaticamente. No modo **Dados brutos**, atributos/valores são exibidos em uma única linha e concatenados com o símbolo **&**. Valores brutos podem ser codificados/decodificados manualmente 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 desejar codificar a variável, 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 codificadas automaticamente quando usadas em campos de consulta ou dados de formulário para variáveis post, mas devem ser codificadas manualmente 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 apenas para aquela etapa. **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 sobrescrito ao especificá-lo no nível da etapa. Por exemplo, atribuir o nome a um cabeçalho, mas não definir valor, irá desabilitar 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 para processar a 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.

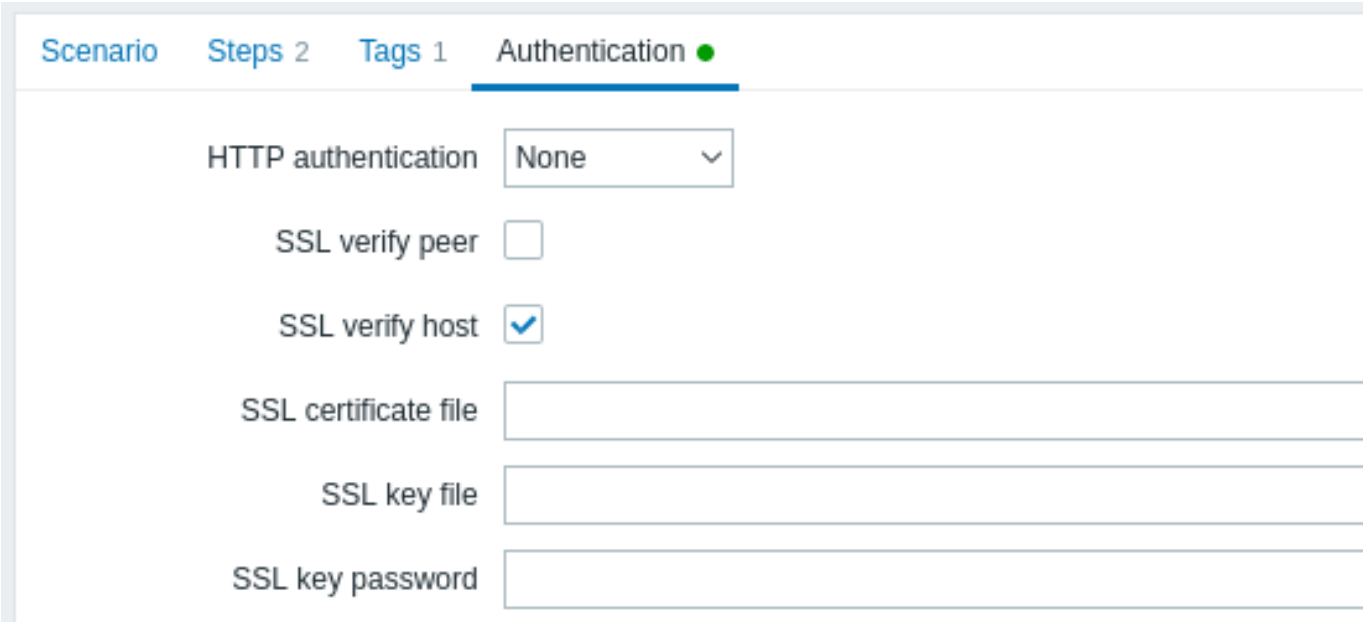
Note:
Quaisquer alterações nas etapas do cenário web só serão salvas quando todo o cenário for salvo.

Veja também um exemplo real de como as etapas de monitoramento web podem ser configuradas.

Configurando tags A aba **Tags** permite definir tags no nível do cenário.

A marcação permite filtrar cenários web e items 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 para verificar o certificado SSL do servidor web. O certificado do servidor será obtido automaticamente 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 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 arquivos de certificado e chave privada apenas no formato PEM. Caso você tenha seu certificado e chave privada em um arquivo no formato PKCS #12 (geralmente 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 aplica alterações em certificados sem 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 items são adicionados automaticamente para monitoramento quando cenários web são criados.

Todos os items herdam tags 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 é 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 colchetes, 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 trigger ”Web scenario failed”, você pode definir uma expressão de trigger:

```
last(/host/web.test.fail[Scenario])<>0
```

Certifique-se de substituir ’Scenario’ pelo nome real do seu cenário.

Exemplo 2

Para criar um trigger ”Cenário web falhou” com uma descrição útil do problema na trigger, você pode definir um trigger com o nome:

```
O cenário web "Scenario" falhou: {ITEM.VALUE}
```

e expressão do trigger:

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

Certifique-se de substituir ’Scenario’ pelo nome real do seu cenário.

Exemplo 3

Para criar um trigger ”Aplicação web está lenta”, você pode definir uma expressão de trigger:

```
last(/host/web.test.in[Scenario,,bps])<10000
```

Certifique-se de substituir ’Scenario’ pelo nome real do seu cenário.

Items de etapa de cenário

Assim que uma etapa é criada, o Zabbix adiciona automaticamente os seguintes items para monitoramento.

Item	Descrição
Velocidade de download para a etapa <Step> do cenário <Scenario>	Este item irá coletar informações sobre a velocidade de download (bytes por segundo) da etapa. Chave do item: web.test.in[Scenario,Step,bps] Tipo: *Numérico(flutuante)*
Tempo de resposta para a etapa <Step> do cenário <Scenario>	Este item irá coletar informações sobre o tempo de resposta da etapa em segundos. O tempo de resposta é contado desde o início da requisição até que todas as informações tenham sido transferidas. Chave do item: web.test.time[Scenario,Step,resp] Tipo: *Numérico(flutuante)*
Código de resposta para a etapa <Step> do cenário <Scenario>	Este item irá coletar os códigos de resposta da etapa. Chave do item: web.test.rspcode[Scenario,Step] Tipo: *Numérico(não assinado)*

Os nomes reais do cenário e da etapa serão usados em vez de ”Scenario” e ”Step”, respectivamente.

Note:
Os items de monitoramento web são adicionados com 30 dias de histórico e 90 dias de retenção de tendências.

Note:
Se o nome do cenário começar com aspas duplas ou contiver vírgula ou colchete, ele será devidamente citado nas chaves dos items. Nos outros casos, nenhuma citação adicional será realizada.

Esses items podem ser usados para criar triggers e definir condições de notificação. Por exemplo, para criar um trigger ”Login na interface web do Zabbix está muito lento”, você pode definir uma expressão de trigger:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 Cenário da vida real

Visão geral

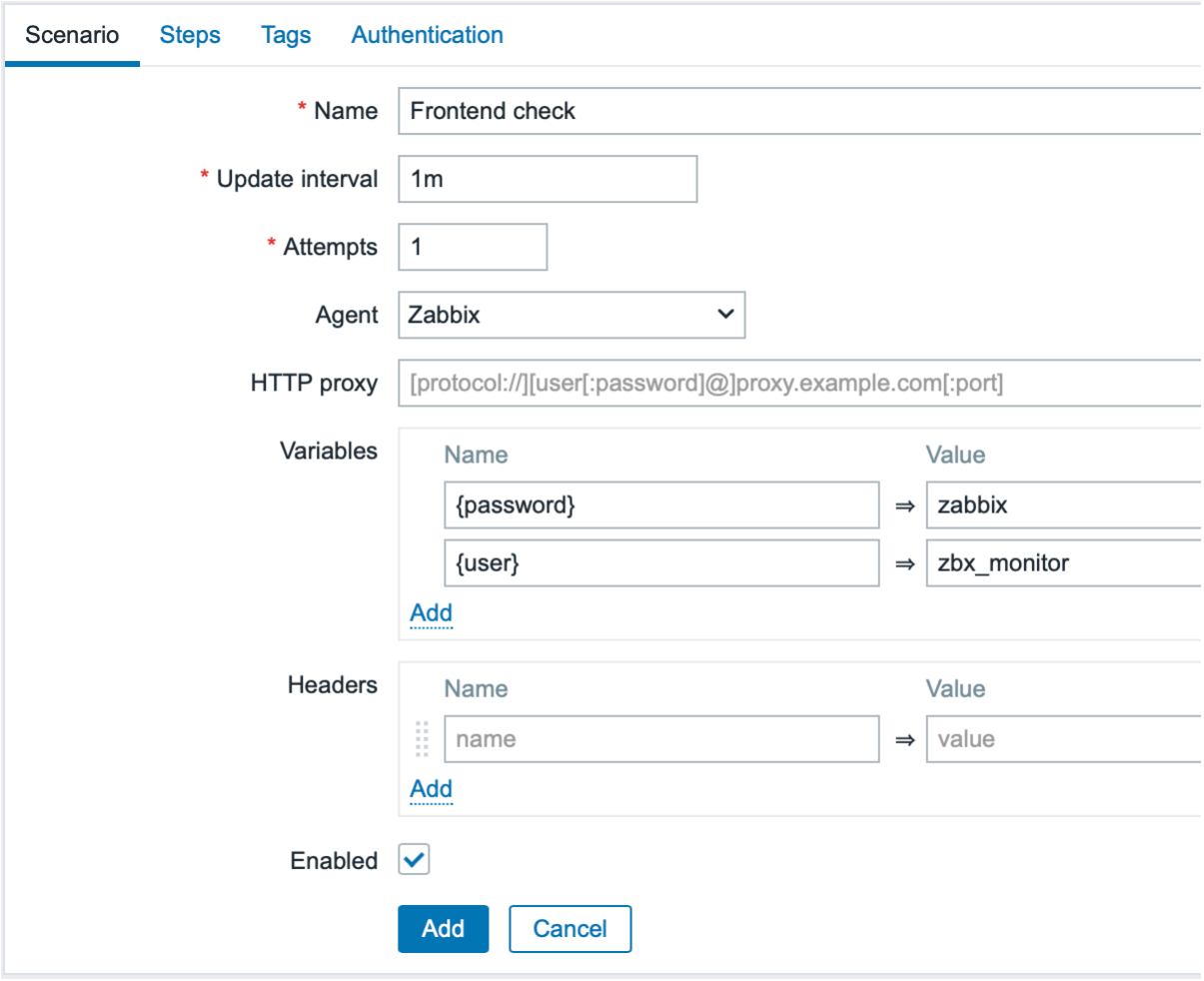
Esta seção apresenta um exemplo passo a passo de como o monitoramento web pode ser usado.

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, várias etapas são necessárias, 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 questões 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 trapper Zabbix ao host. Você pode usar tags de 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

Vá para a aba *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

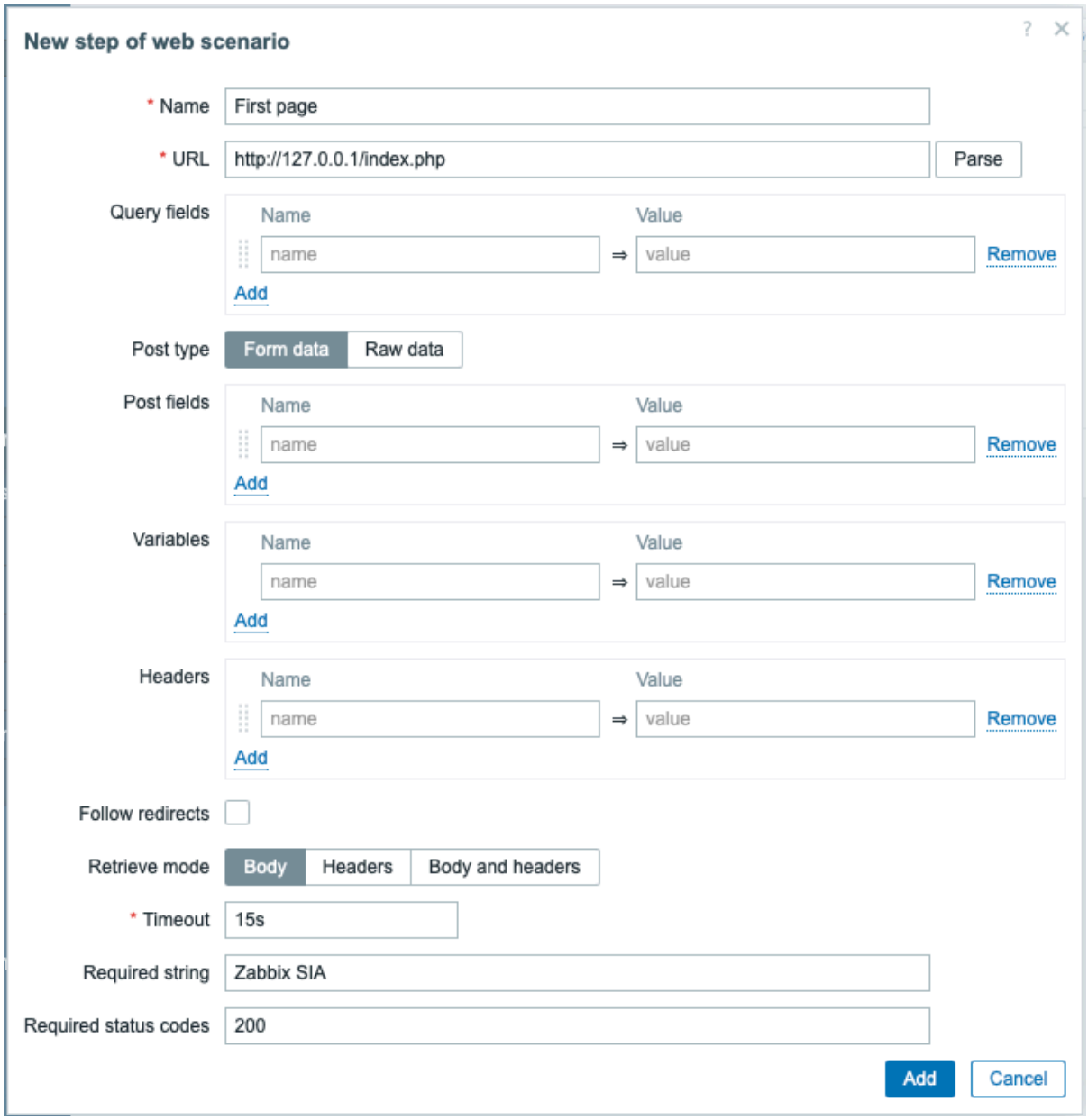
Etapa 1 do cenário da Web

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*.

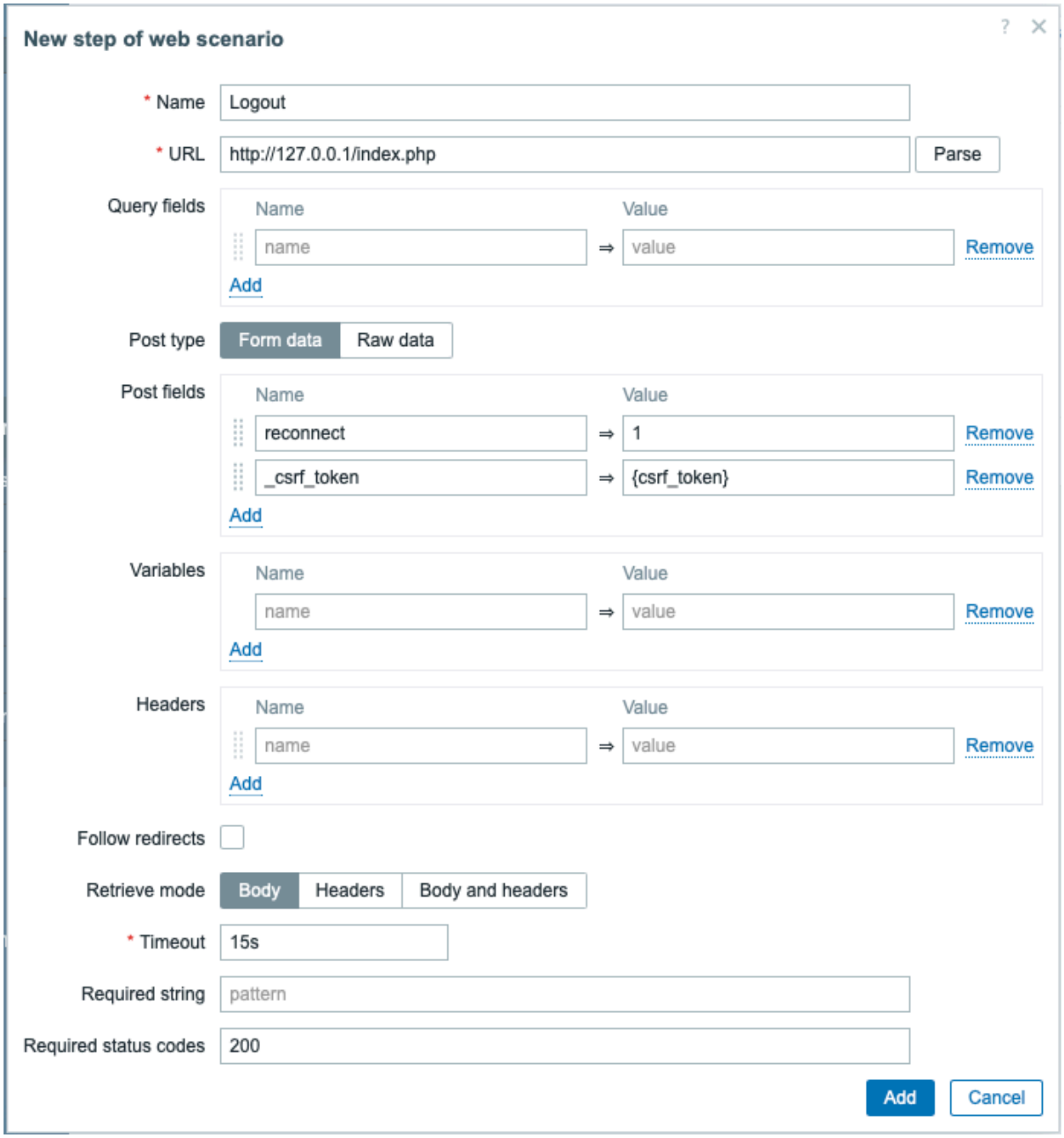
- No campo **Nome**, insira *Verificação de login*.
- No campo **String obrigatória**, insira *Administração*.
- Preencha os campos comuns.

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 envio**, adicione dois campos de envio:
  - *reconnect* com valor *1*
  - *_csrf_token* com valor *{csrf_token}*.
- Preencha os campos comuns.

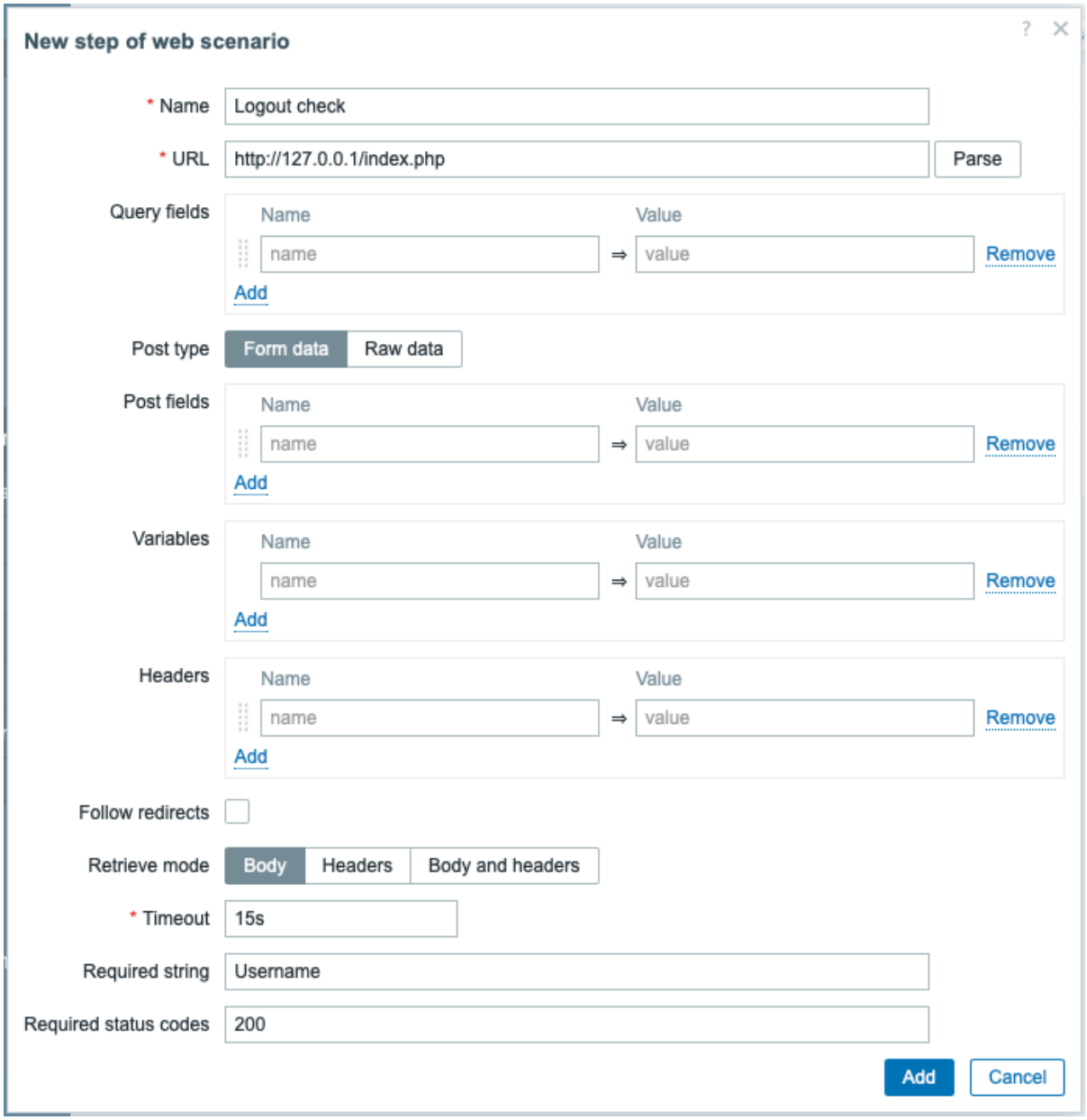
Esta etapa usa a variável {csrf_token} obtida na etapa 2



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 de um cenário web deve ser semelhante a esta:

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati…	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

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 máquinas virtuais

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 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 servidor Zabbix.
2. Os processos *poller* do Zabbix recuperam os dados usando as chaves de item de verificação simples do Zabbix VMware monitoring item keys.

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
- 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 *VMware monitoring item keys*.

Observe que a frequência de recuperação de eventos do VMware depende do intervalo de polling de 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 das estatísticas dos contadores de desempenho do VMware pode ser atrasada pela obtenção dos 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 do 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 da 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 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 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 limites impostos 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 Profiling excessive memory usage with tcmalloc.

1 Chaves de item de monitoramento do VMware

Visão geral Esta página fornece detalhes sobre as verificações simples que podem ser usadas para monitorar ambientes VMware. As métricas são agrupadas pelo alvo de 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 de contadores 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

Chave do item	Descrição	Grupo de item
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 escrita 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.fetchports.get	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 de 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.	
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 escrita 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 do hipervisor VMware.	
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 estado de saúde 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.	

Chave do item	Descrição	Grupo de item
vmware.rp.cpu.usage	O uso de 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 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 que a máquina virtual estava pronta, mas não conseguiu ser agendada para execução na CPU física.	
vmware.vm.cpu.ready	O tempo que a máquina virtual estava pronta, mas não conseguiu ser agendada para execução na CPU física.	
vmware.vm.cpu.swapwait	A porcentagem do tempo de CPU gasto esperando por 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.	
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.maintenance	O status de manutenção do hipervisor da máquina virtual VMware.	
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 de 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 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.	

Chave do item	Descrição	Grupo de item
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.

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** - a 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, 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.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 do VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore do 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 hipervisores 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 do VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore do VMware;
- **prop** - o caminho da propriedade.

vmware.datastore.read[url,datastore,<mode>]

 O tempo gasto 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 do VMware.

Este item funciona com o vSphere 6.5 e mais recente.

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 de alarmes do hipervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a 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 global exclusivo do hipervisor VMware.

vmware.hv.connectionstate[url,uuid]

 O estado de conexão do hypervisor VMware.
 Valor de retorno: *String*: *connected*, *disconnected* ou *notResponding*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.cpu.usage[url,uuid]

 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 global exclusivo 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 global exclusivo do hypervisor VMware.

vmware.hv.datastore.discovery[url,uuid]

 A descoberta de datastores 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.datastore.list[url,uuid]

 A lista de datastores do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a 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 de datastore disponíveis.
 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 hypervisor 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>]

 A quantidade média de tempo para 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 hipervisor VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (padrão).

vmware.hv.discoveryurl

 A descoberta de hypervisors 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** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hypervisor VMware.

vmware.hv.fullname[url,uuid]

 O nome 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.freq[url,uuid]

 A frequência 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.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: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global 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: *Inteiro*.

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 recentes.

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 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.maintenance[url,uuid]

 O status de manutenção do hipervisor 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 hipervisor 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]

 Tamanho da memória usada pelo hypervisor VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a 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 hypervisor VMware.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.network.in[url,uuid,<mode>]

 Estatísticas de entrada 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.network.linkspeed[url,uuid,ifname]

 A velocidade da interface de rede do hypervisor VMware.
 Valor de retorno: *Integer*. Retorna *0* se a interface de rede estiver inativa, caso contrário, retorna o valor da velocidade da interface.

Parâmetros:

- **url** - a 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** - a 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 global exclusivo do hypervisor VMware;
- **max** - o uso máximo de energia permitido.

vmware.hv.sensor.health.state[url,uuid]

 O sensor de resumo do estado de integridade do hipervisor VMware.
 Valor de retorno: *Inteiro*: *0* - cinza; *1* - verde; *2* - amarelo; *3* - vermelho.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware.

Observe que o item pode não funcionar no VMware vSphere 6.5 e mais recentes, porque a VMware descontinuou o sensor *VMware Rollup Health State*.

vmware.hv.sensors.get[url,uuid]

 Os sensores de estado do fornecedor de hardware do hipervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor 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** - 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.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: *Integer*.

Parâmetros:

- **url** - a 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 do VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **rpid** - o ID do pool de recursos do 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** - a 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 global exclusivo 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;
- **name** - o nome do atributo personalizado.

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** - a 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 consegue ser executada porque está competindo pelo acesso à(s) CPU(s) física(s).
 Valor de retorno: *Float*.

Parâmetros:

- **url** - a 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: *Inteiro*.

Parâmetros:

- **url** - o 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 pôde 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 suas cargas de CPU (%).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;

- **uuid** - o identificador global único da máquina virtual VMware.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 A porcentagem do tempo de CPU gasto esperando por swap-in.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - a 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: *Inteiro*.

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: *Inteiro*.

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** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.hv.maintenance[url,uuid]

 O status de manutenção do hypervisor da máquina virtual VMware.
 Valor de retorno: *0* - não está em manutenção; *1* - está em manutenção.

Parâmetros:

- **url** - o 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** - a 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** - o URL do serviço VMware;
- **uuid** - o identificador exclusivo global 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** - o 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** - o URL do serviço VMware;
- **uuid** - o identificador exclusivo global 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** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo 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** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 Uso da memória 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: *Inteiro*.

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 das interfaces de rede 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.

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** - 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.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: *Inteiro*.

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** - o 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.

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: *Inteiro*.

Parâmetros:

- **url** - o 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: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo 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 de convidado 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>]

 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 de sistemas de arquivos 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.

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 VMware do Zabbix:

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in
- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

Mais informações

Consulte Monitoramento de máquinas virtuais para obter 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.

Chave do item		

	”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", veja 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.

	”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.

Chave do item		
	{#DVS.UUID}	Identificador do switch.
vmware.hv.discovery		
Executa a descoberta de hipervisores.	{#HV.UUID}	Identificador único 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"`, veja 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.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"`
		`}]`

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.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 máquinas virtuais.	{#VM.UUID}	Identificador único 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 único 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 sistema operacional da máquina virtual guest, pode estar vazio.
	{#VM.GUESTFULLNAME}	Nome completo do sistema operacional 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" }]

Chave do item		

	”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, veja 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 único 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		
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"
                 }
            },
            {
                    // outros sensores hv hw
                }
        ]
        }
}
```

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 nenhum snapshot existir, 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 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).

Criar um host

1. Vá para *Coleta de dados* → *Hosts*.

2. Crie um host:

- No campo *Nome do host*, insira um nome de host (por exemplo, "VMware VMs").
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, "Virtual machines").
- Na guia *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.

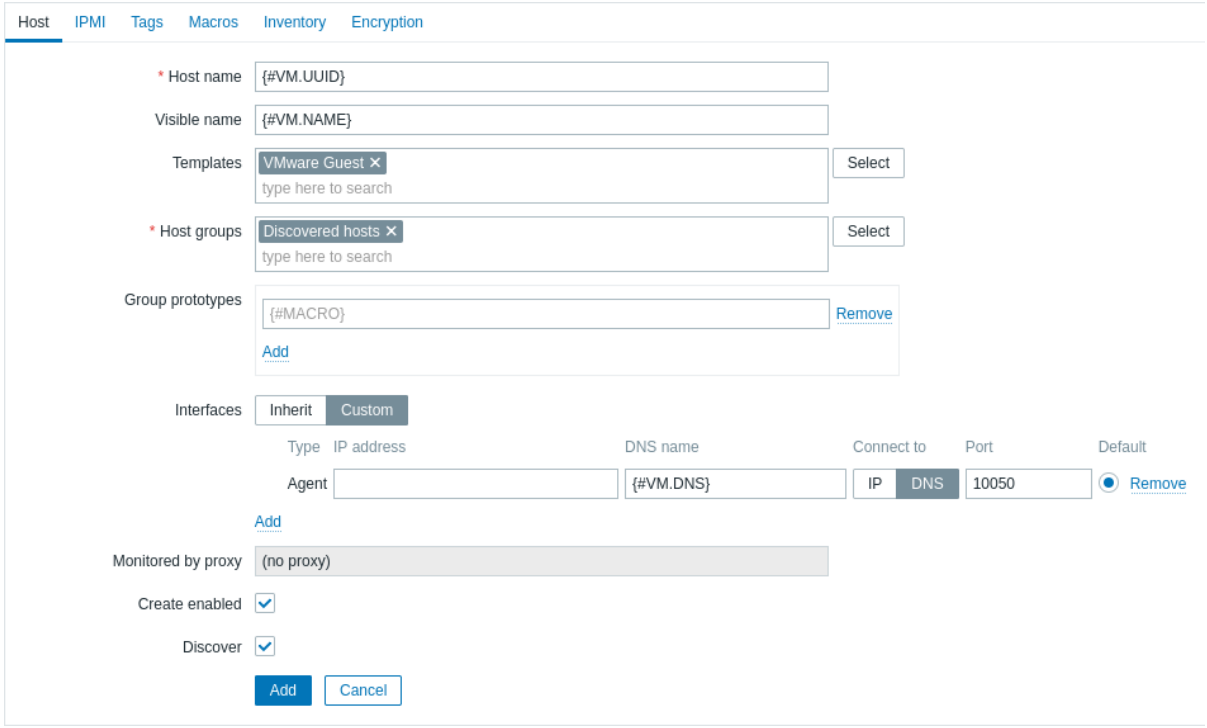
3. Clique no botão *Adicionar* para criar a regra de descoberta de baixo nível. Esta regra de descoberta irá descobrir máquinas virtuais em seu ambiente VMware.

Criar um protótipo de host

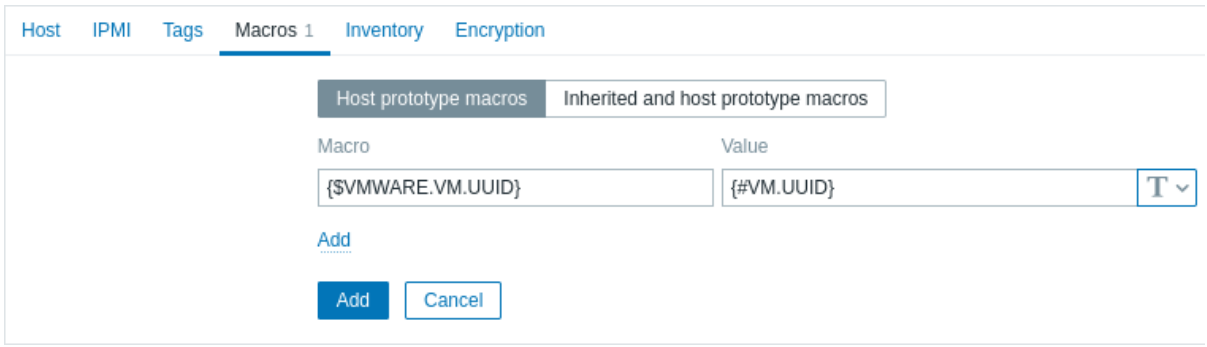
1. Na lista de regras de descoberta de baixo nível, clique em *Protótipos de host* para a regra de descoberta de baixo nível criada anteriormente.

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 {$VMWARE.VM.UUID} com o valor {#VM.UUID}. Isso é necessário para o funcionamento correto do template *VMware Guest*, que utiliza essa macro como macro de usuário em nível de host nos parâmetros dos itens (por exemplo, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

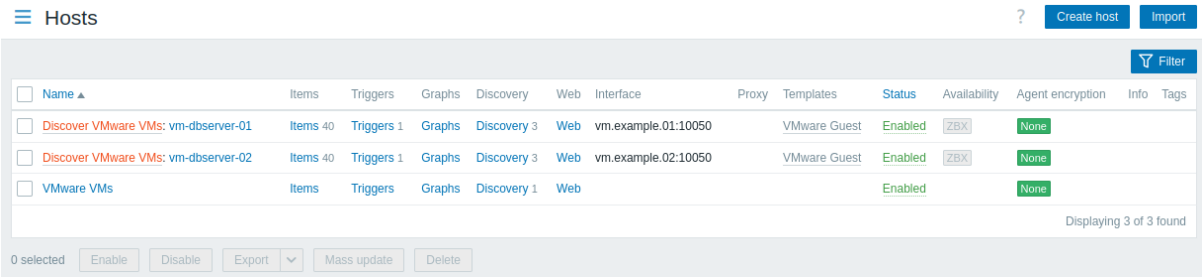


3. Clique no botão *Adicionar* para criar o protótipo de host. Esse protótipo de host será usado para criar hosts para máquinas virtuais descobertas pela regra de descoberta de baixo nível criada anteriormente.

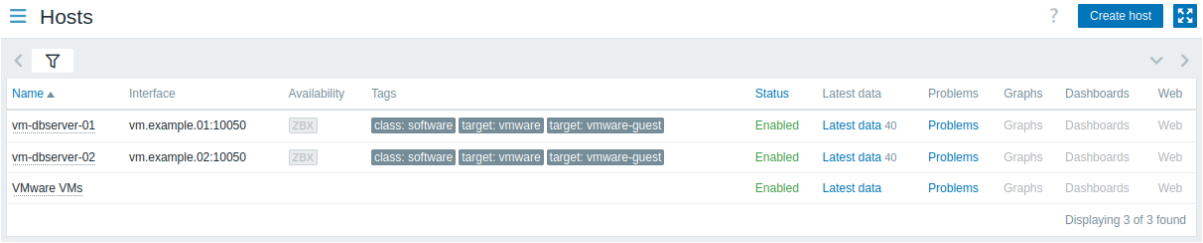
Visualizar hosts e métricas

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 de 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 pausadas 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 as 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:

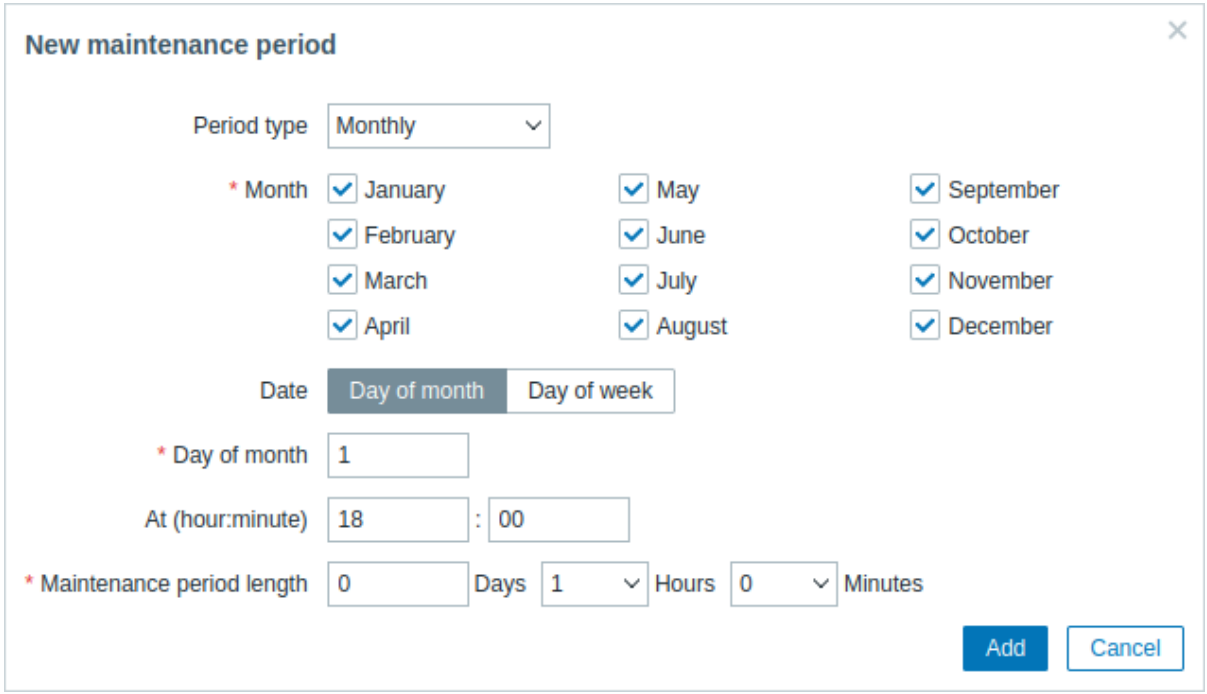
1. Vá para: *Coleta de dados* > *Manutenção*.
2. Clique em *Criar período de manutenção* (ou no nome de um período de manutenção existente).
3. Insira os parâmetros de manutenção no formulário.

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 popup 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 auto-completável, 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 auto-completável, 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 tempo para uma manutenção recorrente ou única. O formulário é dinâmico, com campos disponíveis mudando com base no *Tipo de período* selecionado.



Tipo de período	Descrição
Apenas uma vez	Configure 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.
Diário	Configure 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)* - hora 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 inicial será o dia em que o tempo *Ativo desde* cai. 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.
Semanal	Configure 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)* - hora 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 inicial será a semana em que o tempo *Ativo desde* cai. Para exemplos, veja a descrição do parâmetro *Diário* acima.
Mensal	Configure 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)* - hora 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 servidor Zabbix é utilizado. Para garantir um comportamento previsível dos períodos de manutenção recorrentes, é necessário utilizar 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 horário de verão e o horário atual mudar de 02:00 para 03:00, a manutenção continuará por mais uma hora (até 04:00);
- se após duas horas de manutenção (às 03:00) ocorrer uma mudança de horário de verão e o horário atual mudar de 03:00 para 02:00, a manutenção será interrompida, pois já se passaram duas horas;
- se um período de manutenção começar durante a hora que é pulada por uma mudança de horário de verão, então a manutenção não começará.

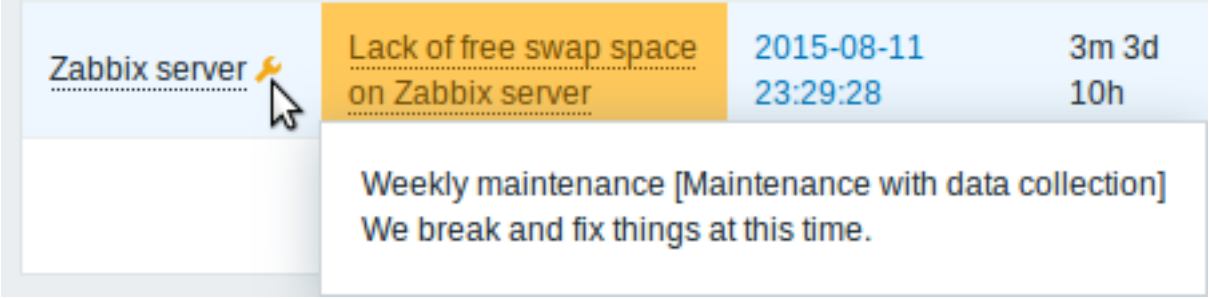
Se um período de manutenção for definido como "1 dia" (o período real da manutenção é de 24 horas, já que o Zabbix calcula dias em horas), começa às 00:00 e termina às 00:00 do dia seguinte:

- 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.

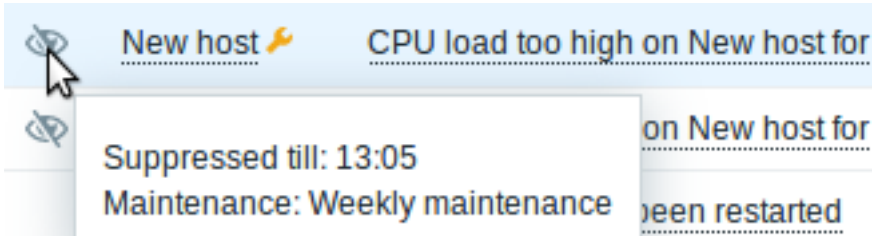
Além disso, hosts em manutenção recebem um fundo laranja em *Monitoramento* > *Mapas*.

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 do 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 do Zabbix não tem conhecimento dos períodos de manutenção porque não há sincronização da configuração de manutenção entre o server e o proxy do Zabbix. Verificações internas calculadas em proxies do Zabbix (por exemplo, `zabbix[queue,,]` e `zabbix[stats,,,queue,,]`) relatam items atrasados independentemente do status de manutenção no server do 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 do Zabbix.

12 Expressões regulares

Visão geral Expressões regulares compatíveis com Perl (PCRE, PCRE2) são suportadas no Zabbix.

Existem duas maneiras de usar expressões regulares no Zabbix:

- inserindo 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 de manual do pcrestack para mais informações.

Observe que, na correspondência multilinha, os delimitadores ^ 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.

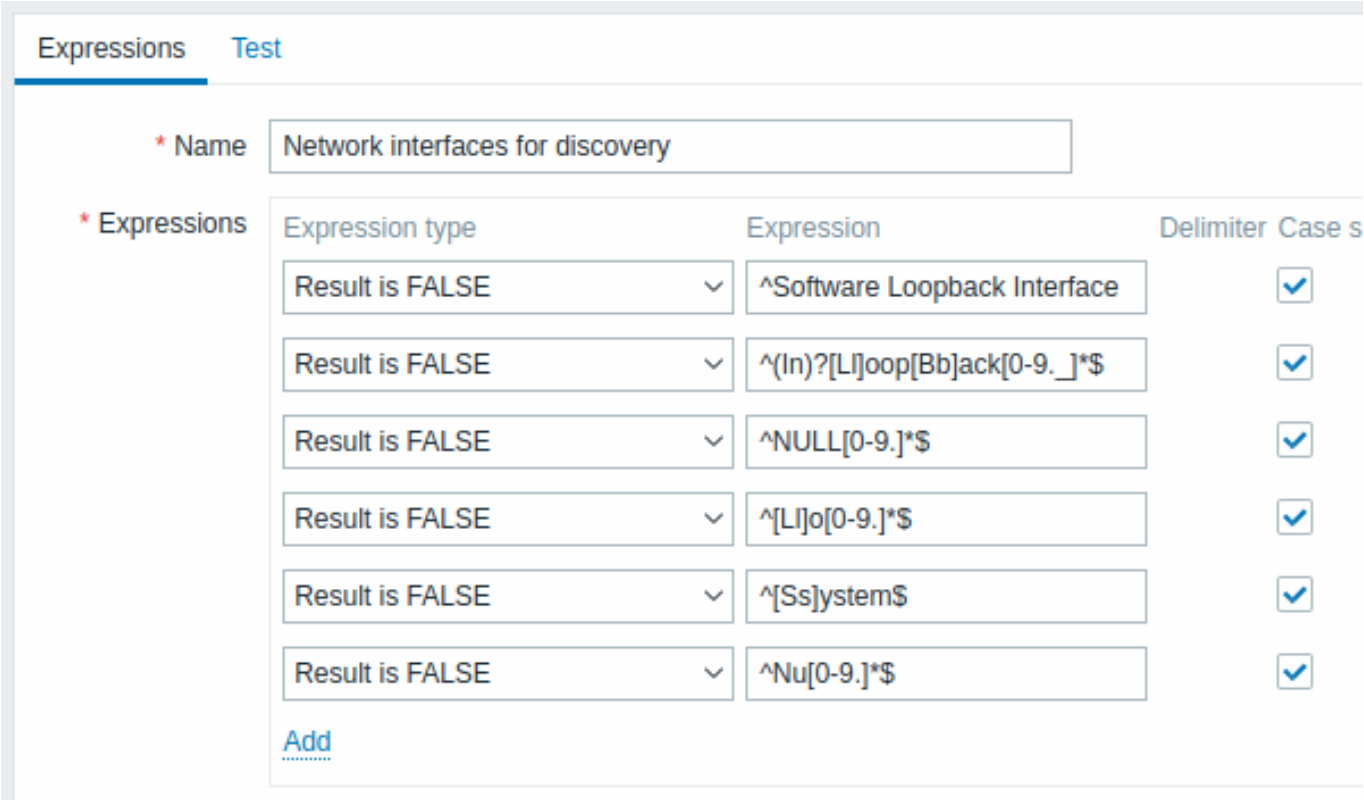
Expressões regulares globais Existe um editor avançado para criar e testar expressões regulares complexas no frontend do Zabbix.

Uma vez que uma expressão regular tenha sido criada dessa forma, ela pode ser usada em vários locais no frontend, referenciando 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.



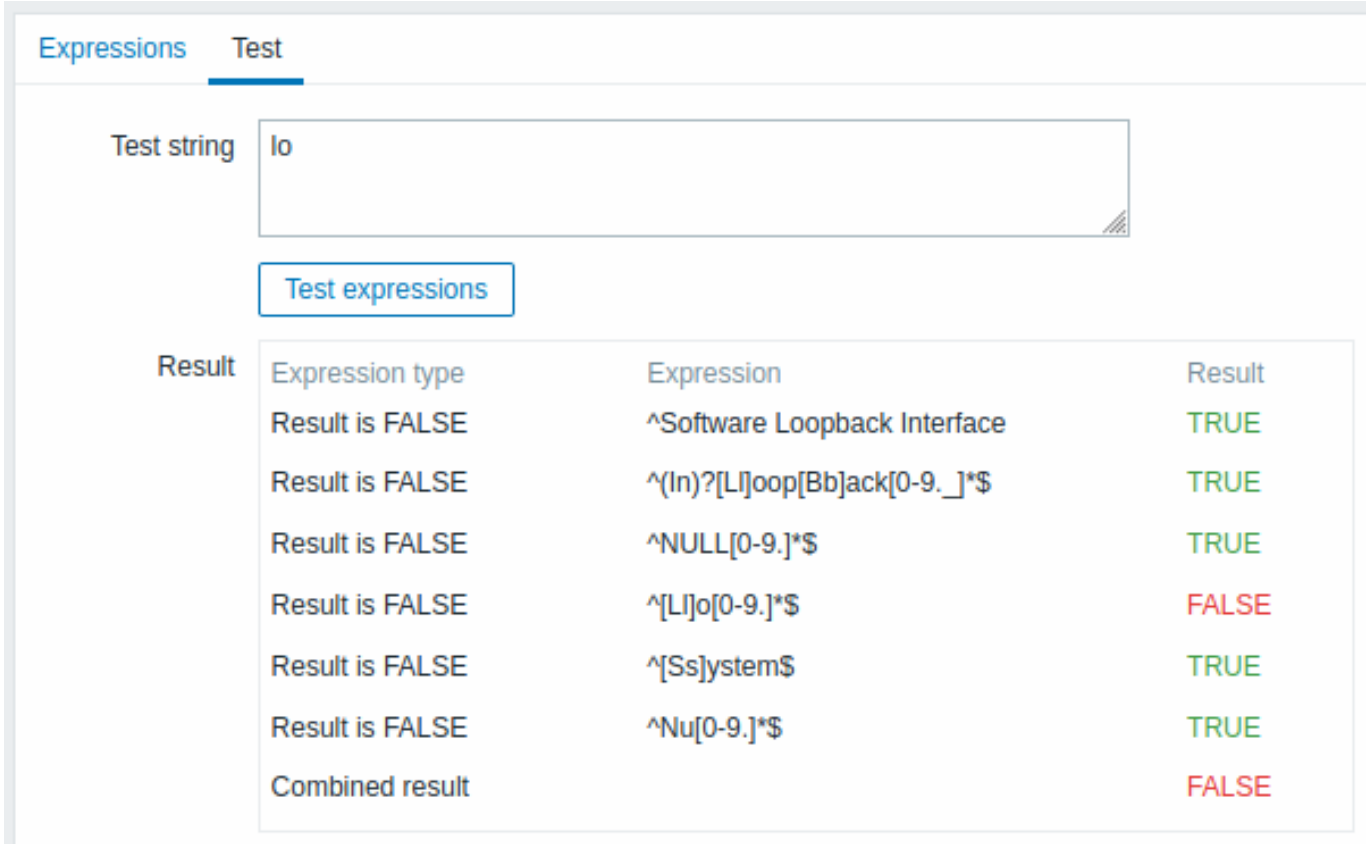
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

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"* é 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.

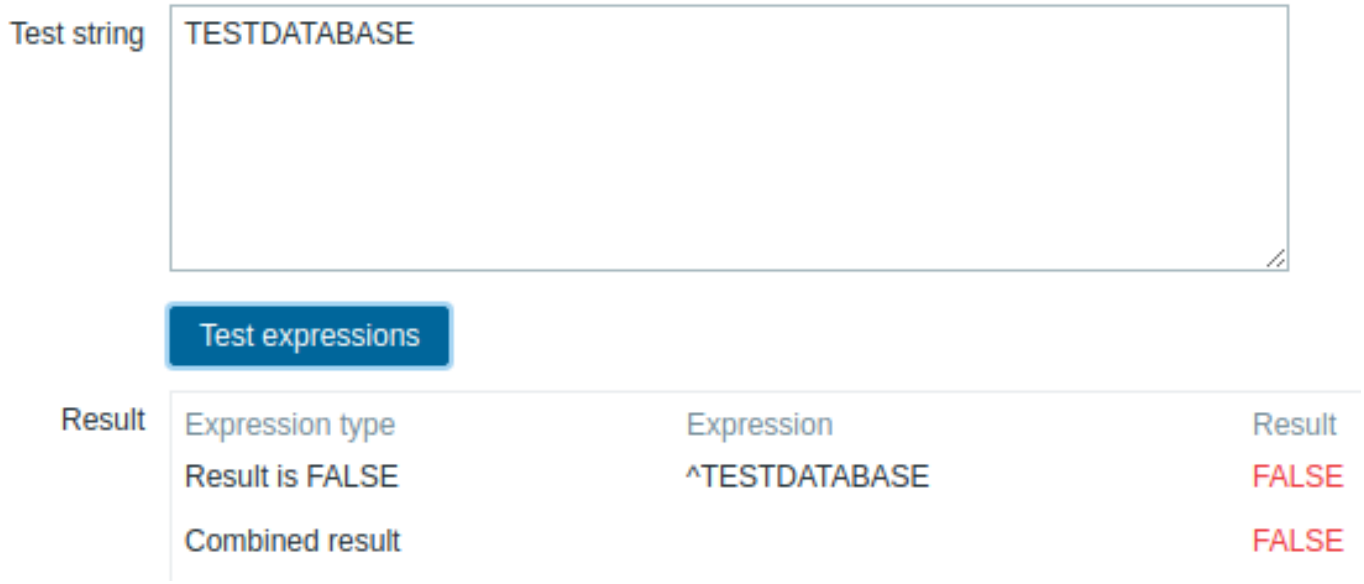
Nome	Expressão	Corresponde
Sistemas de arquivos para descoberta	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)$`	”btrfs” ou ”ext2” ou ”ext3” ou ”ext4” ou ”jfs” ou ”reiser” ou ”xfs” ou ”ffs” ou ”ufs” ou ”jfs” ou ”jfs2” ou ”vxfs” ou ”hfs” ou ”refs” ou ”apfs” ou ”ntfs” ou ”fat32” ou ”zfs”
Interfaces de rede para descoberta	`^Software Loopback Interface`	Strings que começam com ”Software Loopback Interface”.
	`^lo$`	”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 que começam com ”NULL” seguidas opcionalmente por qualquer número de dígitos ou pontos.
	`^[Ll]o[0-9.]*$`	Strings que começam com ”Lo” ou ”lo” e opcionalmente seguidas por qualquer número de dígitos ou pontos.
	`^[Ss]ystem$`	”System” ou ”system”
	`^Nu[0-9.]*$`	Strings que começam com ”Nu” seguidas opcionalmente 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 na descoberta de baixo nível para descobrir bancos de dados, exceto um banco de dados com um nome específico:

`^TESTDATABASE$`

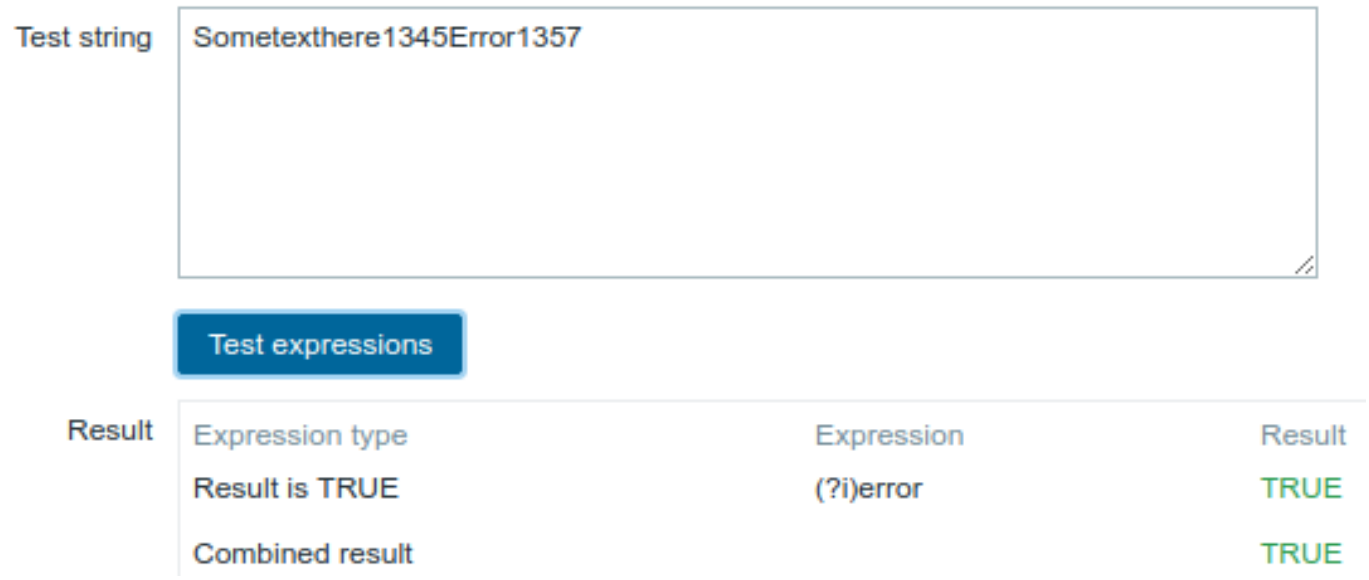


Tipo de expressão escolhido: ”O resultado é FALSO”. Não corresponde ao nome, contendo a string ”*TESTDATABASE*”.

Exemplo com um modificador de regex embutido

Uso da seguinte expressão regular incluindo um modificador embutido (?i) para corresponder aos caracteres ”error”:

`(?i)error`

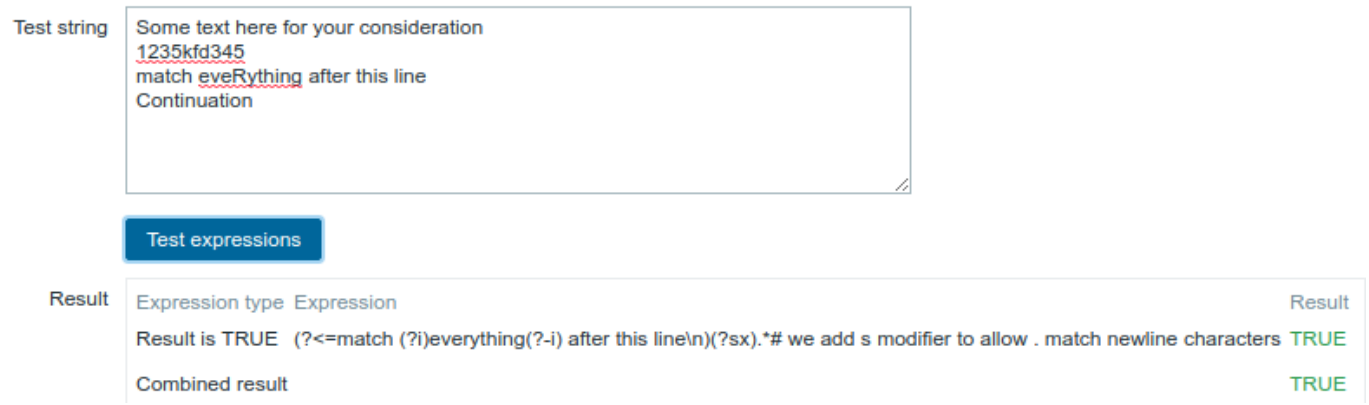


Tipo de expressão escolhido: ”Resultado é VERDADEIRO”. Os caracteres ”error” são correspondidos.

Outro exemplo com um modificador regex embutido

Uso da seguinte expressão regular incluindo múltiplos modificadores embutidos para corresponder aos caracteres após uma linha específica:

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# adicionamos o modificador s para permitir que . c
```



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 do manual pcresyntax. Para mais informações sobre a sintaxe PCRE, consulte a documentação HTML do 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		parâmetro `regexp` suporta ambos, 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 regsub() iregsub()	Sim	Não	Não	No contexto de macro com prefixo **regex:** parâmetro `pattern`
Indicadores de link em mapas	Sim	Não	Não	campo *Padrão* (para itens de texto)
Mapeamento de ícones	Sim	Sim	Não	campo *Expressão*
Mapeamento de valores	Sim	Não	Não	campo *Valor* se o tipo de mapeamento for `regexp`

13 Reconhecimento de problema

Visão geral Eventos de problema no Zabbix podem ser reconhecidos por usuários.

Se um usuário for notificado sobre um evento de problema, ele pode acessar o frontend do Zabbix, abrir a janela pop-up de atualização do problema usando uma das formas listadas abaixo e reconhecer o problema. Ao reconhecer, ele pode inserir um comentário, dizendo que está trabalhando no problema ou qualquer outra informação que desejar.

Dessa forma, se outro usuário do sistema identificar o mesmo problema, ele verá imediatamente se o problema já foi reconhecido e os comentários feitos até o momento.

Assim, o fluxo de trabalho para resolução de problemas por mais de um usuário do sistema pode ocorrer de forma coordenada.

O status de reconhecimento também é usado ao definir operações de ação. Você pode definir, por exemplo, que uma notificação seja enviada para um gerente de nível superior apenas se um evento não for reconhecido por algum tempo.

Para reconhecer eventos e comentar sobre eles, um usuário deve ter pelo menos permissões de leitura para os triggers correspondentes. Para alterar a severidade do problema ou fechá-lo, o usuário deve ter permissões de leitura e gravação para os triggers correspondentes.

Existem **várias** maneiras de acessar a janela pop-up de atualização do problema, que permite reconhecer um problema.

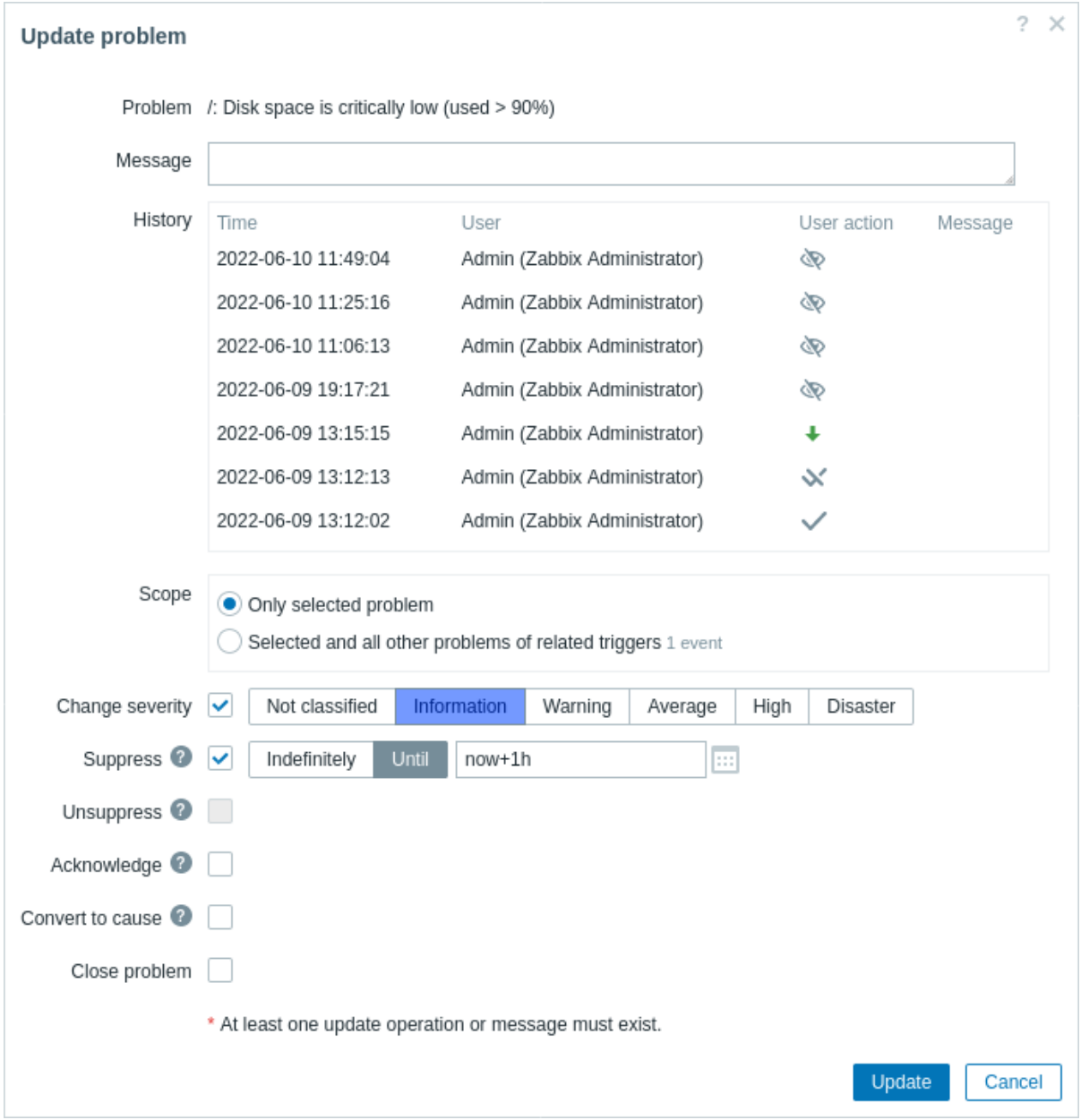
- Você pode selecionar problemas em *Monitoramento* → *Problemas* e então clicar em *Atualização em massa* abaixo da lista
- Você pode clicar em *Atualizar* na coluna *Atualizar* de um problema em:
  - *Dashboards* (widgets *Problemas* e *Problemas por severidade*)
  - *Monitoramento* → *Problemas*
  - *Monitoramento* → *Problemas* → *Detalhes do evento*
- Você pode clicar em uma célula de problema não resolvido em:
  - *Dashboards* (widget *Visão geral dos triggers*)

O menu pop-up contém uma opção *Atualizar* que o levará para a janela de atualização do problema.

Atualizando problemas O popup de atualização de problemas permite:

- comentar sobre o problema
- visualizar comentários e ações até o momento
- alterar a severidade do problema

- suprimir/remover 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 são 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.

Parâmetro	Descrição
Escopo	Defina o escopo de ações como alteração de severidade, reconhecimento ou fechamento manual de problemas: **Apenas o problema selecionado** - afetará apenas este evento **Problema selecionado 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.
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 está disponível se houver permissões de leitura e gravação para pelo menos um dos problemas selecionados. Somente esses problemas que são 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
Remover supressão	Marque a caixa de seleção para remover 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 em problema(s) causa.
Fechar problema	Marque a caixa de seleção para fechar manualmente o(s) problema(s) selecionado(s). A caixa de seleção para fechar um problema está 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 que podem ser 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 a contagem de problemas é exibida no dashboard ou em mapas. Para isso, você deve fazer seleções na opção *Exibição de problemas*, disponível tanto na configuração do mapa quanto no widget *Problemas por severidade* do dashboard. É possível exibir toda a contagem de problemas, a contagem de problemas não reconhecidos separada do total ou apenas a contagem de problemas não reconhecidos.

Com base nas informações de atualização do problema (reconhecimento, etc.), é possível configurar operações de atualização - enviar uma mensagem ou executar comandos remotos.

1 Supressão de problema

Visão geral

A supressão de problemas oferece uma maneira de ocultar temporariamente um problema que pode ser tratado posteriormente. Isso é útil para limpar a lista de problemas, a fim de dar a mais alta prioridade às questões mais urgentes. Por exemplo, às vezes pode surgir um problema no fim de semana que não é urgente o suficiente para ser tratado imediatamente, então ele pode ser "adiado" até a manhã de segunda-feira.

A supressão de problemas permite ocultar um *único* problema, em contraste com a supressão de problemas por meio da manutenção do host, quando todos os problemas do host em manutenção são ocultados.

As operações para ações de trigger serão pausadas para problemas suprimidos da mesma forma que é feito com a manutenção do host.

Configuração

Um problema pode ser suprimido através da janela **atualização de problema**, onde a supressão é uma das opções de atualização do problema juntamente com comentar, alterar a severidade, reconhecer, etc.

Um problema também pode ser dessuprimido através da mesma janela de atualização de problema.

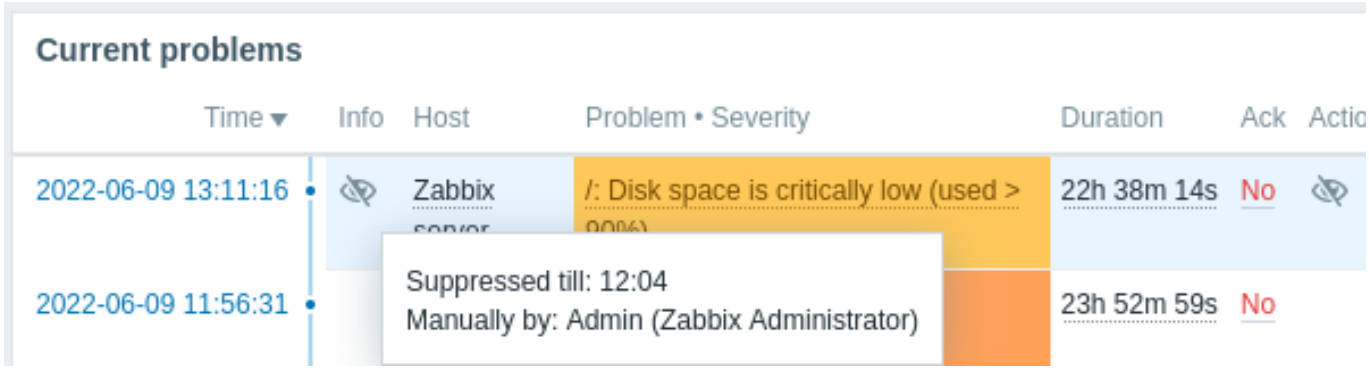
Exibição

Uma vez suprimido, o problema é marcado por um ícone de supressão piscando na coluna *Info*, 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 suprime o problema, o ícone para de piscar. Se o ícone de supressão continuar piscando por muito tempo, isso pode indicar um problema no server, por exemplo, se o server estiver inativo e o gerenciador de tarefas não puder concluir a tarefa. A mesma lógica se aplica à dessupressão. No curto período após o envio da tarefa e antes do server concluí-la, o ícone de dessupressão pisca.

Um problema suprimido pode ser ocultado ou exibido, dependendo das configurações do filtro de problemas/widget.

Quando exibido na lista de problemas, um problema suprimido é marcado pelo ícone de supressão e os detalhes da supressão são exibidos ao passar o mouse:



Os detalhes da supressão também são exibidos em um popup ao posicionar o mouse sobre o ícone de supressão na coluna *Ações*.

14 Exportação/importação de configuração

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

Objetos que podem ser exportados/importados são:

- Grupos de hosts (*apenas através da Zabbix API*)
- Grupos de templates (*apenas através da Zabbix API*)
- Templates
- Hosts

- Mapas de rede
- Tipos de mídia
- Imagens

Formato de exportação

Os dados podem ser exportados usando o frontend web do Zabbix ou a API do Zabbix. 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, expressão regular modificada ou protótipos adicionados 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 estejam 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 a extensão 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: '8.0'
```

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>8.0</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": "8.0"
```

```
    }
}
```

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 de templates. 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 de hosts. 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:

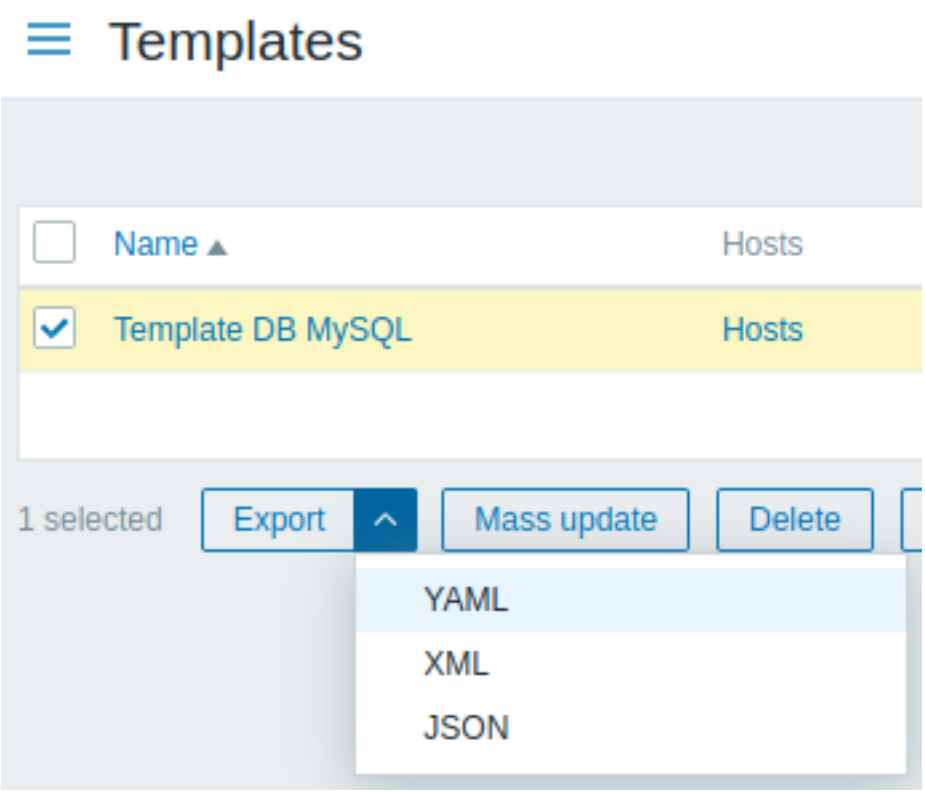
- Grupos de templates vinculados
- Grupos de hosts vinculados (se usados na configuração de prototipagem de host)
- Dados do template
- Vinculação a outros templates
- Vinculação a grupos de templates

- 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.



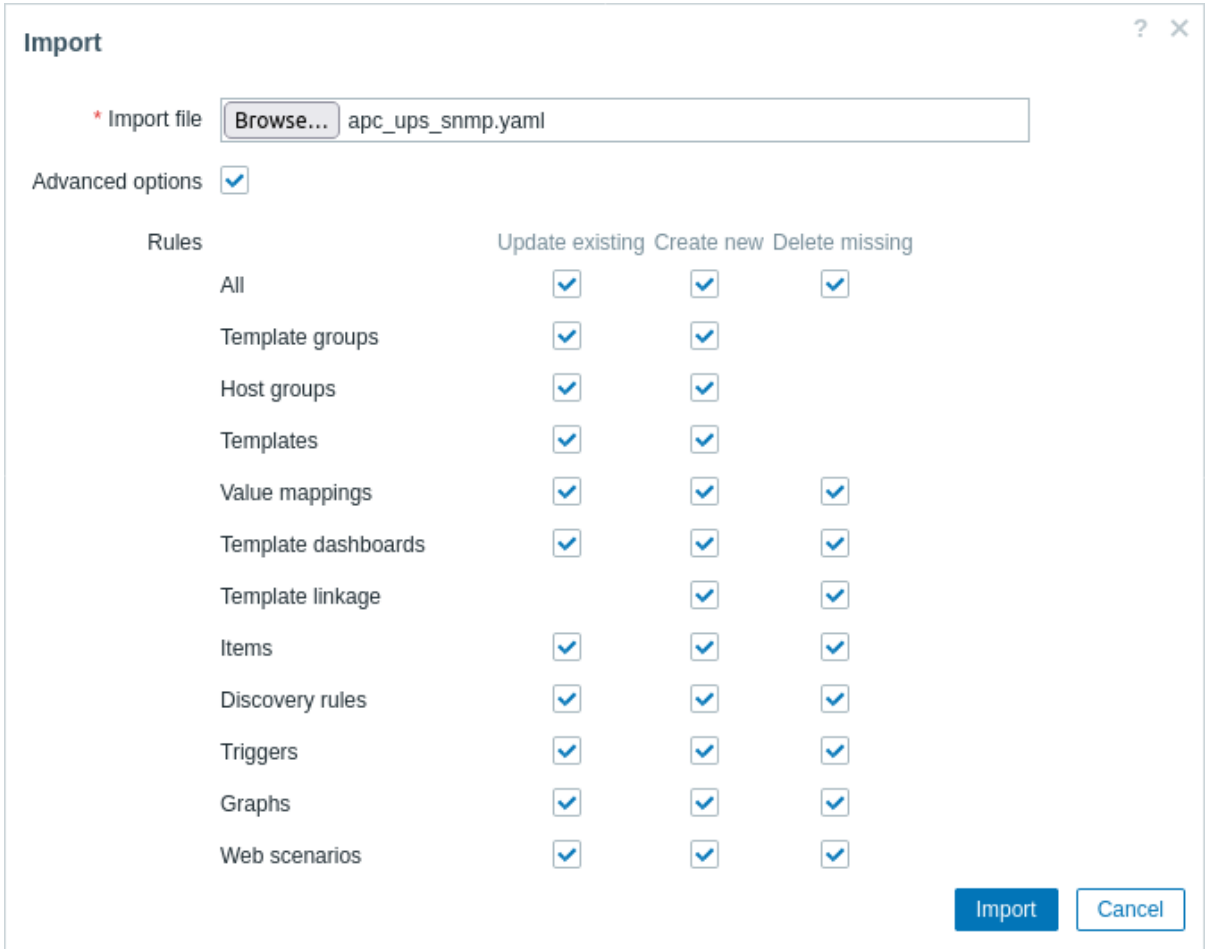
Dependendo do formato selecionado, os templates são exportados para um arquivo local com um nome padrão:

- `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.



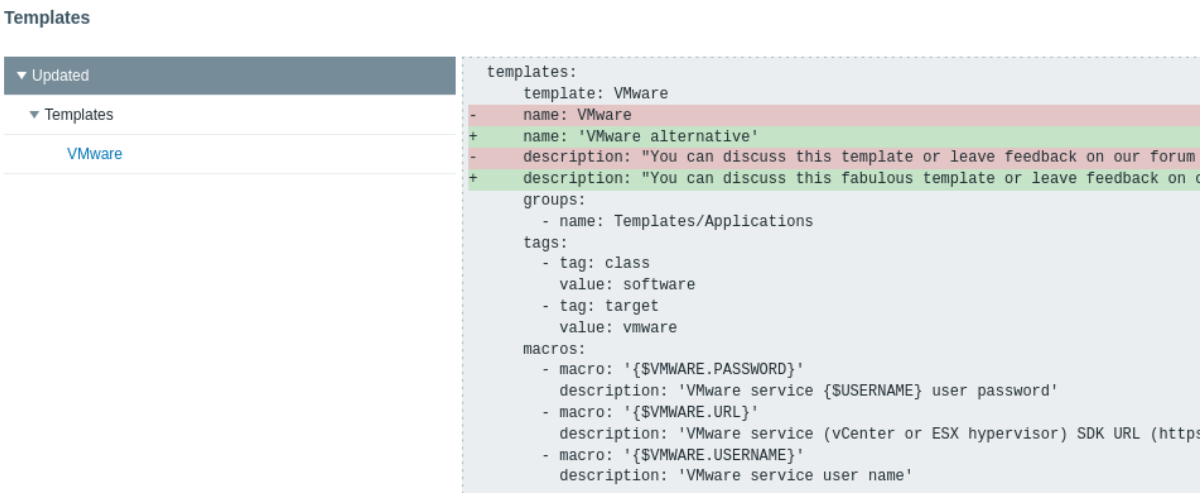
Se você marcar a caixa de seleção *Opções avançadas*, uma lista detalhada de todos os elementos importáveis será exibida - marque ou desmarque cada regra de importação conforme necessário.

Se você clicar na caixa de seleção na linha *Todos*, todos os elementos abaixo dela serão marcados/desmarcados.

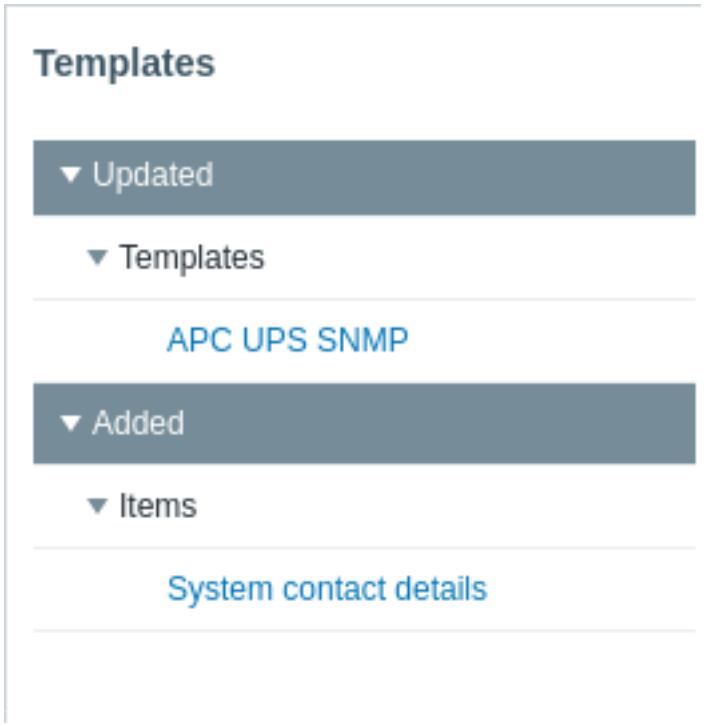
Regras de importação:

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 *Vínculo de template*, o vínculo de template atual que não estiver presente no arquivo de importação será desvinculado. 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 Quando um template é exportado, o Zabbix gera um formato estruturado em YAML, JSON ou XML. A exportação inclui elementos do template, como metadados do template, items, macros, triggers, dashboards e muito mais.

Cada elemento serve a um propósito específico e pode conter elementos aninhados.

As seções a seguir descrevem cada elemento no formato de exportação. Os exemplos usam o template Linux by Zabbix agent.

Uma elipse (`...`) indica elementos omitidos para brevidade. A nota (`veja a tabela abaixo`) é usada quando o elemento é explicado em mais detalhes nas seções posteriores.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (ver tabela abaixo)
```

```
  triggers: (ver tabela abaixo)
  graphs: (ver tabela abaixo)
```

Metadados do template

Elemento	Tipo	Descrição
version	string	(obrigatório) Versão da exportação.
template_groups		(obrigatório) Elemento raiz para grupos de template.
uuid	string	(obrigatório) Identificador único para este grupo de template.
name	string	(obrigatório) Nome do grupo de template.
host_groups		Elemento raiz para grupos de host que são usados por protótipos de host.
uuid	string	(obrigatório) Identificador único para este grupo de host.
name	string	(obrigatório) Nome do grupo de host.
templates		Elemento raiz para templates.
triggers		Elemento raiz para elementos de trigger de template, que são os mesmos que para triggers de item de template.
graphs		Elemento raiz para elementos de gráfico de template, que são os mesmos que para gráficos de host.

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Este é um template oficial do Linux. Requer o agent Zabbix 8.0 ou mais recente. (...)
      wizard_ready: 'YES'
      readme: |
        ## Visão geral

        Este é um template oficial do Linux. Requer o agent Zabbix 8.0 ou mais recente. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (veja a tabela abaixo)
      discovery_rules: (veja a tabela abaixo)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Tempo limite após o qual o agent é considerado indisponível. Funciona apenas para
          config: (veja a tabela abaixo)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Limite crítico de utilização da CPU expresso em %.'
          config: (veja a tabela abaixo)
        (...)
      dashboards: (veja a tabela abaixo)
      valuemaps: (veja a tabela abaixo)
    (...)
```

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.
wizard_ready	text	Se o template está disponível para seleção no Assistente de Host. Valores possíveis:[1] NO (0, padrão), YES (1). Veja também: Objeto template (`wizard_ready`).
readme	text	Instruções de configuração específicas do template para exibir no Assistente de Host. Suporta formatação Markdown.
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.
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, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
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 da 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.
config		Elemento raiz para configuração de macro do template responsável por como a macro é exibida no Assistente de Host.
dashboards		Elemento raiz para dashboards do template.
valuemaps		Elemento raiz para mapas de valores do template.

Itens do template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix agent ping'
          key: agent.ping
          description: 'O agent sempre retorna "1" para este item. Pode ser usado em combinação com `nodat
          valuemap:
            name: 'Zabbix agent ping status'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'Number of installed packages'
          key: system.sw.packages.get
          delay: 1h
          preprocessing: (ver tabela abaixo)
```

```
      tags:
        - tag: component
          value: os
      triggers: (ver tabela abaixo)
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: 'Memory utilization'
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: 'A porcentagem de memória usada é calculada como `100-pavailable`.'
      preprocessing: (ver tabela abaixo)
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: (ver tabela abaixo)
    (...)
```

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) OID 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.
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).
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.

Elemento	Tipo	Descrição
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 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 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 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.
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).

Elemento	Tipo	Descrição
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 se 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 se 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 item do template.

Note:
Veja também: Objeto item (consulte a propriedade relevante com o nome correspondente).

Etapas de pré-processamento de valor de item de template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

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.

Elemento	Tipo	Descrição
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 de item (consulte a propriedade relevante com o mesmo nome).

Triggers de item de template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Number of installed packages has been changed'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
                  value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: High memory utilization'
              event_name: 'Linux: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)'
              priority: AVERAGE
              description: 'The system is running out of free memory.'
              dependencies:
                - name: 'Linux: Lack of available memory'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

Elemento	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 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.

Elemento	Tipo	Descrição
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 de template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (veja a tabela abaixo)
          description: 'The discovery of network interfaces.'
          item_prototypes: (veja a tabela abaixo)
          trigger_prototypes: (veja a tabela abaixo)
          graph_prototypes: (veja a tabela abaixo)
        (...)
```

Attention:
A maioria dos elementos das regras de descoberta de baixo nível de template são os mesmos que para itens de template. A tabela abaixo descreve os elementos que diferem dos itens de 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 regras de descoberta de baixo nível de template.
lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não mais descobertos serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para excluir recursos LLD perdidos. Valores possíveis: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.

Elemento	Tipo	Descrição
enabled_lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não mais descobertos serão desabilitados.
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 template, que são os mesmos que para itens de template.
trigger_prototypes		Elemento raiz para elementos de protótipo de trigger de template, que são os mesmos que para triggers de item de template.
graph_prototypes		Elemento raiz para elementos de protótipo de gráfico de template, que são os mesmos que para gráficos de host.
host_prototypes		Elemento raiz para elementos de protótipo de host de template, que são os mesmos que para hosts.
parent_discovery_rule		Elemento raiz para a regra de descoberta de baixo nível pai (ou protótipo de regra) do protótipo de regra de descoberta de baixo nível. Esta propriedade indica que é um protótipo de regra LLD, filho direto da regra/protótipo de regra referenciado.
key	string	(obrigatório) A chave da regra de descoberta de baixo nível pai (ou protótipo de regra).
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 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 de template, que são os mesmos que para filtros de regras de descoberta de baixo nível de template.
operations		Elemento raiz para operações de sobrescrita de regra de descoberta de baixo nível de 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

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
        - uuid: 2bbdc79f082d4c618e01bec625e9c90a
          (...)
          filter:
            evaltype: AND
```

```
            conditions:
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.MATCHES}'
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
              - macro: '{#DEVTYPE}'
                value: disk
          (...)
```

Elemento	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 condições de filtro.
conditions		Elemento raiz para condições de 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 da condição. Valores possíveis:[1] MATCHES_REGEX (8, padrão), NOT_MATCHES_REGEX (9).
formulaid	string	(obrigatório) ID exclusivo arbitrário 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 regra 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	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).

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 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).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP) ou NTLM.
http_password	string	Senha usada para autenticação BASIC (HTTP) ou NTLM.
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).
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 de cenário web de 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 requisição).
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 de campo de formulário)).
name	string	(obrigatório) Nome do campo do post.
value	string	(obrigatório) Valor do campo do post.

Elemento	Tipo	Descrição
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 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 requisição.
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).

Configuração de macro de template

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Segundos desde a última vez que o agent Zabbix foi visto'
            description: 'Tempo limite após o qual o agent é considerado indisponível.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'Limite de utilização da CPU expresso'
            description: 'Limite crítico de utilização da CPU expresso em %. No intervalo de 0 a 100 inclu
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Controle de interfaces'
            description: 'Dispara um trigger quando o status operacional da interface muda para "Link down
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Elemento	Tipo	Descrição
type	string	(obrigatório) Tipo de campo de entrada da macro. Valores possíveis:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Posição da macro na lista de macros.
section_name	string	Rótulo da seção recolhível onde a macro é agrupada.
label	string	(obrigatório para macros TEXT, LIST, CHECKBOX) Rótulo da macro.
description	text	Texto de ajuda da macro. Suporta formatação Markdown.
required	string	Marca a macro como obrigatória. Valores possíveis:[1] NO (0), YES (1). Suportado para macros TEXT e LIST.
regex	string	Expressão regular para validar a entrada do usuário em um campo de texto. Suportado para macros TEXT.
options		Elemento raiz para itens LIST ou valores CHECKBOX.
value	string	(obrigatório para macros LIST) Valor do item LIST.
text	string	(obrigatório para macros LIST) Texto do item LIST.
checked	string	(obrigatório para macros CHECKBOX) Valor que representa o estado marcado. Valores possíveis:[1] 0 (falso), 1 (verdadeiro). Suportado para macros CHECKBOX.
unchecked	string	(obrigatório para macros CHECKBOX) Valor que representa o estado desmarcado. Valores possíveis:[1] 0 (falso), 1 (verdadeiro). Suportado para macros CHECKBOX.

Note:
Veja também: Objeto de configuração de macro (consulte a propriedade relevante com o nome correspondente).

Dashboards de template

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (veja a tabela abaixo)
        (...)
```

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este dashboard.
name	string	(obrigatório) Nome do dashboard do template.
display pe-riod	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

```yaml
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

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 do template. Valores possíveis:[1] 0-71.
y	integer	Posição vertical a partir do topo do dashboard do 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 do 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.

Elemento	Tipo	Descrição
value	mixed	(obrigatório) Valor do campo do widget, dependendo do tipo de campo.

Note:
Veja também: Objeto widget do dashboard do template (consulte a propriedade relevante com o nome correspondente).

Mapas de valores do template

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
              newvalue: up
        (...)
```

Elemento	Tipo	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

Os hosts são exportados com muitos objetos relacionados e relações de objetos.

A exportação de host contém:

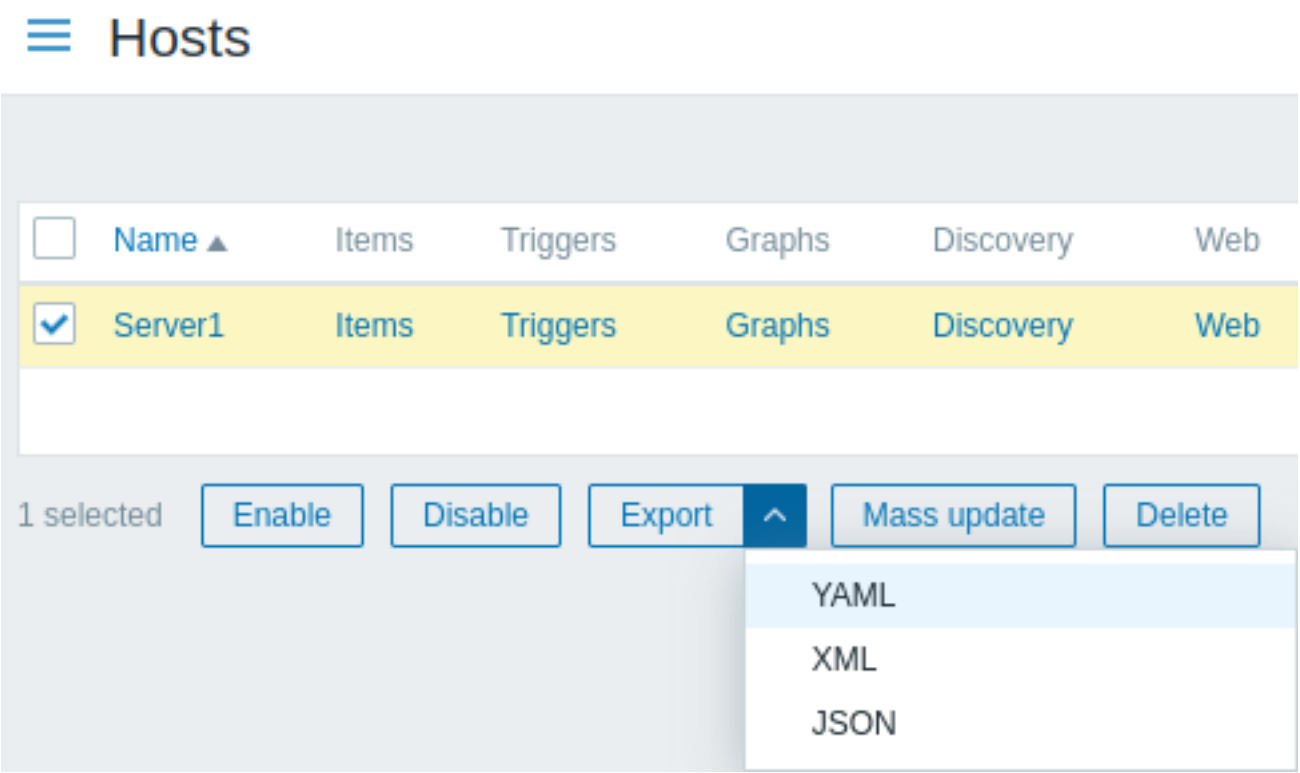
- Grupos de hosts vinculados

- 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.



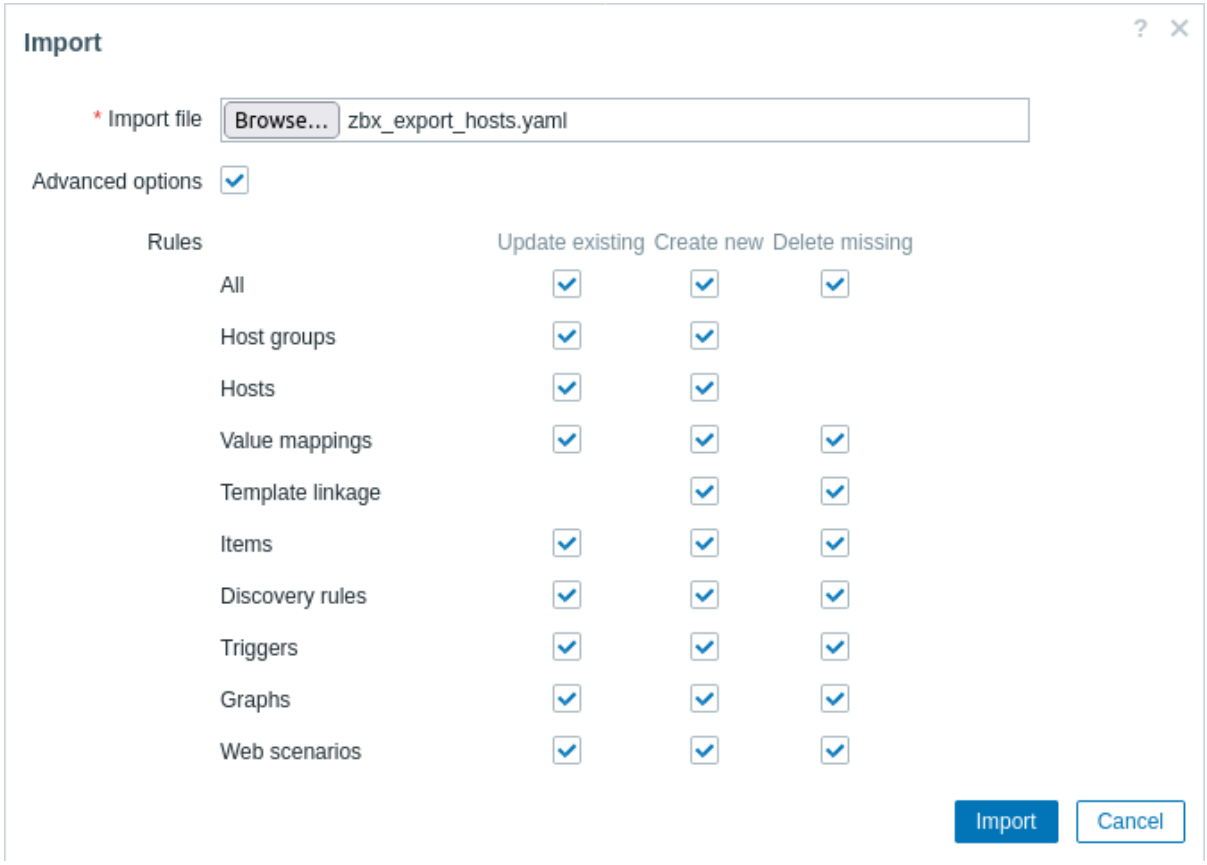
Dependendo do formato selecionado, os hosts são exportados para um arquivo local com um nome padrão:

- `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

Quando um host é exportado, o Zabbix gera um formato estruturado em YAML, JSON ou XML. A exportação inclui elementos do host, como metadados do host, items, macros, triggers e muito mais.

Cada elemento serve a um propósito específico e pode conter elementos aninhados.

As seções a seguir descrevem cada elemento no formato de exportação. Os exemplos usam um host com o template MySQL by Zabbix agent 2. Para fins de demonstração dos elementos, o template é desvinculado após a criação do host. A desvinculação mantém todos os elementos na configuração (diferente de desvincular e limpar, que os remove).

Uma elipse (`...`) indica elementos omitidos para brevidade. A nota (`veja a tabela abaixo`) é usada quando o elemento é explicado em mais detalhes nas seções posteriores.

```yaml
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (veja a tabela abaixo)
  graphs: (veja a tabela abaixo)
```

Metadados do host

Elemento	Tipo	Descrição
version	string	(obrigatório) Versão do Zabbix.
host_groups		(obrigatório) Elemento raiz para grupos de hosts.
uuid	string	(obrigatório) Identificador exclusivo para este grupo de hosts.
name	string	(obrigatório) Nome do grupo de hosts.
hosts		Elemento raiz para hosts.
graphs		Elemento raiz para gráficos de host.

```yaml
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (veja a tabela abaixo)
      items: (veja a tabela abaixo)
      discovery_rules: (veja a tabela abaixo)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: (veja a tabela abaixo)
  (...)
```

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	Como o host é monitorado. Valores possíveis:[1] SERVER (0, padrão), PROXY (1) ou PROXY_GROUP (2).
proxy		Elemento raiz para proxy.
name	string	(obrigatório) Nome do proxy (se houver) que monitora o host.
proxy_group		Elemento raiz para grupo de proxy.
name	string	(obrigatório) Nome do grupo de proxy (se houver) 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).

Elemento	Tipo	Descrição
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 do 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 da 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, veja 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 do host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

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.

Elemento	Tipo	Descrição
max_repetitions	string	Valor máximo de repetição para requisições nativas SNMP bulk (GetBulkRequest-PDUs). Suportado para itens SNMPv2 e SNMPv3 (itens `discovery[]` e `walk[]`). Padrão: 10.
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 requisições 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 do host (consulte a propriedade relevante com o nome correspondente).

Items do host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Número de transações que usaram um cache de disco temporário porque não puderam ca
          preprocessing: (veja a tabela abaixo)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
          params: |
            ( last(//mysql.innodb_buffer_pool_pages_total) -
            last(//mysql.innodb_buffer_pool_pages_free) ) /
            ( last(//mysql.innodb_buffer_pool_pages_total) +
            ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
            ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
```

```
      description: 'Proporção de páginas usadas para o total de páginas no buffer pool.'
      tags:
        - tag: component
          value: memory
      triggers: (veja a tabela abaixo)
    (...)
    - name: Uptime
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Número de segundos que o servidor está ativo.'
      preprocessing: (veja a tabela abaixo)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (veja a tabela abaixo)
    (...)
```

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), ITEM_TYPE_NESTED (23).
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 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.
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).
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.

Elemento	Tipo	Descrição
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 requisição de coleta de 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 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 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.
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).

Elemento	Tipo	Descrição
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 de 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 item 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

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

Elemento	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.

Elemento	Tipo	Descrição
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 de item (consulte a propriedade relevante com o nome correspondente).

Triggers de item de host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: Failed to fetch info data'
              event_name: 'MySQL: Failed to fetch info data (or no data for 30m)'
              priority: INFO
              description: 'Zabbix has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Elemento	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).

Elemento	Tipo	Descrição
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 de host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (veja a tabela abaixo)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
                  value: storage
                - tag: database
                  value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
              path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
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 recursos não mais descobertos serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para exclusão de 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 recursos não mais descobertos serão desabilitados.
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.
parent_discovery_rule		Elemento raiz para a regra de descoberta de baixo nível pai (ou protótipo de regra) do protótipo de regra de descoberta de baixo nível. Esta propriedade denota que é um protótipo de regra LLD, filho direto da regra/protótipo de regra referenciada.
key	string	(obrigatório) A chave da regra de descoberta de baixo nível pai (ou protótipo de regra).
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 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

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
```

```
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Elemento	Tipo	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 condições de 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 da 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 sobrescrita de regra 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), DISCOVERY_PROTOTYPE (4).
operator	string	Operador da condição de sobrescrita. 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 sobrescrita.
status	string	Status do objeto após a operação de sobrescrita.
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 sobrescrita.
history	string	Período de armazenamento do histórico definido para o protótipo de item após a operação de sobrescrita.
trends	string	Período de armazenamento de tendências definido para o protótipo de item após a operação de sobrescrita.
severity	string	Severidade do protótipo de trigger definida após a operação de sobrescrita.
tags		Elemento raiz para as tags definidas para o objeto após a operação de sobrescrita.
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 sobrescrita.
name	string	(obrigatório) Nome do template.
inventory_mode	string	Modo de inventário do protótipo de host definido após a operação de sobrescrita.

Note:
Veja também: Objeto de operação de sobrescrita 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. 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 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).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP) ou NTLM.
http_password	string	Senha usada para autenticação BASIC (HTTP) ou NTLM.
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 host.
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 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 do post) ou um array de campos HTTP (dados de campo de formulário)).
name	string	(obrigatório) Nome do campo post.
value	string	(obrigatório) Valor do campo post.

Elemento	Tipo	Descrição
variables		Elemento raiz das variáveis em 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 (em 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 do host

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (ver tabela abaixo)
    (...)
```

Elemento	Tipo	Descrição
uuid	string	Identificador único do gráfico.
name	string	(obrigatório) Nome do gráfico.
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 mínimo 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), STACKED (1), PIE (2), EXPLODED (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).

Elemento	Tipo	Descrição
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] CALCULATED (0, padrão), FIXED (1), ITEM (2).
ymin_item_1		(obrigatório se `ymin_type_1` estiver 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] CALCULATED (0, padrão), FIXED (1), ITEM (2).
ymax_item_1		(obrigatório se `ymax_type_1` estiver 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 do host.

Note:
Veja também: Objeto gráfico (consulte a propriedade relevante com o nome correspondente).

Itens de gráfico de host

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

Elemento	Tipo	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).
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).

Elemento		Tipo	Descrição
type		string	Tipo de item do gráfico. Valores possíveis:[1] SIMPLE (0, padrão), GRAPH_SUM (2; o valor do item representa o gráfico inteiro; 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

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

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.

5 Mapas de rede

Visão geral

O export do mapa de rede contém:

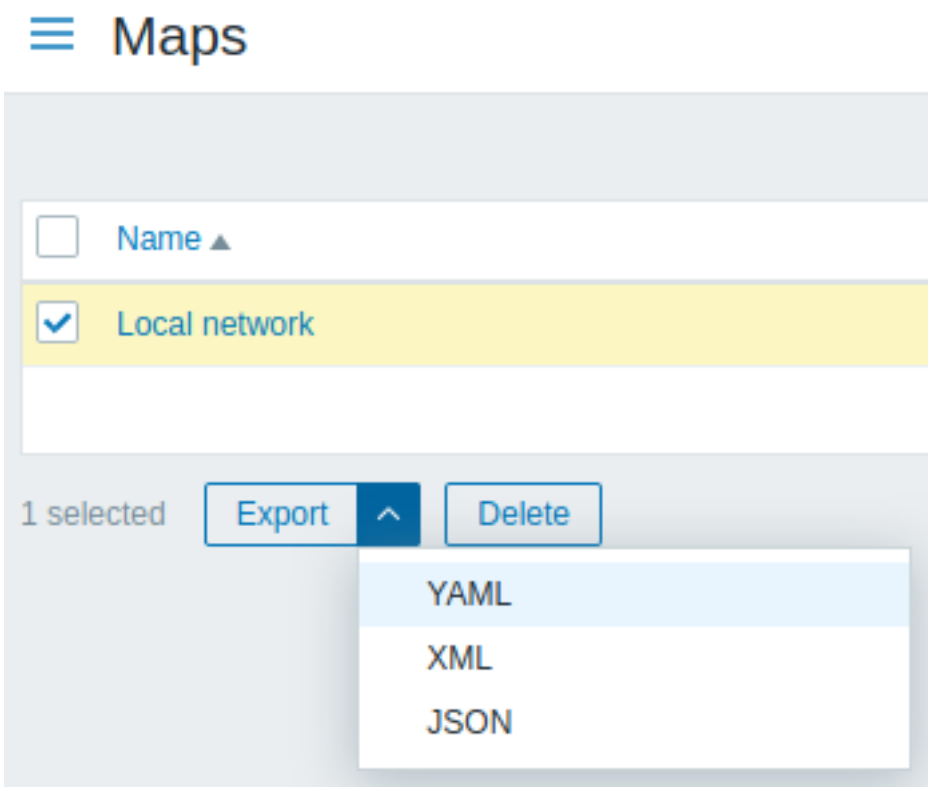
- Todas as imagens relacionadas
- Estrutura do mapa (todas as configurações do mapa, todos os elementos contidos com suas configurações, links do mapa e indicadores de status dos links do mapa)

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.



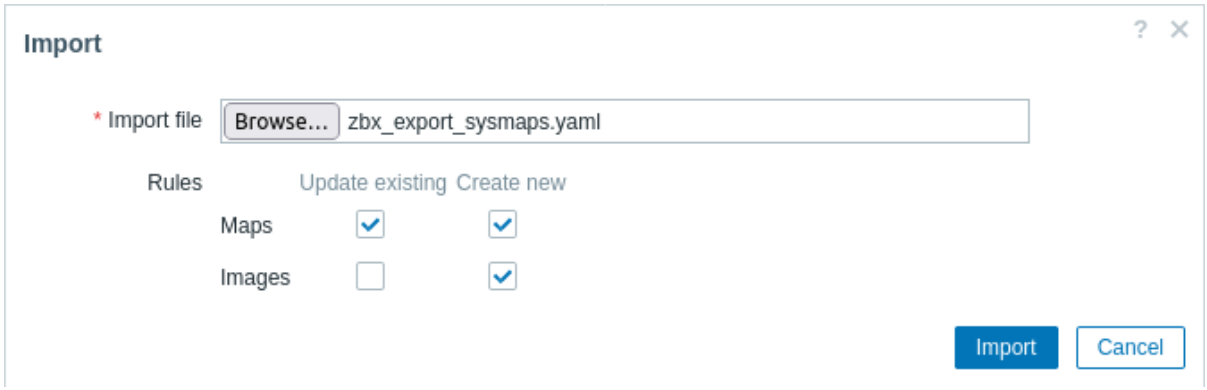
Dependendo do formato selecionado, os mapas são exportados para um arquivo local com um nome padrão:

- `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: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
```

```
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
            name: Zabbix_server_3D_(128)
          icon_on: {  }
          icon_disabled: {  }
          icon_maintenance: {  }
          urls: {  }
          evaltype: '0'
      shapes:
        - type: '0'
          x: '0'
          'y': '0'
          width: '680'
          height: '15'
          text: '{MAP.NAME}'
          font: '9'
          font_size: '11'
          font_color: '000000'
          text_halign: '0'
          text_valign: '0'
          border_type: '0'
          border_width: '0'
          border_color: '000000'
          background_color: ''
          zindex: '0'
      lines: {  }
      links: {  }
```

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.

Elemento	Tipo	Descrição
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 destaque de ícone para triggers ativas e status de host. Valores possíveis: 0 - Não; 1 - Sim.
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 da 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.

Elemento	Tipo	Descrição
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.
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.

Elemento	Tipo	Descrição
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.
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 - Centralizado; 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 - Nenhuma; 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 - Nenhuma; 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).

Elemento	Tipo	Descrição
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).
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 do 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.

Elemento	Tipo	Descrição
use_iconmap	integer	Usar mapeamento de ícones para este elemento. Relevante apenas se o mapeamento de ícones 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.
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 desabilitado.
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 de 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).

6 Tipos de mídia

Visão geral

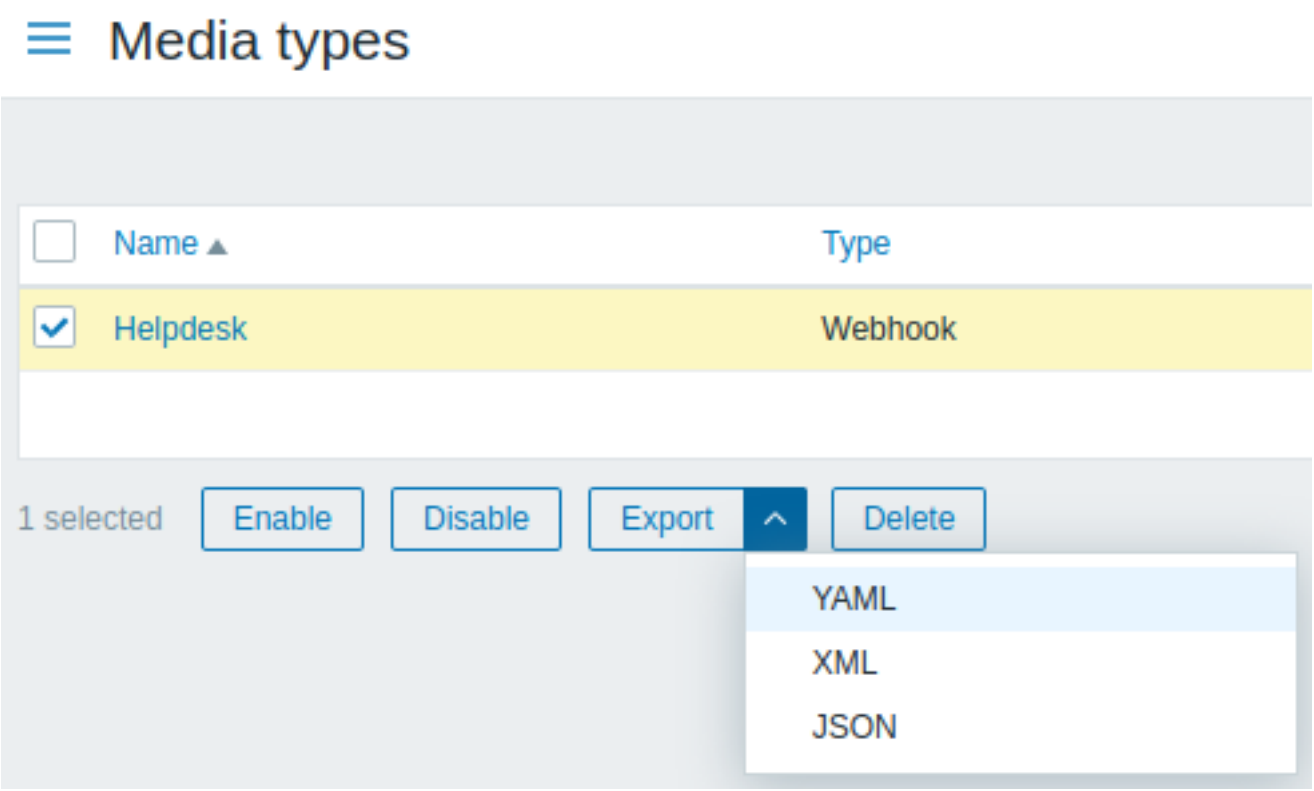
Os tipos de mídia são exportados com todos os objetos relacionados e relações de objetos.

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.



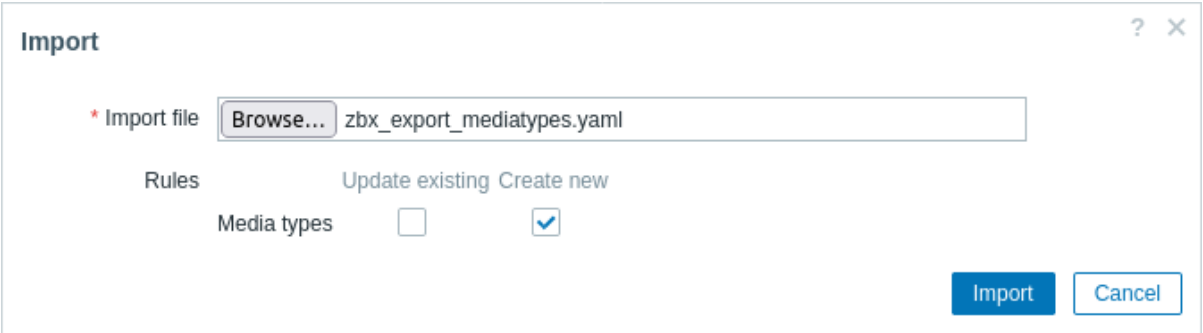
Dependendo do formato selecionado, os tipos de mídia são exportados para um arquivo local com um nome padrão:

- `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:

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
```

```
            {name: 'average', color: '#FFA059'},
            {name: 'high', color: '#E97659'},
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw 'field "retry" should be a number with value of at least 30 if "priority" is set
```

```
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    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}
        Gravidade: {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}
        Gravidade: {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}
        Tempo de atividade 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}
        Tempo de atividade 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 reenvio. 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).

Elemento	Tipo	Descrição
operation_mode	string	Modo de operação. Valores possíveis:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Assunto da mensagem.
message	string	Corpo da mensagem.

Note:
Veja também: Objeto de tipo de mídia (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 e-mail.
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 e-mail 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), OAUTH (2).
username	string	Nome de usuário.
password	string	Senha de autenticação.
redirection_url	string	URL do frontend do Zabbix para redirecionar a autorização OAuth.
client_id	string	Identificador do cliente registrado no servidor de autorização OAuth.
authorization_url	string	URL OAuth, com parâmetros, para obter tokens de acesso e atualização.
token_url	string	URL OAuth para trocar o token de autorização por tokens de acesso e atualização.
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.

Elemento	Tipo	Descrição
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 nome correspondente).

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 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.

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 de topologia de 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 possui um conjunto de verificações de serviço definidas para serem executadas 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 2000000 ou aproximadamente 4 GB de memória. Se a fila ficar cheia, a regra de descoberta será ignorada e uma mensagem de aviso será registrada 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 de ações relevantes, 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. Tais 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), pois um host deve pertencer a um grupo de hosts.

O endereço IP do dispositivo descoberto é o critério para encontrar um host no sistema. Se já existir um host com esse endereço IP e tipo de interface, esse host será o alvo para a execução das operações.

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 do 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** adicionado 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 de string retornado 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 de string retornado 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) é usado.

Se já existir um host com o endereço IP descoberto, um novo host não é criado. No entanto, se a ação de descoberta contiver operações (vincular template, adicionar ao grupo de hosts, etc.), elas serão executadas no host existente.

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 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 de 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 de critérios de exclusividade idênticos, 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. "Tempo de atividade/Tempo de inatividade" não é medido para IPs que são considerados interfaces adicionais.

Alterando a configuração do proxy

Os hosts descobertos por diferentes proxies são sempre tratados como hosts diferentes. Embora isso permita realizar a descoberta em intervalos de IP correspondentes usados por diferentes sub-redes, alterar o proxy para uma sub-rede já monitorada é complicado porque as alterações de proxy também devem ser aplicadas a todos os hosts descobertos.

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
5. habilite a regra de descoberta

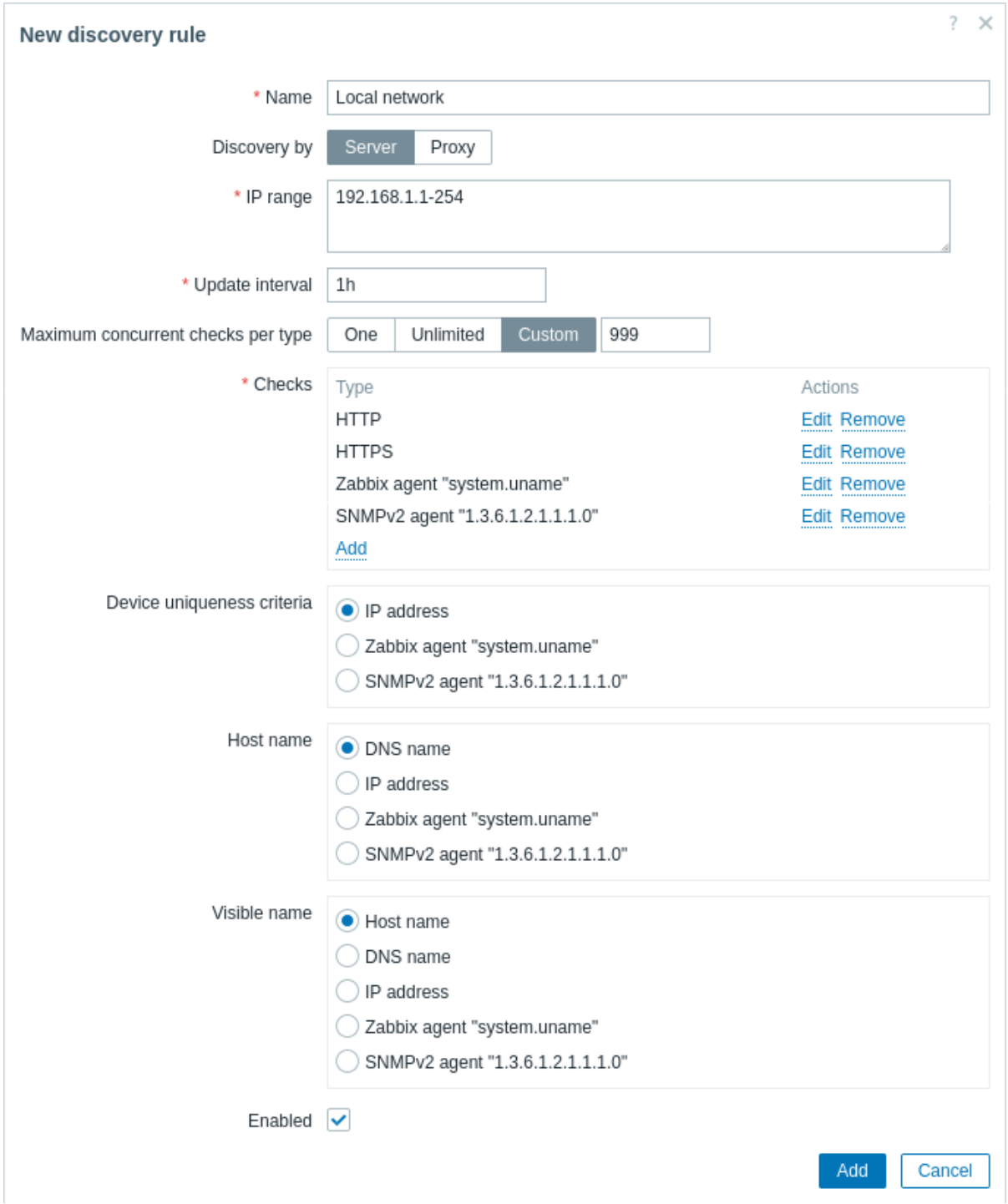
1 Configurando uma regra de descoberta de rede

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: **Server** - 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 processar dispositivos de IP único múltiplos. 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 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 workers de descoberta * 1000. Por padrão, existem 5 workers de descoberta, 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 a rede local com um intervalo de IP de 192.168.1.1-192.168.1.254.

Em nosso cenário, queremos:

- descobrir os hosts que possuem o agent Zabbix em execução
- executar 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 ”Linux servers”
- adicionar hosts Windows ao grupo ”Windows servers”
- usar o template *Linux* para hosts Linux
- usar o template *Windows* para hosts Windows

Passo 1

Definindo uma regra de descoberta de rede para nosso intervalo de IPs.

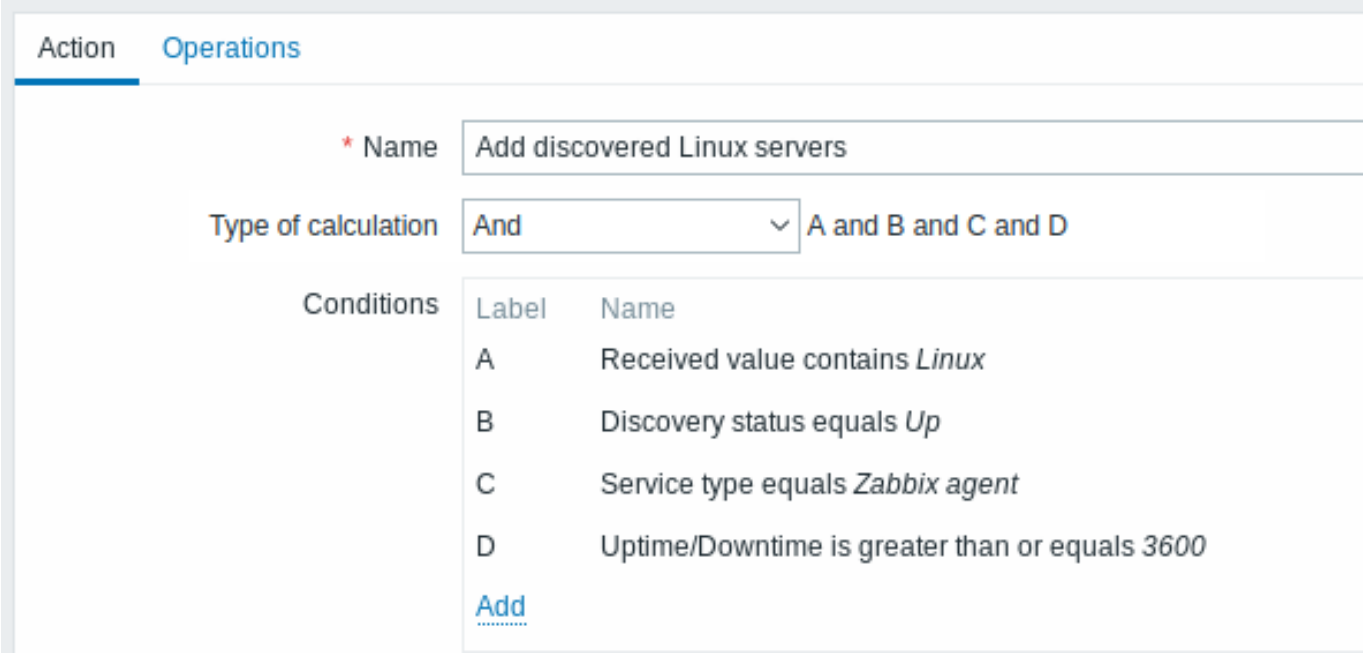
O Zabbix tentará descobrir hosts no intervalo de IPs 192.168.1.1-192.168.1.254 conectando-se aos agents do Zabbix e obtendo o valor da chave **system.uname**. O valor recebido do agent pode ser usado para nomear os hosts e também para aplicar diferentes ações para diferentes sistemas operacionais. Por 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.

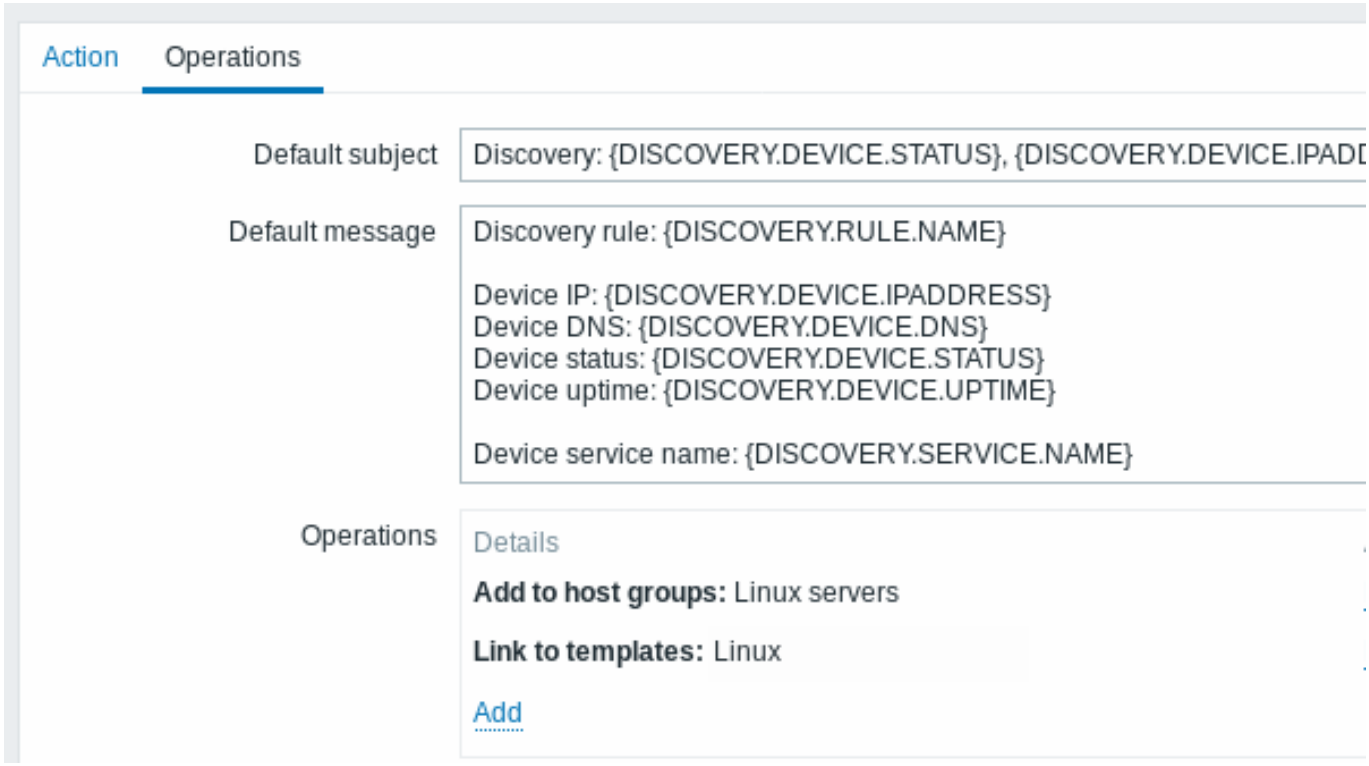
Passo 2

Definindo uma ação de descoberta para adicionar os servidores Linux descobertos ao respectivo grupo/template.



A ação será ativada se:

- o serviço ”Zabbix agent” estiver ”up”
- o valor de system.uname (a chave do Zabbix agent que usamos na definição da regra) contiver ”Linux”
- Uptime for de 1 hora (3600 segundos) ou mais



A ação executará as seguintes operações:

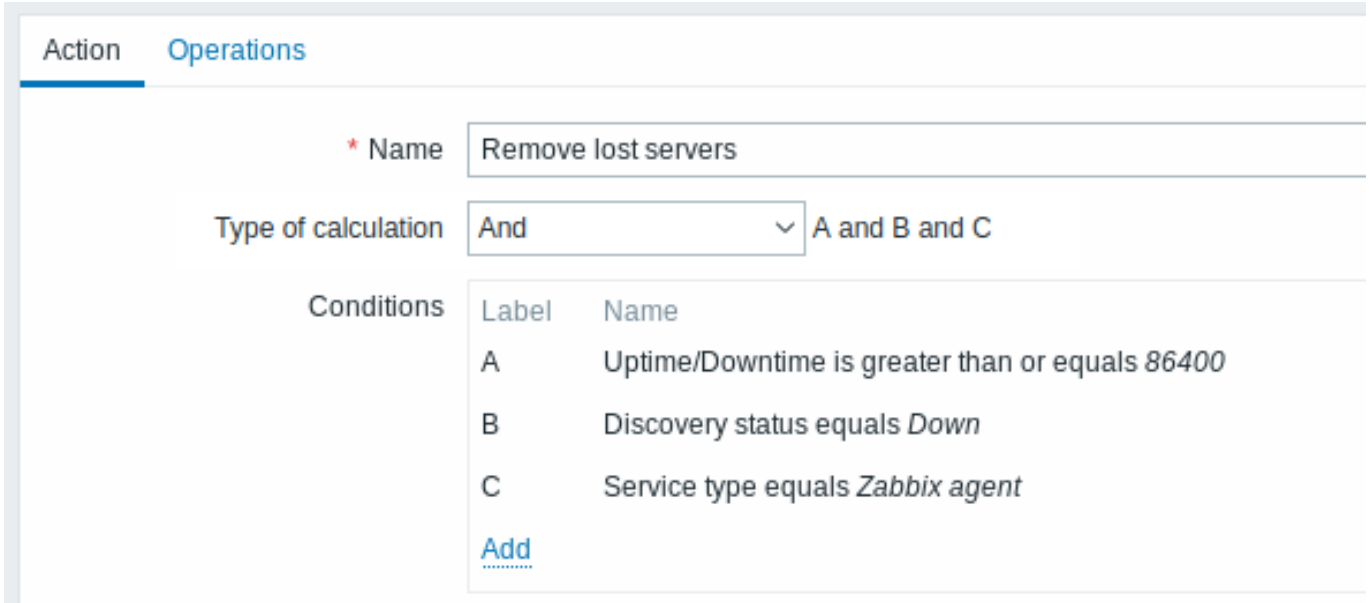
- adicionar o host descoberto ao grupo ”Linux servers” (e também adicionar o host se ele não tiver sido adicionado anteriormente)
- vincular o host ao *Linux* template. O Zabbix começará automaticamente a monitorar o host usando os items e triggers do template ”Linux”.

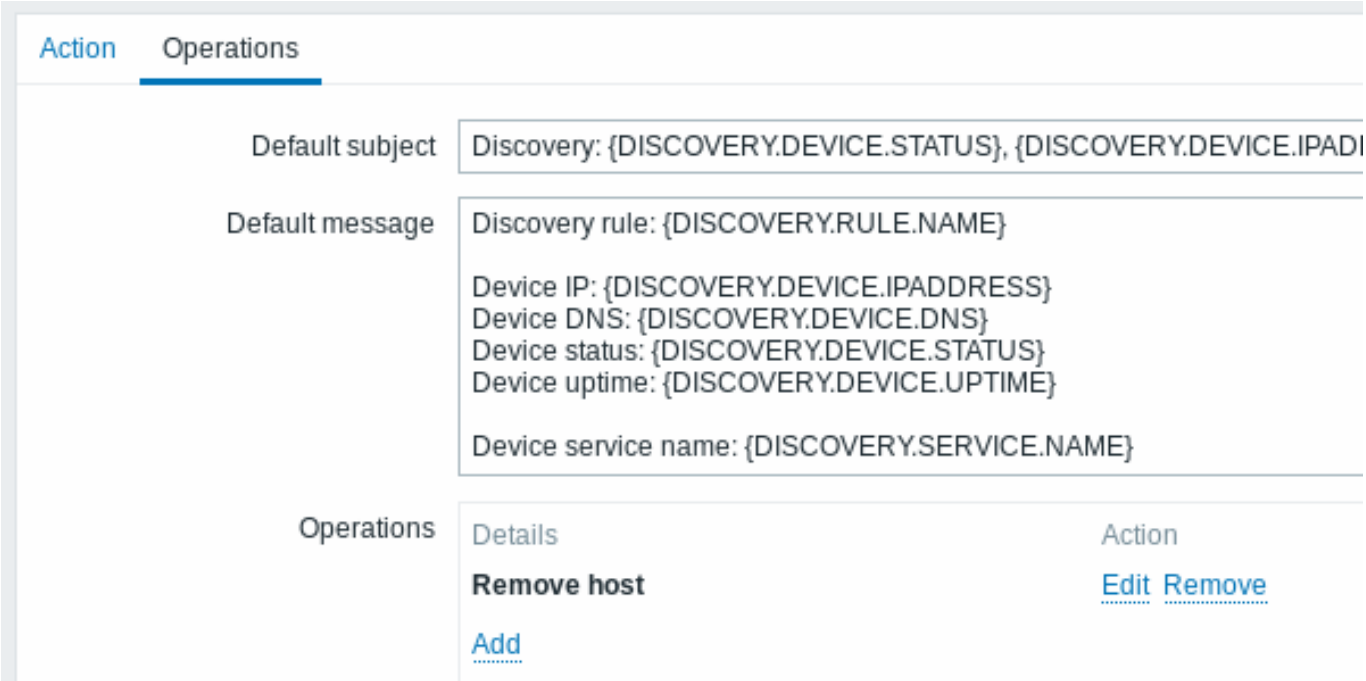
Passo 3

Definindo uma ação de descoberta para adicionar os servidores Windows descobertos ao grupo/template respectivo.

Passo 4

Definindo uma ação de descoberta para remover servers perdidos.





Um server será removido se o serviço ”Zabbix agent” estiver ’down’ por mais de 24 horas (86400 segundos).

2 Autoregistro de agent ativo

Visão geral

É possível permitir a autorregistro de agent ativo 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 previamente 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 tenha os parâmetros de configuração ’ListenIP’ ou ’ListenPort’ 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, o utilizado para a conexão de entrada é usado. Se nenhum valor de porta for recebido, 10050 é usado.

É possível especificar que o host deve ser autorregistrado com um nome DNS como a interface padrão do agent.

O autorregistro é reexecutado:

- se as informações de metadados do host mudarem:
  - 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 autorregistro do agent ativo para o server e proxy do Zabbix é de 120 segundos. Portanto, caso um host descoberto seja excluído, o autorregistro será reexecutado em 120 segundos.

Configuração

Especificar o server

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 hostname do sistema onde o agent está localizado será usado pelo server para nomear o host. O hostname 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 hostnames 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 "Autoregistro" deve ser configurada para o autoregistro de agent.

> **Note:**
> Configurar 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 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 "Metadados do host", veja a próxima seção.
- Na aba Operações, adicione as operações relevantes, como - 'Adicionar host', 'Adicionar ao grupo de hosts' (por exemplo, *Hosts descobertos*), '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 *Hosts descobertos* (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 "Hosts descobertos") e também adicione uma operação *Adicionar ao grupo de hosts* (especificando outro grupo de hosts), pois 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 (para 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-sucedida, 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 habilitada.

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 interface de agent padrão em vez de seu endereço IP. Nesse caso, o nome DNS deve ser especificado ou retornado como 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 consulta 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 em nuvem Amazon) um hostname não é suficiente para o Zabbix server diferenciar 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 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.

Exemplo 1

Usando metadados de host para distinguir entre hosts Linux e Windows.

Suponha que você gostaria que os hosts fossem registrados automaticamente pelo Zabbix server. Você possui agents Zabbix ativos (veja a seção ”Configuração” acima) em sua rede. Existem hosts Windows e hosts Linux em sua rede e você possui 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 para o server durante o registro automático, o que pode não ser suficiente. Para garantir que o template correto seja aplicado ao host, você deve usar metadados de 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

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

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.

Exemplo 2

Passo 1

Usando metadados do host para permitir alguma proteção básica contra hosts indesejados se registrando.

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: E
  - 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

Observe que este método sozinho não fornece proteção forte, pois 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.

Passo 2

É possível adicionar monitoramento adicional para um host já registrado.

Configuração do frontend

Atualize a ação no frontend:

- Nome: Autoregistration action 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: Linux servers
  - Associar templates: Linux
  - Associar templates: MySQL by Zabbix Agent

Configuração do agent

Atualize a próxima 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 fornece uma maneira de criar automaticamente items, triggers e gráficos para diferentes entidades em um computador. Por exemplo, o Zabbix pode iniciar automaticamente o monitoramento de 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. Além disso, é possível configurar o Zabbix para remover entidades desnecessárias automaticamente com base nos resultados reais de descobertas realizadas 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) o valor desse 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 protótipos onde elas são então substituídas pelos valores recebidos para criar items, triggers, gráficos ou até 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 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 retroativa 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.

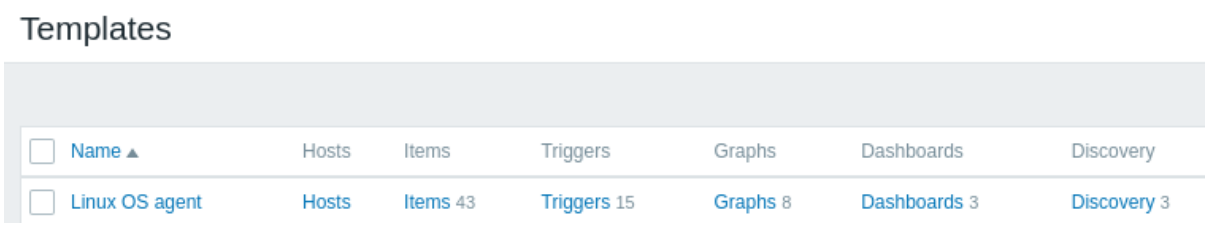
Warning:
Como resultado das alterações acima, agents mais recentes não serão mais capazes de trabalhar com um server Zabbix mais antigo.

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



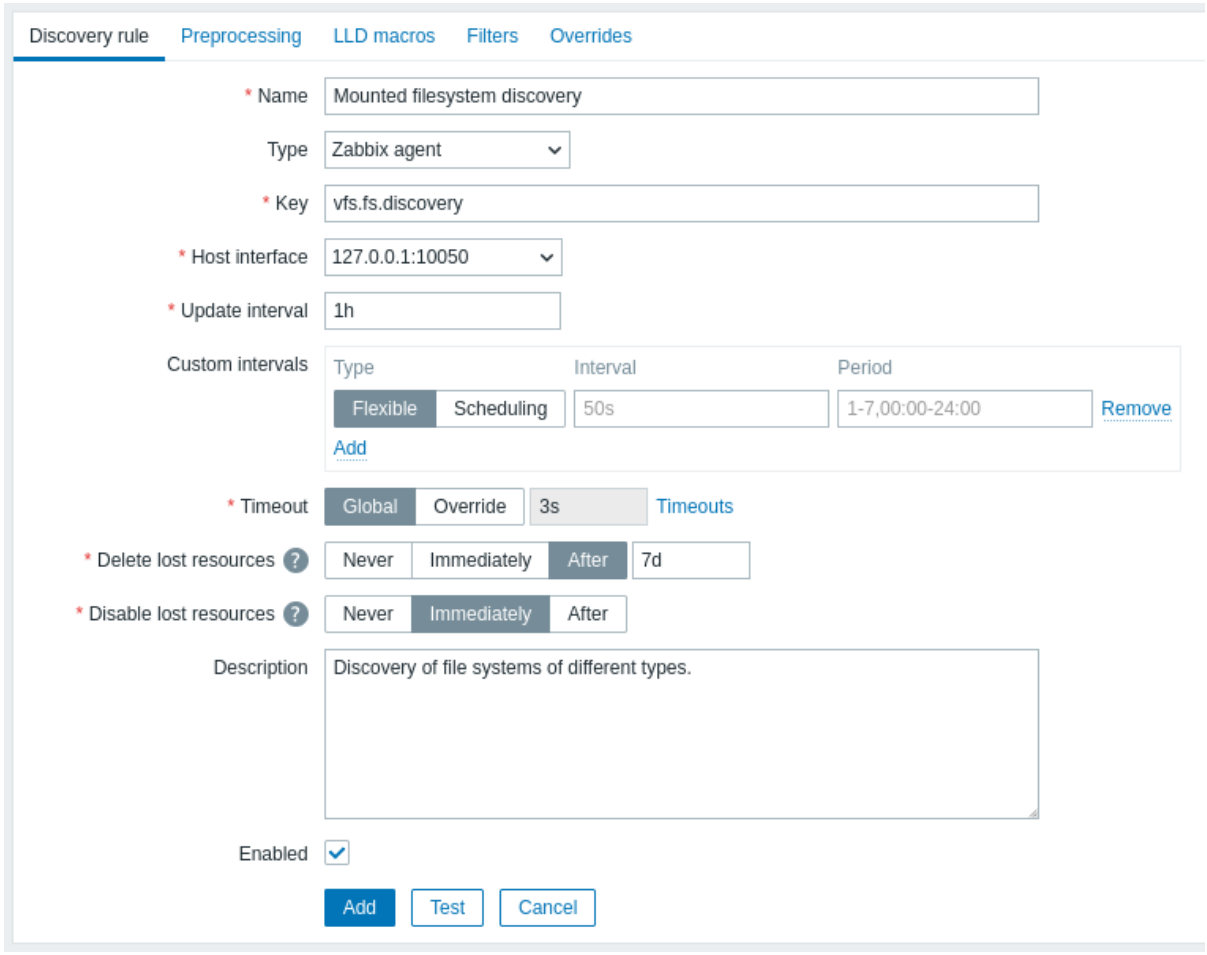
- Clique em *Criar regra de descoberta* no canto superior direito da tela
- Preencha o formulário da regra de descoberta com os detalhes necessários

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.
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*.

Parâmetro	Descrição
Intervalos personalizados	Você pode criar regras personalizadas para verificar 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.

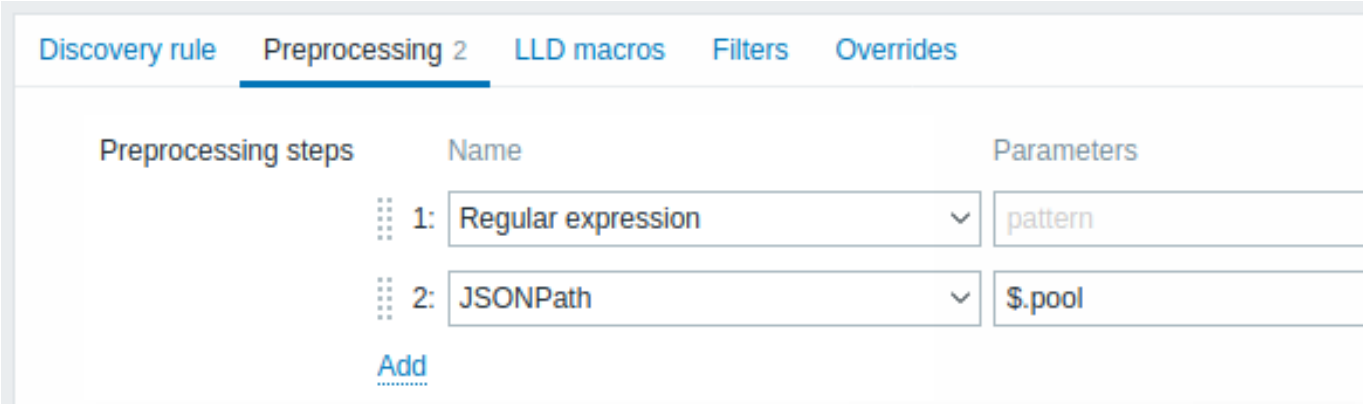
Note:
O histórico da regra de descoberta não é preservado.

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
- Teste de 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	Localiza 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 localizada 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.

	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.

Tipo	
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 no 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: **Sem alteração** - 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 OID 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 utilizado. Para mais informações, veja: Pré-processamento Javascript
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.

Tipo	
Verificar erro em JSON	Verifica uma mensagem de erro em nível de aplicação localizada em JSONPath. Para 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. Para 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.

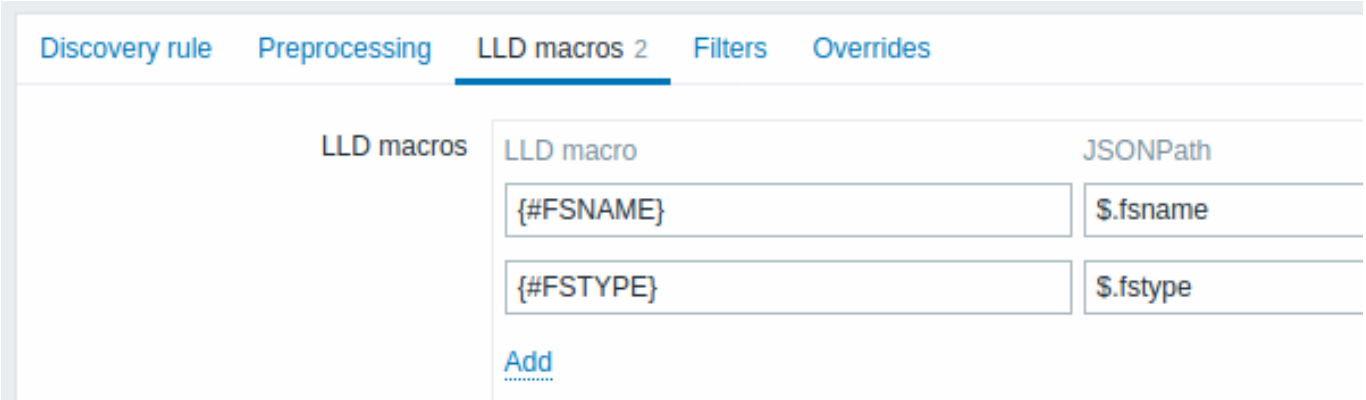
Observe que se a regra de descoberta foi aplicada ao host via template, o conteúdo desta aba é somente leitura.

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 pré-definidas, 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 de sistema de arquivos, mas sem nenhuma macro LLD pré-definida. Neste caso, você pode definir as macros manualmente e mapeá-las para os valores no JSON usando JSONPath:

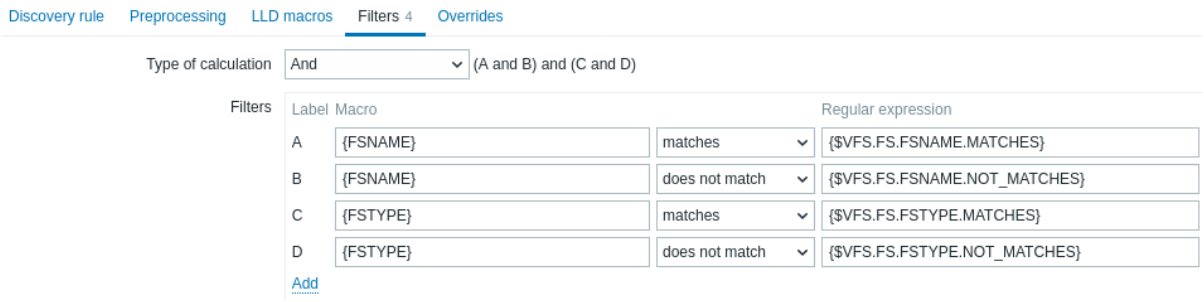


Os valores extraídos podem ser usados em itens, triggers, etc. descobertos. Observe que os valores serão extraídos do resultado da descoberta e de quaisquer etapas de pré-processamento até o momento.

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 de 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 retroativa 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 atendidos; **Ou** - basta que um filtro seja atendido; **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 **Substituição** permite definir regras para modificar a lista de protótipos de item, trigger, gráfico, host e descoberta 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 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 (protótipo de item/protótipo de trigger/protótipo de gráfico/protótipo de host) e uma condição a ser atendida (igual/não igual/contém/não contém/corresponde/não corresponde) **Ações** - links para editar e remover uma operação são exibidos.

Configurando uma operação

Para configurar os detalhes de uma nova operação, clique em no bloco Operações. Para editar uma operação existente, clique em 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 Protótipo de descoberta
Condição	Permite filtrar as entidades às quais a operação deve ser aplicada.
Operador	Operadores suportados: **igual** - aplica a este protótipo **não igual** - aplica a todos os protótipos, exceto este **contém** - aplica se o nome do protótipo contiver esta string **não contém** - aplica se o nome do protótipo não contiver esta string **corresponde** - aplica se o nome do protótipo corresponder à expressão regular **não corresponde** - aplica 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 para especificar intervalos flexíveis/agendados. Para informações detalhadas, consulte Intervalos personalizados.
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 suportadas.
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 suportadas.
Tags	Quando a caixa de seleção estiver marcada, um novo bloco aparecerá, permitindo especificar pares de tag-valor. Essas tags serão adicionadas às tags especificadas no protótipo de item, mesmo que os nomes das tags coincidam.
Objeto: *Protótipo de trigger*	

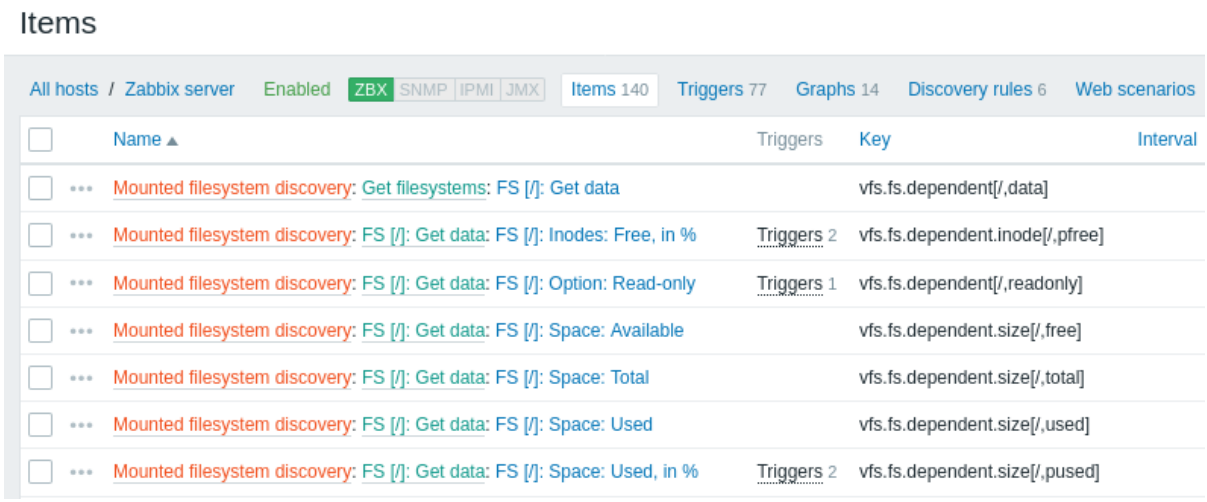
Parâmetro		Descrição
	Criar habilitado	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de trigger: *Sim* - o trigger será adicionado em estado habilitado. *Não* - o trigger será adicionado a uma entidade descoberta, mas em 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.
	Severidade	Quando a caixa de seleção estiver marcada, os botões de severidade do trigger aparecerão, permitindo modificar a severidade do trigger.
	Tags	Quando a caixa de seleção estiver marcada, um novo bloco aparecerá, permitindo especificar pares de 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.
	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 de 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 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. *Nota* 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 dos parâmetros) sem aplicar permanentemente quaisquer alterações.
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 desabilitadas e eventualmente excluídas.

Recursos perdidos podem ser automaticamente desabilitados com base no valor do parâmetro *Desabilitar recursos perdidos*. Isso afeta hosts, items e triggers perdidos.

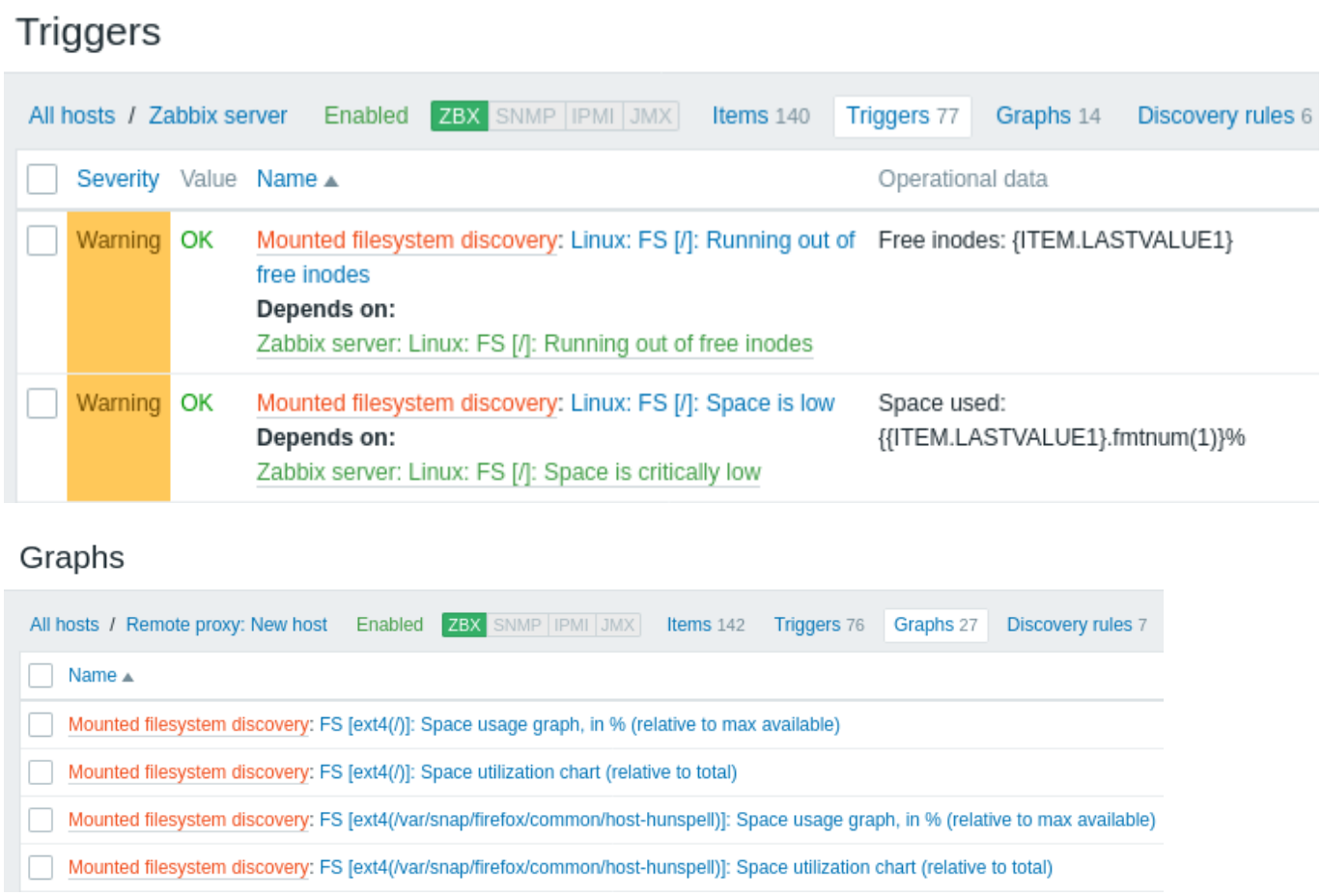
Recursos perdidos podem ser automaticamente excluídos com base no valor do parâmetro *Excluir recursos perdidos*. Isso afeta hosts, grupos de hosts, items, triggers e gráficos perdidos.

Quando as entidades descobertas se tornam 'Não descobertas mais', um indicador de tempo de vida é exibido na lista de entidades. Passe 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 desabilitados), 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:

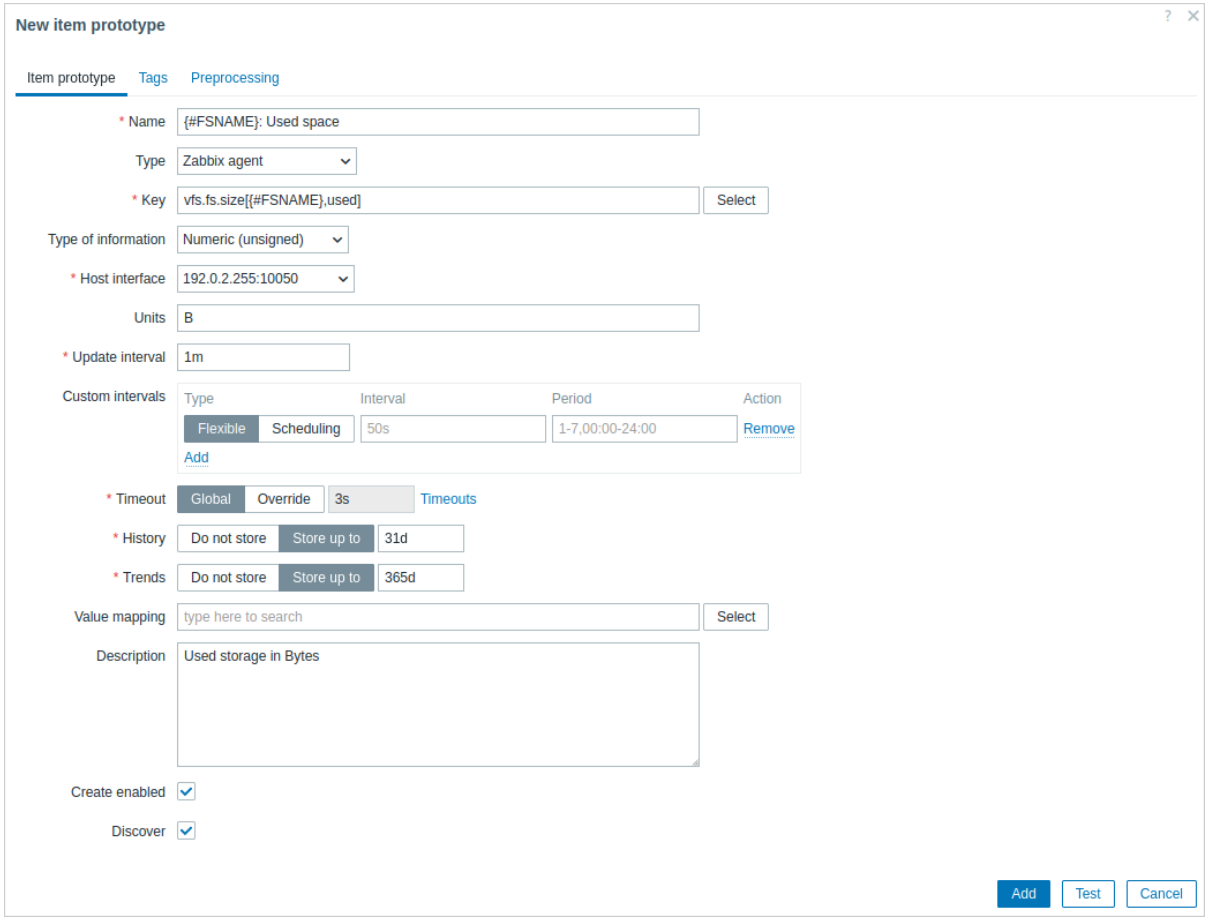
- descoberta de interfaces de rede;
- descoberta de CPUs e núcleos de CPU;
- descoberta de OIDs SNMP;
- descoberta de objetos JMX;
- descoberta usando consultas SQL ODBC;
- descoberta de serviços do Windows;
- descoberta de interfaces de host no Zabbix.

Para mais detalhes sobre o formato JSON para itens de descoberta e um exemplo de como implementar seu próprio descobridor de sistema de arquivos como um script Perl, consulte criando regras LLD personalizadas.

1 Prototipagem de items

Uma vez que uma regra é criada, vá para os items dessa regra e pressione ”Criar protótipo de item” para criar um protótipo de item.

Note 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.



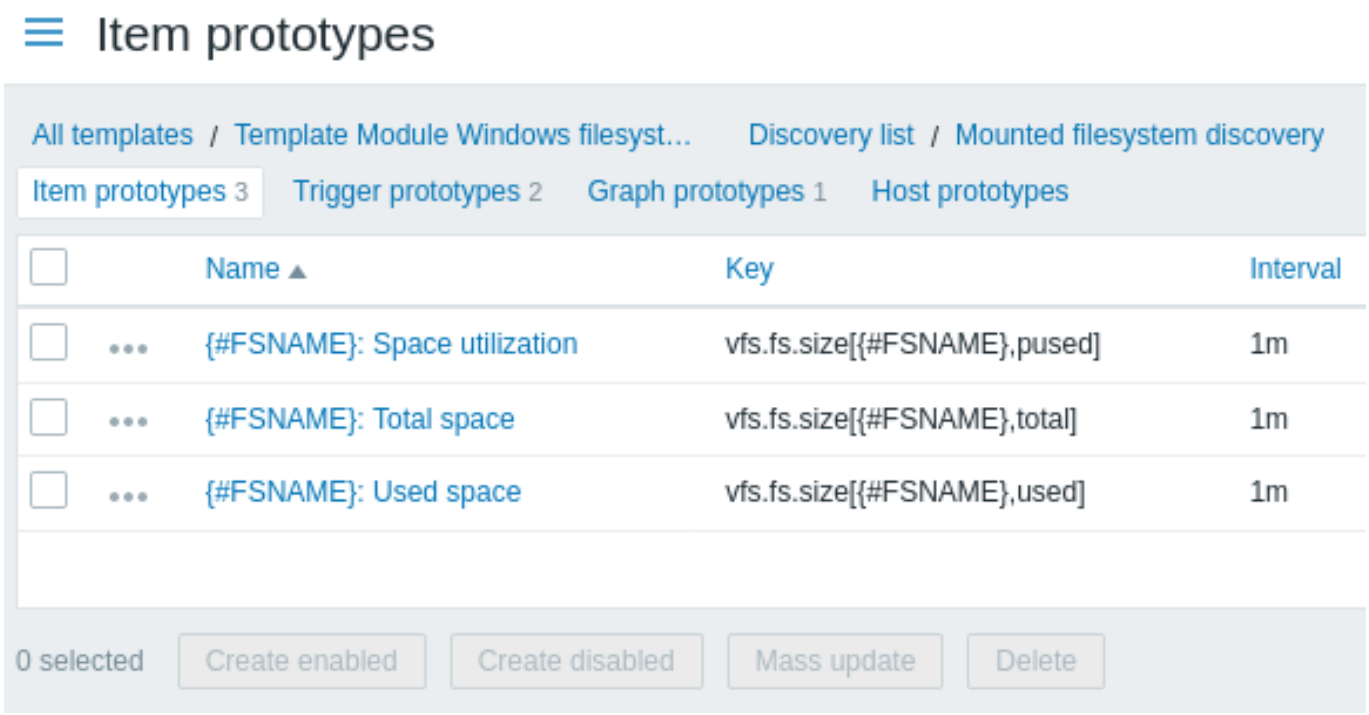
Macros de descoberta de baixo nível e macros de usuário são suportadas na configuração do protótipo de item e nos parâmetros de pré-processamento do valor do item. Observe que, quando usadas em intervalos de atualização, uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas.

Note:
O escape específico de contexto das 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 sobrescrita na regra de descoberta.

Podemos criar vários protótipos de item para cada métrica de sistema de arquivos que nos interessa:



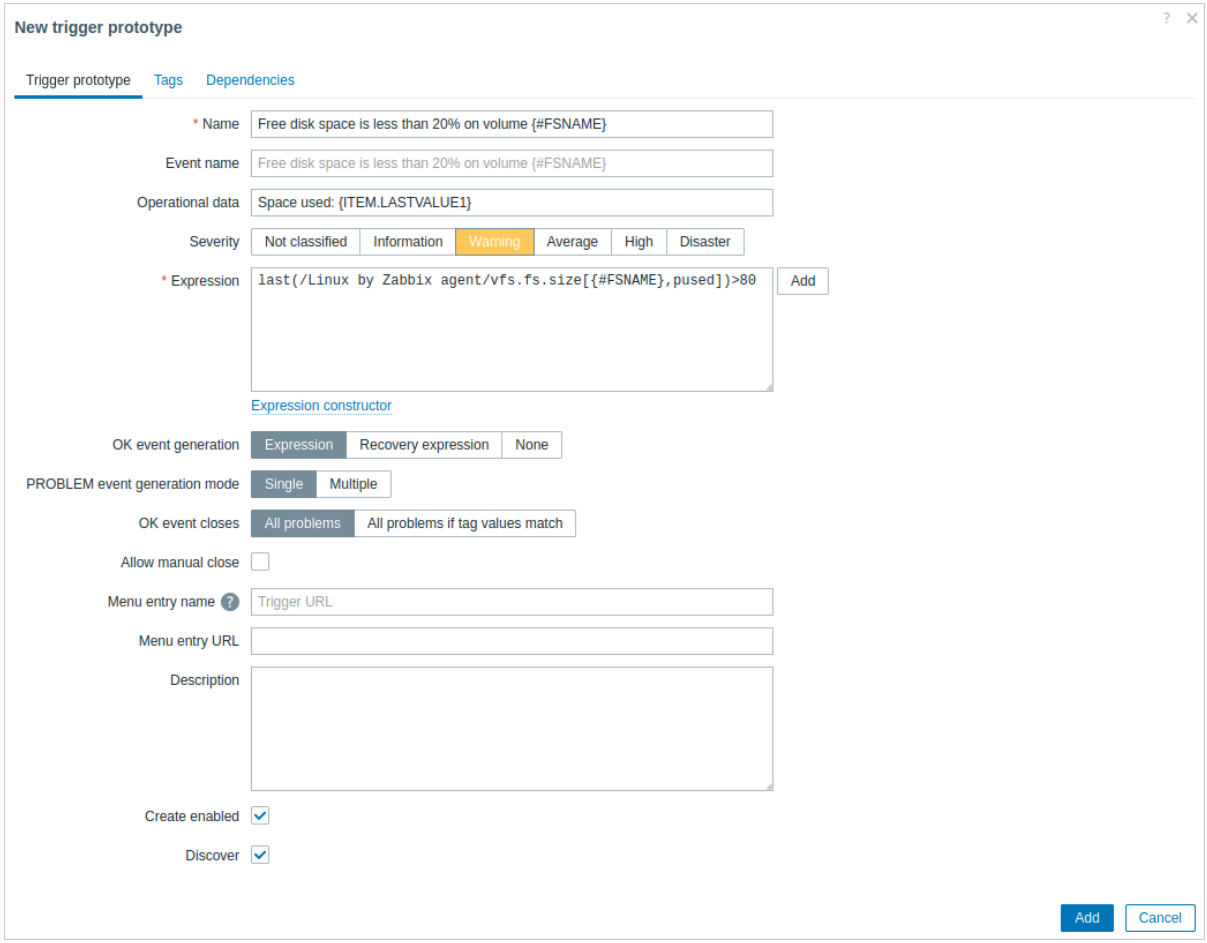
Clique no ícone de três pontos para abrir o menu para o protótipo de item específico com estas opções:

- *Criar protótipo de trigger* - cria um protótipo de trigger baseado 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* - cria um item dependente para este protótipo de item

A opção de *Atualização em massa* está disponível caso você queira atualizar 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:

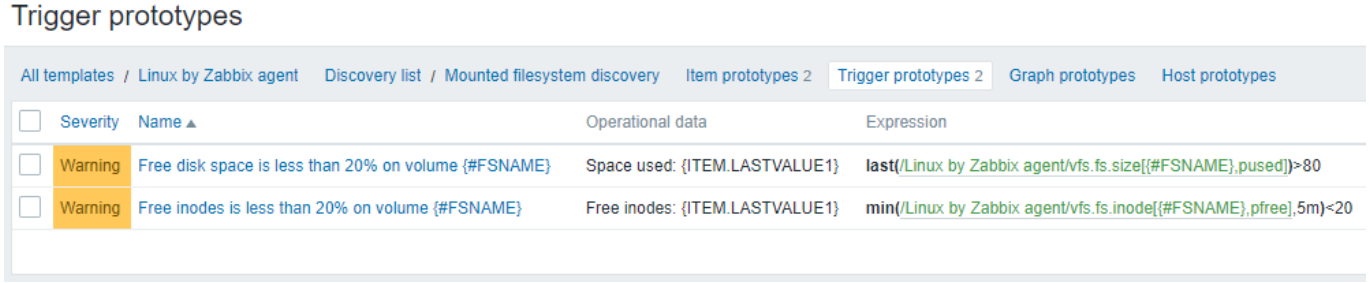


Atributos específicos para protótipos de trigger:

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.

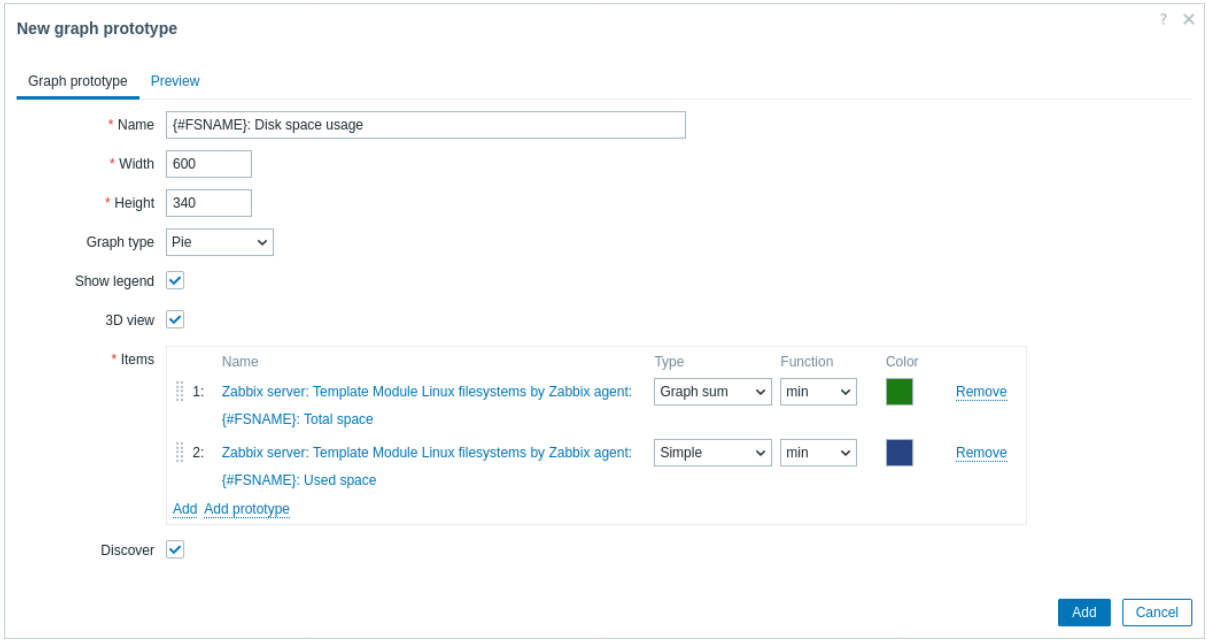
Quando triggers reais são criados a partir dos protótipos, pode ser necessário ser flexível quanto à constante ('20' em nosso exemplo) usada para comparação na expressão. Veja como macros de usuário com contexto podem ser úteis para alcançar tal flexibilidade.

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.



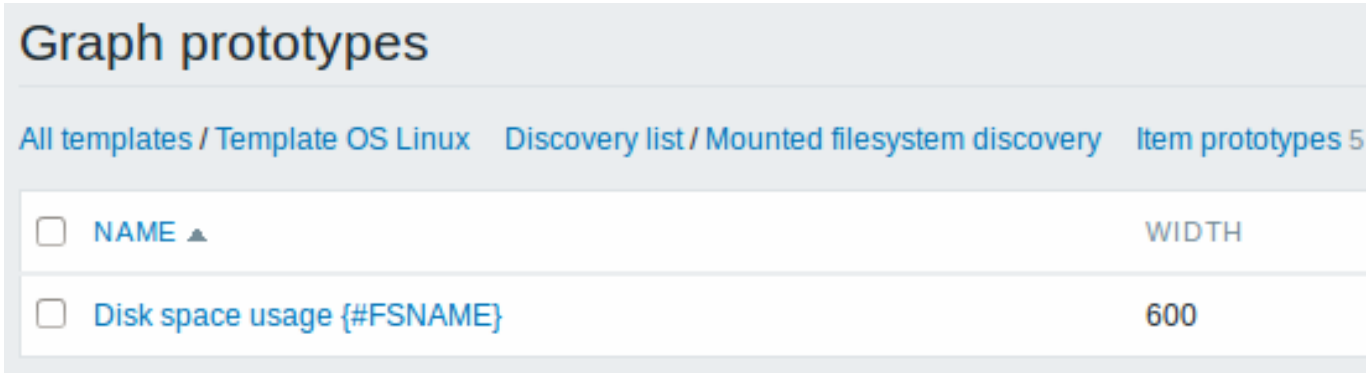
3 Protótipos de gráficos

Também podemos criar protótipos de gráficos:

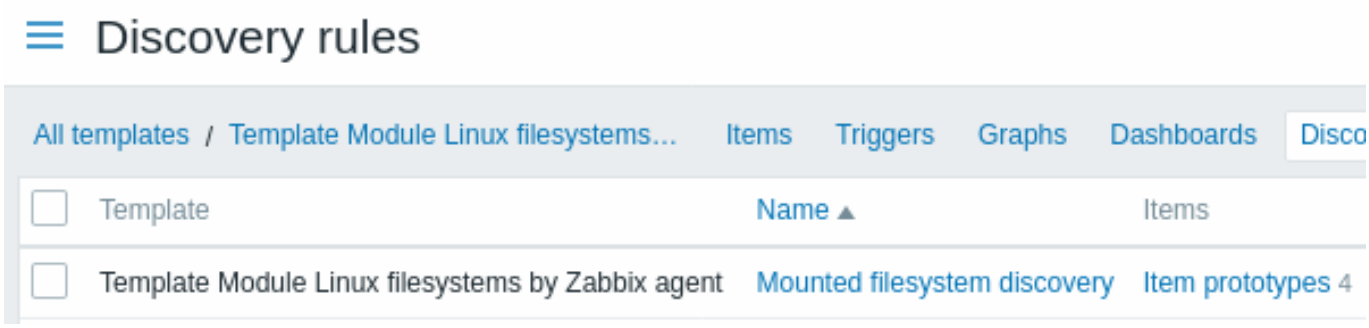


Atributos específicos para protótipos de gráficos:

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 se parece com a mostrada 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

Prototipos de host são configurados em regras de descoberta de baixo nível.

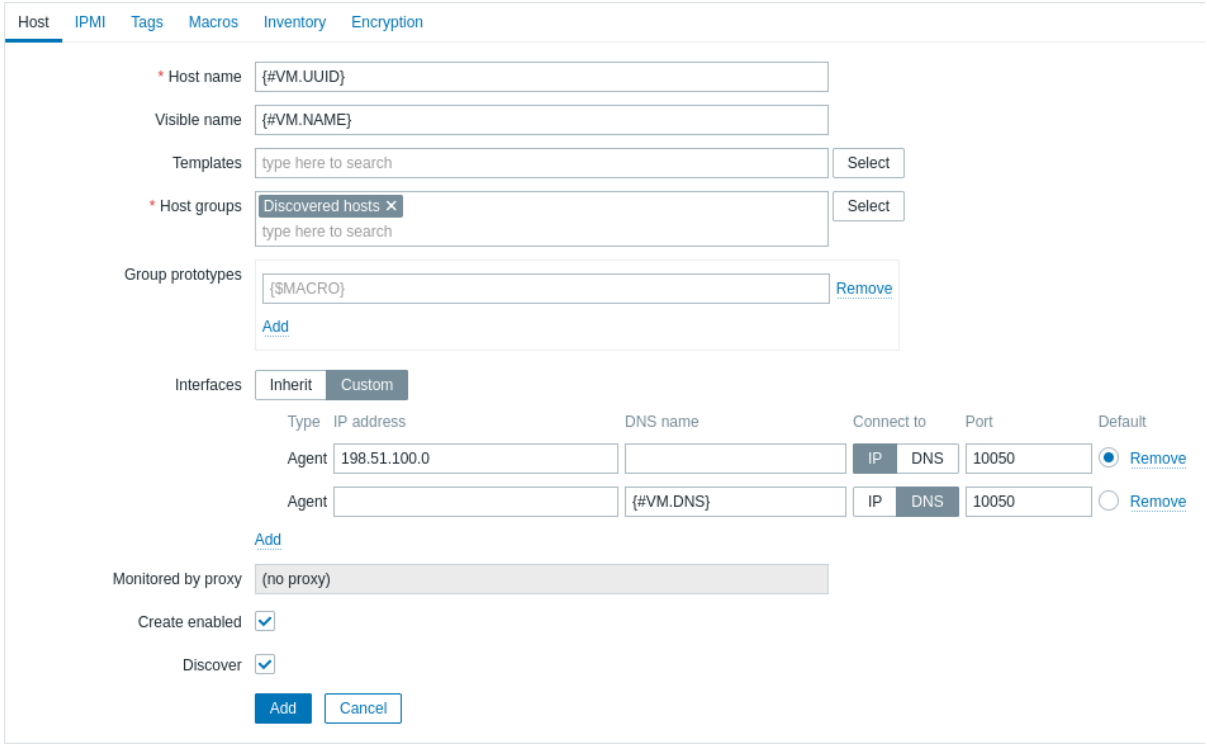
Para criar um protótipo de host:

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 foram 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 do 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 *Interface* de "Herdar" para "Personalizado". Clique em Add e selecione o tipo de interface - agent Zabbix, 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 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 templates 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 tags adicionadas manualmente. As tags herdadas de um protótipo de host não podem ser removidas. Observe que as tags não podem ter duplicatas (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 macros adicionadas manualmente, bem como aquelas herdadas de um protótipo de host; alterar valores e tipos de macro.

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.

Hosts descobertos também podem descobrir outros hosts. Por exemplo, se um protótipo de host (usado para descobrir hipervisores) tiver um template vinculado a ele que inclua uma regra de descoberta de baixo nível com seu próprio protótipo de host (usado para descobrir máquinas virtuais), o Zabbix descobrirá tanto os hipervisores quanto suas máquinas virtuais. Para descobrir adicionalmente containers nessas VMs, você pode criar regras de descoberta de baixo nível ou vincular um novo template com protótipos de host às VMs descobertas ou pré-configurar o protótipo de host da VM vinculando um template que inclua protótipos de host.

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).

5 Protótipos de descoberta

Visão geral

Prototipos de descoberta são regras de **descoberta de baixo nível aninhadas** dentro de uma regra de descoberta "pai", permitindo criar uma descoberta multinível de objetos com seus próprios items, triggers, etc. Por exemplo, você pode querer descobrir todas as instâncias de banco de dados em um servidor de banco de dados, depois descobrir tablespaces para cada instância, depois descobrir tabelas para cada tablespace.

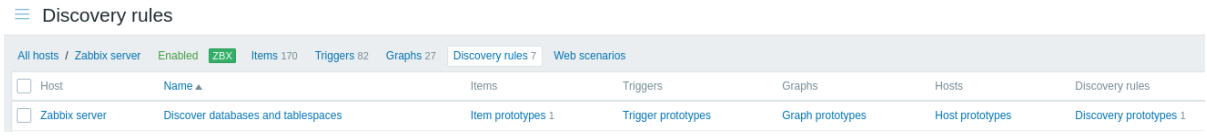
Prototipos de descoberta possuem seus próprios prototipos de item, trigger, gráfico, host e descoberta. Um prototipo de descoberta aninhado usará o mesmo valor JSON da regra pai se você especificar um tipo *Aninhado*.

Os níveis de aninhamento para prototipos de descoberta são ilimitados.

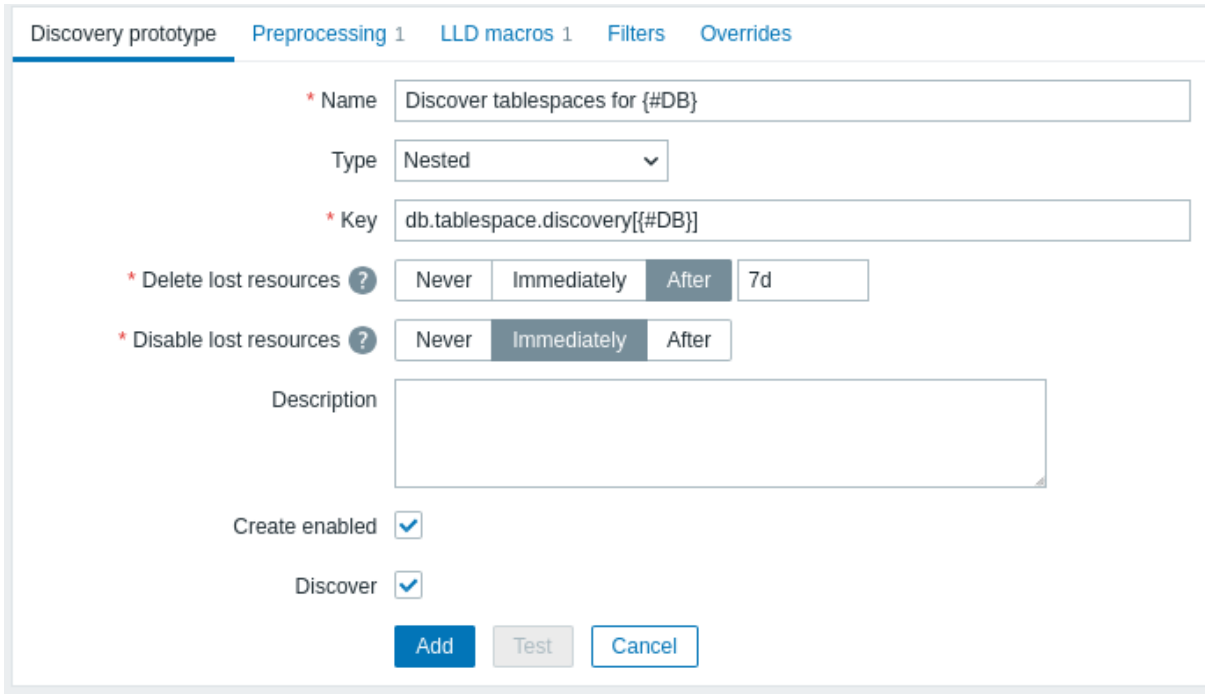
Configuração

Para criar um protótipo de descoberta:

- Clique em *Protótipos de descoberta* na linha de uma regra de descoberta existente



- Clique em *Criar protótipo de descoberta*



Os campos de configuração deste formulário são compartilhados com a descoberta de baixo nível regular.

Se você selecionar ”Aninhado” como *Tipo* no formulário de protótipo de descoberta aberto, então as regras de descoberta (do protótipo de descoberta) são geradas com base em um objeto JSON do mesmo valor JSON que a regra de descoberta pai. Por exemplo, se o JSON original for [<objeto A>, <objeto B>] e houver um protótipo de regra de descoberta aninhada, então duas regras de descoberta seriam geradas com base nos dados do objeto A e do objeto B, respectivamente.

Neste caso, o protótipo de descoberta é ativado ao mesmo tempo que a regra pai. Assim, a regra aninhada pode usar pré-processamento, para trabalhar em outro ”recorte” dos mesmos dados, já adquiridos pela regra pai.

As macros LLD da regra LLD pai estão disponíveis para regras de descoberta aninhadas.

Regras LLD aninhadas em hosts descobertos

Uma regra de descoberta de baixo nível *Aninhada* pode ser usada em um template de host atribuído ao protótipo de host. Se uma regra de descoberta *Aninhada* existir em um host descoberto, o objeto JSON usado para descobrir o host também será enviado para todas as regras LLD do tipo aninhado neste host. Para mais detalhes, veja o exemplo.

As macros LLD da regra de descoberta que criou o host estão disponíveis para regras de descoberta aninhadas.

Exemplo

Vamos ilustrar a possível aplicação de protótipos de descoberta, com base no recebimento do seguinte exemplo de JSON multinível.

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
```

```
  },
  {
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```

Caso 1

Descobrindo instâncias de banco de dados em um servidor de banco de dados e, em seguida, descobrindo os tablespaces para cada instância.

1. Você tem pelo menos um host relacionado à descoberta do servidor de banco de dados.
2. Crie uma regra de LLD para este host chamada *Descobrir bancos de dados e tablespaces*.
3. Altere para a aba *Macros de LLD* nesta regra, adicione a macro `{#DB}=$.database`.
4. Adicione um protótipo de item para esta regra chamado *Conexões ativas para {#DB}* (Tipo: Agent, Chave: `db.connections[{#DB}]`).
5. Os itens relacionados a cada banco de dados são descobertos:

```
Conexões ativas para db1, Conexões ativas para db2, Conexões ativas para db3.
```

6. Crie um protótipo de descoberta para esta regra chamado *Descobrir tablespaces para {#DB}* (Tipo: Aninhado, Chave: `db.tablespace.discovery[{#DB}]`).
7. Altere para a aba *Pré-processamento* deste protótipo de descoberta e adicione a etapa `JSONPath=$.tablespaces`.
8. Altere para a aba *Macros de LLD* deste protótipo de descoberta, adicione a macro `{#TSNAME}=$.name`.
9. Crie um protótipo de item para este protótipo de descoberta chamado *Tamanho do tablespace {#TSNAME} para {#DB}* (Tipo: Agent, Chave: `db.ts.size[{#DB}, {#TSNAME}]`).
10. Os itens relacionados a cada tablespace de cada banco de dados são descobertos:

```
Tamanho do tablespace ts1 para db1, Tamanho do tablespace ts2 para db1, Tamanho do tablespace ts3 para db1
Tamanho do tablespace ts1 para db2, Tamanho do tablespace ts2 para db2, Tamanho do tablespace ts3 para db2
Tamanho do tablespace ts1 para db3, Tamanho do tablespace ts2 para db3, Tamanho do tablespace ts3 para db3
```

com as chaves `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Caso 2

Descobrindo instâncias de banco de dados no servidor de banco de dados, representando-as como hosts descobertos e, em seguida, descobrindo os tablespaces para cada instância.

1. Você tem pelo menos um host (host raiz) relacionado à descoberta do servidor de banco de dados.
2. Crie um template para descobrir os tablespaces de cada banco de dados.
3. Crie um item neste template chamado *Conexões ativas para {#DB}* (Tipo: Agent, Chave: `db.connections[{#DB}]`).
4. Crie uma regra LLD para este template chamada *Descobrir tablespaces* (Tipo: Aninhado).
5. Vá para a aba *Pré-processamento* desta regra e adicione o passo `JSONPath=$.tablespaces`.

6. Vá para a aba *Macros LLD* desta regra, adicione a macro `{#TSNAME}=$.name`.
7. Crie um protótipo de item para esta regra chamado *Tamanho do tablespace {#TSNAME} para {#DB}* (Tipo: Agent, Chave: `db.ts.size[{#DB}, {#TSNAME}]`).
8. De volta ao host raiz, crie uma regra LLD para este host chamada *Descobrir bancos de dados e tablespaces*.
9. Vá para a aba *Macros LLD* nesta regra, adicione a macro `{#DB}=$.database`.
10. Adicione um protótipo de host para esta regra chamado *Host para o banco de dados {#DB}*.
11. Vá para a aba *Macros* neste protótipo de host, adicione a macro `{$DB}={#DB}` (para o nome e chave do item do Passo 3).
12. Vincule o template do passo 2 a este protótipo de host.
13. Os hosts descobertos contêm os itens descobertos relacionados a cada banco de dados e seus tablespaces:

Host	Itens
Host para o banco de dados db1	Conexões ativas para db1
	Tamanho do tablespace ts1 para db1
	Tamanho do tablespace ts2 para db1
	Tamanho do tablespace ts3 para db1
Host para o banco de dados db2	Conexões ativas para db2
	Tamanho do tablespace ts1 para db2
	Tamanho do tablespace ts2 para db2
	Tamanho do tablespace ts3 para db2
Host para o banco de dados db3	Conexões ativas para db3
	Tamanho do tablespace ts1 para db3
	Tamanho do tablespace ts2 para db3
	Tamanho do tablespace ts3 para db3

6 Notas sobre descoberta de baixo nível

Usando macros LLD em contextos de macro de usuário

Macros LLD podem ser usadas dentro do contexto de macro de usuário, por exemplo, em protótipos de trigger.

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 de regras 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.

7 Regras de descoberta

Por favor, 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

Configuração

Item mestre

Crie um item de agent Zabbix usando a seguinte chave:

`vfs.fs.get`

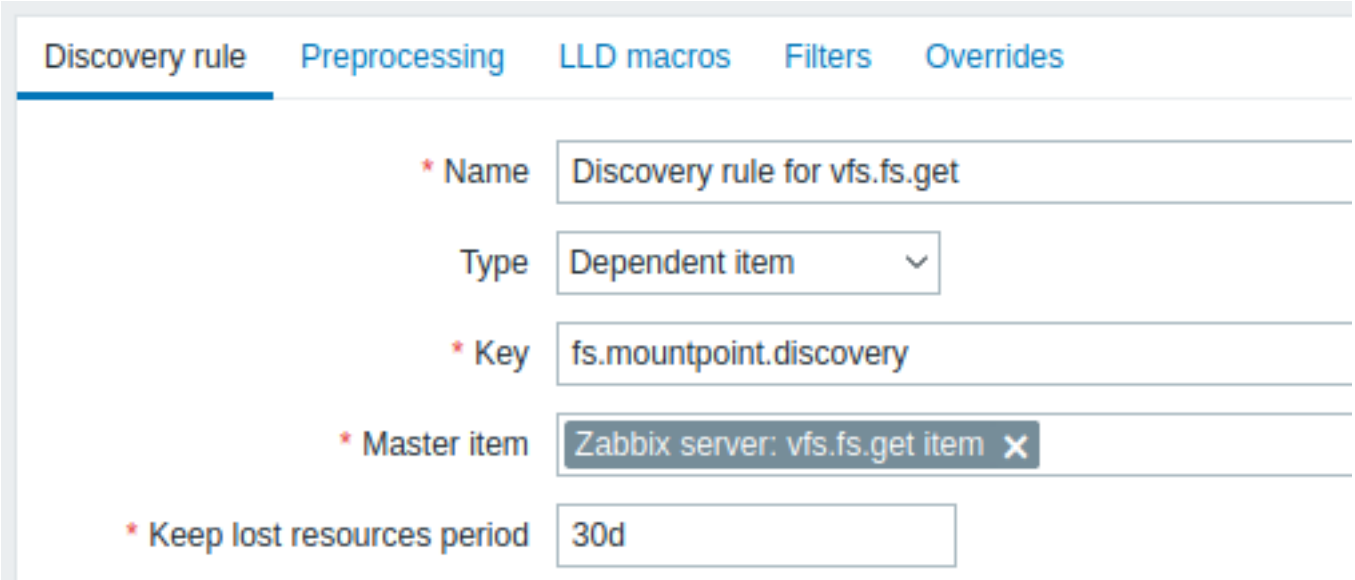
Defina o tipo de informação como "Texto" para possíveis dados JSON grandes.

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"
  }
]
```

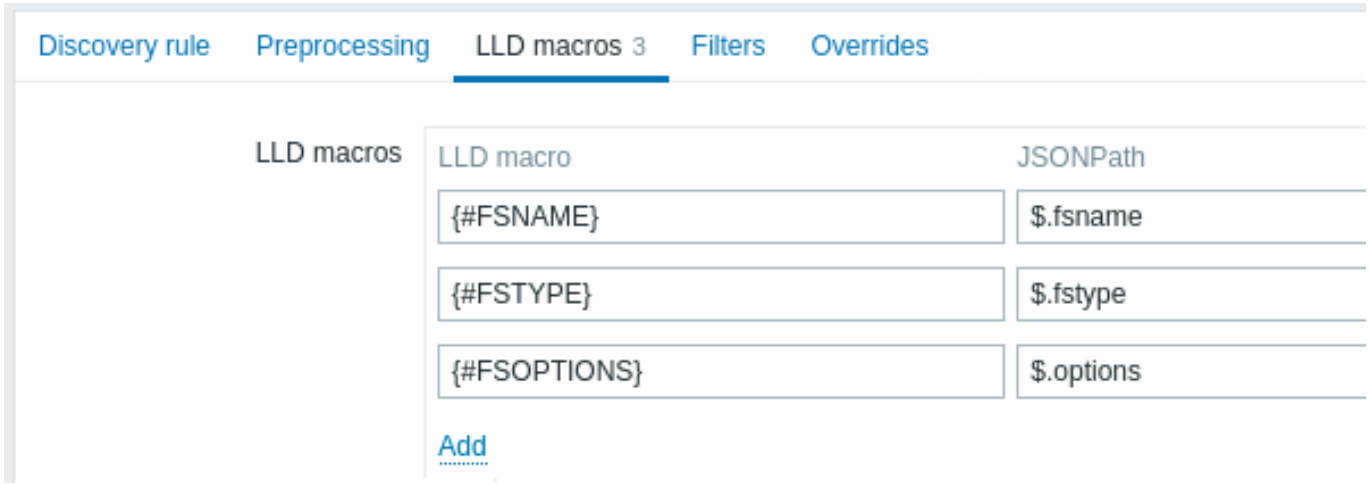
Regra LLD dependente

Crie uma regra de descoberta de baixo nível do tipo "Item dependente":

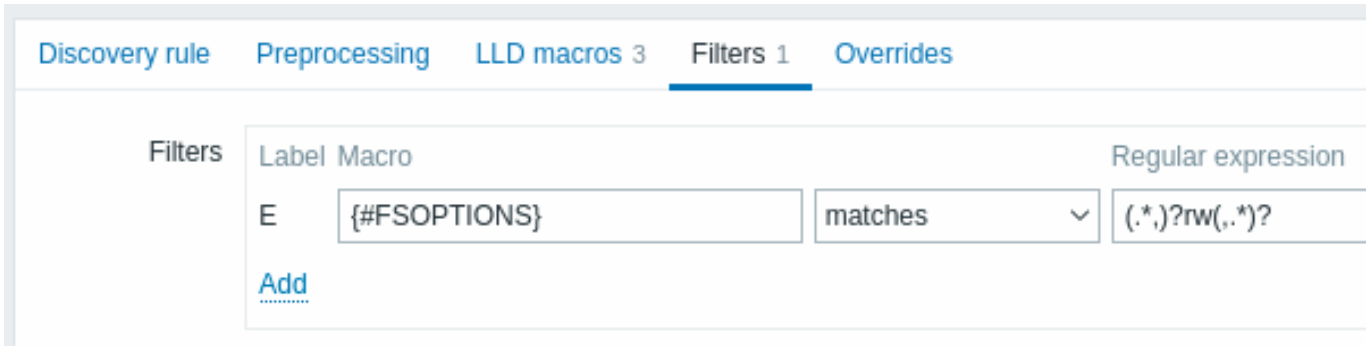


Como item mestre, selecione o item `vfs.fs.get` que criamos.

Na aba "Macros LLD", defina macros personalizadas com o respectivo JSONPath:

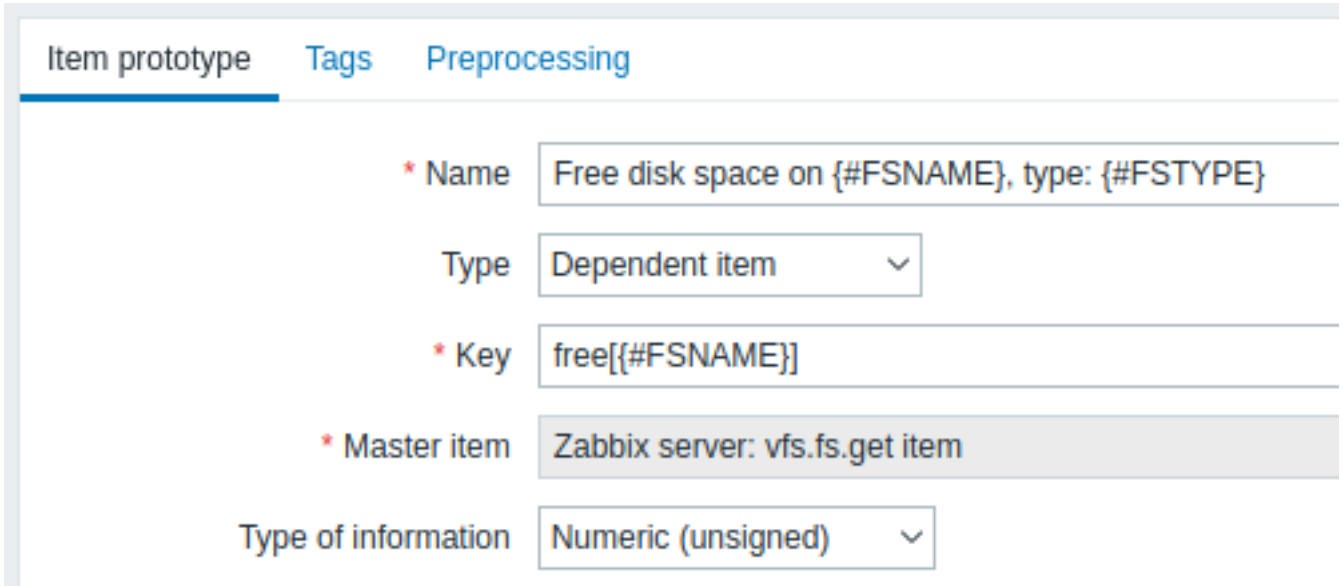


Na aba ”Filtros”, você pode adicionar uma expressão regular que filtre apenas os sistemas de arquivos **read-write**:



Protótipo de item dependente

Crie um protótipo de item com o tipo ”Item dependente” nesta regra de 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 de item:

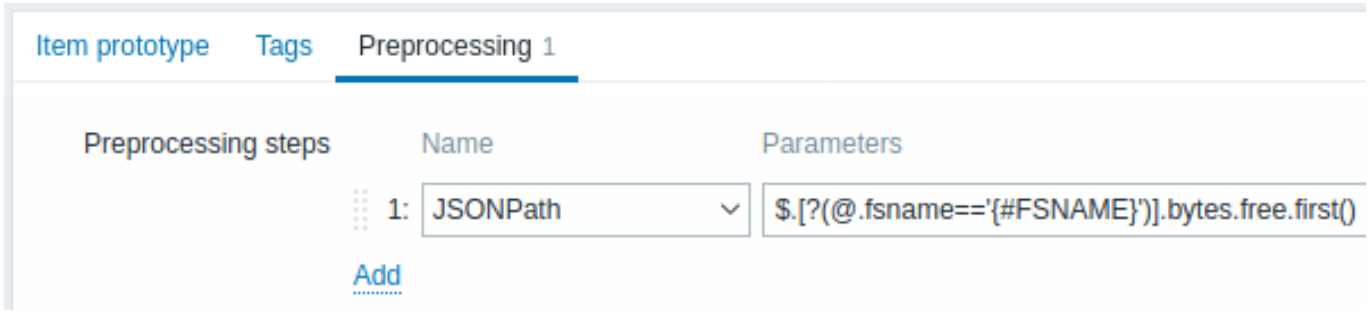
- *Nome*: Espaço livre em disco em `{#FSNAME}`, tipo: `{#FSTYPE}`
- *Chave*: Free[`{#FSNAME}`]

Como tipo de informação, use:

- *Numérico (sem sinal)* para métricas como ’free’, ’total’, ’used’
- *Numérico (float)* para métricas como ’pfree’, ’pused’ (percentual)

Na aba ”Pré-processamento” do protótipo de item, selecione JSONPath e use a seguinte expressão JSONPath como parâmetro:

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Quando a descoberta começar, um item para cada ponto de montagem será criado. Este item retornará o número de bytes livres para o ponto de montagem correspondente.

2 Descoberta de interfaces de rede

De forma semelhante à descoberta de sistemas de arquivos, também é possível descobrir interfaces de rede.

Chave do item

A chave do item a ser usada na regra de descoberta é

`net.if.discovery`

Macros suportadas

Você pode usar a macro {#IFNAME} na regra de descoberta (filtro) e nos protótipos de items, triggers e gráficos.

Exemplos de protótipos de item 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 forma semelhante à descoberta de sistemas de arquivos, também é possível descobrir 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, respectivamente, o número de ordem da CPU e o status. 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 ficar 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, há um bug raro relacionado a grupos de processadores no Windows.

4 Descoberta de OIDs SNMP

Visão geral

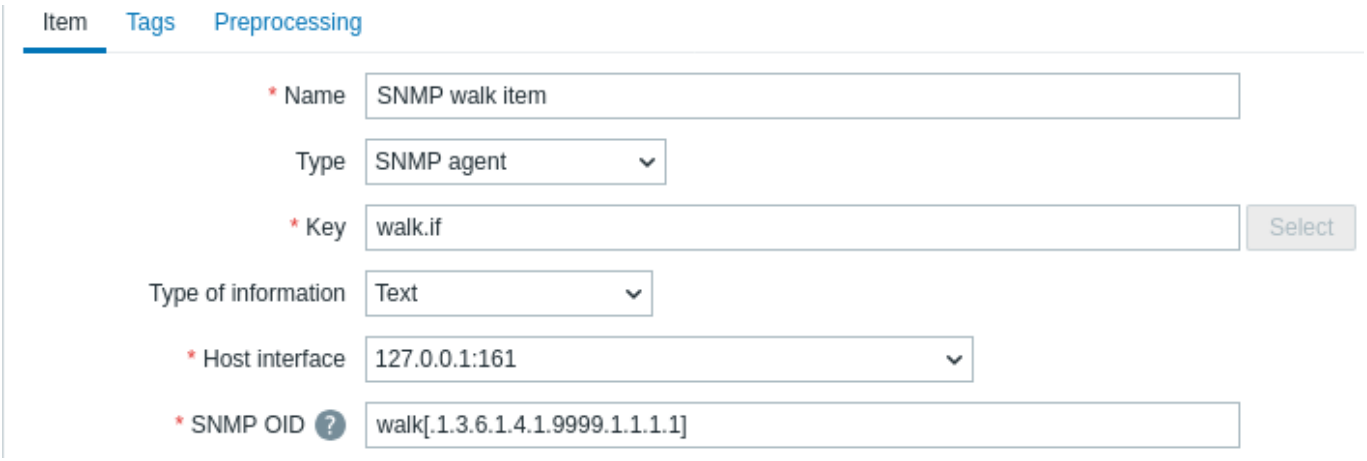
Nesta seção, realizaremos uma descoberta de baixo nível em um dispositivo SNMP.

Este método de descoberta de OIDs SNMP é suportado desde o Zabbix server/proxy 6.4.

Configuração de exemplo

1. Crie um item de agent SNMP com uma chave como:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



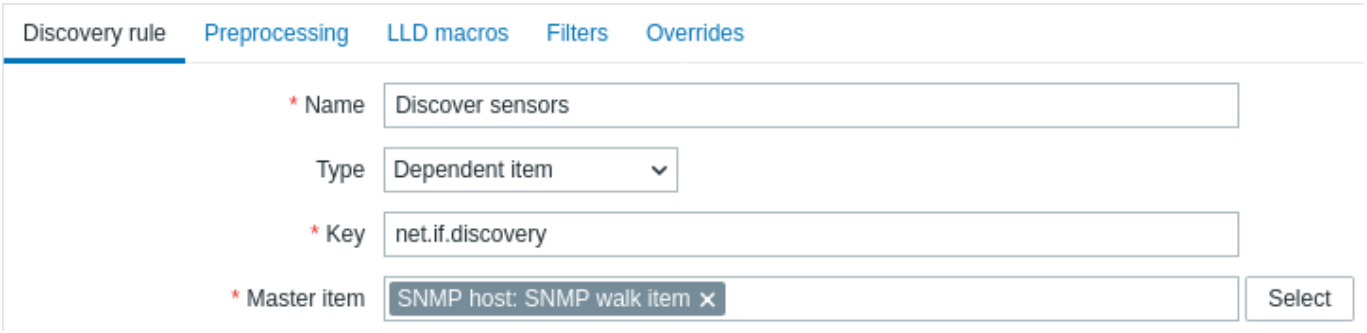
Este item executa uma única varredura de tabela SNMP e retorna todas as entradas da tabela em uma única solicitação, em um formato que corresponde à saída do utilitário `snmpwalk` com as opções de formatação `-Oe -Ot -On`.

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. Sob a 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 item dependente 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 polling, 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 do 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 ser semelhante a:

```
[
    {
        "{#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 este protótipo inclui o nome ”SNMP walk value” com o OID ”1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}” no campo *Parâmetro*. *Formato*: ”Inalterado”.

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 item dependente antigo 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}]
```

Esta 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, iremos realizar uma descoberta SNMP 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 as 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"

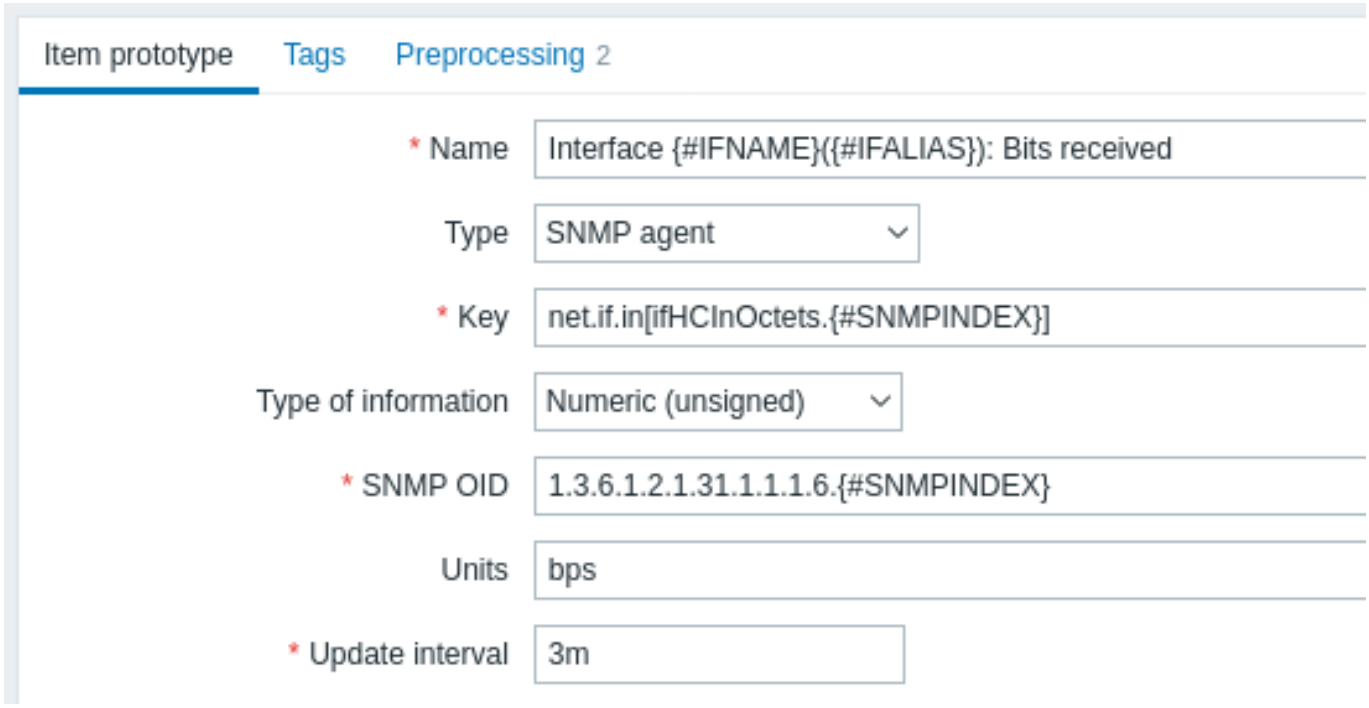
ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

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

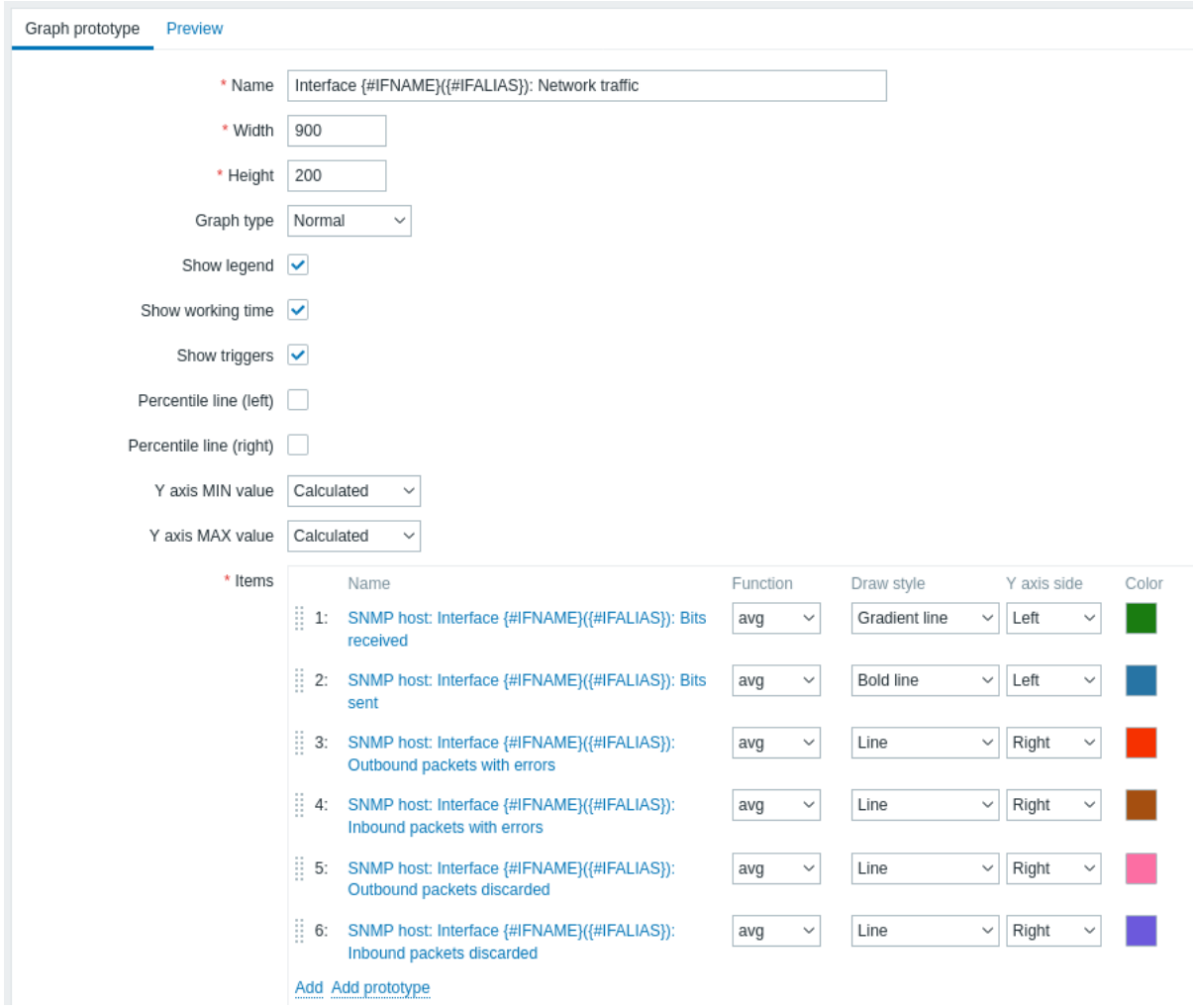
Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): 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

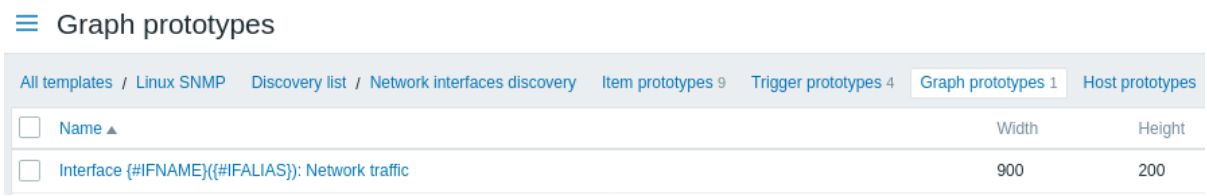
Protótipos de trigger

A captura de tela a seguir ilustra como podemos usar essas macros em protótipos de trigger:

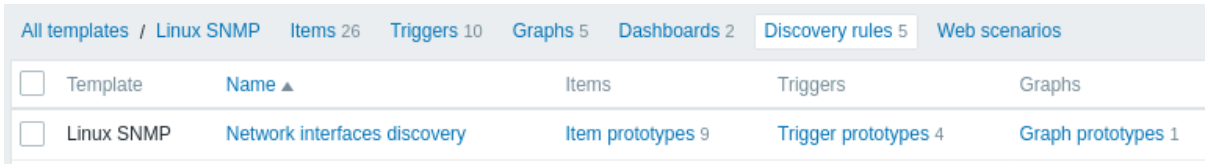
Protótipos de gráficos

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 de onde vieram.

6 Descoberta de objetos JMX

Visão geral

É possível descobrir todos os MBeans JMX ou atributos de MBean ou especificar um padrão para a descoberta desses objetos.

É obrigatório entender a diferença entre um MBean e atributos de MBean para a configuração da regra de descoberta. Um MBean é um objeto que pode representar um dispositivo, uma aplicação ou qualquer recurso que precise ser gerenciado.

Por exemplo, existe um MBean que representa um servidor web. Seus atributos são contagem de conexões, contagem de threads, tempo limite de requisição, cache de arquivos http, uso de memória, etc. Expressando esse pensamento em uma linguagem compreensível para humanos, 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 determinado período de tempo, número de grãos de café necessários por xícara, tempo de reposição de grãos de café e 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* (recupera atributos do JMX MBean, padrão) ou *beans* (recupera 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 do 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* (recupera atributos do JMX MBean, padrão) ou *beans* (recupera 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 suportadas

As seguintes macros são suportadas para uso na regra de descoberta de filtro e protótipos de items, triggers e gráficos:

Macro	Descrição
Descoberta de atributos MBean	
{#JMXVALUE}	Valor do atributo.
{#JMXTYPE}	Tipo do atributo.
{#JMXOBJ}	Nome do objeto.
{#JMXDESC}	Nome do objeto incluindo o nome do atributo.
{#JMXATTR}	Nome do atributo.
Descoberta de MBeans	
{#JMXDOMAIN}	Domínio do MBean. (*Nome reservado pelo Zabbix*)
{#JMXOBJ}	Nome do objeto. (*Nome reservado pelo Zabbix*)
{#JMX<key property>}	Propriedades do MBean (como {#JMXTYPE}, {#JMXNAME}) (veja 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 possui 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 mais dois exemplos práticos de criação de uma regra LLD com o uso de MBean. Para entender melhor a diferença entre uma regra LLD que coleta MBeans e uma regra LLD que coleta atributos de MBean, por favor, observe a tabela a seguir:

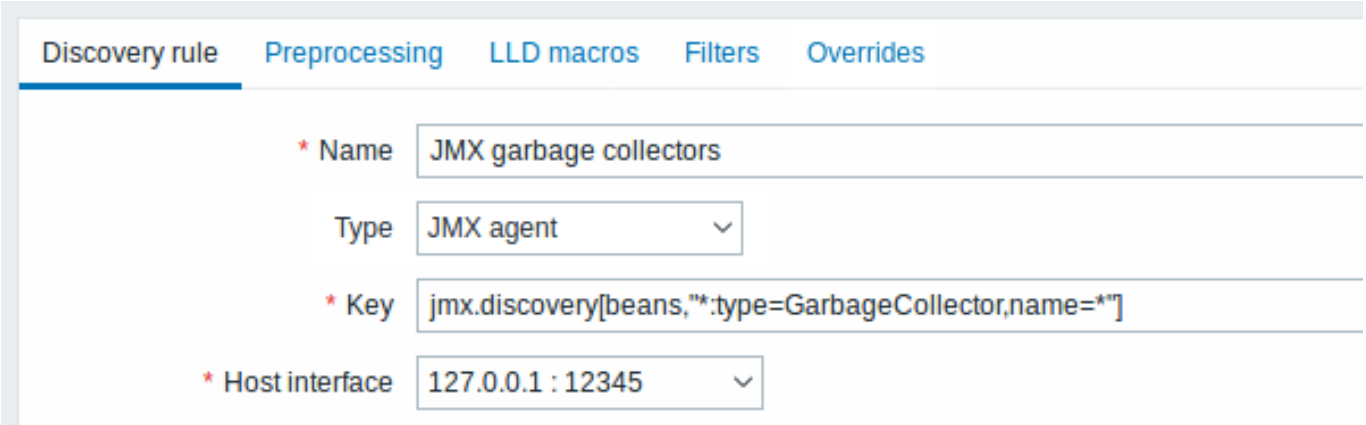
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

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*.

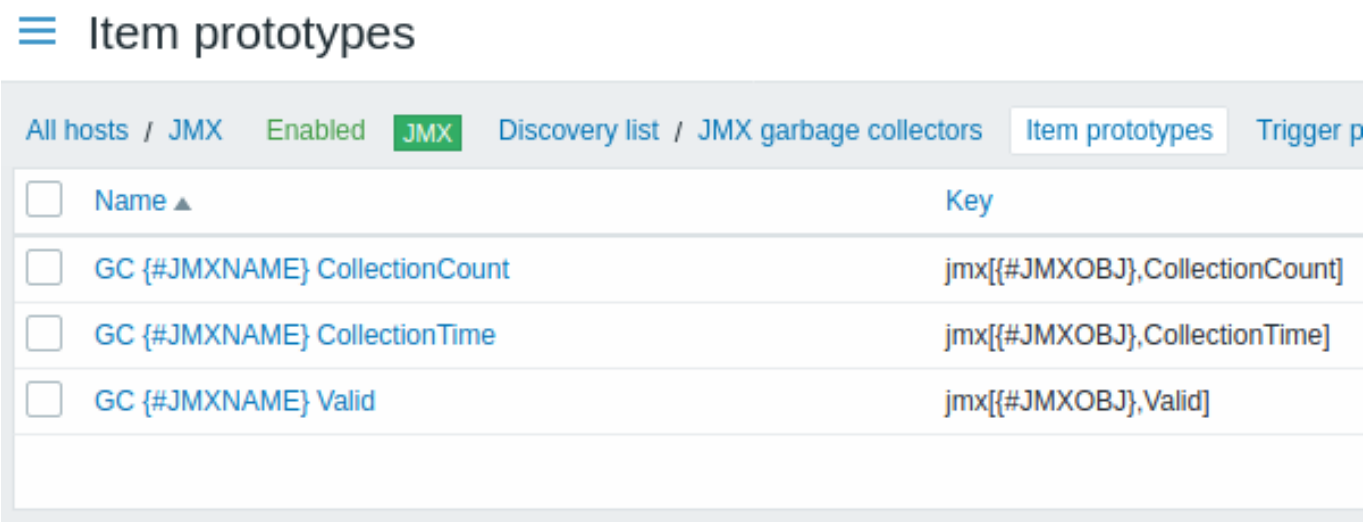
A configuração da regra de descoberta coletando MBeans (sem os atributos) fica assim:



A chave usada aqui:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

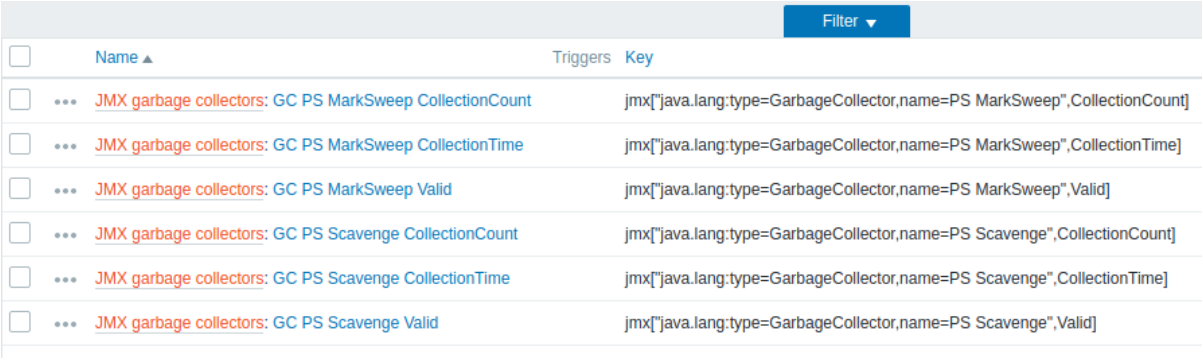
Todos os coletores de lixo sem atributos serão descobertos. Como os coletores de lixo têm o mesmo conjunto de atributos, podemos usar os atributos desejados nos protótipos de item da seguinte forma:



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):

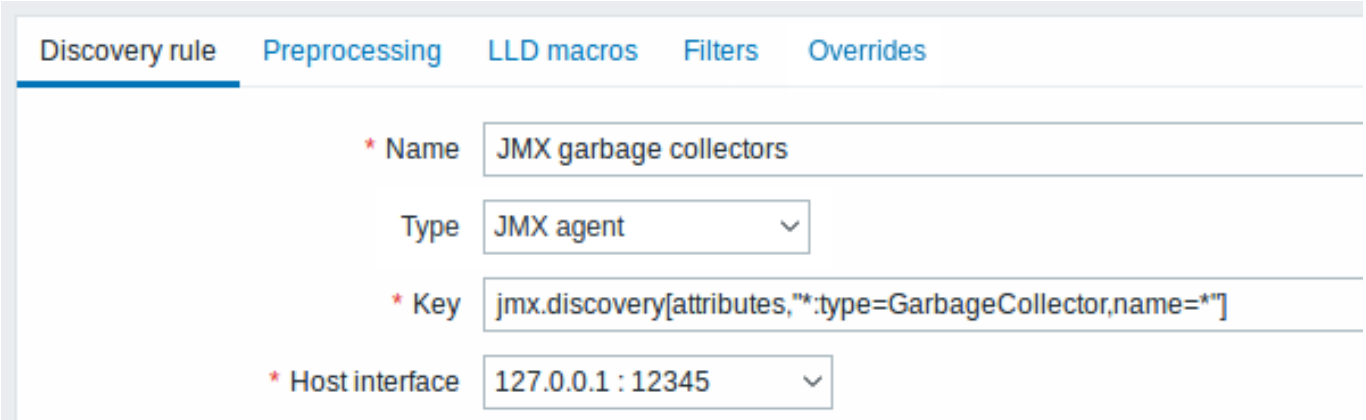


Exemplo 2: Descobrindo atributos MBean

Esta regra retornará 9 objetos com os seguintes campos: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1, MBean1Attribute2, MBean2Attribute2, MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Para mais informações sobre objetos, consulte a tabela de macros suportadas, seção *Descoberta de atributos MBean*.

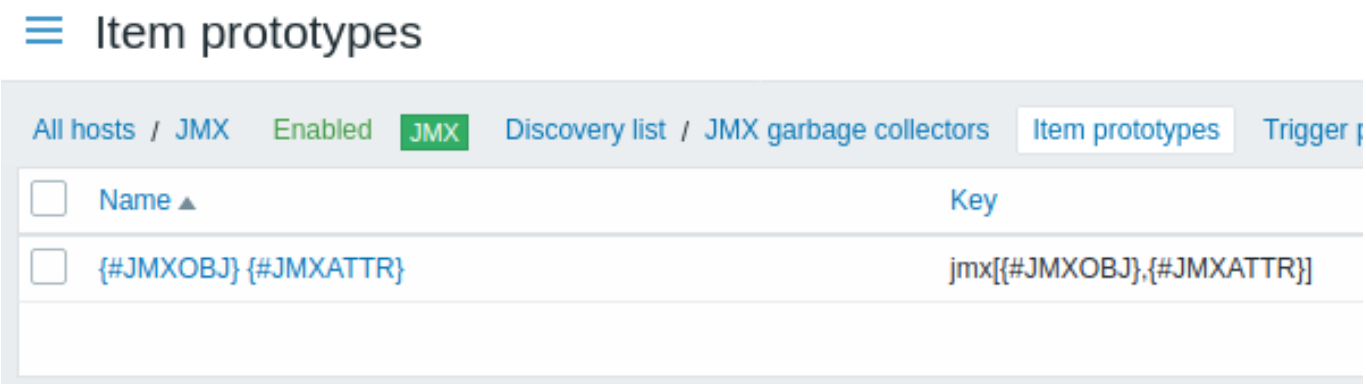
A configuração da regra de descoberta coletando atributos MBean é a seguinte:



A chave usada aqui:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

Todos os coletores de lixo com um único atributo de item serão descobertos.



Neste caso específico, um item será criado a partir do protótipo para cada atributo MBean. A principal desvantagem desta configuração é que a criação de triggers 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 aba 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,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

Descobrindo atributos MBean

Item de descoberta: `jmx.get[attributes,"com.example:type=*,*"]`

Resposta:

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 Descoberta de sensores IPMI

Visão geral

É possível descobrir sensores IPMI automaticamente.

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

Configuração

Item mestre

Crie um item IPMI usando a seguinte chave:

`ipmi.get`

Defina o tipo de informação como ”Texto” para possíveis grandes dados JSON.

Regra de LLD dependente

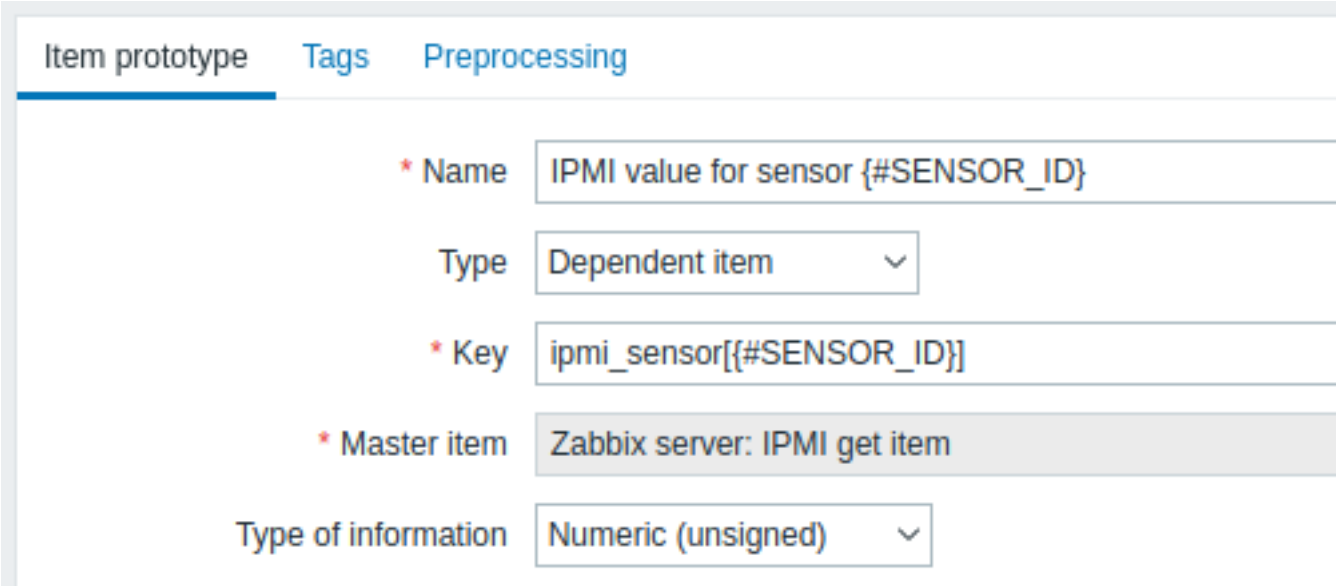
Crie uma regra de descoberta de baixo nível do tipo ”Item dependente”:

Como item mestre, selecione o item `ipmi.get` que criamos.

Na aba ”Macros de LLD”, defina uma macro personalizada com o JSONPath correspondente:

Protótipo de item dependente

Crie um protótipo de item com o tipo ”Item dependente” nesta regra de LLD. Como item mestre para este protótipo, selecione o item `ipmi.get` que criamos.



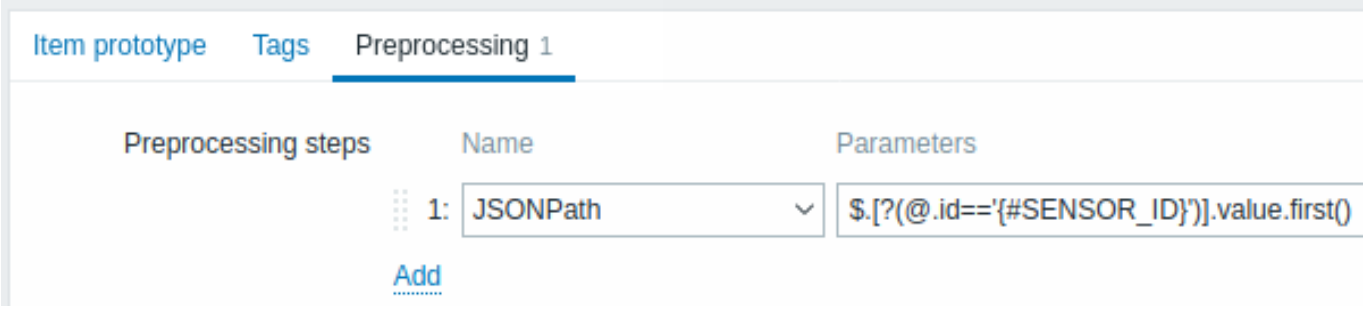
Observe o uso da macro {#SENSOR_ID} no nome e na chave do protótipo de item:

- *Nome*: Valor IPMI para o sensor {#SENSOR_ID}
- *Chave*: ipmi_sensor[{#SENSOR_ID}]

Como tipo de informação, *Numérico (sem sinal)*.

Na aba ”Pré-processamento” do protótipo de item, selecione JSONPath e use a seguinte expressão JSONPath como parâmetro:

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



Quando a descoberta começar, um item para cada sensor IPMI será criado. Este item retornará o valor inteiro do sensor correspondente.

8 Descoberta de serviços systemd

Visão geral

É possível descobrir unidades systemd (serviços, por padrão) com o Zabbix.

Chave do item

O item a ser usado na regra de descoberta é

`systemd.unit.discovery`

Attention:
Esta chave de item é suportada apenas no Zabbix agent 2.

Este item retorna um JSON com informações sobre as unidades do systemd, por exemplo:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

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 suportadas

As seguintes macros são suportadas para uso na regra de descoberta (filtro) e protótipos de items, triggers 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 da unidade foi carregado com sucesso)
{#UNIT.ACTIVESTATE}	Estado ativo (ou seja, se a unidade está atualmente iniciada ou não)
{#UNIT.SUBSTATE}	Subestado (uma versão mais detalhada do estado ativo que é específica para o tipo de unidade, o 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 numérico do job se houver um job na fila para a unidade; 0 caso contrário.
{#UNIT.JOBTYPE}	Tipo do job.
{#UNIT.JOBPATH}	Caminho do objeto do job.
{#UNIT.UNITFILESTATE}	O estado de instalação do arquivo da unidade.

Macro	Descrição
{#UNIT.SERVICETYPE}	Tipo da unidade de serviço (por exemplo, `simple`, `forking`, `oneshot`, `idle`, etc.). Esta macro é retornada apenas se a unidade for um serviço.

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 forma semelhante à descoberta de sistemas de arquivos, também é possível descobrir 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.
{#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 (inicialização por trigger)*, *Automático atrasado (inicialização por trigger)* e *Manual (inicialização 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 múltiplas 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 diversas informações sobre componentes do Windows, serviços, estado e softwares instalados.

Ela pode ser usada para descoberta de discos físicos e coleta de dados de desempenho, descoberta de interfaces de rede, descoberta de convidados Hyper-V, monitoramento de serviços do Windows e muitas outras coisas no sistema operacional Windows.

Esse tipo de descoberta de baixo nível é feito usando consultas WQL, cujos resultados são automaticamente transformados 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 como:

```
[
    {
        "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 JSON retornado, essas macros podem ser definidas pelo usuário como uma etapa adicional, usando a funcionalidade de macro LLD personalizada com JSONPath apontando para os valores descobertos no JSON retornado.

As macros podem então ser usadas para criar protótipos de item, trigger, 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 e única>,<dsn>,<string de conexão>] - este item transforma o resultado da consulta SQL em um array JSON, convertendo 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 e única>,<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 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 macro 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 items internos "zabbix[proxy,<name>,lastaccess]" para monitorar quais proxies estão ativos.

Vamos começar com a configuração da regra de descoberta:

lld_rule_odbc.png

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 se tornam nomes de macros e as linhas selecionadas se tornam os valores dessas macros.

Note:
Se não for óbvio como um nome de coluna seria transformado em um nome de macro, recomenda-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 se torna não suportada, com a mensagem de erro detalhando o número da coluna com problema. Se desejar ajuda adicional, os nomes das colunas obtidas são fornecidos com DebugLevel=4 no arquivo de log do Zabbix server:

```
$ 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} nos protótipos de item:

item_prototype_odbc.png

Uma vez que a descoberta seja realizada, um item será criado para cada proxy:

discovered_items_odbc1.png

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 de baixo nível.

Veja verificações do Prometheus para detalhes de como a consulta de dados do Prometheus é implementada no Zabbix.

Configuração

A regra de descoberta de baixo nível deve ser criada como um item dependente do item mestre HTTP que coleta os dados do Prometheus.

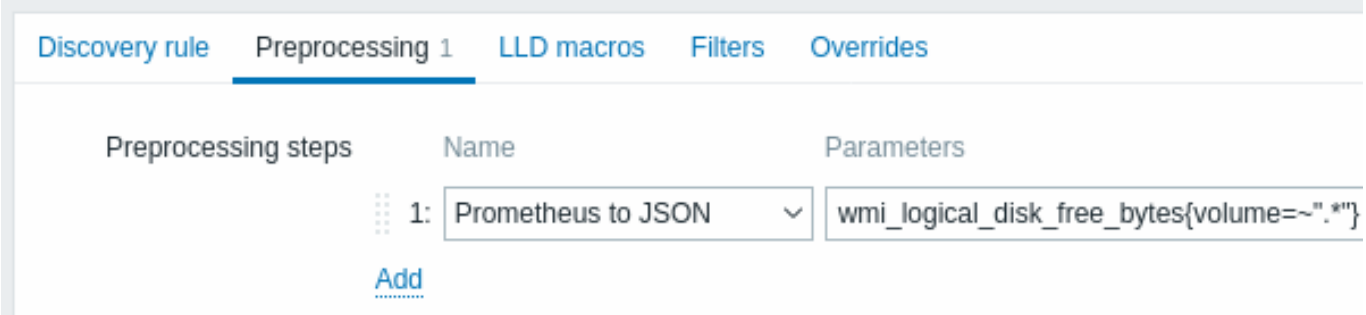
Prometheus para JSON

Na regra de descoberta, vá para a aba Pré-processamento e selecione a opção de pré-processamento *Prometheus para JSON*. Os dados em formato JSON são necessários para a descoberta e a opção de pré-processamento *Prometheus para JSON* retornará exatamente isso, com os seguintes atributos:

- nome da métrica
- valor da métrica
- help (se presente)

- type (se presente)
- labels (se presente)
- linha bruta

Por exemplo, consultando `wmi_logical_disk_free_bytes`:



dessas linhas do Prometheus:

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

retornará:

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

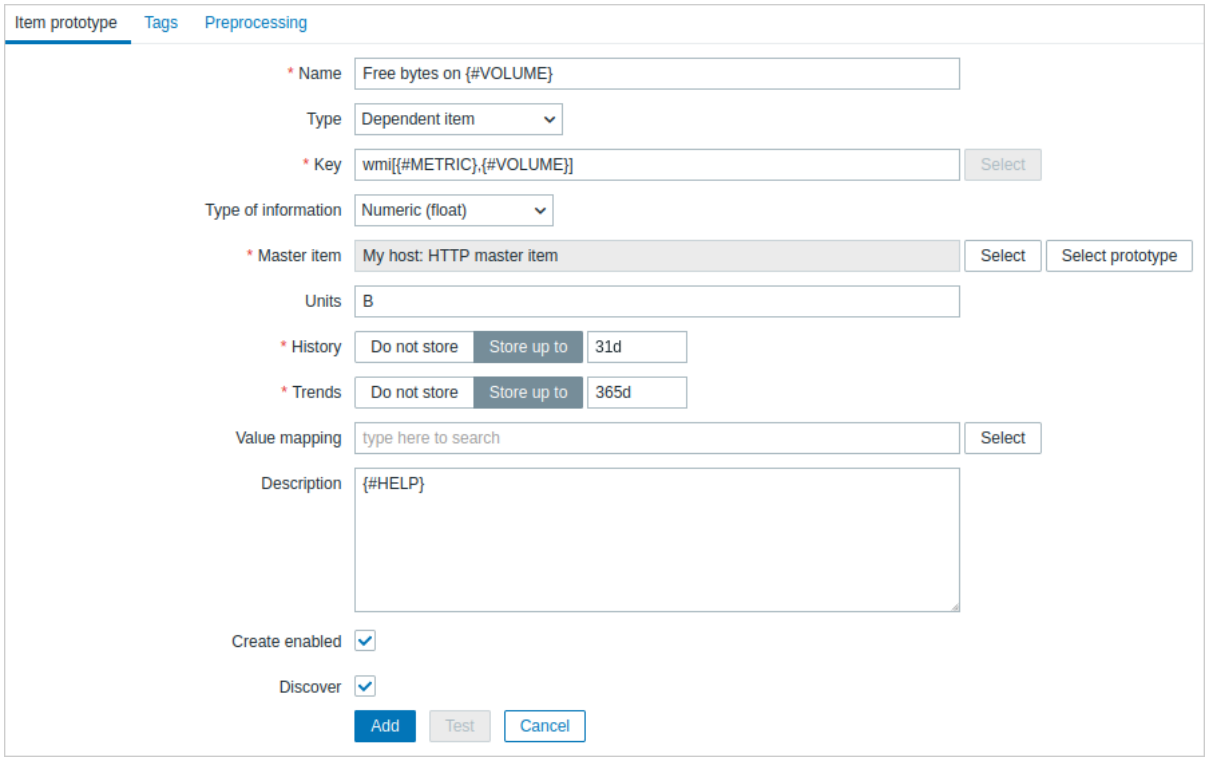
Mapeando macros LLD

Em seguida, você deve ir para a aba de macros LLD e fazer os seguintes mapeamentos:

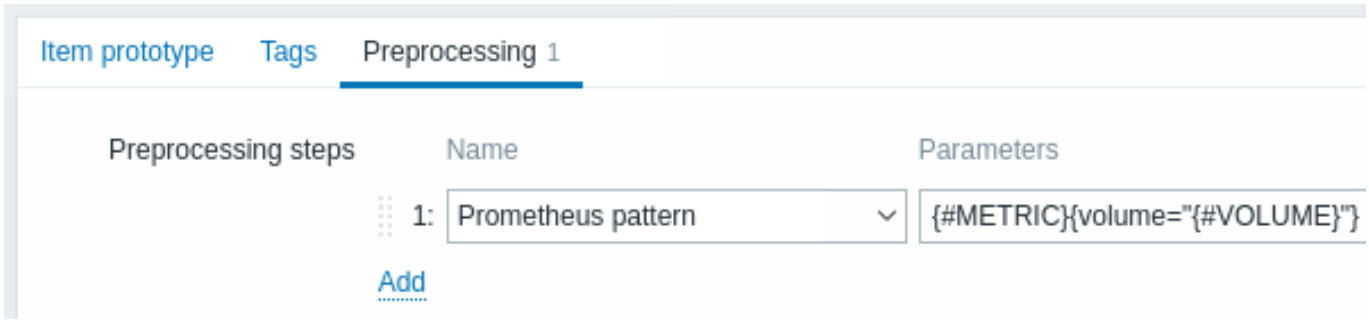
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

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 suportadas

Esta chave de descoberta retorna duas macros - {#DEVNAME} e {#DEVTYPE} identificando respectivamente o nome e o tipo do dispositivo de bloco, por exemplo:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
```

```
    },
    {
        "{#DEVNAME}":"dm-0",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

A descoberta de dispositivos de bloco permite usar os items `vfs.dev.read[]` e `vfs.dev.write[]` para criar protótipos de item usando a macro {#DEVNAME}, por exemplo:

- ”vfs.dev.read[{#DEVNAME},sps]”
- ”vfs.dev.write[{#DEVNAME},sps]”

{#DEVTYPE} é destinado à filtragem de dispositivos.

15 Descoberta de interfaces de host no Zabbix

Visão geral

É possível descobrir todas as 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

`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 (Zabbix agent, 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 no 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 - é a interface padrão

Macro	Descrição
{#IF.SNMP.BULK}	Status de processamento SNMP bulk para a interface: 0 - desabilitado 1 - habilitado Esta macro é retornada apenas se o tipo de interface for “SNMP”.

8 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 a descoberta de sistema de arquivos), é possível retornar mais.

Exemplo

O formato de string JSON necessário é melhor ilustrado com um exemplo. Suponha que você esteja executando um agent Zabbix 1.8 antigo (que não suporta a chave vfs.fs.discovery), mas ainda precisa 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 tanto o nome quanto 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 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.

```
[
    { "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, você pode especificar "{#FSTYPE}" como uma macro e "rootfs|ext3" como uma expressão regular.

Note:
Você não precisa usar nomes de macro como 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 16MB. Para mais detalhes, consulte limites de dados para valores de retorno do LLD.

16 Monitoramento distribuído

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, relatar 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 BD	**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 BD 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 detalhes.

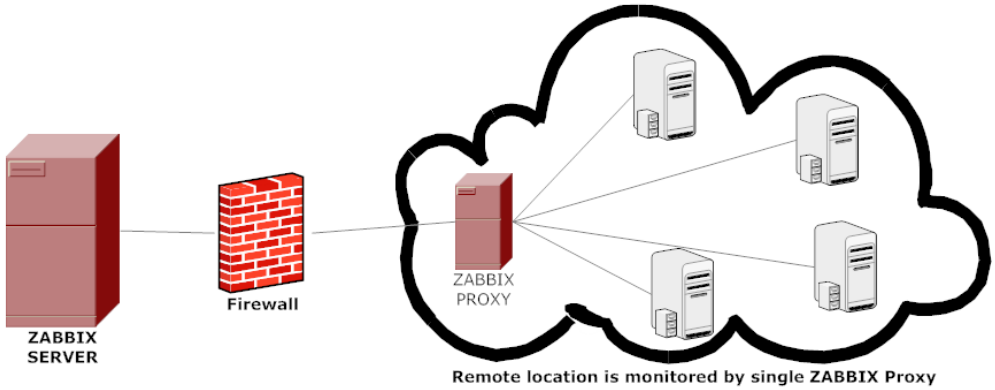
1 Proxies

Visão geral Um proxy Zabbix pode coletar dados de desempenho e disponibilidade em nome do servidor Zabbix. Dessa forma, um proxy pode assumir parte da carga de coleta de dados e aliviar o servidor Zabbix.

Além disso, usar um proxy é a maneira mais fácil de implementar monitoramento centralizado e distribuído, quando todos os agents e proxies reportam para um servidor Zabbix 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 servidor Zabbix ao monitorar milhares de dispositivos
- Simplificar a manutenção do monitoramento distribuído



O proxy requer apenas uma conexão TCP com o servidor Zabbix. 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 servidor Zabbix irá quebrar a configuração.

Todos os dados coletados pelo proxy são armazenados localmente antes de serem transmitidos para o servidor. Dessa forma, nenhum dado é perdido devido a problemas temporários de comunicação com o servidor. Os parâmetros *ProxyLocalBuffer* e *ProxyOfflineBuffer* no arquivo de configuração do proxy controlam por quanto tempo os dados são mantidos localmente.

Attention:
Pode acontecer que um proxy, que recebe as últimas alterações de configuração diretamente do banco de dados do servidor Zabbix, tenha uma configuração mais atualizada do que o servidor Zabbix, 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 servidor Zabbix, 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 do que é a funcionalidade do proxy, consulte a tabela a seguir:

Função		Suportado pelo proxy
Items		
	Verificações do agent Zabbix	**Sim**
	Verificações do agent Zabbix (ativo)	**Sim** [1]
	Verificações simples	**Sim**
	Items trapper	**Sim**
	Verificações SNMP	**Sim**
	Traps SNMP	**Sim**
	Verificações IPMI	**Sim**
	Verificações JMX	**Sim**
	Monitoramento de arquivos de log	**Sim**
	Verificações internas	**Sim**
	Verificações SSH	**Sim**
	Verificações Telnet	**Sim**
	Verificações externas	**Sim**
	Items dependentes	**Sim**
	Items de script	**Sim**
	Items de navegador	**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 servidor) 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 servidor Zabbix para processamento adicional.

Proteção contra sobrecarga

Se o Zabbix server ficar fora do ar por algum tempo e os proxies coletarem muitos dados, quando 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 restringido, interrompendo os processos de coleta de dados do server. O caso mais comum de sobrecarga do cache de histórico ocorre após uma parada do server, quando os proxies estão enviando os dados coletados. Para evitar isso, foi adicionada uma limitação nos proxies (atualmente não pode ser desabilitada).

Quando o uso do cache de histórico atinge 80%, o Zabbix server entra no modo de limitação. No modo de limitação, o server aceita dados do proxy apenas quando o uso do cache de histórico está abaixo de 60%, alternando os proxies aceitos. Assim que o uso do cache de histórico cai abaixo de 20%, o server retorna ao modo normal.

Além disso, no modo normal, o Zabbix server limita proxies individuais que enviam pacotes muito grandes (10.000+ registros) se o uso do cache de histórico exceder 60%. Essa decisão é aplicada no instante em que o server avalia um upload do proxy e, portanto, pode não ser sempre refletida imediatamente nos gráficos de uso do cache de histórico (o item interno `zabbix[wcache,history,pused]` e seu intervalo de atualização podem não capturar picos curtos).

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).
Atinge 60%	Normal, mas preparado para ser limitado	Pode recusar uploads muito grandes do proxy (mais de 10k registros) ao decidir se aceita os dados; continua aceitando outros dados do proxy.
Cai para 20%	Normal	Descarta a lista de limitação e começa a aceitar dados do proxy normalmente.

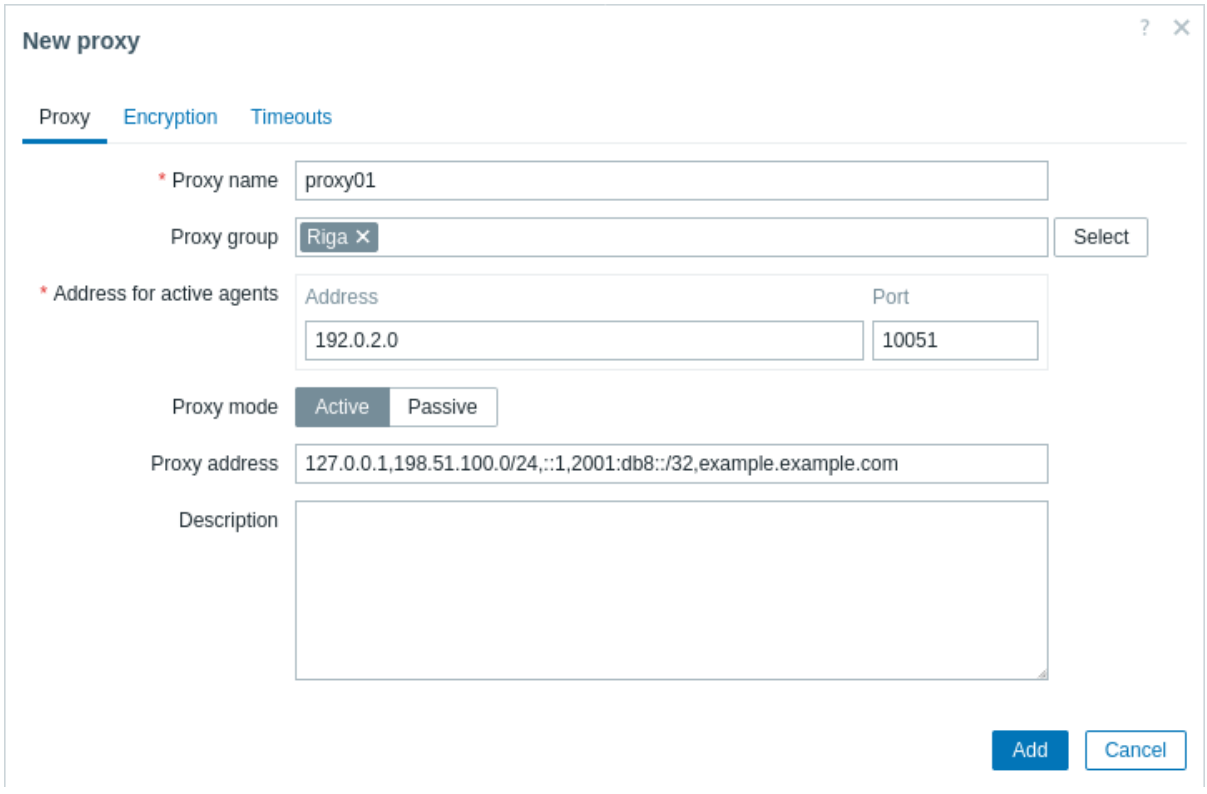
Você pode usar o item interno zabbix[wcache,history,pused] para correlacionar esse comportamento do Zabbix server com uma métrica.

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*



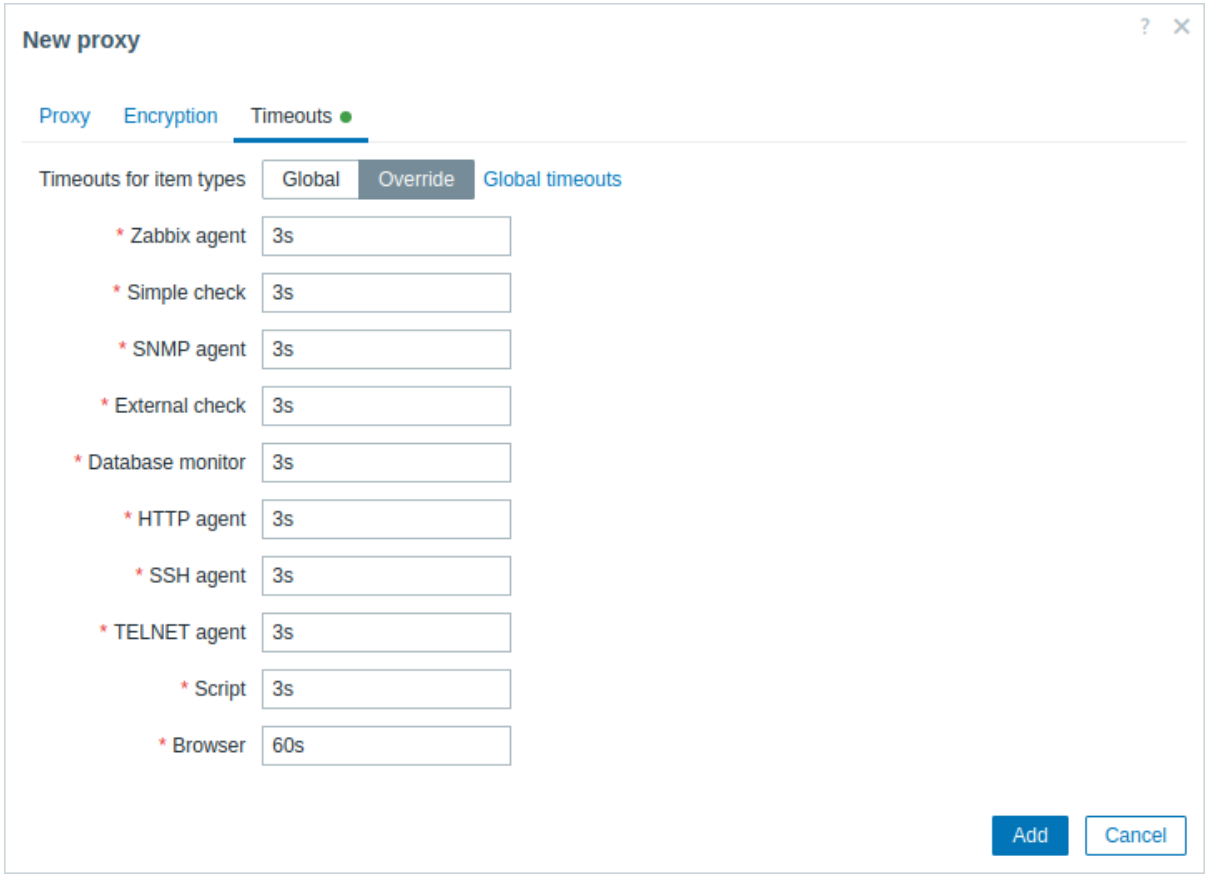
Parâmetro	Descrição
Nome do proxy	Insira o nome do proxy. Deve ser o mesmo nome do parâmetro *Hostname* no arquivo de configuração do proxy.

Parâmetro	Descrição
Grupo de proxy	Selecione um grupo de proxy para balanceamento de carga/alta disponibilidade do proxy.
Endereço para agents ativos	Insira 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.
Modo do proxy	Selecione o modo do proxy. **Ativo** - o proxy irá se conectar ao servidor Zabbix e solicitar dados de configuração **Passivo** - o servidor Zabbix se conecta ao proxy *Nota* que sem comunicações criptografadas (sensível) os dados de configuração do proxy podem ficar disponíveis para partes que tenham acesso à porta trapper do servidor Zabbix 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 os endereços do proxy não forem limitados no campo *Endereço do proxy*.
Endereço do proxy	Se especificado, as solicitações de 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	Insira 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	Insira a descrição do proxy.

A aba **Criptografia** permite exigir conexões criptografadas com o proxy.

Parâmetro	Descrição
Conexões com o proxy	Como o servidor 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 (tanto checagens passivas quanto ativas) - Checagem simples (exceto itens `icmpping*`, `vmware.*`) - SNMP agent (apenas para itens SNMP `walk[OID]` e `get[OID]`) - Checagem 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 servidor, 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 servidor 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 de host

Você pode especificar que um host individual deve ser monitorado por um proxy ou grupo de proxies no formulário de configuração de 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

Visão geral

Esta página fornece detalhes sobre a atualização da configuração de monitoramento para o proxy, ou seja, como as alterações feitas na configuração de monitoramento no server são sincronizadas com o 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 na configuração do proxy são rastreadas usando números de revisão. Somente entidades com revisões maiores do 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 que:

- 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** determina com que frequência a configuração do proxy é sincronizada com o server (10 segundos por padrão).

Observe que ProxyConfigFrequency é:

- parâmetro do server para proxies passivos
- parâmetro do proxy para proxies ativos

Em proxies ativos, ProxyConfigFrequency é um novo parâmetro desde o Zabbix 6.4 e deve ser usado em vez do agora obsoleto ConfigFrequency.

Attention:
Se ambos ProxyConfigFrequency e ConfigFrequency forem usados, o proxy registrará um erro e encerrará.

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.

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 por 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 grupos 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 reatribuídos aos 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 grupos 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

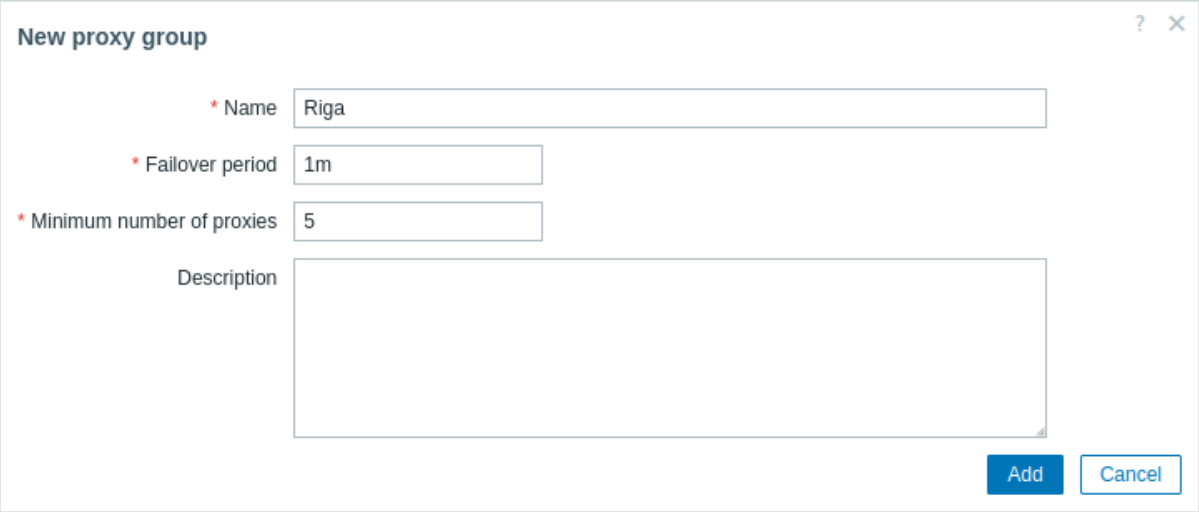
Hosts no proxy	Média do grupo	Reatribuição de hosts
25	50	Sim
15	5	Sim
10	5	Não

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 qualquer 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 checagens passivas, liste todos os proxies do grupo de proxies no parâmetro Server.

- Para checagens ativas, recomenda-se listar todos os proxies do grupo de proxies ou o server Zabbix no parâmetro Server-Active. 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, as checagens ativas serão redirecionadas 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, as checagens ativas podem travar ou falhar durante o redirecionamento ou failover. Por exemplo:

- Durante checagens ativas, 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 estáveis de alta disponibilidade 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 monitorados por um grupo de proxies, e não por proxies individuais (você pode usar a atualização em massa 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 mudança atualizando a lista de hosts em *Administração* > *Proxies*.

Notas importantes

- Traps SNMP não são suportadas por proxies em 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 no grupo de proxies.
- Verificações de banco de dados requerem permissões estendidas no objeto/servidor do 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
- Entre o Zabbix frontend e o Zabbix server

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 recebidas 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 pelos componentes do Zabbix durante a inicialização.
- As chaves pré-compartilhadas são inseridas no frontend do Zabbix e armazenadas no banco de dados do Zabbix em texto simples.
- A criptografia interna não protege as comunicações entre o servidor web que executa o frontend do Zabbix e o navegador web do usuário.
- Atualmente, cada conexão criptografada é aberta com um handshake TLS completo, sem cache de sessão e tickets implementados.
- Adicionar criptografia aumenta o tempo para verificações de item e ações, dependendo da latência da rede:
  - Por exemplo, se o atraso do pacote for de 100ms, abrir uma conexão TCP e enviar uma solicitação não criptografada leva cerca de 200ms. Com criptografia, cerca de 1000 ms são adicionados para estabelecer a conexão TLS.
  - Os timeouts podem precisar ser aumentados, caso contrário, alguns itens e ações que executam scripts remotos em agents podem funcionar com conexões não criptografadas, mas falhar com timeout quando criptografadas.
- A criptografia não é suportada pela descoberta de rede. As verificações do agent Zabbix realizadas pela descoberta de rede serão não criptografadas e, se o agent Zabbix estiver configurado para rejeitar conexões não criptografadas, essas verificações não terão 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 server e 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 (em modo ativo, especifica apenas conexões para o server) e do agent Zabbix (para checagens 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 (em modo passivo, especifica apenas conexões do server) e do agent Zabbix (para checagens 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 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 (checagens ativas) e do zabbix_sender (itens trapper).

Muito provavelmente você irá configurar 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 são sem criptografia por padrão. A criptografia deve ser configurada para cada host e proxy individualmente.

zabbix_get e zabbix_sender com criptografia Consulte as páginas de manual do zabbix_get e zabbix_sender 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 ainda funcionam).

Para bibliotecas de criptografia compiladas com as 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
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 ciphersuites podem ser substituídos por ciphersuites configuradas pelo usuário.

Attention:
Ciphersuites configuradas pelo usuário são um recurso destinado a usuários avançados que compreendem ciphersuites TLS, sua segurança e as consequências de erros, e que estejam confortáveis com a solução de problemas de TLS.

Os critérios internos de seleção de ciphersuites 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)

Escopo de substituição	Parâmetro	Valor	Descrição
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)
Lista combinada de ciphersuites 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 ciphersuites baseadas em certificado **e** PSK for usada. Existem dois casos em que isso ocorre: server (proxy) com um certificado configurado (ciphersuites 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 ciphersuites configuradas pelo usuário

Veja abaixo os seguintes exemplos de ciphersuites configuradas pelo usuário:

- Testando strings de cifra e permitindo apenas ciphersuites 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 o 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 reforçar 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 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 AES128 como padrão interno para dados. Vamos supor que você esteja usando certificados e queira 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
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
Embora apenas os ciphersuites relacionados a certificados sejam usados, os parâmetros `TLSCipherPSK*` também são definidos para evitar seus valores padrão, que incluem cifras menos seguras para maior interoperabilidade. Os ciphersuites PSK não podem ser completamente desabilitados no server/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 para 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 para um arquivo contendo Listas de Revogação de Certificados (CRL).
TLSCertFile	sim	Caminho completo para 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 para 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 de certificado da CA.

Para verificar os certificados dos pares, o Zabbix server deve ter acesso ao arquivo que contém os certificados das CAs raiz autoassinadas 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
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVm0
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. 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
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Proxy Zabbix

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/cadeia de certificados do proxy Zabbix e a chave privada conforme descrito na seção Servidor Zabbix. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do proxy Zabbix conforme necessário.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do proxy Zabbix:

- 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
```

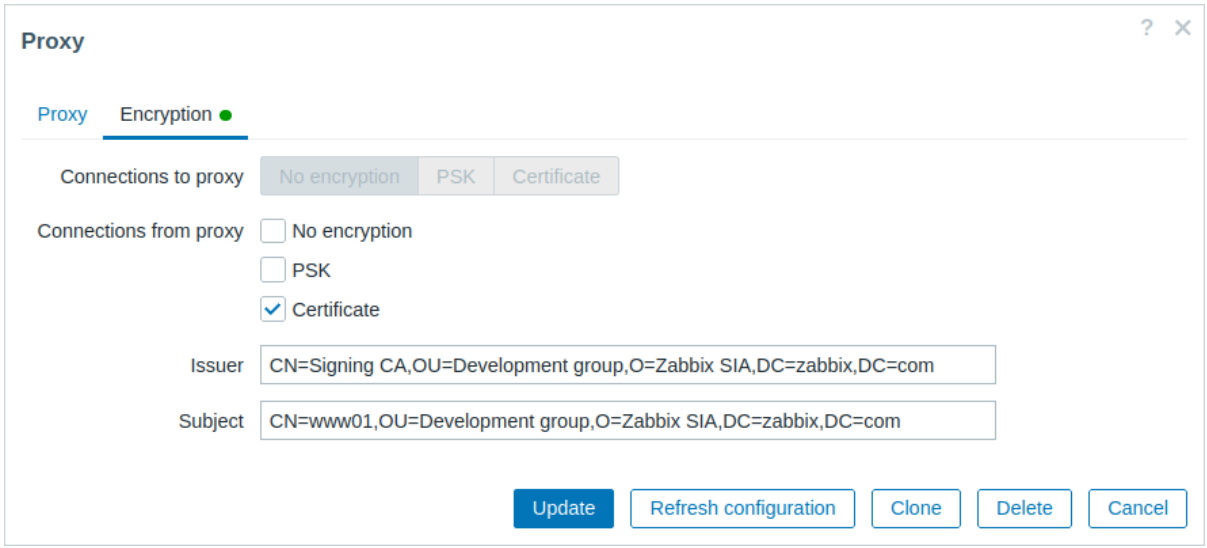
```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configure a criptografia para este proxy no frontend Zabbix:

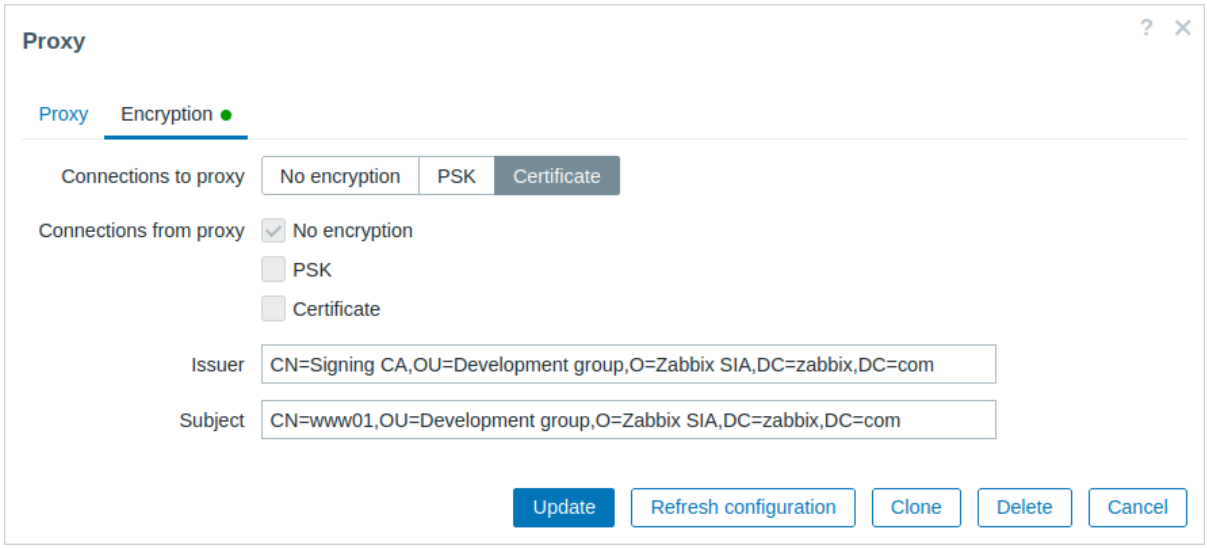
- Vá para: *Administração* → *Proxies*.
- Selecione o proxy e clique na guia *Criptografia*.

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:



Agent Zabbix

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/cadeia de certificados do agent Zabbix e a chave privada conforme descrito na seção Servidor Zabbix. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do agent Zabbix conforme necessário.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do agent Zabbix:

- 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 emissor e assunto do certificado permitido.

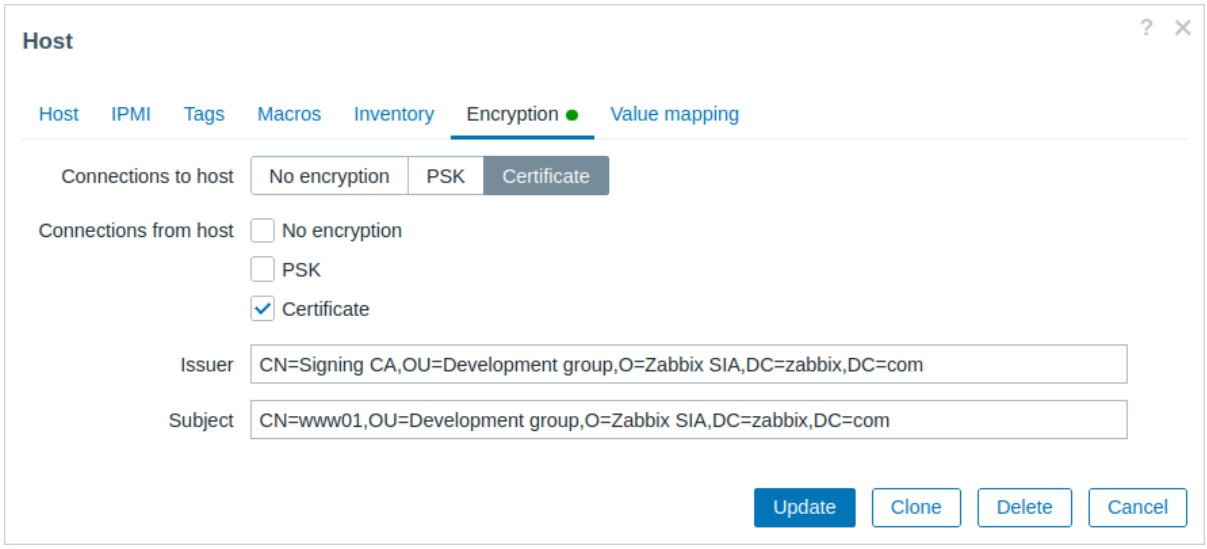
Os parâmetros TLS no arquivo de configuração final do agent podem ser semelhantes ao exemplo abaixo. Observe que o exemplo pressupõe que o host é monitorado por um proxy, portanto, ele é especificado como o Subject do certificado:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file
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 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 emissor e 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 conforme necessário.

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
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
```

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 de peer 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 qualquer outra (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 personificação 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 personificação, pode 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
```

Em contraste, o OpenSSL por padrão exibe 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 campo também varia com base nas opções usadas 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 criptográficos 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 criptográficos. 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 de string PSK secreto.

A string de identidade PSK é uma string UTF-8 não vazia. Por exemplo, "PSK ID 001 Zabbix agentd". É um nome exclusivo pelo qual esta PSK específica é referenciada pelos componentes do Zabbix. Não coloque informações confidenciais na string de identidade PSK - ela é transmitida pela rede sem criptografia.

O valor PSK é uma string difícil de adivinhar composta por dígitos hexadecimais, por exemplo, "e560cb0d918d26d31b4f642181f5f570ad89a39

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 menor:

Componente	Tamanho máximo da identidade do PSK	Tamanho mínimo do valor do PSK	Tamanho máximo do valor do PSK
Zabbix	128 caracteres UTF-8	128 bits (PSK de 16 bytes, inserido como 32 dígitos hexadecimais)	2048 bits (PSK de 256 bytes, inserido como 512 dígitos hexadecimais)
GnuTLS	128 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserido como 512 dígitos hexadecimais)
OpenSSL 1.0.x, 1.1.0	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserido como 512 dígitos hexadecimais)
OpenSSL 1.1.1	127 bytes (pode incluir caracteres UTF-8)	-	512 bits (PSK de 64 bytes, inserido como 128 dígitos hexadecimais)
OpenSSL 1.1.1a e posteriores	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserido 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 menores, 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 Zabbix server se conecte ao agent usando PSK, o server 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 de PSK deve ser pareada com apenas um valor. É responsabilidade do usuário garantir que não existam dois PSKs com a mesma string de identidade, mas valores diferentes. Não cumprir isso pode levar a erros imprevisíveis ou interrupções na comunicação entre os componentes do Zabbix que utilizam PSKs com essa string de identidade.

Gerando PSK

Por exemplo, um PSK de 256 bits (32 bytes) pode ser gerado usando os seguintes comandos:

- com *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- com *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Observe que o "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, portanto, 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 conectar ao server (checagens ativas) e aceitar do server e do `zabbix_get` apenas conexões usando PSK. A identidade do PSK será "PSK 001".

Reinicie o agent. Agora você pode testar a conexão usando o `zabbix_get`, por exemplo:

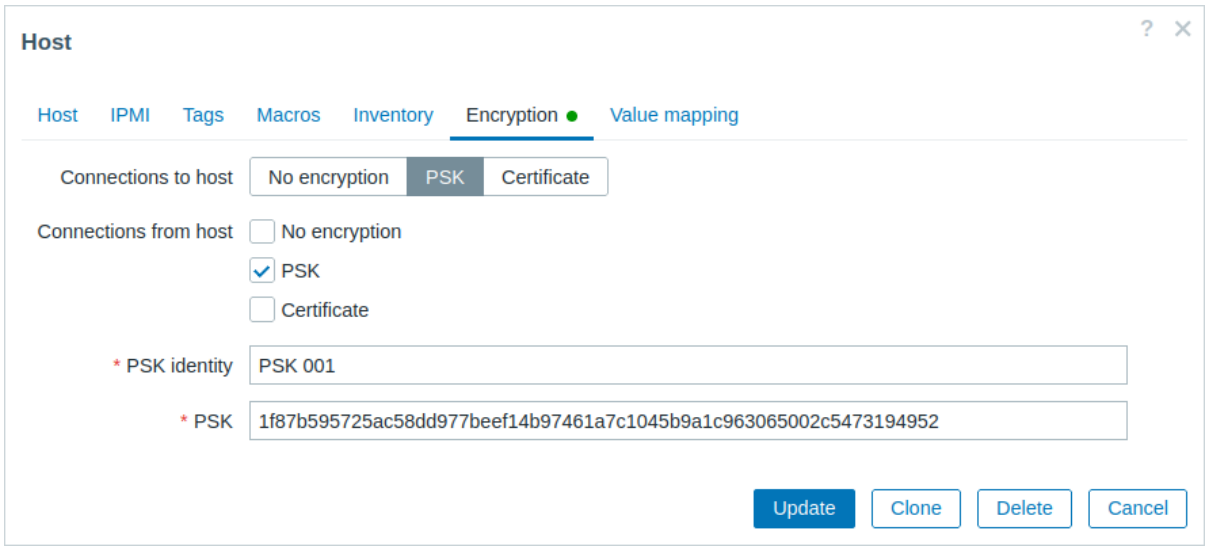
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(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 comunicação server - proxy ativo (exemplo)

No proxy, escreva o valor do PSK em um arquivo, por exemplo, `/home/zabbix/zabbix_proxy.psk`. O arquivo deve conter o PSK na primeira linha de texto, por exemplo:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

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 proxy `zabbix_proxy.conf`, por exemplo, defina:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

O proxy irá conectar ao server usando PSK. A identidade do PSK será "PSK 002".

(Para minimizar o tempo de inatividade, veja como alterar o tipo de conexão em Gerenciamento de criptografia de conexão).

Configure o PSK para este proxy no frontend do Zabbix. Vá para *Administração→Proxies*, selecione o proxy, vá para a guia "Criptografia". Em "Conexões do proxy" marque PSK. Cole no campo "Identidade do PSK" "PSK 002" e "e560cb0d918d26d31b4f642181f5f570ad89a
no campo "PSK". Clique em "Atualizar".

Reinicie o proxy. Ele começará a usar conexões criptografadas baseadas em PSK com o server. Verifique os arquivos de log do server 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 o proxy" no frontend do Zabbix como 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, todos podem funcionar como servidores e clientes TLS.
  Por exemplo, o Zabbix server conectando-se ao agent para uma verificação passiva, atua como cliente TLS. O agent está no papel de servidor TLS.
  O Zabbix agent, solicitando uma lista de verificações ativas ao proxy, atua como 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) frequentemente 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 Problemas de tipo de conexão ou permissão

O server está configurado para conectar com PSK ao agent, mas o agent aceita apenas conexões não criptografadas

No log do server ou proxy (com *GnuTLS* 3.3.16)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

No log do server ou proxy (com *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

Um lado conecta com certificado, mas o outro lado aceita apenas PSK ou vice-versa

Em qualquer log (com *GnuTLS*):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

Em qualquer log (com *OpenSSL* 1.0.2c):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
```

```
    TLS write fatal alert "handshake failure"
```

Tentando usar o Zabbix sender compilado com suporte a TLS para enviar dados ao Zabbix server/proxy compilado sem TLS

No log do lado da conexão:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a  20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

No log do lado de aceitação:

```
...falha ao aceitar uma conexão de entrada: de 127.0.0.1: suporte para TLS não foi compilado
```

Um lado conecta com PSK, mas o outro lado usa LibreSSL ou foi compilado sem suporte a criptografia

O LibreSSL não suporta PSK.

No log do lado que conecta:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

No log do lado que aceita:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

No frontend do Zabbix:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Um lado conecta com PSK, mas o outro lado usa OpenSSL com suporte a PSK desabilitado

No log do lado que está conectando:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

No log do lado que está aceitando:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 Problemas de certificado

OpenSSL usado com CRLs e para alguma CA na cadeia de certificados sua CRL não está incluída em TLSCRLFile

No log do servidor TLS no caso de peer *OpenSSL*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

No log do servidor TLS no caso de peer *GnuTLS*:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL expirada ou expira durante a operação do server

OpenSSL, no log do server:

- antes da expiração:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- após a expiração:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

O ponto aqui é que com uma CRL válida, um certificado revogado é relatado como ”certificate revoked”. Quando a CRL expira, a mensagem de erro muda para ”certificate expired”, o que é bastante enganoso.

GnuTLS, no log do server:

- antes e depois da expiração, o mesmo:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

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 o mesmo Issuer e Subject, embora tenha sido assinado por uma CA. Issuer e Subject 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 Issuer e Subject, execute:

```
openssl x509 -in <seucertificado.crt> -noout -text
```

É aceitável que o certificado raiz (de nível superior) tenha valores idênticos para Issuer e Subject.

3 Problemas com PSK

PSK contém um número ímpar de dígitos hexadecimais

O 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 de 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, informado como 128 dígitos hexadecimais).

Veja também: Limites de tamanho de valor

18 Interface web

Visão geral Para um fácil acesso ao Zabbix de qualquer lugar e de qualquer plataforma, a interface baseada na web é fornecida.

> **Note:**
> Se estiver usando mais de uma instância do frontend, certifique-se de que os locais e bibliotecas (LDAP, SAML etc.) estejam instalados e configurados de forma idêntica para todos os frontends.

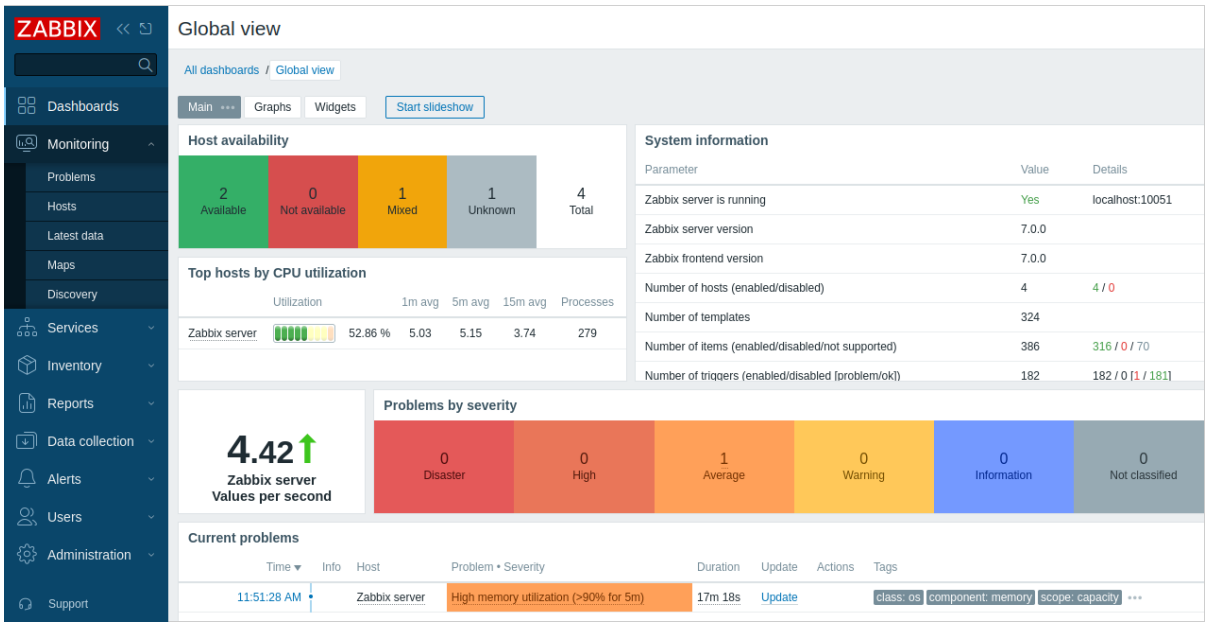
Ajuda do frontend Um link de ajuda é fornecido nos formulários do frontend do Zabbix com links diretos para as partes correspondentes da documentação.

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.

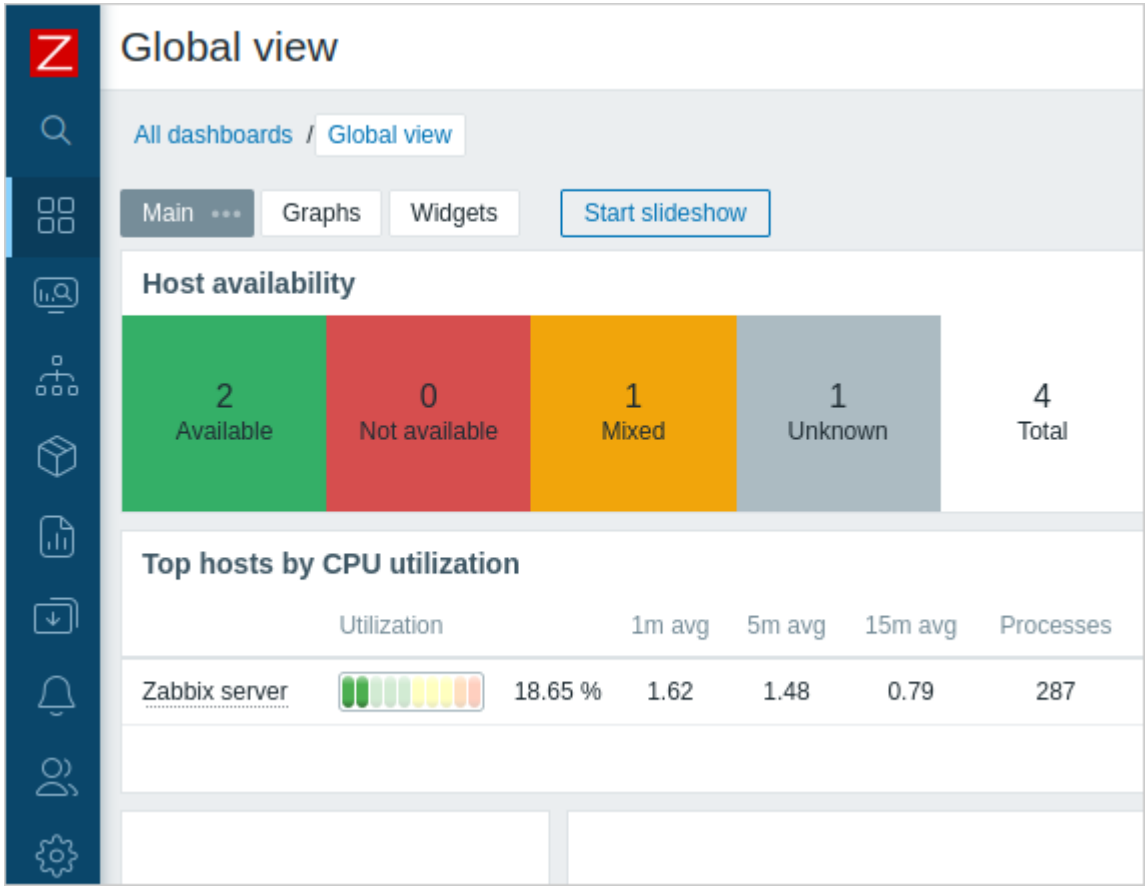


Trabalhando com o menu

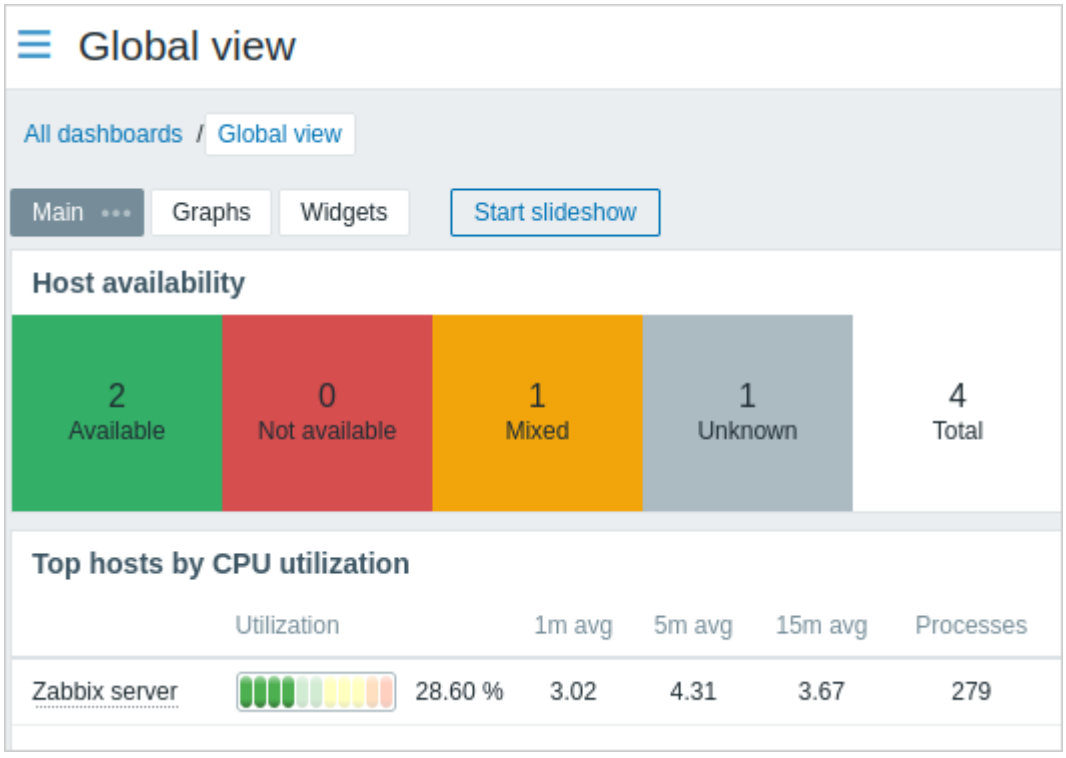
Uma caixa de pesquisa global está localizada abaixo do logo do Zabbix.

O menu pode ser recolhido ou ocultado completamente:

- Para recolher, clique em ao lado do logo do Zabbix. No menu recolhido, apenas os ícones ficam visíveis.



- Para ocultar, clique em  ao lado do logo do Zabbix. No menu oculto, tudo fica escondido.



Menu recolhido

Quando o menu é recolhido para exibir apenas 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 com que um menu recolhido reapareça 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.

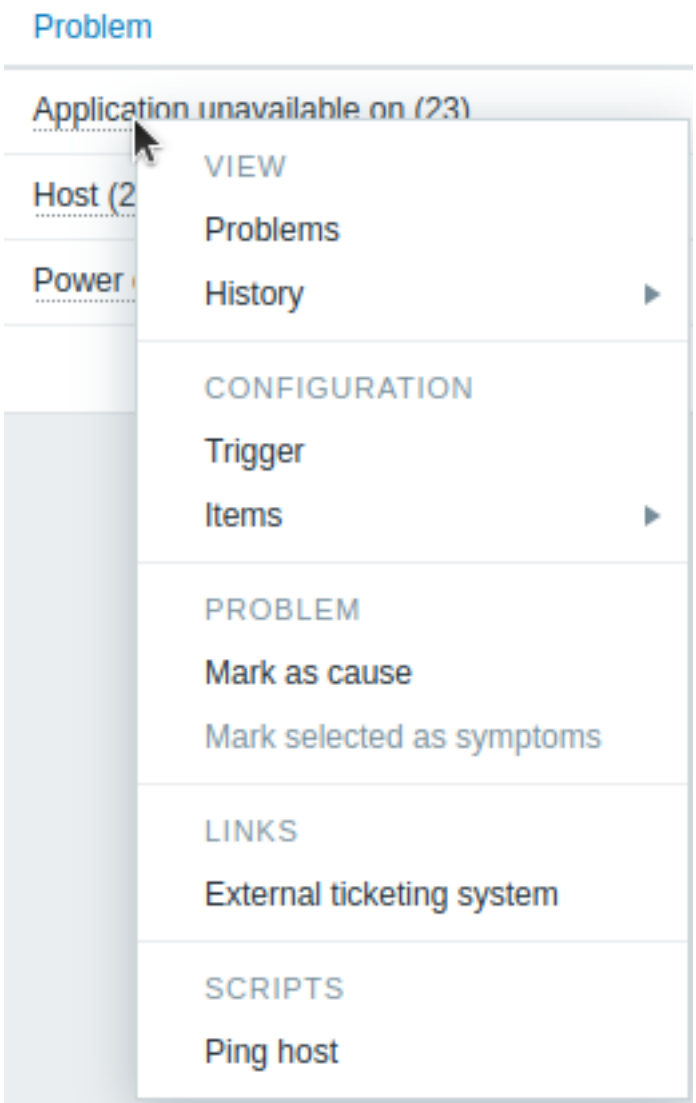
Os menus de contexto são acessíveis clicando no nome do host, item ou problema/trigger nos locais suportados.

1 Menu de evento

Visão geral

O menu de evento contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um evento.

O menu de evento 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 nos widgets *Visão geral dos 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 host

Visão geral

O menu de host contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um host.

O menu de host pode ser aberto clicando no nome do host ou no ícone de três pontos, dependendo da seção do frontend, por exemplo:

Monitoramento → Últimos dados *Coleta de dados → Hosts*

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 em 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:

- **Assistente de Host** - abre o Assistente de Host para o host atual (desabilitado para hosts descobertos).
- **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 de trigger configurada.
- acessar links personalizados configurados em Scripts globais (com escopo *Ação manual de 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 de host* para estarem disponíveis no menu do host.

Locais suportados

O menu do host é acessível clicando no nome do host em várias seções do frontend, por exemplo:

- Dashboards widgets, como Problemas, Principais items, Visão geral dos triggers, etc.
- Monitoramento → Problemas
- Monitoramento → Problemas → Detalhes do evento
- Monitoramento → Hosts
- Monitoramento → Hosts → Monitoramento Web
- Monitoramento → Últimos dados
- Monitoramento → Mapas
- Inventário → Hosts
- Relatórios → Top 100 triggers

O menu do host é acessível clicando no ícone de três pontos em Coleta de dados → Hosts.

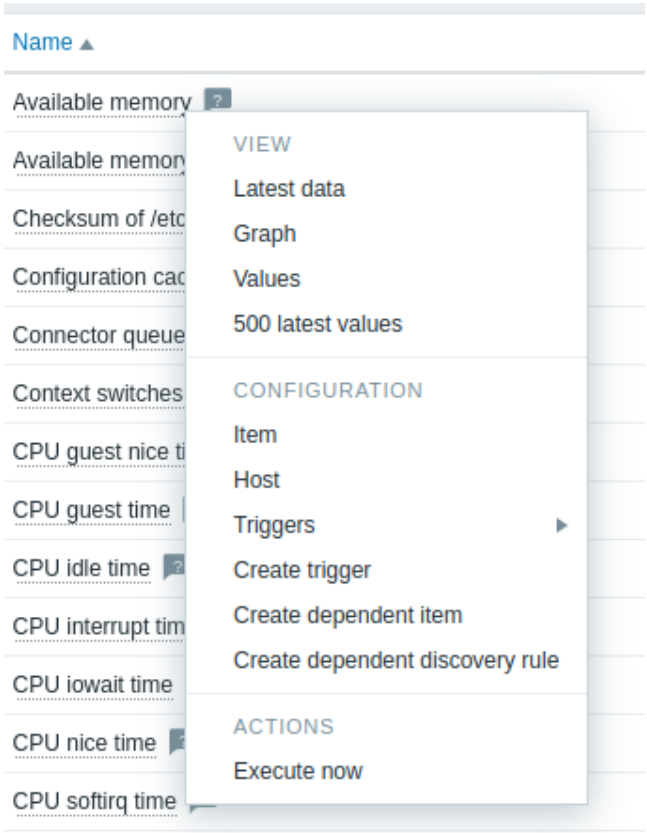
3 Menu de item

Visão geral

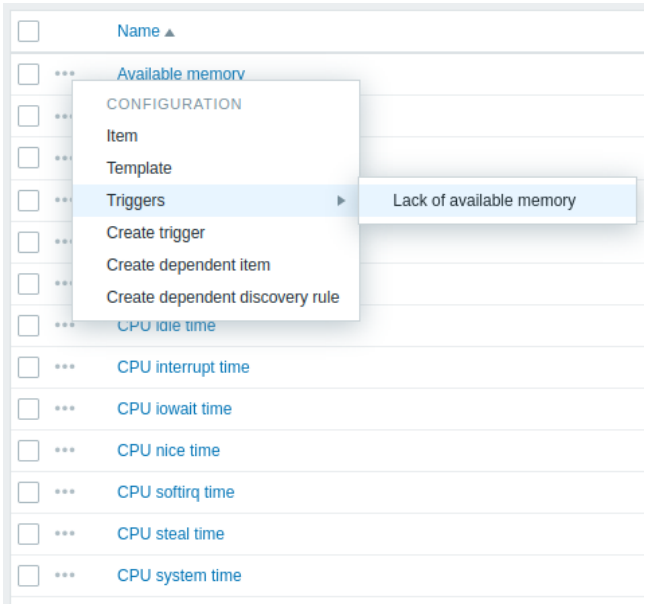
O menu de item contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um item.

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 do 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 (disponível apenas quando o menu do item é acessado em *Coleta de dados* → *Templates* → *Itens*);
- **Host** - abre o formulário de configuração do host ao qual o item pertence;
- **Triggers** - ao passar o mouse, abre uma lista de triggers do item, se houver; ao clicar em uma trigger, abre seu formulário de configuração de trigger;
- **Criar trigger** - abre o formulário de configuração de trigger para criar uma 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

O menu de item é acessível clicando no valor de um item no widget de dashboard Itens principais.

2 Seções do frontend

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
    - Imagens
    - Mapeamento de ícones
    - Expressões regulares
    - Opções de exibição de triggers
    - Mapas geográficos
    - Módulos
    - 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

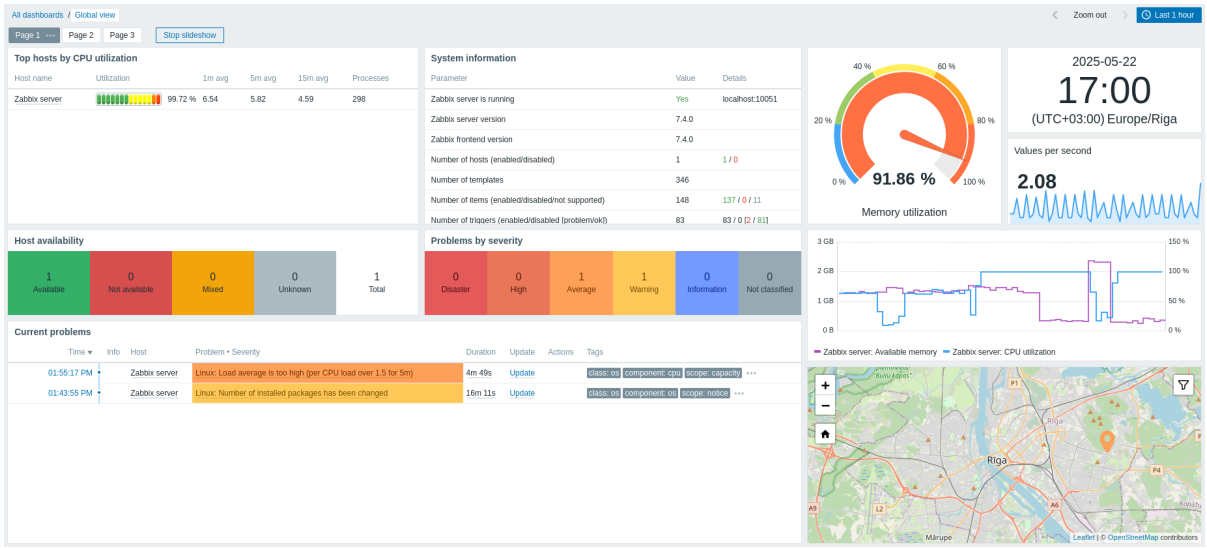
Visão geral

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 podem 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. Seu rótulo exibe o período de tempo atualmente selecionado. Clicar no rótulo permite expandir e recolher o seletor de período de tempo.

Observe que, quando o dashboard é exibido no modo quiosque e apenas widgets são exibidos, é possível reduzir o período do gráfico clicando duas vezes no gráfico.

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 é 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.

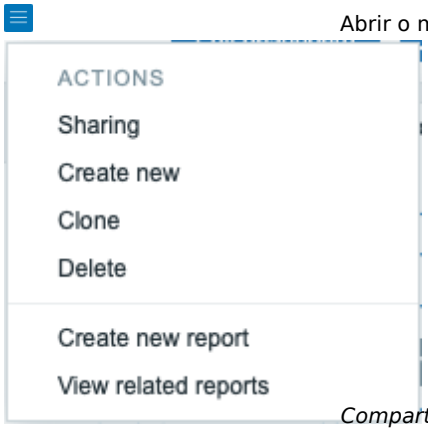
Visualizando um dashboard

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.



Abrir 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, você será solicitado a inserir 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 - abrir 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 - abrir 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.

	Exibir 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.

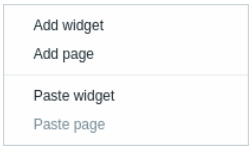
Para editar o status de compartilhamento de um dashboard, clique na opção *Compartilhamento* no menu de ações ao visualizar um único dashboard:

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 foi 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 foi 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:

dashboard_properties.png

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.

Parâmetro	Descrição
Iniciar apresentação de slides automaticamente	Marque esta caixa para executar uma apresentação de slides automaticamente quando existir mais de uma página de dashboard.

Ao clicar em *Aplicar*, um dashboard vazio é aberto:

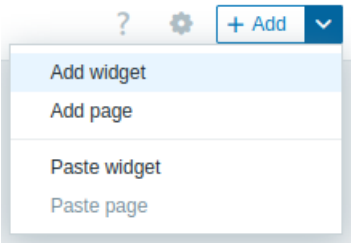
Para preencher o dashboard, você pode adicionar widgets e páginas.

Clique no botão *Salvar alterações* para salvar o dashboard. Se você clicar em *Cancelar*, o dashboard não será criado.

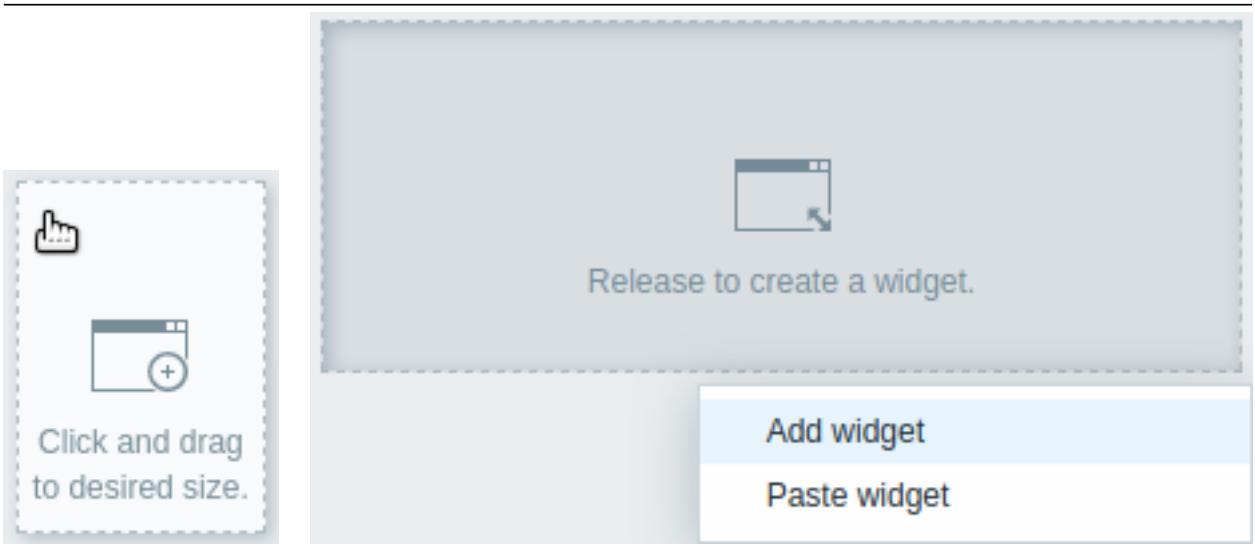
Adicionando widgets

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.

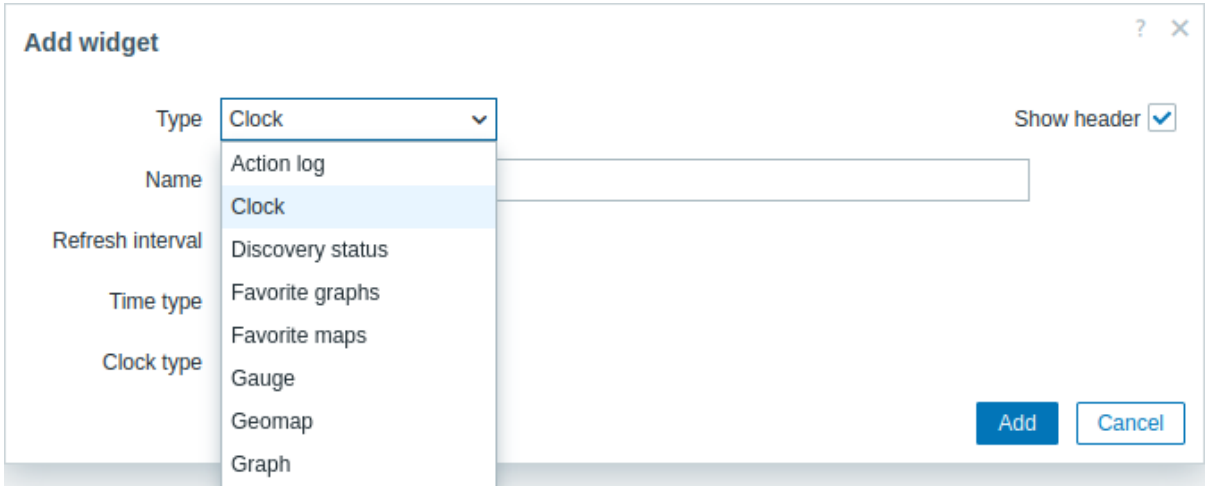


- Mova o mouse para um espaço vazio no dashboard. Observe que um espaço reservado aparece, então clique nele para abrir o formulário de configuração do widget. Após configurar o widget, ele será adicionado em seu tamanho padrão ou redimensionado para caber no espaço disponível. Alternativamente, arraste 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.



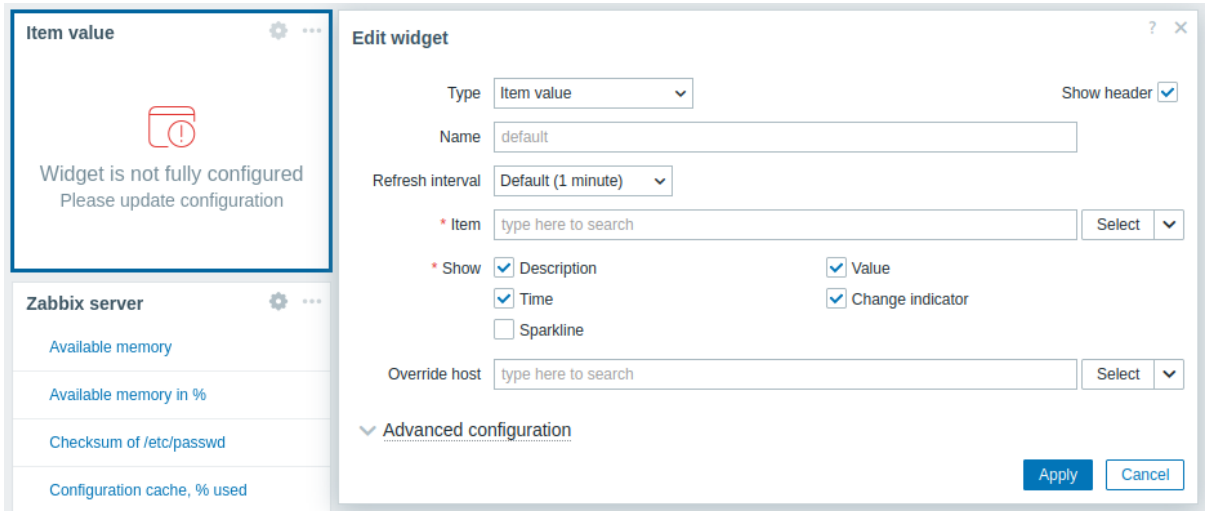
Editando widgets

Os widgets podem ser redimensionados e reposicionados no modo de edição do dashboard, arrastando sua barra de título. Você também pode reposicionar o formulário de configuração do widget arrastando sua barra de título.

Cada widget inclui controles no canto superior direito:

-  - editar o widget
- - abrir o menu do widget

Durante a edição, o widget é destacado e as alterações são visualizadas em tempo real. Se os parâmetros obrigatórios do widget não forem configurados, o widget entra no estado *Widget não está totalmente configurado*.



Após a edição, clique em *Aplicar* no widget e, em seguida, 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 menu do 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



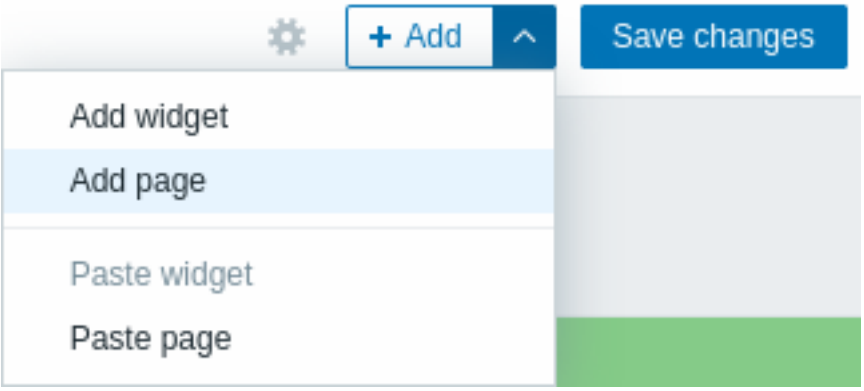
Os controles relacionados à apresentação de slides também estão disponíveis no modo quiosque (onde apenas o conteúdo da página é exibido):

-  - parar apresentação de slides
-  - iniciar apresentação de slides
-  - voltar uma página
-  - ir para a próxima página

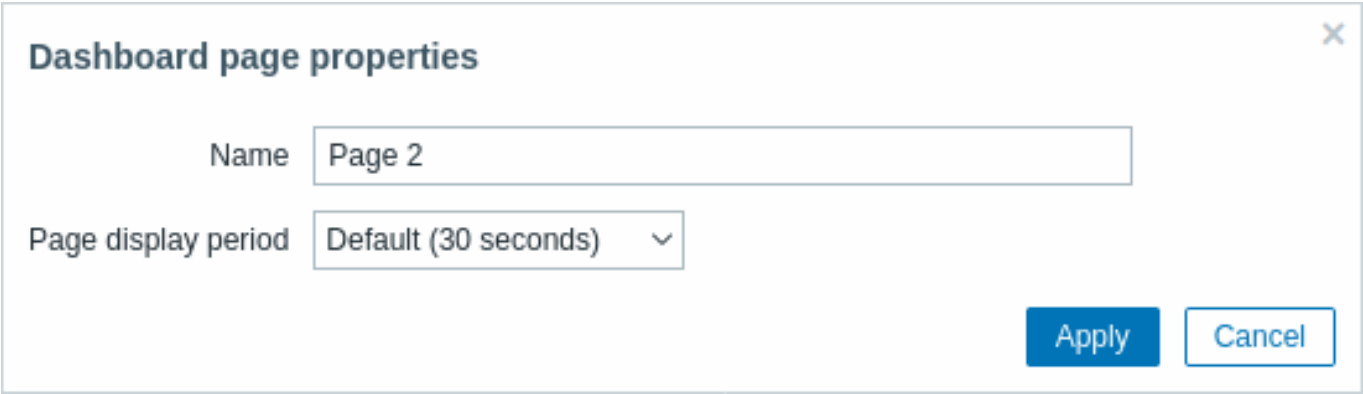
Adicionando páginas

Para adicionar uma nova página a um dashboard:

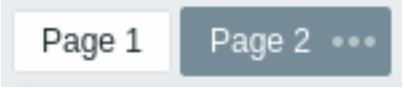
- Certifique-se de que o dashboard está no modo de edição
- Clique na seta ao lado do botão *Adicionar* e selecione a opção *Adicionar página*



- 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 guia (*Página 2*).



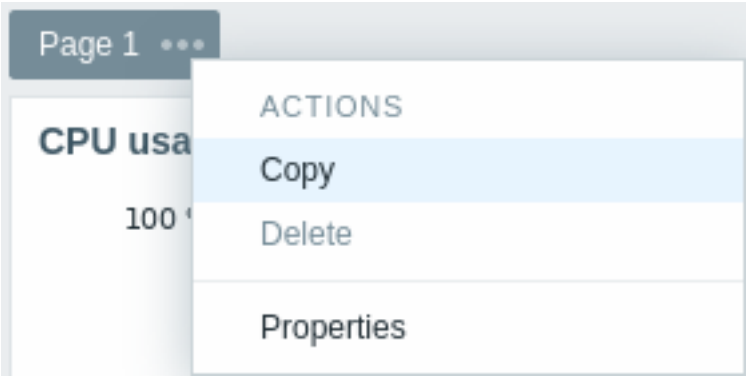
As páginas podem ser reordenadas arrastando e soltando as guias das páginas. A reordenação mantém a nomenclatura original das páginas. Sempre é possível ir para cada página clicando em sua guia.

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 já 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:

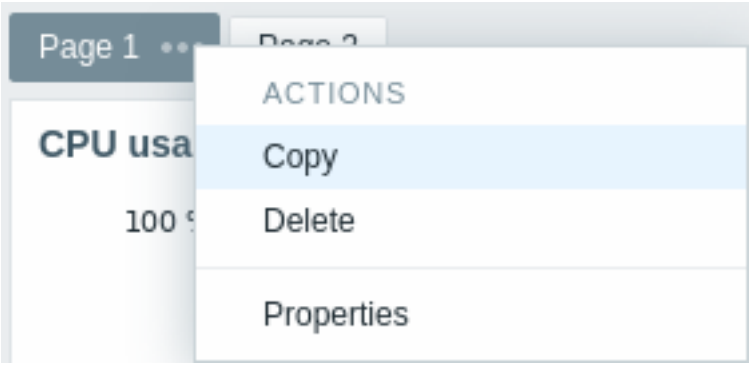


Para colar a página copiada:

- 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:

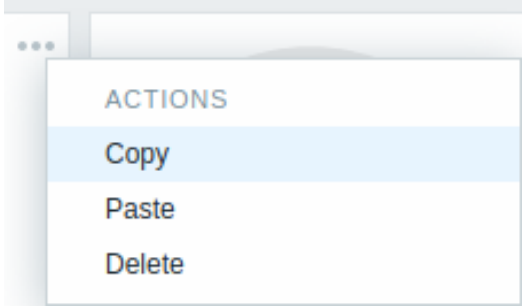


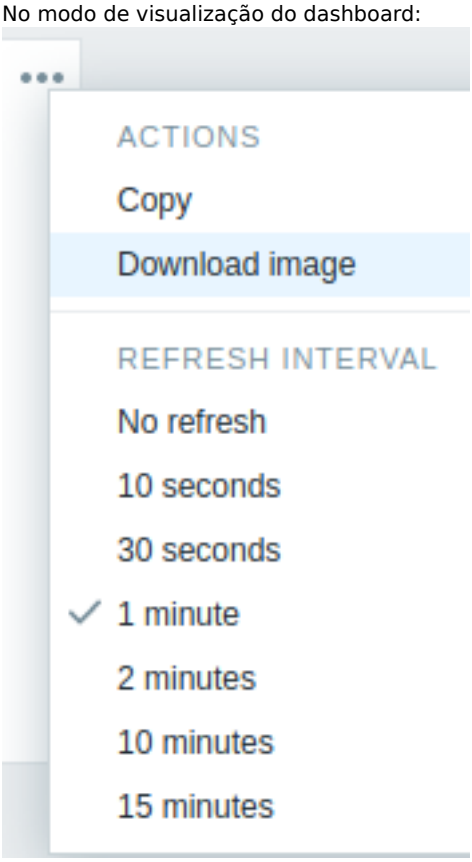
Ele contém as seguintes opções:

- *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

Menu do widget	Opções
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 regulares 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 da seguinte forma:

- Eles podem visualizar e clonar um dashboard público mesmo quando nenhum usuário ou grupo de usuários é 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 Widgets do dashboard

Visão geral

O Zabbix fornece uma variedade de widgets de dashboard que visualizam e resumem os dados de monitoramento, fornecendo uma visão clara do desempenho e disponibilidade do sistema.

Abaixo está uma visão geral de todos os widgets, agrupados por seu propósito principal.

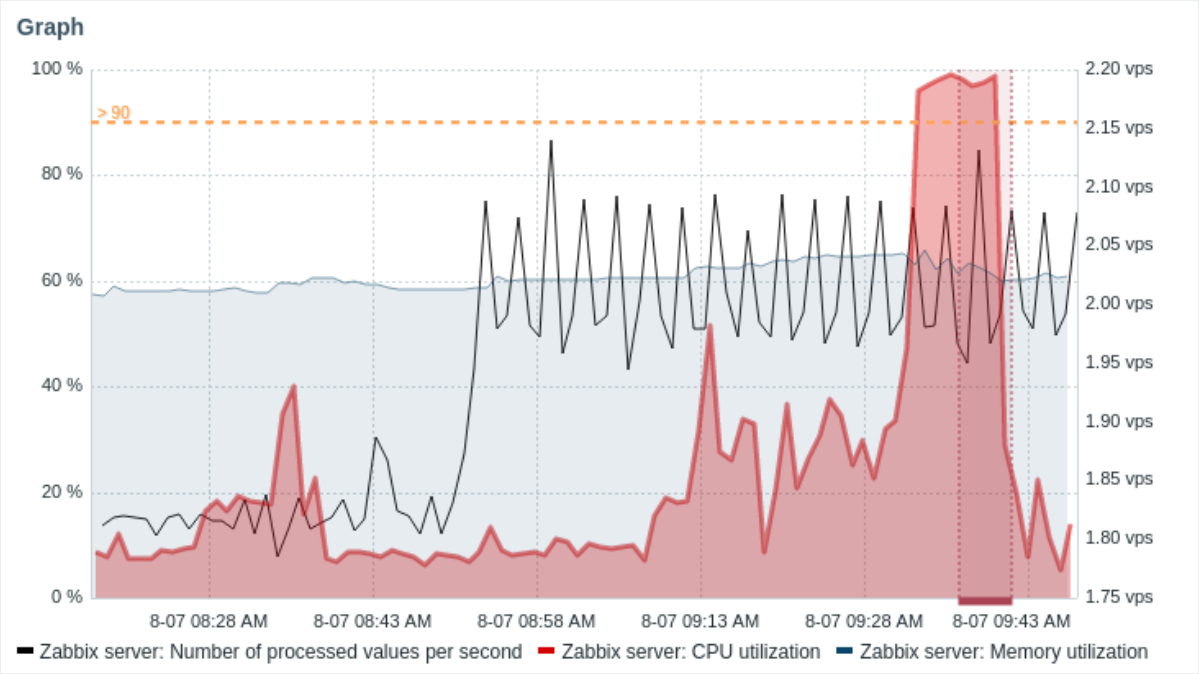
Para detalhes de configuração dos widgets, consulte Parâmetros dos widgets do dashboard.

Visualização de dados

Esses widgets apresentam dados em vários formatos visuais, ajudando você a monitorar dados e problemas de diferentes maneiras.

Gráfico

O widget Gráfico exibe dados numéricos de item como um gráfico vetorial. Ele pode ajudar você a acompanhar métricas, identificar problemas e comparar valores ao longo do tempo e entre hosts.



Widgets relacionados:

- Gráfico (clássico) - o widget Gráfico legado que exibe dados numéricos de item como um gráfico baseado em imagem.
- Protótipo de gráfico - exibe uma grade de gráficos personalizados criados automaticamente com base em protótipos de gráficos de regras de descoberta de baixo nível ou protótipos de item.
- Gráficos favoritos - exibe uma lista alfabética de atalhos para gráficos marcados como favoritos pelo usuário atual.

Gráfico de pizza

O widget Gráfico de pizza exibe dados numéricos de item como um gráfico de pizza ou rosquinha vetorial. Este widget pode ajudar a visualizar como os items ou hosts contribuem para o conjunto de dados geral.

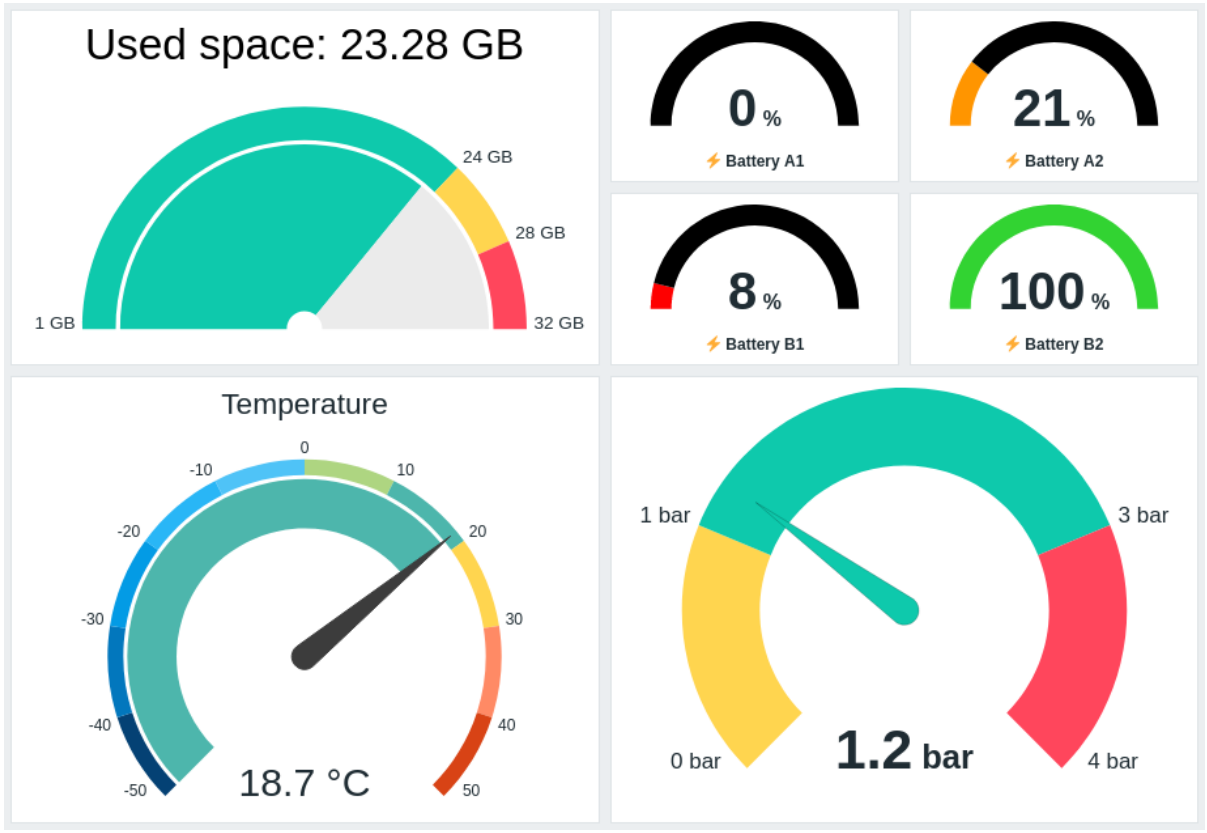
—

—

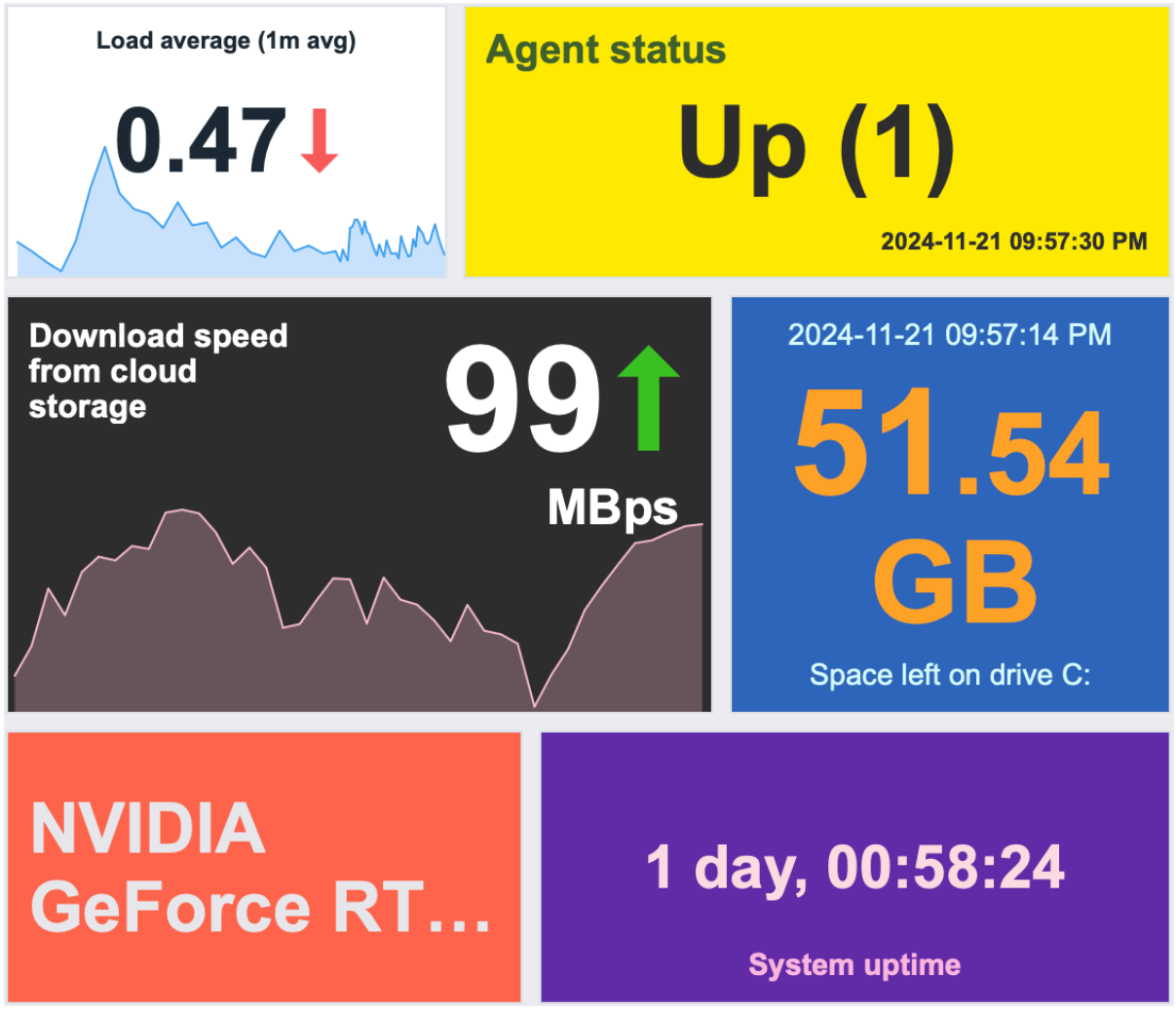
Medidor/Valor do item

Os widgets Medidor/Valor do item exibem o valor de um único item, útil para monitorar métricas importantes, visualizar limites e detectar mudanças repentinas nos dados.

Widget Medidor:



Widget Valor do item:



Histórico do item

O widget Histórico do item exibe os dados mais recentes para todos os tipos de item (numérico, texto, etc.) em formato de tabela. Ele pode mostrar barras de progresso e imagens (útil para itens de navegador) e pode destacar valores (útil para monitoramento de arquivos de log).

Principais items

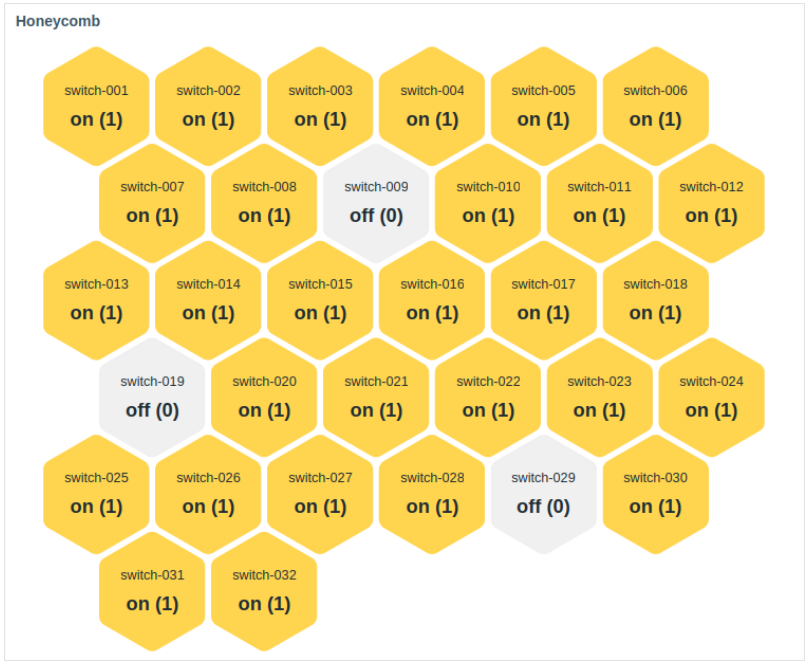
O widget Principais items exibe os maiores/menores valores para os items selecionados, fornecendo uma visão rápida do desempenho deles.

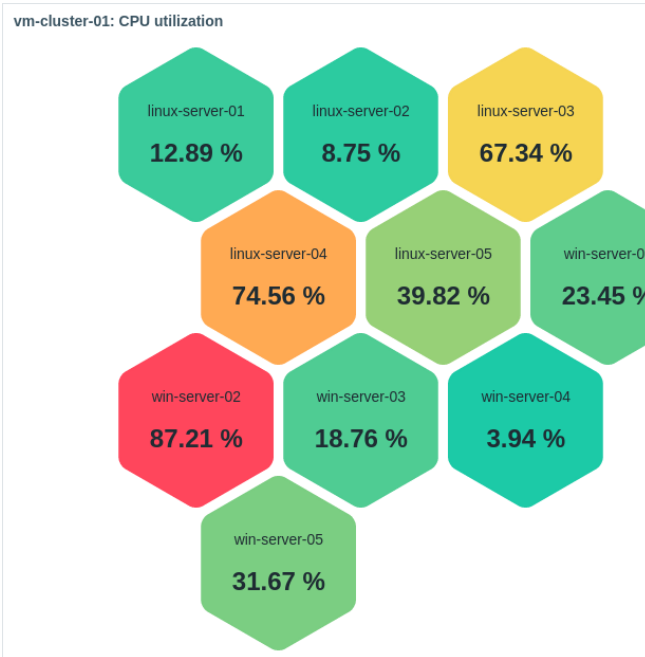
Widgets relacionados:

- Principais hosts - exibe os maiores/menores valores para os items selecionados de múltiplos hosts.
- Principais triggers - exibe triggers com o maior número de problemas.

Honeycomb

O widget Honeycomb exibe hosts ou items como células hexagonais em uma grade. Esse layout facilita a detecção de hosts/items com problemas, identificação de clusters de problemas, comparação de grupos rapidamente e muito mais.



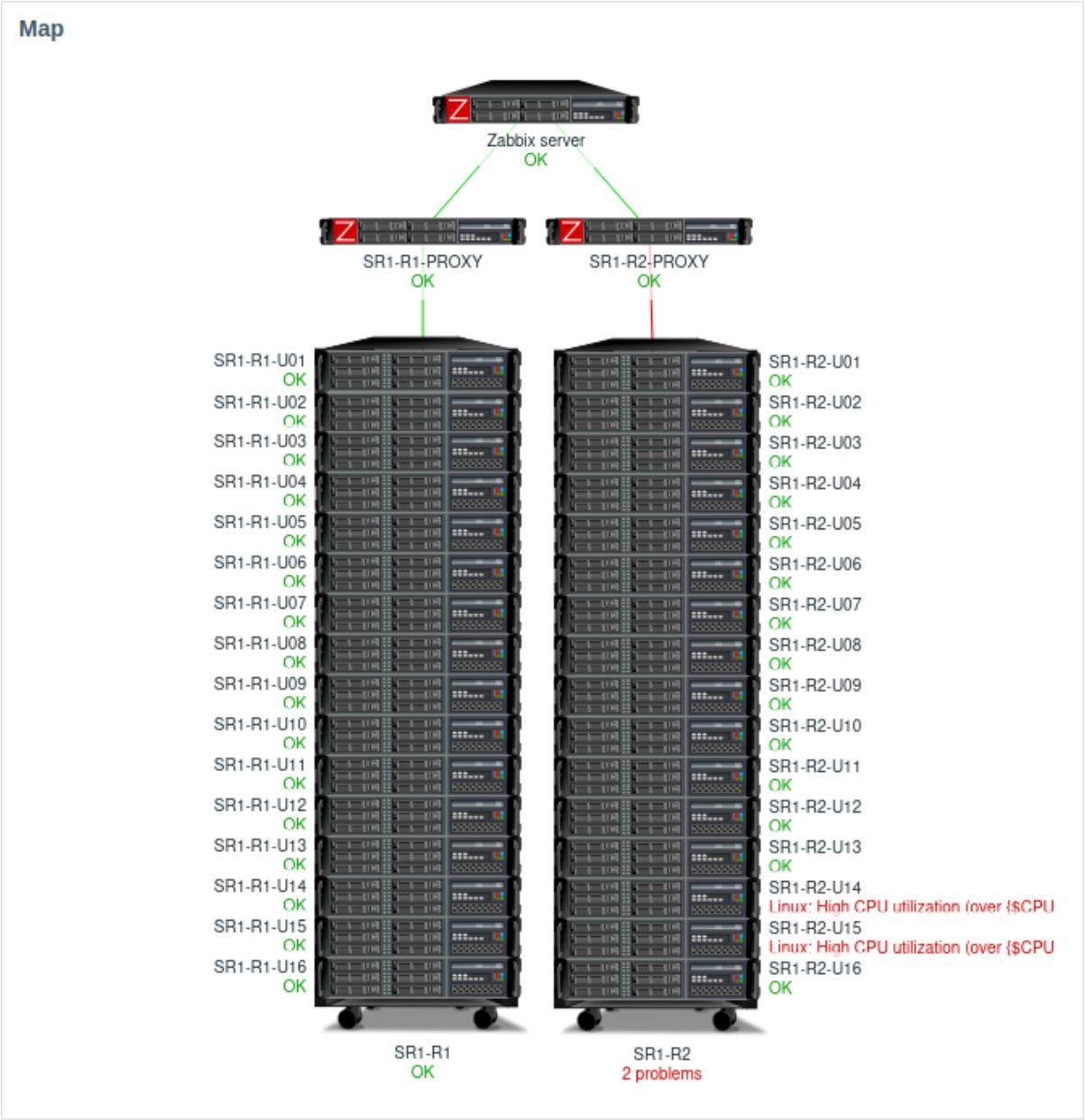


Mapas

Os widgets de mapa visualizam a topologia da rede ou a localização dos hosts, ajudando a avaliar a integridade geral e específica de cada local.

Mapa

O widget Mapa pode exibir um mapa de rede ou uma visualização semelhante, fornecendo uma visão geral dinâmica da sua rede.



Widgets relacionados:

- Árvore de navegação de mapas - exibe uma hierarquia dos mapas existentes e a contagem de problemas para cada mapa e grupo de mapas.
- Mapas favoritos - exibe uma lista de atalhos para mapas marcados como favoritos pelo usuário atual, classificados em ordem alfabética.

Geomap

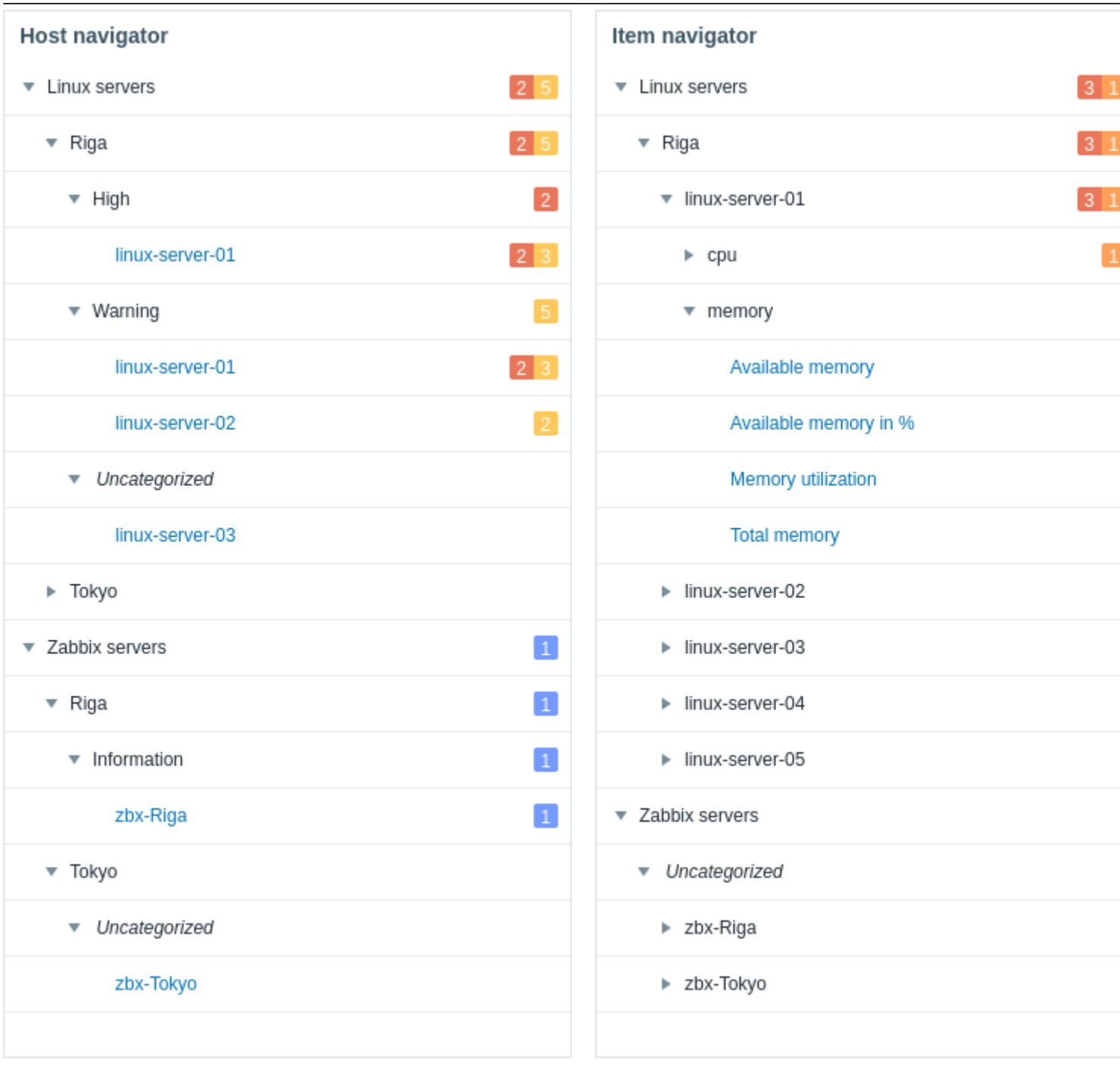
O widget Geomap exibe hosts como marcadores em um mapa interativo baseado em Leaflet, com as cores dos marcadores indicando o problema mais grave de cada host.

Detalhes do host/item

Esses widgets fornecem listas e visualizações detalhadas de hosts ou items.

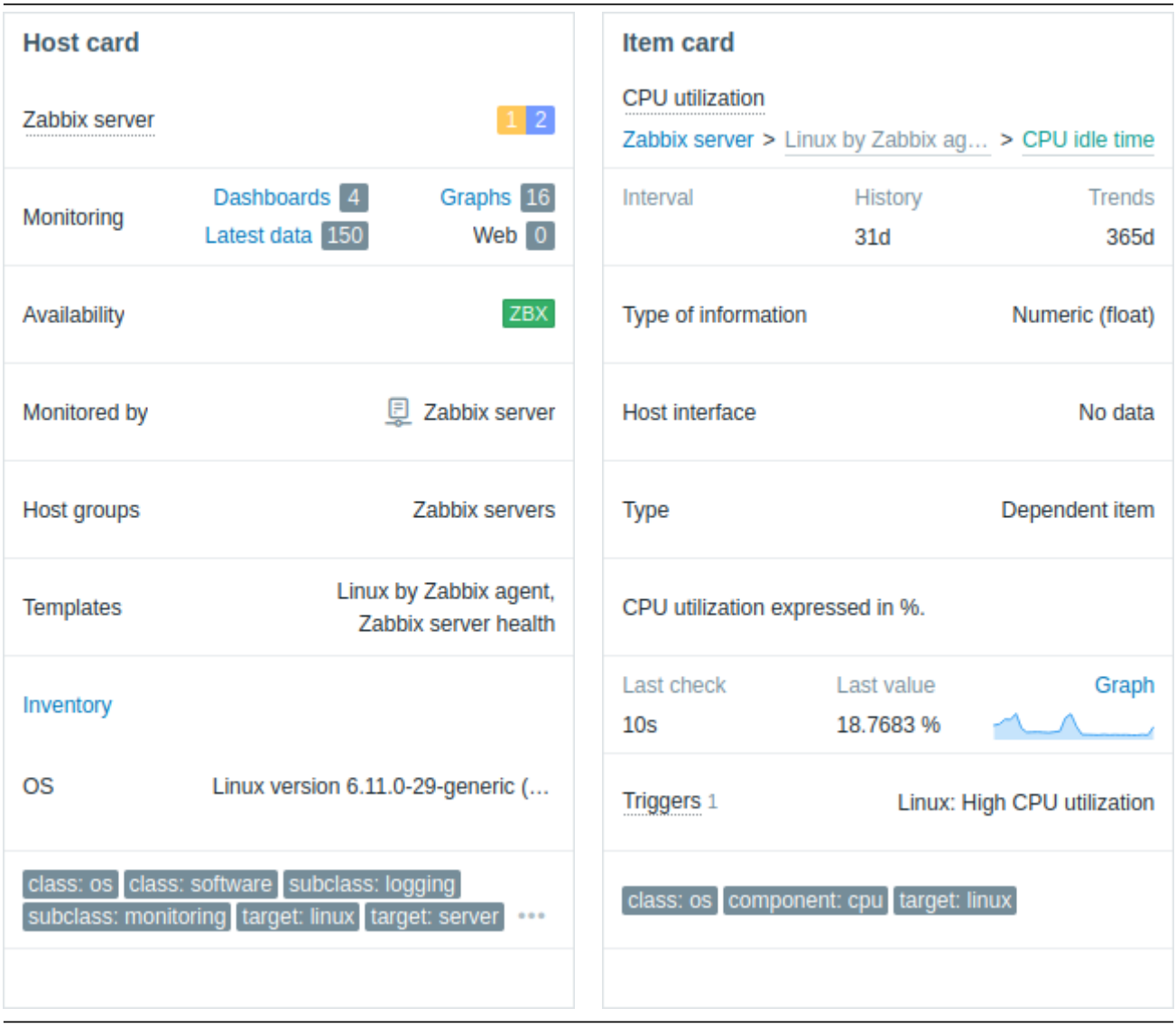
Navegador de Host/Item

Os widgets Navegador de host/Navegador de item exibem uma lista de hosts/items com base em várias opções de filtragem e agrupamento. Esses widgets são mais úteis para controlar o que outros widgets exibem com base no host/item selecionado.



Cartão de host/item

Os widgets Cartão de host/Cartão de item exibem detalhes de um único host/item, permitindo que você avalie rapidamente o status e a configuração de um host ou item (disponibilidade do host, problemas, inventário, dados mais recentes do item, erros, descrição, etc.).



Status e relatórios

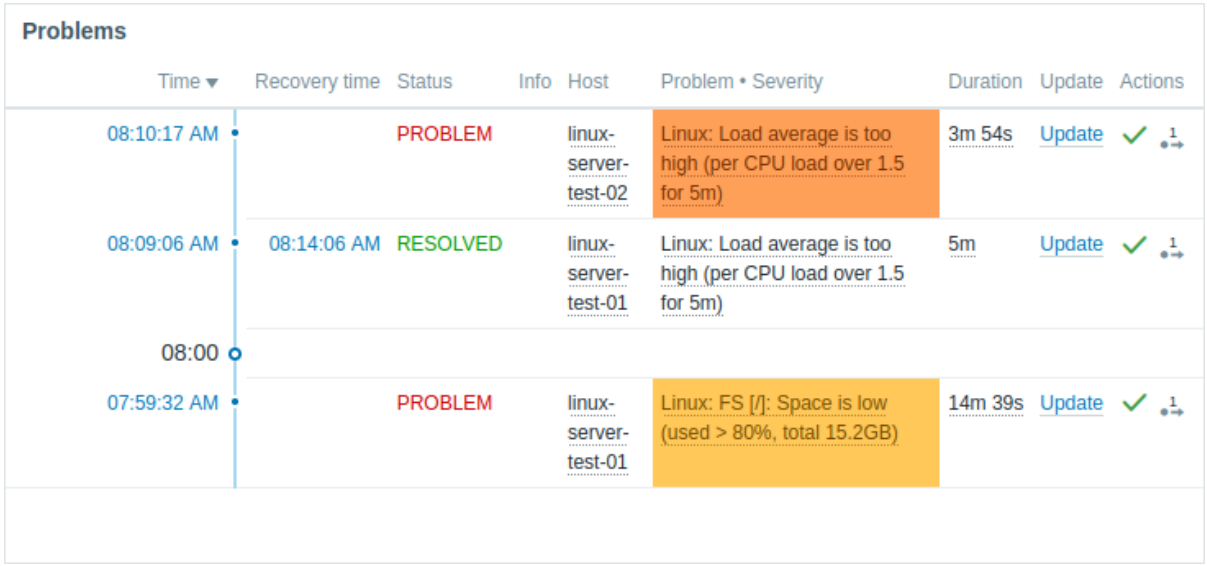
Esses widgets exibem dados de status (por exemplo, disponibilidade, triggers, problemas), bem como relatórios.

Disponibilidade do host

O widget Disponibilidade do host exibe a disponibilidade dos hosts nos grupos de hosts selecionados, permitindo monitorar o número de hosts ativos ou inativos.

Problemas

O widget Problemas exibe os problemas atuais filtrados por diferentes parâmetros (grupos de hosts, hosts, nomes de problemas, etc.), fornecendo uma visão clara do que precisa de atenção.



Widgets relacionados:

- Problemas por severidade - exibe o número de problemas de causa agrupados por severidade.
- Hosts com problemas - exibe o número de problemas de causa por grupo de hosts e mostra a maior severidade de problema em cada grupo.

Visão geral dos triggers

O widget Visão geral dos triggers exibe os estados atuais dos triggers dos hosts como uma tabela de blocos coloridos. Ele fornece uma visão rápida e visual da integridade e atividade dos triggers em diferentes hosts.

Log de ação

O widget Log de ação exibe detalhes das operações (notificações, comandos remotos) executadas em uma ação. O log detalhado auxilia na auditoria, solução de problemas e monitoramento da execução de cada ação.

Relatório de SLA

O widget Relatório de SLA exibe relatórios de SLA, ajudando você a monitorar o desempenho do serviço em relação às metas definidas.

Monitoramento web

O widget Monitoramento web exibe o status dos cenários web ativos.

Status da descoberta

O widget Status da descoberta exibe o status dos dispositivos descobertos por regras de descoberta de rede habilitadas. Ele ajuda você a monitorar se a descoberta de rede está funcionando corretamente e o número de dispositivos que estão ativos ou inativos.

Utilitários do dashboard e informações do sistema

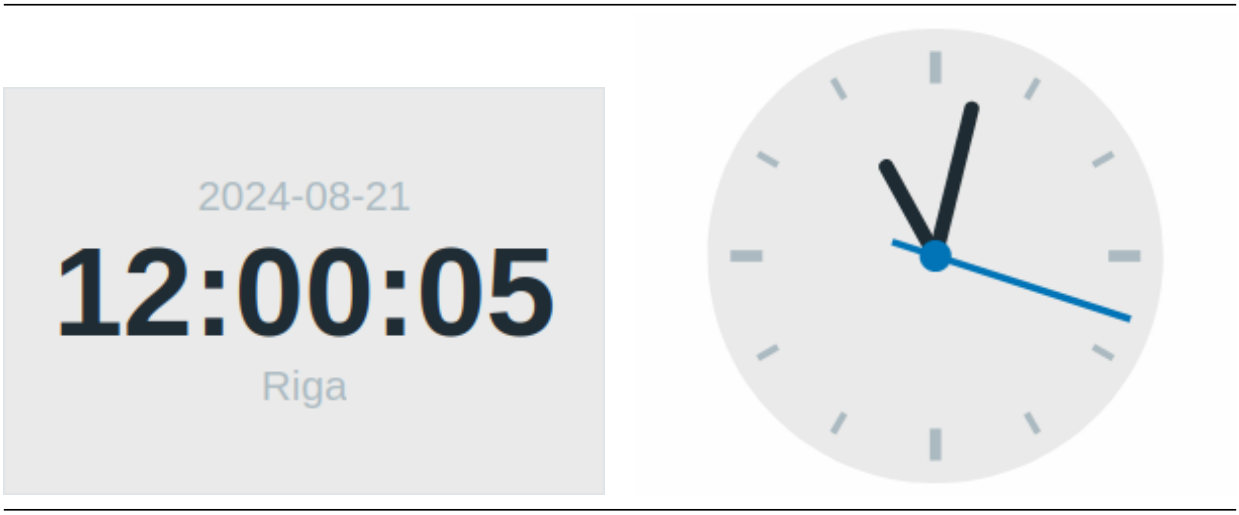
Esses widgets fornecem informações e ferramentas suplementares para aprimorar seus dashboards.

Informações do sistema

O widget Informações do sistema exibe um resumo dos principais dados do sistema e do Zabbix server ou detalhes dos nós de alta disponibilidade.

Relógio

O widget Relógio exibe a hora local, do server ou do host nos formatos analógico ou digital.



URL

O widget URL exibe o conteúdo recuperado de uma URL.

2 Parâmetros do widget do dashboard

Visão geral

Esta página descreve os parâmetros que são comuns a todos os widgets do dashboard, bem como os parâmetros dinâmicos que permitem que vários widgets troquem dados de configuração entre si ou com o dashboard.

Para ver os parâmetros específicos de cada widget, utilize a barra lateral para navegar até a página do widget correspondente.

Parâmetros comuns

Os seguintes parâmetros são comuns para todos os widgets:

Nome	Insira 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 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 único do usuário tem prioridade sobre a configuração do widget e é mantido 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 dinâmicos

Vários parâmetros de widget (por exemplo, *Hosts*, *Substituir host*, *Período de tempo*) suportam fontes de dados dinâmicas. Em vez de configurar valores estáticos, você pode vincular esses parâmetros a outros widgets ou ao próprio dashboard. Os widgets são então atualizados automaticamente com base nas seleções ou transmissões de outras partes do dashboard.

Compatibilidade dos widgets

Alguns widgets podem transmitir dados de configuração para outros widgets, alguns podem escutar dados e alguns podem fazer ambos. Por exemplo:

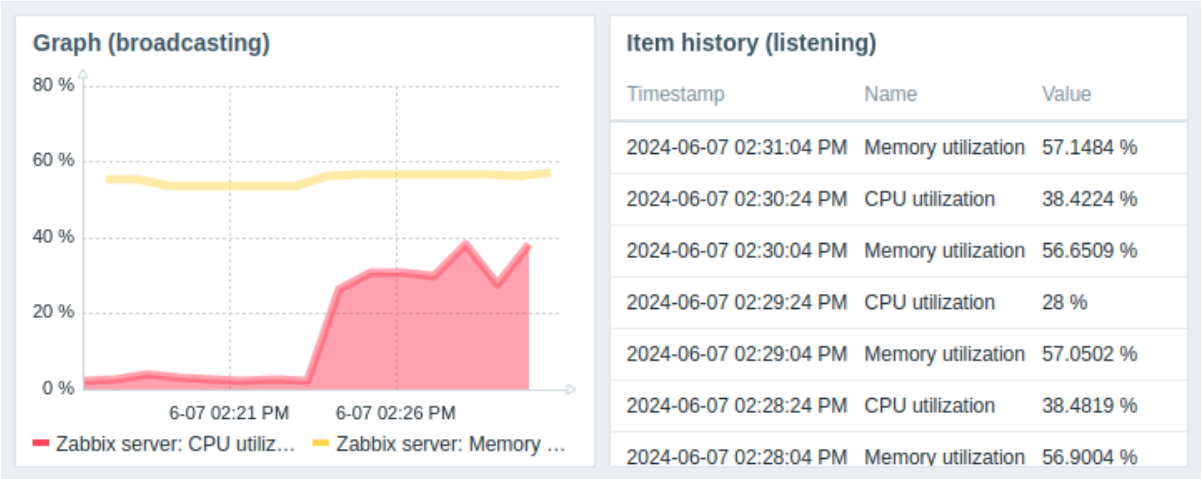
- O widget *Log de ações* só pode recuperar dados de período de tempo dos widgets *Gráfico*, *Gráfico (clássico)* e *Protótipo de gráfico*.
- O widget *Geomap* pode transmitir dados de host para widgets que os escutam (*Honeycomb*, *Itens principais*, etc.) e também pode escutar dados de grupo de hosts e hosts de widgets que os transmitem (*Honeycomb*, *Hosts com problemas*, etc.).
- O widget *Relógio* não pode transmitir nem escutar nenhum dado.

A tabela a seguir descreve as capacidades de transmissão e escuta de cada widget.

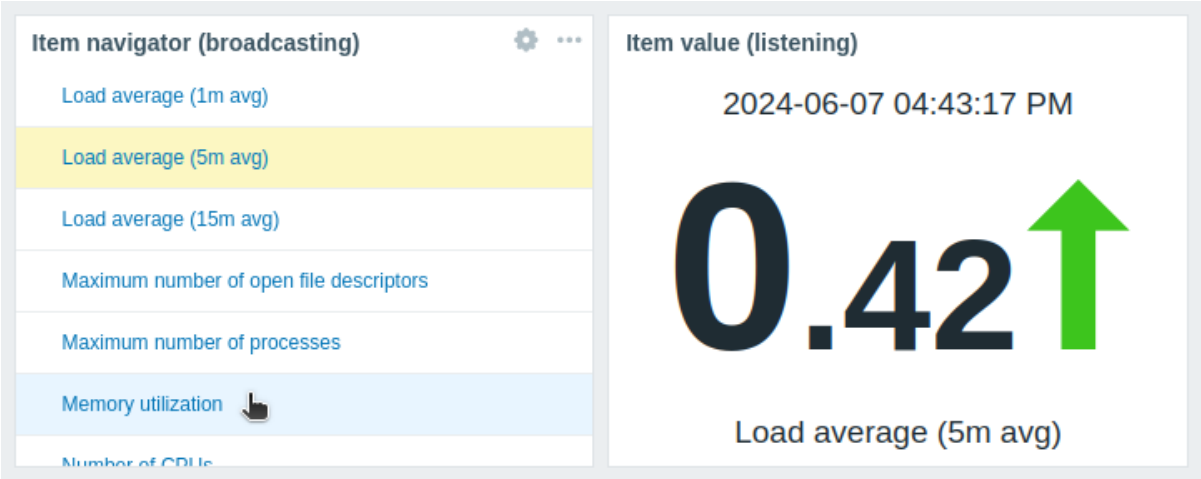
Widget	Transmite	Escuta
Log de ações	-	Período de tempo
Relógio	-	-
Status de descoberta	-	-
Gráficos favoritos	-	-
Mapas favoritos	-	-
Medidor	-	Hosts, Itens
Geomap	Hosts	Grupos de hosts, Hosts
Gráfico	Período de tempo	Período de tempo, Itens, Hosts
Gráfico (clássico)	Período de tempo	Hosts, Itens, Gráficos, Período de tempo
Protótipo de gráfico	Período de tempo	Hosts, Período de tempo
Honeycomb	Hosts, Itens	Grupos de hosts, Hosts
Disponibilidade do host	-	Grupos de hosts
Cartão do host	-	Hosts
Navegador de hosts	Hosts	Grupos de hosts
Cartão do item	-	Hosts, Itens, Período de tempo
Histórico do item	Itens	Hosts, Período de tempo
Navegador de itens	Itens	Grupos de hosts, Hosts
Valor do item	-	Hosts, Itens, Período de tempo
Mapa	Grupos de hosts, Hosts	Mapas
Árvore de navegação de mapas	Mapas	-
Gráfico de pizza	-	Período de tempo, Itens
Hosts com problemas	Grupos de hosts	Grupos de hosts, Hosts
Problemas	Eventos	Grupos de hosts, Hosts
Problemas por severidade	Grupos de hosts	Grupos de hosts, Hosts
Relatório de SLA	-	-
Informações do sistema	-	-
Hosts principais	Hosts	Grupos de hosts, Hosts
Itens principais	-	Grupos de hosts, Hosts
Triggers principais	-	Período de tempo
Visão geral dos triggers	-	Grupos de hosts, Hosts
URL	-	Hosts
Monitoramento web	Grupos de hosts	Grupos de hosts, Hosts

Os widgets diferem em como **transmitem** dados para outros widgets.

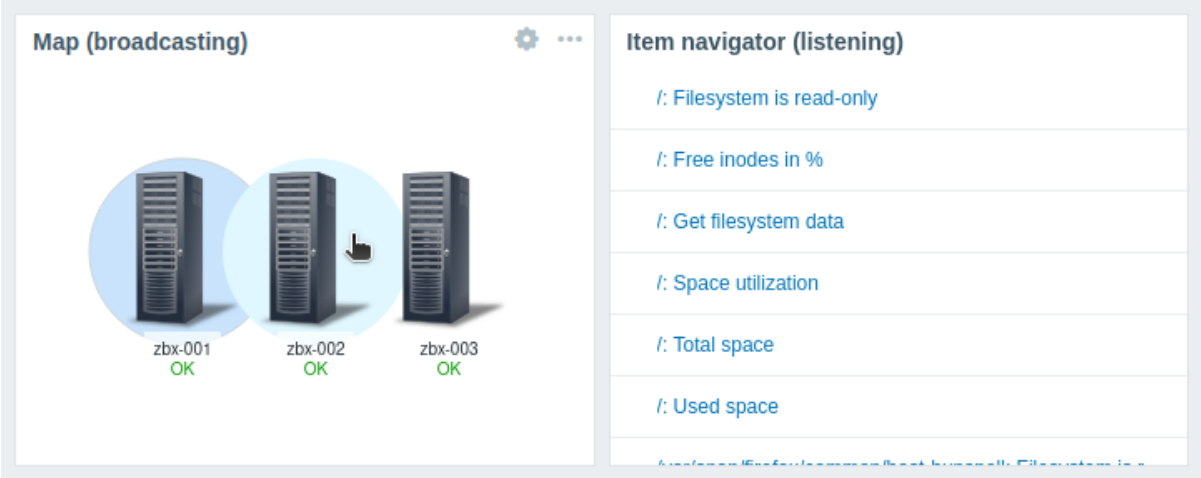
Os widgets que suportam transmissão começam a fazê-lo automaticamente quando criados. Por exemplo, o widget *Gráfico* transmite imediatamente os dados do período de tempo para os widgets ouvintes.



Os widgets que suportam seleção de entidade 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.

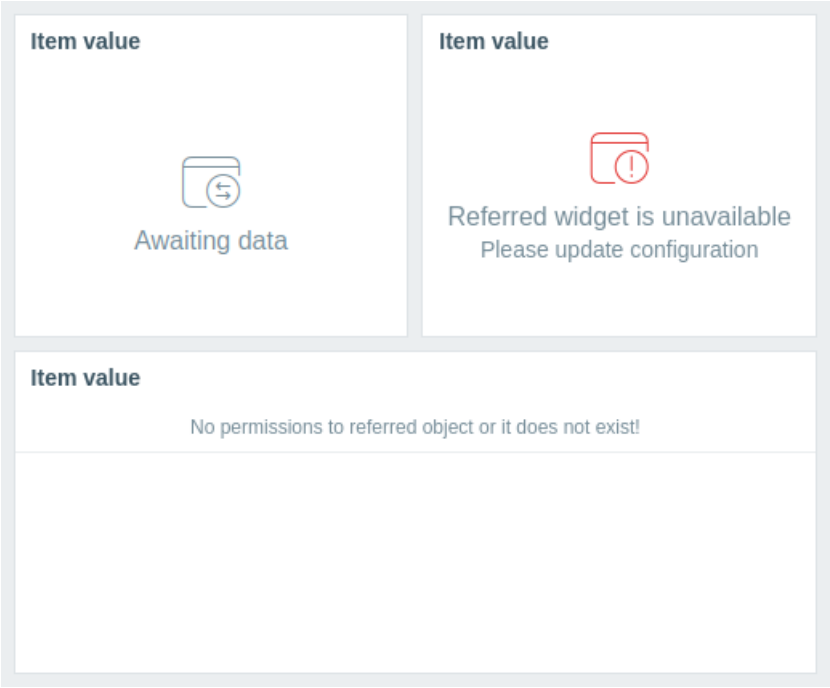


O widget *Mapa* se comporta de maneira semelhante, transmitindo dados para o primeiro elemento mais próximo do canto superior esquerdo da área visível do widget (para o widget *Geomap*, o mais próximo do centro). Quando um elemento diferente é selecionado, ele transmite dados para esse elemento. Ao passar o mouse, o elemento é destacado em azul claro; ao selecionar, é destacado em azul escuro.



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 tiver sido 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 possuir a entidade configurada no widget ouvinte (item, gráfico, mapa, etc.) ou se o usuário não tiver permissões para acessar o host, o widget ouvinte exibirá a seguinte mensagem: *"Sem permissão para o objeto referenciado ou ele não existe!"*



1 Log de ações

Visão geral

O widget *Log de ação* exibe detalhes das operações (notificações, comandos remotos) executadas dentro de uma ação. O log detalhado auxilia na auditoria, solução de problemas e monitoramento da execução de cada ação.

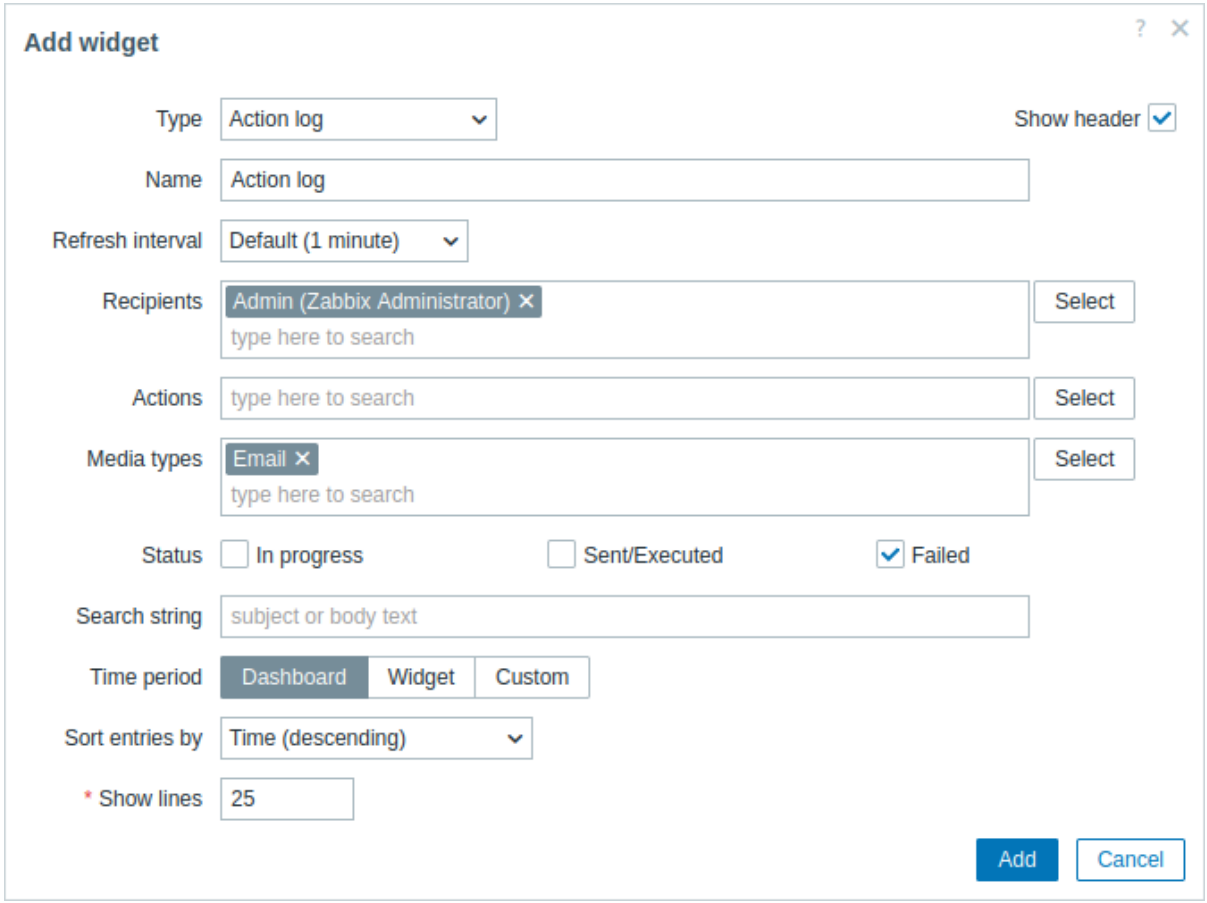
Action log

Time ▾	Action	Media type	Recipient	Message	Status	Info
2025-08-15 07:58:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Resolved in 1m 59s: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem has been resolved at 10:58:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Problem duration: 1m 59s Host: Linux server Severity: Average Original problem ID: 45	Sent	
2025-08-15 07:56:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Problem: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem started at 10:56:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Host: Linux server Severity: Average Operational data: Load averages(1m 5m 15m): (6.256836 4.808105 3.398438), # of CPUs: 2 Original problem ID: 45	Sent	
2025-08-15 07:42:30 AM	Report problems to Zabbix administrators		Admin (Zabbix Administrator)		Failed	i

Ele exibe os mesmos dados de *Relatórios* > *Log de ação* e pode exibir até 1000 registros.

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:

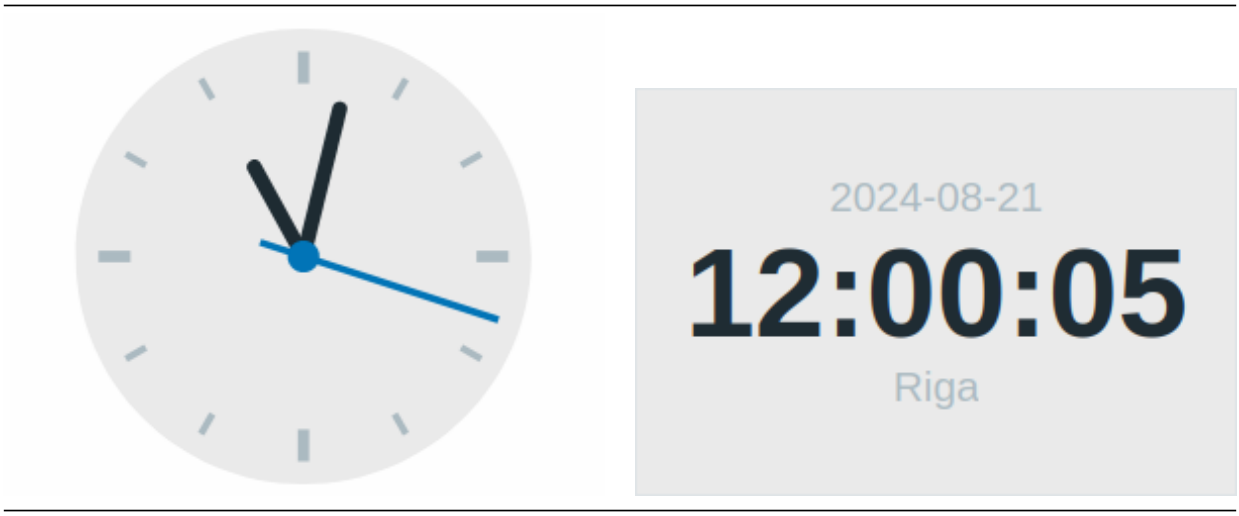
Destinatários	Filtra as entradas por destinatários. Este campo é de preenchimento automático, então ao começar a digitar o nome de um destinatário será exibida uma lista suspensa de destinatários correspondentes. Se nenhum destinatário for selecionado, serão exibidos detalhes das operações de ação para todos os destinatários.
Ações	Filtra as entradas por ações. Este campo é de preenchimento automático, então ao começar a digitar o nome de uma ação será exibida uma lista suspensa de ações correspondentes. Se nenhuma ação for selecionada, serão exibidos detalhes das operações de ação para todas as ações.
Tipos de mídia	Filtra as entradas por tipos de mídia. Este campo é de preenchimento automático, então 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, serão exibidos detalhes das operações de ação para todos os tipos de mídia.
Status	Marque a caixa de seleção para filtrar as entradas pelo respectivo status: **Em andamento** - são exibidas operações de ação que estão em andamento; **Enviado/Executado** - são exibidas operações de ação que enviaram uma notificação ou foram executadas; **Falhou** - são exibidas operações de ação que falharam.
String de busca	Filtra as entradas pelo conteúdo da mensagem/comando remoto. Se você inserir uma string aqui, apenas 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** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
Widget	Insira 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	Insira 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	Insira 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”.
Ordenar entradas por	Ordena as entradas por: **Tempo** (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

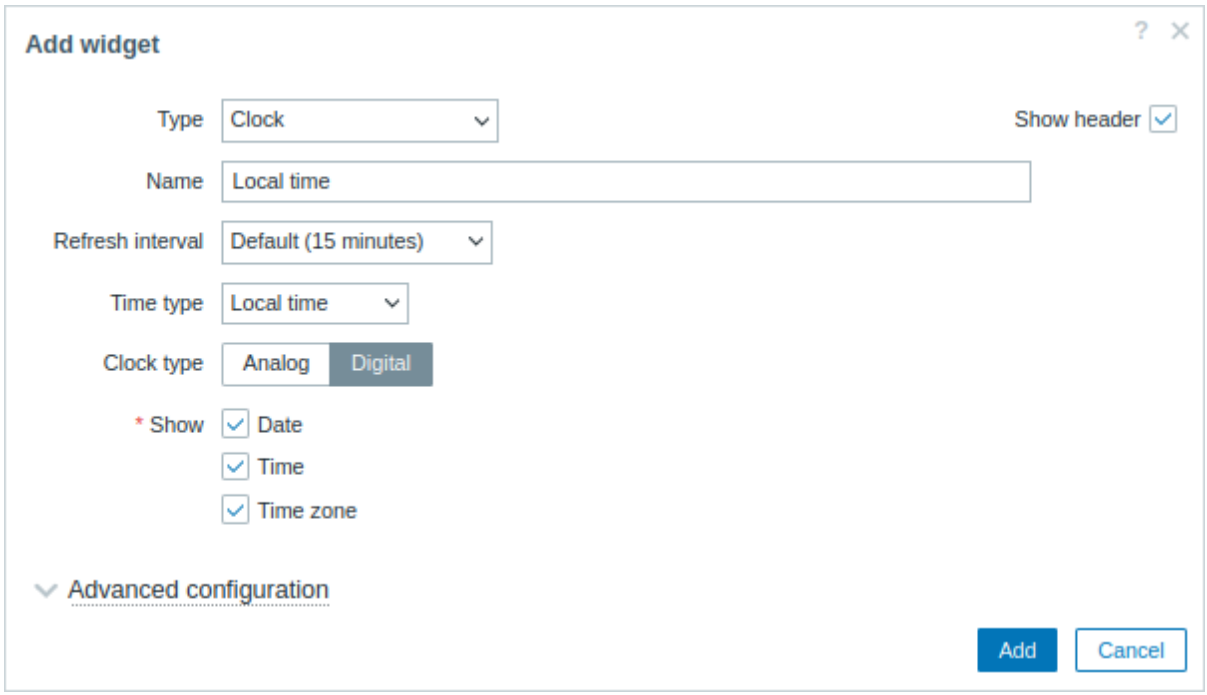
Visão geral

O widget *Relógio* exibe a hora local, do servidor ou do host no formato analógico ou digital.



Configuração

Para configurar, selecione *Relógio* como tipo:



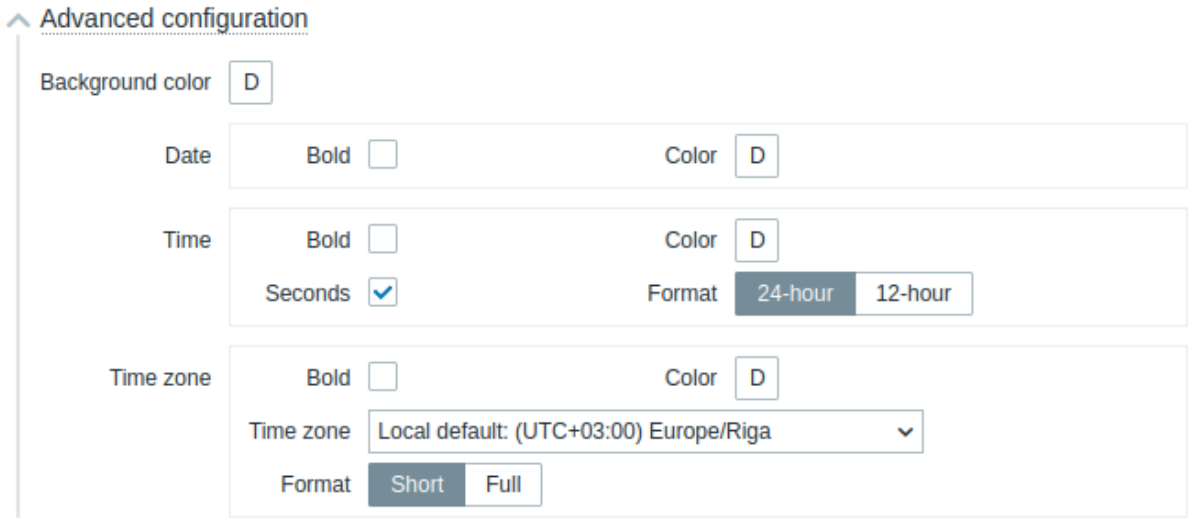
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Tipo de hora	Selecione a fonte de hora: **Local** - hora do sistema do dispositivo executando o navegador; **Servidor** - fuso horário definido globalmente ou para o usuário Zabbix; **Host** - hora obtida de um item do host.
Item	Selecione o item para a hora do host (use o item system.localtime[local]). Este parâmetro está disponível se o *Tipo de hora* estiver definido como "Hora do host".
Tipo de relógio	Selecione o tipo de relógio (analógico ou digital).
Mostrar	Selecione uma ou mais unidades de informação (data, hora, fuso horário) para exibir no relógio digital. O tamanho da unidade se adapta automaticamente às unidades exibidas, ao tamanho do widget e à escala de exibição. Este parâmetro está disponível se o *Tipo de relógio* estiver definido como "Digital".
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada para o relógio digital. Esta seção está disponível se o *Tipo de relógio* estiver definido como "Digital".

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.



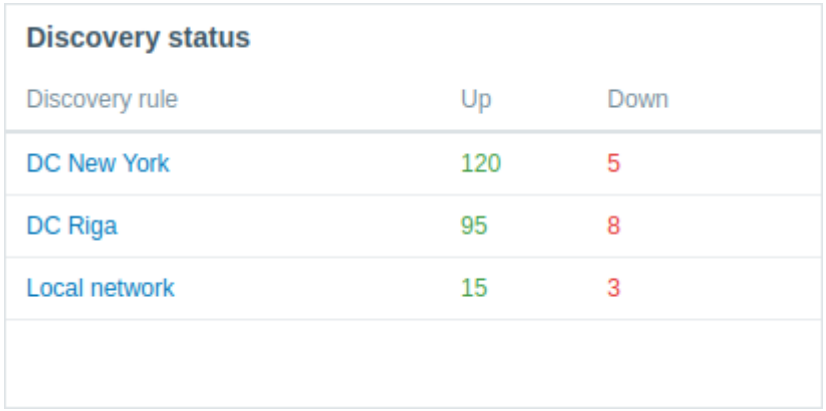
Cor de fundo	Selecione a cor de fundo no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Data	
Negrito	Marque a caixa de seleção para exibir a data em negrito.
Cor	Selecione a cor da data no seletor 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`).

3 Status da descoberta

Visão geral

O widget *Status de descoberta* exibe o status dos dispositivos descobertos por regras de descoberta de rede habilitadas. Ele ajuda a monitorar se a descoberta de rede está funcionando corretamente e o número de dispositivos que estão ativos ou inativos.



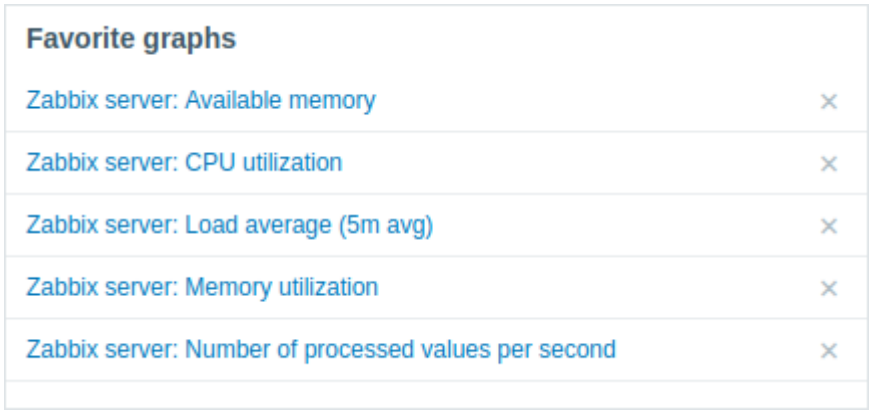
Configuração

Todos os parâmetros de configuração para este widget são comuns a todos os widgets.

4 Gráficos favoritos

Visão geral

O widget *Gráficos favoritos* exibe uma lista alfabética de atalhos para gráficos marcados como favoritos pelo usuário atual.



Clicar em x remove o gráfico da lista e dos favoritos.

Os gráficos são marcados como favoritos ao clicar no botão  *Adicionar aos favoritos* ao visualizar gráficos de item em *Monitoramento* > *Últimos dados*.



Configuração

Todos os parâmetros de configuração para este widget são comuns a todos os widgets.

5 Mapas favoritos

Visão geral

O widget *Mapas favoritos* exibe uma lista de atalhos para mapas marcados como favoritos pelo usuário atual, ordenados alfabeticamente.

Clicar em x remove o mapa dos favoritos.

Os mapas são marcados como favoritos clicando no botão *Adicionar aos favoritos* ao visualizar mapas em *Monitoramento* > *Mapas*.

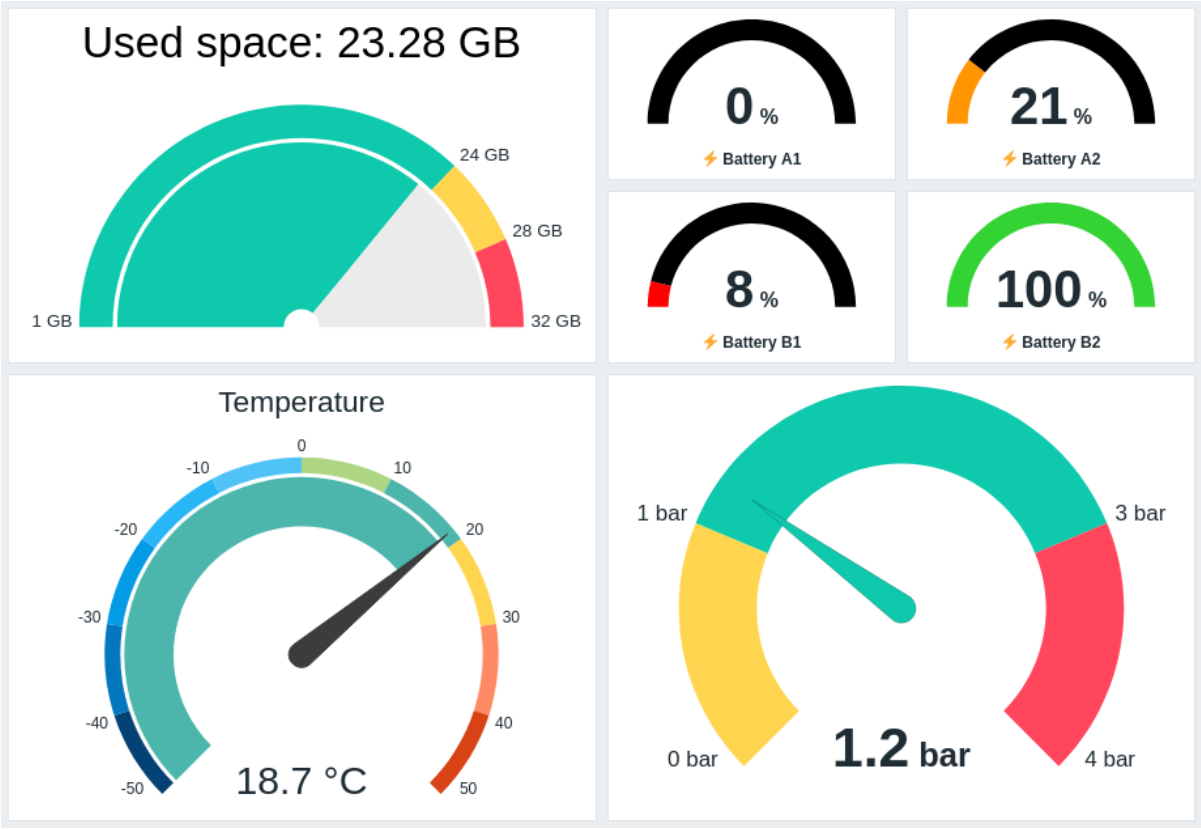
Configuração

Todos os parâmetros de configuração para este widget são comuns a todos os widgets.

6 Gauge

Visão geral

O widget *Gauge* exibe o valor de um único item numérico como um medidor. É útil para monitorar métricas importantes, visualizar limites e detectar mudanças repentinas.



Você pode configurar o widget para exibir o seguinte:

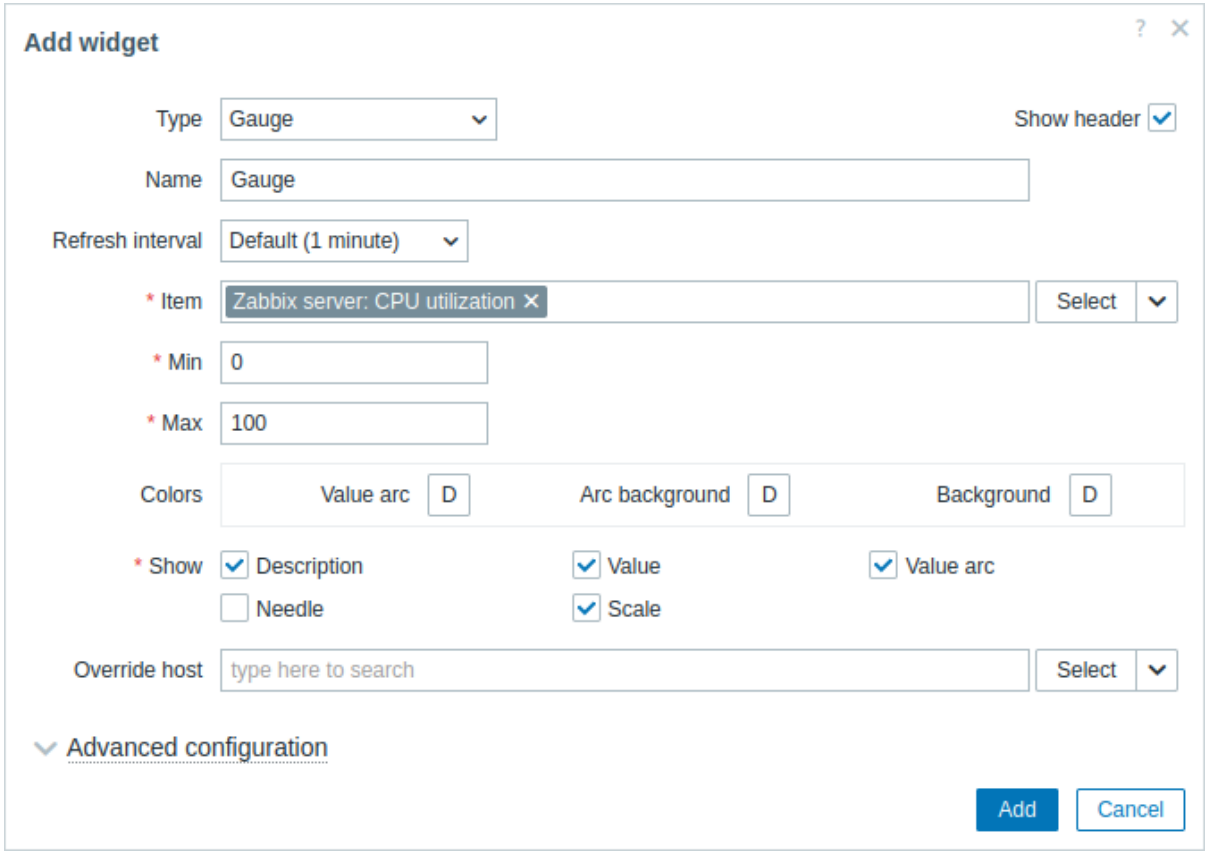
- Descrição do item (Espaço usado: {ITEM.LASTVALUE}, Temperatura)
- Valor do item e unidades (21 %, 18.7 °C, 1.2 bar)
- Escala do medidor (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Arco do medidor (arco do valor do medidor e arco dos limites do medidor)
- Ponteiro do medidor

Clicar no widget abre um gráfico simples para o item.

Você também pode exportar o widget renderizado como uma imagem .png.

Configuração

Para configurar, selecione *Gauge* como tipo:



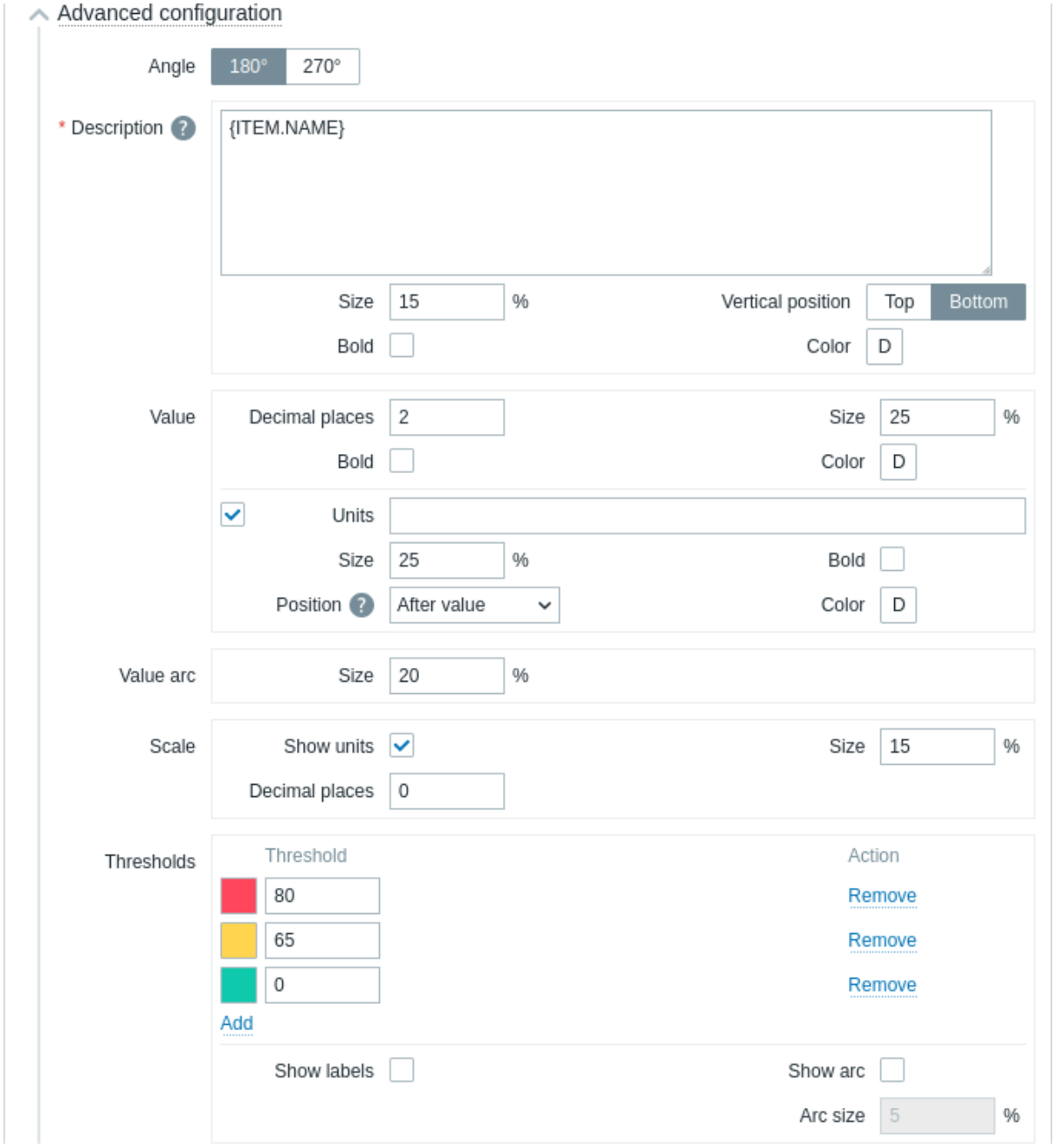
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Item	Selecione o item. Como alternativa, selecione um widget compatível como fonte de dados para itens. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um item, será exibida uma lista suspensa de itens correspondentes. Observe que você só pode selecionar itens que retornam dados numéricos (exceto dados binários).
Min	Insira o valor mínimo do gauge. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valores são suportados.
Max	Insira o valor máximo do gauge. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valores 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 seletor de host do 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, relativa 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, relativa 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, relativa 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, relativa 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, relativa 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 valor 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, relativa ao raio do arco do gauge).

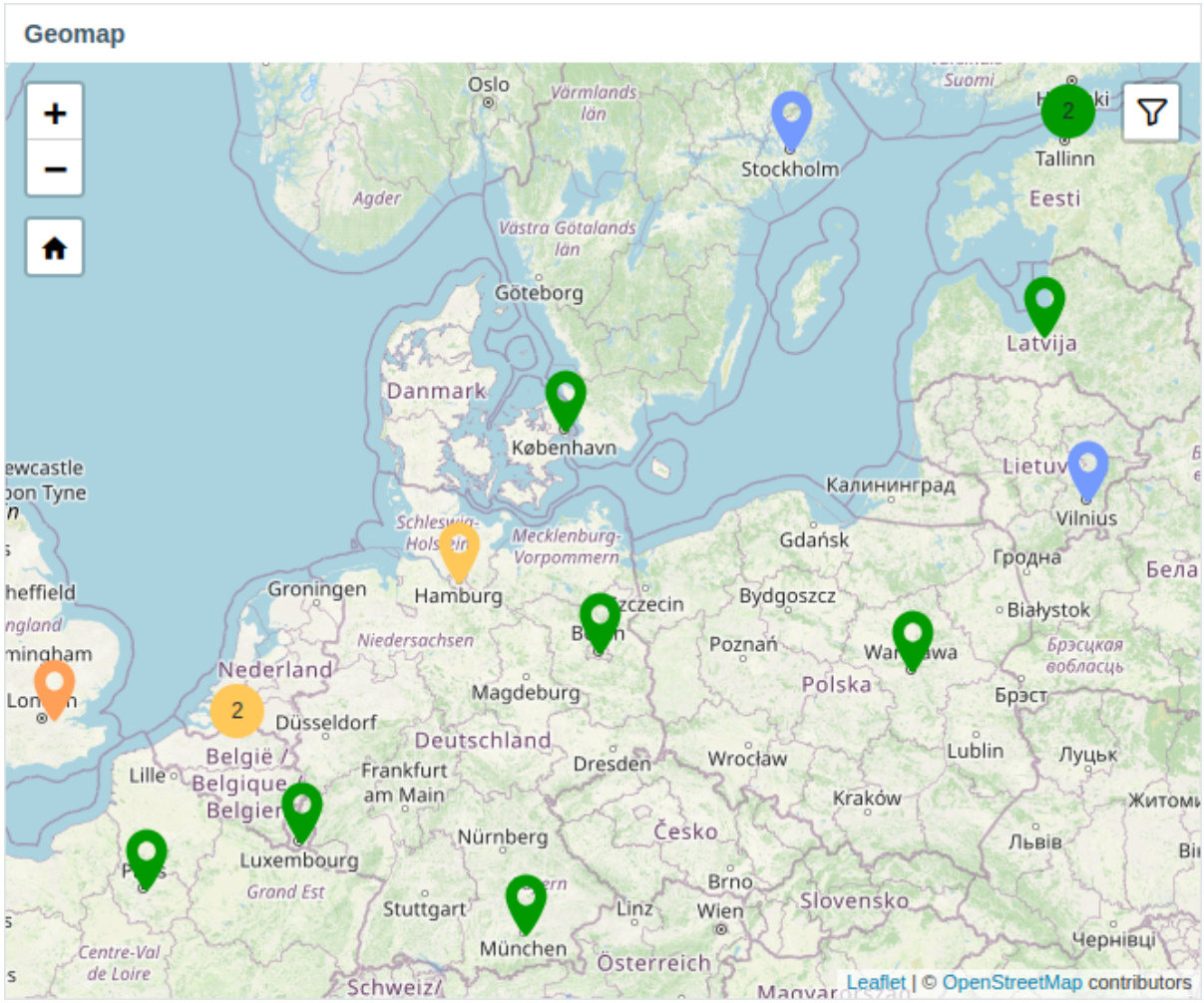
As informações exibidas pelo widget de medidor podem ser baixadas como uma imagem *.png* usando o menu do widget:

Uma captura de tela do widget será salva na pasta *Downloads*.

7 Geomapa

Visão geral

O widget *Geomap* exibe hosts como marcadores em um mapa interativo baseado em Leaflet, com cores de marcador indicando o problema mais grave de cada host.

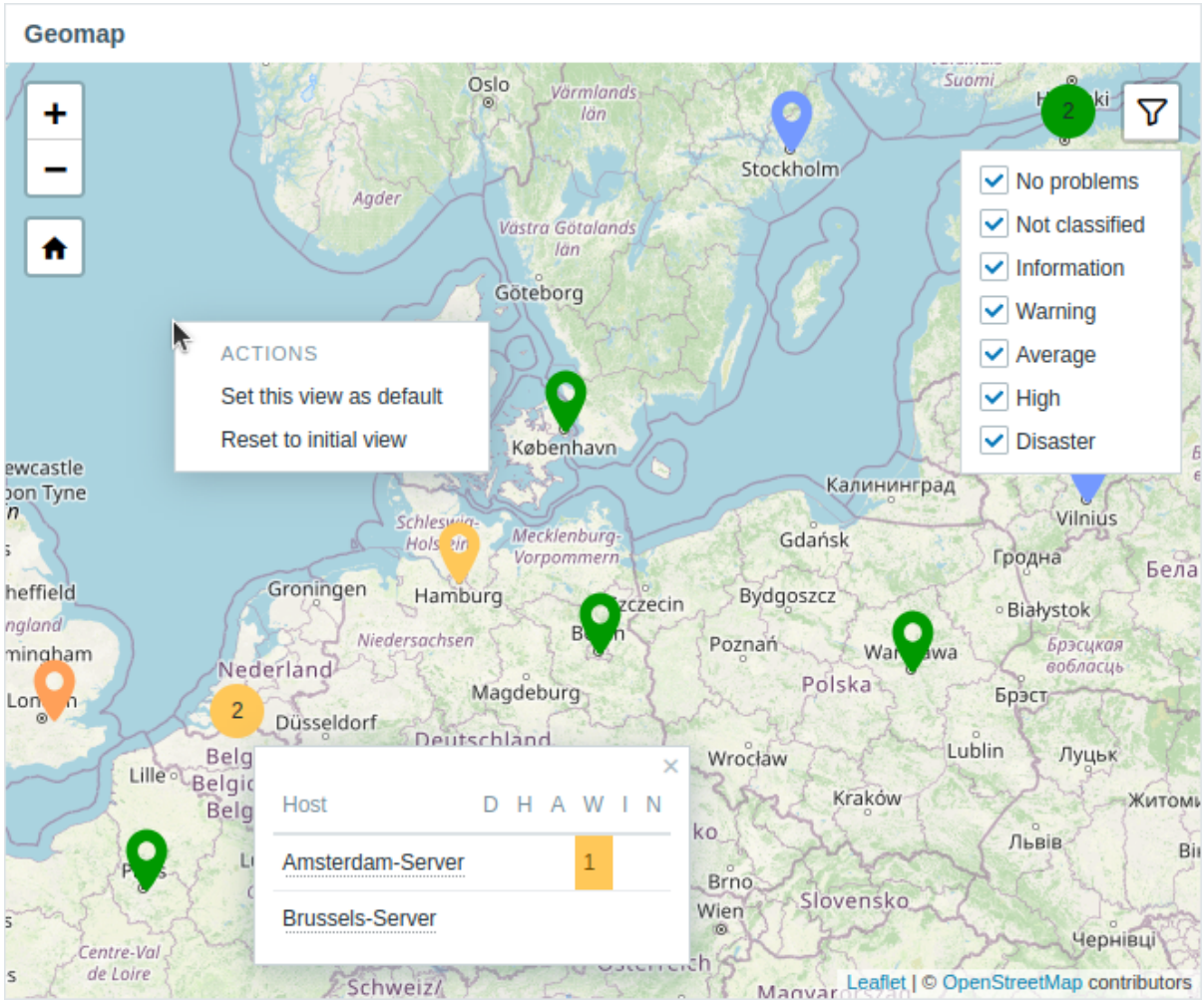


Note:
O Zabbix oferece vários provedores de mapas predefinidos e permite adicionar provedores personalizados em *Administração* > *Geral* > *Mapas geográficos*.

Por padrão, todos os hosts habilitados com coordenadas válidas (latitude -90 a 90, longitude -180 a 180) são exibidos.

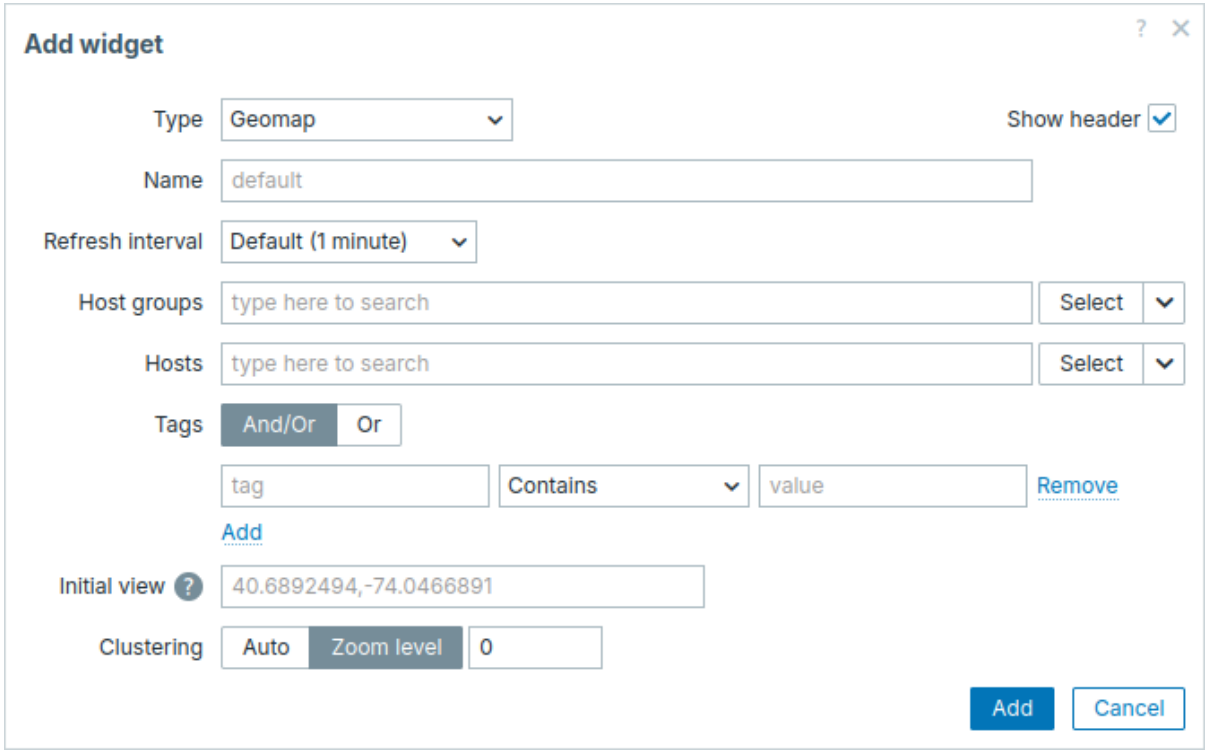
Use as seguintes opções para interagir com o widget:

- Clicar em um marcador exibe o nome do host e problemas não resolvidos; clicar no nome do host abre o menu do host.
- Filtre hosts por gravidade do problema usando o ícone de filtro no canto superior direito.
- Clique com o botão direito no mapa para definir ou redefinir a visualização padrão; use o ícone de casa para retornar à visualização padrão.



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:

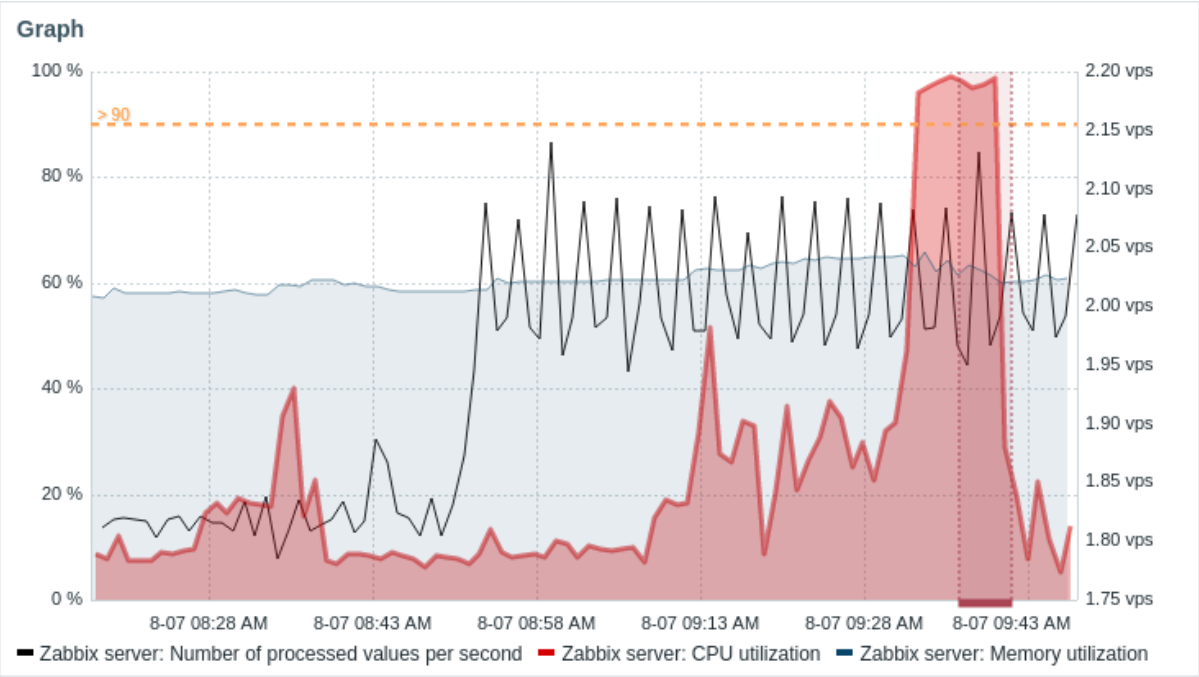
Grupos de hosts	Selecione os grupos de hosts a serem exibidos no mapa. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo possui preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. 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 de 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 onde os valores de 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 de 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.
Visualização inicial	Coordenadas centrais separadas por vírgula e um nível de zoom opcional para exibir quando o widget é 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 é calculado como metade do zoom máximo para o provedor de tiles em particular. A visualização inicial é ignorada se a visualização padrão estiver definida (veja abaixo). Exemplos: 40.6892494,-74.0466891 40.6892494,-122.0466891
Agrupamento	Especifique como os marcadores de hosts próximos são combinados em um único marcador com uma contagem: **Automático** - os marcadores são combinados automaticamente; **Nível de zoom** - os marcadores são combinados com base no nível de zoom especificado (0 - zoom máximo para fora, 30 - zoom máximo para dentro). Por exemplo, se o parâmetro for definido como 0, os marcadores não serão combinados em nenhum nível de zoom; se definido como 15, os marcadores serão combinados a partir do zoom máximo para fora até o nível de zoom 15; se definido como 30, os marcadores serão combinados em todos os níveis de zoom. Observe que o nível máximo de zoom para dentro é determinado pelo provedor de tiles do mapa.

8 Gráfico

Visão geral

O widget *Gráfico* exibe dados numéricos de item como um gráfico vetorial. Ele pode ajudar você a acompanhar tendências,

identificar problemas e comparar valores ao longo do tempo e entre hosts.

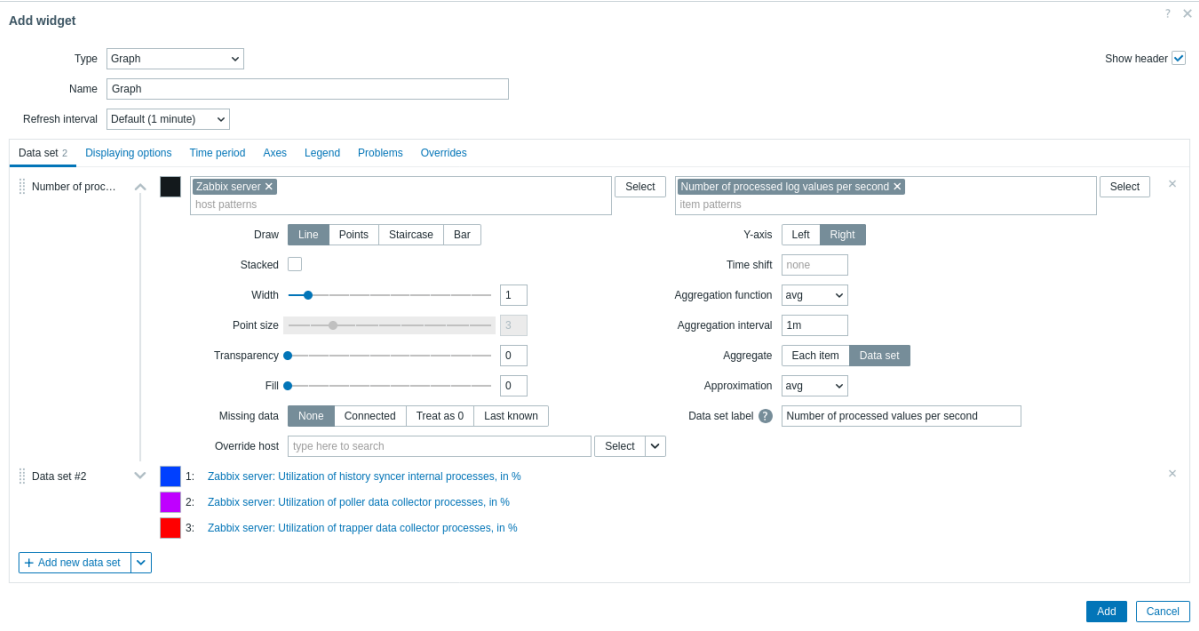


Você pode exportar o gráfico renderizado como uma imagem .png.

O widget legado Gráfico (clássico) também está disponível.

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. Você pode escolher uma única cor base ou selecionar uma linha de paleta para atribuir cores distintas a cada item correspondente.
- *Lista de itens* - os dados dos itens selecionados são exibidos. Você pode escolher a cor de cada item individualmente no seletor.

Por padrão, um conjunto de dados *Padrões de item* é adicionado.

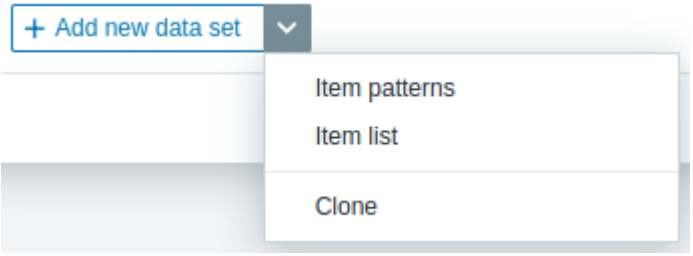
Conjunto de dados	Para o conjunto de dados **Padrões de item**: Selecione ou insira padrões de host e de item; os dados dos itens que correspondem a esses padrões serão exibidos no gráfico; até 50 itens podem ser exibidos. Padrões curinga podem ser usados para seleção (por exemplo, ∗ retornará resultados 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 itens para o gráfico clicando no botão *Adicionar item*. 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 em branco; **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 do último valor conhecido; não aplicável para os tipos de desenho *Pontos* e *Barra*.
Substituir host	Selecione um widget compatível ou o seletor de host do dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
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 de valores; **first** - exibe o primeiro valor; **last** - exibe o último valor; **none** - exibe todos os valores (sem agregação).
Intervalo de 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. 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* definida como *Tendências* ou *Auto*), recomenda-se 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.
Agregação	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 longo 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 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

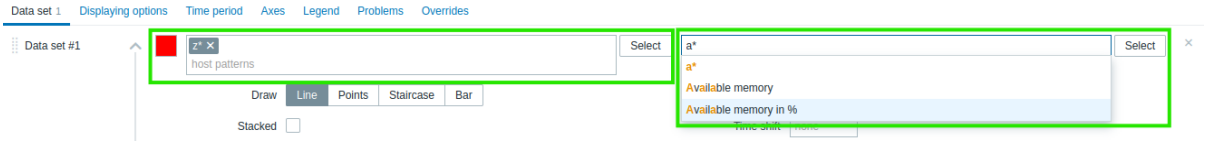
Os conjuntos de dados existentes são exibidos em uma lista. Você pode:

- 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 abrir o seletor. Você pode inserir um código hexadecimal, escolher uma amostra sólida ou alternar para a guia *Paleta* e selecionar uma linha de cores predefinidas. A cor escolhida é aplicada diretamente para conjuntos de dados do tipo *Lista de items* ou como base para tons gerados em *Padrões de items*. Use Tab para mover entre os controles do diálogo, as teclas de seta para navegar entre as amostras ou linhas da paleta, Enter para selecionar e Esc para cancelar.
- 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 do tipo *Padrões de items* ou *Lista de items* ou permitindo *Clonar* o conjunto de dados atualmente aberto. Se todos os conjuntos de dados estiverem recolhidos, a opção *Clonar* não estará disponível.

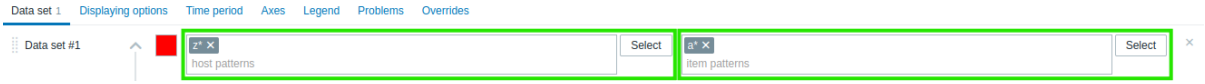


O conjunto de dados **Padrões de item** contém os campos *Padrões de host* e *Padrões de item* que reconhecem nomes completos ou padrões contendo um símbolo curinga (*). Essa funcionalidade permite selecionar todos os nomes de host e nomes de item que contenham o padrão selecionado. Ao digitar o nome 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.

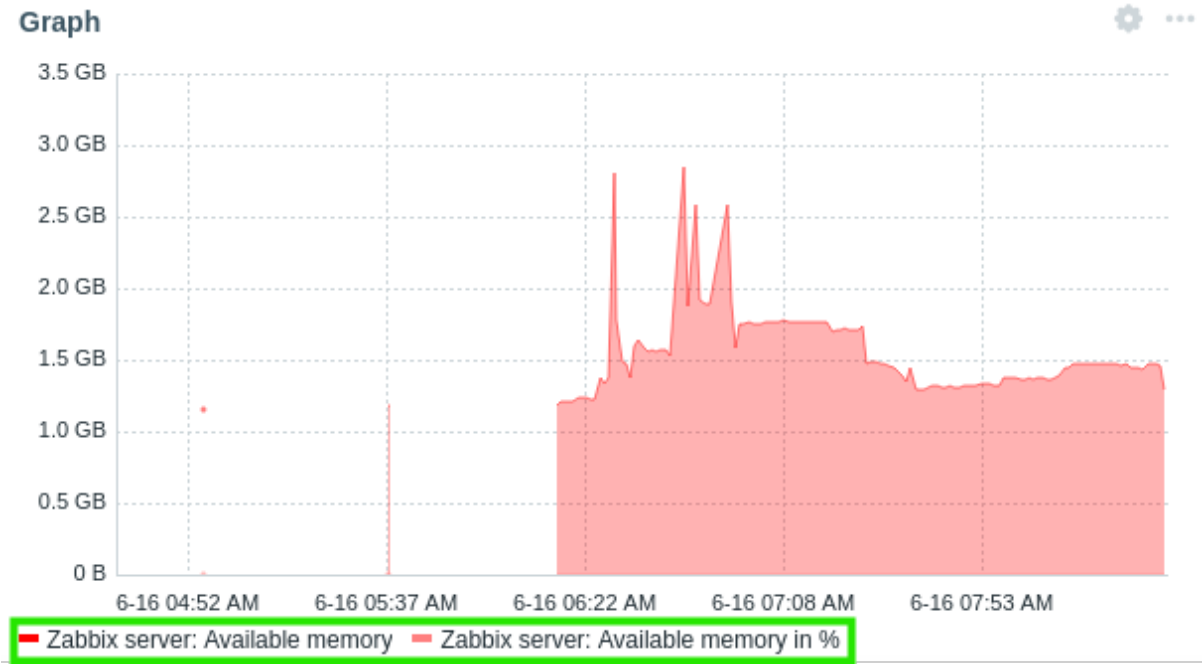
Por exemplo, ao digitar o padrão **z*** no campo *Padrões de host*, a lista suspensa exibe todos os nomes de host que contêm esse padrão: **z*, Zabbix server, Zabbix proxy**. Após pressionar *Enter*, esse padrão é aceito e exibido como **z***. Da mesma forma, ao digitar o padrão **a*** no campo *Padrões de item*, a lista suspensa exibe todos os nomes de item que contêm esse padrão: **a*, Available memory, Available memory in %**.



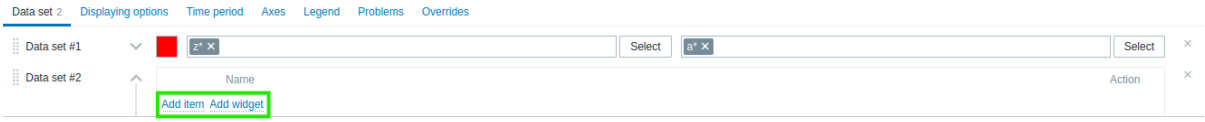
Após pressionar Enter, o padrão é aceito e exibido como a*.



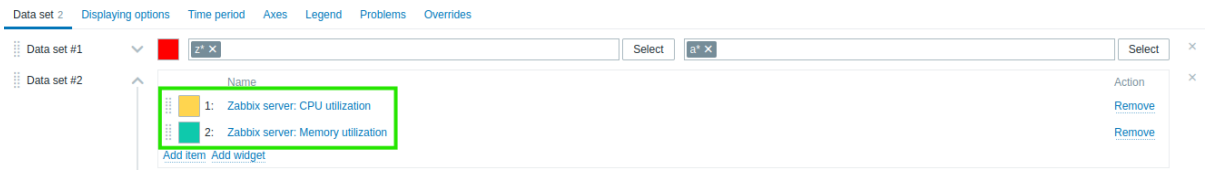
O gráfico então exibe todos os itens pertencentes ao(s) nome(s) de host selecionado(s).



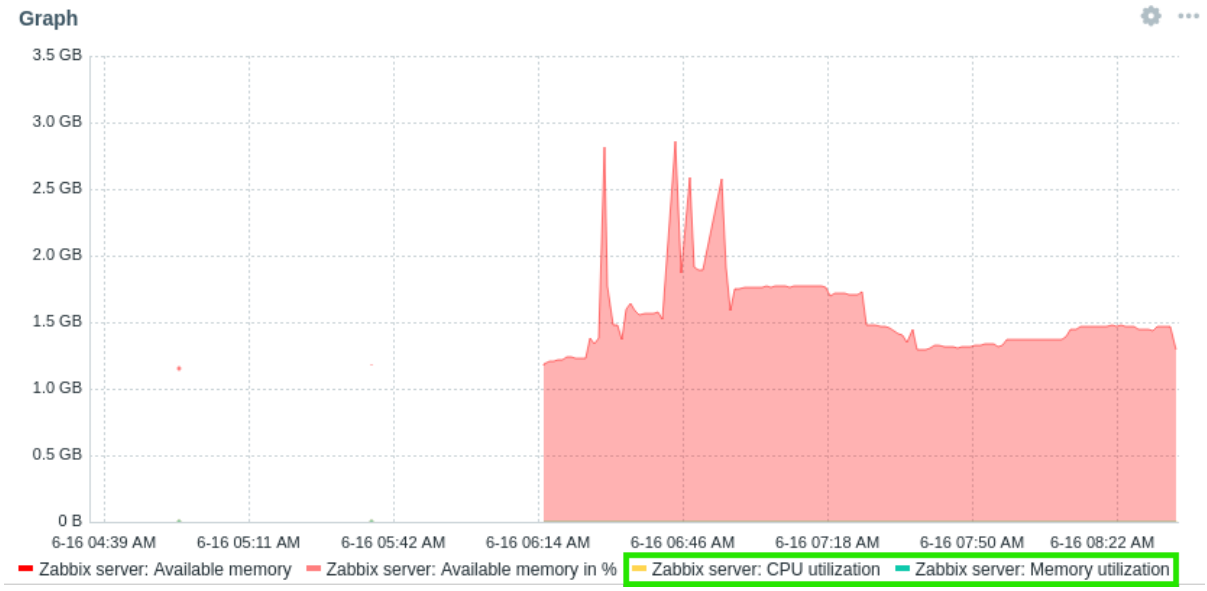
O conjunto de dados **Lista de items** contém o botão *Adicionar item* que permite adicionar items para serem exibidos no gráfico. Você também pode adicionar widgets compatíveis como fonte de dados para items clicando no botão *Adicionar widget*.



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.

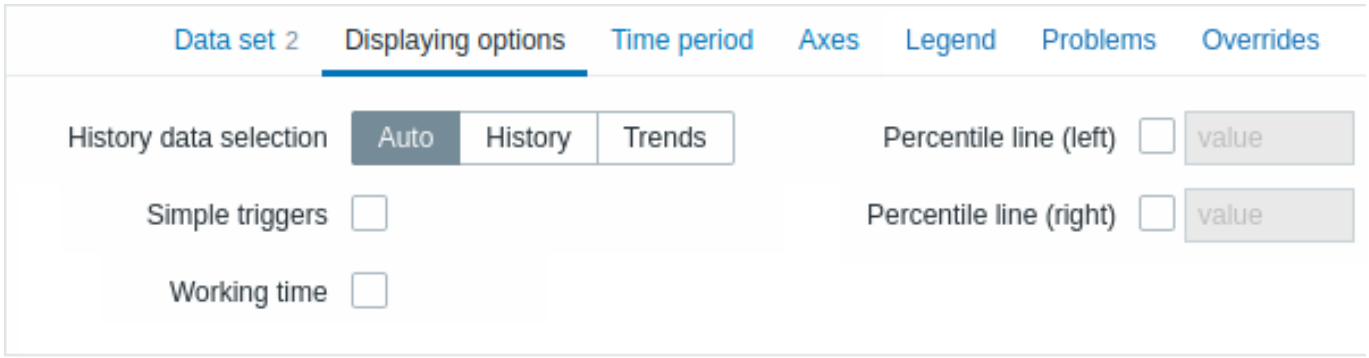


Após selecionar um ou mais items, eles serão exibidos na lista de items do conjunto de dados e no gráfico.



Opções de exibição

A aba **Opções de exibição** permite definir a seleção de dados do histórico:

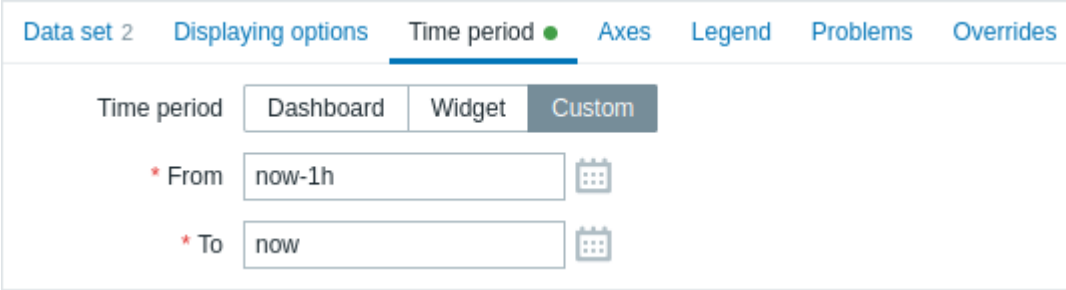


Seleção de dados do histórico	Defina a fonte dos dados do gráfico: **Auto** - os dados são obtidos de acordo com o algoritmo de gráfico clássico (padrão); **Histórico** - dados do histórico; **Tendências** - dados das tendências.
Triggers simples	Marque a caixa de seleção para mostrar os limiares de 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.
Horário de trabalho	Marque a caixa de seleção para mostrar o horário de trabalho no gráfico. O horário de trabalho (dias úteis) é exibido nos gráficos como fundo branco, enquanto o horário não útil é exibido em cinza (com o tema padrão *Original blue* do frontend).
Linha de percentil (esquerda)	Marque a caixa de seleção 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 de seleção 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

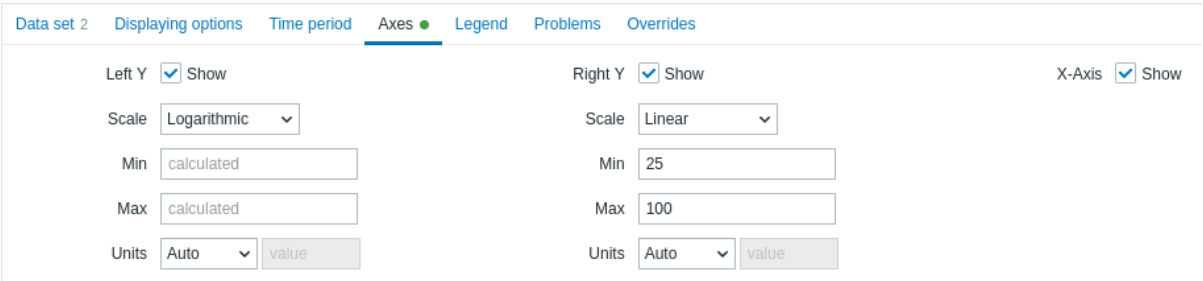
A aba **Período de tempo** permite definir um período de tempo para exibir dados no gráfico:



Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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 fim 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".

Eixos

A aba **Eixos** permite personalizar como os eixos são exibidos:

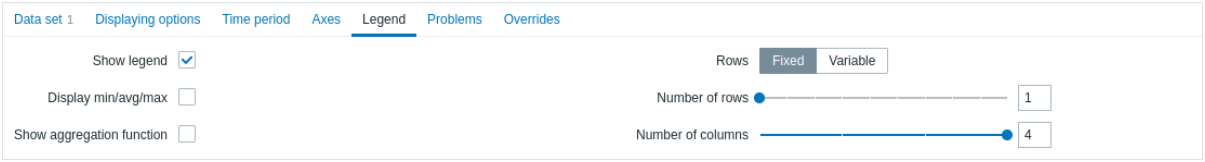


Y Esquerdo	Marque esta caixa de seleção para tornar o eixo Y esquerdo visível. A caixa de seleção pode estar desabilitada se não estiver selecionada na aba *Conjunto de dados* ou *Substituições*.
Y Direito	Marque esta caixa de seleção para tornar o eixo Y direito visível. A caixa de seleção pode estar desabilitada se não estiver selecionada na aba *Conjunto de dados* ou *Substituições*.
Eixo X	Desmarque esta caixa de seleção para ocultar o eixo X (marcada por padrão).

Escala	Escolha a escala para os valores do eixo do gráfico no menu suspenso: **Linear** - os valores do eixo aumentam por uma quantidade fixa (por exemplo, 10, 20, 30), adequado para dados que mudam de forma constante ou cobrem uma faixa pequena a moderada; **Logarítmica** - os valores do eixo aumentam exponencialmente (por exemplo, 10, 100, 1000), adequado para dados que mudam rapidamente ou cobrem uma faixa grande.
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: **Auto** - os valores do eixo são exibidos usando a unidade do primeiro item no conjunto de dados; **Estático** - os valores do eixo são exibidos usando a unidade especificada no campo de entrada *valor*; se o campo for deixado em branco, apenas valores numéricos serão exibidos.

Legenda

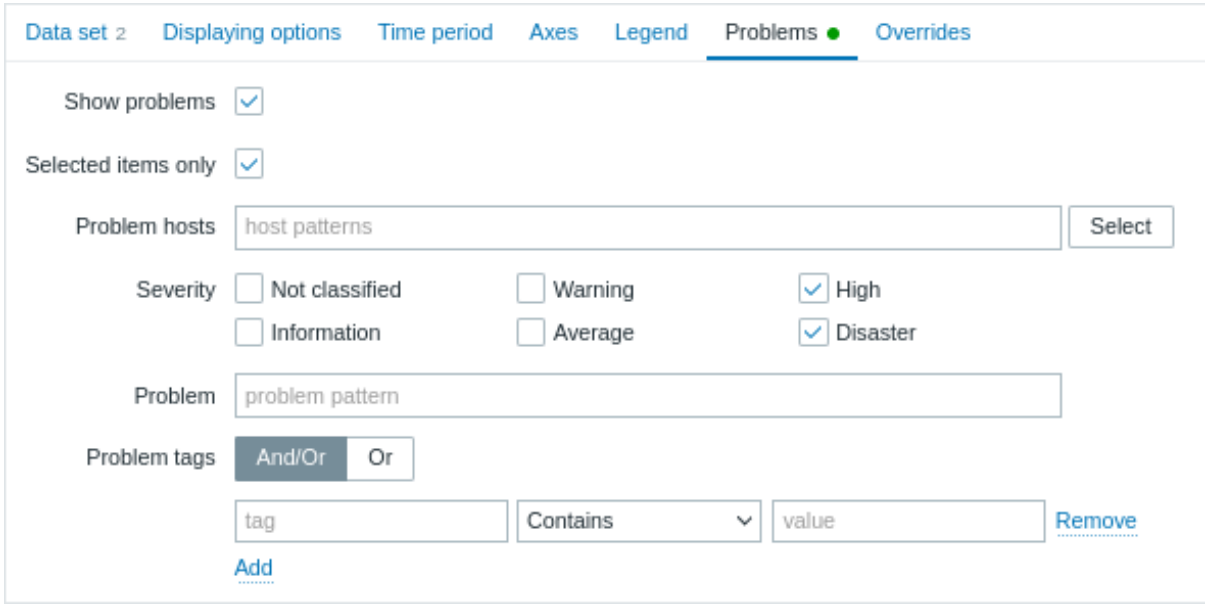
A aba **Legenda** permite personalizar a legenda do gráfico:



Exibir legenda	Desmarque esta caixa de seleção para ocultar a legenda no gráfico (marcada por padrão).
Exibir min/méd/máx	Marque esta caixa de seleção para exibir os valores mínimo, médio e máximo do item na legenda.
Exibir função de agregação	Marque esta caixa de seleção para exibir 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 *Exibir min/méd/máx* estiver desmarcado.

Problemas

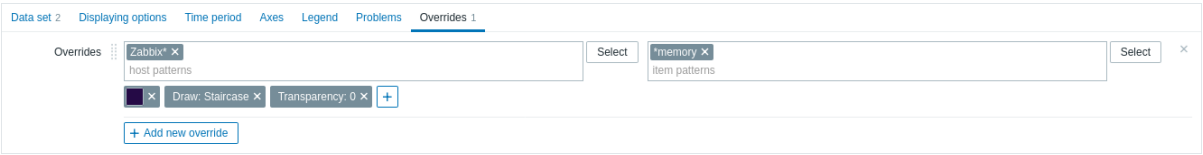
A aba **Problemas** permite personalizar a exibição de problemas:



Exibir problemas	Marque esta caixa para habilitar a exibição de problemas no gráfico (desmarcada, ou seja, desabilitada por padrão).
Apenas itens selecionados	Marque esta caixa 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. 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.

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).

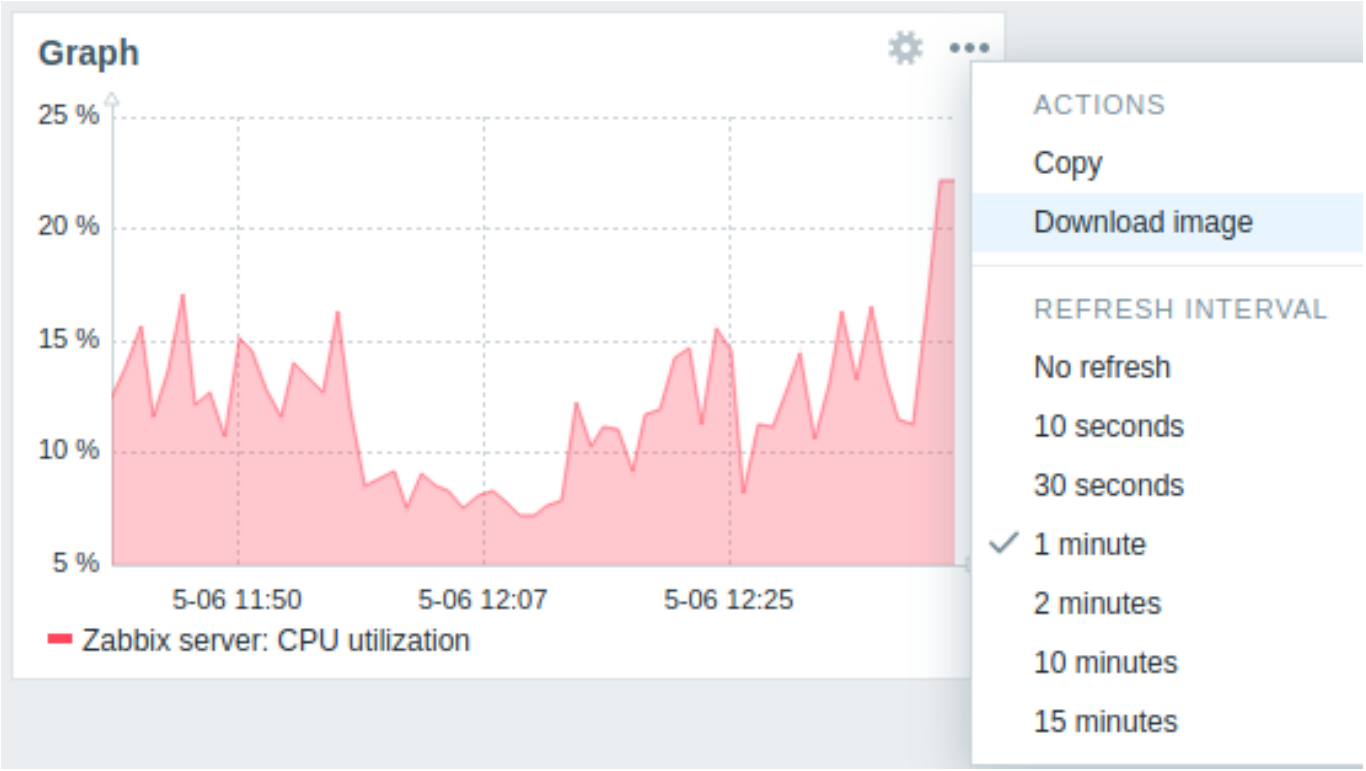
As substituições existentes (se houver) são exibidas em uma lista. Para adicionar uma nova substituição:

- Clique no botão 

- Selecione hosts e items para a substituição. Alternativamente, você pode inserir padrões de host e item. 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. 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.

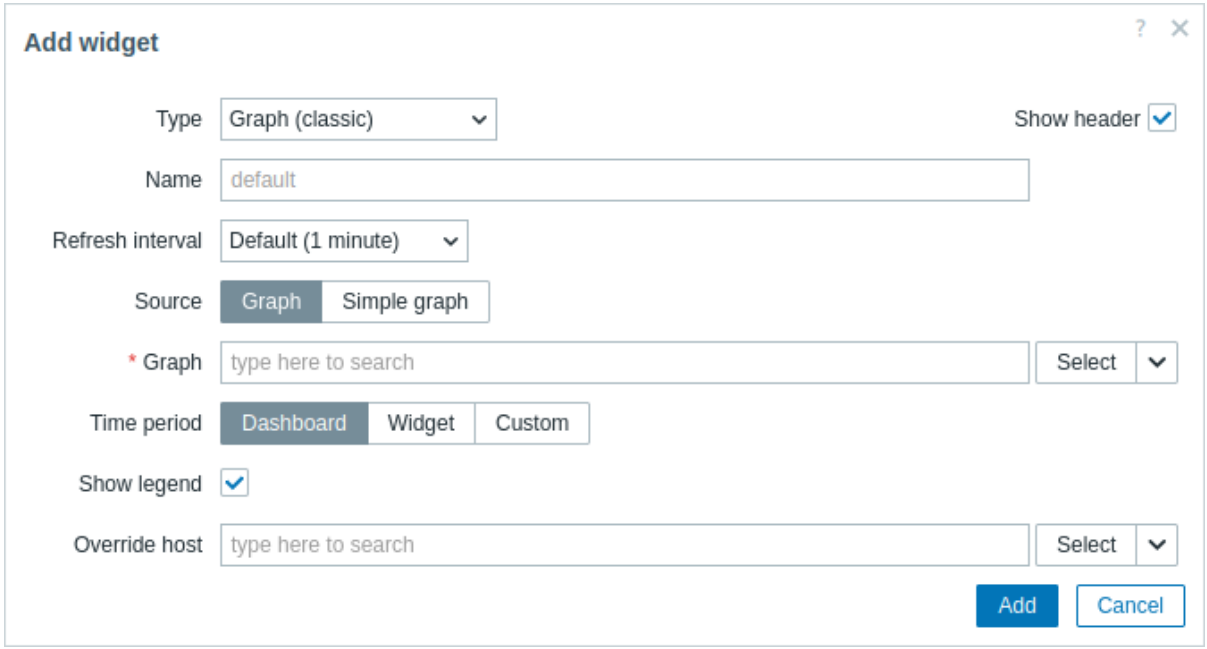
9 Gráfico (clássico)

Visão geral

O widget *Gráfico (clássico)* exibe dados numéricos de item como um gráfico personalizado ou gráfico simples baseado em imagem. Ele suporta seleção de período de tempo, linhas de trigger e exibição opcional de legenda.

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:

Origem	Selecione o tipo de gráfico: **Gráfico** - gráfico personalizado; **Gráfico simples** - gráfico simples.
Gráfico	Selecione o gráfico personalizado a ser exibido. Alternativamente, selecione um widget compatível como fonte de dados para gráficos. Este parâmetro está disponível se *Origem* estiver definido como ”Gráfico”.
Item	Selecione o item a ser exibido em um gráfico simples. Alternativamente, selecione um widget compatível como fonte de dados para itens. Este parâmetro está disponível se *Origem* 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** - use o seletor de período de tempo do dashboard; **Widget** - use um widget compatível (definido no parâmetro *Widget*); **Personalizado** - use um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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”.
Exibir legenda	Desmarque esta caixa para ocultar a legenda no gráfico (marcada por padrão).
Substituir host	Selecione um widget compatível ou o seletor de host do dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

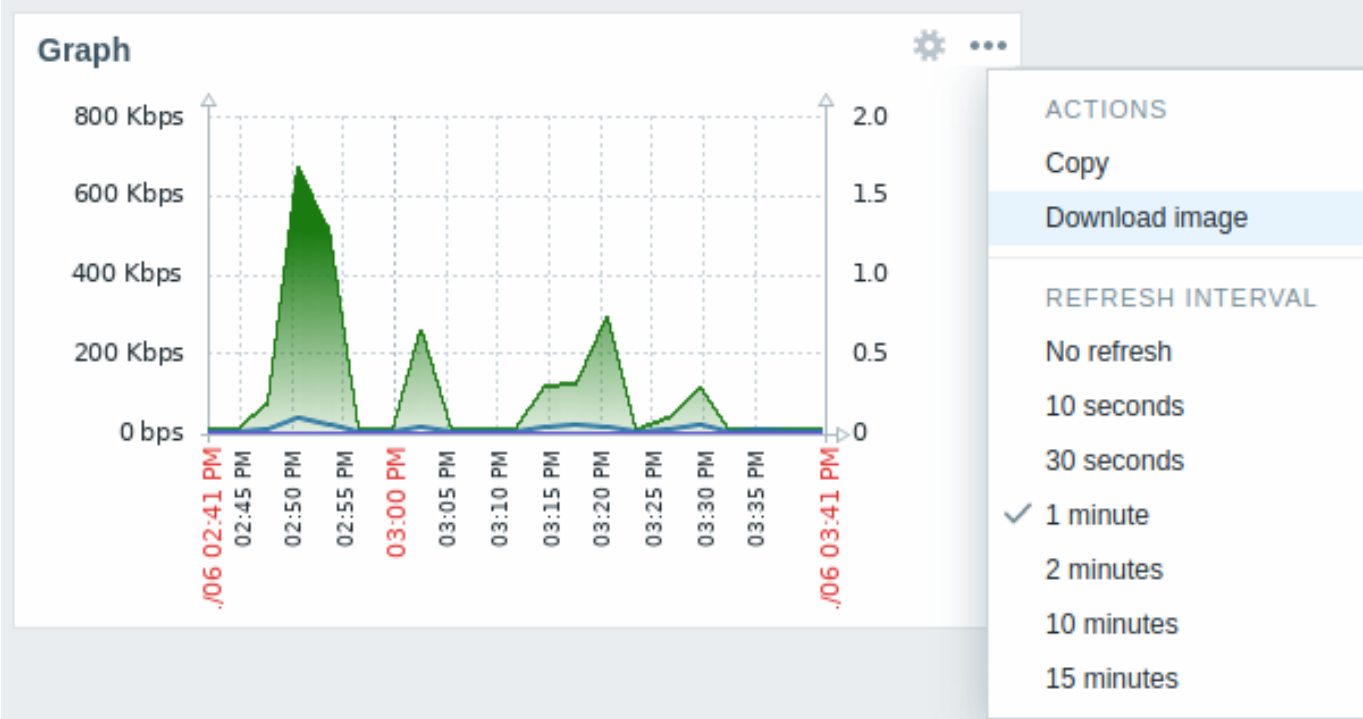
A legenda do gráfico consiste em três seções:

- Itens e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver triggers associadas 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:

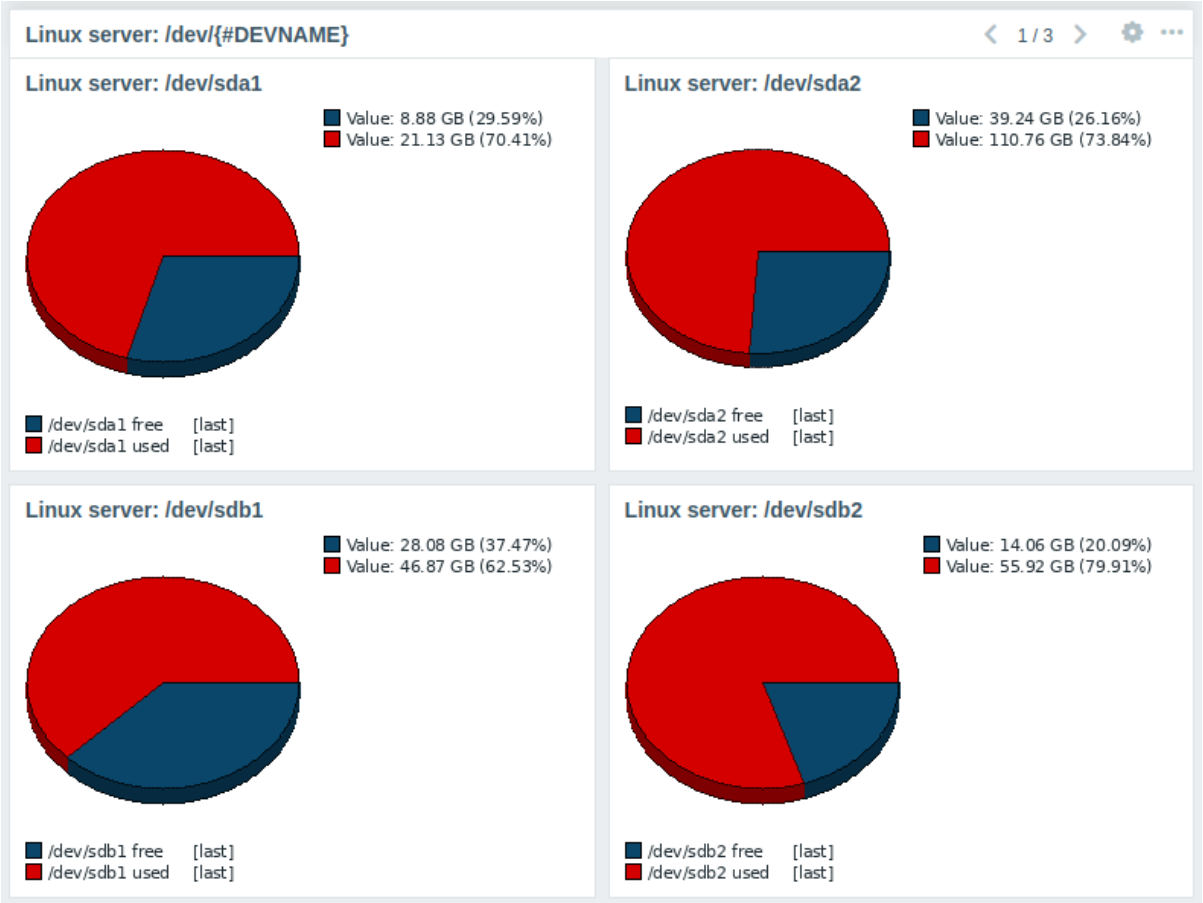


Uma captura de tela do widget será salva na pasta *Downloads*.

10 Protótipo de gráfico

Visão geral

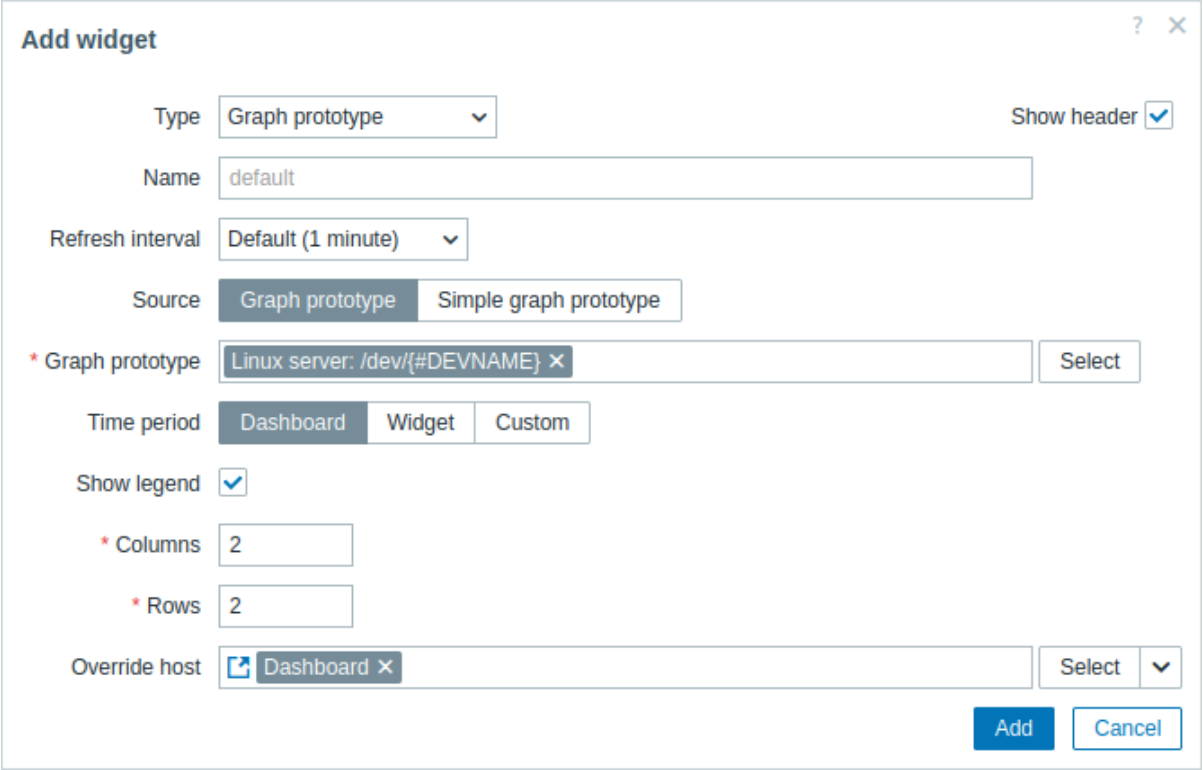
O widget *Protótipo de gráfico* exibe uma grade de gráficos personalizados criados automaticamente com base em protótipos de gráficos de regras de descoberta de baixo nível prototipos de gráficos ou prototipos de item.



Os gráficos são organizados em um layout de grade configurável; se o número de gráficos exceder o layout, a paginação é ativada no widget.

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:

Fonte	Selecione a fonte 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 *Fonte* 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 *Fonte* 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** - use o seletor de período de tempo do dashboard; **Widget** - use um widget compatível (definido no parâmetro *Widget*); **Personalizado** - use um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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".
Mostrar legenda	Desmarque esta caixa de seleção para ocultar a legenda nos gráficos (marcada por padrão).
Substituir host	Selecione um widget compatível ou o seletor de host do 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.

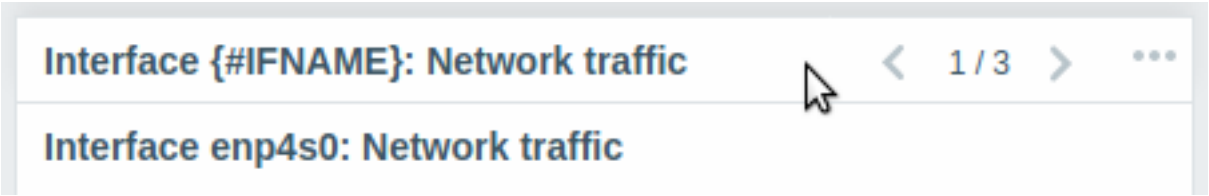
A legenda do gráfico consiste em três seções:

- Itens e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver triggers associadas 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.

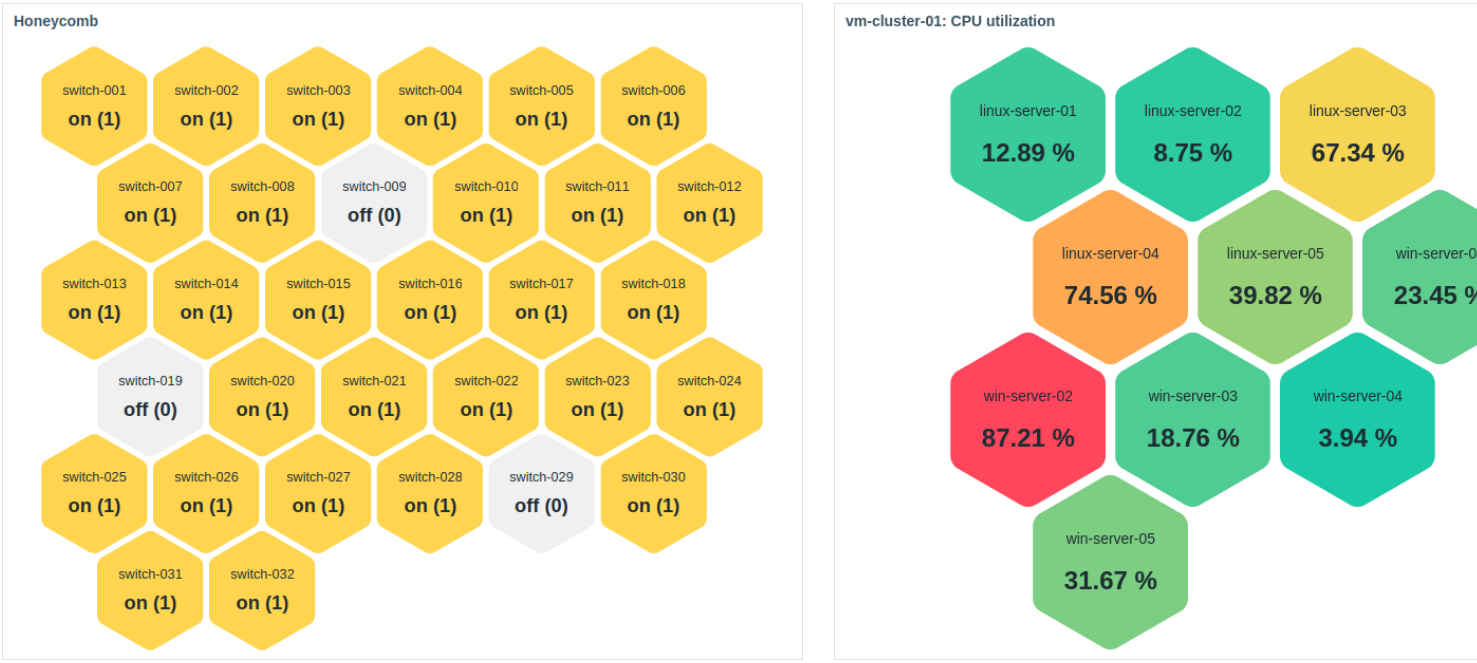
Embora os parâmetros *Colunas* e *Linhas* permitam ajustar mais de um gráfico no widget, ainda pode haver mais gráficos descobertos do que colunas/linhas no widget. Neste caso, a paginação fica disponível no widget, e um cabeçalho deslizante permite alternar entre as páginas usando as setas esquerda e direita:



11 Honeycomb

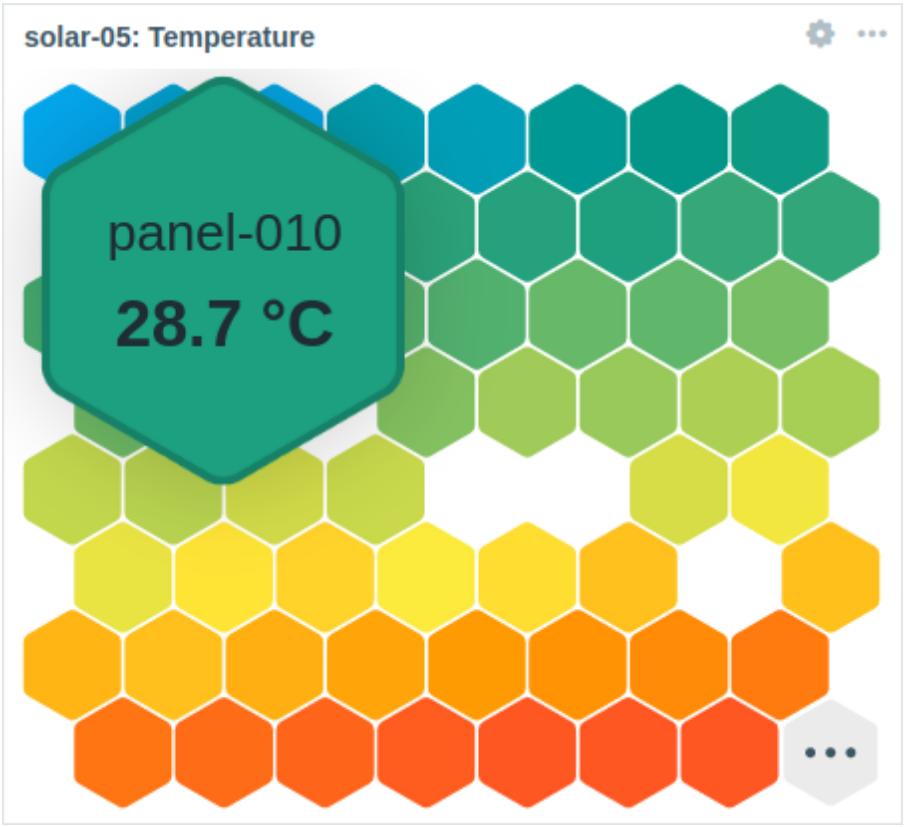
Visão geral

O widget *Honeycomb* exibe hosts ou items como células hexagonais em uma grade. Esse layout facilita a detecção de hosts/items com problemas, identificação de clusters de problemas, comparação de grupos rapidamente e muito mais.



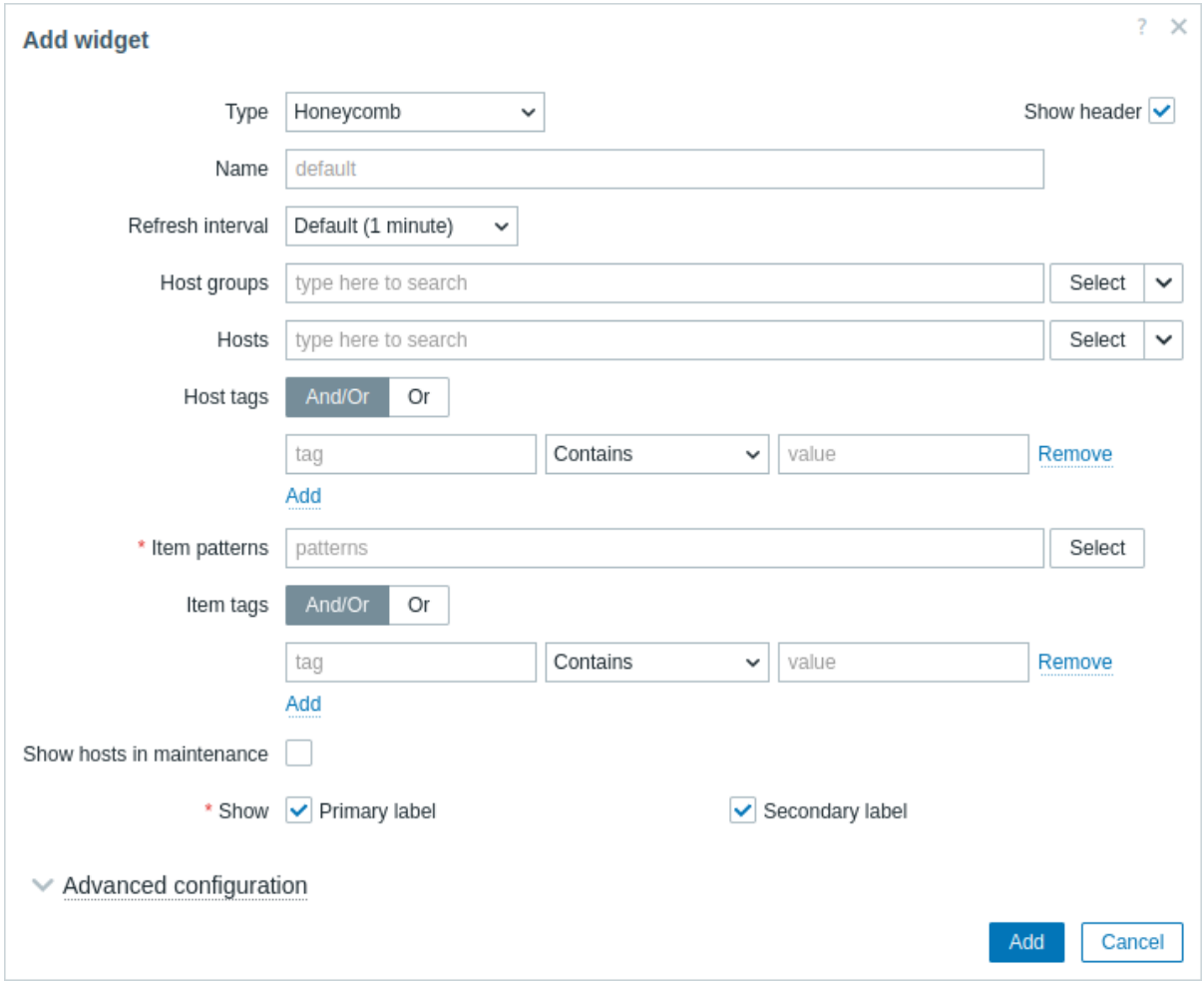
O número de células exibidas é limitado pelo tamanho do widget e pelo tamanho mínimo da célula (32px). Se todas as células não couberem, a última célula exibirá uma reticência. O widget pode ser redimensionado, com a contagem, o tamanho e o posicionamento das células ajustando-se dinamicamente.

Ao passar o mouse, a célula em foco é ampliada para melhor visibilidade. Clicar em uma célula destaca sua borda até que outra célula seja selecionada.



Configuração

Para configurar, selecione *Honeycomb* como tipo:



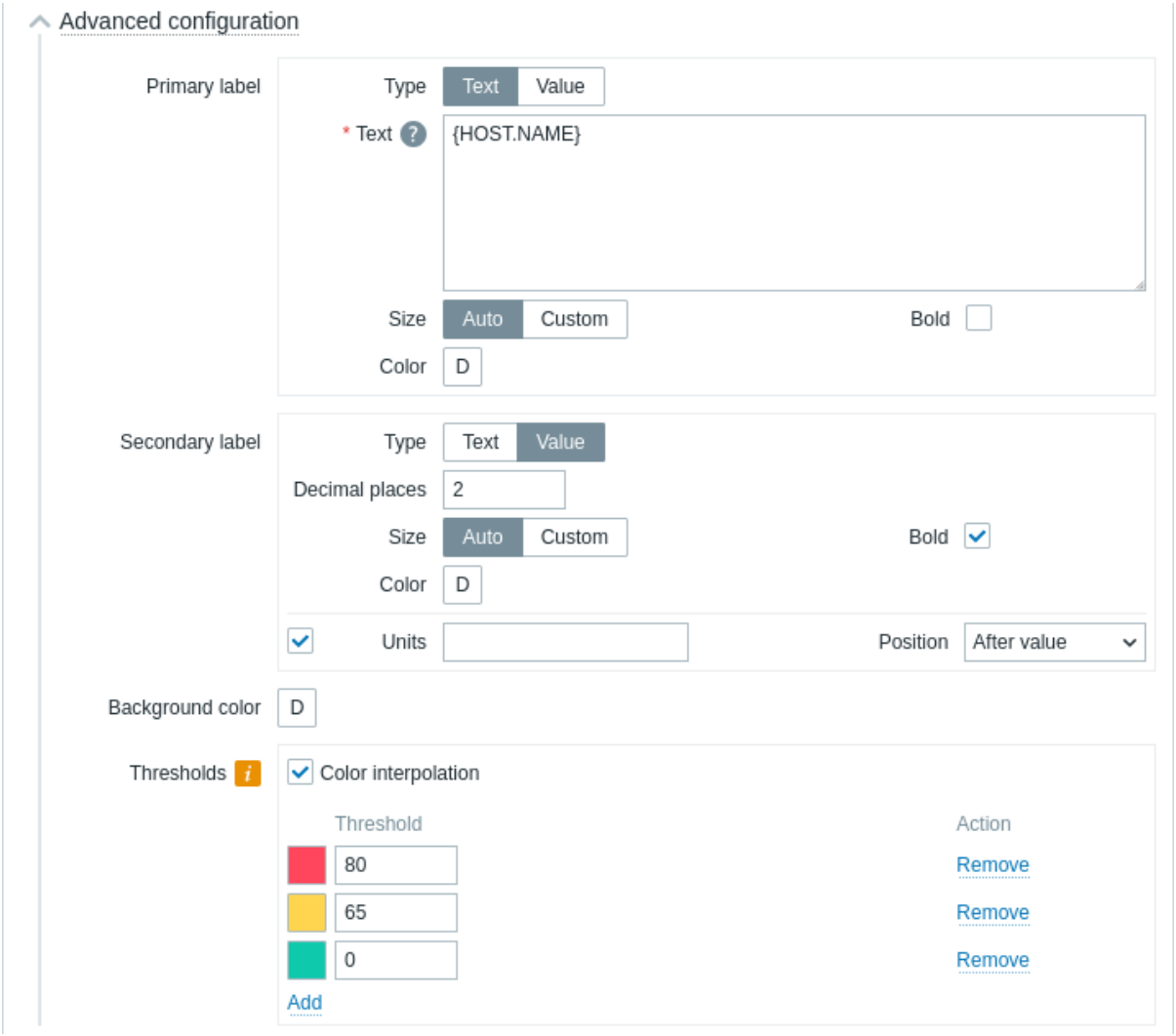
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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 com os 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 com os 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. 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 de tag 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 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. 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 curinga 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 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 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 *Exibir* (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 - usa o tamanho do 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 habilitar 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*.

12 Disponibilidade do host

Visão geral

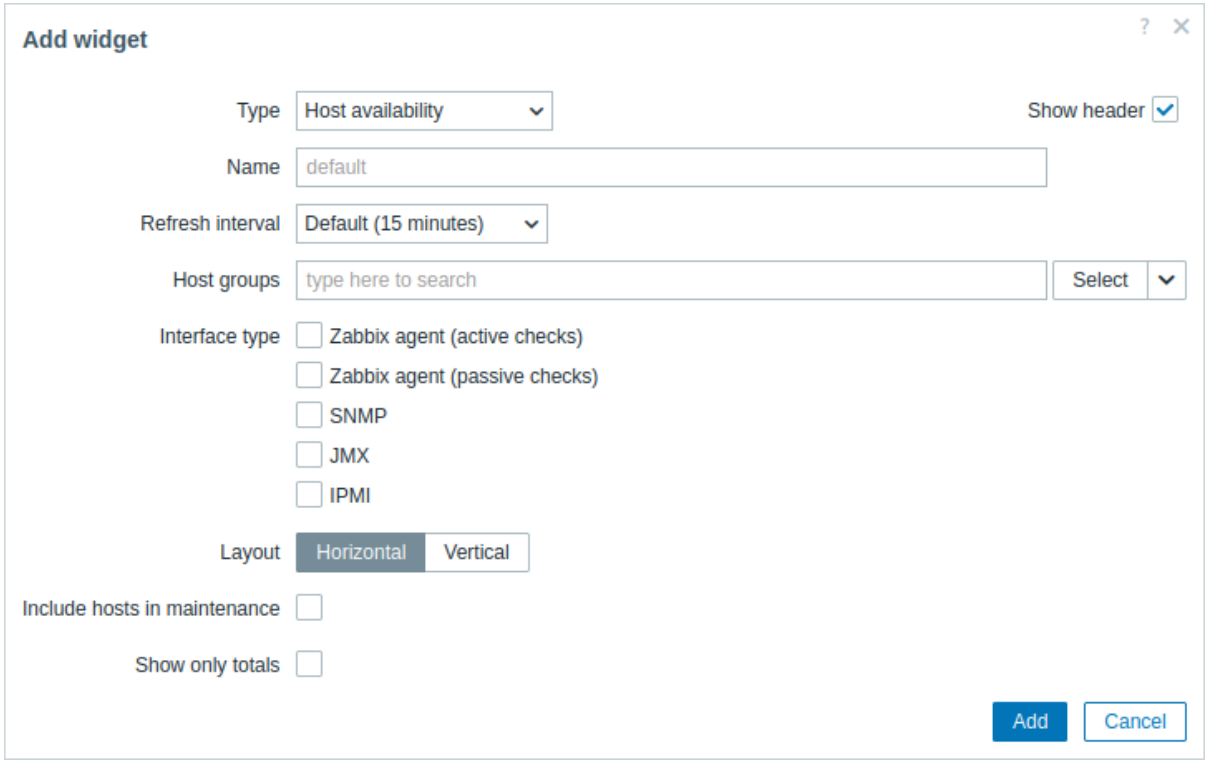
O widget *Disponibilidade do host* exibe a disponibilidade do host em grupos de hosts selecionados, permitindo monitorar o número de hosts que estão ativos ou inativos.

A disponibilidade do host 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
- *Desconhecido* - hosts com pelo menos uma interface desconhecida, mas nenhuma indisponível
- *Total* - total de todos os hosts

Configuração

Para configurar, selecione *Disponibilidade do host* como tipo:



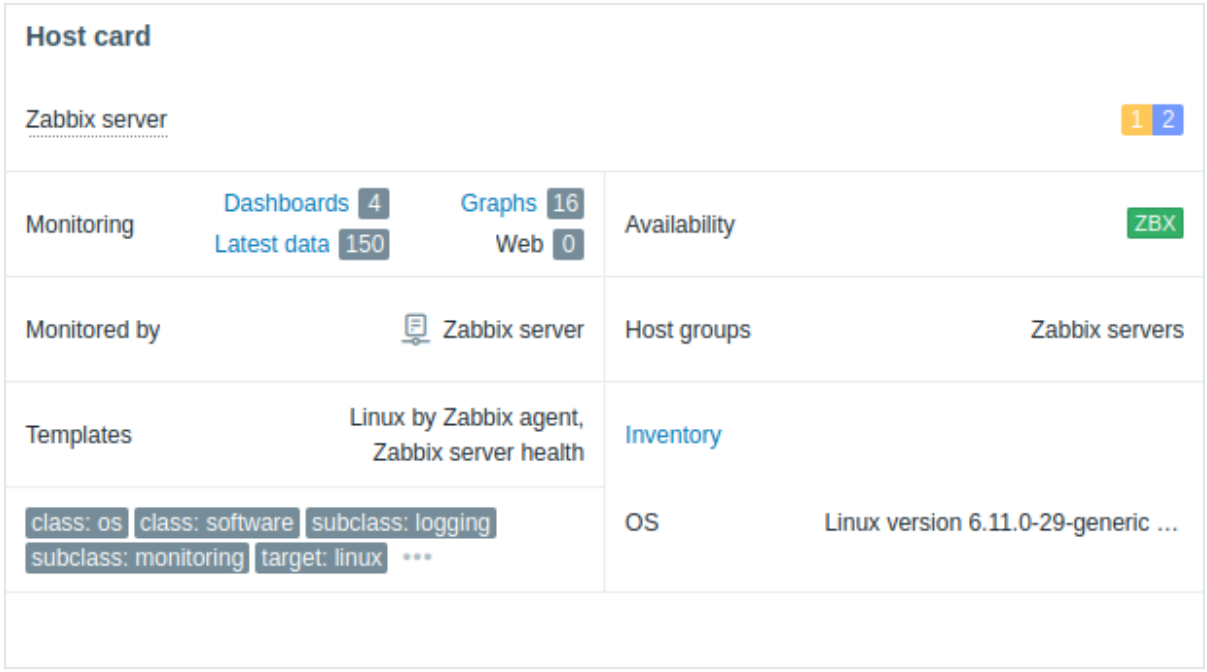
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, 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 — *Zabbix agent (active checks)*, *Zabbix agent (passive checks)*, *SNMP*, *JMX*, *IPMI*. Para *Zabbix agent (active checks)*, a célula *Misto* estará sempre vazia, pois esse tipo de item não pode ter várias interfaces. 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, será exibido apenas o total de hosts, sem detalhamento por interfaces. Esta opção é desabilitada se apenas uma interface for selecionada.

13 Cartão de host

Visão geral

O widget *Host card* exibe detalhes de um único host, permitindo que você avalie rapidamente o status e a configuração de um host (disponibilidade, problemas, inventário, etc.).

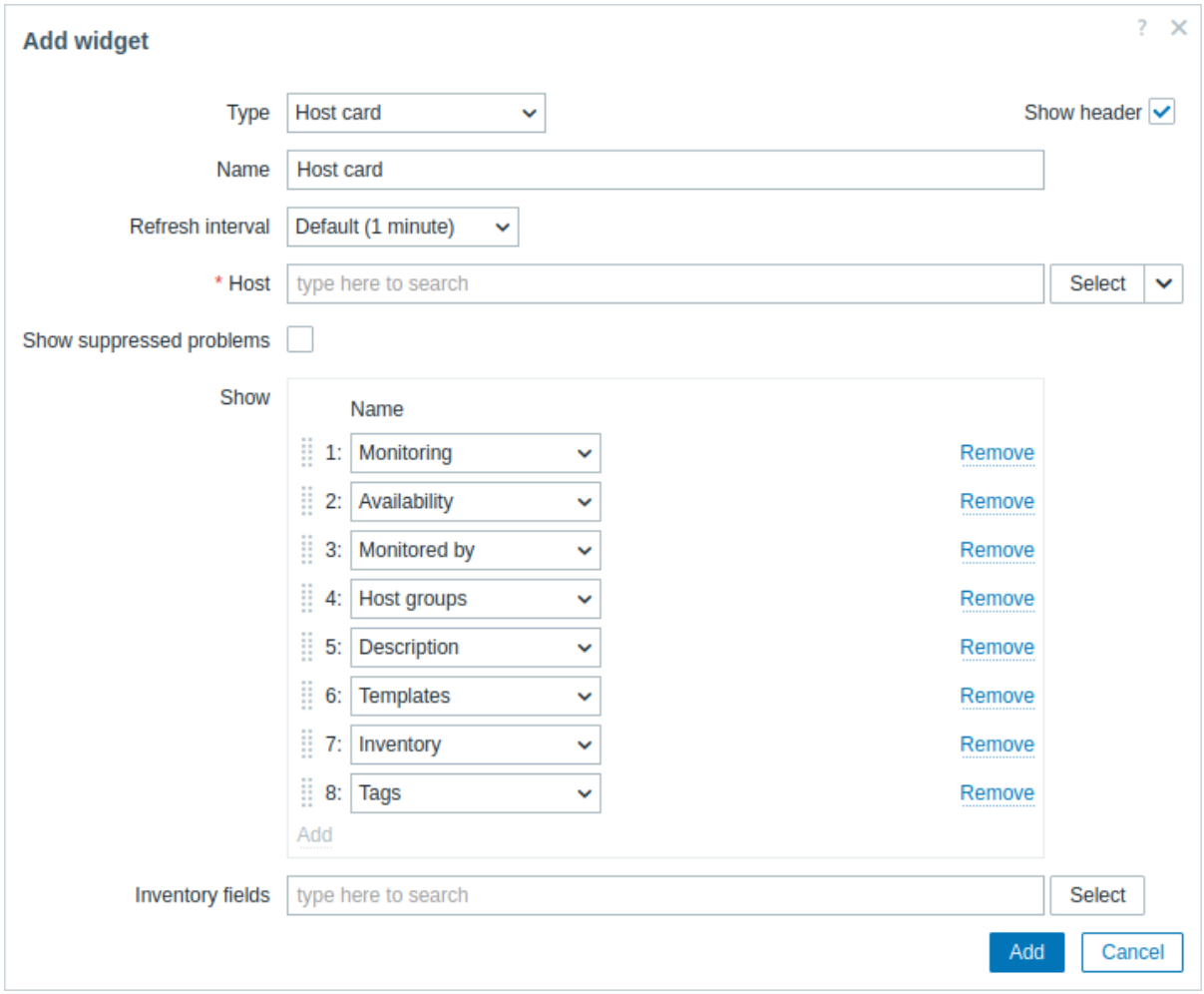


O host pode ser definido diretamente na configuração do widget ou selecionado a partir de um widget compatível ou do *Host selector* no dashboard.

O widget pode exibir várias seções, cada uma mostrando informações diferentes. O layout das seções é ajustado automaticamente com base na largura do widget; quando expandido horizontalmente, as seções são automaticamente reorganizadas em várias colunas.

Configuração

Para configurar, selecione *Host card* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir os seguintes parâmetros específicos:

Host	Selecione o host. Alternativamente, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Sugestões correspondentes aparecem enquanto você digita. A seção superior do widget sempre exibe o **nome do host** (clicando nele abre o menu do host): - Se o host estiver desabilitado, um rótulo vermelho *Desabilitado* é exibido ao lado do nome do host. - Se o host estiver em manutenção, o ícone de manutenção é exibido ao lado do nome do host. A seção superior do widget também exibe o número de **problemas do host** agrupados por severidade (clicando neles abre os problemas do host). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Mostrar problemas suprimidos	Marque esta caixa para incluir problemas suprimidos no cálculo de problemas.
Mostrar	Adicione seções para exibir no widget. As seções podem ser reorganizadas arrastando e soltando.
Grupos de hosts	Exibe os grupos de hosts aos quais o host pertence.
Descrição	Exibe a descrição do host.
Monitoramento	Exibe links rápidos para navegar até os Dashboards, Últimos dados, Gráficos e Cenários web do host. O número ao lado de cada link indica o número de entidades respectivas.
Disponibilidade	Exibe a disponibilidade do host por interface.
Monitorado por	Exibe se o host é monitorado pelo Zabbix server ou por um proxy ou grupo de proxies específico.
Templates	Exibe os templates vinculados ao host.
Inventário	Exibe os campos de inventário do host.
Tags	Exibe as tags do host.
Campos de inventário	Selecione os campos de inventário a serem exibidos. Sugestões correspondentes aparecem enquanto você digita. Se nenhum campo de inventário for especificado, todos os campos de inventário preenchidos serão exibidos. Este parâmetro está disponível se *Mostrar* contiver *Inventário*.

14 Navegador de hosts

Visão geral

O widget *Navegador de hosts* exibe uma lista de hosts com base em várias opções de filtragem e agrupamento.

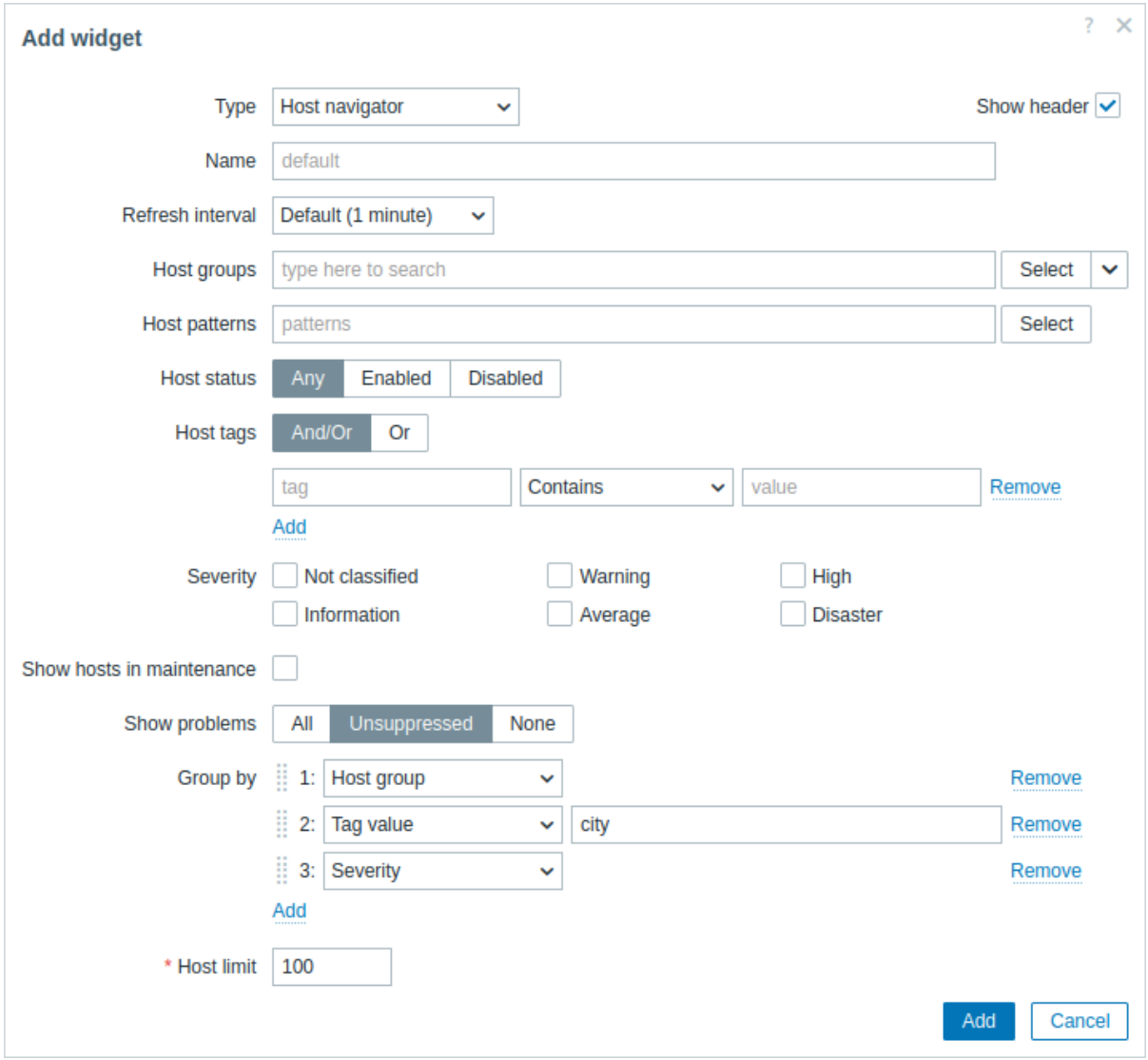
Os grupos pelos quais os hosts são organizados podem ser expandidos ou recolhidos.

Para grupos, problemas e hosts em manutenção, detalhes adicionais são acessíveis por dicas ao passar o mouse.

O widget é mais útil para controlar o que outros widgets exibem com base no host selecionado.

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 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 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 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. 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 aos 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.

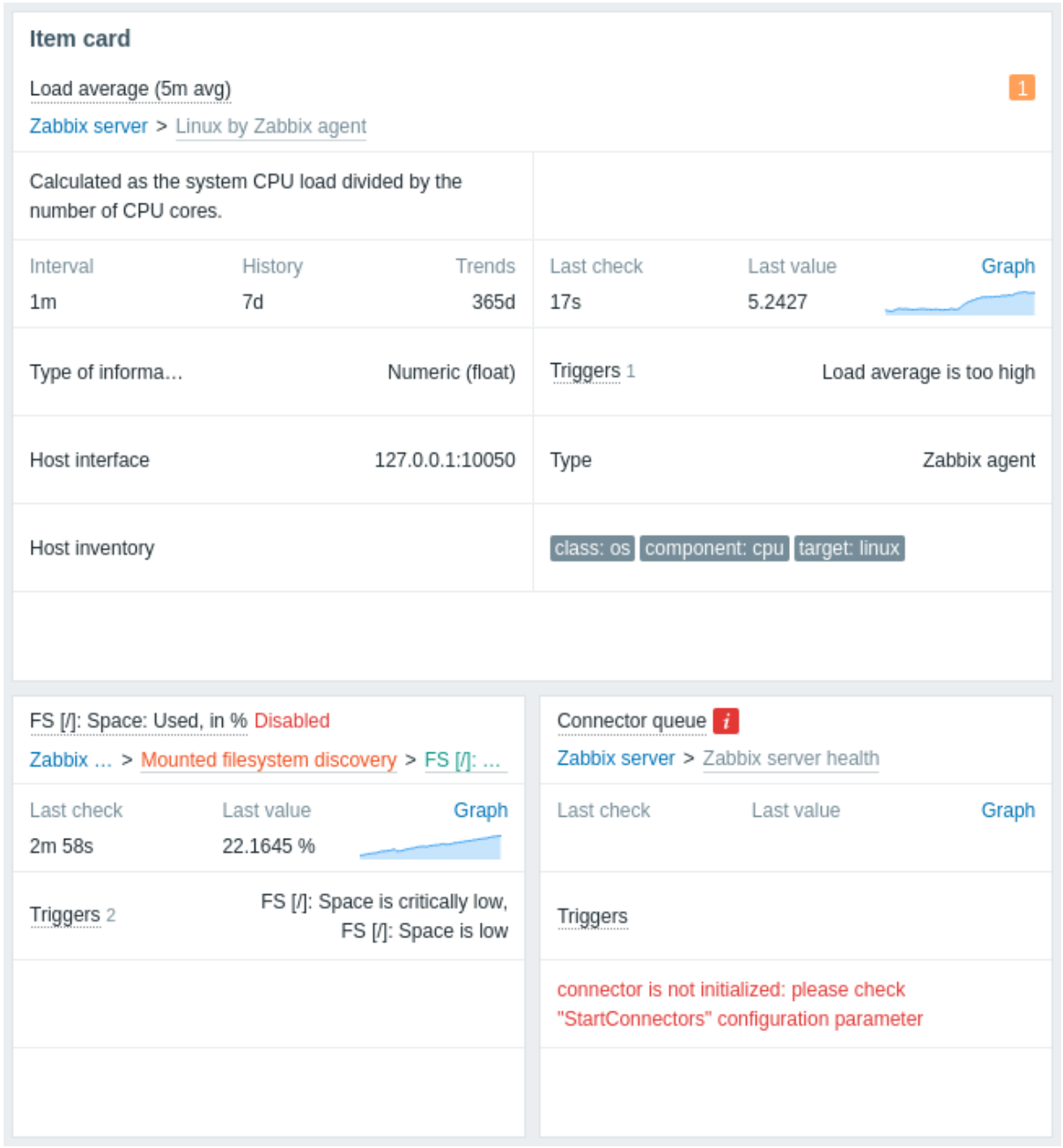
Tags do host	Especifique tags para filtrar os hosts exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - 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; **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 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 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 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	
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 correspondam aos atributos de agrupamento configurados são exibidos no grupo *Sem categoria*. Até 10 atributos de agrupamento podem ser especificados, e todos devem ser únicos. 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 Cartão de item

Visão geral

O widget *Item card* exibe detalhes de um único item, permitindo que você avalie rapidamente o status e a configuração de um item (dados mais recentes, erros, descrição, etc.).

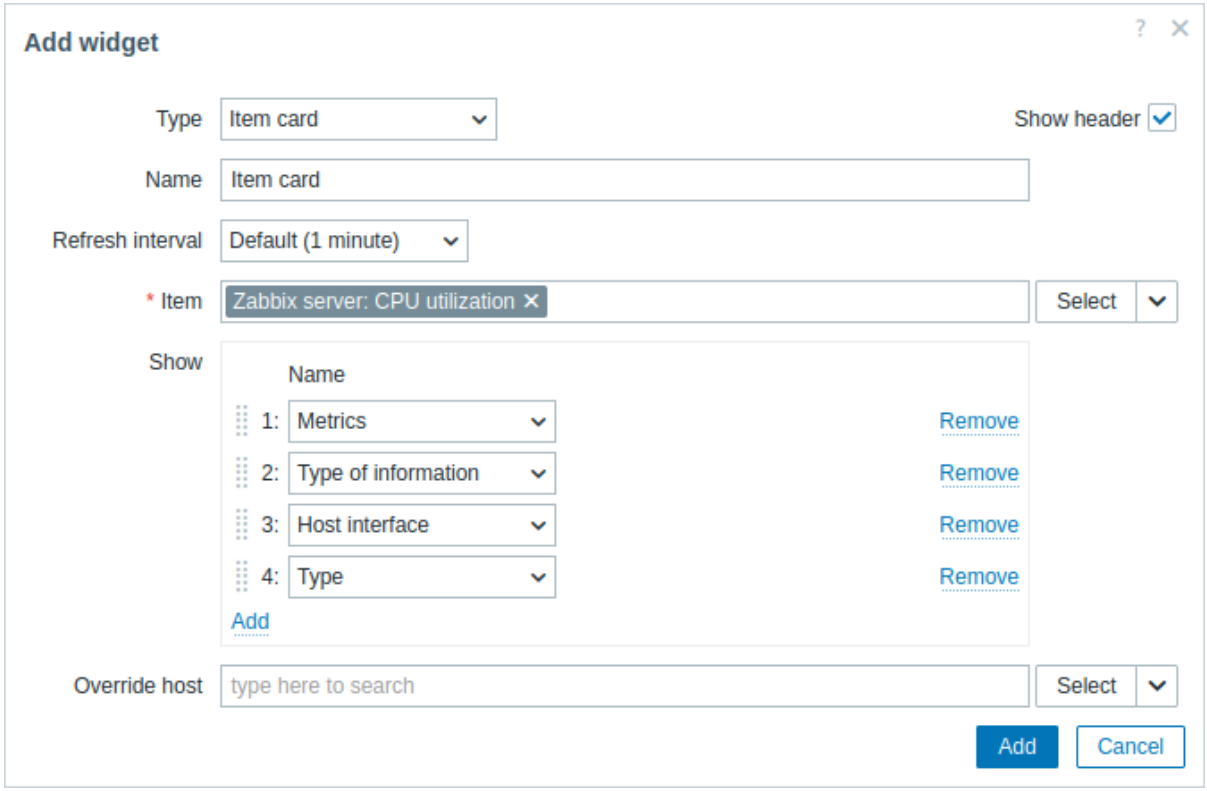


O item pode ser definido diretamente na configuração do widget ou selecionado a partir de um widget compatível no dashboard.

O widget pode exibir várias seções, cada uma mostrando informações diferentes. O layout das seções é ajustado automaticamente com base na largura do widget; quando expandido horizontalmente, as seções são automaticamente reorganizadas em várias colunas.

Configuração

Para configurar, selecione *Item card* como tipo:



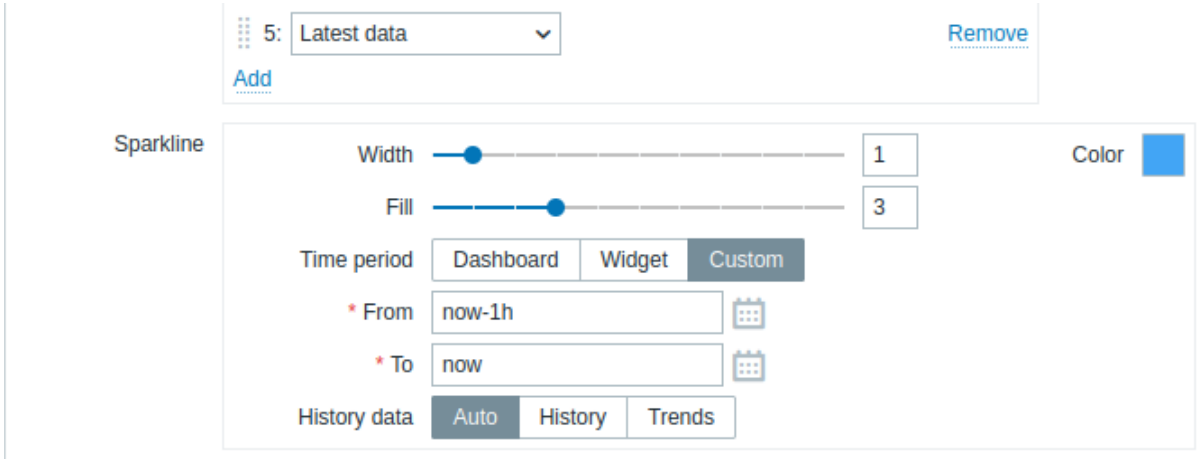
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Item	Selecione o item. Como alternativa, selecione um widget compatível como fonte de dados para os itens. Sugestões correspondentes aparecem enquanto você digita. A seção superior do widget sempre exibe o **nome do item** (clicar nele abre o menu do item): - Se o item estiver desabilitado, um rótulo vermelho *Desabilitado* é exibido ao lado do nome do item. - Em caso de erros, um ícone quadrado com a letra "i" é exibido. Em caso de problemas, um ícone quadrado com a contagem de problemas é exibido para cada severidade de problema. Passar o mouse sobre um ícone abre um pop-up com detalhes. - Em um dashboard de template, o nome técnico (em vez do nome visível) do item é exibido. A seção superior do widget também exibe o **caminho do item**—seu host e, se aplicável, template, regra de descoberta de baixo nível (LLD) ou item mestre (clicar em uma entidade abre sua configuração). O caminho do item é exibido apenas para usuários com permissões para as respectivas entidades.
Mostrar	Adicione seções para exibir no widget. As seções podem ser reorganizadas arrastando e soltando.
Descrição	Exibe a descrição do item. Macros são resolvidas.
Texto de erro	Exibe o erro do item (se houver).
Métricas	Exibe as seguintes informações: - Intervalo de atualização do item (exceto para itens trapper); - Configuração de histórico do item (se houver); - Configuração de tendências do item (se houver). Macros são resolvidas. Se o item usar intervalos personalizados, um ícone de ajuda é exibido ao lado do intervalo de atualização do item. Passar o mouse sobre o ícone abre um pop-up com detalhes do intervalo personalizado.

Últimos dados	Exibe as seguintes informações: - Tempo desde a última verificação do item; - Último valor do item; - Link para gráfico simples/histórico dos valores do item (se o histórico ou tendências forem armazenados); - Gráfico sparkline configurável (se o item for numérico). Para itens do tipo binário, uma miniatura ou a opção *Mostrar* é exibida em vez do último valor. Passar o mouse sobre a miniatura abre um pop-up com a imagem. Passar o mouse sobre *Mostrar* abre um pop-up com o valor do item (string Base64). Em um dashboard de template, *Sem dados* é exibido em vez do último valor do item.
Tipo de informação	Exibe o tipo de informação do item.
Triggers	Exibe as seguintes informações: - Rótulo *Triggers* com a contagem de triggers (passar o mouse sobre *Triggers* abre um pop-up com detalhes das triggers); - Lista de nomes das triggers (macros são resolvidas).
Interface do host	Exibe a interface do host usada pelo item, ou *Sem dados* se o item não tiver uma interface de host (por exemplo, itens dependentes, verificações simples). Em um dashboard de template, *Sem dados* é exibido.
Tipo	Exibe o tipo do item.
Inventário do host	Exibe o campo de inventário do host que o valor do item preenche.
Tags	Exibe as tags do item.
Substituir host	Selecione um widget compatível ou o seletor de host do dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Sparkline

As opções de configuração do gráfico Sparkline estão disponíveis apenas se a seção *Últimos dados* for adicionada ao widget.



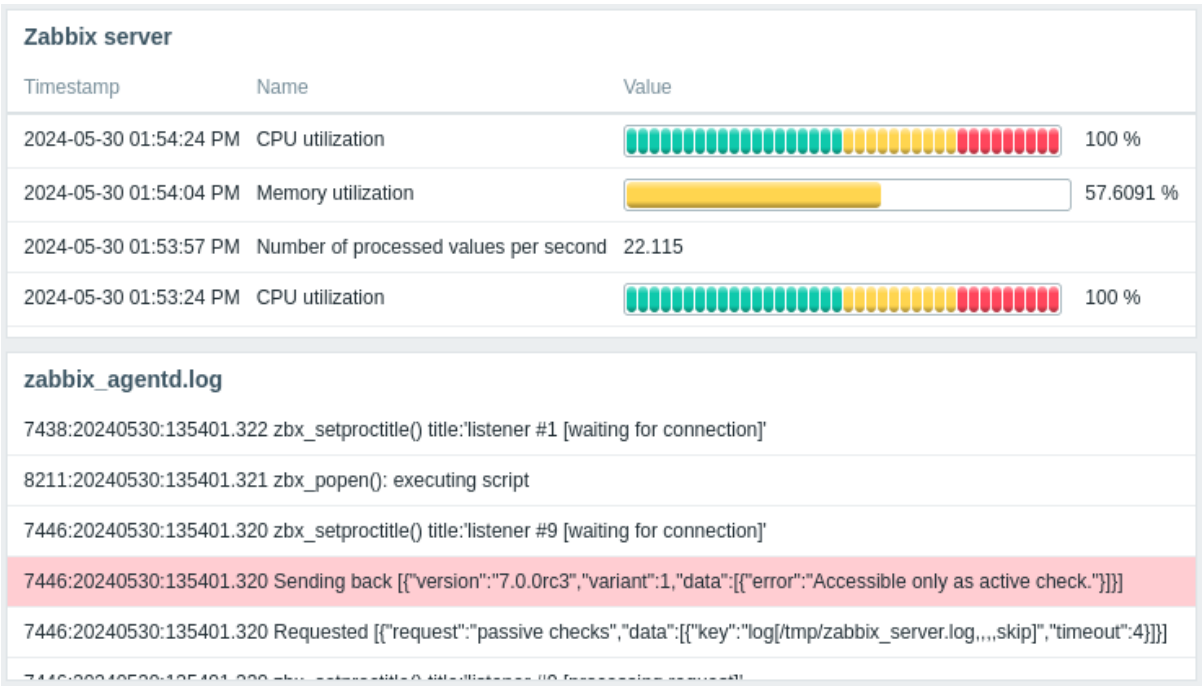
Largura	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Cor	Selecione a cor da linha e do preenchimento.
Preenchimento	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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”.

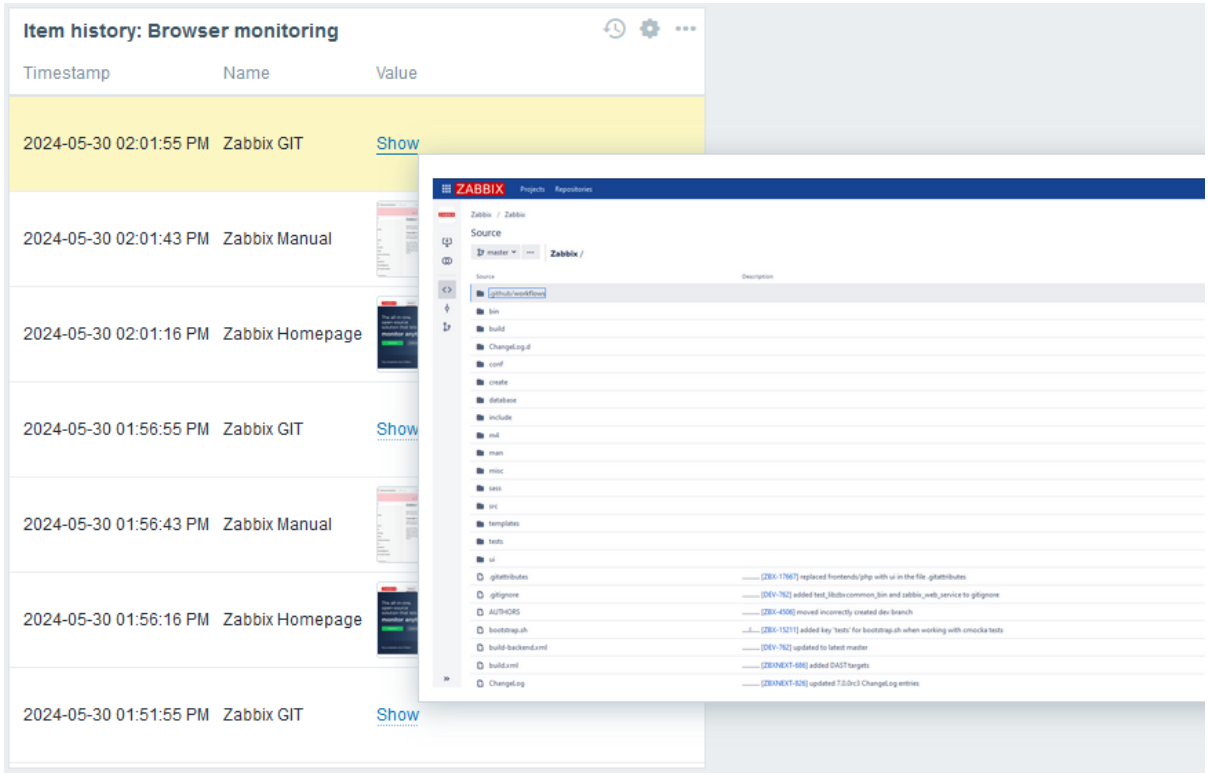
De	Digite ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Dados do histórico	Obtenha dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obtém dados do histórico; **Tendências** - obtém dados das tendências. Este parâmetro se aplica apenas a dados numéricos. Dados não numéricos sempre serão obtidos do histórico.

16 Histórico de item

Visão geral

O widget *Histórico de item* exibe os dados mais recentes para todos os tipos de item (numérico, texto, etc.) em formato de tabela. Ele pode mostrar barras de progresso e imagens (útil para itens de navegador) e pode destacar valores (útil para monitoramento de arquivos de log).

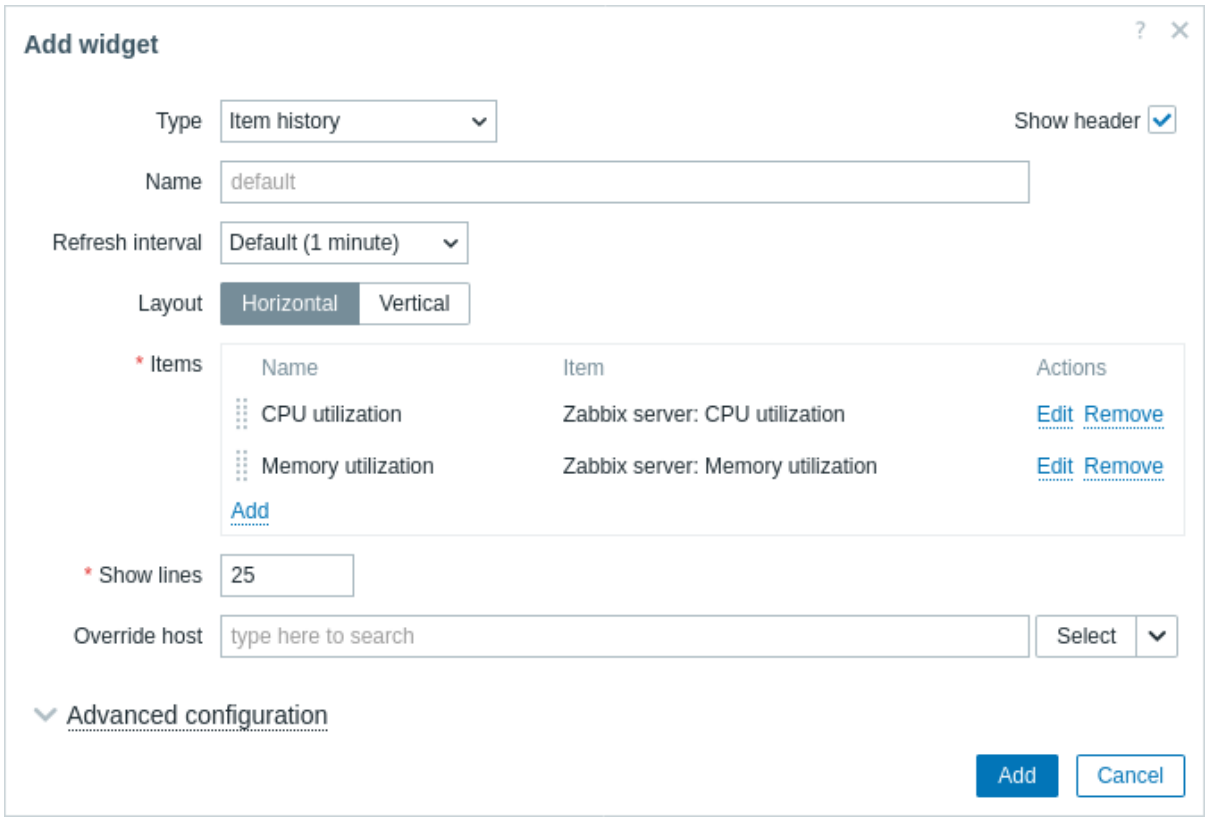




Até 1000 registros podem ser exibidos.

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 dos items: **Horizontal** - os items serão exibidos horizontalmente, os valores verticalmente; **Vertical** - os items serão exibidos verticalmente, os valores horizontalmente.

Items	Adicione colunas de item para exibir. A ordem dos items determina sua ordem de exibição. Os items podem ser reordenados arrastando-os para cima ou para baixo pela alça antes do nome do item.
Exibir linhas	Especifique o número de linhas de valores de item a serem exibidas.
Substituir host	Selecione um widget compatível ou o seletor de host do 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 coluna

Para configurar as colunas de item, clique em *Adicionar* no parâmetro *Itens*:

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 items 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	Informe 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	Informe 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 ser salva. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.
Dados de histórico	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.

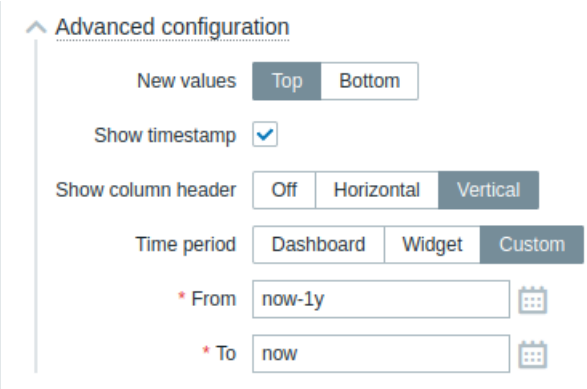
Exibir	Selecione como o valor do item deve ser exibido: **Como está** - exibido como está: - quebras de linha são observadas se o valor recebido tiver retornos de linha; - sem quebra de linha com layout horizontal ou layout vertical com uma coluna; quebra de linha com layout vertical e mais de uma coluna. Passe o mouse ou clique no valor para abrir um pop-up com o valor. **HTML** - como texto 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 de seleção para exibir o valor do item em fonte monoespaçada (desmarcada por padrão).
Exibir hora local	Marque esta caixa de seleção para exibir a hora local em vez do timestamp na coluna de timestamp. Observe que a caixa de seleção *Mostrar 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 são imagens. Desmarque esta caixa de seleção para exibir uma opção ”Mostrar” para todos os valores de item binário. 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 abre 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; **Inferior** - na parte inferior das colunas.
Exibir data/hora	Marque esta caixa para exibir a coluna de data/hora (desmarcada por padrão).
Exibir 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** - usar o seletor de período de tempo do dashboard; **Widget** - usar um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usar um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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 fim 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".

17 Navegador de item

Visão geral

O widget *Navegador de items* exibe uma lista de items com base em várias opções de filtragem e agrupamento.

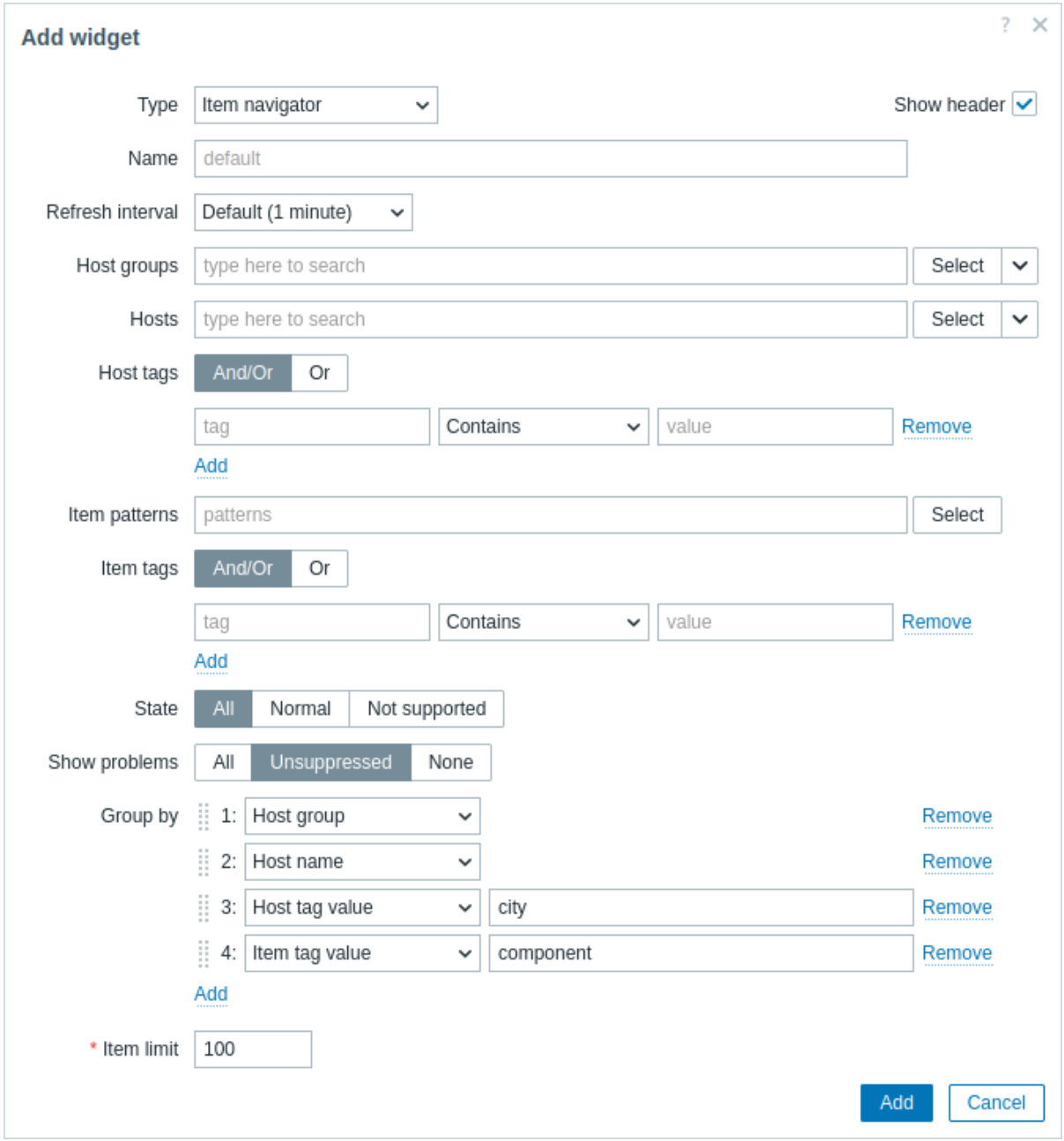
Os grupos pelos quais os items são organizados podem ser expandidos ou recolhidos.

Para grupos e problemas, detalhes adicionais são acessíveis por dicas ao passar o mouse.

O widget é mais útil para controlar o que outros widgets exibem com base no item selecionado.

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 com os 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 com os 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.

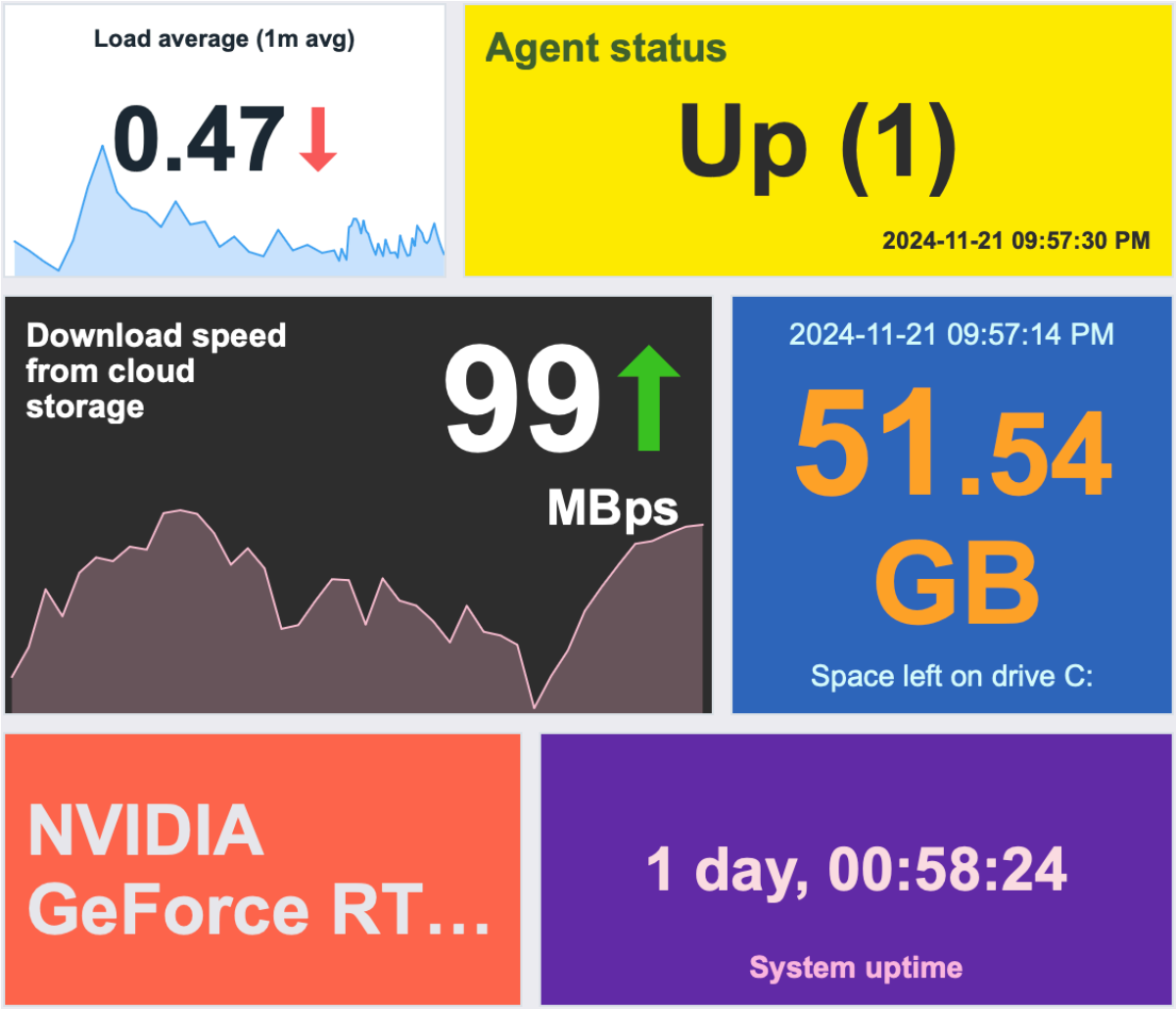
Tags de host	Especifique tags para filtrar os items exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - 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.
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 comecem 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. 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.
Estado	Filtra quais items exibir com base em seu estado (todos, normal, não suportado).

Exibir problemas	Filtra quais problemas exibir com base em seu status (todos, não suprimidos, nenhum). Os problemas são indicados com blocos coloridos no final da linha do item. As cores são baseadas na severidade do problema, que pode ser personalizada em *Administração > Geral > Opções de exibição de triggers*. As severidades dos problemas podem ser alteradas ao atualizar problemas.
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 de 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”, ”Tokyo”, etc.); **Valor da tag de 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, Tokyo, 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 todos os items subordinados; - para cada item, é exibida apenas sua contagem de problemas. 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*.

18 Valor do item

Visão geral

O widget *Valor do item* exibe o valor de um único item numérico ou de texto. É útil para monitorar métricas importantes, visualizar limites e detectar mudanças repentinas nos dados.



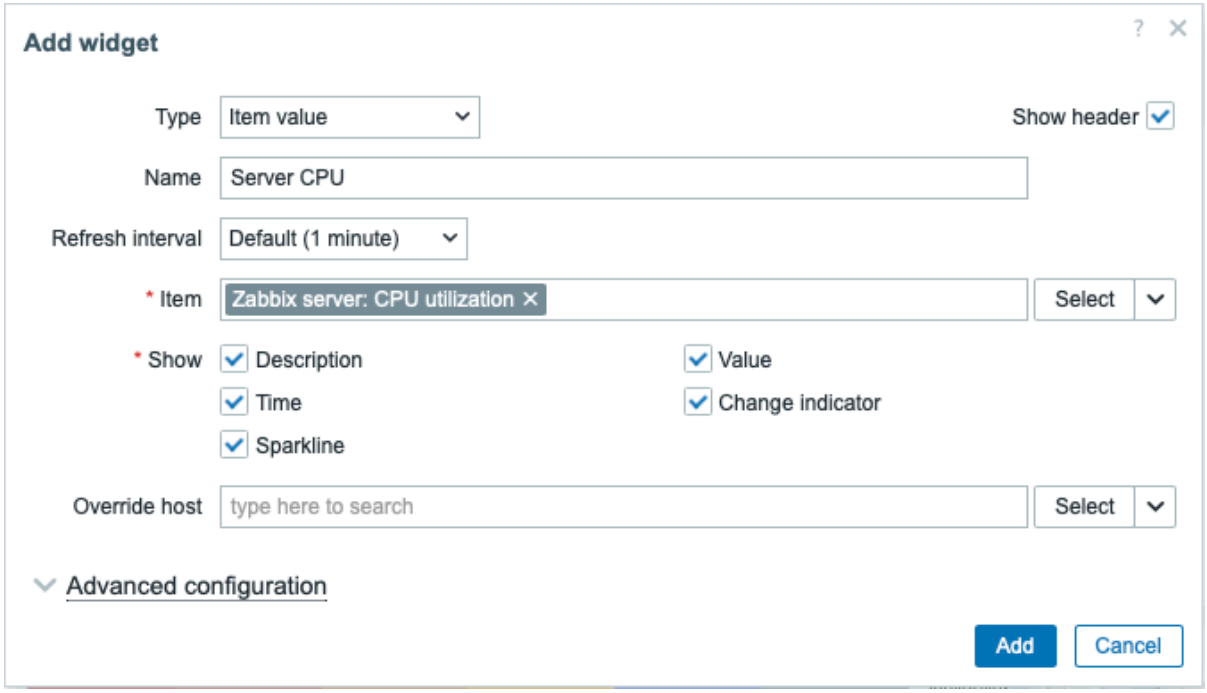
Você pode configurar o widget para exibir o seguinte:

- Data/hora da métrica (2024-11-21 09:57:30)
- Descrição do item (Load average (1m), Agent status)
- Valor do item, indicador de alteração e unidades (0.47 ↓, 99 ↑ Mbps)
- Cor de fundo
- Gráfico de linha para valores do período de tempo especificado

Clicar no widget abre um gráfico ad-hoc para itens numéricos ou os últimos dados para itens de texto.

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 itens. Itens que retornam dados binários não são suportados.
Mostrar	Marque a caixa de seleção para exibir o respectivo elemento: descrição, valor, hora, indicador de alteração, sparkline. Desmarque para ocultar. Pelo menos um elemento deve ser selecionado.
Substituir host	Selecione um widget compatível ou o seletor de host do 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.

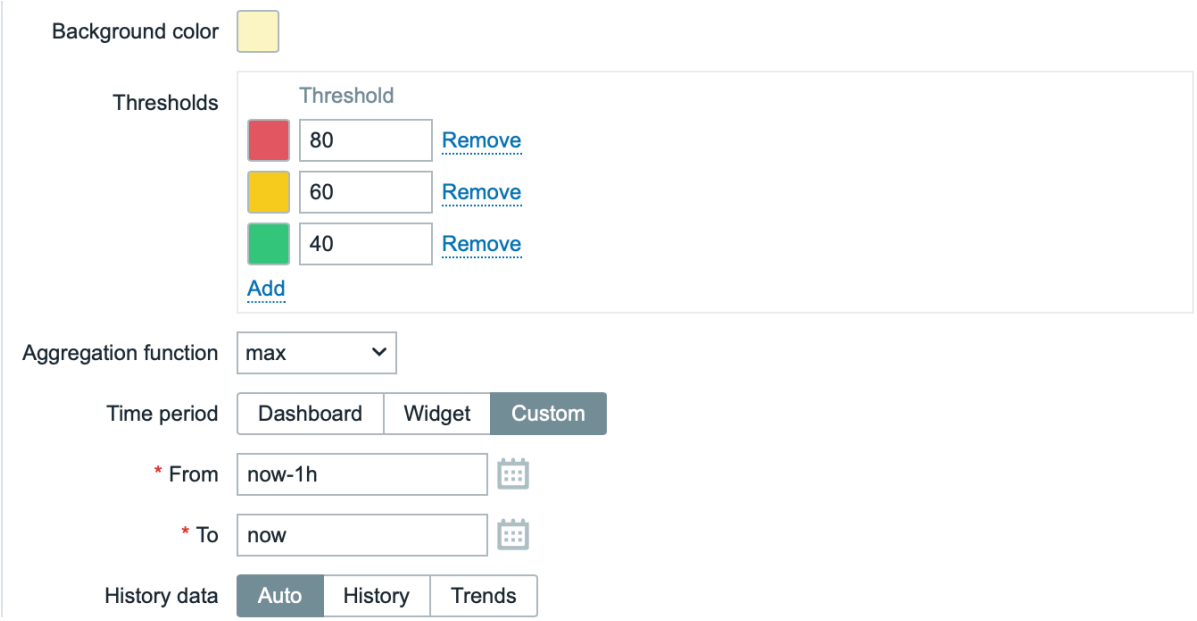
Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*, e apenas para os elementos que estão selecionados no campo *Mostrar* (veja acima).

Observe que vários elementos não podem ocupar o mesmo espaço; se forem colocados no mesmo espaço, uma mensagem de erro será exibida.

Parâmetros genéricos do widget

Esses parâmetros determinam a cor de fundo (estática ou dinâmica) para todo o widget e uma função de agregação para exibir valores.

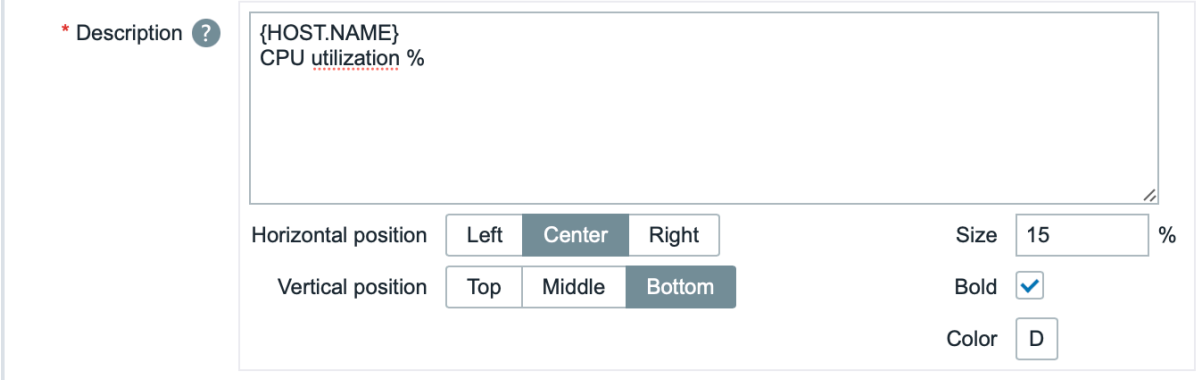


Cor de fundo	Selecione a cor de fundo para todo o widget no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Limiares	Configure a cor de fundo dinâmica para todo o widget. Clique em *Adicionar* para 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 mudará. A lista será classificada em ordem crescente ao salvar. Observe que a cor de fundo dinâmica será exibida corretamente apenas para items 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	Período de tempo a ser usado para agregar valores. Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo. Este parâmetro não está disponível se a *Função de agregação* estiver definida como "não usado".
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".

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.

Descrição

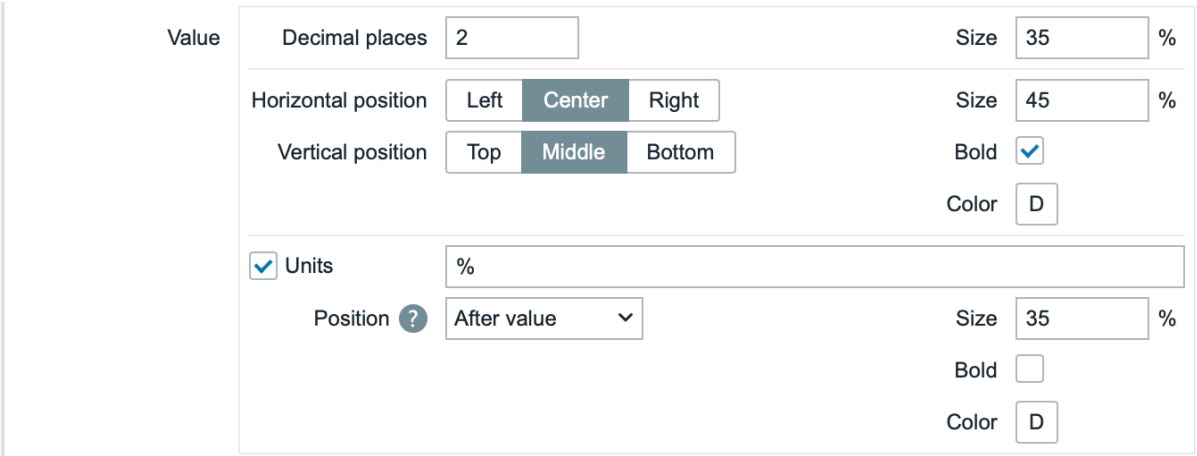
Esses parâmetros determinam como a descrição do item deve ser exibida.



Descrição	Insira a descrição do item. Esta descrição pode substituir o nome padrão do item. Descrições multilinha são suportadas. Uma combinação de texto e macros suportadas é possível. {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 porcentagem relativa 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

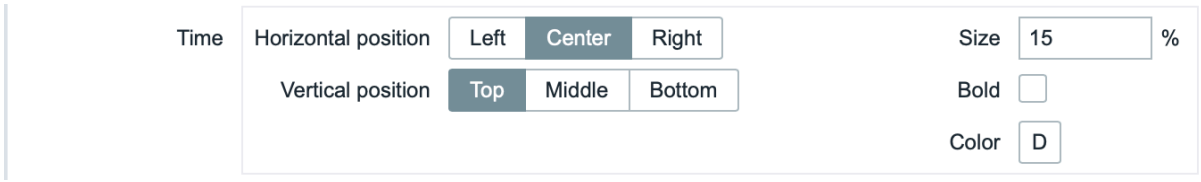
Esses parâmetros determinam como o valor do item deve ser exibido.



Casas decimais	Selecione quantas casas decimais serão exibidas com o valor. Esse valor afetará apenas itens do tipo float. Para itens com unidades definidas como ”s”, se esse 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 para uma string de tempo legível (por exemplo, ”4h 56m 30s”).
Tamanho	Informe o tamanho da fonte para as casas decimais (em porcentagem relativa à altura total do widget).
Posição horizontal	Selecione a posição horizontal do valor do item - esquerda, direita ou centro.
Posição vertical	Selecione a posição vertical do valor do item - topo, inferior ou meio.
Tamanho	Informe o tamanho da fonte para o valor do item (em porcentagem relativa à altura total do widget). Observe que o tamanho do valor do item tem prioridade; outros elementos precisam ceder espaço para o valor. 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	Informe o tamanho da fonte para a unidade do item (em porcentagem relativa 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

Esses parâmetros determinam como a hora (valor do relógio do histórico do item) deve ser exibida.



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 porcentagem relativa 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

Esta seção permite selecionar a cor dos indicadores de alteração no seletor de cores.



Os indicadores de alteração são os seguintes:

- ↑ - o valor do item subiu (para itens numéricos)
- ↓ - o valor do item caiu (para itens numéricos)
- ↕ - o valor do item mudou (para itens de string e itens com mapeamento de valor)

Nota:

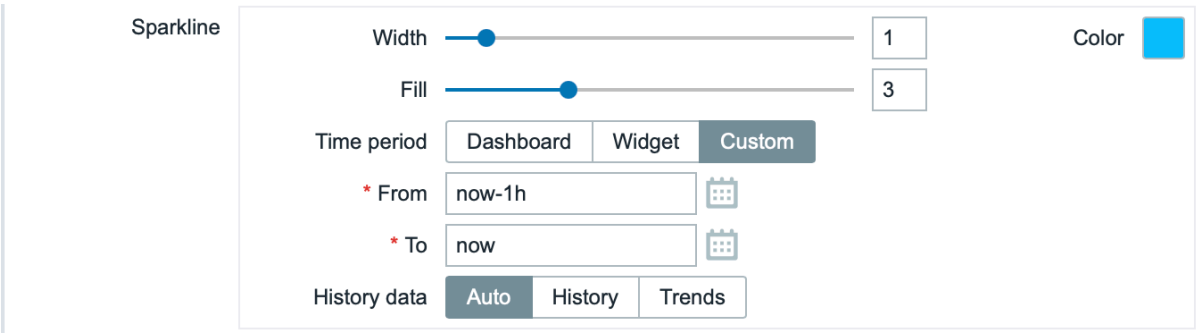
- Os indicadores de subida (↑) e descida (↓) não são exibidos se houver apenas um valor.
- Os valores são sempre comparados dentro do mesmo escopo; por exemplo:
  - O valor mais recente é comparado ao valor anterior.
  - Um valor mensal é comparado ao mês anterior.
  - Para agregações, o período anterior tem a mesma duração do período selecionado, terminando imediatamente antes do início do período selecionado.

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).

Sparkline

Esses parâmetros determinam como o gráfico sparkline deve ser exibido.

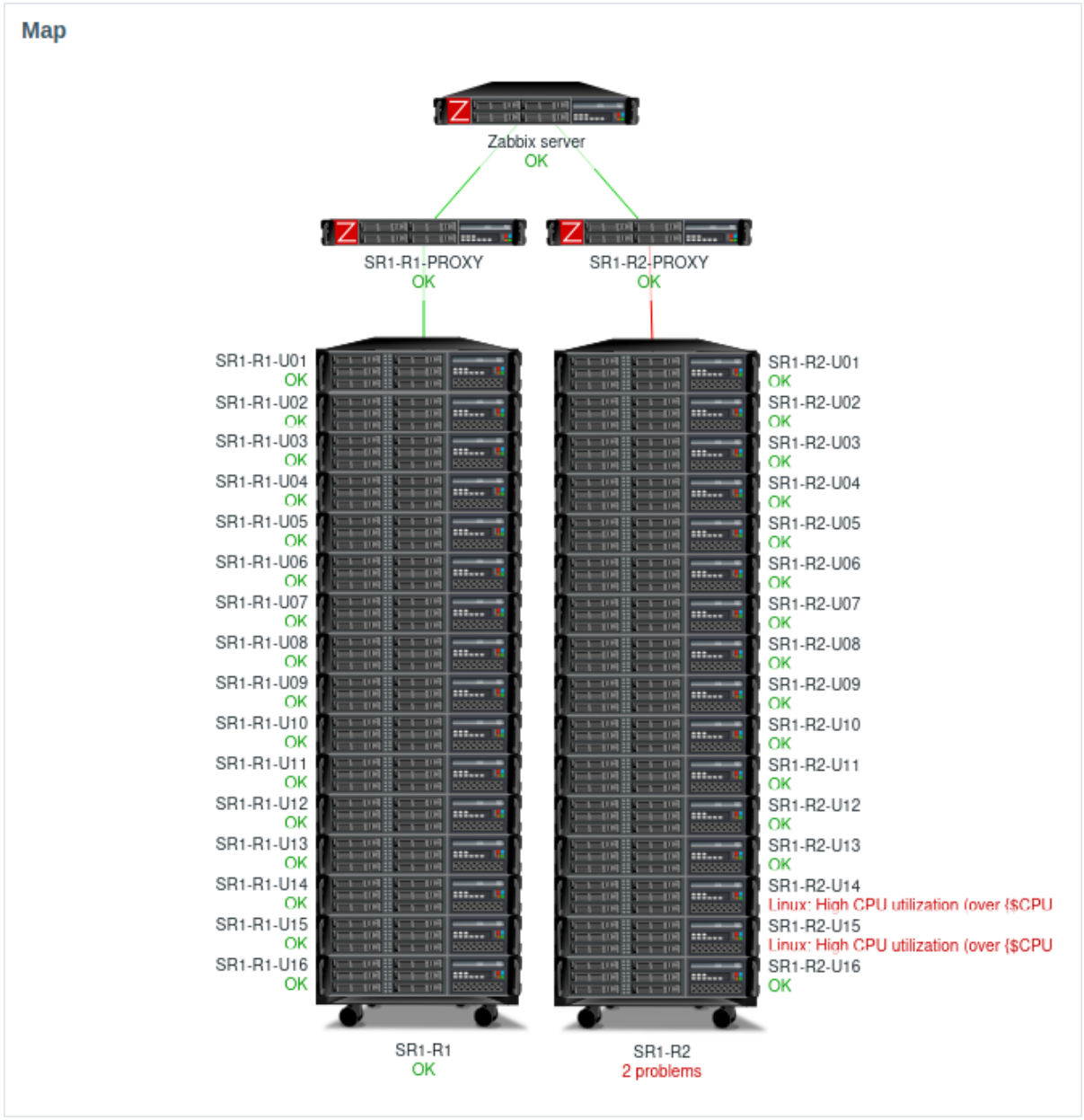


Largura	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Cor	Selecione a cor da linha e do preenchimento.
Preenchimento	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Período de tempo	Especifique o período de tempo para os valores a serem incluídos no gráfico sparkline. Selecione a fonte de dados para o período de tempo: **Dashboard** - use o seletor de período de tempo do dashboard; **Widget** - use um widget compatível (definido no parâmetro *Widget*); **Personalizado** - use um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo. Este parâmetro não está disponível se a *Função de agregação* estiver definida como "não usada".
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".
Dados de histórico	Obtenha dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obter dados do histórico; **Tendências** - obter dados das tendências.

19 Mapa

Visão geral

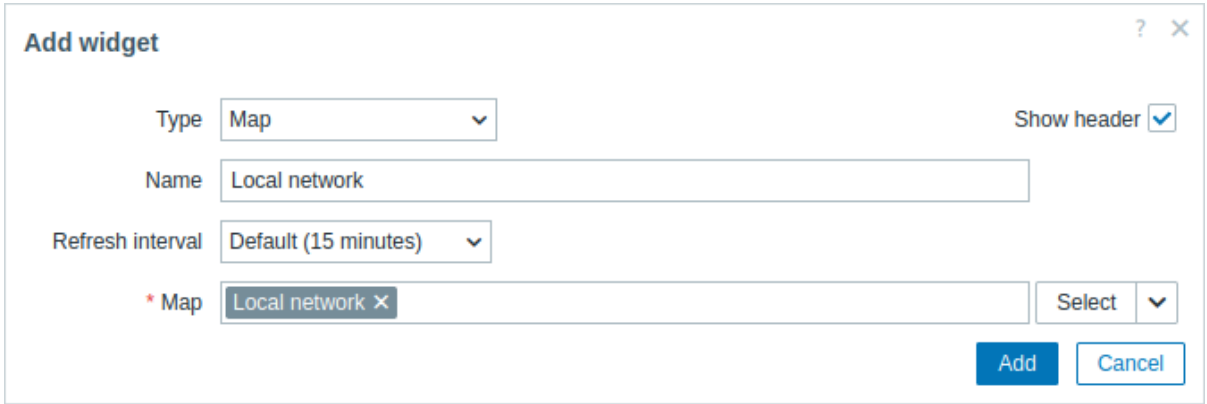
O widget *Mapa* pode exibir um mapa de rede ou uma visualização semelhante, fornecendo uma visão geral dinâmica da sua rede.



Você pode combiná-lo com o widget Árvore de navegação de mapas para exibir dinamicamente o mapa selecionado na árvore.

Configuração

Para configurar, selecione *Mapa* como tipo:



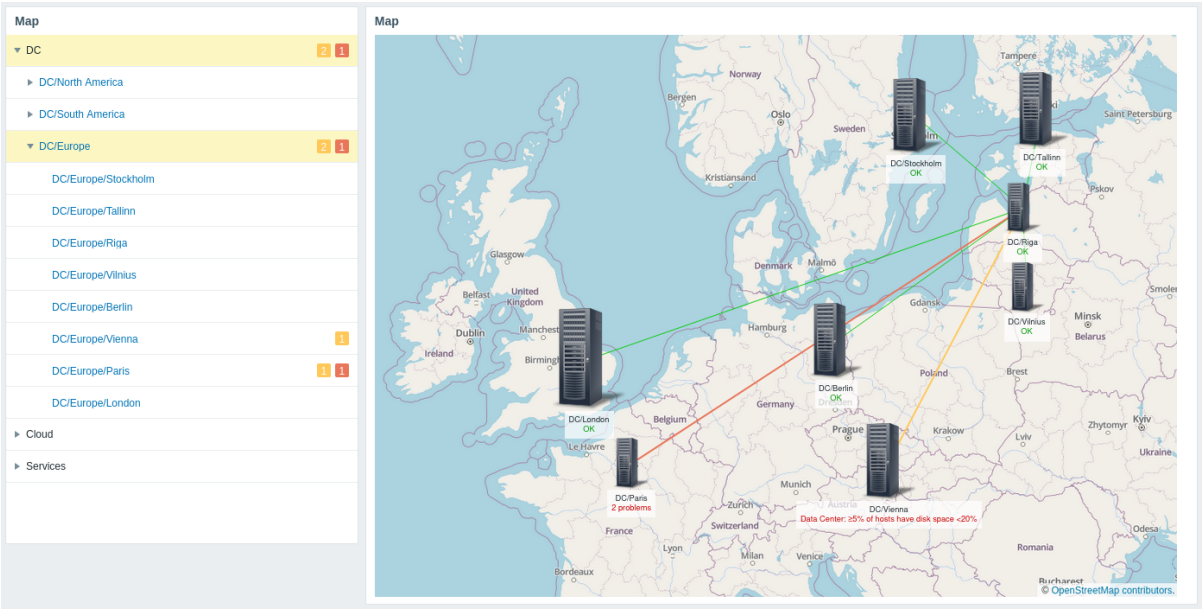
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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 de preenchimento automático, portanto, ao começar a digitar o nome do mapa ou widget, será exibida uma lista suspensa de mapas ou widgets correspondentes.

20 Árvore de navegação do mapa

Visão geral

O widget *Árvore de navegação de mapas* exibe uma hierarquia dos mapas existentes e a contagem de problemas para cada mapa e grupo de mapas.

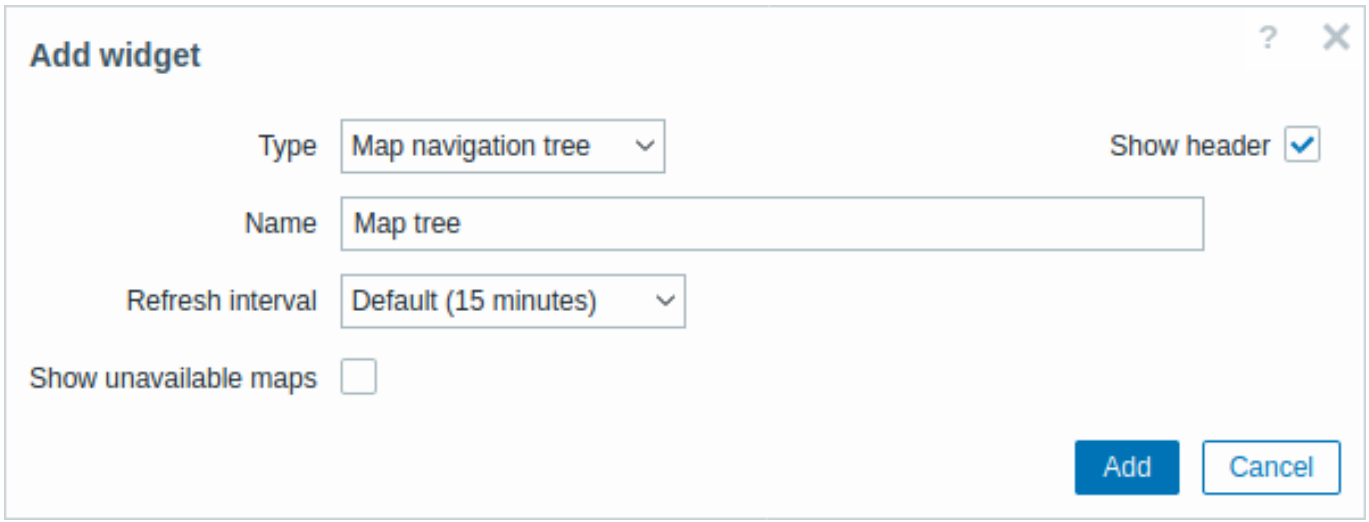


Você pode combiná-lo com o widget Mapa para exibir dinamicamente o mapa selecionado na árvore.

A contagem de problemas para o mapa de nível superior na hierarquia inclui todos os problemas dos submapas e os seus próprios.

Configuração

Para configurar, selecione *Árvore de navegação de mapas* como tipo:

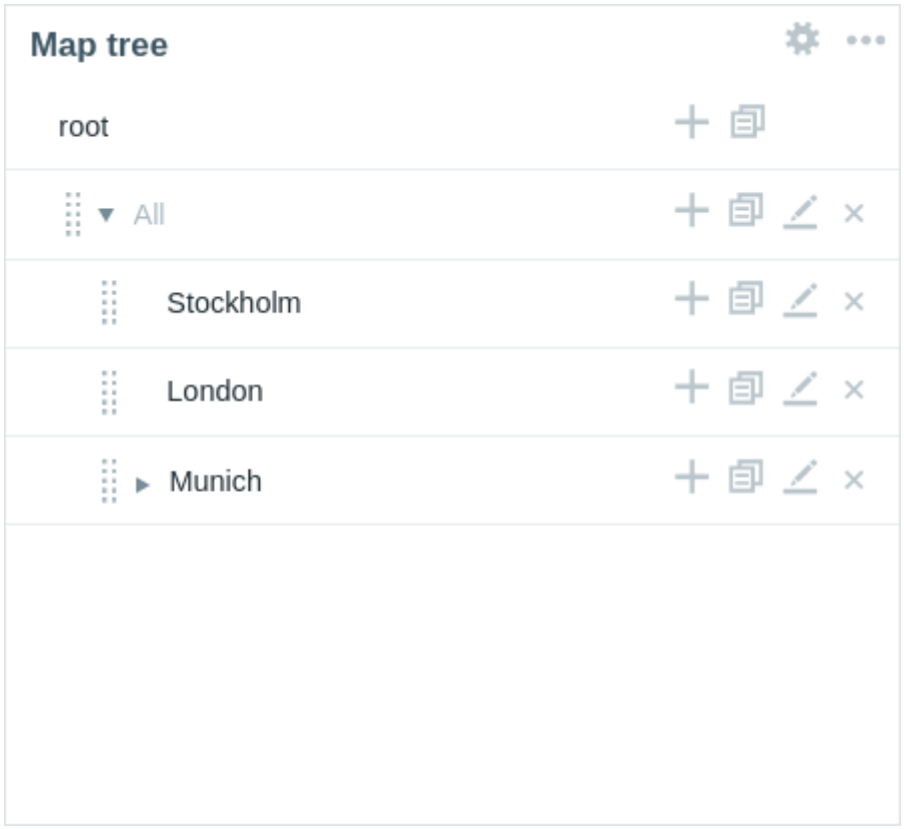


Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

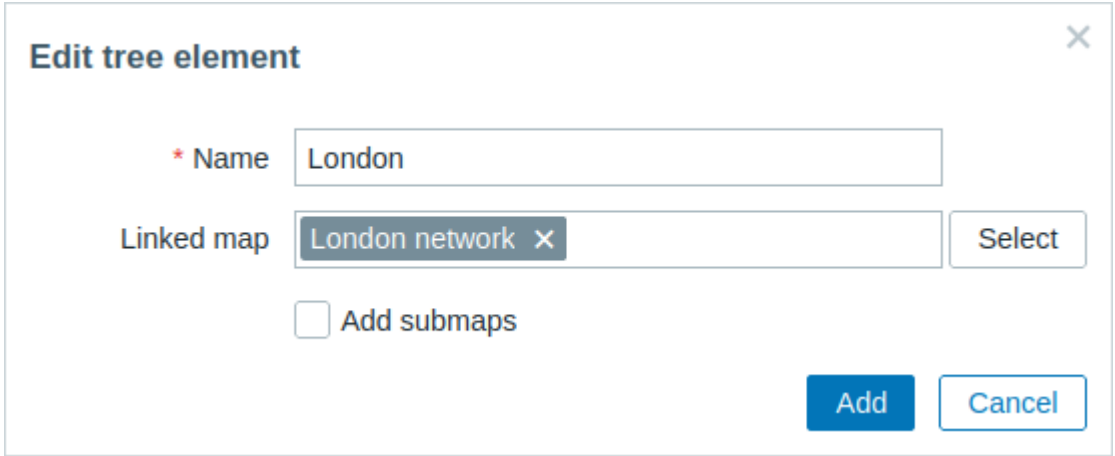
Os elementos da árvore de navegação são exibidos em uma lista. Você pode:

- 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.



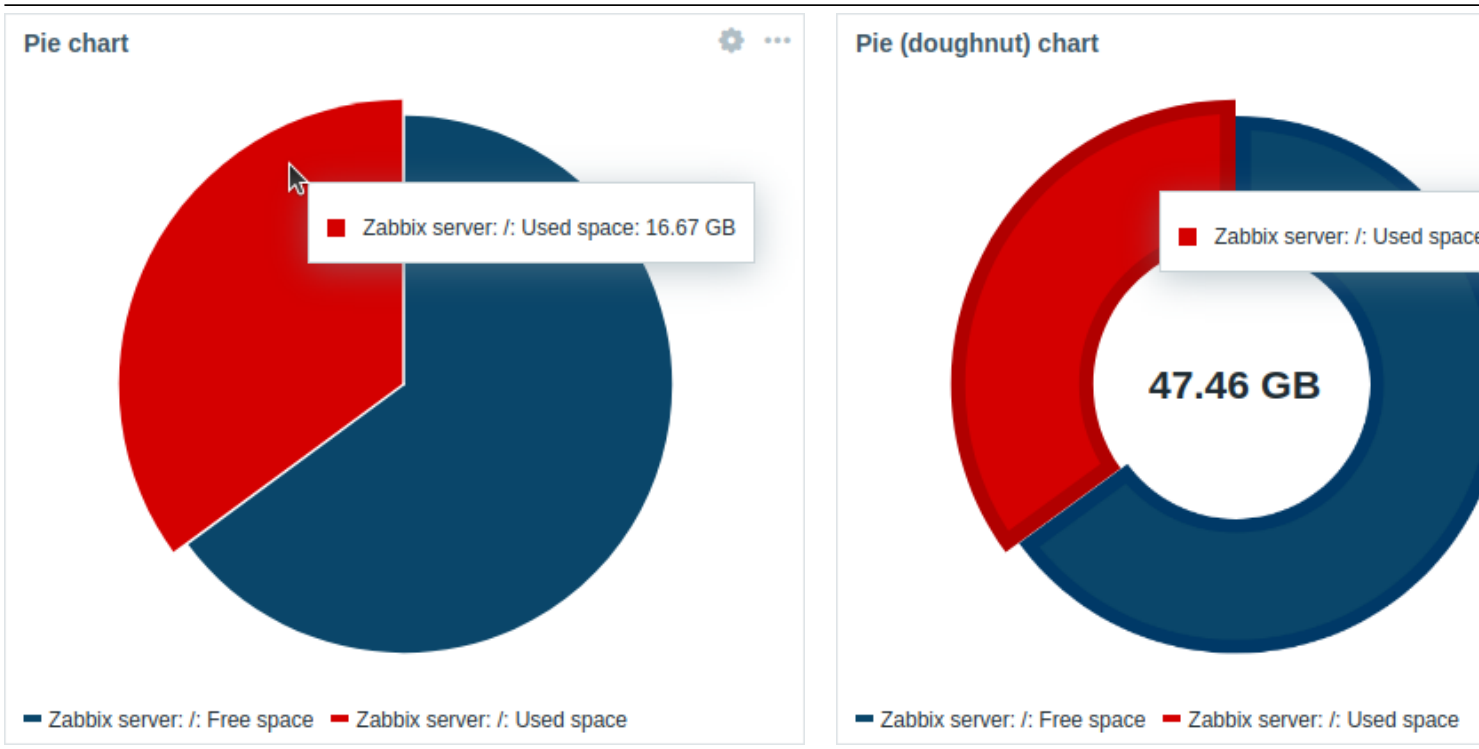
Os seguintes parâmetros de configuração do elemento da árvore de navegação estão disponíveis:

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 autocompletável, portanto, ao começar a digitar o nome de um mapa, será exibida uma lista suspensa com os 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.

21 Gráfico de pizza

Visão geral

O widget *Gráfico de pizza* exibe dados numéricos de item como um gráfico de pizza ou de rosca vetorial. Este widget pode ajudar você a visualizar como os items ou hosts contribuem para o conjunto geral de dados.



Ao passar o mouse, o setor destacado se expande e mostra sua legenda; ao clicar no setor, ele permanece expandido e sua legenda visível até ser fechado.

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. Você pode escolher uma única cor base ou selecionar uma linha de paleta para atribuir cores distintas a cada item correspondente.
- *Lista de itens* - os dados dos itens selecionados são exibidos. Você pode escolher a cor de cada item individualmente no seletor.

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 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 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 usado". 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 de 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 usado** - 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 de 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...”).

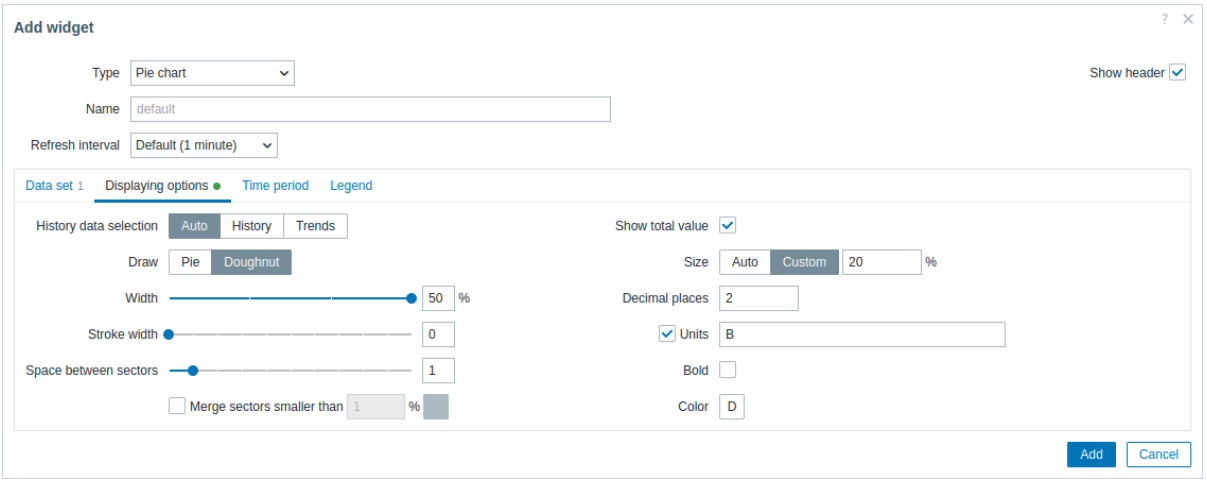
Detalhes da configuração do conjunto de dados

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 os 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 a seleção de dados históricos e opções de visualização para o gráfico de pizza:

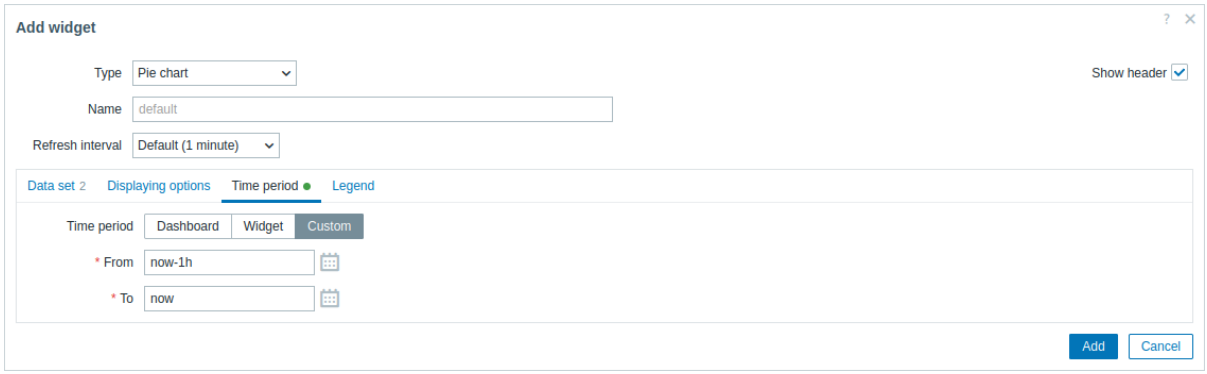


Seleção de dados históricos	Selecione a fonte de dados: **Auto** - os dados são obtidos de acordo com o algoritmo clássico (padrão); **Histórico** - dados do histórico; **Tendências** - dados das tendências.
Desenhar	Selecione o estilo de visualização do gráfico de pizza: **Pizza** - uma pizza completa (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	Selecione a largura da rosquinha: 20, 30, 40 ou 50% (padrão) do raio.
Largura da borda	Selecione a largura da borda do setor da rosquinha (0-10).
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 percentual 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

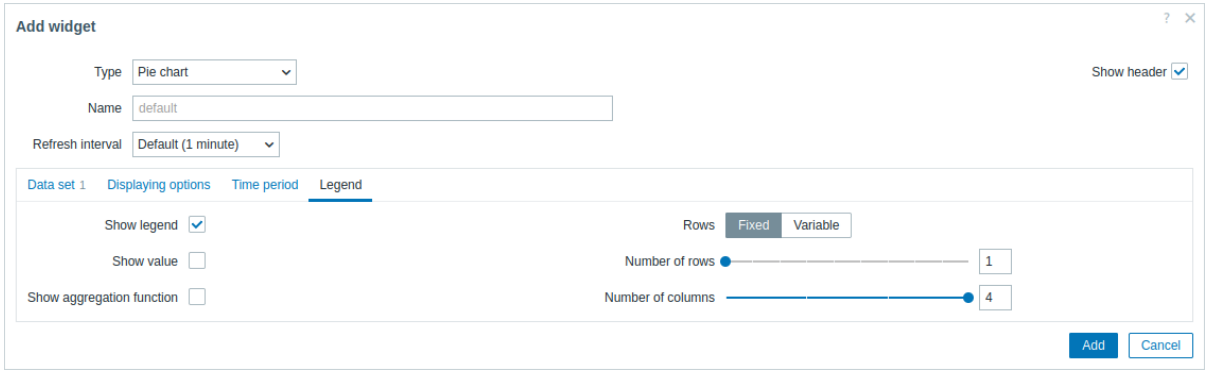
A aba **Período de tempo** permite definir um período de tempo personalizado para as configurações de agregação do gráfico de pizza:



Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado especificado nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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".

Legenda

A aba **Legenda** permite personalizar a legenda do gráfico de pizza:



Exibir legenda	Desmarque esta caixa para ocultar a legenda no gráfico de pizza (marcada por padrão).
Exibir valor	Marque esta caixa para exibir o valor 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 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 *Exibir valor* estiver desmarcado.

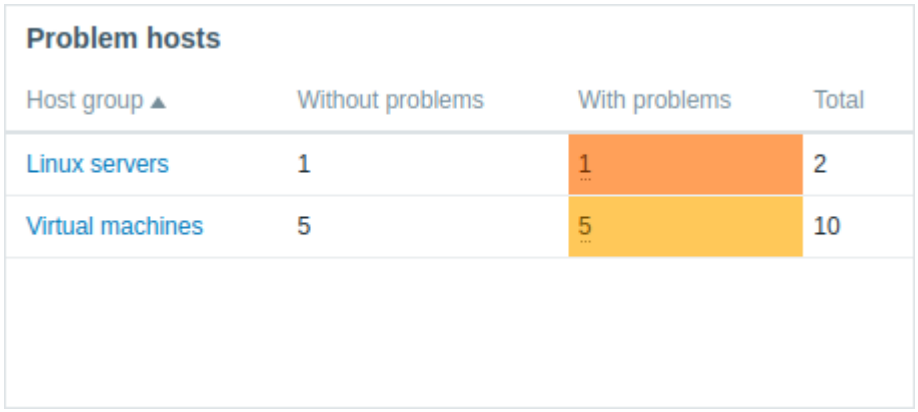
As informações exibidas pelo widget de gráfico de pizza podem ser baixadas como uma imagem *.png* usando o menu do widget.

Uma captura de tela do widget será salva na pasta *Downloads*.

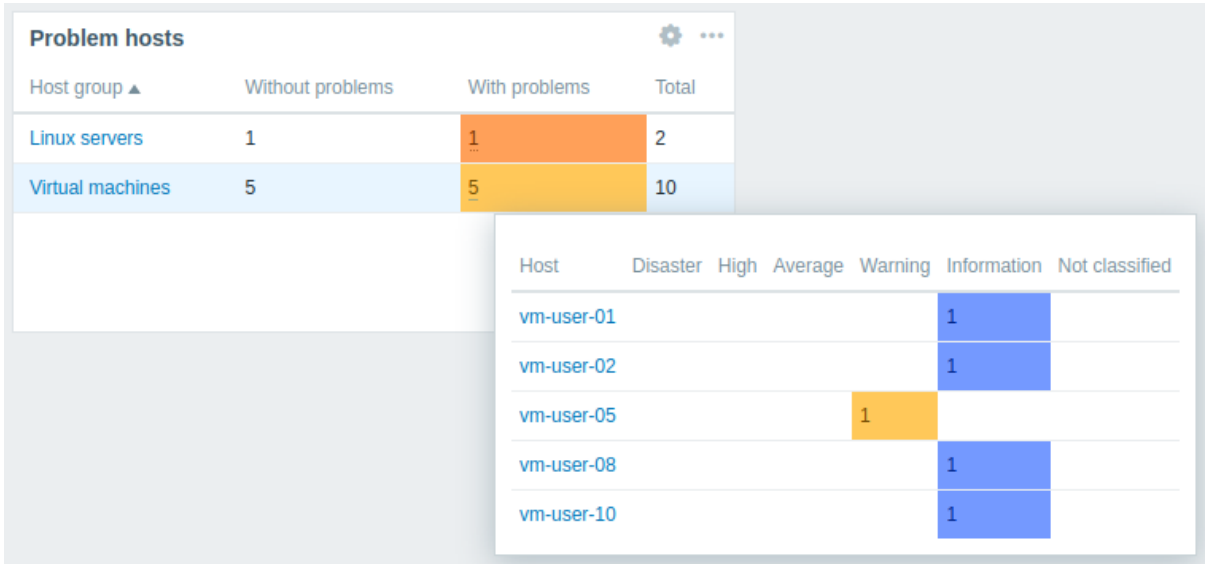
22 Hosts com problemas

Visão geral

O widget *Hosts com problemas* exibe o número de problemas de causa por grupo de hosts e mostra a maior severidade de problema em cada grupo. Ele permite filtrar problemas por diferentes parâmetros (grupos de hosts, hosts, nomes de problemas, etc.).

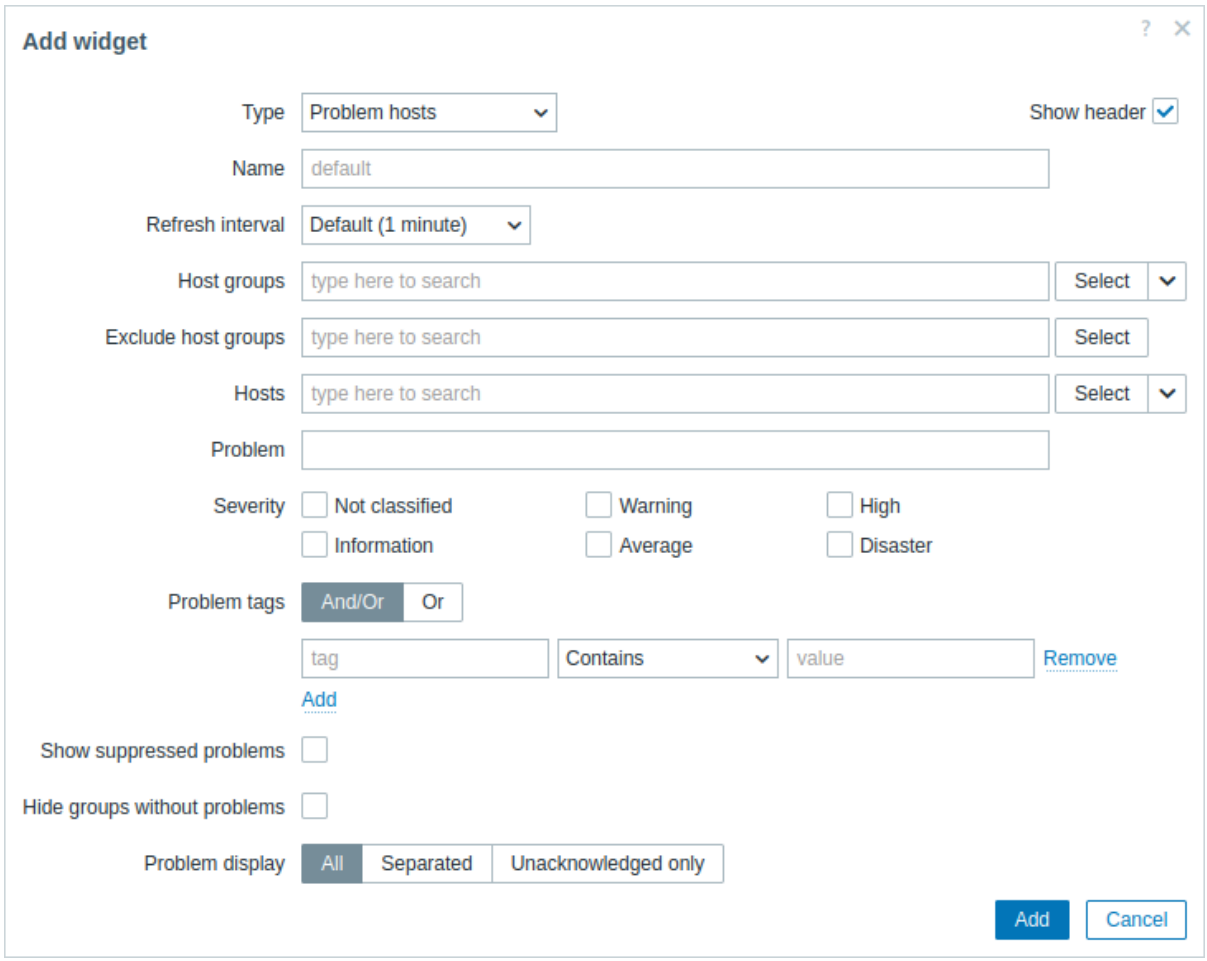


Ao passar o mouse sobre a contagem de problemas, é exibida a lista de hosts com problemas:



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:

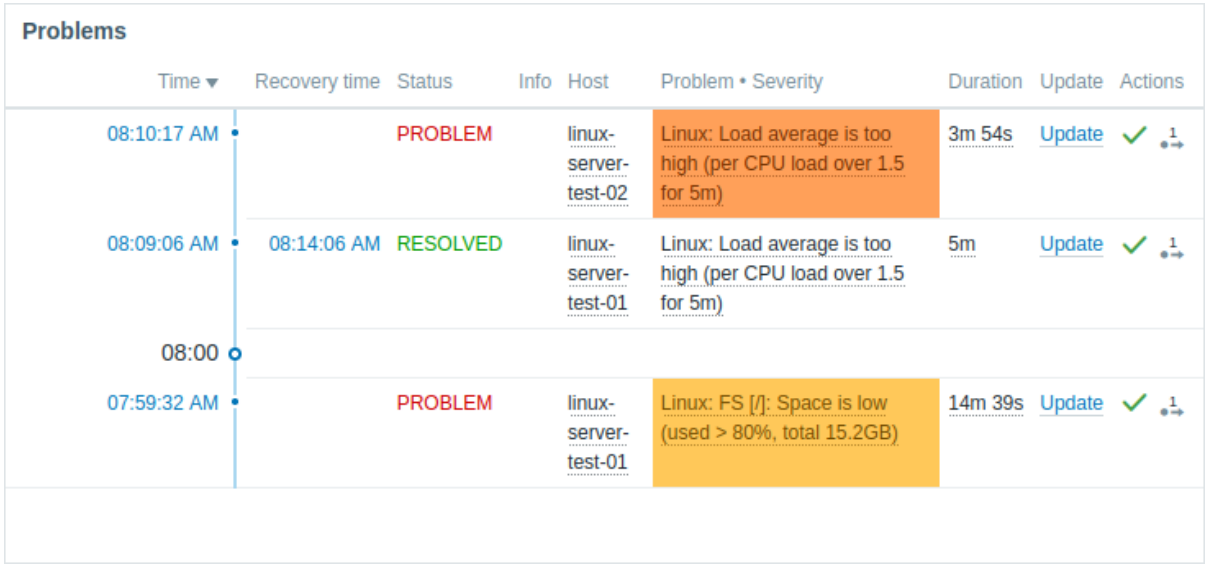
Grupos de hosts	Selecione grupos de hosts para exibir no widget. Alternativamente, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. 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 para *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 hosts para exibir 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, 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 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 das tags especificadas; **Igual** - inclui os nomes e valores das tags especificadas (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes das tags especificadas onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes das tags especificadas; **Diferente** - exclui os nomes e valores das tags especificadas (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes das tags especificadas 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.
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	Exibir a contagem de problemas como: **Todos** - a contagem total de problemas será exibida; **Separados** - a contagem de problemas não reconhecidos será exibida separada como um número da contagem total de problemas; **Apenas não reconhecidos** - apenas a contagem de problemas não reconhecidos será exibida.

23 Problemas

Visão geral

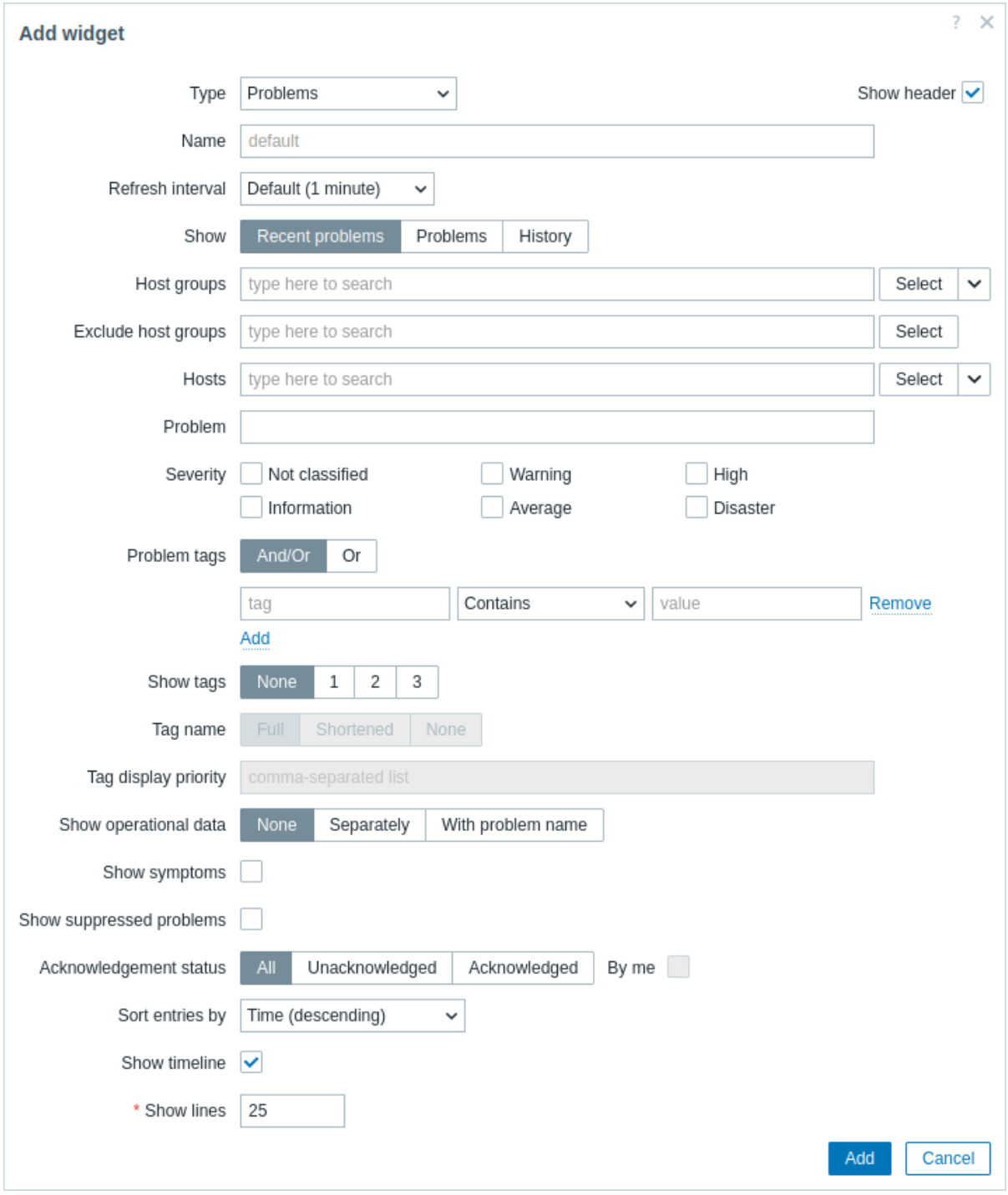
O widget *Problemas* exibe os problemas atuais filtrados por diferentes parâmetros (grupos de hosts, hosts, nomes de problemas, etc.), proporcionando uma visão clara do que precisa de atenção.



Ele exibe os mesmos dados de *Monitoramento* > *Problemas* e pode exibir até 1000 registros.

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 host, host, tag de evento, status de reconhecimento, etc.

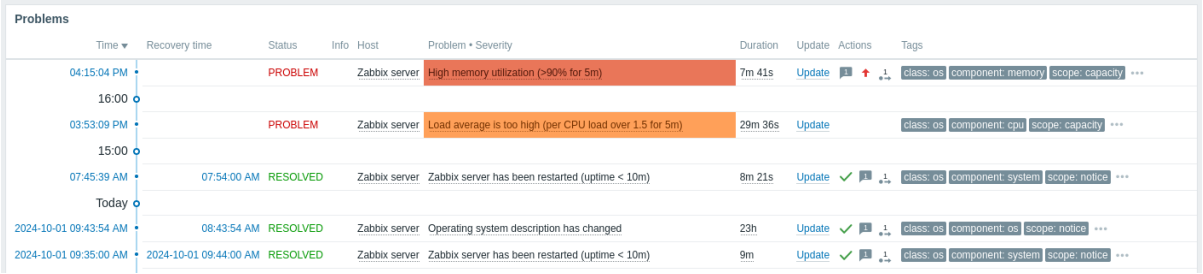
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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 host pai seleciona implicitamente todos os grupos de hosts aninhados. Problemas desses grupos de hosts serão exibidos no widget; se nenhum grupo de host 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 host 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 aqueles 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. 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 de tag 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 de tag 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. 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).
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.
Mostrar linhas	Especifique o número de linhas de problemas a serem exibidas.

Usando o widget

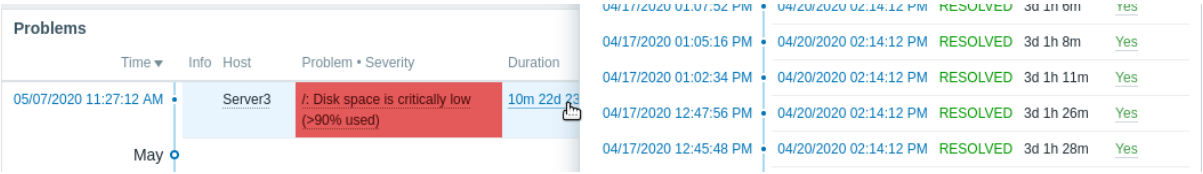


O widget de problemas oferece acesso rápido a informações adicionais:

- Clique na data do problema (na coluna Hora) ou na data de recuperação (na coluna Hora de recuperação) para ver 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 ver 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 apenas se você clicar novamente na duração.

24 Problemas por gravidade

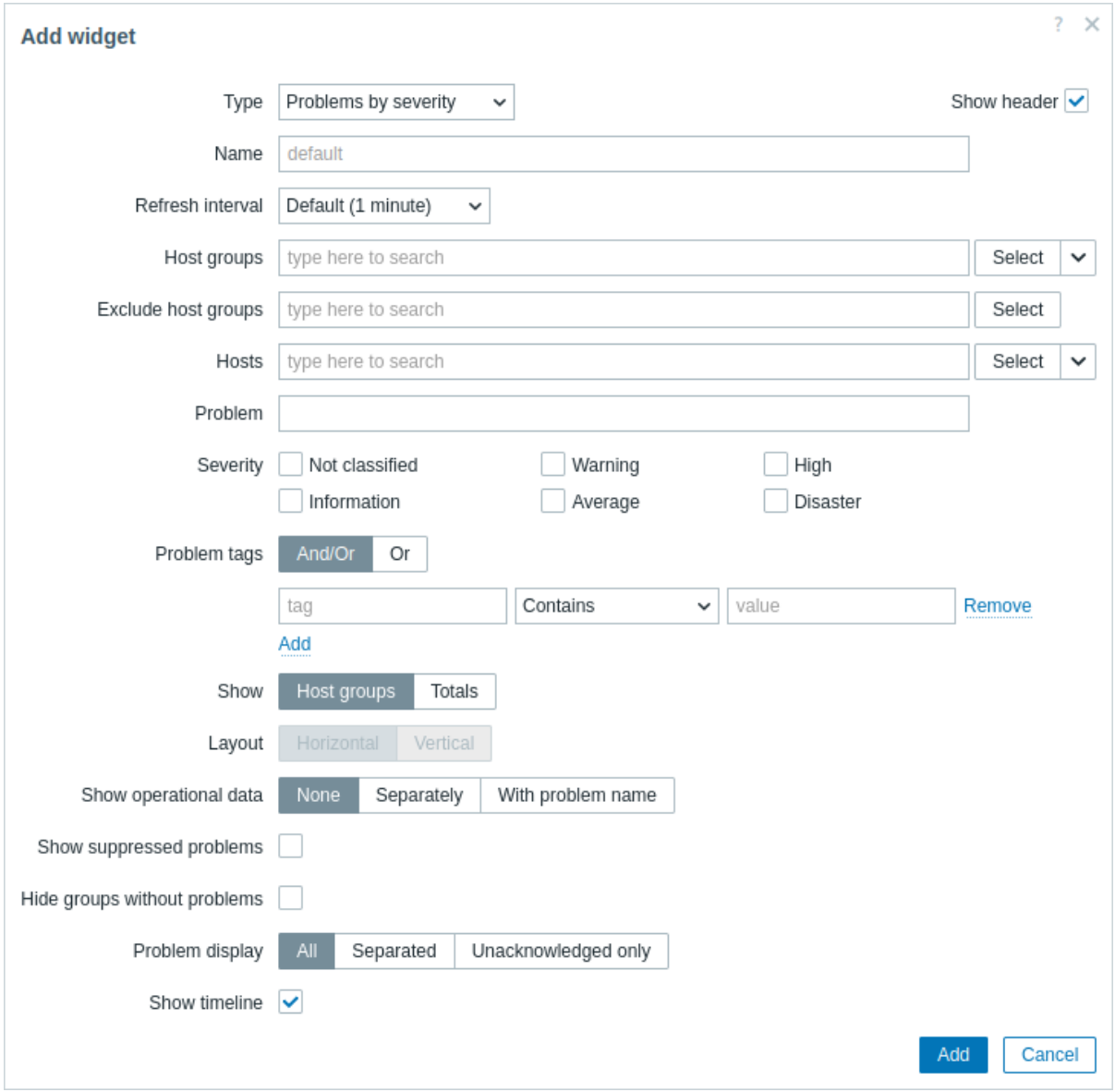
Visão geral

O widget *Problemas por severidade* exibe o número de problemas de causa agrupados por severidade. Ele permite filtrar problemas por diferentes parâmetros (grupos de hosts, hosts, nomes de problemas, etc.).

Ao passar o mouse sobre a contagem de problemas, é exibida a lista de problemas relacionados:

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. 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. 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. 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, 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 os 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	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** - os dados operacionais não serão exibidos **Separadamente** - os dados operacionais serão exibidos como uma linha separada **Com o nome do problema** - os 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 do problema	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.

25 Relatório de SLA

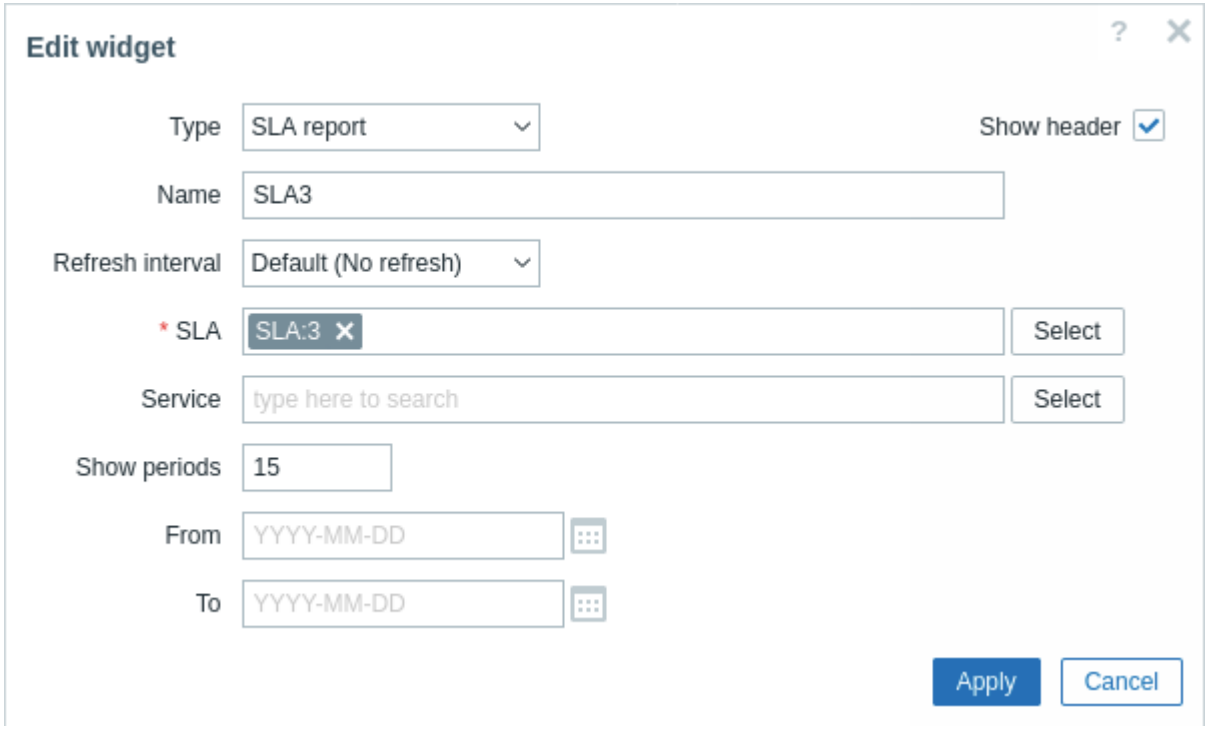
Visão geral

O widget *Relatório de SLA* exibe relatórios de SLA, ajudando você a monitorar o desempenho do serviço em relação às metas definidas.

Ele exibe os mesmos dados que *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.

26 Informações do sistema

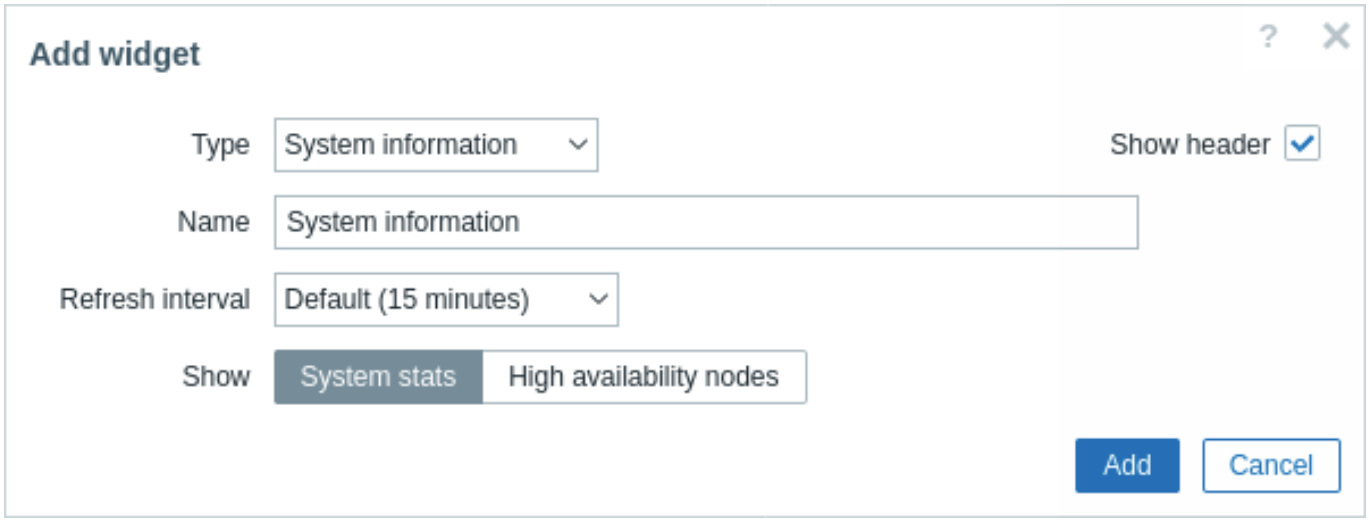
Visão geral

O widget *Informações do sistema* exibe um resumo dos principais dados do sistema e do Zabbix server ou detalhes dos nós de alta disponibilidade.

Ele exibe os mesmos dados de *Relatórios* > *Informações do sistema*.

Configuração

Para configurar, selecione *Informações do sistema* como tipo:



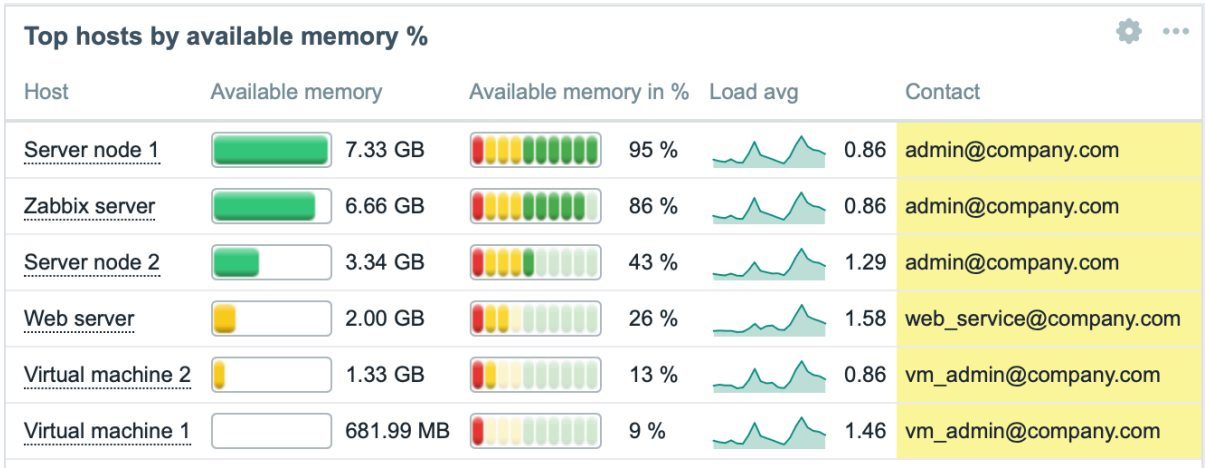
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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*.

27 Principais hosts

Visão geral

O widget *Top hosts* exibe os maiores/menores valores para os items selecionados de múltiplos hosts.



O widget suporta a exibição de até 1000 hosts simultaneamente.

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:

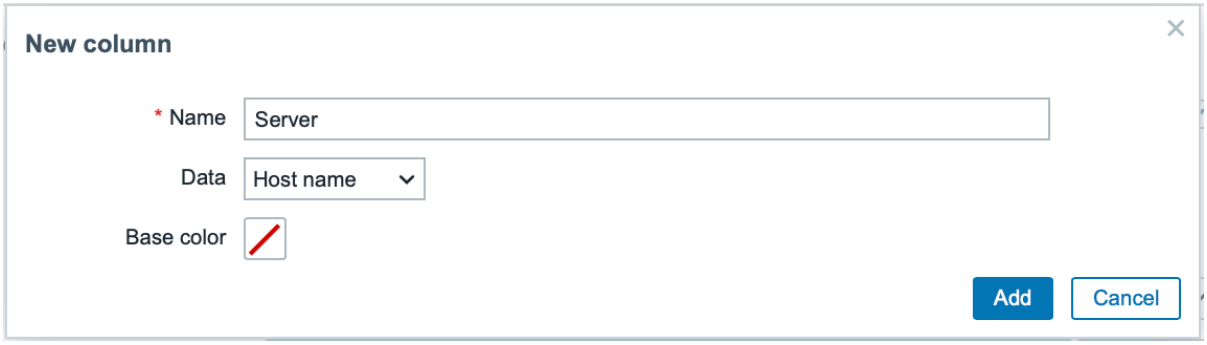
Grupos de hosts	Selecione os grupos de hosts a serem exibidos no widget. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa com os grupos 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 de tag 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 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. 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 exibido 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 ou para baixo pelo ícone antes do nome da coluna.
Ordenar por	Especifique a coluna da lista *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). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Configuração de coluna

É possível adicionar colunas de três tipos de dados: nome do host, valor do item ou texto. A lista de parâmetros de coluna disponíveis depende do tipo de dado da coluna e, para o tipo valor do item, do formato do valor.

Coluna Nome do host

A coluna Nome do host é usada para exibir o nome do host.

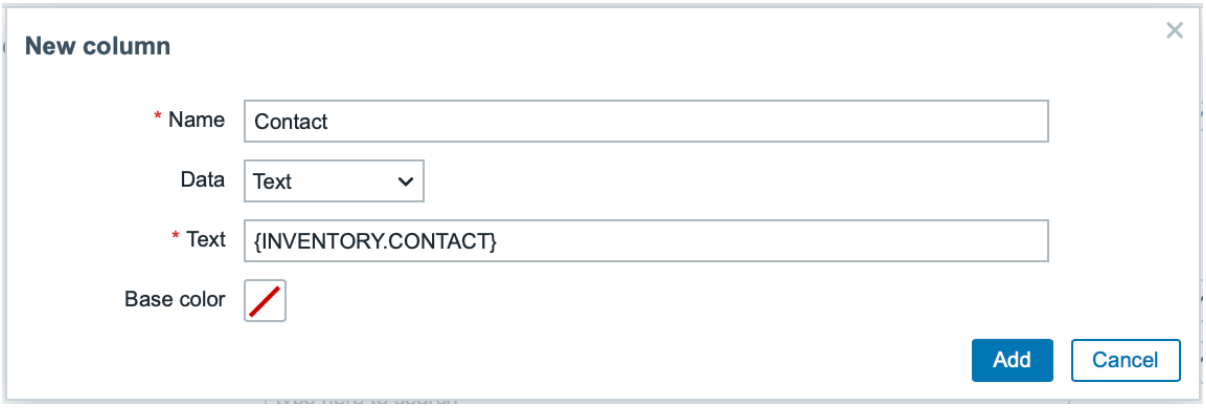


Parâmetros suportados:

Nome	Nome da coluna.
Dados	Tipo de dado a ser exibido na coluna, selecione **Nome do host**.
Cor base	Cor de fundo da coluna.

Coluna de texto

A coluna de texto é usada para exibir qualquer string de texto especificada.

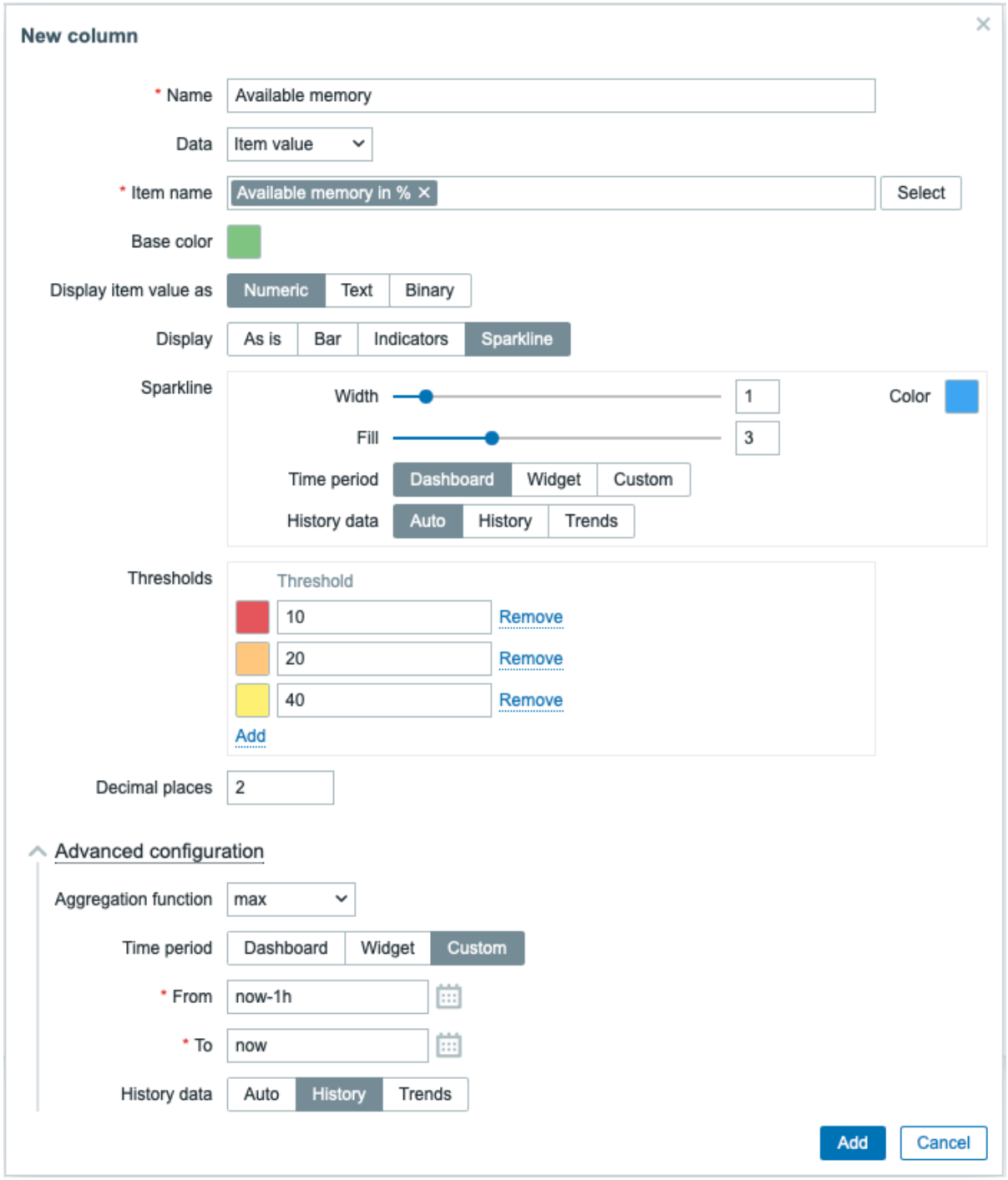


Parâmetros suportados:

Nome	Nome da coluna.
Dados	Tipo de dado a ser exibido na coluna, selecione **Texto**.
Cor base	Cor de fundo da coluna.
Texto	Digite a string a ser exibida. Pode conter macros de host e inventário macros.

Coluna de valor do item

A coluna de valor do item é usada para exibir o valor do item especificado; ela suporta vários formatos e opções de exibição de valor.



Parâmetros suportados:

Nome	Nome da coluna.
Dados	Tipo de dado a ser exibido na coluna, selecione **Valor do item**.
Nome do 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.
Cor base	Cor de fundo da coluna; cor de preenchimento se o dado *Valor do item* for exibido como barra/indicadores. A cor padrão pode ser substituída por uma cor personalizada, se o valor do item ultrapassar um dos limites especificados.
Exibir valor do item como	Formato para exibir o valor do item: **Numérico**, **Texto** ou **Binário**. A opção selecionada determina quais parâmetros adicionais estarão disponíveis. Consulte a lista de parâmetros para cada formato abaixo.
Configuração avançada	

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. Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo. Este parâmetro não está disponível se *Função de agregação* estiver definido como "não usado".
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".
Dados do histórico	Obter dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obter dados do histórico; **Tendências** - obter dados das tendências. Este parâmetro está disponível apenas para valores numéricos de itens. Para valores de texto e binários, os dados sempre serão obtidos do histórico.

Parâmetros de **valor numérico do item**:

Exibir	Defina como o valor deve ser exibido: **Como está** - como texto normal; **Barra** - como barra sólida e colorida; **Indicadores** - como barra segmentada e colorida; **Sparkline** - mini gráfico de linha.
Limiares	Especifique os valores de limite quando a cor de fundo/preenchimento deve mudar. A lista será classificada em ordem crescente ao salvar. Para sparklines, os limites são aplicados apenas ao último valor do item.
Casas decimais	Especifique quantas casas decimais serão exibidas com o valor.
Barra/Indicadores	
Mín	Valor mínimo.
Máx	Valor máximo.
Sparkline	
Largura	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Cor	Selecione a cor da linha e do preenchimento.
Preenchimento	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.

Período de tempo	Especifique o período de tempo para os valores a serem incluídos no gráfico sparkline. Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
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".
Dados do histórico	Obter dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obter dados do histórico; **Tendências** - obter dados das tendências.

Parâmetros de **valor de item de texto**:

Destaques	Especifique as expressões regulares nas quais, ao corresponder, a cor de fundo/preenchimento deve mudar.

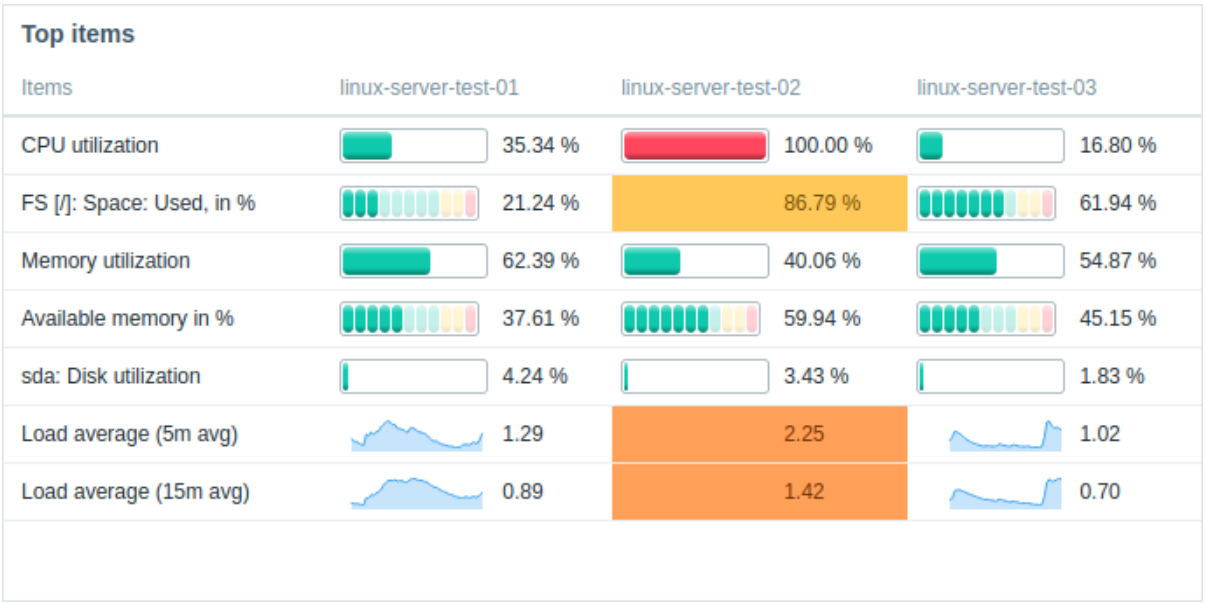
Parâmetros de **valor de item binário**:

Exibir miniatura	Especifique se deve criar e exibir uma miniatura para a imagem contendo dados binários ou exibir um hiperlink *Exibir* levando à imagem em tamanho real na coluna de valor.

28 Principais items

Visão geral

O widget *Top items* exibe os maiores/menores valores para os items selecionados, fornecendo uma visão rápida do seu desempenho.

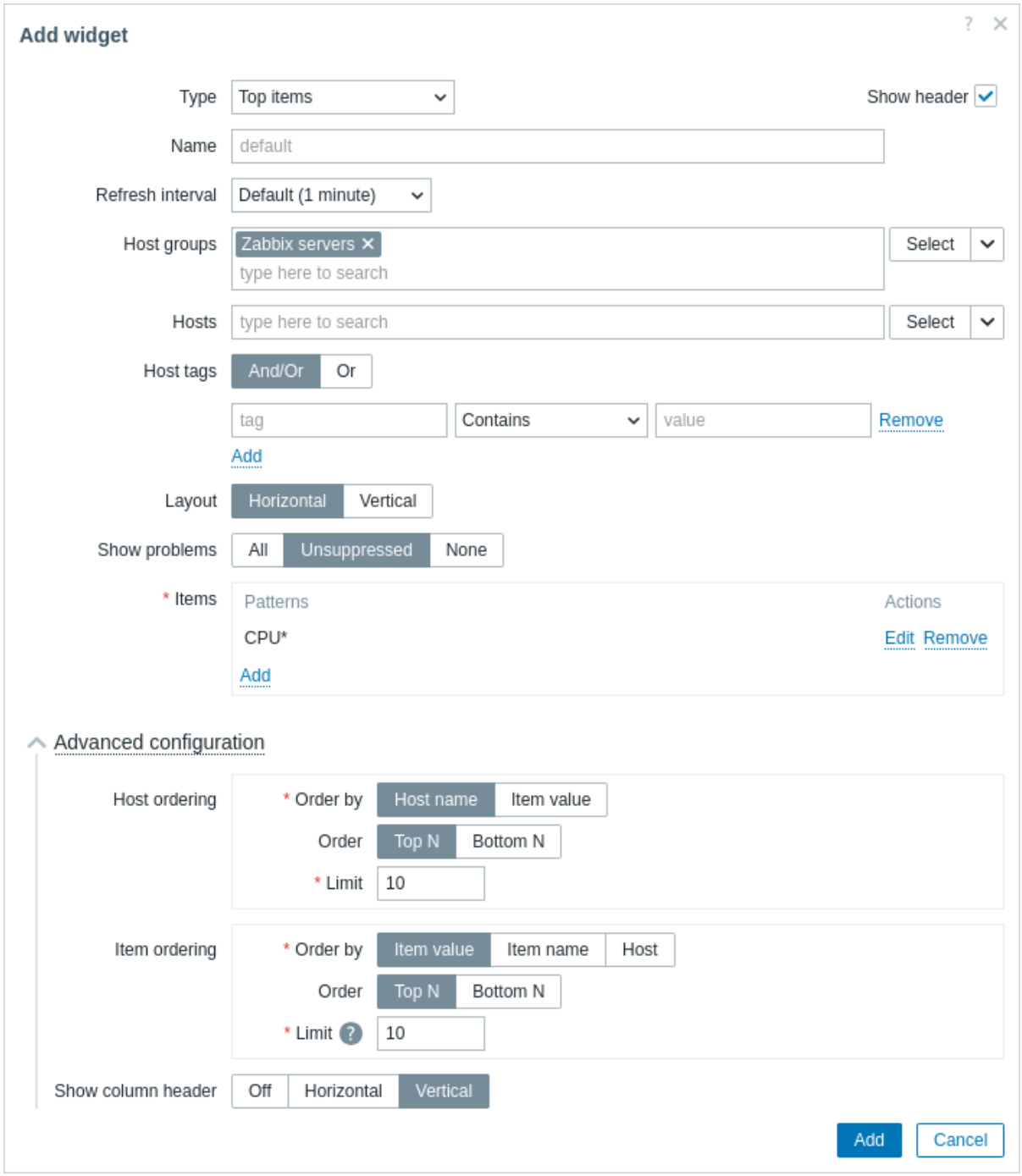


Por padrão, o widget exibe valores das últimas 24 horas. Esse período pode ser alterado em *Administração* > *Geral* > *GUI* (*Período máximo de exibição do histórico*).

Clicar no valor do item abre o menu do item.

Configuração

Para configurar, selecione *Top items* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

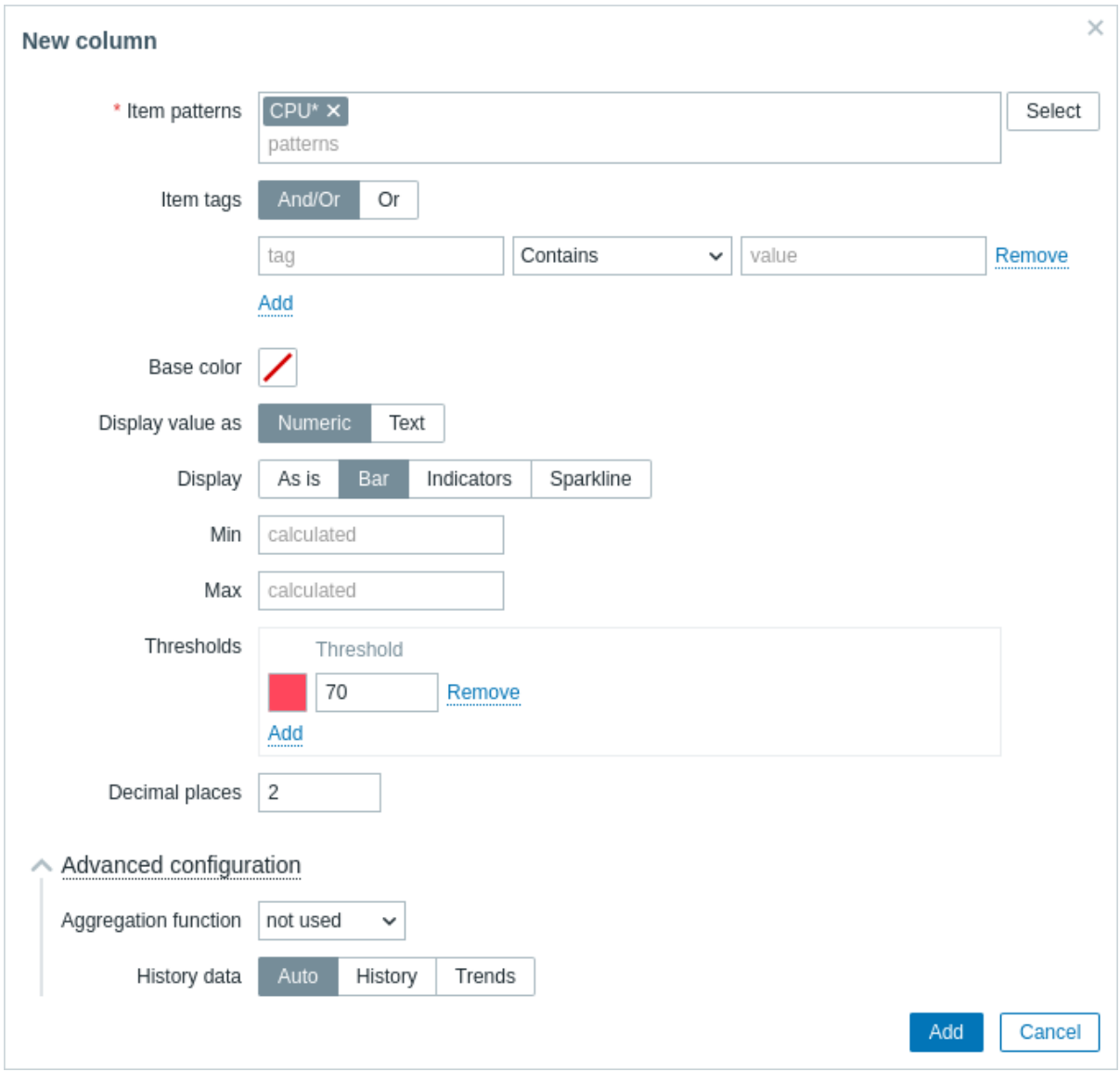
Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Hosts	Selecione hosts. 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. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tags de host	Especifique tags para limitar a quantidade de dados de host 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.
Layout	Selecione a opção de layout: **Horizontal** - os nomes dos hosts serão exibidos à esquerda; **Vertical** - os nomes dos hosts serão exibidos no topo.
Exibir problemas	Filtre quais problemas exibir com base em seu status (todos, não suprimidos, nenhum). Os problemas são indicados com valores de item coloridos, substituindo as configurações de exibição da coluna do item. As cores são baseadas na severidade do problema, que pode ser personalizada em *Administração* > *Geral* > *Opções de exibição de trigger*. As severidades dos problemas podem ser alteradas ao atualizar problemas.
Itens	Adicione padrões de item ou itens específicos para exibição (veja Configuração de coluna).
Configuração avançada	
Ordenação de host	Selecione opções de ordenação para a coluna/linha de host.
Ordenar por	Ordenar hosts por: **Nome do host** - os hosts serão ordenados pelo nome do host; **Valor do item** - os hosts serão ordenados pelo valor dos itens selecionados.
Ordem	Selecione se deseja exibir os maiores ou menores valores: **TopN** - os N maiores valores; **BottomN** - os N menores valores. O valor de N é selecionado no campo *Limite*.
Limite	Informe o limite para hosts exibíveis (intervalo 1-1000; padrão 10). Este valor será o valor de N no campo *Ordem*.
Ordenação de item	Selecione opções de ordenação para a coluna/linha de item.
Ordenar por	Ordenar itens por: **Valor do item** - os itens serão ordenados pelo valor do item; **Nome do item** - os itens serão ordenados pelo nome do item; **Host** - os itens serão ordenados pelo padrão de host selecionado.
Ordem	Selecione se deseja exibir os maiores ou menores valores: **TopN** - os N maiores valores; **BottomN** - os N menores valores. O valor de N é selecionado no campo *Limite*.
Limite	Informe o limite para itens exibíveis (intervalo 1-1000; padrão 10). Este valor será o valor de N no campo *Ordem*.

Exibir cabeçalho da coluna	Selecione as opções de exibição do cabeçalho da coluna: **Desligado** - não exibe o cabeçalho da coluna; **Horizontal** - exibe o texto horizontalmente no cabeçalho; **Vertical** - exibe o texto verticalmente no cabeçalho.

Configuração de coluna

Para configurar as colunas de item, clique em *Adicionar* no parâmetro *Itens*:



Parâmetros comuns de coluna:

Padrões de item	Especifique um ou vários padrões de item, usando o caractere curinga. Alternativamente, selecione os itens. Ao configurar o widget em um dashboard de template, apenas itens configurados no template podem ser selecionados.

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 tag 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 tag especificados; **Igual** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tag 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 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.
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 sobrescrita por cores de limite ou de destaque.
Exibir valor como	Formato para exibir o valor do item: **Numérico** ou **Texto**. A opção selecionada determina quais parâmetros adicionais estarão disponíveis. Consulte a lista de parâmetros para cada formato abaixo.

Parâmetros de coluna para valores numéricos:

Exibir	Defina como o valor deve ser exibido: **Como está** - como texto normal; **Barra** - como barra sólida preenchida com cor; **Indicadores** - como barra segmentada preenchida com cor; **Sparkline** - mini gráfico de linha.
Mín	Valor mínimo para exibição de barra/indicadores.
Máx	Valor máximo para exibição de barra/indicadores.
Largura	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10. Este parâmetro é apenas para exibição de sparkline.
Preenchimento	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10. Este parâmetro é apenas para exibição de sparkline.
Cor	Selecione a cor da linha e do preenchimento. Este parâmetro é apenas para exibição de sparkline.
Veja *Configuração avançada* para descrição dos campos relacionados ao período de tempo e seleção de dados de histórico.	
Limiares	Especifique os valores de limiar quando a cor de fundo/preenchimento deve mudar. A lista será classificada em ordem crescente ao salvar. Para sparklines, os limiares são aplicados apenas ao último valor do item.
Casas decimais ***Configuração avançada***	Especifique quantas casas decimais serão exibidas com o valor.

Função de agregação	Especifique qual função de agregação usar: **min** - exibir o menor valor; **max** - exibir o maior valor; **avg** - exibir o valor médio; **count** - exibir a contagem de valores; **sum** - exibir a soma dos valores; **first** - exibir o primeiro valor; **last** - exibir o último valor; **não usado** - exibir 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. Selecione a fonte de dados para o período de tempo: **Dashboard** - use o seletor de período de tempo do dashboard; **Widget** - use um widget compatível (definido no parâmetro *Widget*); **Personalizado** - use um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo. Este parâmetro não está disponível se *Função de agregação* estiver definido como ”não usado”.
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”.
Dados de histórico	Obtenha dados do histórico ou de tendências: **Auto** - seleção automática; **Histórico** - obter dados do histórico; **Tendências** - obter dados de tendências. Este parâmetro está disponível apenas para valores numéricos de item. Para valores de texto, os dados sempre serão obtidos do histórico.

Parâmetros de coluna para valores de texto:

Destaques	Especifique as expressões regulares nas quais, ao corresponder, a cor de fundo/preenchimento deve mudar.

29 Principais triggers

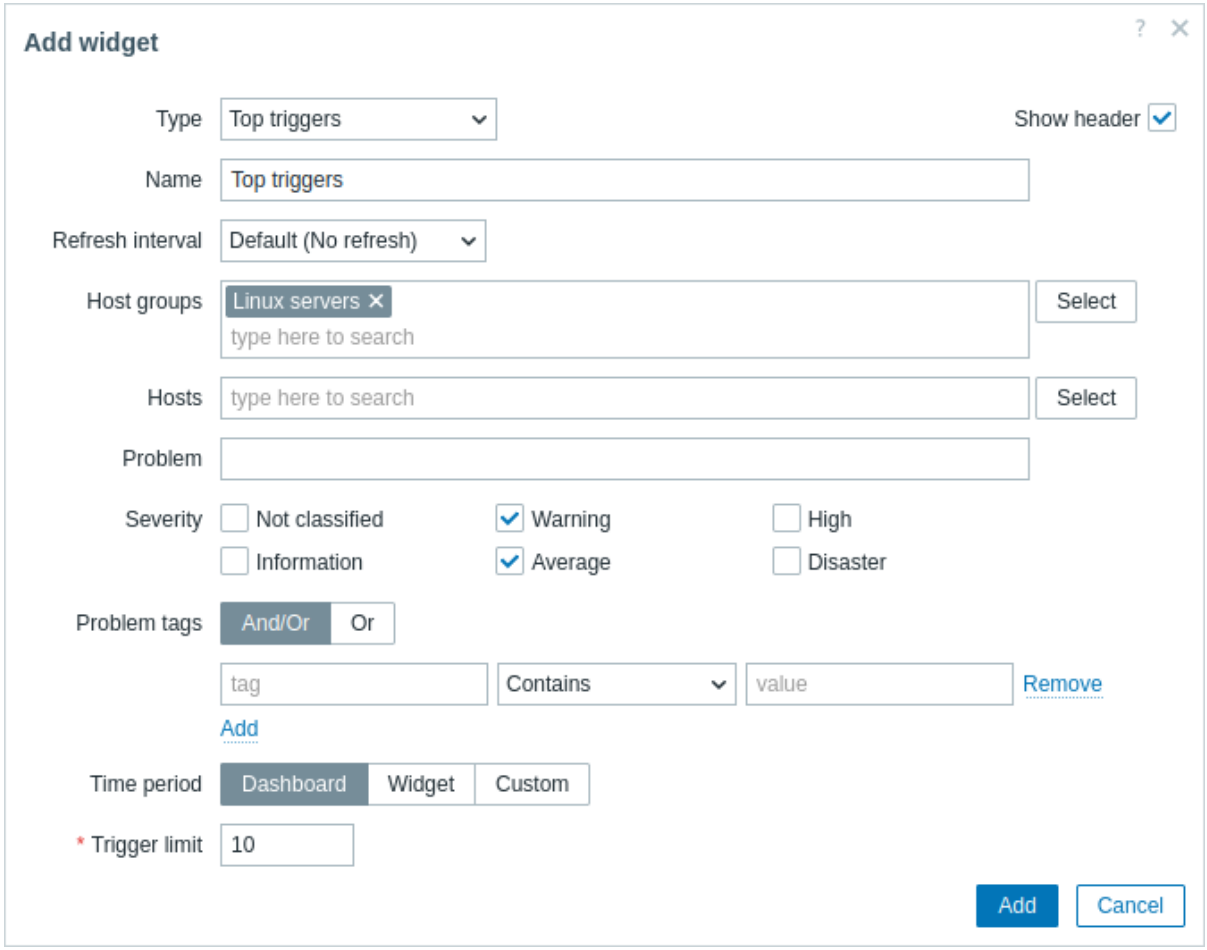
Visão geral

O widget *Top triggers* exibe triggers com o maior número de problemas.

Ele exibe os mesmos dados que *Relatórios > Top 100 triggers* e pode exibir até 1000 registros.

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 autocompletável, 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 autocompletável, 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. 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.
Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - usa o seletor de período de tempo do dashboard; **Widget** - usa um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usa um período de tempo personalizado definido nos parâmetros *De* e *Até*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse. Observe que, independentemente da configuração de *Período de tempo* do widget, widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
Widget	Insira 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	Insira 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	Insira ou selecione o fim 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".
Limite de triggers	Defina o número de triggers a serem exibidos. Faixa de valores possível: 1-1000.

30 Visão geral do trigger

Visão geral

O widget *Visão geral dos triggers* exibe os estados atuais dos triggers dos hosts como uma tabela de blocos coloridos. Ele fornece uma visão rápida e visual da integridade e atividade dos triggers em diferentes hosts.



O widget utiliza vários elementos visuais para transmitir detalhes dos triggers:

- As cores dos blocos para triggers em PROBLEMA correspondem às severidades dos problemas.
- Blocos piscando indicam mudanças recentes no estado do trigger (nos últimos 2 minutos).
- Setas cinzas indicam triggers com dependências (passe o mouse sobre elas para ver detalhes).
- Ícones de caixa de seleção indicam problemas reconhecidos; todos os problemas ativos ou resolvidos para o trigger devem ser reconhecidos para que este ícone apareça.

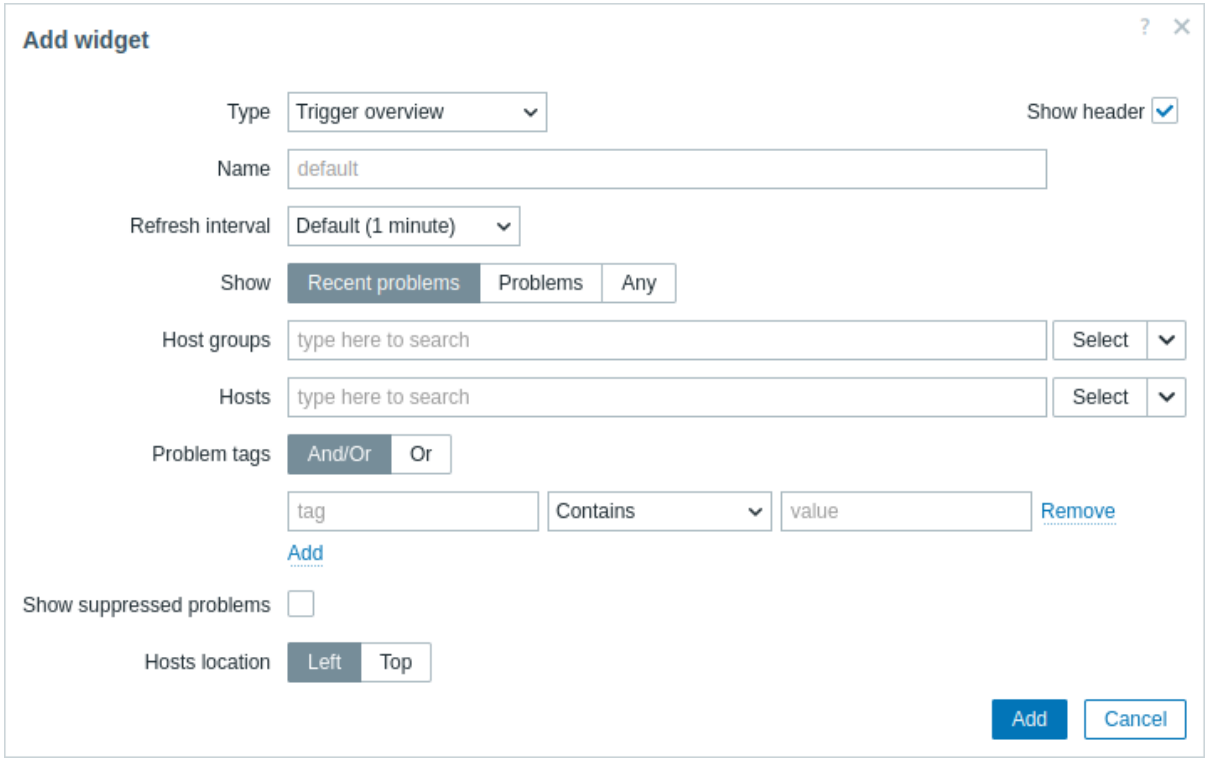
Note:
As cores de severidade dos triggers podem ser ajustadas em *Administração* > *Geral*. As severidades dos triggers podem ser modificadas ao atualizar problemas.

Clicar em um bloco fornece links para informações relacionadas ao trigger (problemas, configuração do trigger, etc.).

Por padrão, até 50 registros são exibidos (configurável em *Administração* > *Geral* > *GUI* em *Número máximo de colunas e linhas nas tabelas de visão geral*). Se existirem mais triggers, uma mensagem solicita que você refine os critérios de filtragem. Esse limite é aplicado antes de qualquer filtragem adicional (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	Filtra 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 auto-completável, 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 auto-completável, 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 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. **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. Há 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.
Mostrar problemas suprimidos	Marque a caixa para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
Layout	Selecione a opção de layout: **Horizontal** - os nomes dos hosts serão exibidos à esquerda; **Vertical** - os nomes dos hosts serão exibidos no topo.

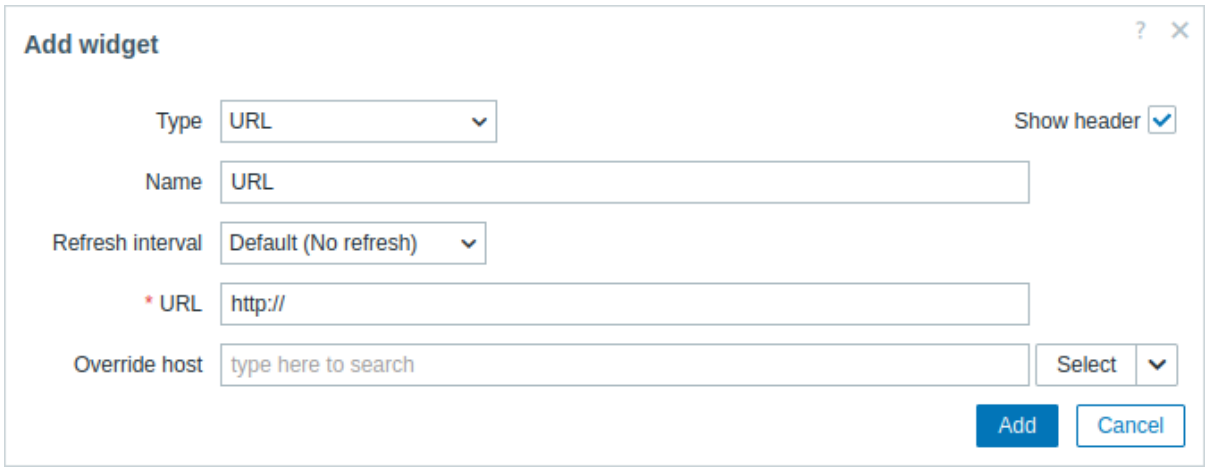
31 URL

Visão geral

O widget *URL* exibe o conteúdo recuperado de uma URL.

Configuração

Para configurar, selecione *URL* como tipo:



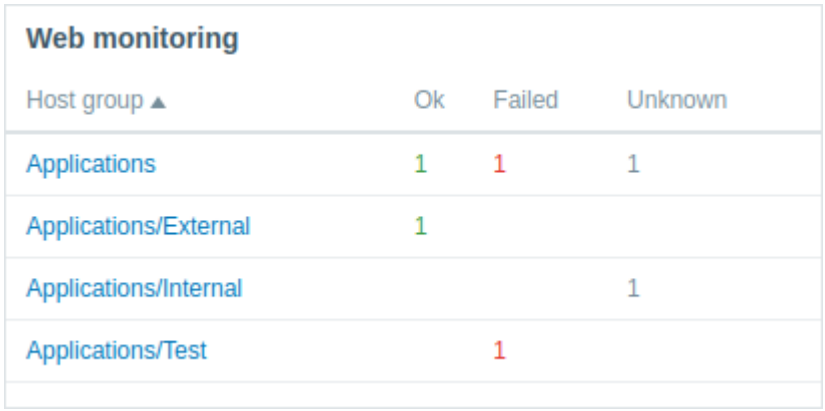
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

URL	Insira 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 seletor de host do 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. Além disso, pode ser necessário modificar o cabeçalho de resposta HTTP X-Frame-Options para exibir conteúdo externo ao frontend do Zabbix.

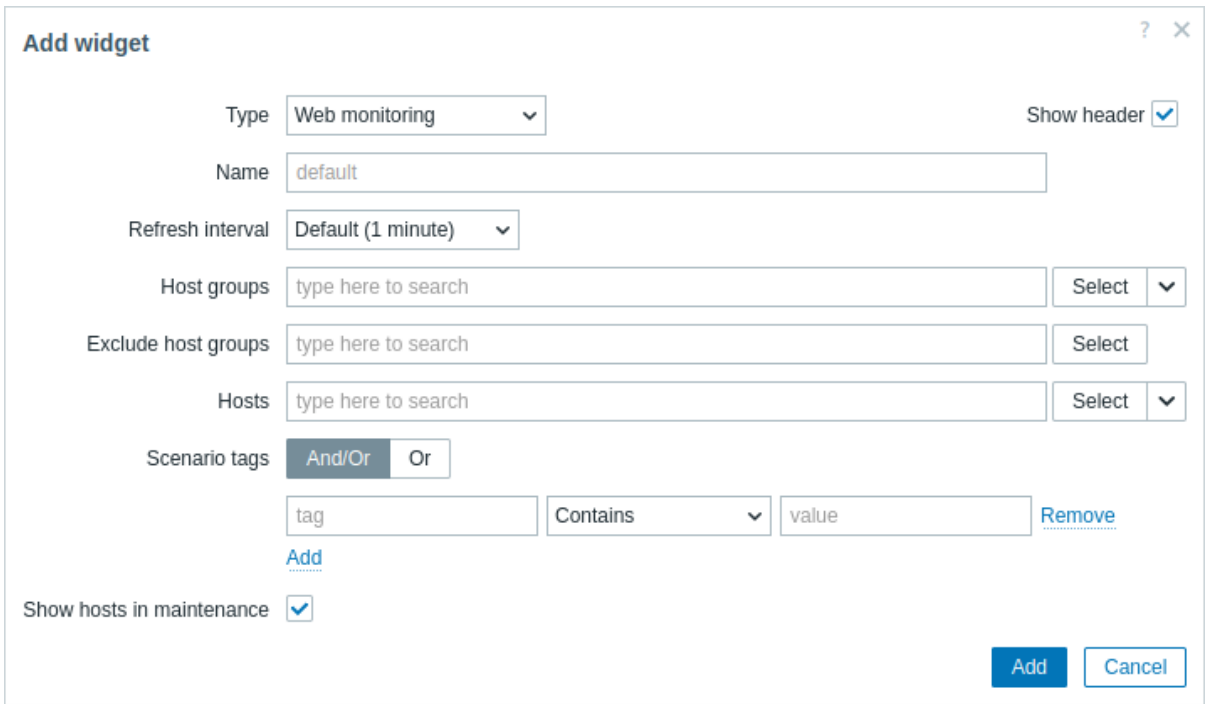
Visão geral

O widget *Monitoramento web* exibe o status dos cenários web ativos.



Clicar no nome de um grupo de hosts abre *Monitoramento* > *Hosts* > *Cenários web*, onde você pode visualizar os detalhes do cenário.

Configuração



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. Alternativamente, 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 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 *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. Alternativamente, 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 cenários	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. 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.
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.

2 Monitoramento

Visão geral

O menu Monitoramento é totalmente voltado para exibição de dados. Quaisquer informações que o Zabbix esteja configurado para coletar, visualizar e agir, serão exibidas 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:

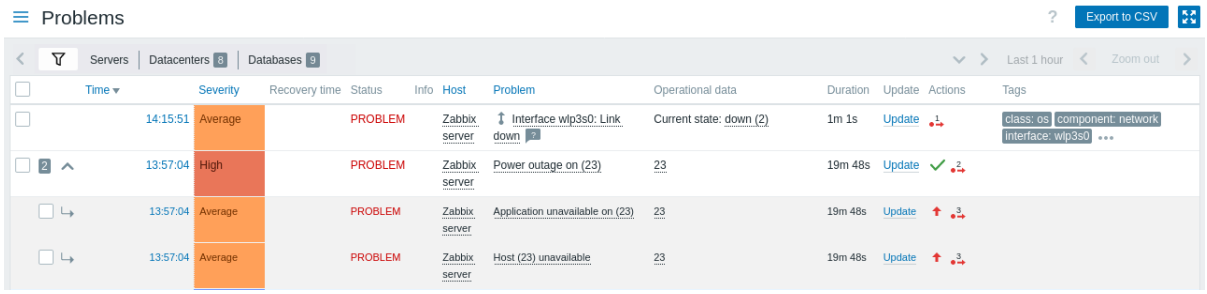
	Exibe a página no modo quiosque. Neste modo, apenas o conteúdo da página é exibido. Para sair do modo quiosque, mova o cursor do mouse até que o botão de saída apareça e clique nele. Você será levado de volta ao modo normal.

1 Problemas

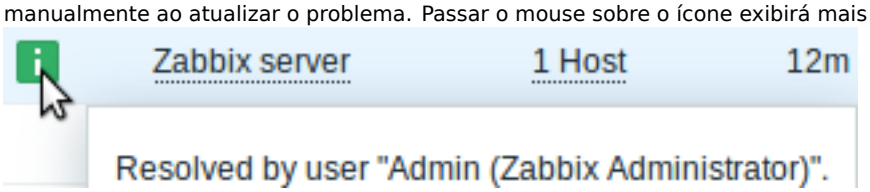
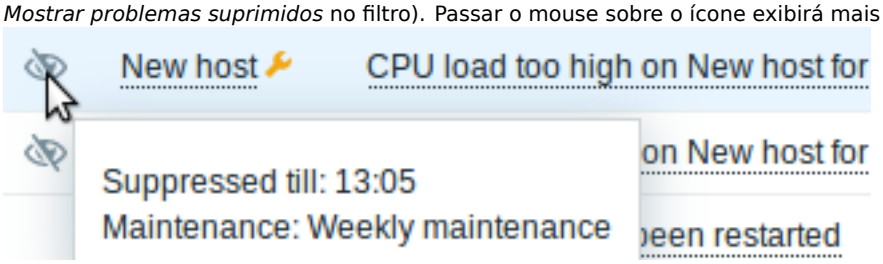
Visão geral

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: 2 - 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, porém, 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*.

Coluna	Descrição
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 de item mais recentes. 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).
Dados operacionais	Dados operacionais são exibidos contendo os valores mais recentes dos itens. Os 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)  - 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, popups com detalhes sobre a atividade são exibidos. Veja visualizando detalhes para saber mais sobre os ícones usados no popup 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 de 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, sendo a 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, fechamento manual por um usuário ou fechamento por correlação).

Observe que fechar o problema manualmente não gera 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 de problema negativa, 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 à ausência de dados recebidos 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 de problema negativa;
- 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 de problema negativa 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 para 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:

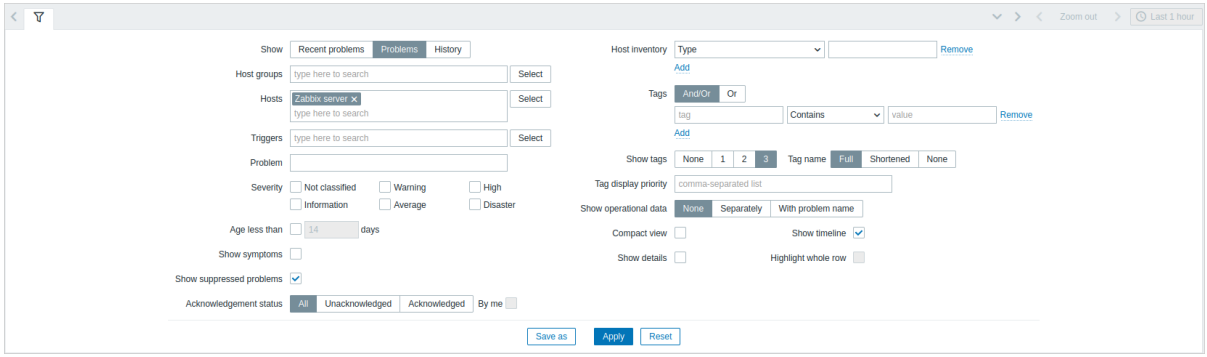
Export to CSV	Exporta o conteúdo de todas as páginas para um arquivo CSV.

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 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.



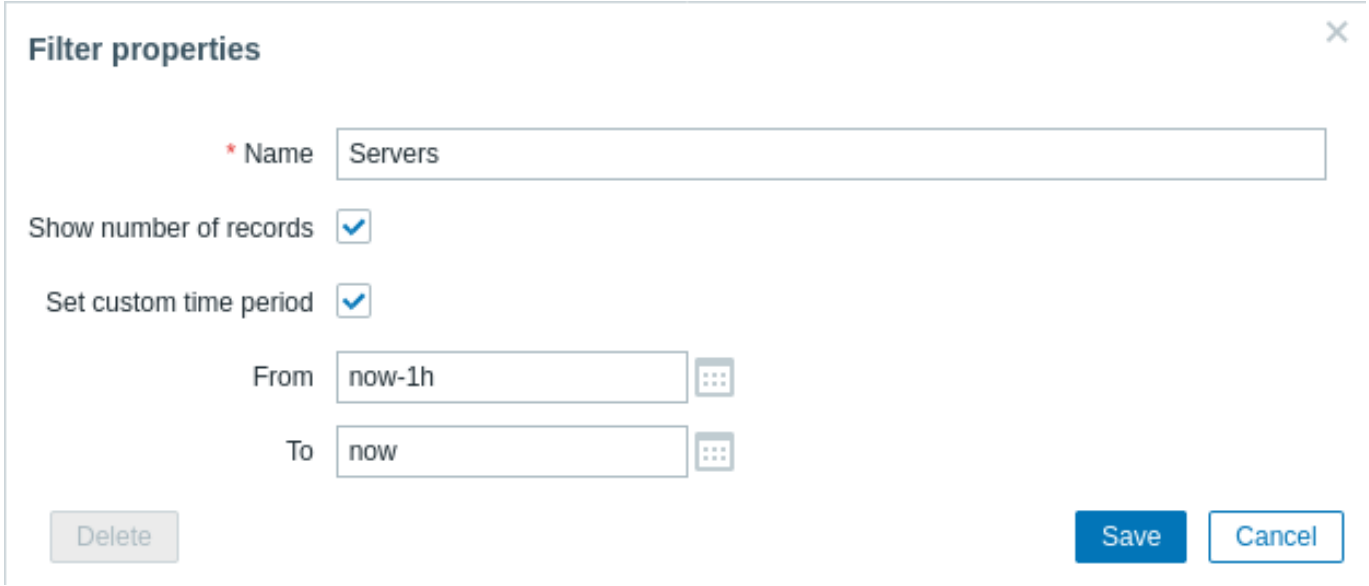
Parâmetro	Descrição
Mostrar	Filtrar pelo status do problema: **Problemas recentes** - problemas não resolvidos e recentemente resolvidos são exibidos (padrão) **Problemas** - apenas problemas não resolvidos são exibidos **Histórico** - o histórico de todos os eventos é exibido
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados.
Hosts	Filtrar por um ou mais hosts.
Triggers	Filtrar por um ou mais triggers.
Problema	Filtrar pelo nome do problema.
Severidade	Filtrar pela severidade do trigger (problema).
Idade menor que	Filtrar por quanto tempo o problema existe.
Mostrar sintomas	Marque a caixa de seleção para exibir em sua própria linha os 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	Filtrar 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.
Inventário do host	Filtrar por tipo e valor do inventário.
Tags	Filtrar por nome e valor da tag do evento. 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. Há 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, insensível 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, 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 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).
Mostrar 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

Parâmetro	Descrição
Prioridade de exibição da tag	Digite a prioridade de exibição da tag para um problema, como uma lista separada por vírgulas de tags (por exemplo: `cliente,escopo,componente`). Devem ser usados apenas nomes de tags, sem valores. As tags desta lista serão sempre 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
Visualização compacta	Marque a caixa de seleção para ativar a visualização compacta.
Mostrar detalhes	Marque a caixa de seleção para exibir as expressões dos triggers subjacentes dos problemas. Desativado se a caixa de seleção *Visualização compacta* estiver marcada.
Mostrar linha do tempo	Marque a caixa de seleção para exibir a linha do tempo visual e o agrupamento. Desativado se a caixa de seleção *Visualização compacta* estiver marcada.
Destacar linha inteira	Marque a caixa de seleção para destacar toda a linha para problemas não resolvidos. A cor da severidade do problema é usada para o destaque. Ativado apenas se a caixa de seleção *Visualização compacta* estiver marcada nos temas padrão azul e escuro. *Destacar linha inteira* não está disponível nos temas de alto contraste.

Abas para filtros favoritos

Conjuntos de parâmetros de filtro frequentemente usados podem ser salvos em abas.

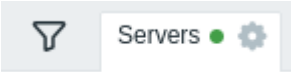
Para salvar um novo conjunto de parâmetros de filtro, abra a aba principal e configure as configurações do filtro, depois pressione o botão *Salvar como*. Em uma nova janela pop-up, defina as *Propriedades do filtro*.



Parâmetro	Descrição
Nome	O nome do filtro a ser exibido na lista de abas.
Mostrar número de registros	Marque se quiser 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 de tempo absoluto (`Y-m-d H:i:s`) ou relativo (`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 ativado.

Para editar as propriedades do filtro de um filtro existente, pressione o símbolo de engrenagem ao lado do nome da aba ativa.



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 filtros salvos. Alternativamente, o botão apontando para baixo abre um menu suspenso com todos os filtros salvos e você pode clicar no que precisar.
- 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.

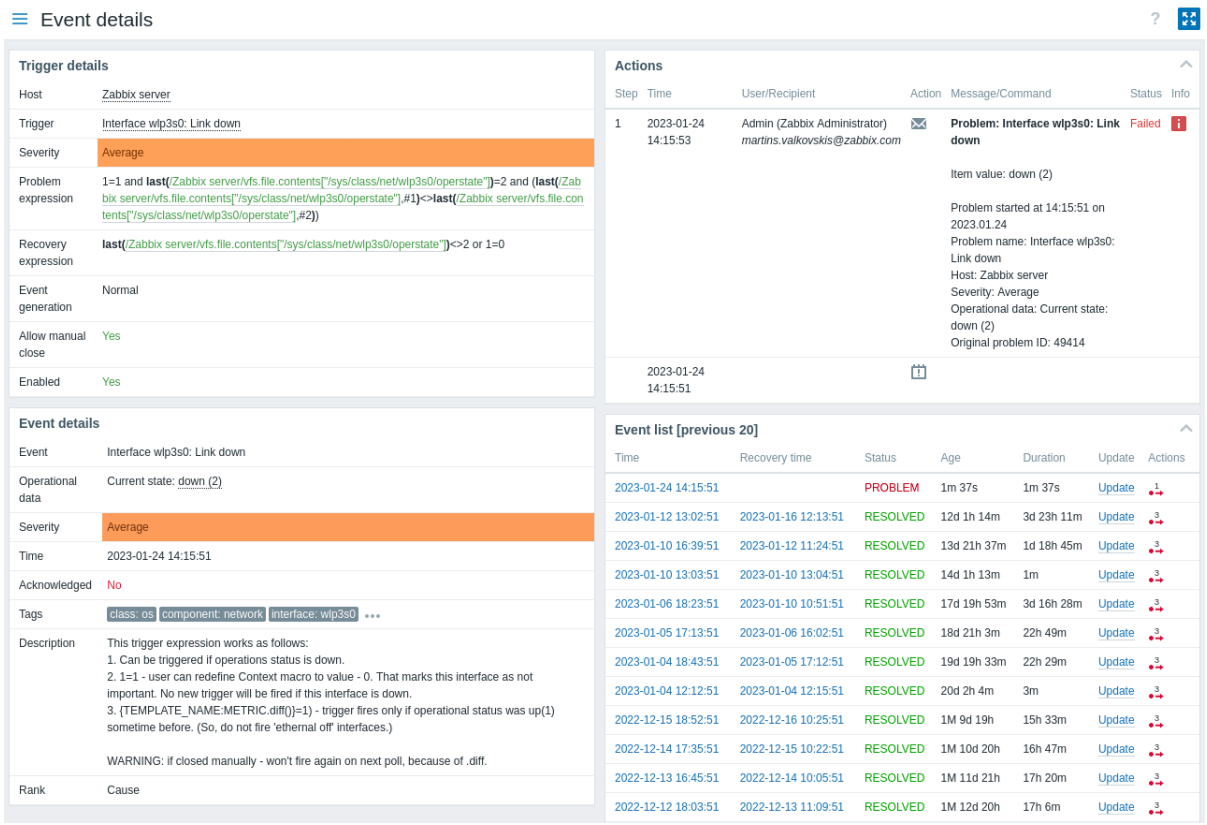
Note:
Para compartilhar filtros, copie e envie para outros a URL de um filtro ativo. Após abrir esta URL, outros usuários poderão salvar este conjunto de parâmetros como um filtro permanente em sua conta Zabbix.
Veja também: Parâmetros de página.

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 de problema - se ela tiver sido atualizada para o evento de 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 reduzida (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. Neste 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.

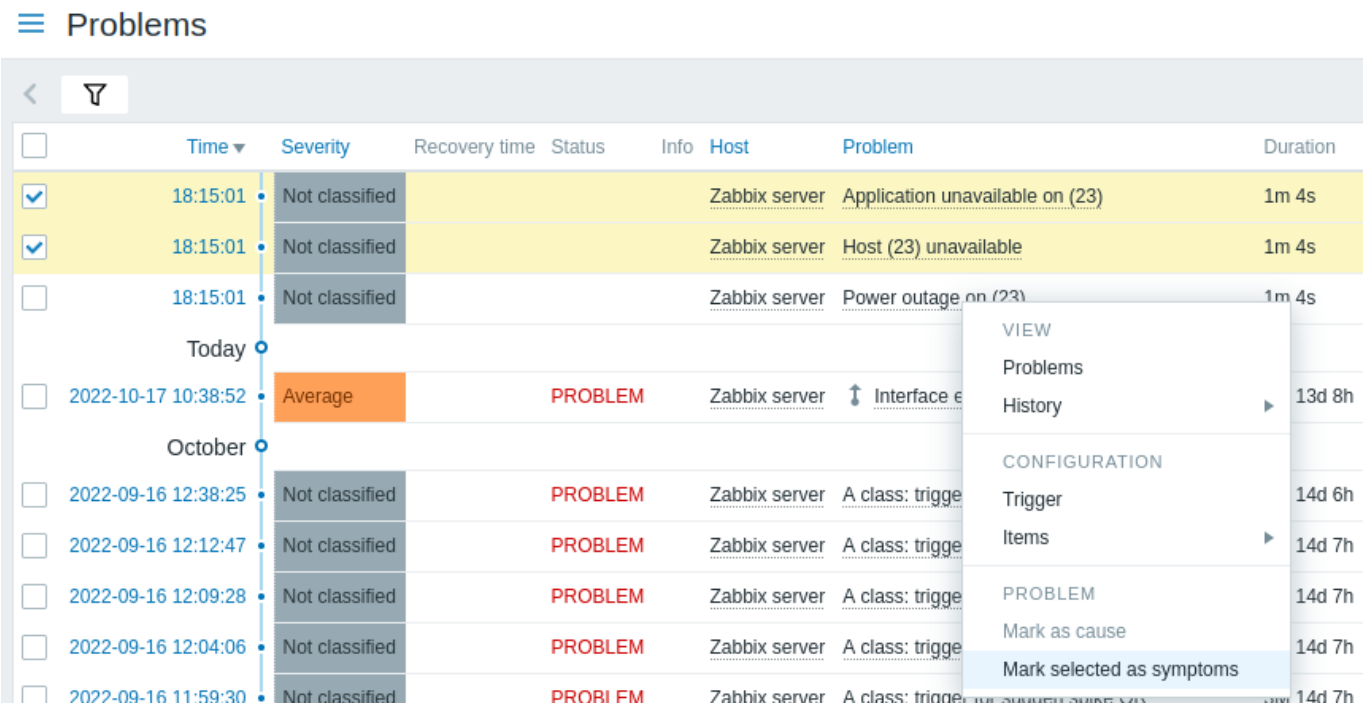
Apenas problemas de causa são contabilizados nos totais de problemas em mapas, widgets do dashboard como *Problemas por severidade* ou *Hosts com problemas*, etc. No entanto, a classificação de problemas não afeta os serviços.

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 sintoma, primeiro selecione-o na lista de problemas. Um ou vários problemas podem ser selecionados.

Em seguida, vá até o problema de causa e, em seu menu de contexto, clique na opção *Marcar selecionados como sintomas*.



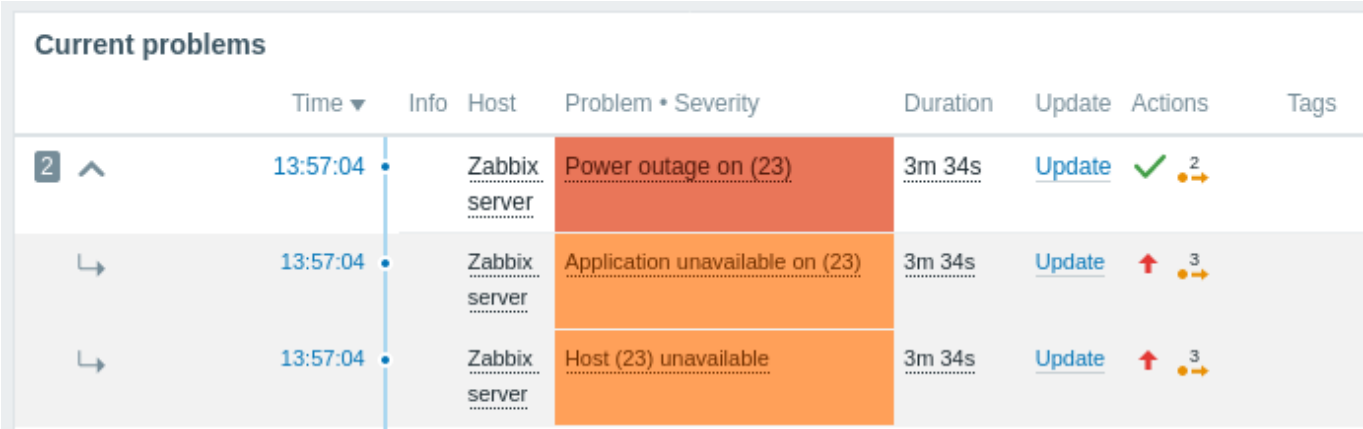
Após isso, os problemas selecionados serão atualizados pelo server para sintomas do problema de causa.

Enquanto o status do problema está sendo atualizado, ele é exibido de uma das duas maneiras:

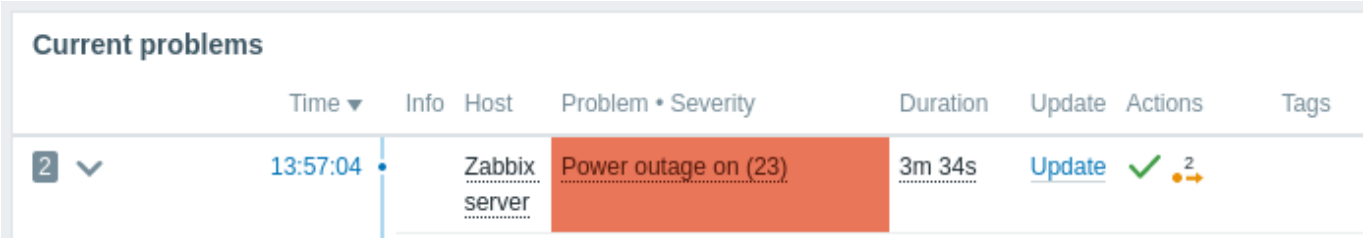
- Um status ”ATUALIZANDO” piscando é exibido na coluna Status;
- Um ícone piscando ou na coluna Info (isto ocorre se apenas *Problemas* estiverem selecionados no filtro e, portanto, a coluna Status não for exibida).

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

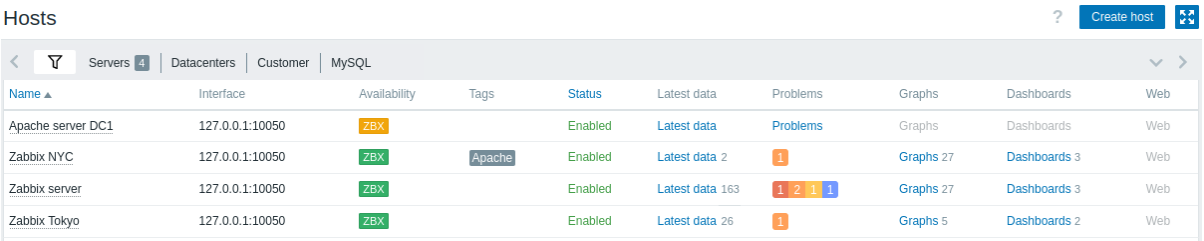
Um problema de sintoma pode ser revertido para um problema de causa. Para fazer isso, você pode:

- clicar na opção *Marcar como causa* no menu de contexto do problema de sintoma;
- marcar a opção *Converter para causa* na tela de atualização de problema e clicar 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 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. **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.
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 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).
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 cinza) e nenhum número é exibido.

Coluna	Descrição
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 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 cinza) e nenhum número é exibido.

Botões

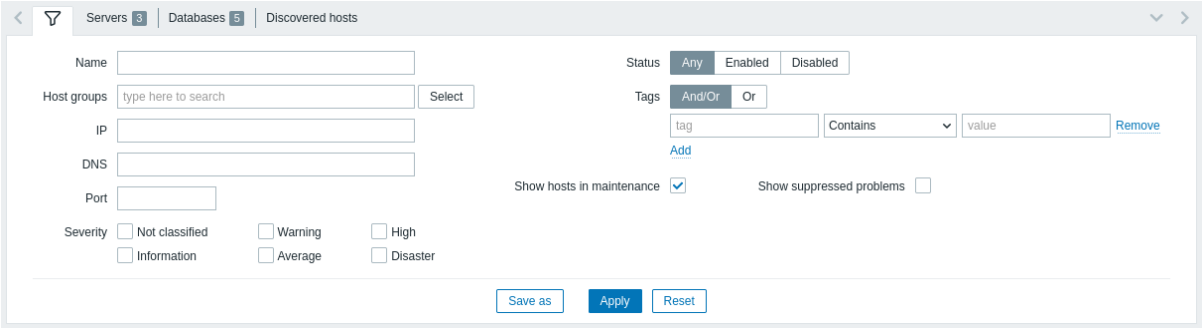
Criar host permite criar um novo host. Este botão está disponível apenas para usuários Admin e Super Admin.

Os botões do modo de visualização, comuns a todas as seções, são descritos na página Monitoramento.

Usando o filtro

Você pode usar o filtro para exibir apenas os hosts que lhe interessam. Para melhor desempenho na busca, 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. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados.
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.
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 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
Exibir hosts em manutenção	Marque a caixa para exibir hosts que estão em manutenção (exibido por padrão).
Exibir problemas suprimidos	Marque a caixa 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.

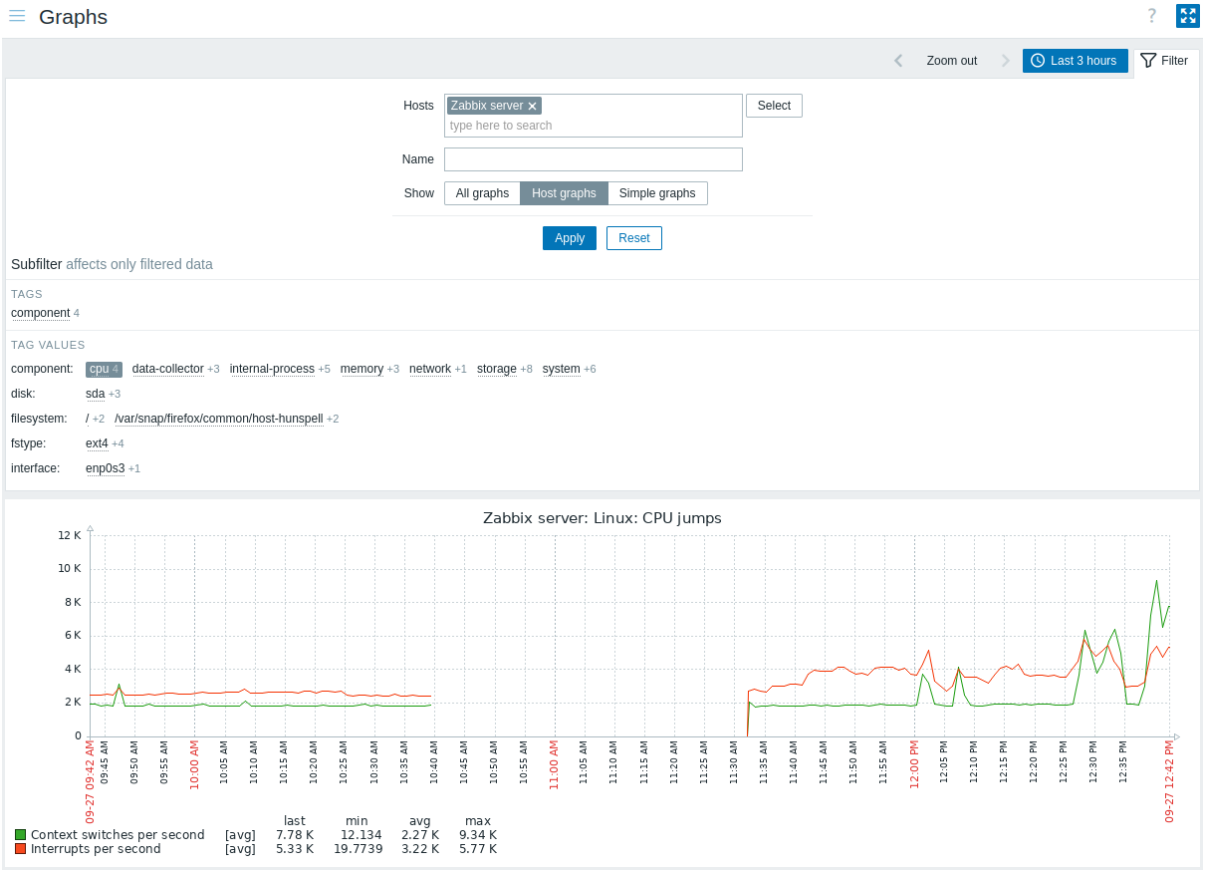
Veja mais detalhes sobre salvar filtros.

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 *Exibir* (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 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 filtrados imediatamente (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 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 três pontos 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 três pontos 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* abaixo do 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.

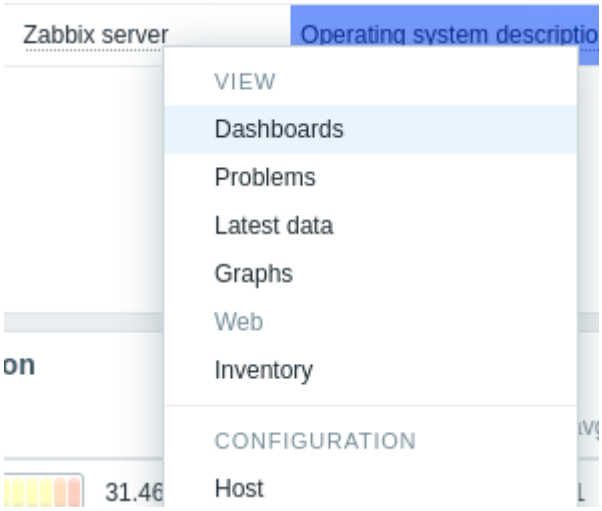
> **Note:**
> Os dashboards de host eram telas de host antes do Zabbix 5.2. Ao importar um template antigo que contenha telas, a importação da tela será ignorada.

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);

- a partir do menu do host clicando em *Dashboards*.

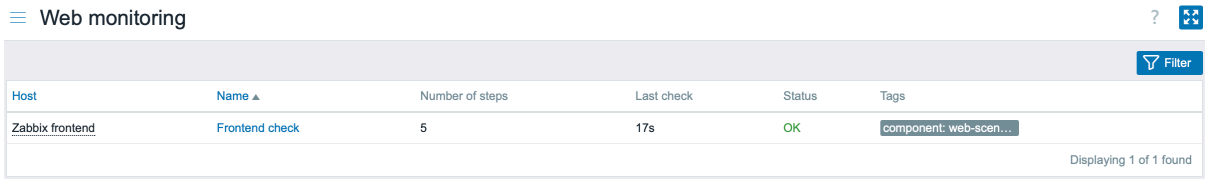


Observe que os dashboards de host *não podem* ser acessados diretamente na seção *Dashboards*, que é reservada para dashboards globais.

3 Cenários web

Visão geral

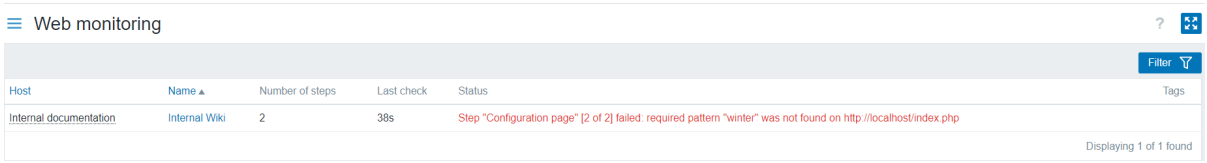
As informações do cenário web do host podem ser acessadas em *Monitoramento* → *Hosts* clicando em Web para o respectivo host.



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 status de um cenário web pode ser:

- OK (verde) - todas as etapas foram bem-sucedidas e o Zabbix coletou dados para o cenário.
- Erro (vermelho) - uma ou mais etapas falharam (veja a mensagem de erro para dicas sobre quais parâmetros podem precisar de correção).
- Desconhecido (cinza) - nenhum status de etapa está disponível ainda ou nenhum dado foi coletado.



O número máximo de cenários exibidos por página depende da configuração de *Linhas por página* no perfil do usuário.

Por padrão, apenas 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. 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 o filtro

A página mostra uma lista de todos os cenários web do host selecionado. Para visualizar cenários web de outro host ou grupo de hosts sem retornar à página *Monitoramento* → *Hosts*, selecione esse host ou grupo no filtro. Você também pode filtrar cenários com base em tags.

Botões

Os botões do modo de visualização, comuns a todas as seções, estão descritos na página Monitoramento.

3 Últimos dados

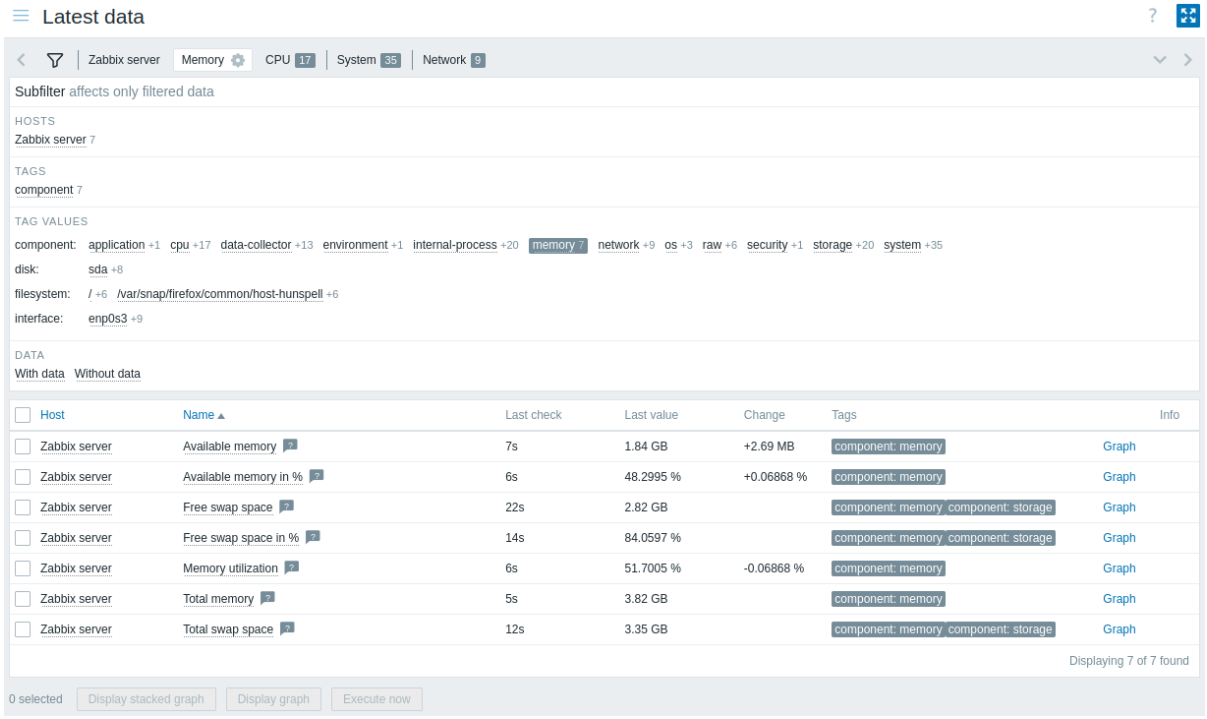
Visão geral

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 é exibido após o nome do host. Se um host estiver desabilitado, o nome do host é 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. 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 items que contenham 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 é 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:

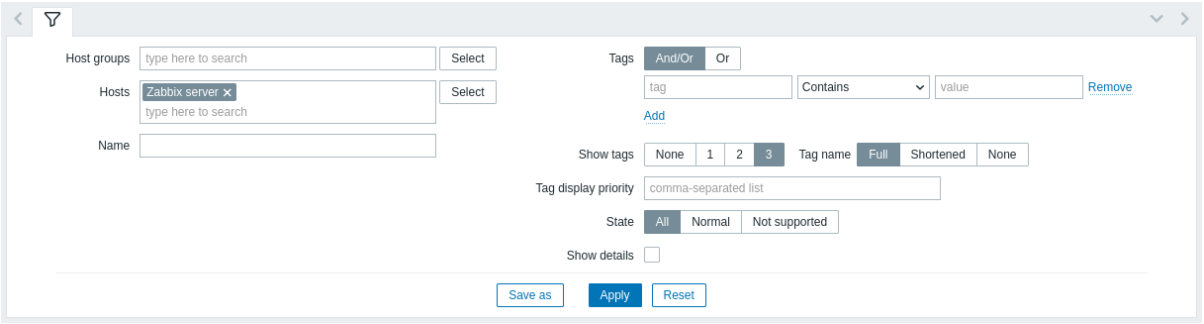
- *Exibir gráfico empilhado* - exibe um gráfico ad-hoc empilhado.
- *Exibir gráfico* - exibe um gráfico ad-hoc simples.
- *Executar agora* - executa uma checagem para novos valores de item imediatamente. Suportado apenas para checagens **passivas** (veja mais detalhes). Esta opção está disponível apenas para hosts com acesso de leitura e escrita. 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 de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone de 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 *Exibir 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

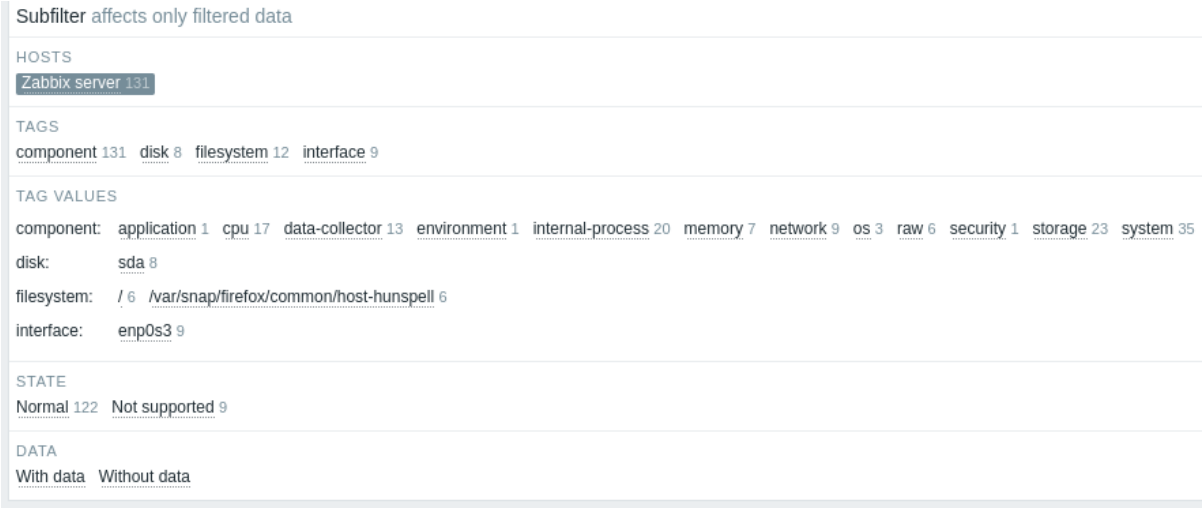
As configurações de filtro favoritas podem ser salvas como abas e, em seguida, acessadas rapidamente clicando na respectiva aba acima do filtro.

Veja mais detalhes sobre salvar filtros.

Usando o subfiltro

O subfiltro permite modificar ainda mais o filtro do 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 solicitação de atualização da tabela para sempre ter informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

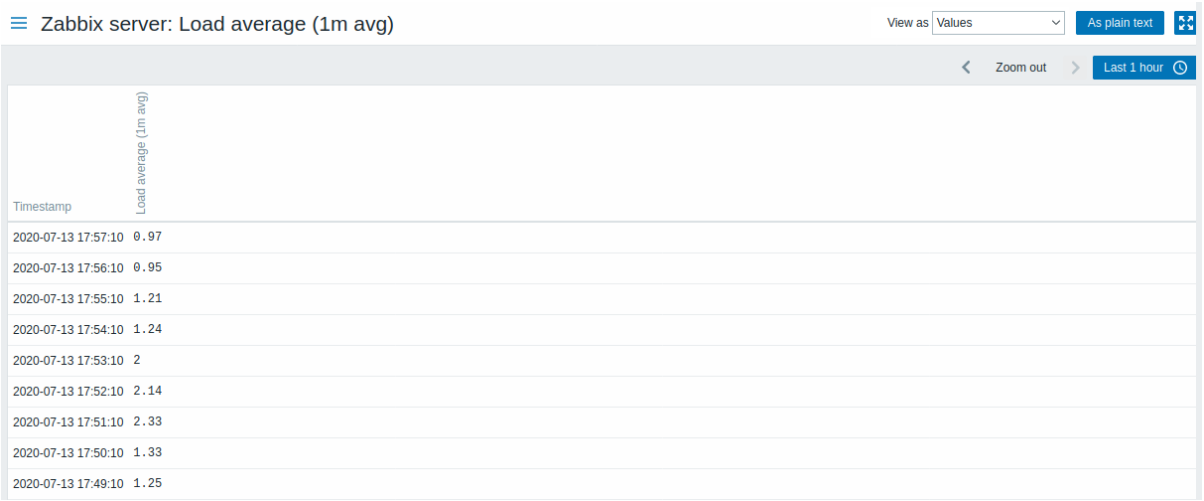
Para cada grupo de entidades (hosts, tags, valores de tags, etc.), até 10 linhas de entidades são exibidas. Se houver mais entidades, essa lista pode ser expandida para exibir no máximo 1000 entradas (o valor de SUBFILTER_VALUES_PER_GROUP em definições do frontend) clicando no ícone de três pontos ••• 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

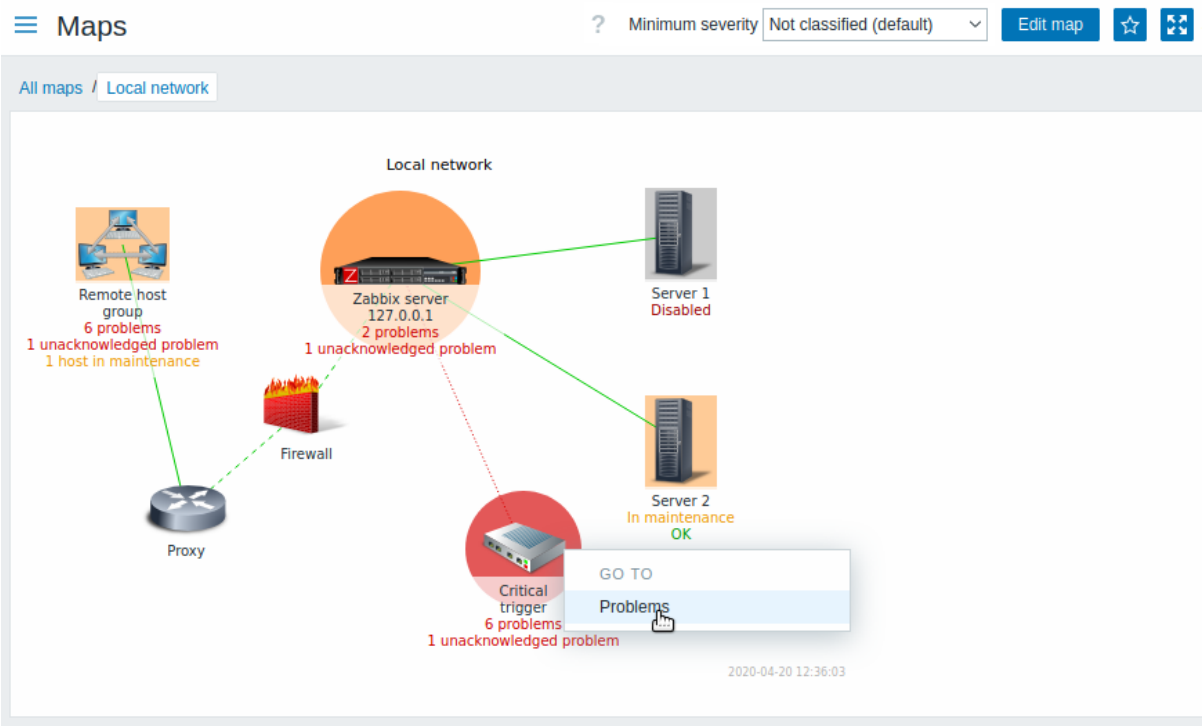
Para usar essas opções, marque as caixas de seleção antes dos respectivos mapas, em seguida, clique no botão desejado.

Usando 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 severidade dos triggers de problema a serem exibidos. A severidade marcada como *padrão* é o nível definido na configuração do mapa. Se o mapa contiver um submapa, navegar até o submapa manterá a severidade 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 de 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, são descritos na página Monitoramento.

Resumo legível em mapas

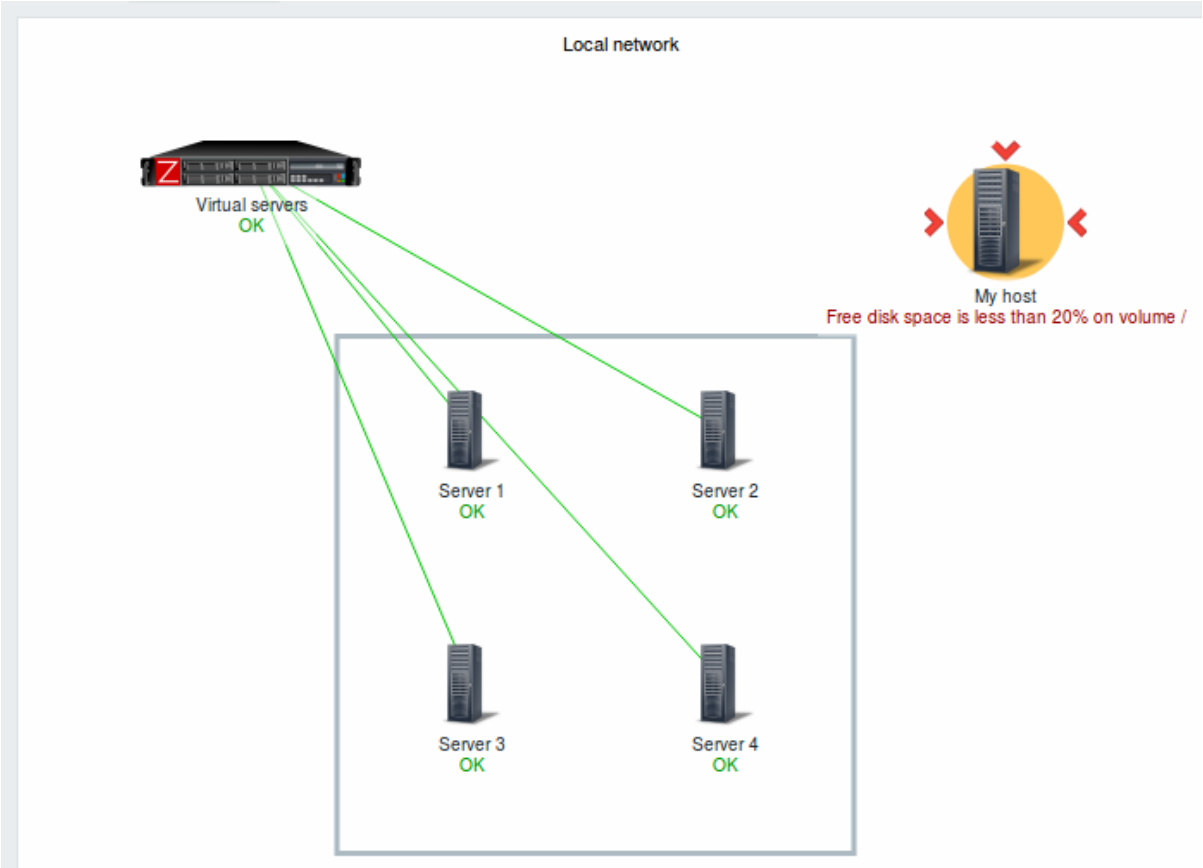
Uma propriedade oculta "aria-label" está disponível, permitindo que as informações do mapa sejam lidas por um leitor de tela. Tanto a descrição geral do mapa quanto a descrição individual dos elementos estão disponíveis, no seguinte formato:

- 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 vários problemas: `<Tipo de elemento>, Status <Status do elemento>, <Nome 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 problem, 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
```

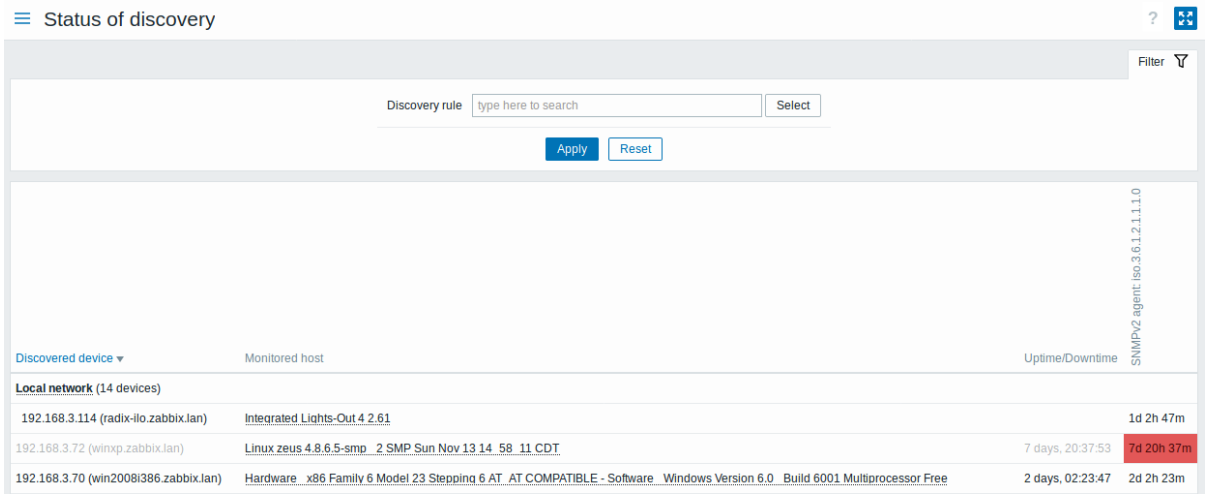
abrirá o mapa com esse nome (Local network).

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 em que o dispositivo foi 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 página Monitoramento.

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.

Sem nada selecionado no filtro, todas as regras de descoberta habilitadas são exibidas. Para selecionar uma regra de descoberta específica para exibição, comece a digitar seu nome no filtro. Todas as regras de descoberta habilitadas correspondentes serão listadas para seleção. Mais de uma regra de descoberta pode ser selecionada.

3 Serviços

Visão geral

O menu Serviços é destinado às funções de monitoramento de serviços do Zabbix.

1 Serviços

Visão geral

Nesta seção, você pode ver o status de alto nível de todos os serviços que foram configurados no Zabbix, com base em sua infraestrutura.

Um serviço pode ser uma hierarquia composta por vários níveis de outros serviços, chamados de serviços ”filhos”, que contribuem 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 respectivo nome e cor da gravidade do problema.

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:

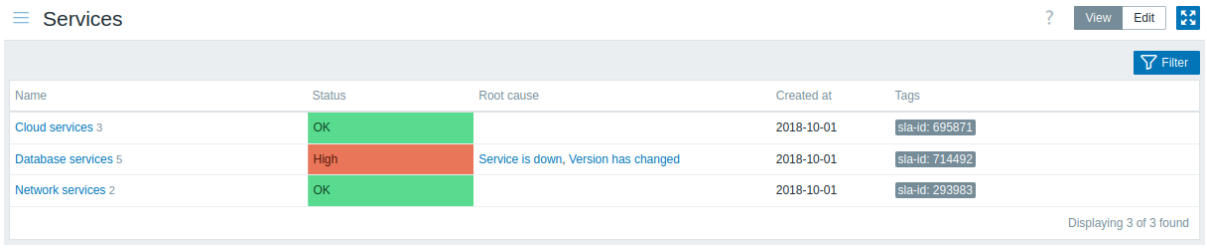
- 

- visualizar serviços
- 

- adicionar/editar serviços e serviços filhos

Observe que o acesso à edição depende das configurações de função de 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 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 retornados pela macro {SERVICE.ROOTCAUSE}. 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.

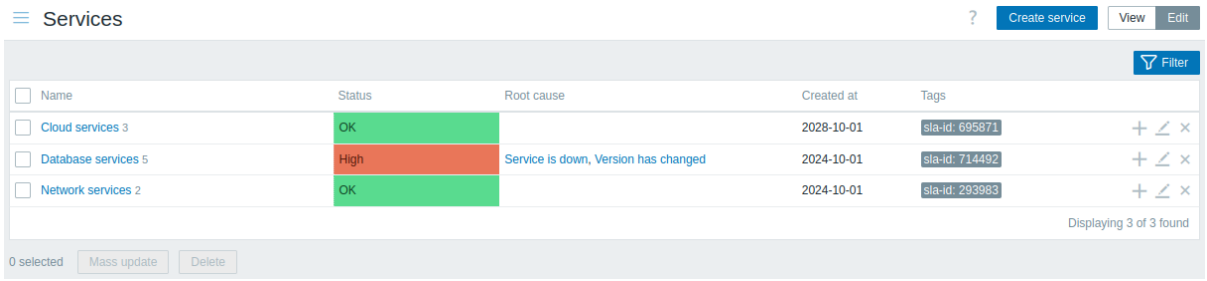
Usando o filtro

Você pode usar o filtro para exibir apenas os serviços que lhe interessam.

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 pai (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:

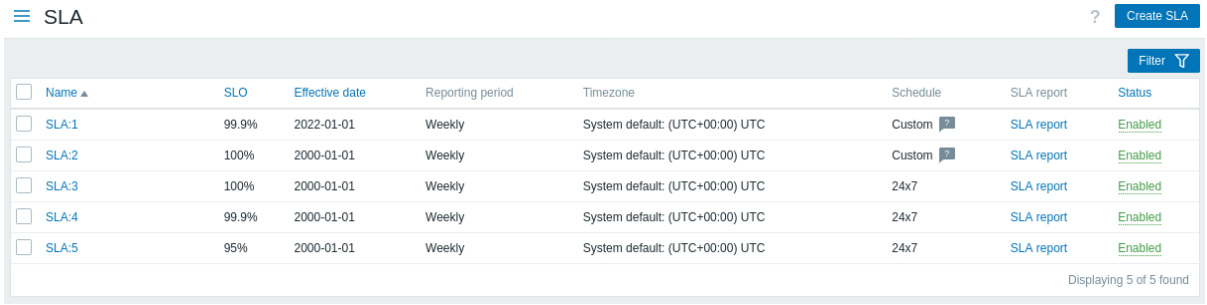
-  - adicionar um serviço filho a este serviço
-  - editar este serviço
-  - excluir este serviço

2 SLA

Visão geral

Esta seção permite visualizar e configurar 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 de 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 de 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

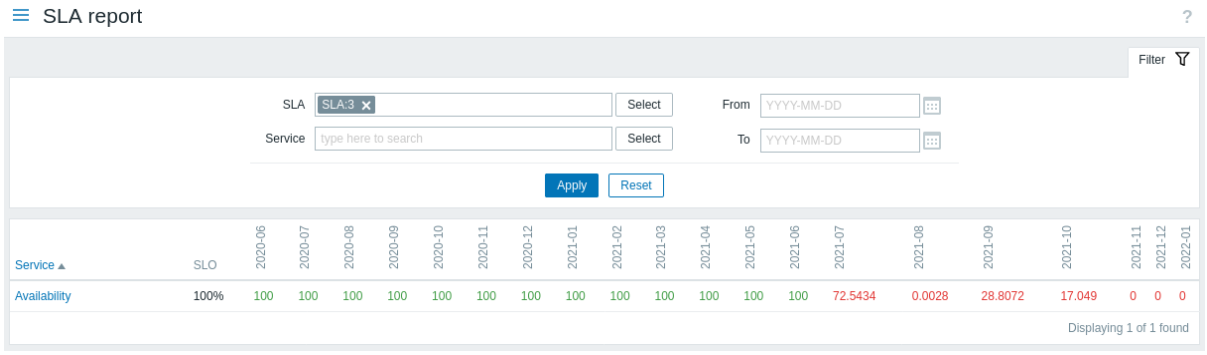
Visão geral

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 violando o SLO definido são destacados em vermelho.

20 períodos são exibidos no relatório. Um máximo de 100 períodos pode ser exibido, se ambas as datas *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 a geração de relatórios de SLA.

4 Inventário

Visão geral

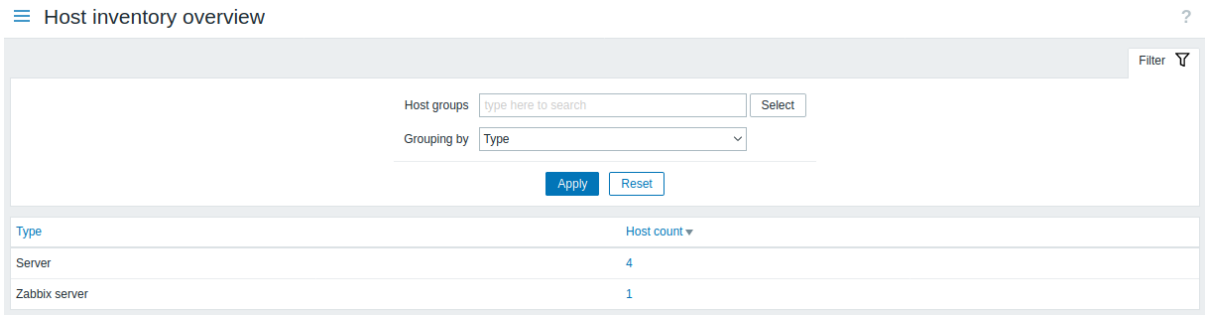
O menu Inventário apresenta seções que fornecem uma visão geral dos dados de inventário do host por um parâmetro escolhido, bem como a capacidade de visualizar detalhes do inventário do host.

1 Visão geral

Visão geral

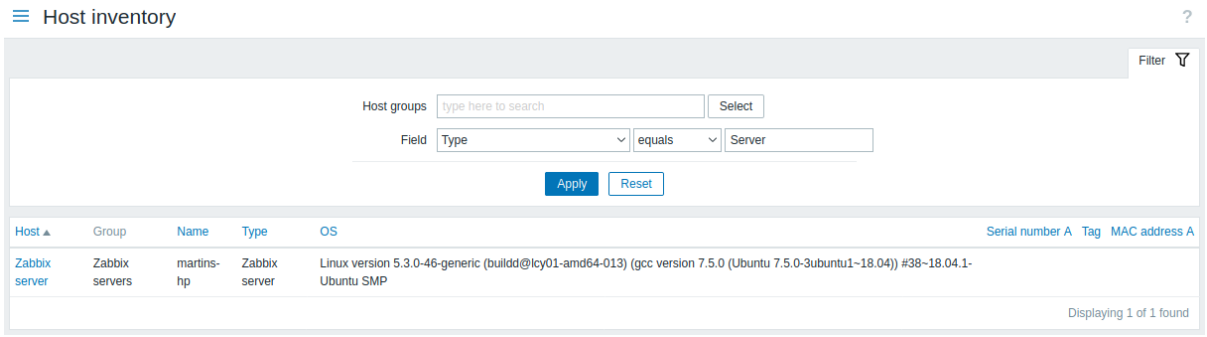
A seção *Inventário* → *Visão geral* fornece maneiras de obter uma visão geral dos dados de inventário de host.

Para que uma visão geral seja exibida, escolha grupos de hosts (ou nenhum) e o campo de inventário pelo qual exibir os dados. O número de hosts correspondente a cada entrada do campo escolhido será exibido.



A completude de uma visão geral depende de quanta informação de inventário é mantida com os hosts.

Os números na coluna *Contagem de hosts* são links; eles levam a esses hosts sendo filtrados na tabela *Inventário de hosts*.

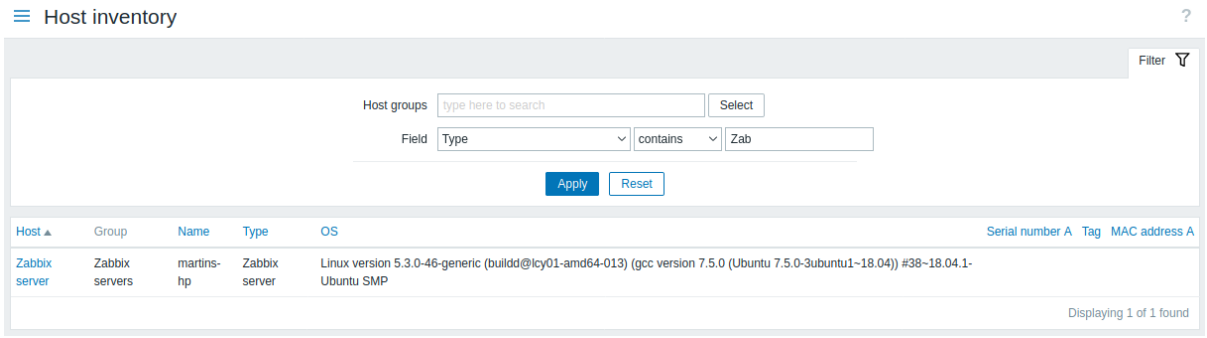


2 Hosts

Visão geral

Na seção *Inventário* → *Hosts*, os dados de inventário dos hosts são exibidos.

Você pode filtrar os hosts por grupo(s) de host e por qualquer campo de inventário para exibir apenas os hosts do seu interesse.

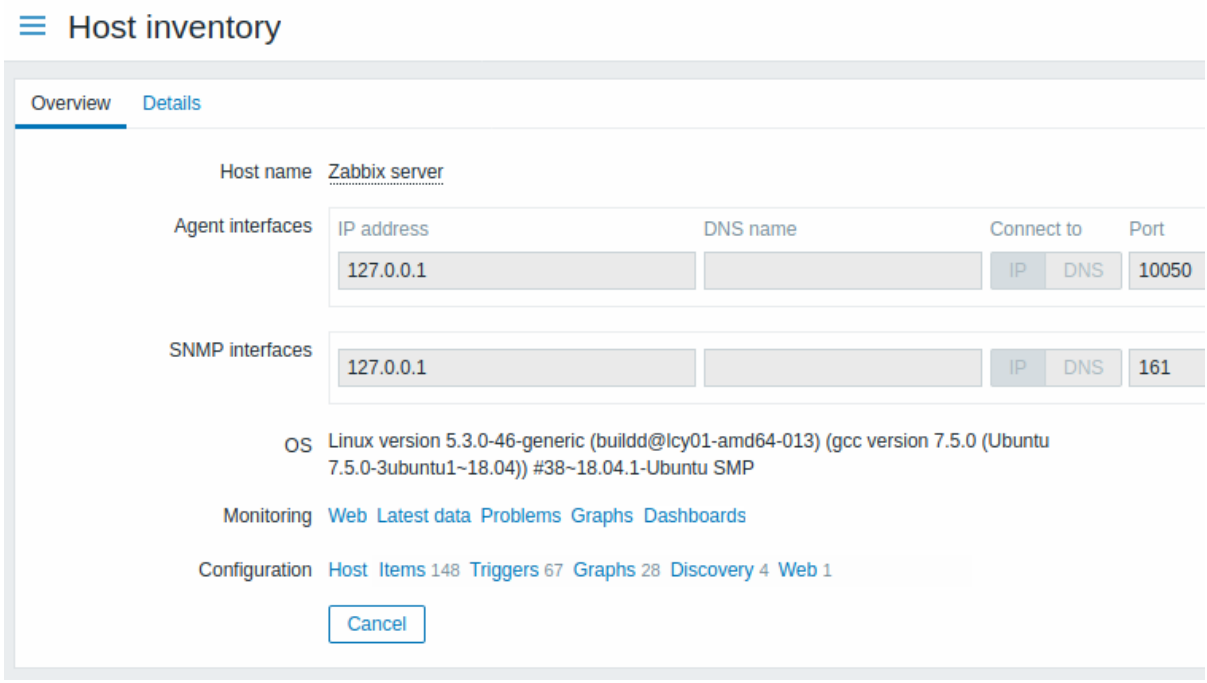


Para exibir todos os inventários de host, pressione o botão ”Resetar”.

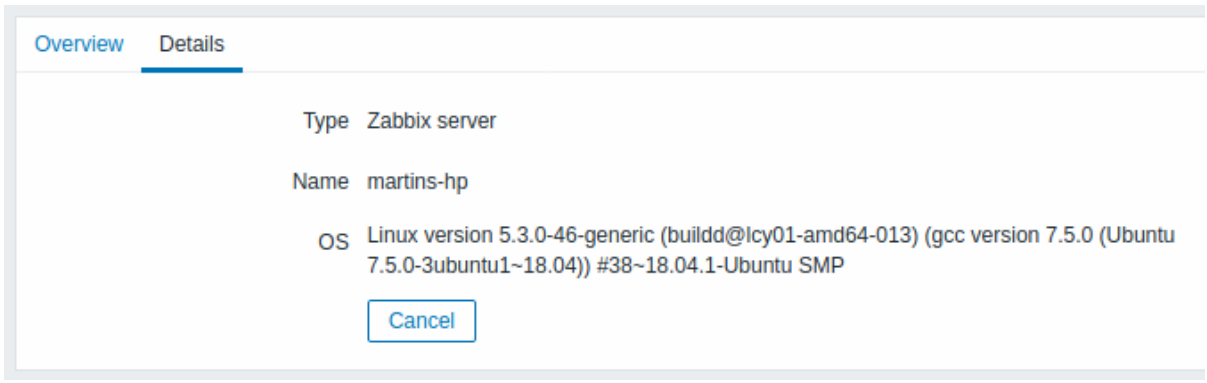
Embora apenas alguns campos-chave de inventário sejam exibidos na tabela, você também pode visualizar todas as informações de inventário disponíveis para esse host. Para isso, clique no nome do host na primeira coluna.

Detalhes do inventário

A aba **Visão geral** contém algumas informações gerais sobre o host, os dados de monitoramento mais recentes e 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, triggers e dados coletados.

1 Informações do sistema

Visão geral

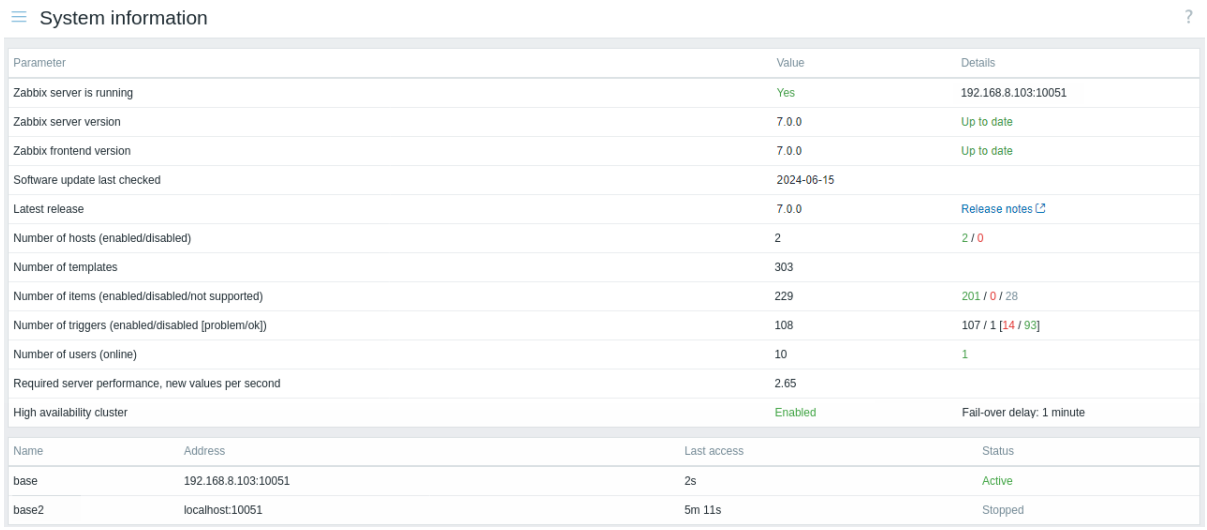
Em *Relatórios* → *Informações do sistema*, é exibido um resumo dos principais dados do server e do sistema Zabbix. Os dados do sistema são coletados usando items internos.

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 `$ZBX_SERVER_PORT` para um server diferente do que está ativo. Observe que, ao definir apenas `$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 nodes de alta disponibilidade. Este bloco é visível apenas para usuários *Super Admin* do Zabbix.

Informações do sistema também estão disponíveis como um widget no dashboard.

Estatísticas do sistema



Dados exibidos:

Parâmetro	Valor	Detalhes
Zabbix server está em execução	Status do Zabbix server: **Sim** - server está em execução **Não** - 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.
Versão do Zabbix server	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: **Atualizado** - usando a versão mais recente; **Nova atualização disponível** - uma versão mais atualizada está disponível; **Desatualizado** - o período de suporte total 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 se não existem dados sobre a versão atual.
Versão do Zabbix frontend	O número da versão do Zabbix frontend é exibido.	O status da versão do Zabbix frontend é exibido: **Atualizado** - usando a versão mais recente; **Nova atualização disponível** - uma versão mais atualizada está disponível; **Desatualizado** - o período de suporte total 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 se não existem dados sobre a versão atual.
Última verificação de atualização de software	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.	

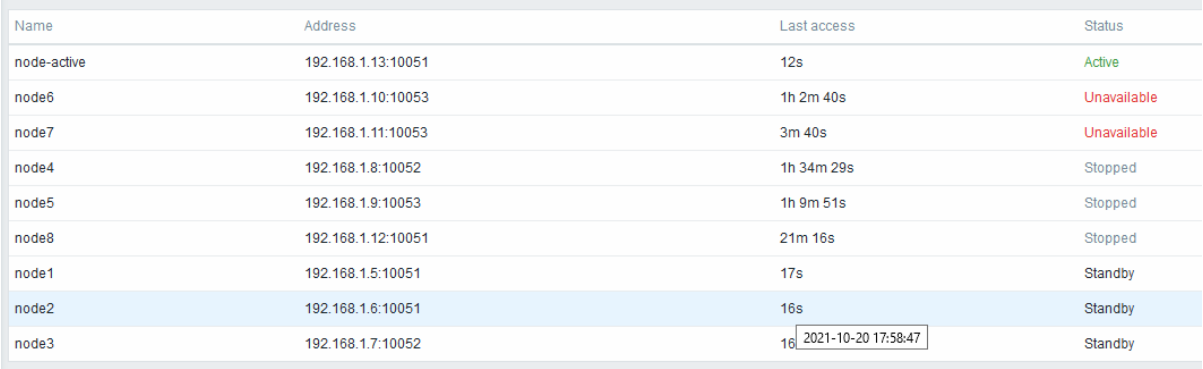
Parâmetro	Valor	Detalhes
Último release	O número de um release mais recente (se disponível) para a versão atual do Zabbix é exibido. Esta informação só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server. Nada é exibido se a última verificação de atualização de software foi realizada há mais de uma semana ou se não existem dados sobre a versão atual.	Um link para as notas de release do último release disponível do Zabbix é exibido.
Número de hosts	O número total de hosts configurados é exibido.	Número de hosts monitorados/hosts não monitorados.
Número de templates	O número total de templates é exibido.	
Número de 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.
Número de triggers	O número total de triggers é exibido.	Número de triggers de host habilitadas/desabilitadas; divisão das triggers habilitadas de acordo com os estados ”Problema”/”OK”. Triggers listadas sob o estado ”OK” incluem triggers com o status ”Desconhecido”. Triggers dependentes de items desabilitados ou atribuídas a hosts desabilitados são contadas como desabilitadas.
Número de usuários	O número total de usuários configurados é exibido.	Número de usuários online.
Desempenho necessário do server, novos valores por segundo	O número esperado de novos valores processados pelo Zabbix server por segundo é exibido.	*Desempenho necessário do server* 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. O cálculo não é ajustado durante um período de manutenção ”nodata”. Items trapper não são contados.
Scripts globais no Zabbix server	**Desabilitado** será exibido neste campo se os scripts globais estiverem desabilitados no Zabbix server, configurando EnableGlobalScripts=0 na configuração do server.	
Cluster de alta disponibilidade	Status do cluster de alta disponibilidade para o Zabbix server: **Desabilitado** - server standalone **Habilitado** - pelo menos um nó de alta disponibilidade existe	Se habilitado, o atraso de failover é exibido.

Informações do sistema 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*).
- Housekeeping para TimescaleDB está configurado incorretamente (tabelas de histórico ou tendência contêm chunks comprimidos, mas as opções *Substituir período de histórico do item* ou *Substituir período de tendência do item* 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 dashboards, 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 filtro abaixo.

Dados exibidos:

Coluna	Descrição
Nome	Nome do relatório. Clicando 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 para o qual o relatório é preparado.
Último envio	A data e hora em que o último relatório 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) 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 em massa, 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 no estado de problema/OK.

Para cada estado, é exibida uma porcentagem de tempo, facilitando a determinação da 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 restringir 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 dos parâmetros listados abaixo.

Parâmetro	Descrição
Grupo de template	Filtra hosts por triggers herdados de templates pertencentes ao grupo de template selecionado. Especificar um grupo de template pai seleciona implicitamente todos os grupos de template aninhados.
Template	Filtra hosts por triggers herdados do template selecionado, incluindo templates aninhados. Se um template aninhado tiver seus próprios triggers, esses triggers não serão exibidos.
Trigger do template	Filtra hosts pelo trigger selecionado. Outros triggers dos hosts filtrados não serão exibidos.
Grupo de host	Filtra hosts pertencentes ao grupo de host selecionado.

Filtrando por host

No modo *Por host*, os resultados podem ser filtrados por host ou 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 de tempo comumente usados com um clique. O seletor pode ser expandido e recolhido clicando na guia *Período de tempo* ao lado do filtro.

Clicar em *Exibir* na coluna *Gráfico* exibe informações de disponibilidade em um gráfico de barras onde cada barra representa uma semana passada do ano atual.

O verde de uma barra representa o tempo OK e o vermelho - o tempo de problema.

Efeito dos períodos de manutenção

A manutenção não exclui automaticamente o tempo do relatório de disponibilidade. Apenas 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

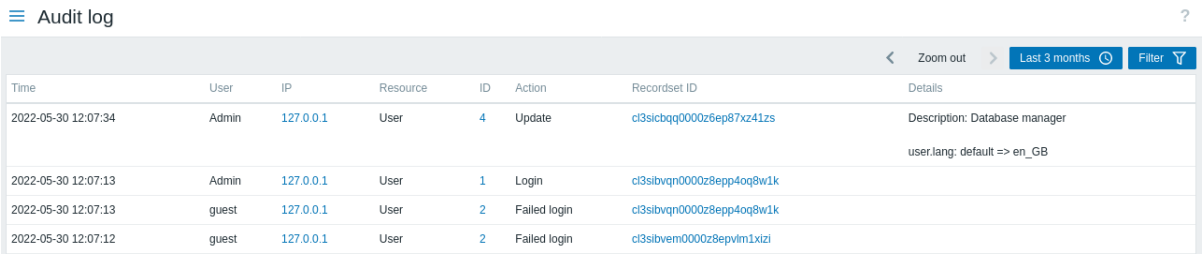
O seletor de período de tempo ao lado do filtro permite selecionar períodos frequentemente necessários com um clique do mouse. Para mais informações, consulte Seletor de período de tempo e host.

5 Log de auditoria

Visão geral

Na seção *Relatórios* → *Log de auditoria*, os registros de atividade do usuário e do sistema podem ser visualizados.

Note:
Para que os registros de auditoria sejam coletados e exibidos, a caixa de seleção *Habilitar log 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:

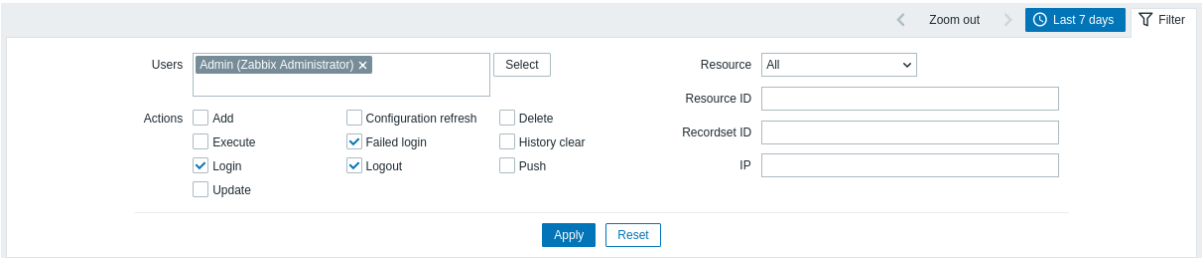
Coluna	Descrição
Hora	Carimbo de data/hora do registro de auditoria.
Usuário	Usuário que realizou a atividade.
IP	IP de onde a atividade foi iniciada. Clicar no hiperlink resultará na filtragem dos registros do log de auditoria por esse IP.
Recurso	Tipo de recurso afetado (*Token de API*, *Ação*, *Autenticação*, *Autoregistro*, etc.).
ID	ID do recurso afetado. Clicar no hiperlink resultará na filtragem dos registros do log de auditoria por esse ID de recurso.
Ação	Tipo de atividade (*Adicionar*, *Atualização de configuração*, *Excluir*, *Executar*, *Falha de login*, *Limpar histórico*, *Login*, *Logout*, *Push*, *Atualizar*).

Coluna	Descrição
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 log 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*. Clicar no hiperlink resultará na filtragem dos registros do log de auditoria 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. Dependendo do recurso, uma ou mais ações específicas podem ser selecionadas no filtro.

Para melhor desempenho na pesquisa, todos os dados são pesquisados com macros não resolvidas.

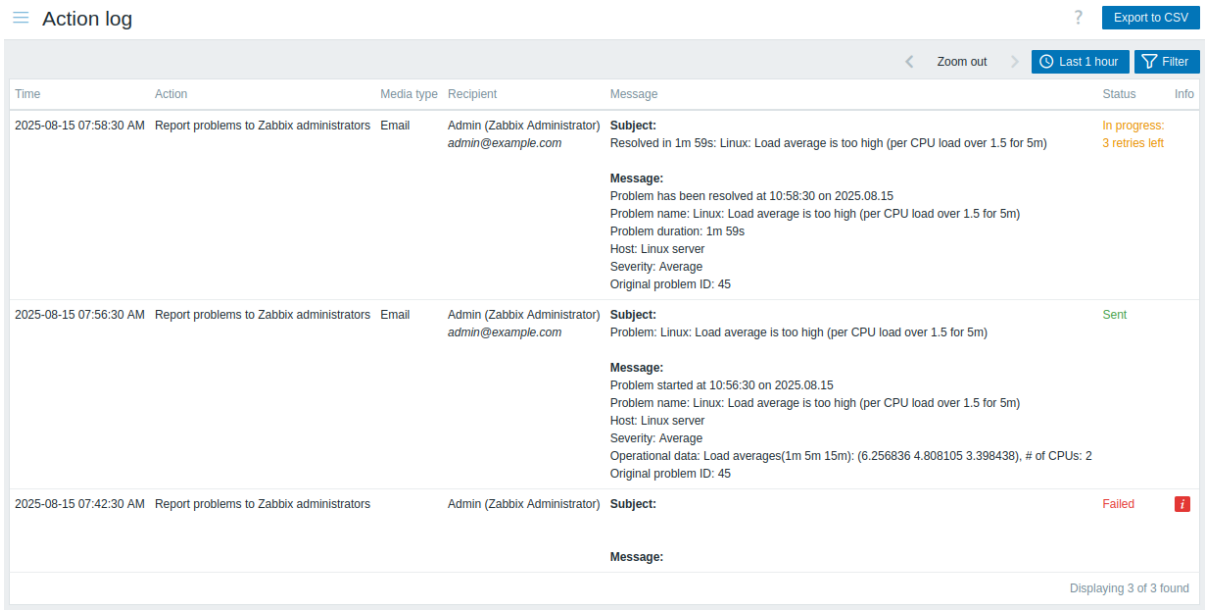
Seletor de período de tempo

O seletor de período de tempo ao lado do filtro permite selecionar períodos frequentemente necessários com um clique do mouse. Para mais informações, consulte Seletor de período de tempo e host.

6 Log de ações

Visão geral

Em *Relatórios* → *Log de ações*, você pode 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.
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 por 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 tentativas que o server fará para 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) sobre a execução da operação.

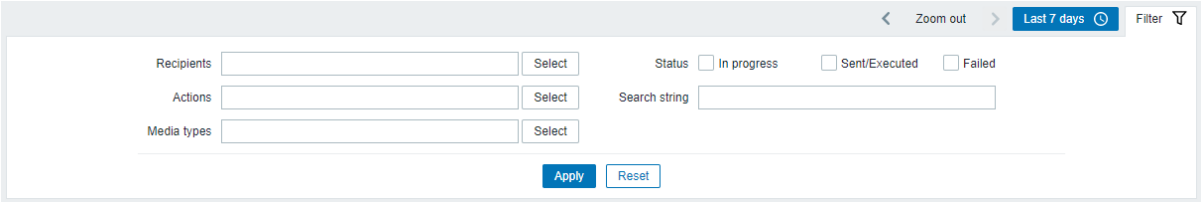
Botões

O botão no canto superior direito da página oferece a seguinte opção:

Export to CSV	Exporta os registros do log de ações de todas as páginas 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 de pesquisa, os dados são pesquisados com macros não resolvidas.

Seletor de período de tempo

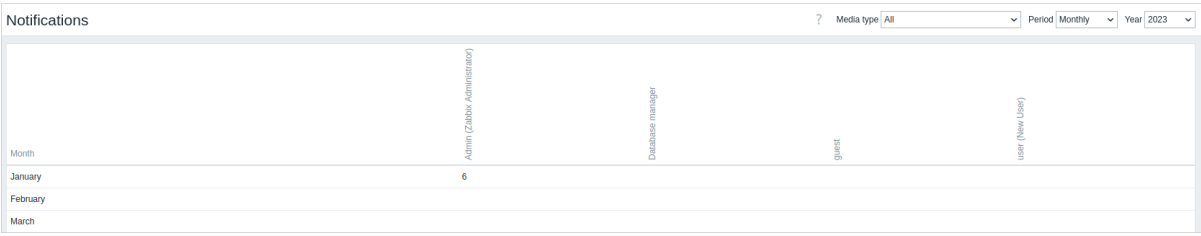
O seletor de período de tempo ao lado do filtro permite selecionar períodos frequentemente necessários com um clique do mouse. Para mais informações, consulte Seletor de período de tempo e host.

7 Notificações

Visão geral

Na seção *Relatórios* → *Notificações* é exibido um relatório sobre o número de notificações enviadas para cada usuário.

Nos menus suspensos no canto superior direito, você pode escolher o tipo de mídia (ou todos), o período (dados para cada dia/semana/mês/ano) e o ano para as notificações enviadas.



Cada coluna exibe totais por usuário do sistema.

6 Coleta de dados

Visão geral

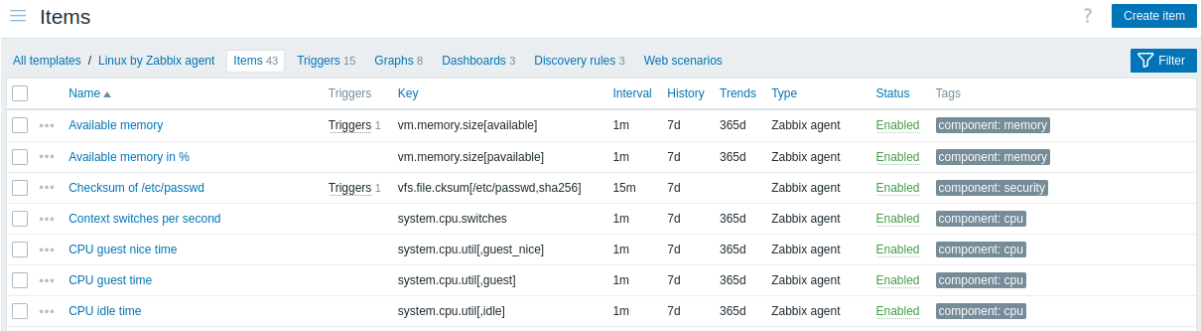
Este menu apresenta seções relacionadas à configuração da coleta de dados.

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.

Coluna	Descrição
Template	Template ao qual o item pertence. 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 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, verificação 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 que permite 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:

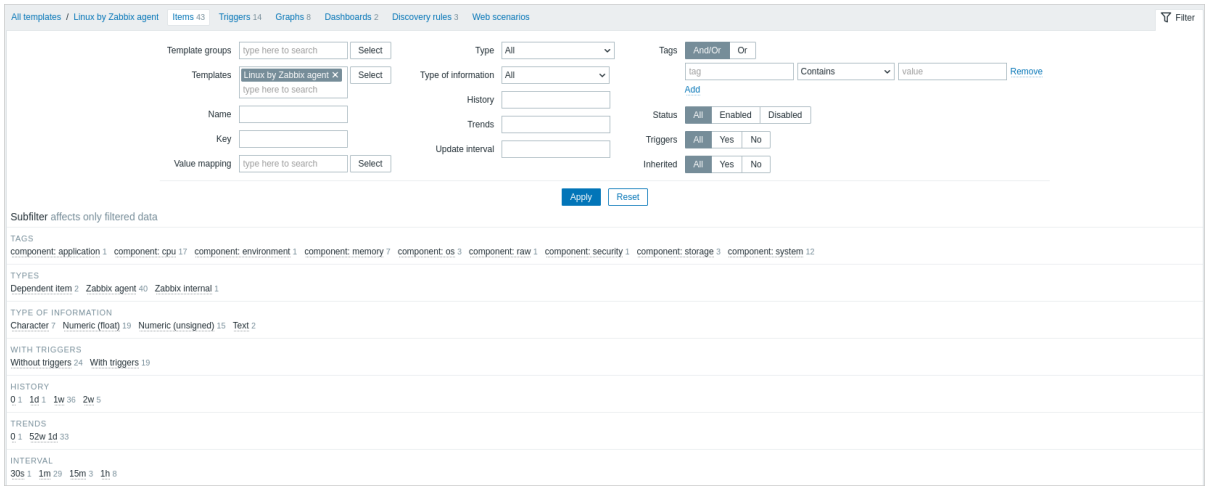
- *Ativar* - altera o status do item para *Ativado*.
- *Desativar* - altera o status do item para *Desativado*.
- *Copiar* - copia os items para outros hosts ou templates.
- *Atualização em massa* - atualiza várias propriedades de vários items de uma só vez.
- *Excluir* - exclui os items.

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 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.



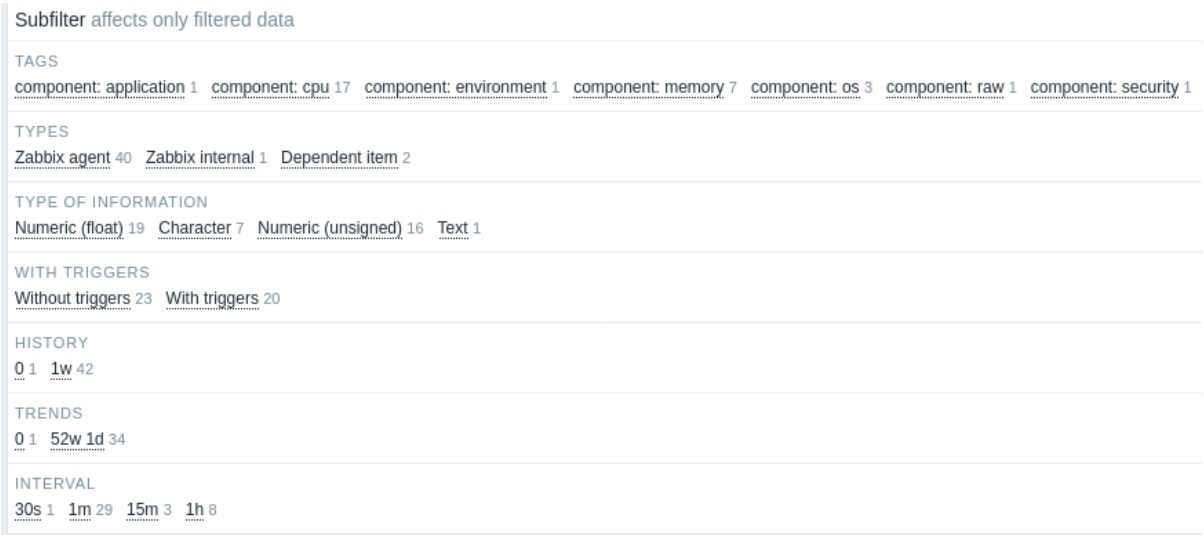
Parâmetro	Descrição
Grupos de templates	Filtrar por um ou mais grupos de templates. Especificar um grupo de templates pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.

Parâmetro	Descrição
Nome	Filtrar por nome do item.
Chave	Filtrar por chave do item.
Mapeamento de valor	Filtrar pelo mapa de valor 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, ponto flutuante, 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
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.

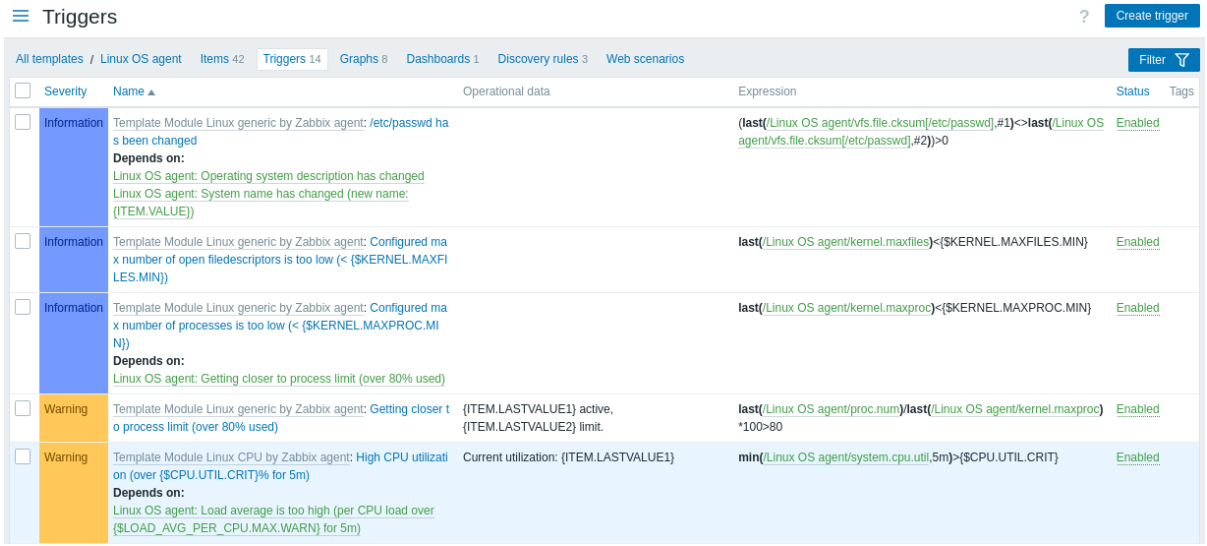
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de três pontos é exibido no final; ele não é clicável.

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. 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 link do 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 naquele nível de template.
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 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 trigger para *Ativado*
- *Desativar* - altera o status do trigger para *Desativado*
- *Copiar* - copia os triggers para outros hosts ou templates

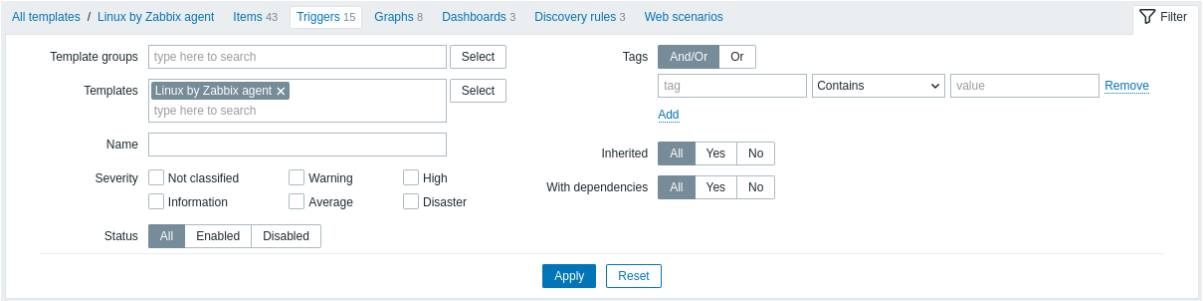
- *Atualização em massa* - atualiza várias propriedades de vários triggers de uma só vez
- *Excluir* - exclui os triggers

Para usar essas opções, marque as caixas de seleção antes dos triggers correspondentes e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os triggers que lhe interessam. 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.



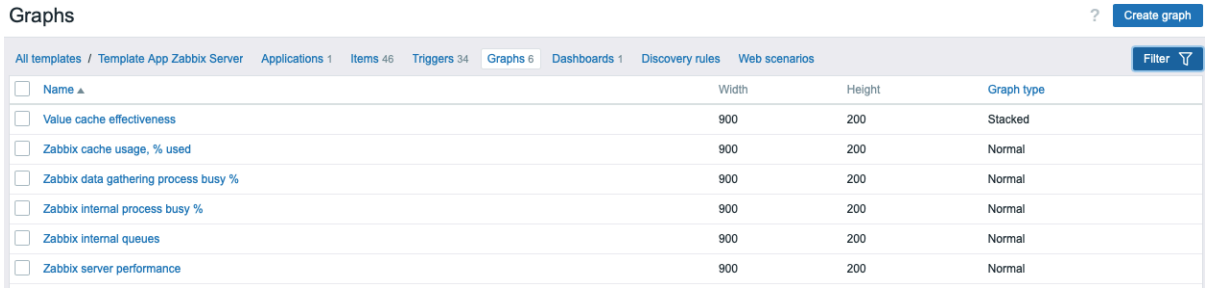
Parâmetro	Descrição
Grupos de template	Filtrar por um ou mais grupos de template. Ao especificar um grupo de template pai, todos os grupos aninhados são implicitamente selecionados.
Templates	Filtrar por um ou mais templates. Se grupos de template já estiverem selecionados acima, a seleção de template é 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 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, não sensível 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, 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 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. 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* - copia os gráficos para outros hosts ou templates
- *Excluir* - exclui os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos e clique no botão desejado.

Usando o filtro

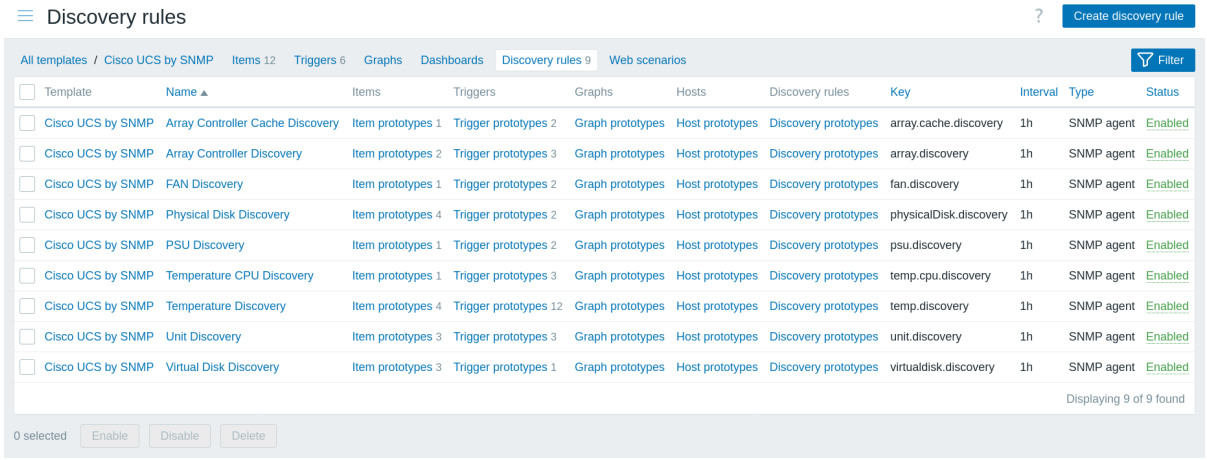
Você pode filtrar gráficos por grupo de template e template. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

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. Clicar no nome do template abre o formulário de configuração do template.
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 será exibido antes do nome da regra como um link cinza. Clicar no link do template abre a lista de regras de descoberta naquele nível de 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 (Zabbix agent, SNMP agent, 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:

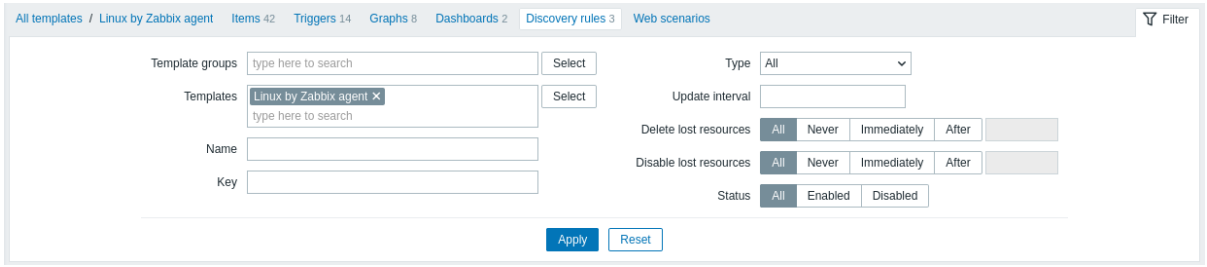
- *Ativar* - altera o status da regra de descoberta de baixo nível para *Ativado*
- *Desativar* - altera o status da regra de descoberta de baixo nível para *Desativado*
- *Excluir* - exclui as regras de descoberta de baixo nível

Para usar essas opções, marque as caixas de seleção antes 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 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 do item, etc.



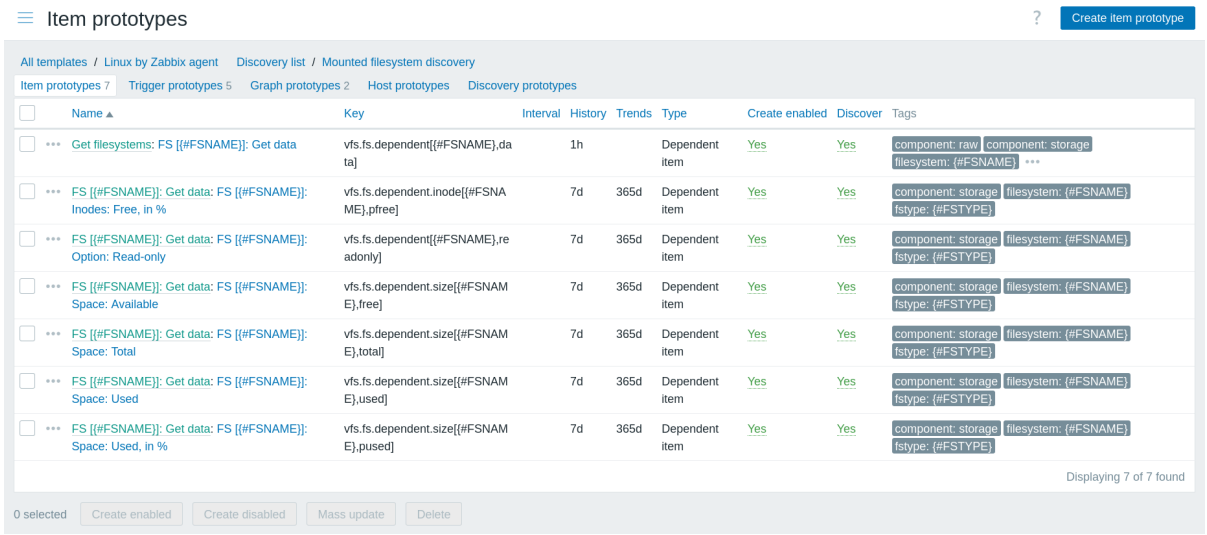
Parâmetro	Descrição
Grupos de templates	Filtrar por um ou mais grupos de templates. Especificar um grupo de templates pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.
Nome	Filtrar por nome da regra de descoberta.
Chave	Filtrar por chave do item de descoberta.
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 items

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. Clicar no nome abre o formulário de configuração do protótipo de item. 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. Clicar no link do template abre a lista de protótipos de item no nível do template vinculado. 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).

Coluna	Descrição
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 (agent Zabbix, agent SNMP, 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* - criar esses items como *Habilitado*
- *Criar desabilitado* - criar esses items como *Desabilitado*
- *Atualização em massa* - atualizar em massa esses protótipos de item
- *Excluir* - excluir esses protótipos de item

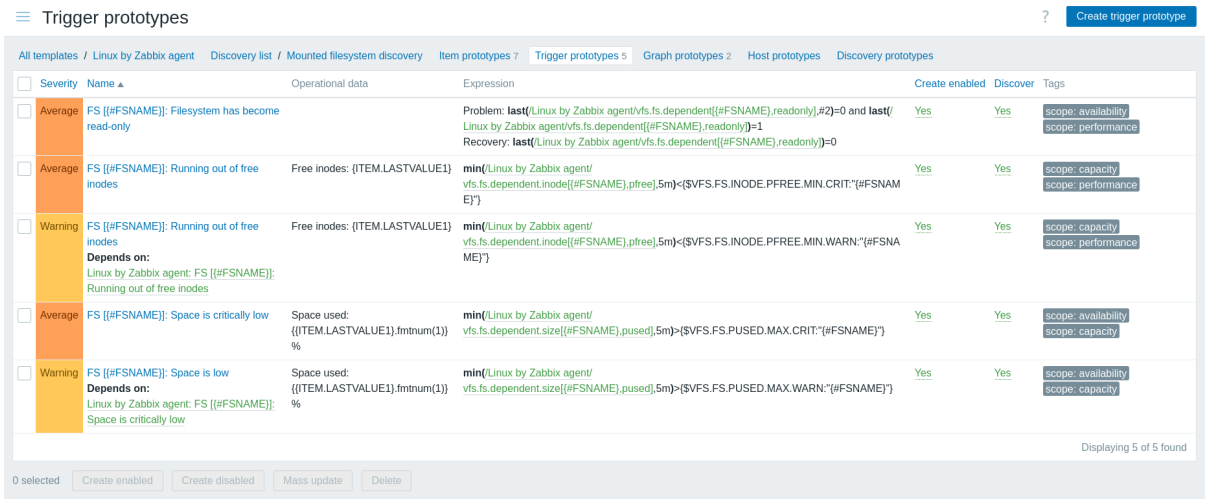
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de item e clique no botão desejado.

2 Protótipos de trigger

Visão geral

Nesta seção são exibidos os protótipos de trigger configurados de uma regra de descoberta de baixo nível no template.

Se o template estiver vinculado ao host, os protótipos de trigger se tornarão a 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.

Coluna	Descrição
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 sobre eles.
Descobrir	Descobrir o trigger com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando sobre eles.
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

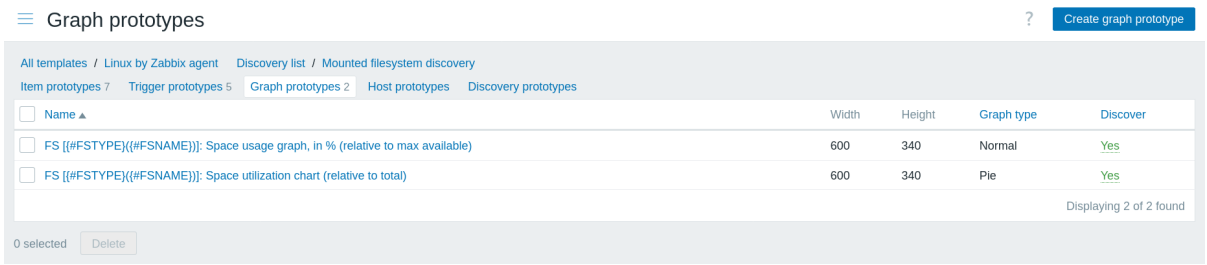
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de trigger e clique no botão desejado.

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. Ao clicar no nome, abre-se 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. Ao clicar no link do template, a lista de protótipos de gráficos é aberta 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 sobre eles.

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

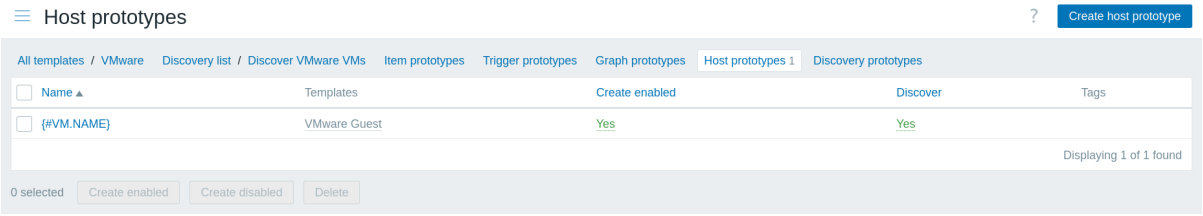
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráficos e clique no botão necessário.

4 Protótipos de host

Visão geral

Nesta seção, os protótipos de host configurados em uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de host servirão de 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.
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

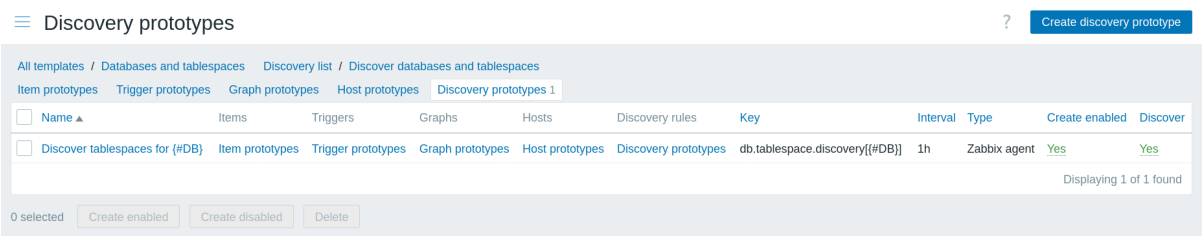
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e clique no botão desejado.

5 Protótipos de descoberta

Visão geral

Nesta seção, os protótipos de descoberta configurados de uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de descoberta se tornarão a base para a criação de protótipos de descoberta reais da regra de descoberta pai.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de descoberta, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de descoberta. Se o protótipo de descoberta pertencer a um template vinculado, o nome do template será exibido antes do nome do protótipo de descoberta como um link cinza. Clicar no link do template abre a lista de protótipos de descoberta no nível do template vinculado.
Itens	É exibido um link para a lista de protótipos de item. O número de protótipos de item existentes é exibido em cinza.
Triggers	É exibido um link para a lista de protótipos de trigger. O número de protótipos de trigger existentes é exibido em cinza.
Gráficos	É exibido um link para a lista de protótipos de gráfico. O número de protótipos de gráfico existentes é exibido em cinza.
Hosts	É exibido um link para a lista de protótipos de host. O número de protótipos de host existentes é exibido em cinza.
Regras de descoberta	É exibido um link para a lista de protótipos de descoberta. O número de protótipos de descoberta 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 (Zabbix agent, SNMP agent, etc).
Criar habilitado	O status de criação do protótipo de descoberta é exibido - criar habilitado (*Sim*) ou criar desabilitado (*Não*). Clicando no status você pode alterá-lo - de Sim para Não (e vice-versa).
Descobrir	O status de descoberta do protótipo de descoberta é exibido - descobrir (*Sim*) ou não descobrir (*Não*). Clicando no status você pode alterá-lo - de Sim para Não (e vice-versa).

Para configurar um novo protótipo de descoberta, clique no botão *Criar protótipo de descoberta* 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 protótipos de descoberta como *Habilitado*
- *Criar desabilitado* - criar esses protótipos de descoberta como *Desabilitado*
- *Excluir* - excluir esses protótipos de descoberta

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de descoberta 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 ou Nenhum.
Proxy HTTP	Exibe o proxy HTTP ou 'Não' se não for usado.
Status	O status do cenário web é exibido - *Ativado* ou *Desativado*. Clicando no status você pode alterá-lo.

Coluna	Descrição
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.

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:

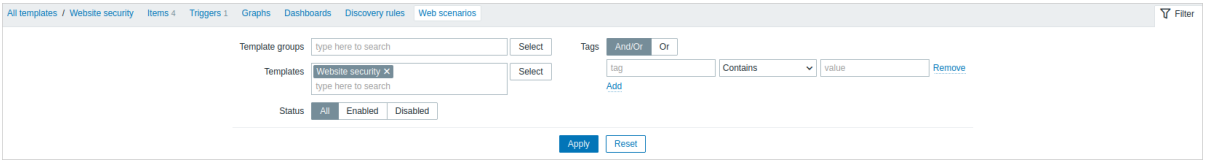
- *Habilitar* - altera o status do cenário para *Habilitado*
- *Desabilitar* - altera o status do cenário para *Desabilitado*
- *Excluir* - exclui os cenários web

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 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 se torna disponível onde você pode filtrar cenários por grupo de template, template, status e tags.

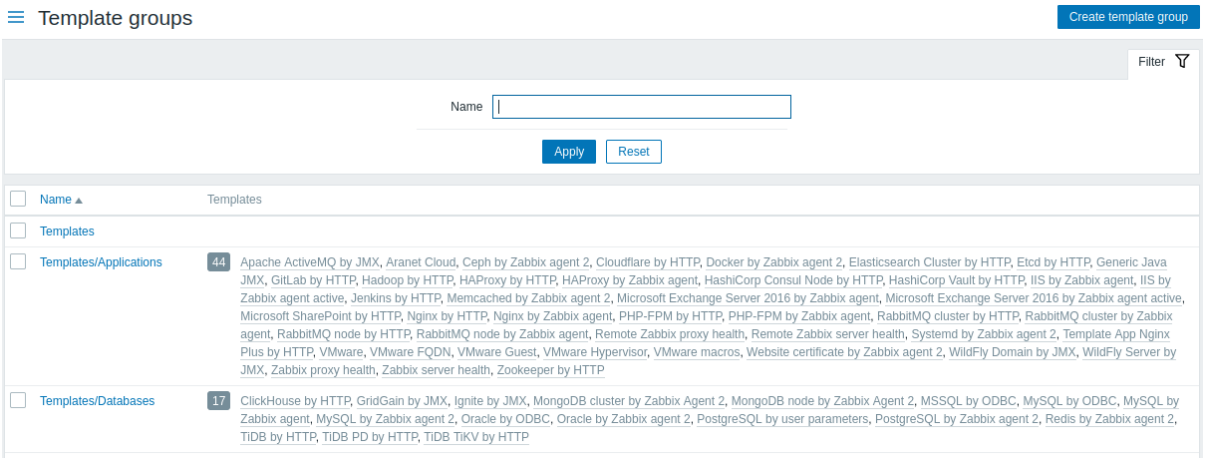


1 Grupos de templates

Visão geral

Na seção *Coleta de dados* → *Grupos de templates*, os usuários podem configurar e manter grupos de templates.

Uma listagem 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 templates de uma vez, marque as caixas de seleção antes dos respectivos grupos e clique no botão Excluir abaixo da lista.

Usando o filtro

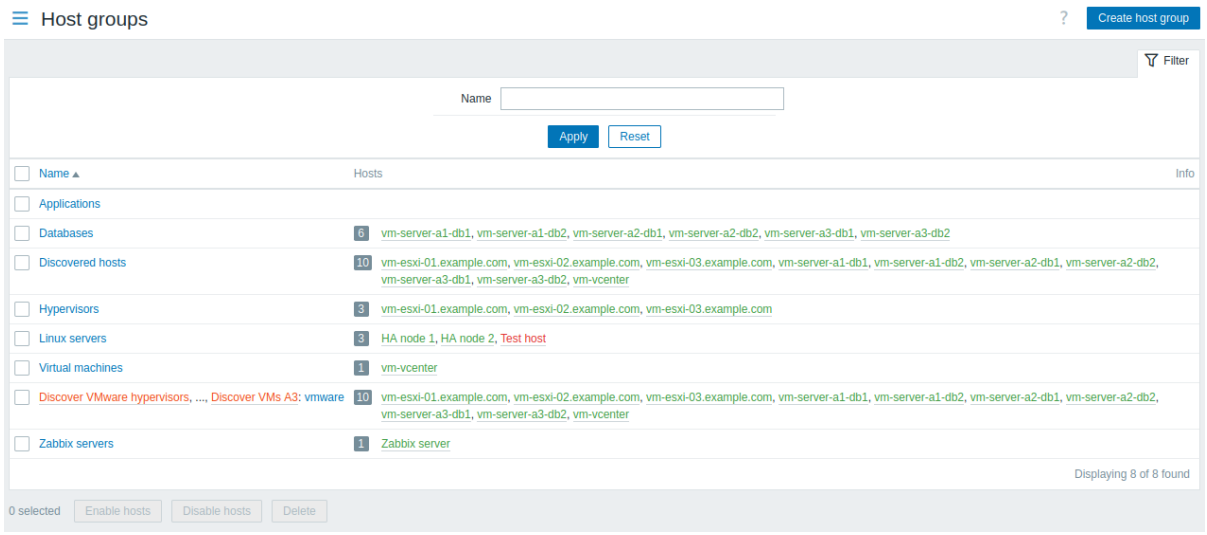
Você pode usar o filtro para exibir apenas os grupos de templates que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

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 irá, na listagem completa de hosts, filtrar 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:

- *Ativar hosts* - altera o status de todos os hosts do grupo para ”Monitorado”
- *Desativar hosts* - altera o status de todos os hosts do grupo para ”Não monitorado”
- *Excluir* - exclui os grupos de hosts

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de hosts e clique no botão desejado.

Usando o filtro

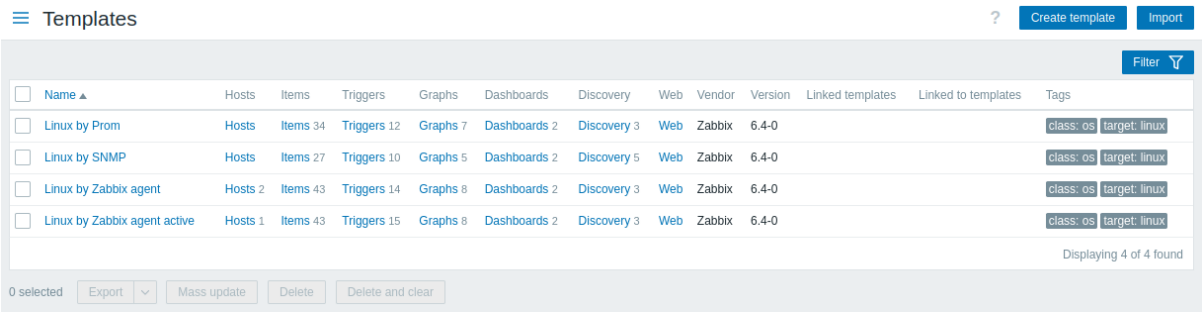
Você pode usar o filtro para exibir apenas os grupos de hosts que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

3 Templates

Visão geral

Na seção *Coleta de dados* → *Templates*, os usuários podem configurar e manter templates.

Uma listagem 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, na listagem completa dessa entidade, filtrará 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 nova versão principal do Zabbix). Por exemplo, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Tags	Tags do template, com macros não resolvidas.

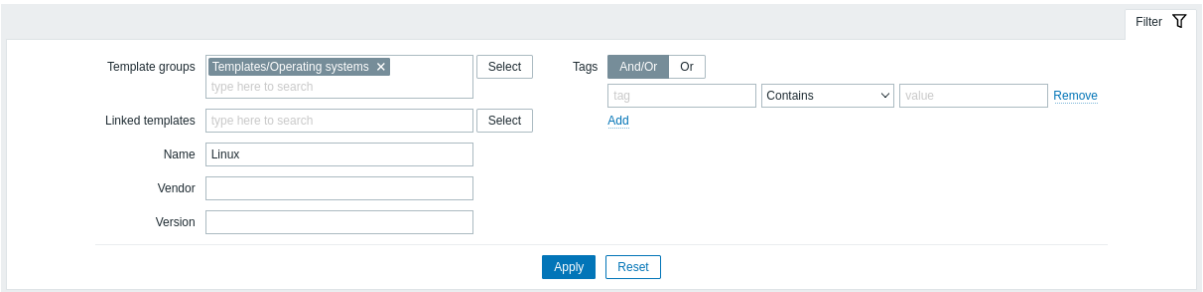
Para configurar um novo template, clique no botão *Criar template* no canto superior direito.

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 do 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 template, templates vinculados, nome e tags.



Parâmetro	Descrição
Grupos de template	Filtra por um ou mais grupos de template. Especificar um grupo de template pai seleciona implicitamente todos os grupos aninhados.
Templates vinculados	Filtra por templates vinculados diretamente.
Nome	Filtra por nome do template.
Fornecedor	Filtra por fornecedor do template.
Versão	Filtra por versão do template.
Tags	Filtra por 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 só 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.

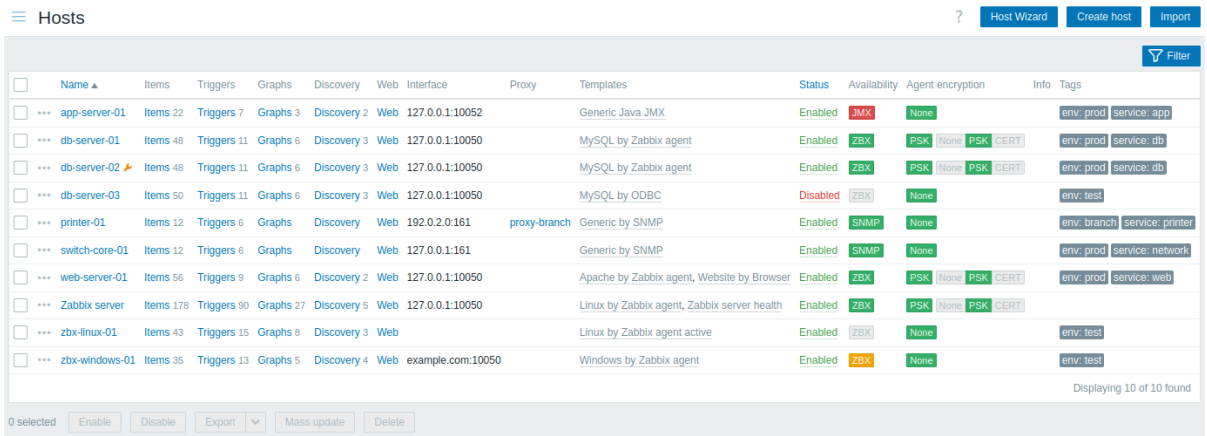
Para usar essas opções, marque as caixas de seleção antes dos respectivos templates e clique no botão desejado.

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
Menu do host	Clique no ícone de três pontos para abrir o menu do host.
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.
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 ou 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 desconhecida 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 é desconhecida, mas nenhuma está indisponível. **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.
Criptografia do agent	O status da criptografia para conexões com o host e do 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 *Assistente de host* ou *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 vez;

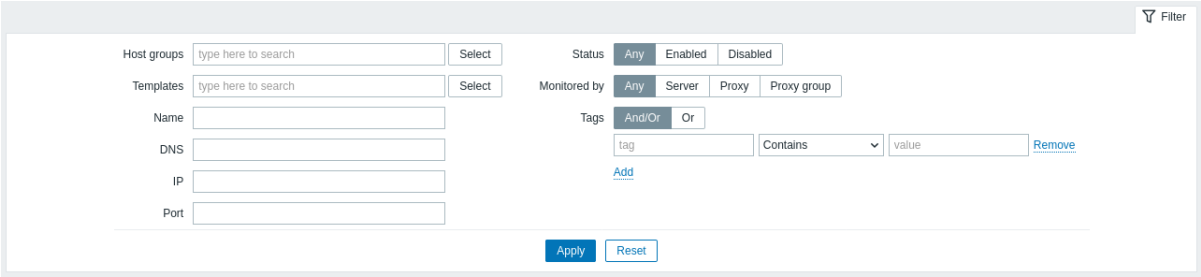
- *Excluir* - exclui os hosts.

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 nos quais está interessado. 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.
Templates	Filtrar por templates vinculados.
Nome	Filtrar por nome visível do host.
DNS	Filtrar por nome DNS.
IP	Filtrar por endereço IP.
Porta	Filtrar por número da porta.
Status	Filtrar por status do host.
Monitorado por	Filtrar hosts que são monitorados pelo Zabbix server, proxy ou grupo de proxy.
Proxies	Filtrar hosts que são monitorados pelos proxies especificados aqui. Este campo só está disponível se ”Proxy” for selecionado no campo *Monitorado por*.
Grupos de proxy	Filtrar hosts que são monitorados pelos grupos de proxy especificados aqui. Este campo só está disponível se ”Grupo de proxy” for selecionado no campo *Monitorado por*.
Tags	Filtrar por 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; **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.

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 "Desconhecida" após a sincronização do cache de configuração do Zabbix server.

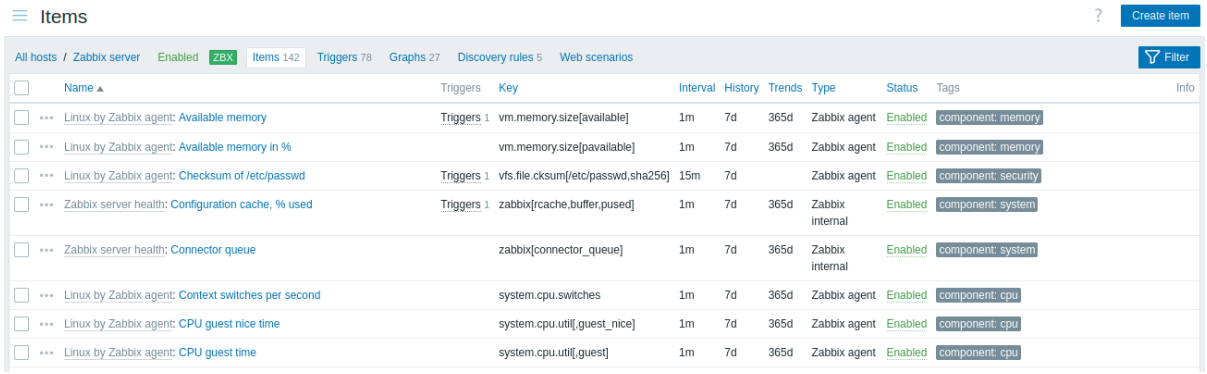
A disponibilidade da interface (Disponível/Não disponível) em hosts monitorados por proxies é restaurada após a sincronização do cache de configuração do proxy.

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. Clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida apenas se múltiplos 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 checados 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 (agent Zabbix, agent SNMP, 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:

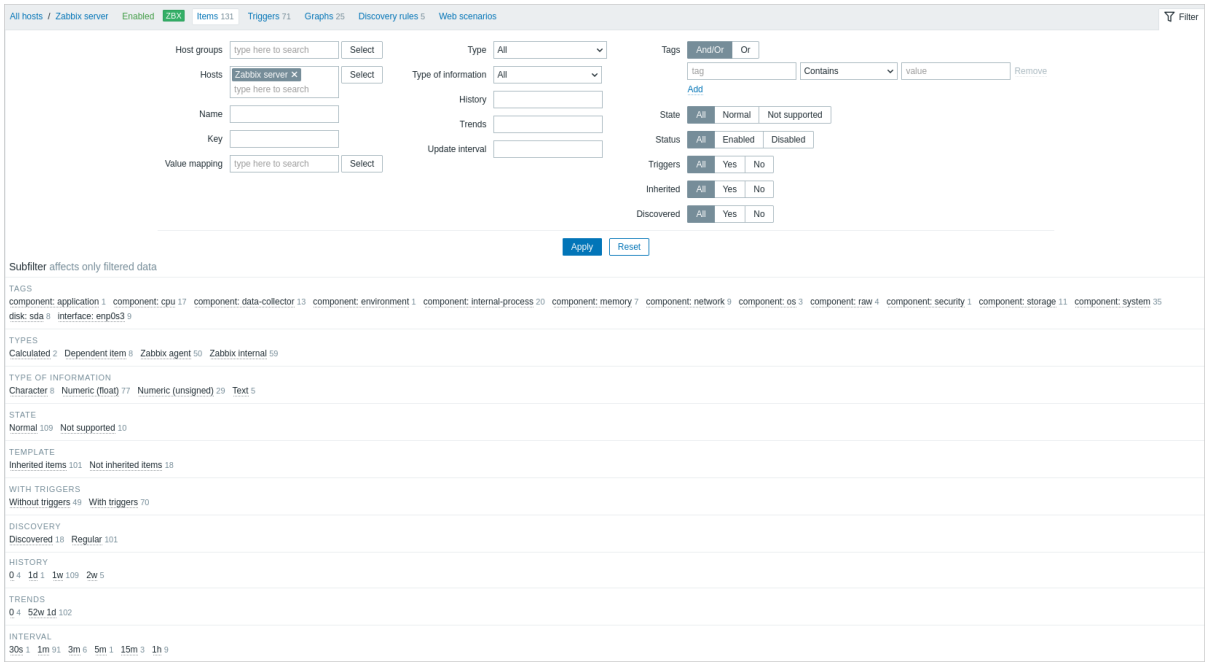
- *Ativar* - altera o status do item para *Ativado*
- *Desativar* - altera o status do item para *Desativado*
- *Executar agora* - executa uma verificação para novos valores de item imediatamente. Suportado apenas para verificações **passivas** (veja mais detalhes). Observe que, ao verificar os valores 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 os dados de histórico e tendências dos itens.
- *Copiar* - copia os itens para outros hosts ou templates.
- *Atualização em massa* - atualiza várias propriedades para 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.



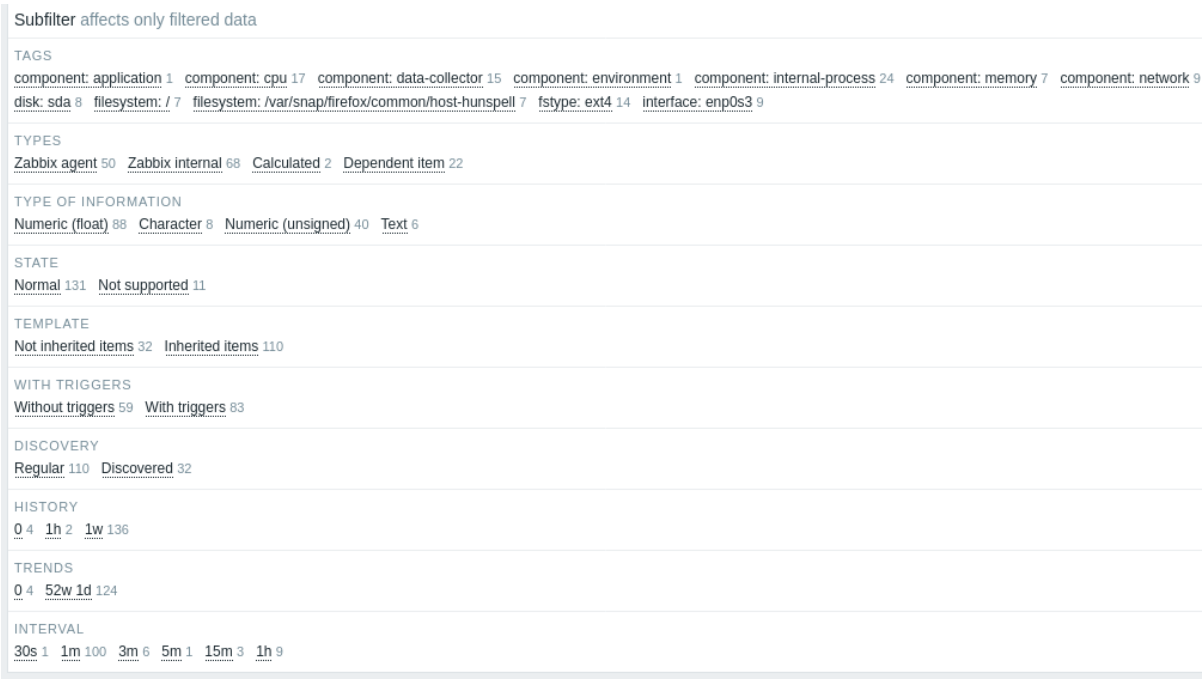
Parâmetro	Descrição
Grupos de hosts	Filtra por um ou mais grupos de hosts. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados. Grupos de hosts contendo apenas templates não podem ser selecionados.
Hosts	Filtra por um ou mais hosts.
Nome	Filtra pelo nome do item.
Chave	Filtra pela chave do item.
Mapeamento de valores	Filtra pelo mapa de valores utilizado. Este parâmetro não é exibido se a opção *Hosts* estiver vazia.
Tipo	Filtra pelo tipo de item (agent Zabbix, agent SNMP, etc.).
Tipo de informação	Filtra pelo tipo de informação (Numérico sem sinal, ponto flutuante, etc.).

Parâmetro	Descrição
Histórico	Filtra por quanto tempo o histórico do item é mantido.
Tendências	Filtra por quanto tempo as tendências do item são mantidas.
Intervalo de atualização	Filtra pelo 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. 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
Estado	Filtra pelo estado do item - *Normal* ou *Não suportado*.
Status	Filtra pelo status do item - *Habilitado* ou *Desabilitado*.
Triggers	Filtra items com (ou sem) triggers.
Herdado	Filtra items herdados (ou não herdados) de um template.
Descoberta	Filtra 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, é necessário aumentar o valor do parâmetro *Limite para resultados de busca 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 das opções de filtragem disponíveis e seus números de contagem.

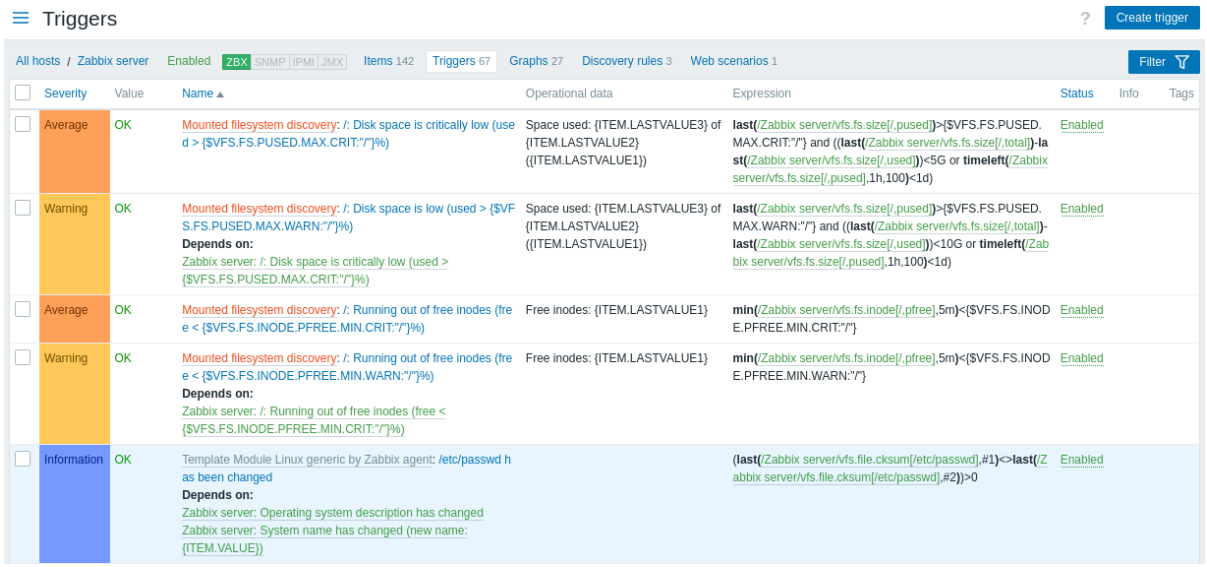
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de três pontos é exibido no final; ele não é clicável.

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 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. 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 será exibido antes do nome do trigger como um link cinza. Clicar no link do template abrirá 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 abre 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*. Clicando 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 de ferramenta fornece detalhes sobre seu status.

Coluna	Descrição
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.
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:

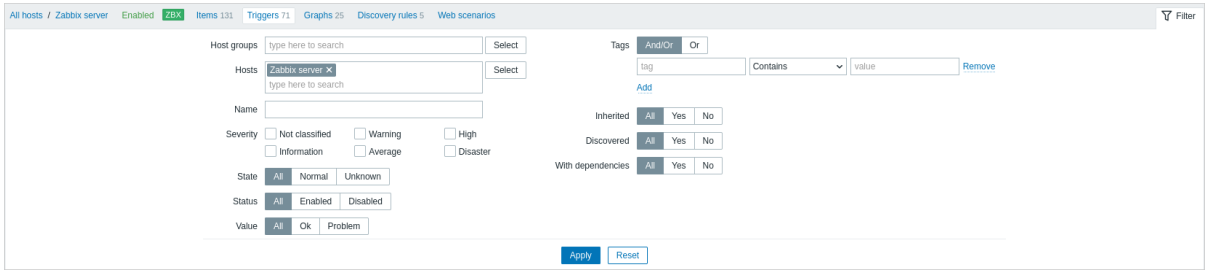
- *Ativar* - altera o status do trigger para *Ativado*.
- *Desativar* - altera o status do trigger para *Desativado*.
- *Copiar* - copia os triggers para outros hosts ou templates.
- *Atualização em massa* - atualiza várias propriedades de vários triggers de uma só vez.
- *Excluir* - exclui os triggers.

Para usar essas opções, marque as caixas de seleção antes dos triggers correspondentes e clique no botão desejado.

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.

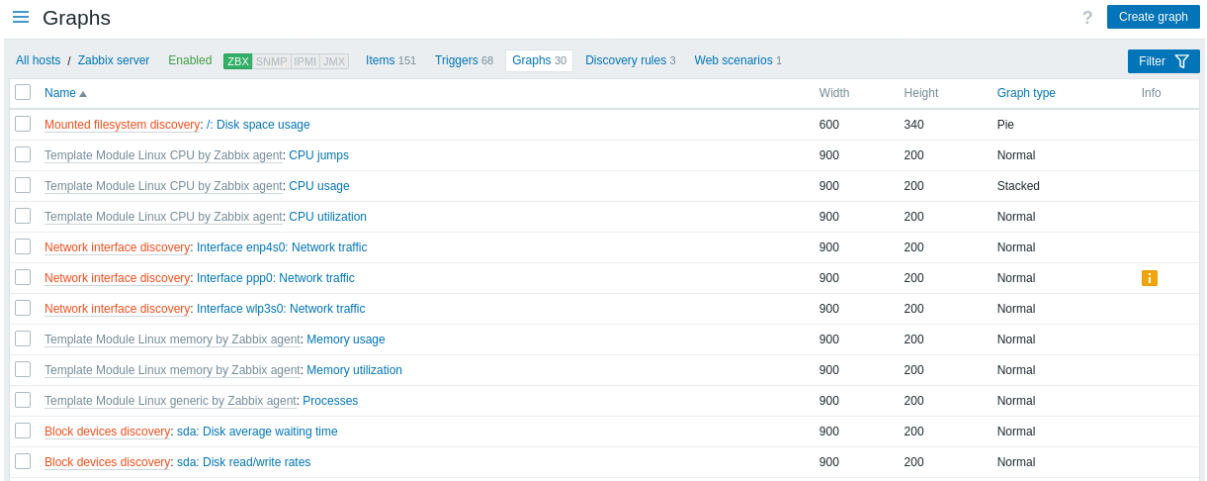
Parâmetro	Descrição
Tags	Filtra pelo nome e valor da tag do trigger. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - 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. Clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida apenas se vários hosts ou nenhum host estiver selecionado no filtro.

Coluna	Descrição
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 pertence a um template, o nome do template é exibido antes do nome do gráfico como um link cinza. Clicar no link do template abrirá 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 abre 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 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 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* - copia os gráficos para outros hosts ou templates
- *Excluir* - exclui os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos e clique no botão desejado.

Usando filtro

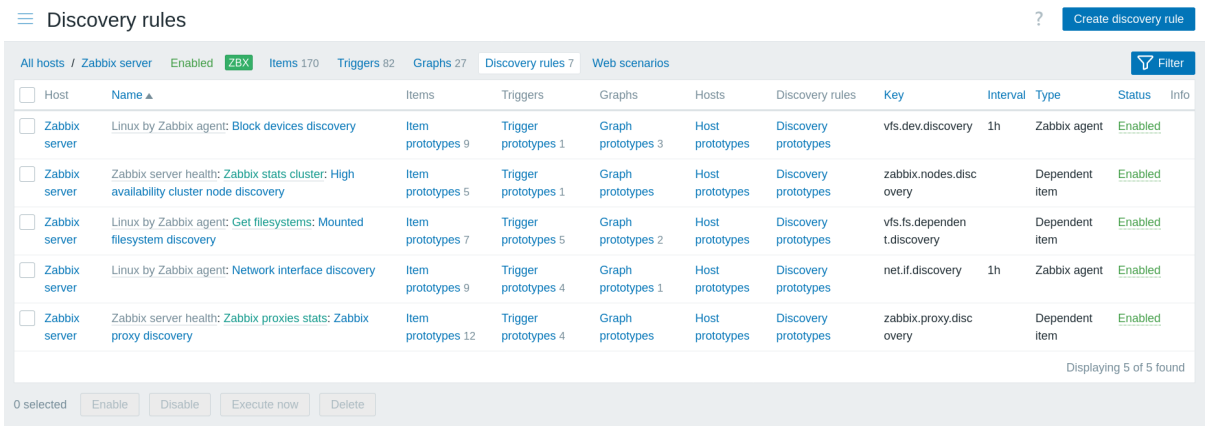
Você pode filtrar gráficos por grupo de hosts e host. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

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. Clicar no nome do host abre o formulário de configuração do host. Na ausência de um nome visível, o nome técnico do host é exibido.

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 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:

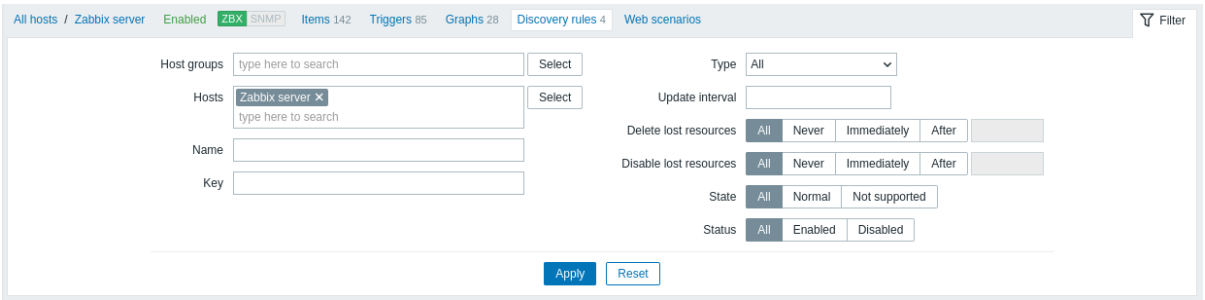
- *Habilitar* - altera o status da regra de descoberta de baixo nível para *Habilitado*.
- *Desabilitar* - altera o status da regra de descoberta de baixo nível para *Desabilitado*.
- *Executar agora* - executa a descoberta com base nas regras de descoberta imediatamente. Veja mais detalhes. Observe 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.



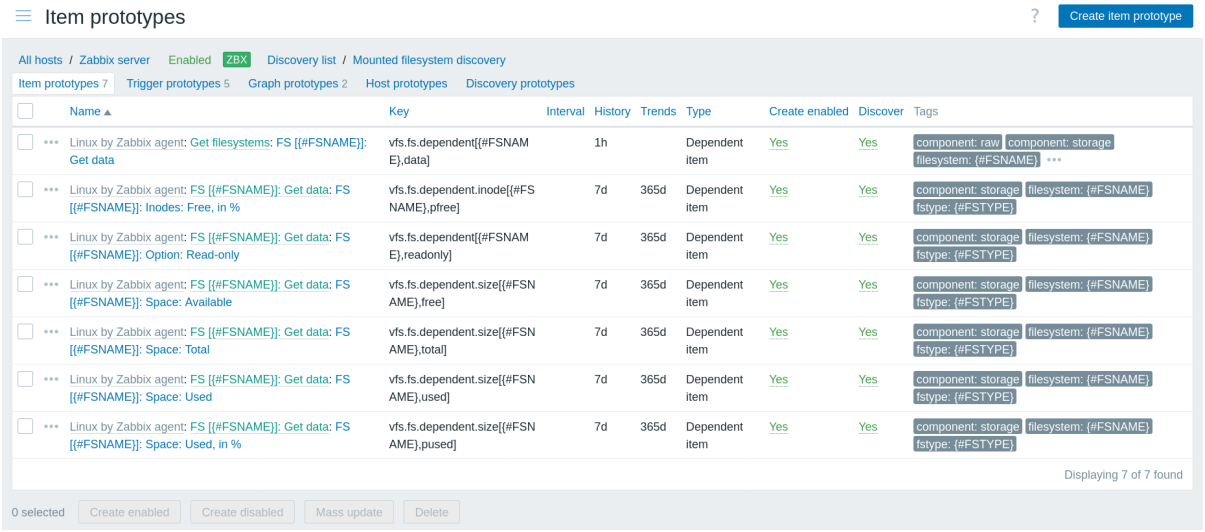
Parâmetro	Descrição
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados.
Hosts	Filtrar por um ou mais hosts.
Nome	Filtrar pelo nome da regra de descoberta.
Chave	Filtrar pela chave do item de descoberta.

Parâmetro	Descrição
Tipo	Filtrar pelo tipo do item de descoberta.
Intervalo de atualização	Filtrar pelo intervalo de atualização. Não disponível para Zabbix trapper e itens dependentes.
Excluir recursos perdidos	Filtrar pelo período de *Excluir recursos perdidos*.
Desabilitar recursos perdidos	Filtrar pelo período de *Desabilitar recursos perdidos*.
SNMP OID	Filtrar por SNMP OID. Disponível apenas se *SNMP agent* estiver selecionado como tipo.
Estado	Filtrar pelo estado da regra de descoberta (Todos/Normal/Não suportado).
Status	Filtrar pelo 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 items 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 pertencer a um template, o nome do template será 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 será 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:

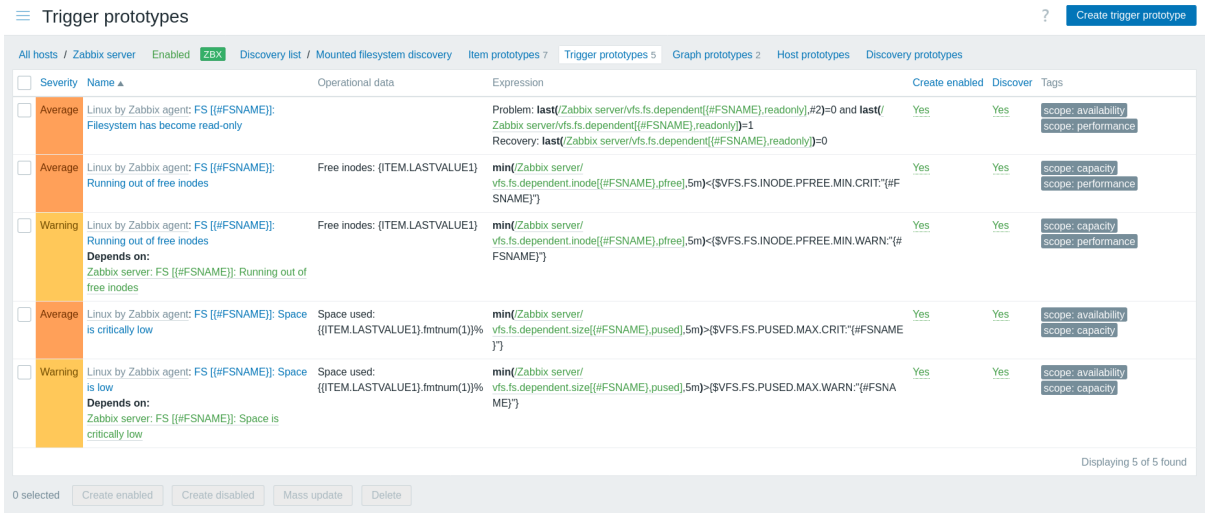
- *Criar habilitado* - cria esses items como *Habilitado*
- *Criar desabilitado* - cria esses items como *Desabilitado*
- *Atualização em massa* - atualiza em massa esses protótipos de item
- *Excluir* - exclui esses protótipos de item

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 pertence a um template vinculado, o nome do template é 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:

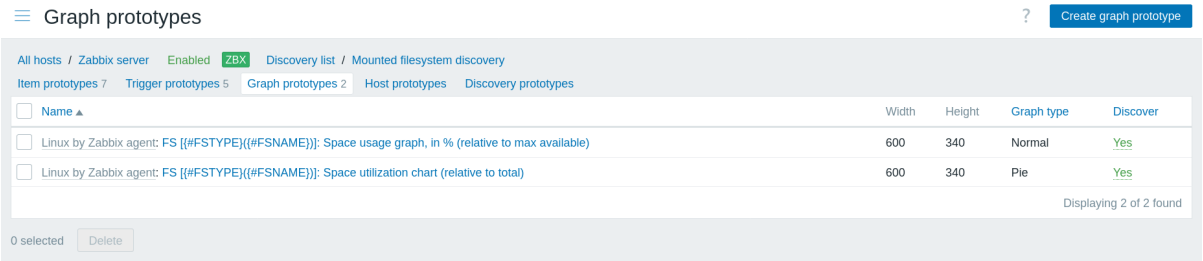
- *Criar habilitado* - criar esses triggers como *Habilitado*
- *Criar desabilitado* - criar esses triggers como *Desabilitado*
- *Atualização em massa* - atualizar em massa esses protótipos de trigger
- *Excluir* - excluir esses protótipos de trigger

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:

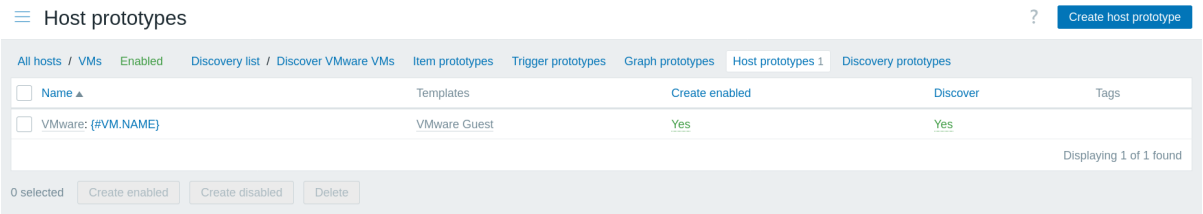
- *Excluir* - exclui esses protótipos de gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráficos e clique no botão necessário.

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	Criar o host com base neste protótipo como: **Sim** - ativado **Não** - desativado. Você pode alternar entre 'Sim' e 'Não' clicando sobre eles.
Descobrir	Descobrir o host com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando sobre eles.
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:

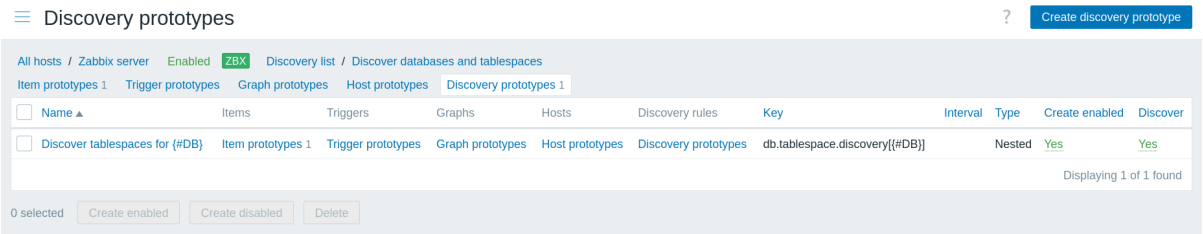
- *Criar habilitado* - cria esses hosts como *Habilitado*
- *Criar desabilitado* - cria esses hosts como *Desabilitado*
- *Excluir* - exclui esses protótipos de host

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e clique no botão desejado.

5 Protótipos de descoberta

Visão geral

Nesta seção, os protótipos de descoberta de uma regra de descoberta de baixo nível no host são exibidos. Os protótipos de descoberta são regras de descoberta aninhadas da regra de descoberta pai.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de descoberta, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de descoberta. Se o protótipo de descoberta pertence a um template vinculado, o nome do template é exibido antes do nome do host como um link cinza. Clicar no link do template abre a lista de protótipos de descoberta no nível do template vinculado.
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.
Regras de descoberta	Um link para a lista de protótipos de descoberta é exibido. O número de protótipos de descoberta existentes é exibido em cinza.
Chave	A chave do item usada para descoberta é exibida.

Coluna	Descrição
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).
Criar habilitado	O status de criação do protótipo de descoberta é exibido - criar habilitado (*Sim*) ou criar desabilitado (*Não*). Clicando no status você pode alterá-lo - de Sim para Não (e vice-versa).
Descobrir	O status de descoberta do protótipo de descoberta é exibido - descobrir (*Sim*) ou não descobrir (*Não*). Clicando no status você pode alterá-lo - de Sim para Não (e vice-versa).

Para configurar um novo protótipo de descoberta, clique no botão *Criar protótipo de descoberta* 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 protótipos de descoberta como *Habilitado*
- *Criar desabilitado* - criar esses protótipos de descoberta como *Desabilitado*
- *Excluir* - excluir esses protótipos de descoberta

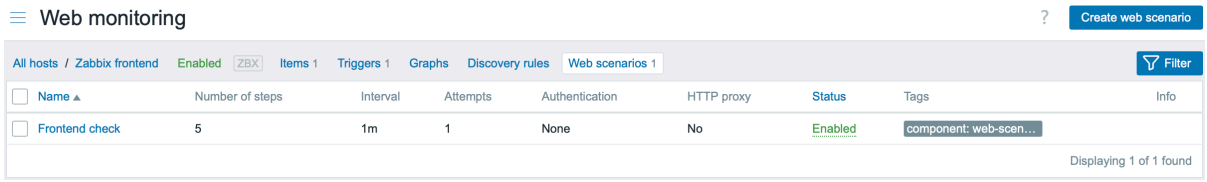
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de descoberta e clique no botão desejado.

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 pertencer a um template, o nome do template será 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 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:

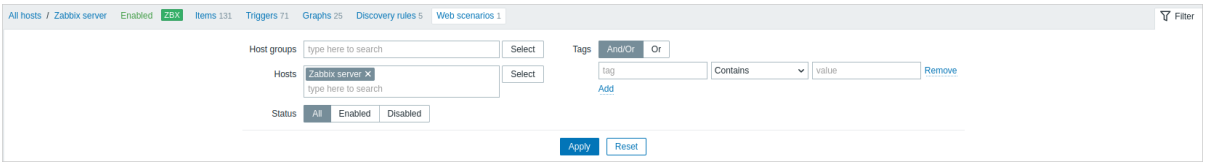
- *Ativar* - altera o status do cenário para *Ativado*
- *Desativar* - altera o status do cenário para *Desativado*
- *Limpar histórico e tendências* - limpa os dados de histórico e tendências dos cenários
- *Excluir* - exclui os cenários web

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 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 se torna disponível onde você pode filtrar cenários por grupo de host, host, status e tags.

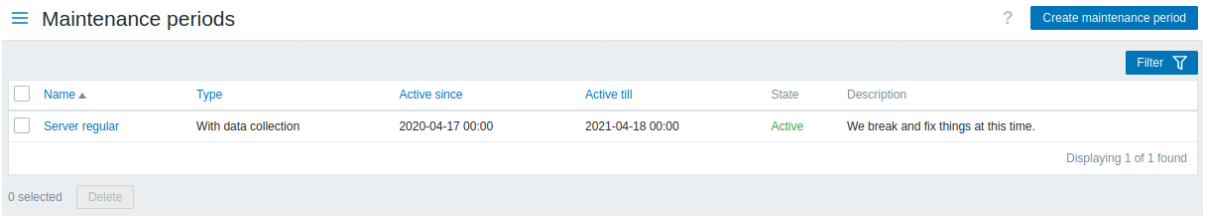


5 Manutenção

Visão geral

Na seção *Coleta de dados* → *Manutenção*, os usuários podem configurar e manter períodos de manutenção para hosts.

Uma lista 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 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 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.

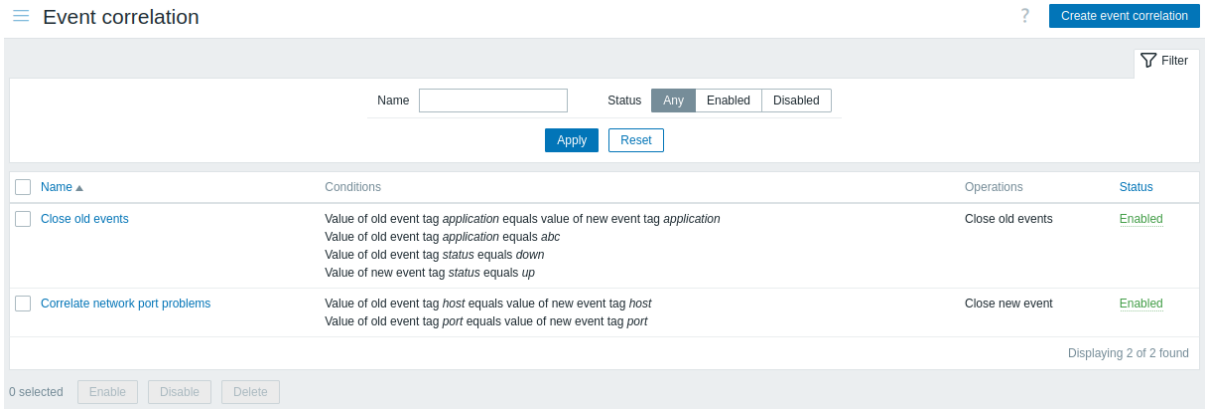
Cálculo de filas durante a manutenção

Attention:
O proxy Zabbix não tem conhecimento dos períodos de manutenção; veja Cálculo de filas durante a manutenção para 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 - *Habilitado* ou *Desabilitado*. 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 necessário.

Usando filtro

Você pode usar o filtro para exibir apenas as regras de correlação que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

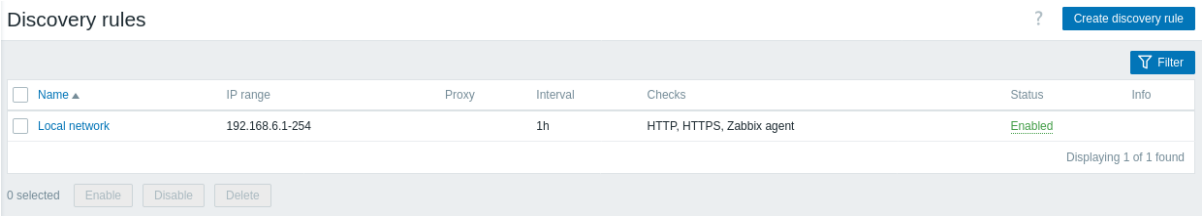
O link *Filtro* está disponível acima da lista de regras de correlação. Se você clicar nele, um filtro ficará disponível onde você pode filtrar as regras de correlação por nome e status.

7 Descoberta

Visão geral

Na seção *Coleta de dados* → *Descoberta* os usuários podem configurar e manter regras de descoberta.

Uma lista 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 - *Ativado* ou *Desativado*. 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:

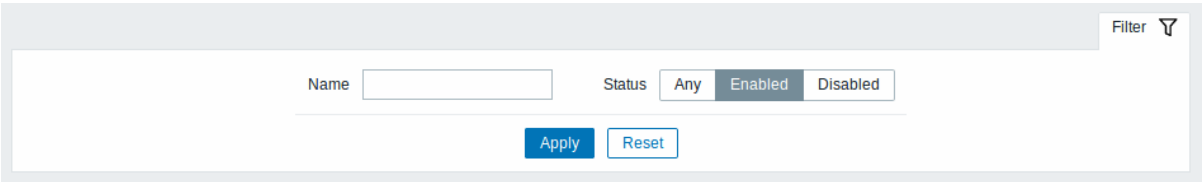
- *Ativar* - altera o status da regra de descoberta para *Ativado*
- *Desativar* - altera o status da regra de descoberta para *Desativado*
- *Excluir* - exclui as regras de descoberta

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 de 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.

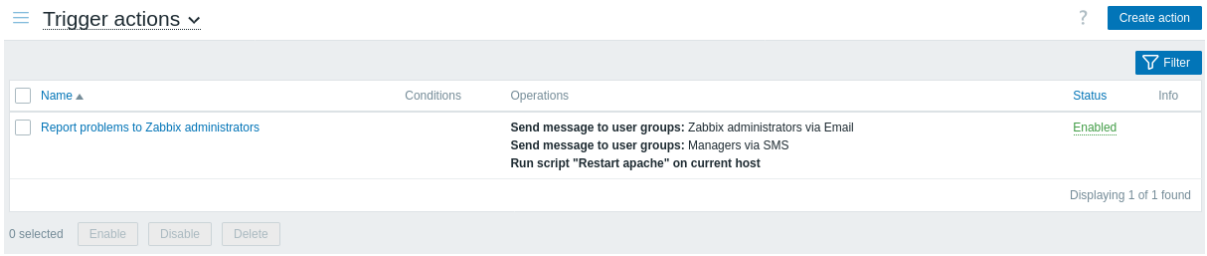
1 Ações

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, auto-registro, 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 de ação ausentes ou condições 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.

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-escrita a grupos de hosts, hosts, templates e triggers nas condições da ação
- O usuário tem acesso de leitura-escrita 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* - altera o status da ação para *Ativado*
- *Desativar* - altera o status da ação para *Desativado*
- *Excluir* - exclui as ações

Para usar essas opções, marque as caixas de seleção antes das respectivas ações e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as ações 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 ações. Se você clicar nele, um filtro ficará disponível onde você pode filtrar ações por nome e status.

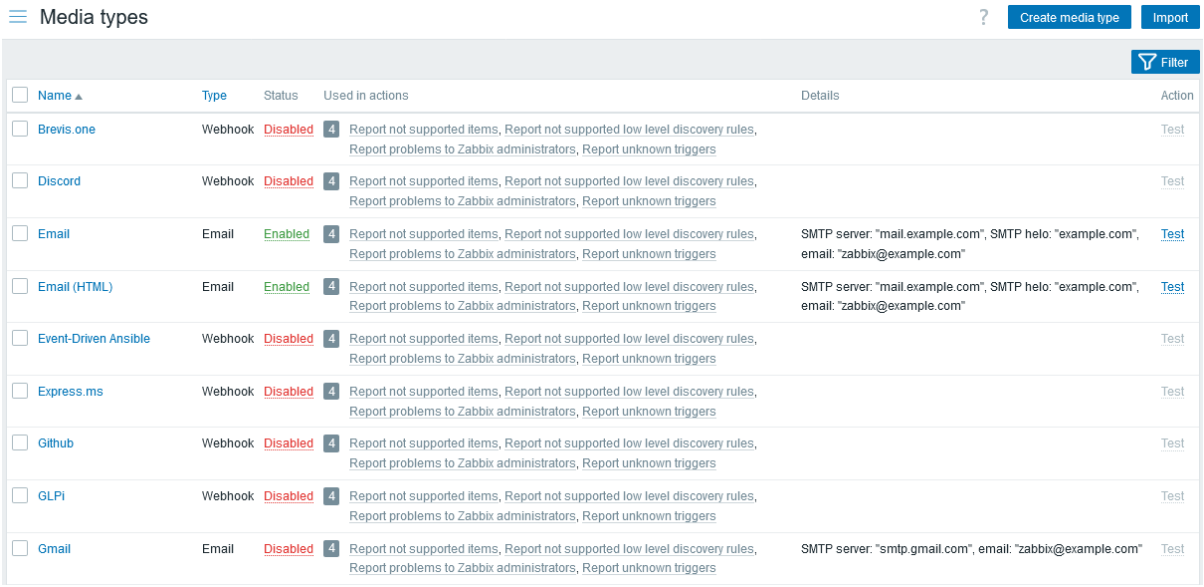
2 Tipos de mídia

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 listagem 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 da 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, 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: Testando tipos de mídia.

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:

- *Habilitar* - altera o status do tipo de mídia para *Habilitado*
- *Desabilitar* - altera o status do tipo de mídia para *Desabilitado*
- *Exportar* - exporta os tipos de mídia para um arquivo YAML, XML ou JSON
- *Excluir* - exclui os tipos de mídia

Para usar essas opções, marque as caixas de seleção antes dos respectivos tipos de mídia e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os tipos de mídia 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 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, 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.

Scripts globais, dependendo do escopo configurado e também das permissões do usuário, estão disponíveis para execução:

- a partir do menu de host em vários locais do frontend (*Dashboard*, *Problemas*, *Últimos dados*, *Mapas*, etc.)
- a partir do menu de evento
- podem ser executados como uma operação de ação

Os scripts são executados no agent Zabbix, server Zabbix (proxy) ou apenas no server Zabbix. Veja também Execução de comandos.

Tanto no agent Zabbix quanto no proxy Zabbix, scripts remotos estão desabilitados por padrão. Eles podem ser habilitados por:

- Para comandos remotos executados no agent Zabbix:
  - 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 proxy Zabbix:
  - **Atenção: Não é necessário habilitar comandos remotos no proxy Zabbix se os comandos remotos forem executados no agent Zabbix que é monitorado pelo proxy Zabbix.** Se, no entanto, for necessário executar comandos remotos no proxy Zabbix, defina o parâmetro *EnableRemoteCommands* como '1' na configuração do proxy.

A execução de scripts globais no server Zabbix pode ser desabilitada definindo EnableGlobalScripts=0 na configuração do server. Para novas instalações, desde o Zabbix 7.0, a execução de scripts globais no server Zabbix está desabilitada por padrão.

Uma lista 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, 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 agent Zabbix, proxy Zabbix ou server, ou apenas no server Zabbix.
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

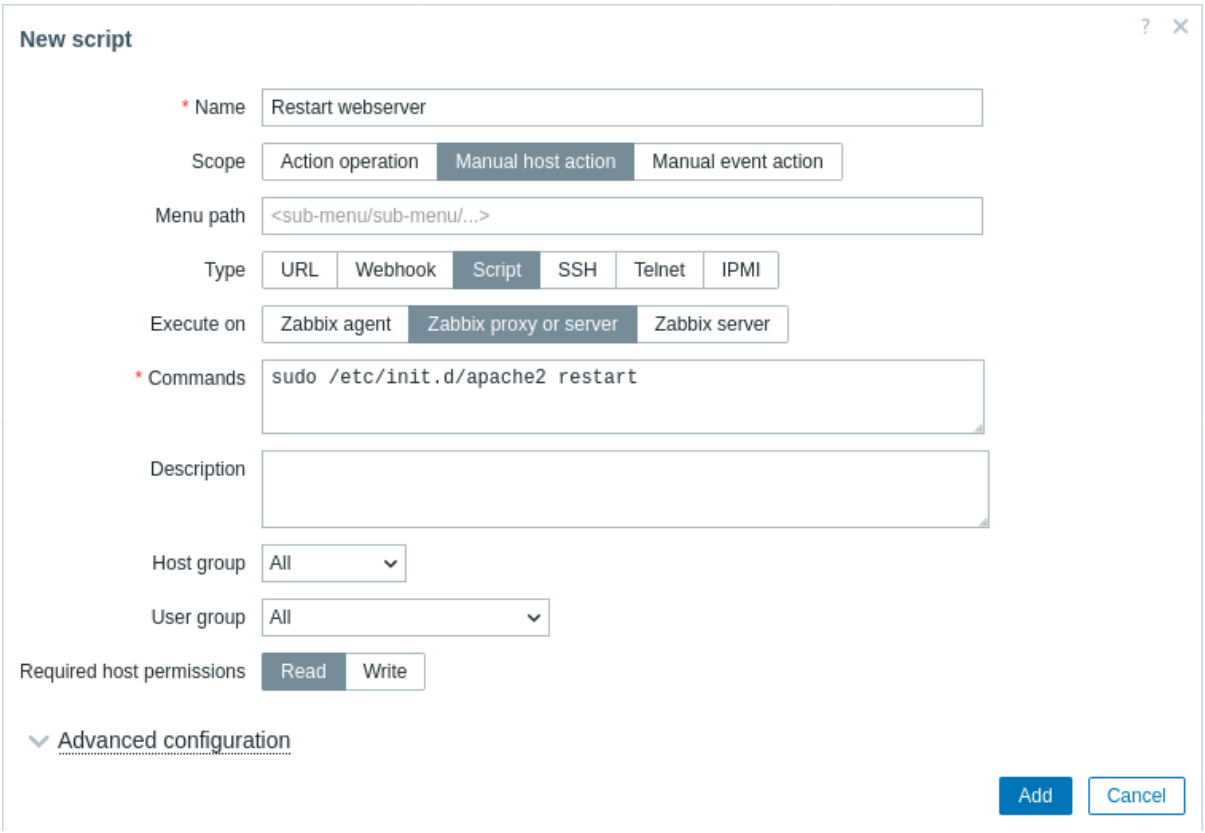
Para usar esta opção, marque as caixas de seleção antes dos respectivos scripts e clique em *Excluir*.

Usando o filtro

Você pode usar o filtro para exibir apenas os scripts que lhe interessam. 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



Atributos do script:

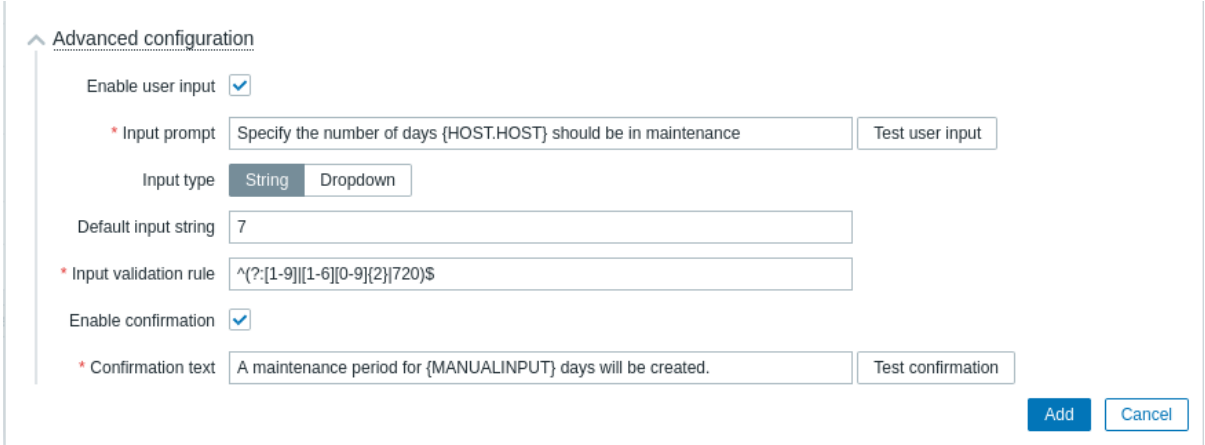
Parâmetro	Descrição
Nome	Nome único do script. Ex.: `Limpar sistema de arquivos /tmp`

Parâmetro	Descrição
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 for realmente usado em uma ação, seu escopo não poderá ser alterado de 'operação de ação'. **Suporte a macro** O escopo afeta a gama de macros disponíveis. Por exemplo, macros relacionadas ao usuário ({USER.*}) são suportadas em scripts para permitir o envio de informações sobre o usuário que lançou 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 ação manual de evento. Para descobrir 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 respectivo botão 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	Timeout de execução do JavaScript (1-60s, padrão 30s). Sufixos de tempo são suportados, por exemplo, 30s, 1m.
Tipo de script: **Script**	
Executar em	Clique no respectivo botão 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. Ele será executado no proxy se habilitado por EnableRemoteCommands. Ele será executado no server se scripts globais estiverem habilitados pelo parâmetro EnableGlobalScripts do server. **Zabbix server** - o script será executado apenas pelo Zabbix server. Esta opção não estará disponível se scripts globais estiverem desabilitados pelo parâmetro EnableGlobalScripts do server.

Parâmetro	Descrição
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.
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.
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 de dropdown.
Opções do dropdown	Insira valores únicos para o dropdown de entrada do usuário em uma lista separada por vírgulas. Para incluir uma opção vazia no dropdown, 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 que possam 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 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 número inteiro diferente ou um endereço de URL diferente para o script durante a execução.

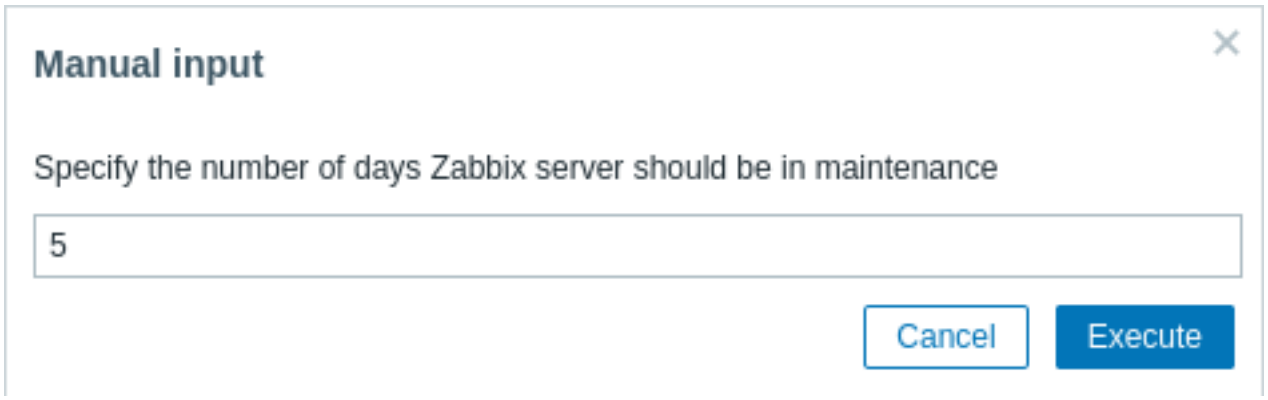
Para habilitar a entrada manual do usuário:

- use a macro {MANUALINPUT} no script (comandos, script, parâmetro do script) onde necessário; ou no campo URL de scripts de URL;

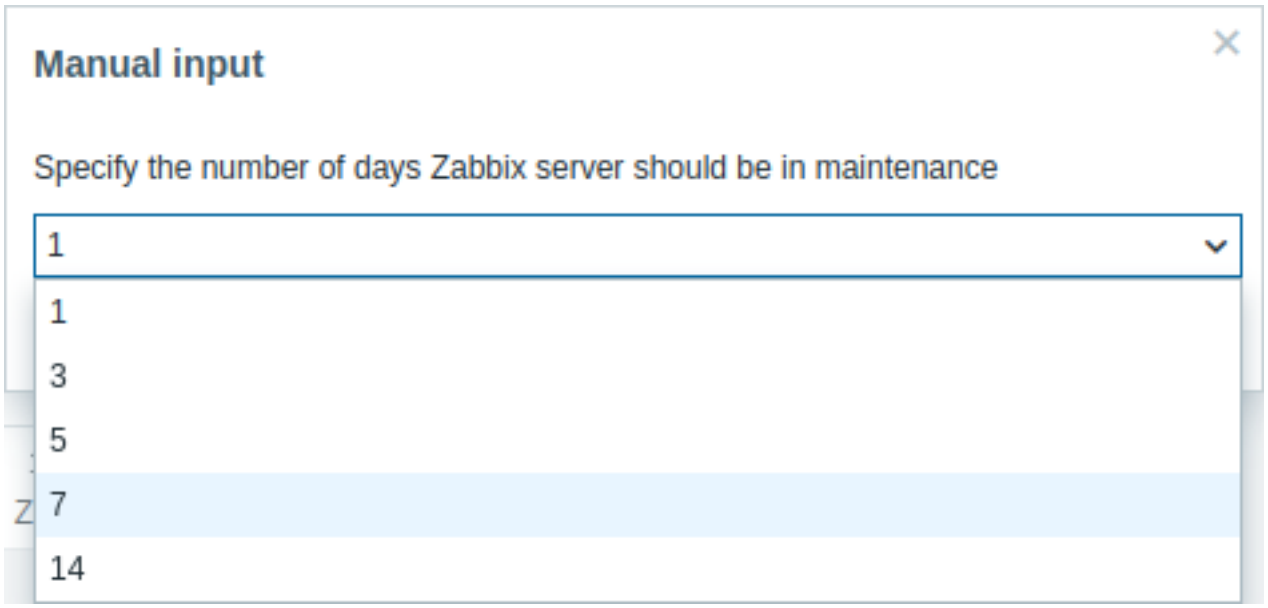
- na configuração avançada do script, habilite a entrada manual do usuário e configure as opções de entrada.

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 de host' ou 'Ação manual de 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:

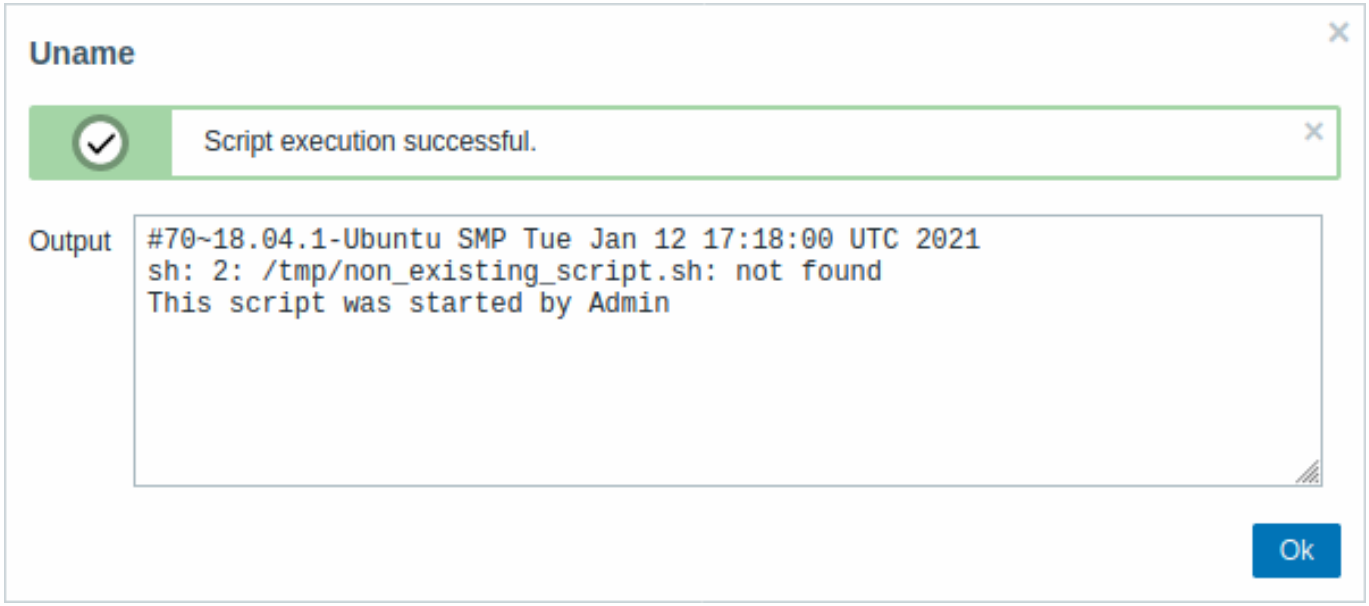
- Se o script for finalizado com sucesso (código de saída 0), o valor de retorno é limitado a 16MB (incluindo espaços em branco à direita, que são truncados).
- Se o script for finalizado com erro (código de saída diferente de zero), o valor de retorno é um erro padrão limitado a 2KB.

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 "Este script foi iniciado por {USER.USERNAME}"
```



O resultado do script não exibe o próprio script.

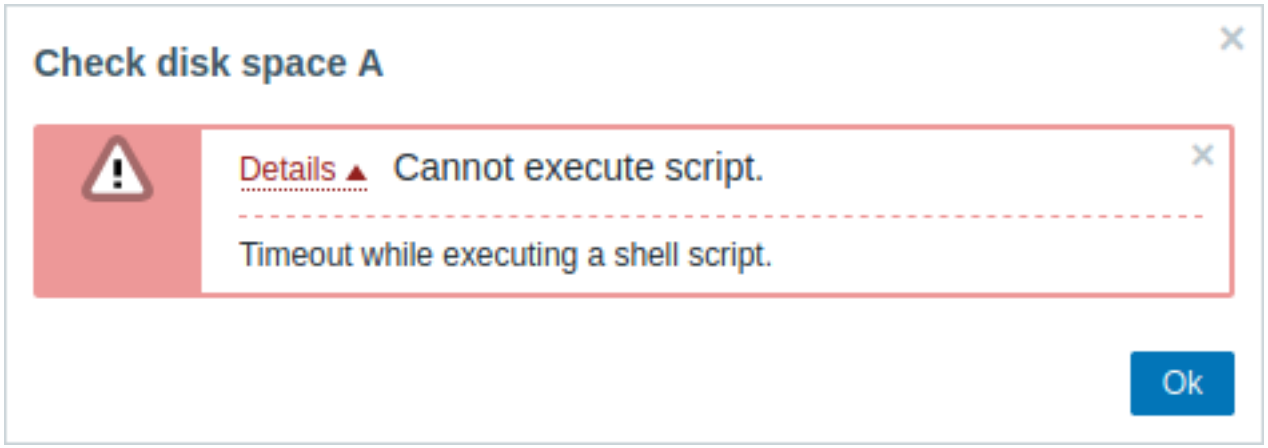
Tempo limite do script

Zabbix agent

Você pode se deparar com uma situação em que ocorre um timeout ao executar um script.

Veja abaixo um exemplo de script rodando no Zabbix agent e a janela de resultado:

```
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 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 agent ativo é suportada a partir do Zabbix agent 7.0.

Caso o parâmetro `Timeout` tenha sido alterado na configuração do Zabbix agent, a seguinte mensagem de erro aparecerá:

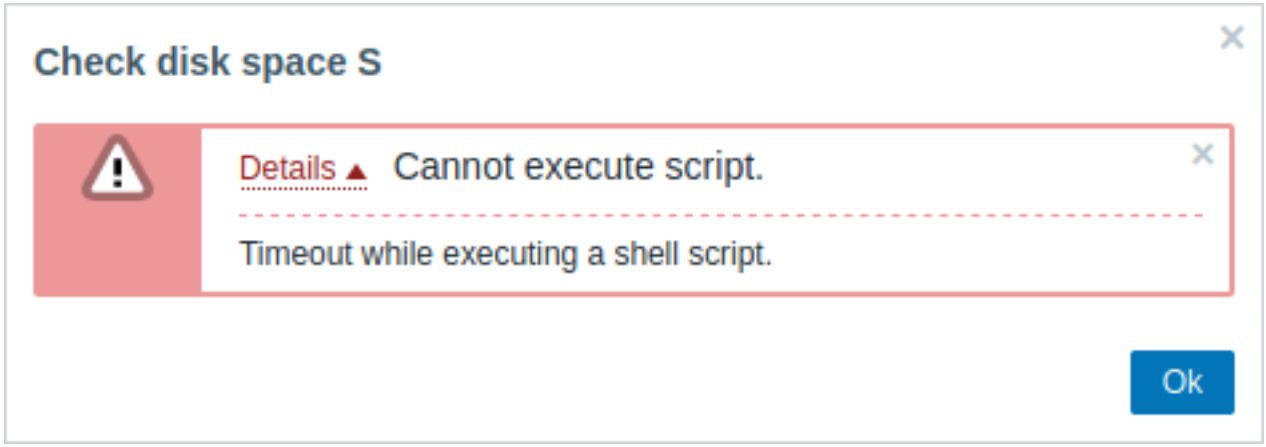
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

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 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.

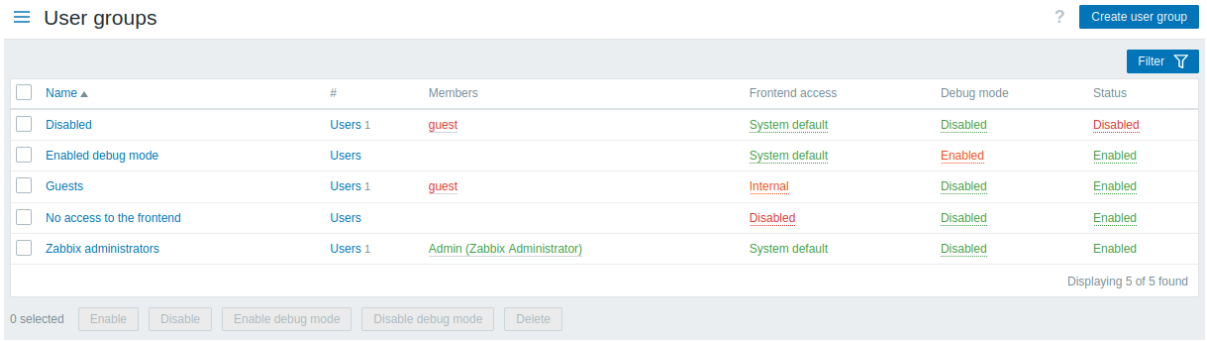
1 Grupos de usuários

Visão geral

Na seção *Usuários* → *Grupos de usuários* são mantidos os grupos de usuários do sistema.

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 de usuários.
#	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 de usuários (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.

Coluna	Descrição
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:

- *Habilitar* - altera o status do grupo de usuários para *Habilitado*
- *Desabilitar* - altera o status do grupo de usuários para *Desabilitado*
- *Habilitar modo de depuração* - habilita o modo de depuração para os grupos de usuários
- *Desabilitar modo de depuração* - desabilita 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 dos respectivos grupos de usuários e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os grupos de usuários que são do seu interesse. Para melhor desempenho na busca, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de grupos de usuários. Se você clicar nele, um filtro ficará disponível onde você pode filtrar os grupos de usuários por nome e status.

2 Funções de 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 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 esmaecidas; os usuários não poderão acessar o elemento mesmo inserindo uma URL direta para esse elemento no navegador.

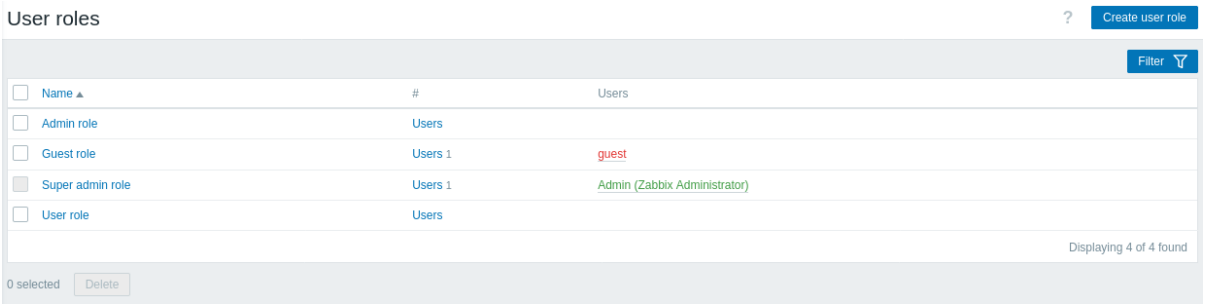
> **Note:**
> Restringir o acesso a alguns elementos da interface do usuário apenas impede a abertura dessa página — não remove a capacidade de acessar os dados subjacentes em outras partes da interface.

As funções de usuário podem ser atribuídas aos 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

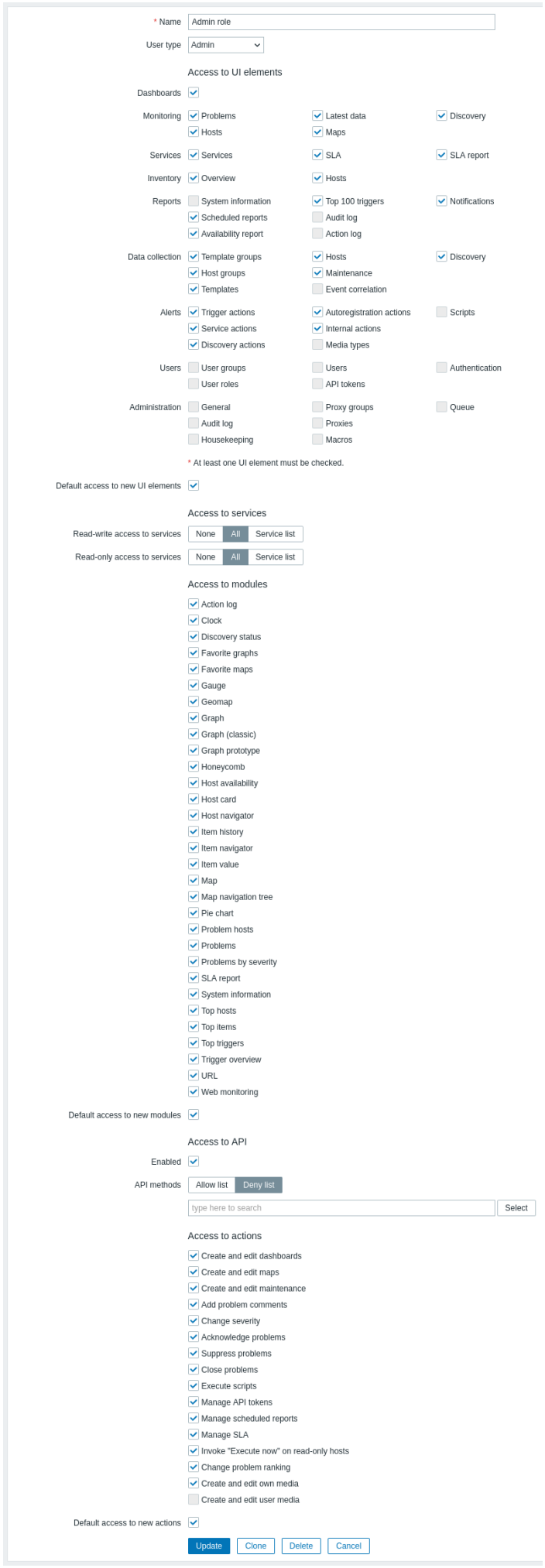


Estas são baseadas nos principais tipos de usuário do Zabbix. A lista de todos os usuários atribuídos à respectiva função é exibida. Os usuários incluídos em grupos desabilitados são indicados em vermelho. A *Função de convidado* é uma função do tipo usuário com a única permissão de visualizar algumas seções do frontend.

Note:
A *Função de superadministrador* padrão não pode ser modificada ou excluída, porque pelo menos um usuário superadministrador com privilégios ilimitados deve existir 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
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Default access to new actions
Update
Clone
Delete
Cancel

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** - nenhum acesso **Todos** - acesso de leitura e gravação a todos os serviços **Lista de serviços** - selecione 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** - nenhum acesso **Todos** - acesso somente leitura a todos os serviços **Lista de serviços** - selecione 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 o 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 na seção *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 não estiver marcada, 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 Criar e editar manutenção Adicionar comentários ao problema Alterar gravidade Reconhecer problemas Suprimir problemas Fechar problemas	Desmarcar esta caixa também revogará os direitos de usar os métodos da API `.create`, `.update` e `.delete` para os elementos correspondentes. Desmarcar esta caixa também revogará os direitos de executar a ação correspondente via método da API `event.acknowledge`.	Sim	Sim	Sim Não Sim	Não
Executar scripts	Desmarcar esta caixa também revogará os direitos de usar o método da API `script.execute`.				
Gerenciar tokens da 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.				
Criar e editar mídia própria	Permitir criar/editar sua própria mídia.				
Criar e editar mídia do usuário	Permitir criar/editar mídia para usuários. Esta opção está disponível apenas para usuários Superadministradores.		Não	Não	
Acesso padrão a novas ações	Esta opção especifica como novas ações estarão acessíveis após uma atualização do Zabbix.		Sim	Sim	

Veja também:

- Configurando um usuário

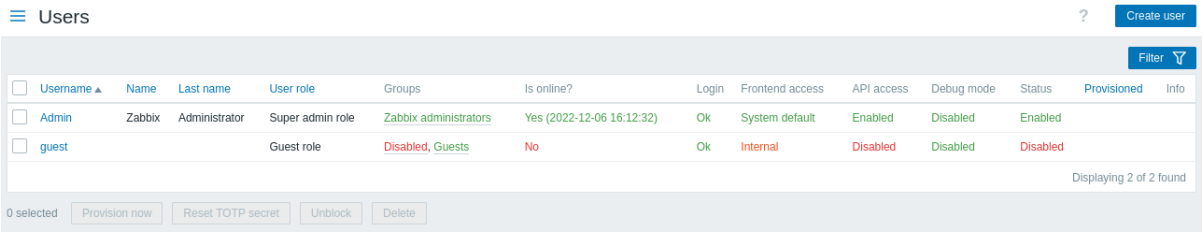
3 Usuários

Visão geral

Na seção *Usuários* → *Usuários*, os usuários do sistema são mantidos.

Usuários

Uma lista dos 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.
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 de 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 se torna disponível onde você pode filtrar usuários por nome de usuário, nome, sobrenome, função do usuário e grupo de usuários.

4 Tokens de 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 os tokens ou status (habilitado/desabilitado). Clique no status do token na lista para habilitar/desabilitar rapidamente um token. Você também pode habilitar/desabilitar tokens em massa 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 só pode ser atribuído 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 precisar criar um token em estado desabilitado.

Pressione *Adicionar* para criar um token.

Copie o valor do *Token de autenticação* e salve em um local seguro **antes de fechar a página**, depois pressione Fechar. O token aparecerá na lista.

Warning:
O valor do *Token de autenticação* não poderá ser visualizado novamente posteriormente. Ele só estará disponível imediatamente após a criação do 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 utiliza 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, acesse a aba *LDAP* e configure os parâmetros do LDAP, depois retorne à aba *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.

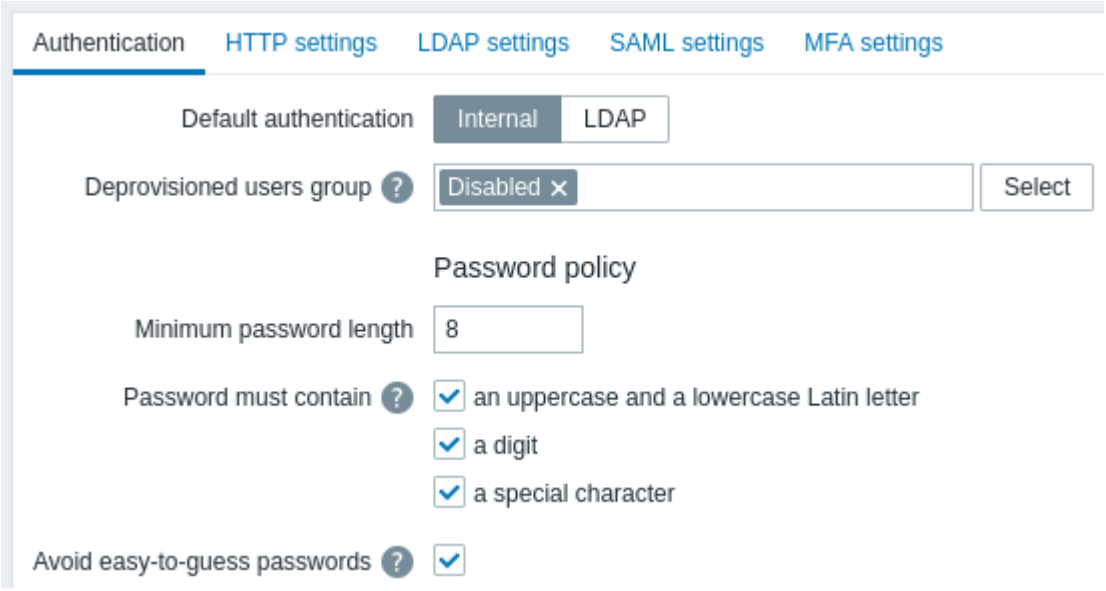
Veja também:

- 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.
A senha deve conter	Marque uma ou várias caixas de seleção para exigir o uso dos 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 dos 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

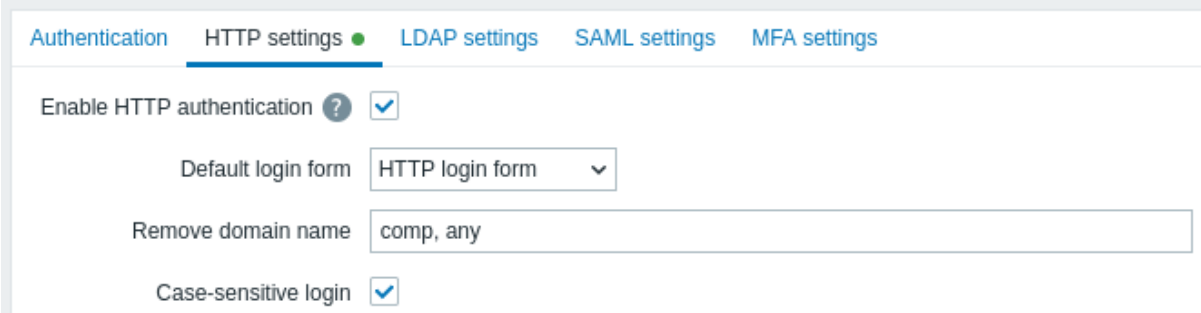
Visão geral

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 o usuário também deve existir no Zabbix, porém sua senha do Zabbix não será usada.

Attention:
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 está 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.
Formulário de login padrão	Especifique se deseja direcionar usuários não autenticados para: **Formulário de login do Zabbix** - página de login padrão do Zabbix. **Formulário de login HTTP** - página de login HTTP. Recomenda-se habilitar a autenticação baseada em 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, autenticar 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 regular 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 o Microsoft Active Directory e o OpenLDAP.

Se apenas o login LDAP estiver configurado, o usuário também deve existir no Zabbix, porém, sua senha do Zabbix não será usada. Se a autenticação for bem-sucedida, o Zabbix irá corresponder 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. Neste 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* 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.

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.

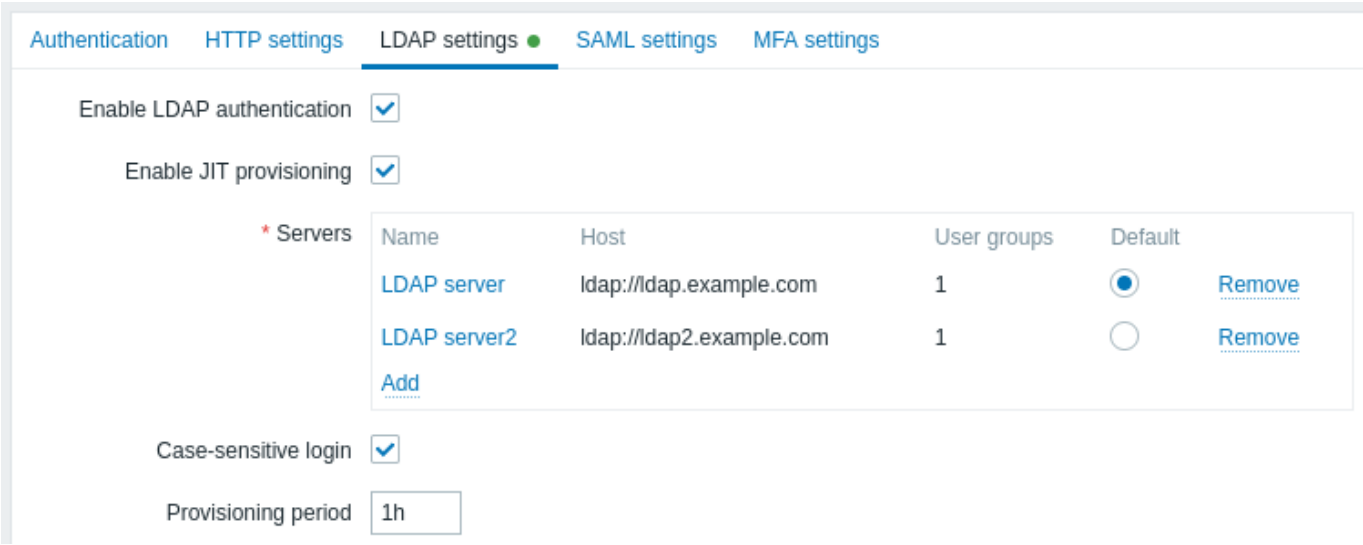
O provisionamento JIT do LDAP está disponível apenas quando o LDAP está configurado para usar "anônimo" ou "usuário especial" para binding. Para binding direto do usuário, o provisionamento será feito apenas para a ação de login do usuário, pois a senha do usuário de login é usada para esse tipo de binding.

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.

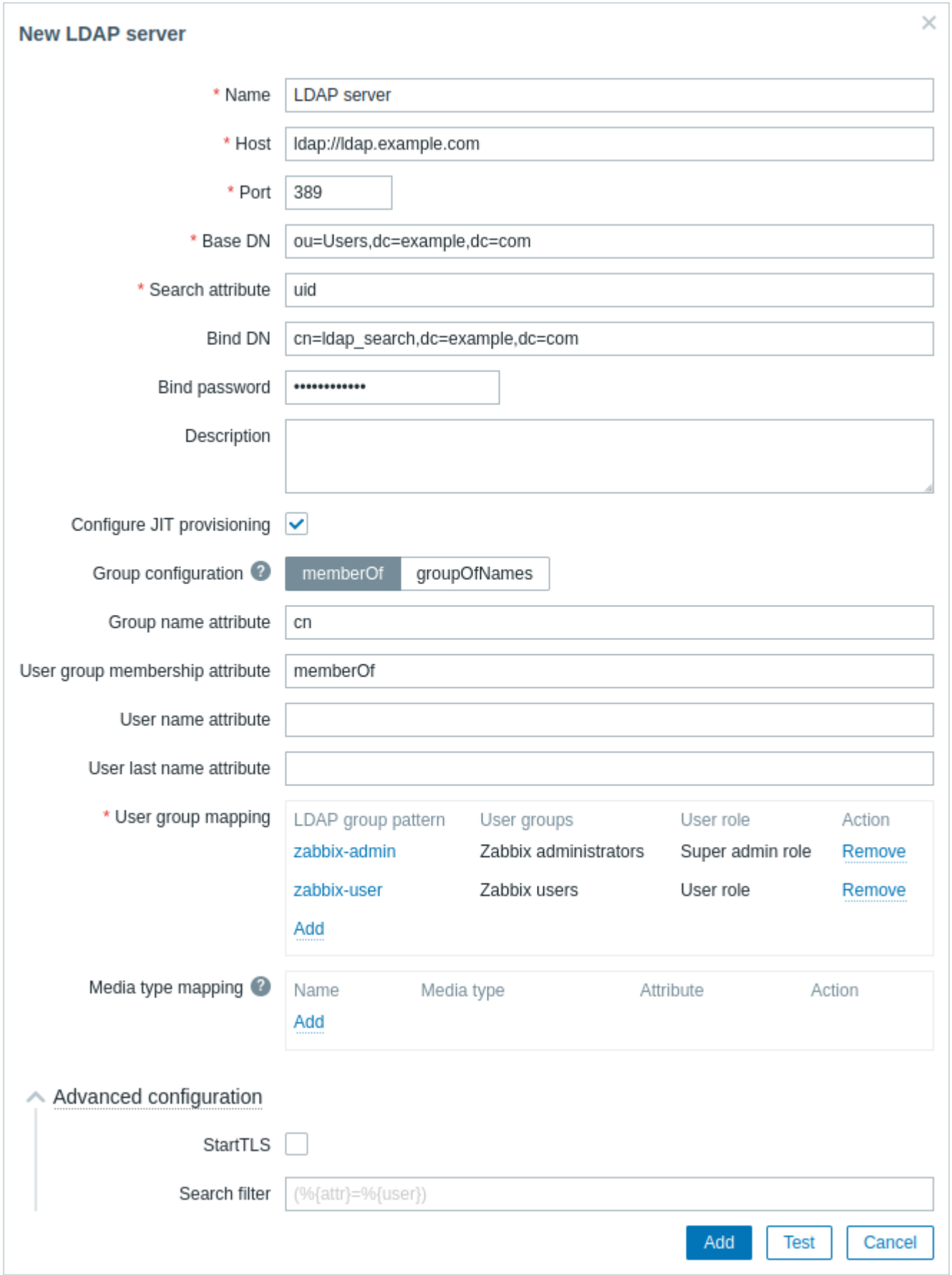
Configuração



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:

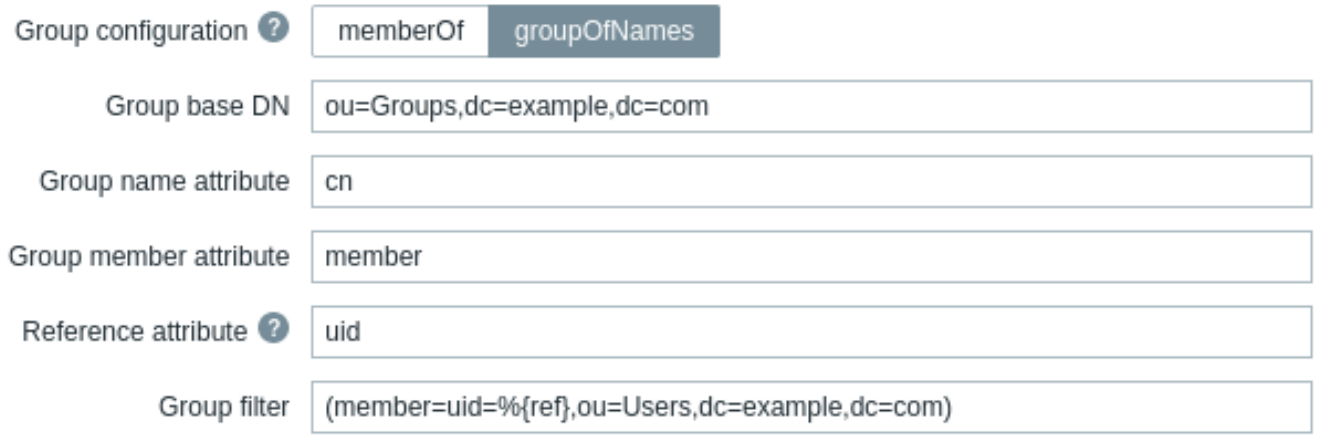
Parâmetro	Descrição
Nome	Nome do servidor LDAP na configuração do Zabbix.

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 uma URI LDAP completa 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 vinculação direta 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 vinculação 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) Também é suportada a vinculação anônima. Observe que a vinculação anônima potencialmente expõe a configuração do domínio para usuários não autorizados (informações sobre usuários, computadores, servidores, grupos, serviços, etc.). Por motivos de segurança, desative as vinculações anônimas nos hosts LDAP e use acesso autenticado.
Senha de vinculação	Senha LDAP da conta para vinculação 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 terá 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 nível de permissão mais alto entre elas.
Mapeamento de tipo de mídia	Mapeie os atributos de mídia LDAP do usuário (por exemplo, e-mail) para a mídia de usuário do Zabbix para envio de notificações.
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 sensível a maiúsculas e minúsculas em um ambiente LDAP ou Microsoft Active Directory que não diferencia maiúsculas de minúsculas, 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 **vinculação direta 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 vinculação* em branco. Ao autenticar, o placeholder %{user} será substituído pelo nome de usuário inserido durante o login.

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*). Em seguida, 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 a "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 a vinculação 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 fornece mais segurança e não exige a alteração da *Senha de vinculação* 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 conseguir autenticar o usuário de teste.
Senha do usuário	Senha do usuário LDAP a ser testada.

3 SAML

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 usada. Se a autenticação for bem-sucedida, o Zabbix corresponderá um nome de usuário local ao 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 de usuário pode ser criada quando o usuário faz 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 usado, 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 a autenticação SAML no frontend.

Para usar a autenticação SAML, o Zabbix deve ser configurado da seguinte forma:

1. Forneça a chave privada e os certificados do SP. Um certificado IdP deve ser fornecido (colar ou escolher arquivo) para ativar o SAML via o frontend. Um certificado SP e uma chave privada SP devem ser fornecidos se as opções *Assinar* ou *Criptografar* forem selecionadas.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido em zabbix.conf.php, você pode colar o texto do certificado ou fazer upload do arquivo do certificado no frontend durante a configuração do SAML — nenhum arquivo é necessário no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido em zabbix.conf.php, o certificado deve estar disponível no sistema de arquivos (por padrão em ui/conf/certs ou no caminho configurado em zabbix.conf.php) e o frontend não armazenará certificados no banco de dados. Observe que se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será assumido.

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

Attention:
O material do certificado pode ser importado via o frontend colando o texto ou fazendo upload de arquivos. Embora importações não criptografadas sejam permitidas para conveniência em alguns ambientes, é altamente recomendável usar transporte criptografado (por exemplo, habilite HTTPS/TLS para o frontend do Zabbix). Quando certificados/chaves SAML são armazenados no banco de dados, habilite TLS/SSL para conexões com o banco de dados para proteger o tráfego entre o aplicativo e o banco de dados.
 Criptografe backups do banco de dados que possam conter certificados SAML ou chaves privadas e limite o acesso aos arquivos de backup e às tabelas do banco de dados que armazenam material SAML. Restrinja os privilégios do usuário do banco de dados ao mínimo necessário.

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.
URL do serviço SSO	A URL para a qual os usuários serão redirecionados ao fazer login.
URL do serviço SLO	A URL para a qual os usuários serão redirecionados ao fazer logout. Se deixado em branco, o serviço SLO não será usado.

Parâmetro	Descrição
Atributo de nome de usuário	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.
Formato do nome de entidade do SP	Solicite um formato específico de nome de entidade 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
Certificado IdP	Conteúdo do certificado do provedor de serviço (SP) para configuração do serviço SAML single sign-on (SSO).
Chave privada do SP	Conteúdo da chave privada do provedor de serviço (SP) para configuração do serviço SAML single sign-on (SSO). Fornece autenticação segura e troca de dados com o Provedor de Identidade (IdP).
Certificado do SP	Conteúdo do certificado do provedor de serviço (SP) para configuração do serviço SAML single sign-on (SSO).
Assinar	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*
Criptografar	Marque as caixas de seleção para selecionar as entidades para as quais a criptografia SAML deve ser habilitada: *Nome de entidade* *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 sempre será 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 mostrar opções relacionadas ao provisionamento de usuário JIT.
Atributo de nome do grupo	Especifique o atributo de nome do grupo para provisionamento de usuário JIT.
Atributo de nome do usuário	Especifique o atributo de nome do usuário para provisionamento de usuário JIT.
Atributo de sobrenome do usuário	Especifique o atributo de sobrenome do usuário para provisionamento de usuário JIT.

Parâmetro	Descrição
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 receberá 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 receberá o nível de permissão mais alto entre elas.
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 provisionamento SCIM

Para 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 Zabbix utiliza a biblioteca OneLogin's SAML PHP Toolkit (versão 3.4.1). A estrutura da seção $SSO['SETTINGS'] deve ser semelhante à estrutura usada pela biblioteca. Para a descrição das opções de configuração, 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 *SP name ID* da configuração SAML, defina:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

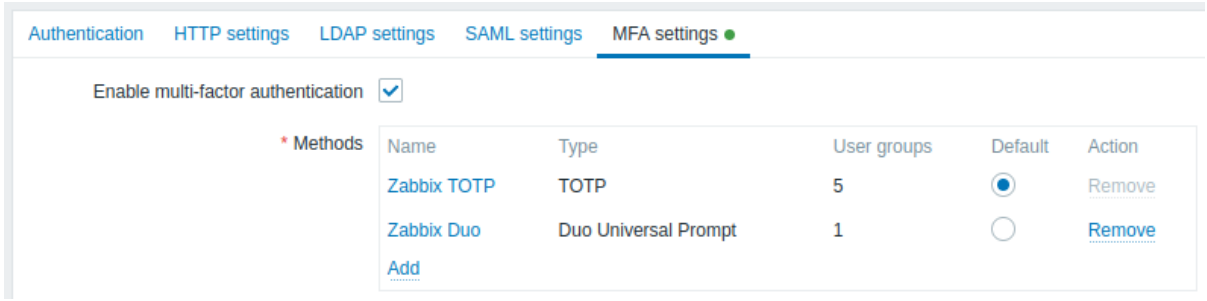
Visão geral

A autenticação multifator (MFA) pode ser usada para fazer login no Zabbix, fornecendo uma camada adicional de segurança além do nome de usuário e senha.

Com 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.

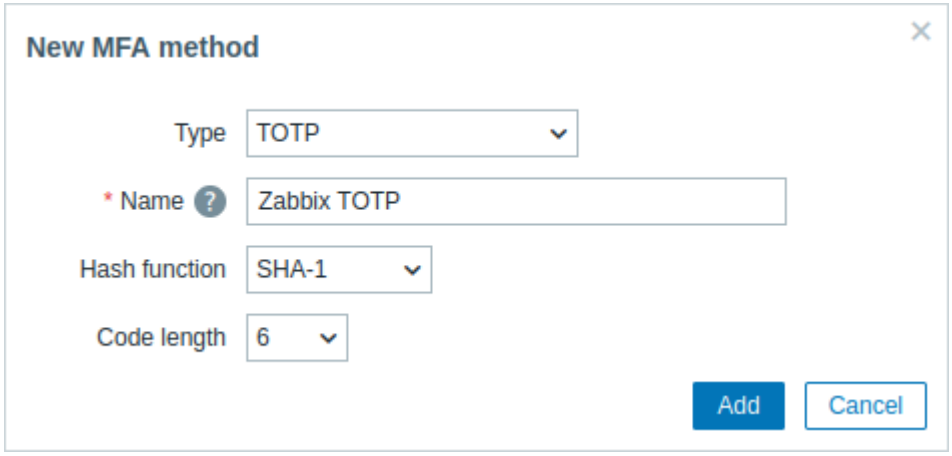
Configuração



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	Insira 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	Insira 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”.
Client ID	Insira o client ID fornecido pelo serviço de autenticação Duo. Este parâmetro está disponível se o tipo de método MFA estiver definido como ”Duo Universal Prompt”.

Parâmetro	Descrição
Client secret	Insira o client secret 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 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: Grupo TOTP
- 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 pela aplicação *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 na 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.

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

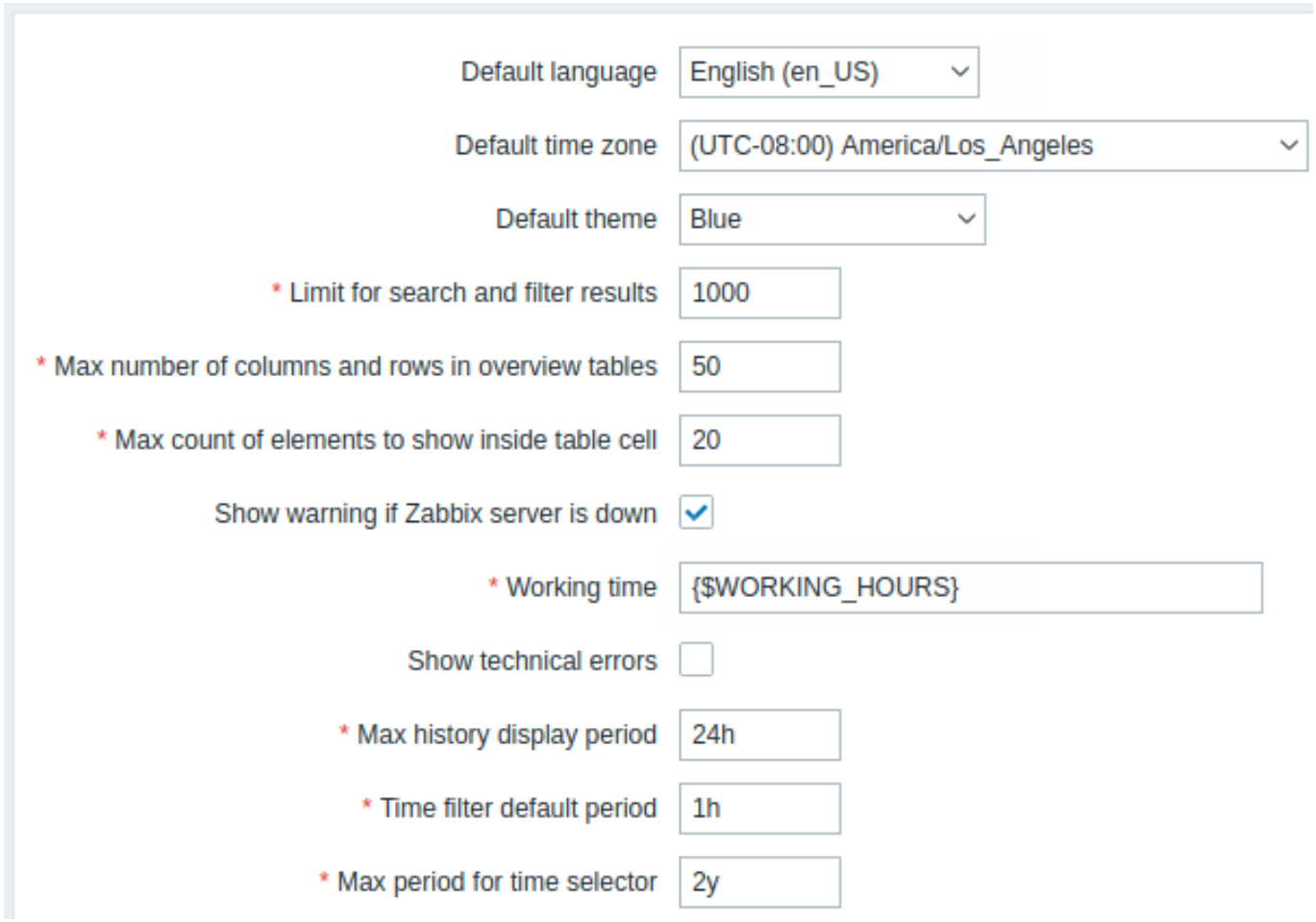
Visão geral

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.



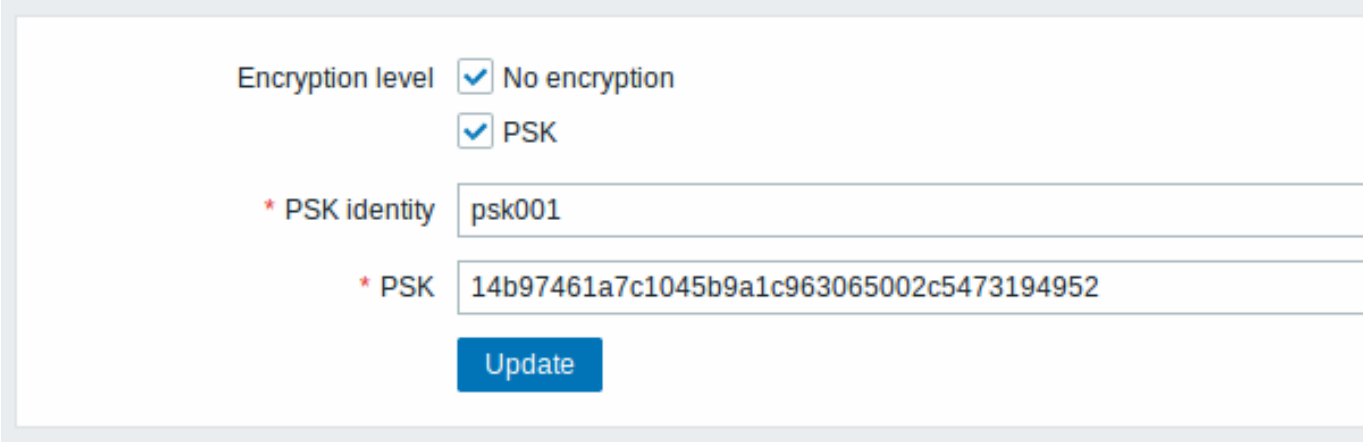
Parâmetros de configuração:

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 for utilizado um filtro 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 em tabelas de visão geral	Número máximo de colunas e linhas a serem exibidas no widget de painel *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. 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 de tabela, não serão mostradas mais do que o configurado aqui.
Mostrar 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 que o usuário role 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 de sistema define o horário de trabalho. Em gráficos, o horário de trabalho é exibido como fundo branco e o horário fora do expediente é exibido como cinza. Consulte a página Especificação de período de tempo para descrição do formato de tempo. Macros de usuário são suportadas.
Mostrar erros técnicos	Se marcado, todos os usuários registrados poderão ver erros técnicos (PHP/SQL). Se desmarcado, a informação estará disponível 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 de cenário web do host em *Monitoramento* > *Hosts* e no widget de painel *Top items*. 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.



Parâmetros marcados com um asterisco são obrigatórios.

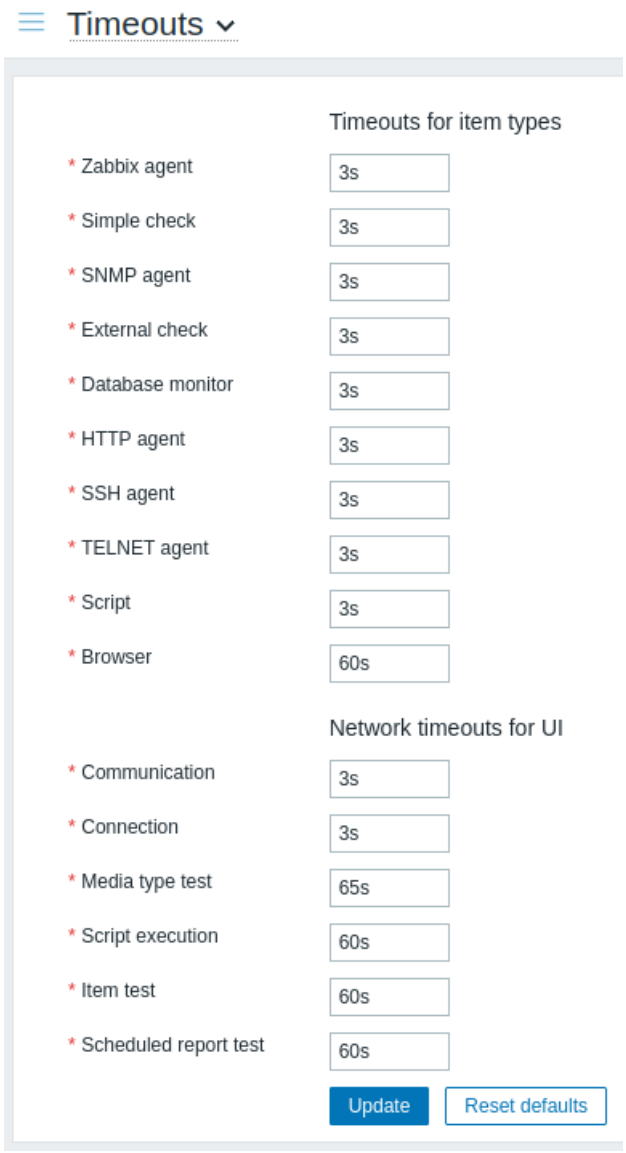
Parâmetros de configuração:

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 sensíveis 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.

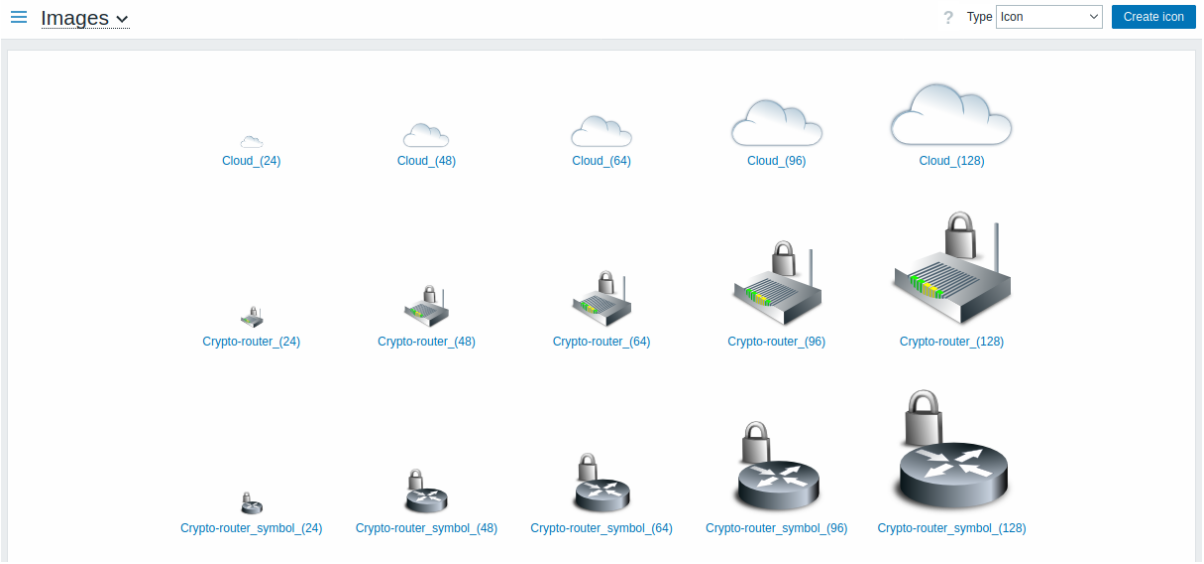


Parâmetro	Descrição
Timeouts por tipo de item	Quantos segundos esperar por uma resposta de um item monitorado (com base em seu tipo). Intervalo permitido: 1 - 600s (padrão: 3s; padrão para o tipo de item Browser: 60s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são suportados. Tipos 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 utilizado e possuir timeouts configurados, as configurações de timeout do proxy irão sobrescrever as globais. Se houver timeouts definidos para itens específicos, eles irão sobrescrever as configurações do proxy e as globais.
Timeouts de rede para a UI	

Parâmetro		Descrição
	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).
	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.

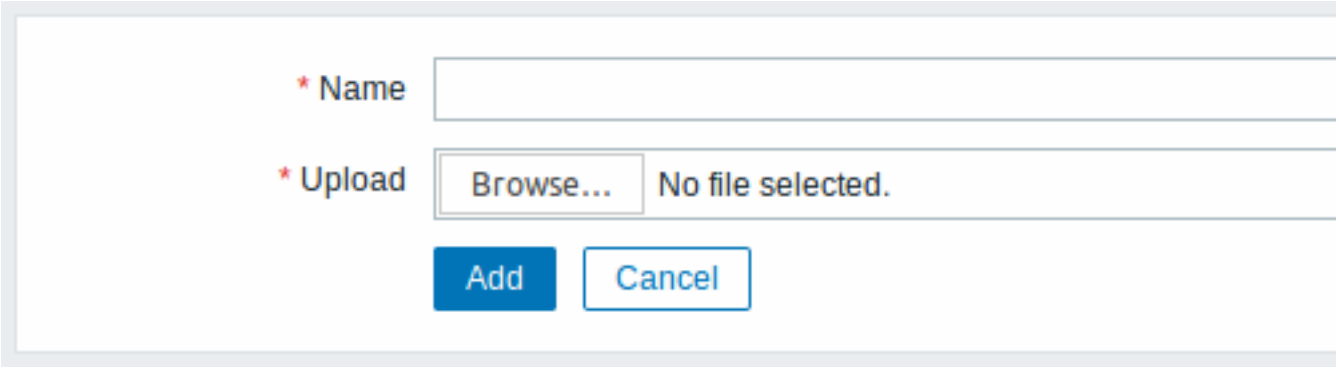


O menu suspenso *Tipo* permite alternar entre ícones e imagens de fundo:

- Ícones são usados para exibir elementos de mapas de rede
- Fundos são usados como imagens de fundo de 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) de um 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 exige 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.

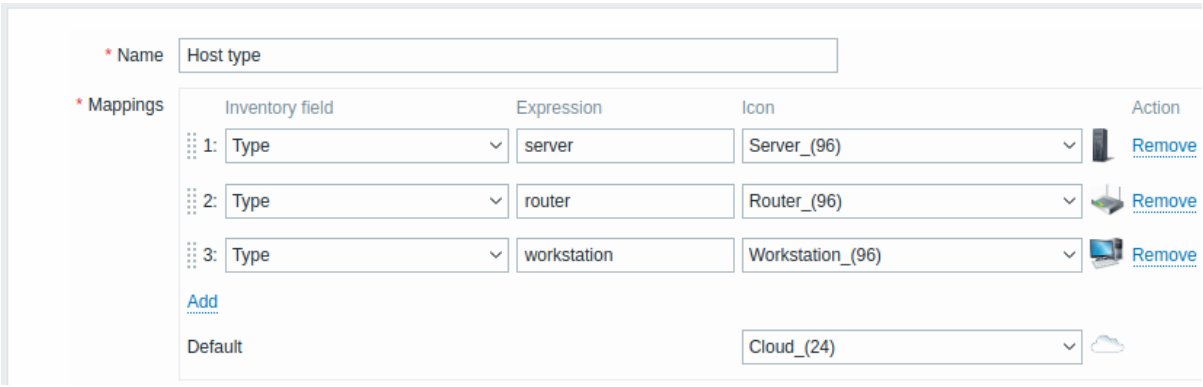
O upload de uma imagem pode falhar se o tamanho da imagem estiver próximo de 1 MB e o parâmetro de configuração do MySQL `max_allowed_packet` estiver no padrão de 1 MB. Nesse caso, aumente o parâmetro max_allowed_packet.

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 locais no frontend. Consulte a seção Expressões regulares para obter detalhes.

Opções de exibição de trigger

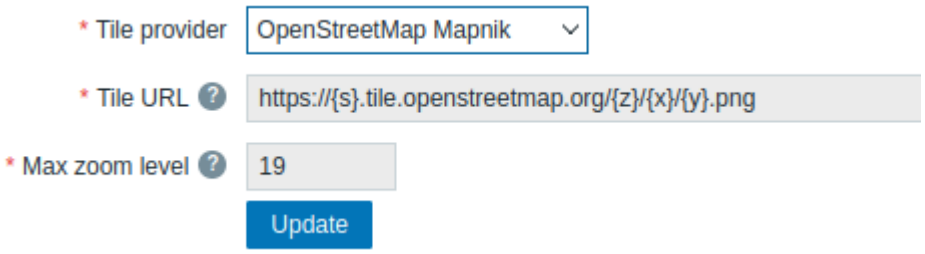
Esta seção permite personalizar como o status do trigger é exibido no frontend e os nomes e cores da severidade do trigger.



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 *piscar* 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 de status, triggers piscam por	Duração do piscar do trigger. Faixa permitida: 0 - 24 horas. Sufixos de tempo, por exemplo, 5m, 2h, 1d, são suportados.
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 definições do provedor para o widget de Geomap 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 *Other* para adicionar outro provedor de tiles ou tiles auto-hospedados (veja Usando um provedor 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 *Tile provider* estiver definido como *Other*. 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 *Tile provider* estiver definido como *Other*.
Max zoom level	Nível máximo de zoom do mapa. Este campo é editável apenas se *Tile provider* estiver definido como *Other*.

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 do frontend personalizados, bem como os módulos do frontend integrados.

Clique em *Procurar diretório* para registrar/remover o registro de quaisquer módulos personalizados. Os módulos registrados aparecerão na lista; os módulos removidos do registro serão removidos da lista.

Clique no status do módulo na lista para habilitar/desabilitar um módulo. Você também pode habilitar/desabilitar vários módulos de uma vez, selecionando-os na lista e clicando nos botões *Habilitar/Desabilitar* 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 *Habilitado* e clique em *Atualizar*.

Você pode filtrar os módulos por nome ou status (habilitado/desabilitado).

Conectores

Esta seção permite configurar conectores para streaming 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*.

Parâmetro	Descrição
Grupo de usuários para mensagem de banco de dados indisponível	Grupo de usuários para envio de mensagem de alarme ou 'Nenhum'. O Zabbix server depende da disponibilidade do banco de dados backend. Ele não pode funcionar sem um banco de dados. Se o banco de dados estiver indisponível, os usuários selecionados podem ser notificados pelo Zabbix. As notificações serão enviadas para o grupo de usuários definido aqui usando as mídias 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 esteja configurada e habilitada, ela não acionará notificações. O Zabbix server não será interrompido; ele aguardará até que o banco de dados volte para continuar o processamento. A notificação consiste no seguinte conteúdo: `[MySQL|PostgreSQL] 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 Zabbix server) tenta estabelecer uma nova conexão com o banco de dados a cada 10 segundos. Se o banco de dados ainda estiver indisponível, o gerenciador de alertas repete o envio de alertas, mas não mais frequentemente do que a cada 15 minutos.
Registrar SNMP traps não correspondentes	Registrar SNMP trap se nenhuma interface SNMP correspondente for encontrada.

Autorização

Parâmetro	Descrição
Tentativas de login *Intervalo de bloqueio de login*	Número de tentativas de login malsucedidas antes que a possibilidade de login seja bloqueada. Período de tempo durante o qual o login será proibido quando o limite de *Tentativas de login* for excedido. Intervalo permitido: 0 - 3600 segundos. Sufixos de tempo, por exemplo, 90s, 5m, 1h, são suportados.

Armazenamento de segredos

Parâmetro	Descrição
Vault provider	Selecione o software de gerenciamento de segredos para armazenar valores de macro de usuário - *HashiCorp Vault* (padrão) ou *CyberArk Vault*.
Resolve secret vault macros by	Resolve macros de cofre de segredos por: **Zabbix server** - os segredos são recuperados pelo Zabbix server e encaminhados para os proxies quando necessário (padrão); **Zabbix server and proxy** - os segredos são recuperados tanto pelo Zabbix server quanto pelos proxies, permitindo que eles resolvam macros de forma independente.

Veja também: Armazenamento de segredos.

Segurança

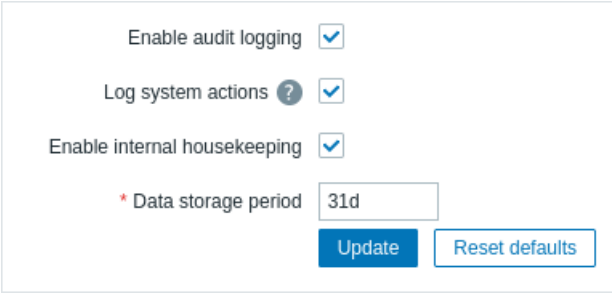
Parâmetro	Descrição
Validar esquemas de URI	Desmarque esta caixa para desabilitar a validação de esquemas de URI (habilitado por padrão). Se marcada, você pode especificar uma lista separada por vírgulas de esquemas de URI permitidos (padrão: http,https,ftp,file,mailto,tel,ssh). Aplica-se a todos os campos no frontend onde URIs são usados (por exemplo, URLs de elementos de mapas).

Parâmetro	Descrição
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 sandbox 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 sandbox para o 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:

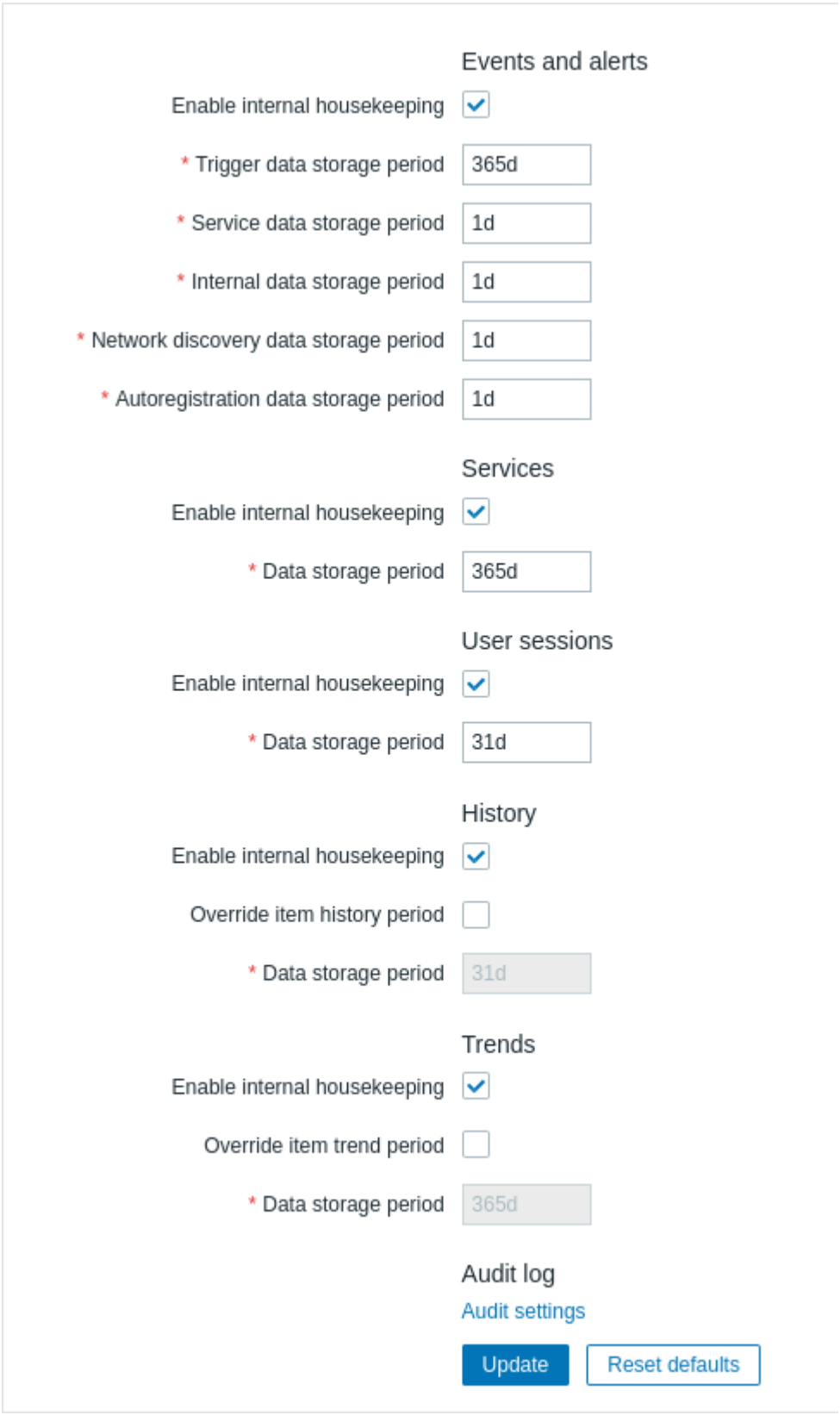
Parâmetro	Descrição
Habilitar log de auditoria	Habilita (padrão) ou desabilita o log de auditoria.
Registrar ações do sistema	Habilita (padrão) ou desabilita o log de auditoria das atividades de descoberta de baixo nível, descoberta de rede e autorregistro realizadas pelo server (usuário do sistema).
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 compressão do log de auditoria estão disponíveis na seção *Administração > Limpeza*, dentro do bloco *Compressão de histórico, tendências e log de auditoria*, que se torna visível quando o TimescaleDB é utilizado.

3 Limpeza

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
- Sessões de usuário
- Histórico
- Tendências

As configurações de housekeeping de auditoria estão disponíveis em uma seção de menu separada.

Configuração

A tabela abaixo descreve os parâmetros de configuração da limpeza automática.

Parâmetro	Descrição
Habilitar limpeza automática interna	Habilita (padrão) ou desabilita a limpeza automática interna para este grupo de entidades.
Período de armazenamento de dados	Tempo pelo qual os registros devem ser mantidos antes de serem removidos pelo housekeeper. Obrigatório se a limpeza automática estiver habilitada. Intervalo: 1 dia (1 hora para histórico) - 25 anos; ou ”0”. Sufixos de tempo são suportados, por exemplo, 1d (um dia), 1w (uma semana). Para *Eventos e alertas*, o período de armazenamento de dados é definido separadamente para triggers, serviços, dados internos, descoberta de rede e autorregistro. Para items, o período de armazenamento de dados também determina o período pelo qual os dados estarão visíveis na seção *Monitoramento* > *Últimos dados*, mesmo que a limpeza automática interna esteja desabilitada.
Sobrescrever período de histórico do item	Se marcado, o período de armazenamento de histórico/tendência especificado na configuração do item será sobrescrito pela configuração global. O período de armazenamento não será sobrescrito para items que tenham a opção de configuração *Não armazenar* habilitada. É possível sobrescrever o período de armazenamento de histórico/tendência mesmo que a limpeza automática interna esteja desabilitada. 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.
Sobrescrever período de tendência do item	

O botão *Restaurar padrões* permite reverter quaisquer alterações feitas.

Note:

- Excluir um item/trigger também excluirá os problemas gerados por esse item/trigger.
- Eventos de problema ou recuperação não serão excluídos até que o registro de problema relacionado seja excluído. O housekeeper exclui primeiro os problemas e depois os eventos, para evitar dados obsoletos.

Configurações para TimescaleDB

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 tendências do item* devem ser habilitadas, assim como a opção *Habilitar limpeza interna* 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 o descarte de partições desatualizadas estiver habilitado, o Zabbix server e o frontend não irão mais rastrear itens excluídos, e o histórico desses itens será limpo quando uma partição desatualizada for excluída.

As configurações de compressão para histórico, tendências e log de auditoria estão disponíveis no bloco *Compressão de histórico, tendências e log de auditoria*, que aparece quando o TimescaleDB é utilizado.

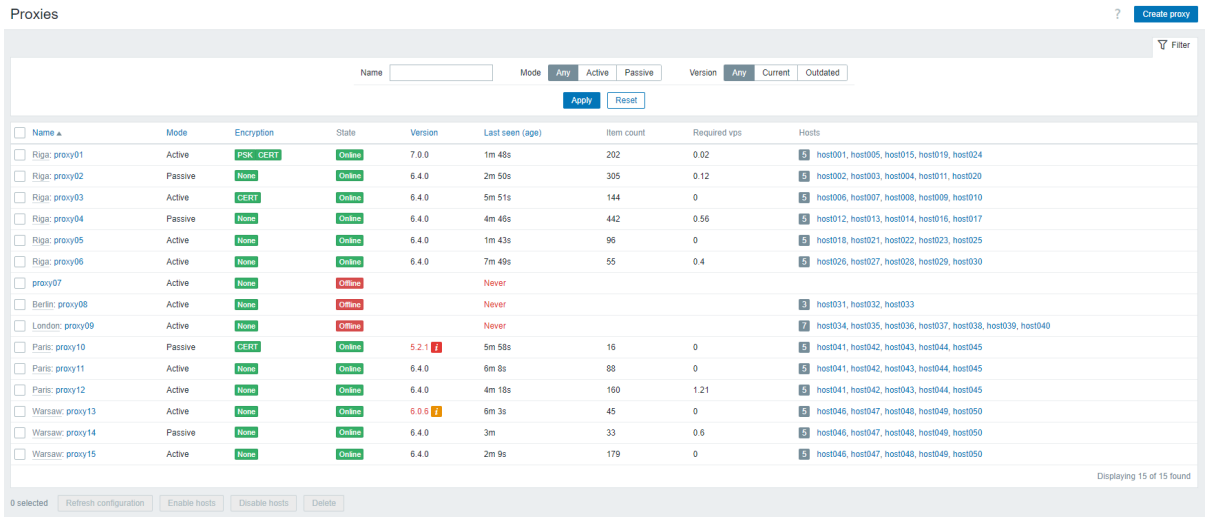
4 Proxies

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*.
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 ver detalhes.
Última vez visto (idade)	O horário em que o proxy foi visto pelo server pela última vez.
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 se comunicou 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 se comunicou 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	Quantidade 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 *Quantidade 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:

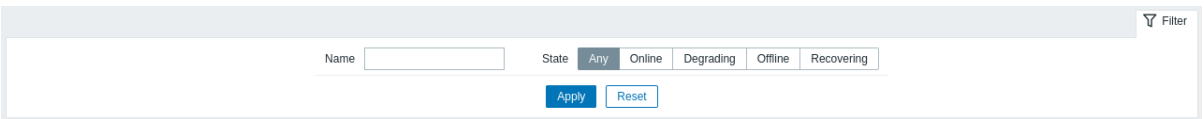
- *Excluir* - exclui os grupos de proxy.

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de proxy e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os grupos de proxy que lhe interessam. Para melhor desempenho de 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 atualização.

Idealmente, ao abrir esta seção, ela deve estar toda "verde", significando 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.

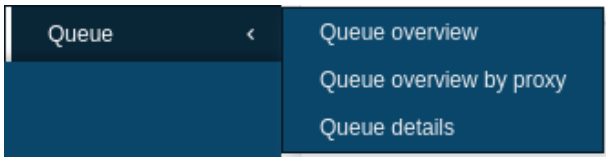
Note:
O proxy Zabbix não reconhece períodos de manutenção; consulte Cálculo de filas durante a manutenção para detalhes.

Note:
A fila está disponível apenas se o server Zabbix estiver em execução. Os items não são contados na fila se a interface do item ficar indisponível devido a problemas de conexão ou se o agent não estiver funcionando corretamente.

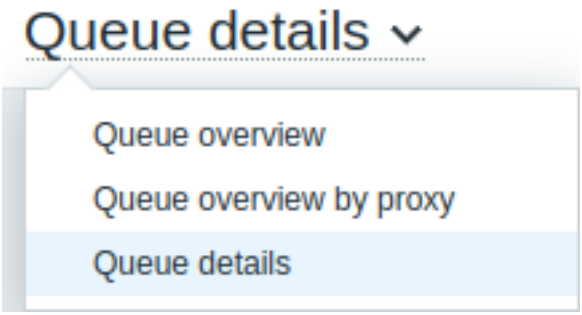
A seção *Administração* → *Fila* contém as seguintes páginas:

- 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.



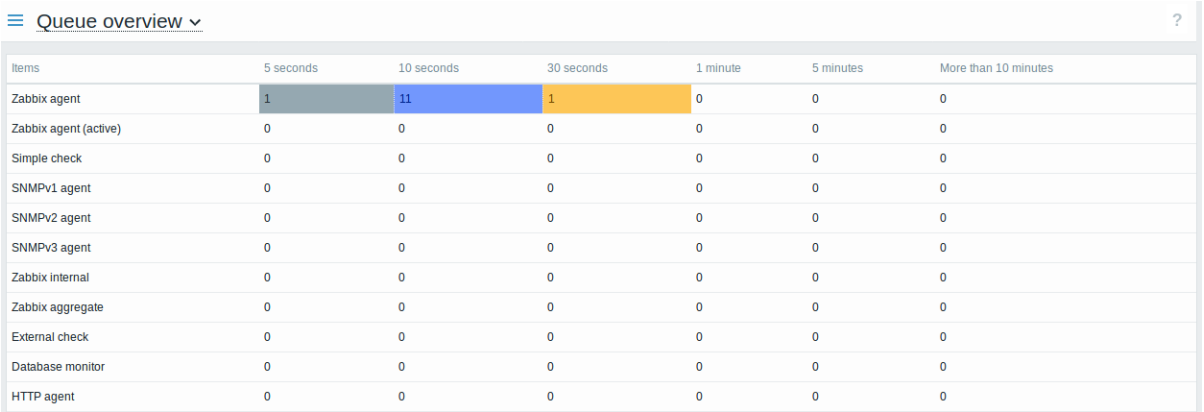
Menu de terceiro nível.



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 item.

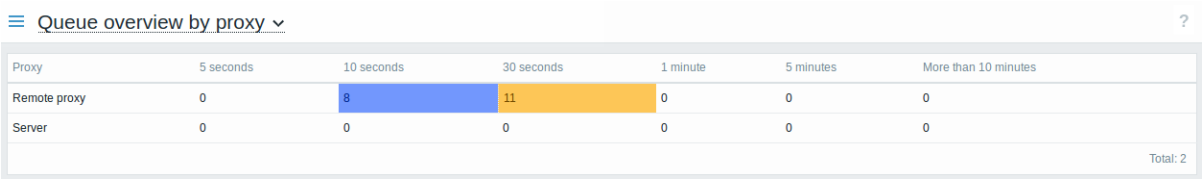

Queue overview

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 itens em espera - esperando por 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

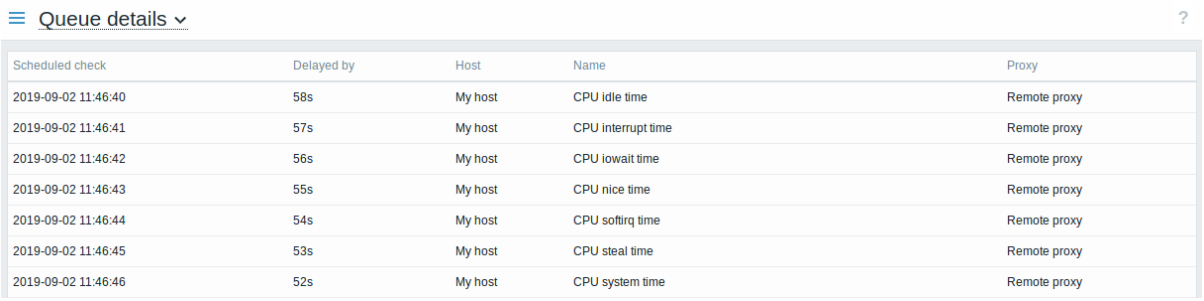
Nesta tela é fácil localizar se o problema está relacionado a um dos proxies ou ao server.



Cada linha contém um proxy, com o server por último na lista. Cada coluna mostra o número de items em espera - esperando por 5-10 segundos/10-30 segundos/30-60 segundos/1-5 minutos/5-10 minutos ou mais de 10 minutos, respectivamente.

Lista de items em espera

Nesta tela, cada item em espera 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 em espera é exibido.
Proxy	O nome do proxy é exibido, se o host for monitorado por proxy.

Possíveis mensagens de erro

Você pode encontrar 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.

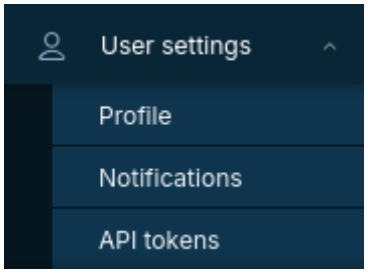
3 Configurações do usuário

Visão geral

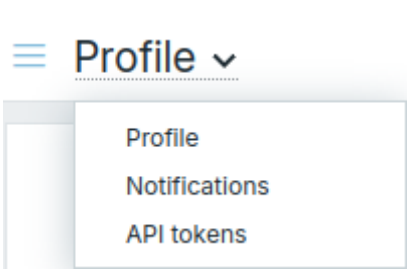
Dependendo das permissões da função do usuário, a seção *Configurações do usuário* pode conter as seguintes páginas:

- *Perfil* ou *Perfil do usuário* - para personalizar certos recursos do frontend do Zabbix;
- *Notificações* - para personalizar as notificações do usuário atual;
- *Tokens de API* - para gerenciar os 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.

profile.png

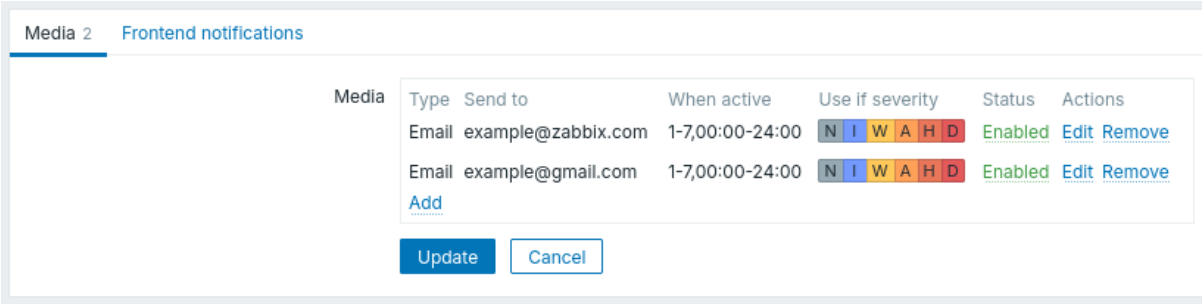
Parâmetro	Descrição
Nome	O nome e sobrenome do usuário são exibidos. Na ausência de nome e sobrenome, o nome de usuário é exibido.
Senha	Clique no botão *Alterar senha* para abrir três campos: *Senha antiga*, *Nova senha*, *Nova senha (mais uma vez)*. Ao alterar a senha com sucesso, o usuário será desconectado 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 no 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 seu perfil: **Padrão do sistema** - usa as configurações padrão do sistema; **Azul** - tema azul padrão; **Escuro** - tema escuro alternativo; **Claro de alto contraste** - tema claro com alto contraste; **Escuro de alto contraste** - tema escuro com alto contraste.
Login automático	Marque esta caixa para que o Zabbix 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.

Parâmetro	Descrição
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 será 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.
Atualização	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*.

Notificações

A seção *Notificações* fornece opções para personalizar as notificações do usuário atual.

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:
As permissões do usuário para alterar os detalhes de mídia para si mesmo podem ser concedidas/revogadas com base em sua função de usuário (veja a opção *Criar e editar própria mídia*). As permissões do superadministrador para alterar os detalhes de mídia de outros também podem ser concedidas/revogadas com base em sua função de usuário (veja a opção *Criar e editar mídia do usuário*)

Observe que para usuários provisionados:

- a mídia do usuário provisionada não pode ser excluída;
- a mídia do usuário provisionada pode ser desabilitada/habilitada;
- os campos de mídia do usuário provisionados, 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 do 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.

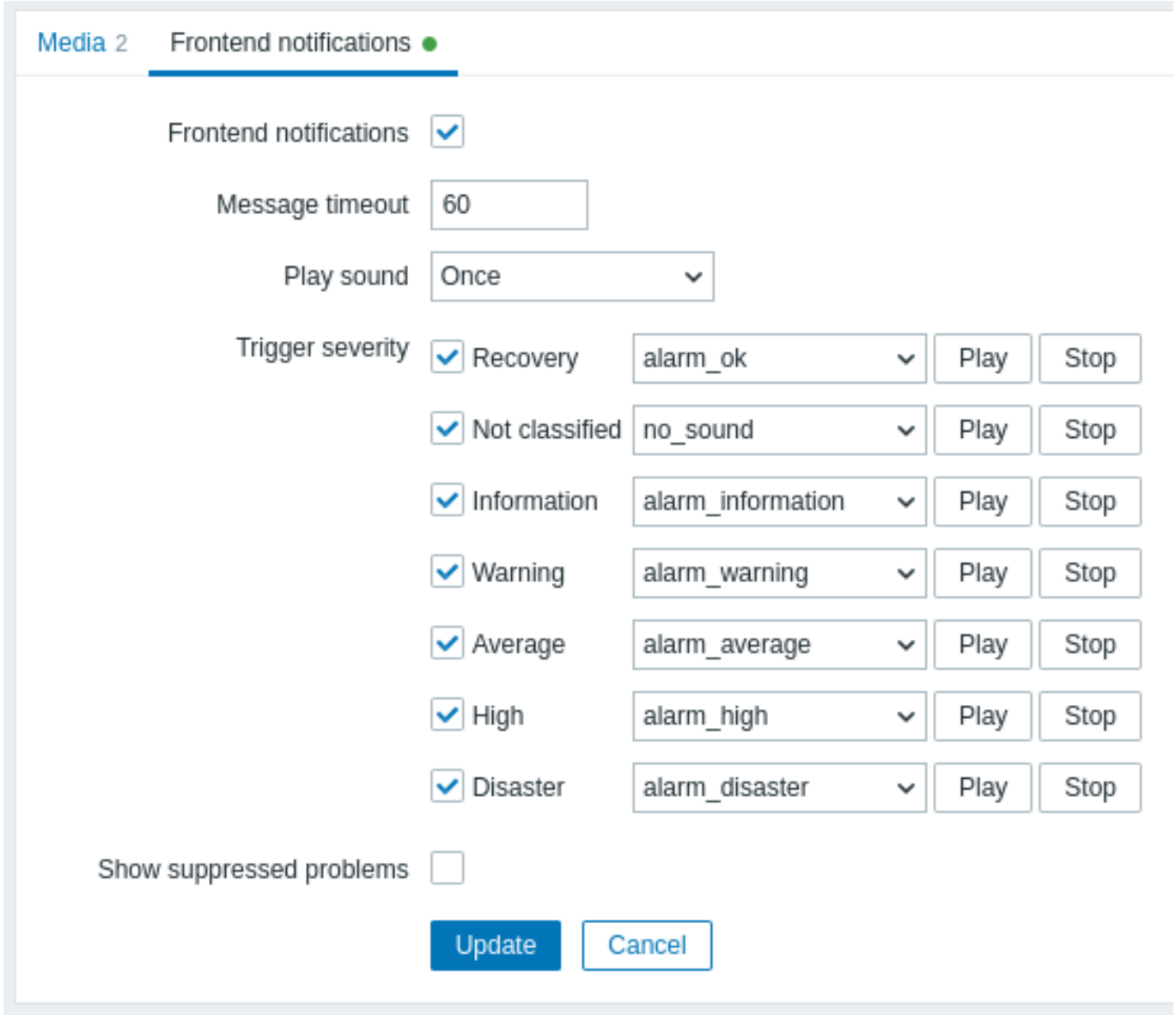
Sem notificações globais, ao trabalhar fora das seções *Problemas* ou *Dashboard*, você não receberia nenhuma informação sobre problemas atuais. As notificações globais garantem que essas informações sejam exibidas, independentemente de sua localização atual dentro do frontend do Zabbix.

As notificações globais incluem tanto exibir uma mensagem quanto tocar um som.

Attention:
A reprodução automática de sons pode estar desabilitada (por padrão) em versões recentes dos navegadores. Nesses casos, você precisa ativar essa configuração manualmente.

Configuração

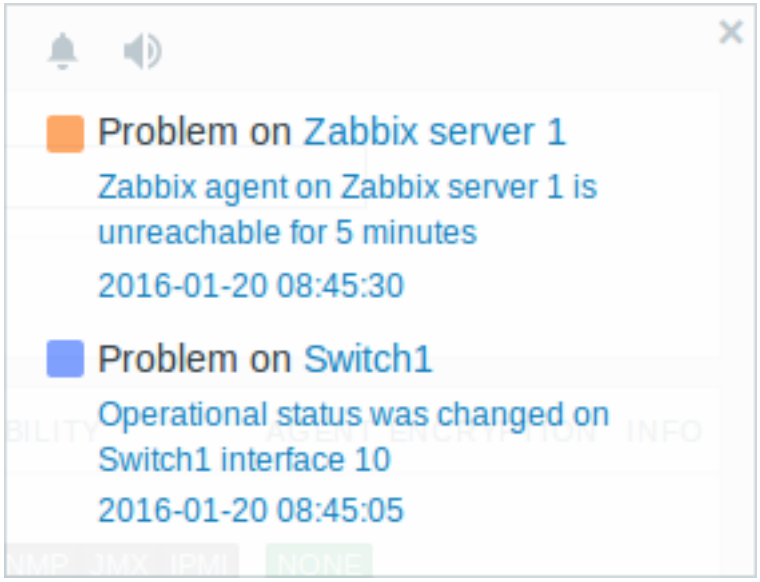
As notificações globais podem ser ativadas por usuário na aba **Notificações do frontend** da seção Notificações.



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 do trigger 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, as 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 **Ativar/Desativar som** alterna entre reproduzir e não reproduzir os sons de alarme.

2 Som nos navegadores

Visão geral

O som é utilizado em notificações globais.

Para que os sons sejam reproduzidos no frontend do Zabbix, as *Notificações do frontend* devem estar habilitadas na aba *Notificações do frontend* do perfil do usuário, com todas as gravidades de trigger marcadas. Além disso, os sons devem estar habilitados 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 nas versões recentes dos navegadores Firefox e Opera no Linux, e nos navegadores Chrome, Firefox, Microsoft Edge e Opera no Windows.

Attention:
A reprodução automática de sons pode estar desabilitada (por padrão) nas versões recentes dos navegadores. Nesses casos, você precisa habilitar essa configuração manualmente.

4 Pesquisa global

É possível pesquisar no frontend do Zabbix por hosts, grupos de hosts, templates e grupos de templates.

A caixa de entrada de pesquisa está localizada abaixo do logotipo do Zabbix no menu. A pesquisa pode ser iniciada pressionando *Enter* ou clicando no ícone de pesquisa .



Se houver um host que contenha a string inserida em qualquer parte do nome, um menu suspenso aparecerá, listando todos esses hosts (com a parte correspondente destacada em laranja). O menu suspenso também listará um host se o nome visível desse host corresponder ao nome técnico inserido como string de pesquisa; o host correspondente será listado, mas sem nenhum destaque.

Atributos pesquisáveis

Os hosts podem ser pesquisados pelas seguintes propriedades:

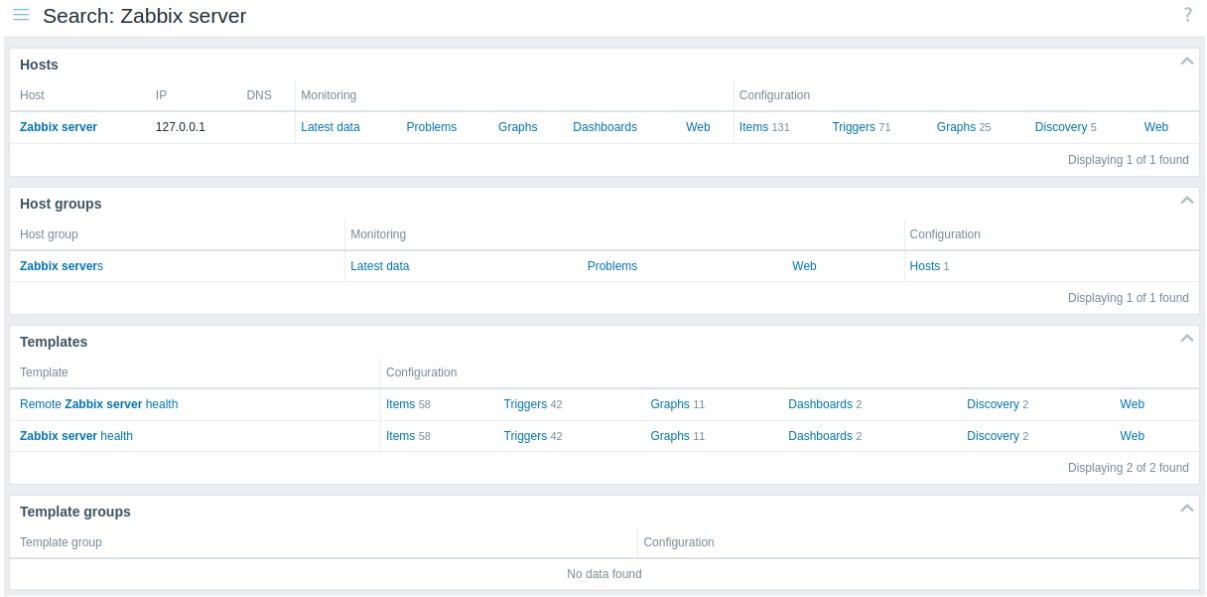
- Nome do host
- Nome visível
- Endereço IP
- Nome DNS

As templates podem ser pesquisadas por nome ou nome visível. Se você pesquisar por um nome que seja diferente do nome visível (de uma template/host), nos resultados da pesquisa ele será exibido abaixo do nome visível entre parênteses.

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 do frontend

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` no 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 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 as 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, a localização para:

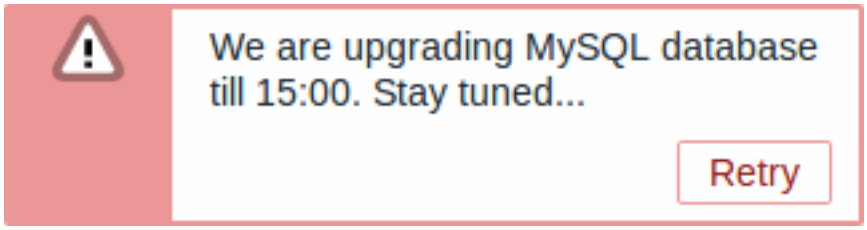
- SUSE e RedHat é `/etc/zabbix/web/maintenance.inc.php`.
- Sistemas baseados em Debian é `/usr/share/zabbix/conf/`.

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 definido, 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 de 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 parâmetro=valor após a URL, separados da URL por um ponto de interrogação (?) e uns dos outros por e comercial (&).

Monitoramento > Problemas

Os seguintes parâmetros são suportados:

Parâmetro	Descrição	Exemplo
show	Opção de filtro *Mostrar*. Valores possíveis: 1 - problemas recentes; 2 - todos; 3 - em estado de problema.	show=1
name	Opção de filtro *Problema*: string livre.	name=Zabbix agent

Parâmetro	Descrição	Exemplo
severities	Opção de filtro *Gravidade*: array de gravidades selecionadas no formato `severities[*]=*` (substitua * pelo nível de gravidade). Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 5 - desastre.	severities[3]=3
inventory	Opção de filtro *Inventário do host*: array de campos de inventário `[field],[value]`	inventory[0][field]=type&invento
evaltype	Opção de filtro *Tags*, estratégia de filtragem de tags. Valores possíveis: 0 - e/ou; 2 - ou.	evaltype=0
tags	Opção de filtro *Tags*: array de tags definidas `[tag], [operator], [value]` Valores possíveis para o operador: 0 - contém; 1 - igual; 2 - não contém; 3 - não é igual; 4 - existe; 5 - não existe.	tags[0][tag]=target&tags[0][ope
show_tags	Opção de filtro *Mostrar tags*. Valores possíveis: 0 - nenhuma; 1 - uma; 2 - duas; 3 - três.	show_tags=3
tag_name_format	Opção de filtro *Nome da tag*. Valores possíveis: 0 - nome completo; 1 - nome abreviado; 2 - nenhum.	tag_name_format=1
tag_priority	Opção de filtro *Prioridade de exibição da tag*: string separada por vírgulas da prioridade de exibição da tag	tag_priority=customer, target
show_suppressed	Opção de filtro *Mostrar problemas suprimidos*. Valores possíveis: 1 - mostrar; 2 - não mostrar.	show_suppressed=1
acknowledgement_status	Opção de filtro *Status de reconhecimento*. Valores possíveis: 0 - todos; 1 - não reconhecidos; 2 - reconhecidos.	acknowledgement_status=0
acknowledged_by_me	Opção de filtro *Por mim*; suportado apenas com `acknowledgement_status=2`. Valores possíveis: 0 - desabilitado; 1 - habilitado.	acknowledged_by_me=1

Parâmetro	Descrição	Exemplo
compact_view	Opção de filtro *Visualização compacta*. Valores possíveis: 0 - desabilitado; 1 - habilitado.	compact_view=1
highlight_row	Opção de filtro *Destacar linha inteira*; suportado apenas com `compact_view=1`. Valores possíveis: 0 - desabilitado; 1 - habilitado.	highlight_row=1
filter_name	Opção de propriedades do filtro *Nome*: string livre	filter_name=Databases
filter_show_counter	Opção de propriedades do filtro *Mostrar número de registros*. Valores possíveis: 0 - desabilitado; 1 - habilitado.	filter_show_counter=1
filter_custom_time	Opção de propriedades do filtro *Definir período de tempo personalizado*. Valores possíveis: 0 - desabilitado; 1 - habilitado.	filter_custom_time=1
sort	Coluna para ordenar. Valores possíveis: clock - ordenar pela coluna *Hora*; host - ordenar pela coluna *Host*; severity - ordenar pela coluna *Gravidade*; name - ordenar pela coluna *Problema*.	sort=clock
sortorder	Ordem de classificação dos resultados. Valores possíveis: DESC - decrescente; ASC - crescente.	sortorder=DESC
age_state	Opção de filtro *Idade menor que*; suportado apenas com `show=3`. Valores possíveis: 0 - desabilitar o parâmetro `age`; 1 - habilitar o parâmetro `age`.	age_state=1
age	Opção de filtro *Idade menor que*: inteiro, número de dias; suportado apenas com `age_state=1` e `show=3`.	age=7
groupids	Opção de filtro *Grupos de hosts*: array de IDs de grupos de hosts.	groupids[]=4
hostids	Opção de filtro *Hosts*: array de IDs de hosts	hostids[]=10084
triggerids	Opção de filtro *Triggers*: array de IDs de triggers	triggerids[]=22382
show_timeline	Opção de filtro *Mostrar linha do tempo*; não suportado com `compact_view=1`. Valores possíveis: 0 - não mostrar; 1 - mostrar.	show_timeline=1
details	Opção de filtro *Mostrar detalhes*. Valores possíveis: 0 - não mostrar; 1 - mostrar.	details=1
from	Início do intervalo de datas, pode ser relativo (por exemplo, now-1m); suportado apenas com `filter_custom_time=1`	from=now-2h
to	Fim do intervalo de datas, pode ser relativo (por exemplo, now-1m); suportado apenas com `filter_custom_time=1`	to=now

Veja também: opções de filtro da página de Problemas.

Modo quiosque

O modo quiosque nas páginas suportadas do frontend pode ser ativado usando parâmetros de URL. Por exemplo, em dashboards:

- `/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 dashboard:

- `/zabbix.php?action=dashboard.view&slideshow=1` - ativa a apresentação de slides

7 Definições

Visão geral

Enquanto muitas coisas no frontend podem 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 nome do cookie de sessão do frontend do Zabbix.

Padrão: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Tempo limite 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 seguir o procedimento passo a passo fornecido aqui para criar o seu próprio. Sinta-se à vontade para compartilhar o resultado do seu trabalho com a comunidade Zabbix se você criou algo interessante.

Passo 1

Para definir seu próprio tema, você precisará criar um arquivo CSS e salvá-lo na pasta `assets/styles/` (por exemplo, *custom-theme.css*). Você pode copiar os arquivos de um tema diferente e criar seu tema com base nele ou começar do zero.

Passo 2

Adicione seu tema à lista de temas retornados pelo método APP::getThemes(). Você pode fazer isso sobrescrevendo o método ZBase::getThemes() na classe APP. Isso pode ser feito adicionando o seguinte código antes da chave de fechamento em *include/classes/core/APP.php*:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:
Observe que o nome especificado dentro do primeiro par de aspas deve corresponder ao nome do arquivo do tema sem extensão.

Para adicionar vários temas, basta listá-los abaixo do primeiro tema, por exemplo:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

Observe que todos os temas, exceto o último, devem ter uma vírgula ao final.

Note:
Para alterar as cores dos gráficos, a entrada deve ser adicionada na tabela *graph_theme* do banco de dados.

Passo 3

Ative o novo tema.

No frontend do Zabbix, você pode definir este tema como o padrão ou alterar o tema no perfil do usuário.

Aproveite o novo visual!

9 Modo de depuração

Visão geral

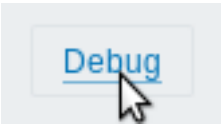
O modo de depuração pode ser usado para diagnosticar problemas de desempenho com as páginas do frontend.

Configuração

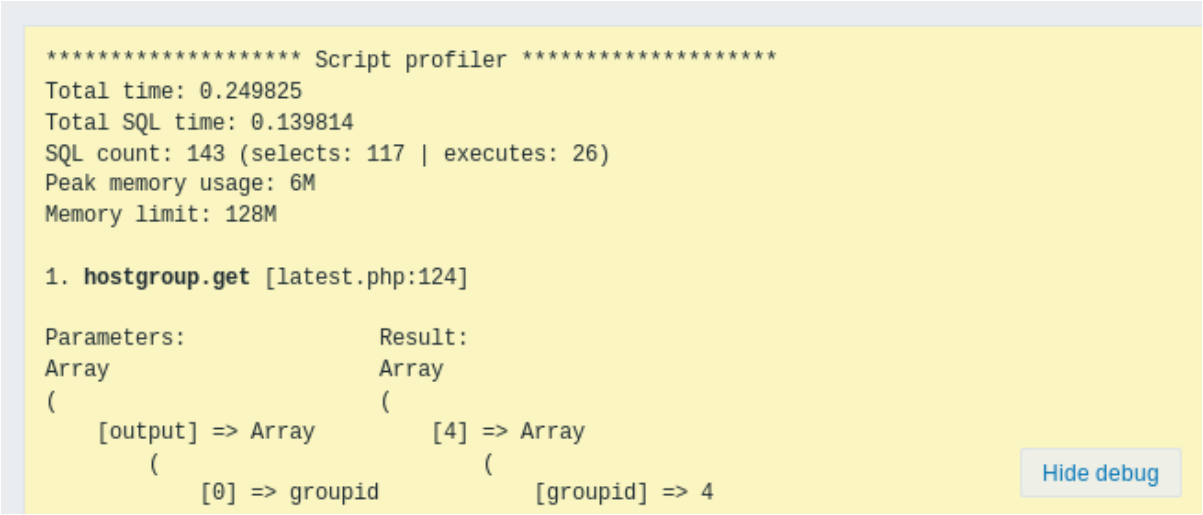
O modo de depuração pode ser ativado para usuários individuais que pertencem a um grupo de usuários:

- 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_SESSION_NAME	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 *Action*) e é criado quando um usuário navega de uma aba primária para outra aba (como a aba *Tags* ou *Dependencies*). 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)	-	-
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 parâmetro 'HttpOnly' nos cookies do Zabbix por uma diretiva do servidor web não é suportado.

Notas de rodapé

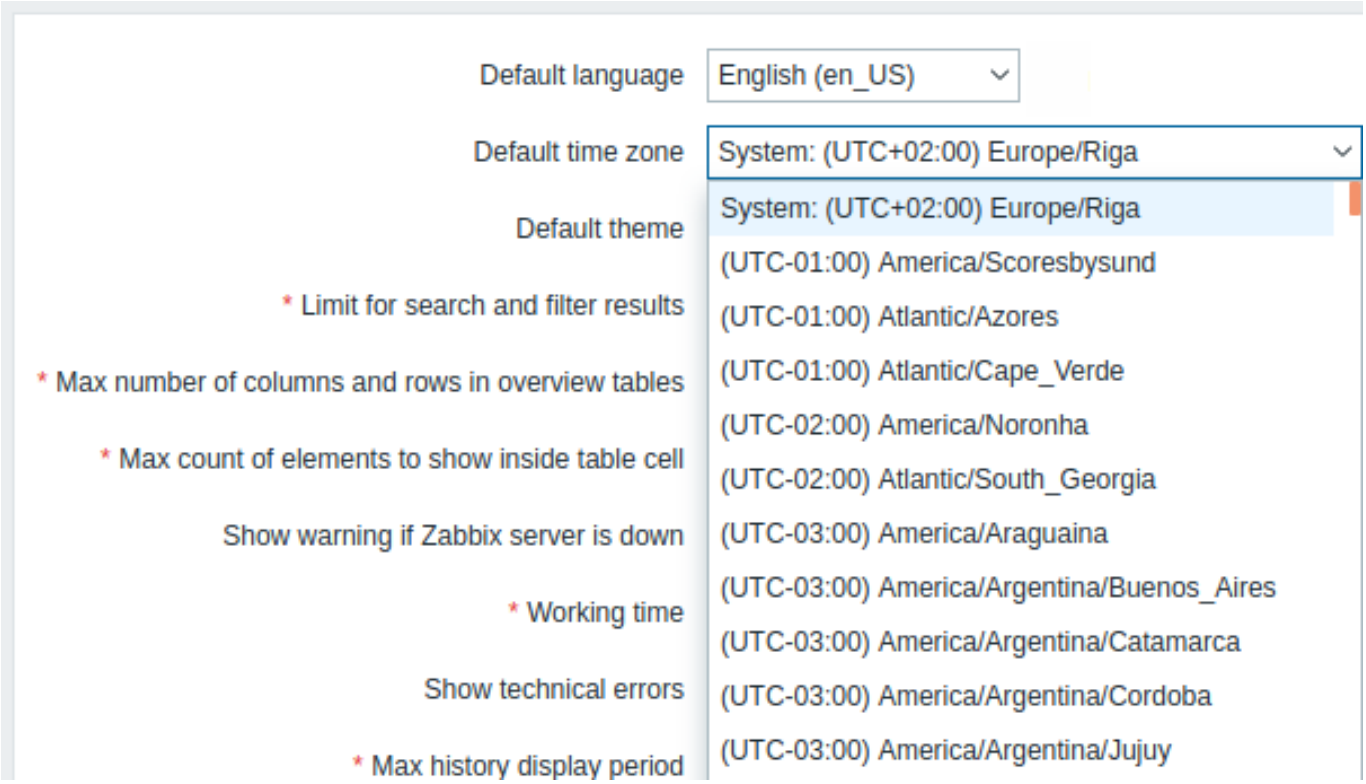
[1] Quando `HttpOnly` é 'true', o cookie será acessível apenas pelo protocolo HTTP. Isso significa que o cookie não estará acessível por linguagens de script, como JavaScript. Essa configuração pode ajudar a reduzir o roubo de identidade por meio de ataques XSS (embora não seja suportada por todos os navegadores).

[2] `Secure` indica que o cookie só deve ser transmitido por uma conexão HTTPS segura a partir do cliente. Quando definido como 'true', o cookie só será definido se existir uma conexão segura.

11 Fusos horários

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 usará o fuso horário global/usuário especificado apenas 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 a senha

Visão geral Esta seção descreve as etapas 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 do usuário.

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.

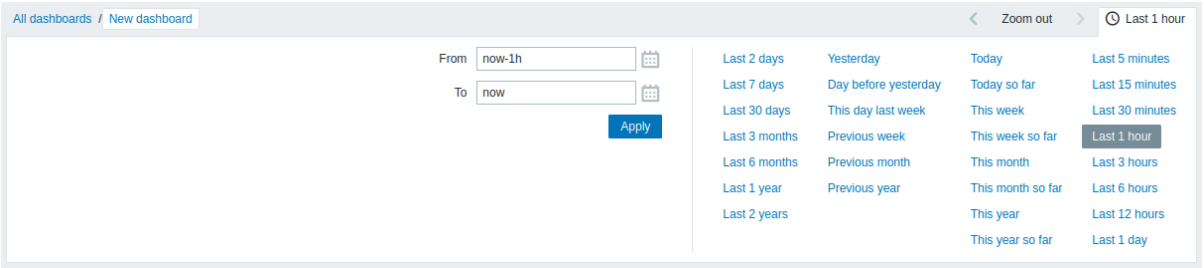
13 Seletor de período de tempo e host

Visão geral

Esta página descreve os seletores de período de tempo e host disponíveis nos dashboards. Esses seletores permitem ajustar dinamicamente os dados exibidos em todos os widgets de dashboard compatíveis sem editar as configurações individuais dos widgets.

Seletor de período de tempo

O seletor de período de tempo permite selecionar períodos frequentemente necessários com um clique do mouse. Ele pode ser expandido ou recolhido clicando no botão do seletor de período de tempo no canto superior direito do dashboard.



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 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, 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. Também é possível reduzir o zoom 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:

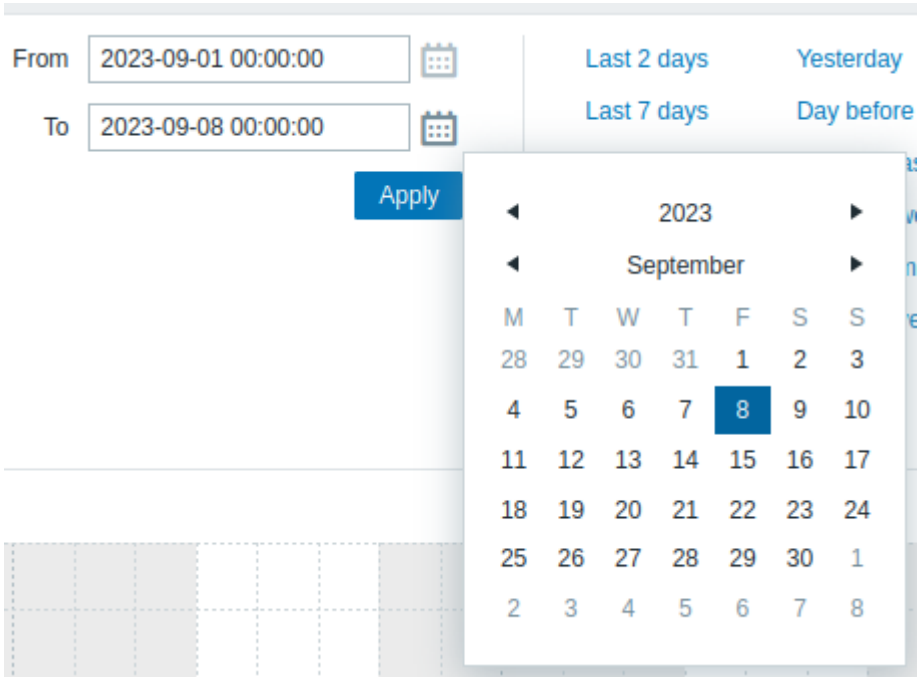
- `now`
- `s` (segundos)
- `m` (minutos)

- `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 data. Para abri-lo, clique no ícone de calendário ao lado dos campos *De/Até*.



Note:
Dentro do seletor de data, 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".

Seletor de host

O seletor de host permite selecionar um host para exibir seus dados em widgets de dashboard compatíveis.

Ele fica disponível após editar e salvar um dashboard, se pelo menos um widget estiver configurado com o parâmetro *Substituir host* definido para usar o dashboard como fonte de dados.



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, implementá-las é altamente recomendado 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, recomenda-se usar os caminhos padrão fornecidos pelo instalador. O uso de caminhos personalizados sem as permissões adequadas pode comprometer a segurança da instalação.

Avisos de segurança do Zabbix e banco de dados CVE

Consulte *Avisos de segurança do Zabbix e banco de dados CVE*.

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 através da API do Zabbix. Nesse caso, limitar as permissões dos scripts e excluir informações sensíveis dos scripts (por exemplo, credenciais de acesso) pode ajudar a evitar a exposição de informações sensíveis 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 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 é desativada por padrão (pode ser ativada definindo EnableRemoteCommands=1 na configuração do proxy);
- a execução de scripts globais nos agentes Zabbix é desativada por padrão (pode ser ativada adicionando um parâmetro AllowKey=system.run[<comando>,*] 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.

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 frontend e API do Zabbix:

```
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 do 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 o 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
```

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.

Habilitando o Zabbix no diretório raiz da URL

Em sistemas baseados em RHEL, adicione um host virtual à configuração do Apache (`/etc/httpd/conf/httpd.conf`) e defina um redirecionamento permanente do diretório 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
```

Ativando HTTP Strict Transport Security (HSTS) no servidor web

Para proteger o frontend do Zabbix contra ataques de downgrade de protocolo, recomendamos ativar a política HSTS no servidor web.

Para ativar a política HSTS 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 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 via HTTPS, evitando 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 Política de Segurança de Conteúdo (CSP) no servidor web

Para proteger o frontend do Zabbix contra Cross Site Scripting (XSS), injeção de dados e ataques semelhantes, recomendamos habilitar a Política de Segurança de Conteúdo no servidor web. Para isso, configure o servidor web para retornar o cabeçalho HTTP.

Attention:
A configuração de 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 de cabeçalho CSP pode ser necessária se você estiver, por exemplo, configurando o widget *URL* para exibir conteúdo dos subdomínios do site ou 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, certifique-se de adicionar "duo.com" à diretiva CSP no arquivo de configuração do seu host virtual.

Para habilitar o CSP para 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 Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Reinicie o serviço Apache para aplicar as alterações:

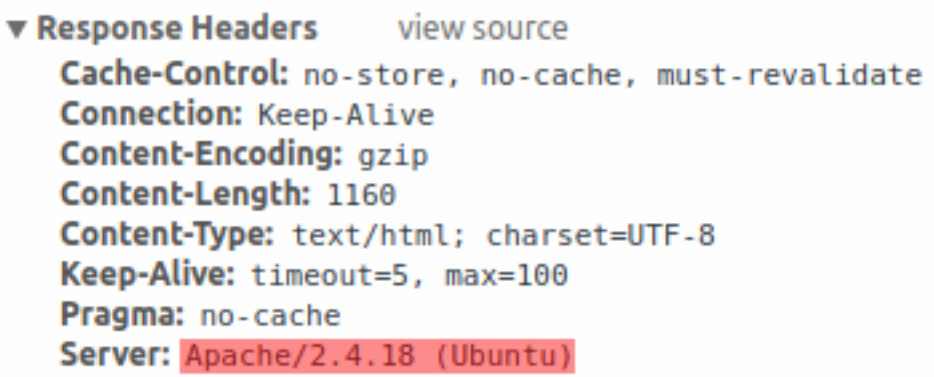
```
#### Em sistemas baseados em RHEL:
systemctl restart httpd.service

#### No Debian/Ubuntu
systemctl restart apache2.service
```

Desabilitando a exposição de informações do servidor web

Para melhorar a segurança, recomenda-se desabilitar todas as assinaturas do servidor web.

Por padrão, o servidor web expõe 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.

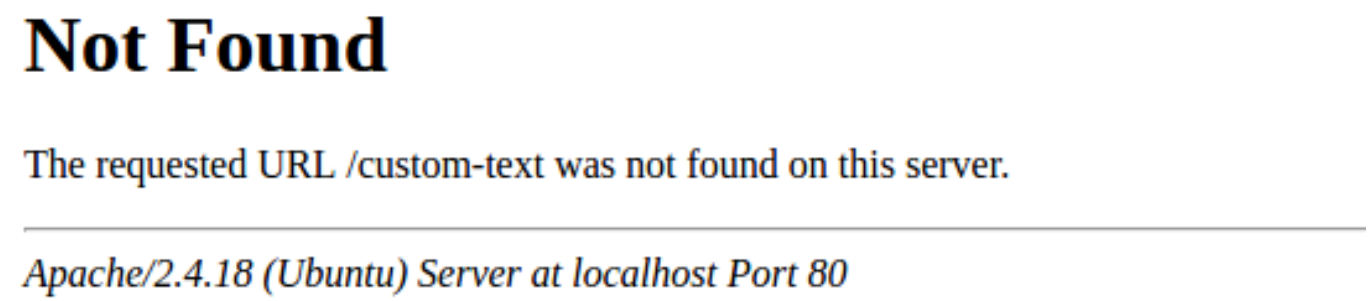
Para segurança adicional, você pode usar a ferramenta *mod_security* com o Apache (pacote *libapache2-mod-security2*). Esta ferramenta permite remover a assinatura do servidor em vez de remover apenas a versão da assinatura do servidor. A assinatura do servidor pode ser alterada para qualquer valor definindo "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:



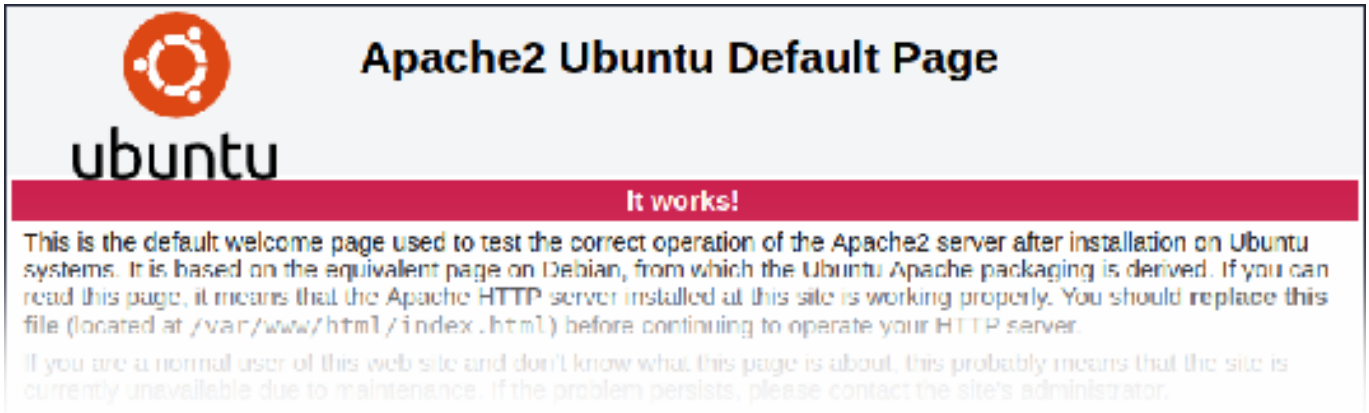
Essas páginas de erro padrão devem ser substituídas/removidas. Por exemplo, a diretiva "ErrorDocument" pode ser usada para definir uma página/texto de erro personalizado para o servidor web Apache.

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, recomenda-se remover a 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 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 XSS e clickjacking. Usuários *Super admin* podem modificar os parâmetros *Usar sandboxing de iframe* e *Usar 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, é recomendável limitar o acesso ao arquivo `ui/data/top_passwords.t`
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 habilitado 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 desempenho ideal e facilidade de uso. As recomendações são baseadas nos conselhos dos desenvolvedores do Zabbix e na experiência prática de treinadores 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 você! 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 sim 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 a desorganização 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 possui vários hosts semelhantes, como *Switch de rede 1* e *Switch de rede 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, *Template de switch de rede*. Em seguida, agrupe os hosts semelhantes em um grupo de hosts, para o exemplo acima poderia ser *Switches de rede*. 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, items dependentes podem

ser configurados para extrair partes específicas dos dados dessa coleta 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 em 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 servidor Zabbix 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 servidor Zabbix 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 proxy Zabbix 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 servidor Zabbix, reduzindo a necessidade de abrir várias portas no firewall.
2. Você está monitorando locais remotos, filiais e/ou redes. Em caso de interrupção da rede entre o servidor Zabbix e seus locais remotos, os proxies Zabbix implantados nos locais remotos continuarão a coleta de dados e enviarão os dados coletados de volta ao servidor Zabbix sempre que a conexão de rede for restabelecida.
3. Você possui uma implantação em larga escala e deseja reduzir a carga no servidor Zabbix 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.

Macros secretas

Você pode querer usar macros de usuário secretas como texto secreto ou macros secretas de cofre.

Para maior segurança ao usar macros secretas de cofre, é recomendável configurar os valores das macros para serem recuperados pelo Zabbix server e proxies Zabbix de forma independente. Por padrão, os valores das macros secretas são recuperados pelo Zabbix server e propagados para os proxies Zabbix.

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 uma API 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 a biblioteca Python zabbix_utils para a API do Zabbix.

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 requisições Depois de configurar o frontend, você pode usar requisições HTTP remotas para chamar a API. Para isso, é necessário enviar requisições HTTP POST para o arquivo `api_jsonrpc.php` localizado no diretório do frontend. Por exemplo, se o seu frontend Zabbix estiver instalado em `https://example.com/zabbix`, uma requisição 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}'
```

A requisição deve ter o cabeçalho `Content-Type` definido para um destes valores: `application/json-rpc`, `application/json` ou `application/jsonrequest`.

O objeto da requisição 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 a ser chamado;
- `params` - os parâmetros que serão passados para o método da API;
- `id` - um identificador arbitrário da requisição (se omitido, a API trata a requisição como uma notificação).

Se a requisição estiver correta, a resposta retornada pela API deve ser semelhante a esta:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "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 requisição 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 requisição 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 "Authorization"

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 Authorization.

A API do Zabbix aceita cabeçalhos de forma case-insensitive (por exemplo, `authorization`, `Authorization` e `AUTHORIZATION` são tratados da mesma forma).

O cabeçalho Authorization é suportado em solicitações de origem cruzada (CORS).

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 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 próximas solicitações:

```
{
    "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 de 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 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 graciosamente 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 sua aplicação para usar recursos específicos de cada versão.

O Zabbix garante compatibilidade retroativa de recursos dentro de uma versão principal. Ao fazer alterações incompatíveis entre versões principais, o Zabbix normalmente 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 já tem conhecimento suficiente para começar a trabalhar com a API do Zabbix, no entanto, não pare por aqui. Para leitura adicional, recomenda-se dar uma olhada na lista de APIs disponíveis.

Referência de métodos

Esta seção fornece uma visão geral das funções fornecidas pela API do Zabbix e ajudará você a se orientar nas classes e métodos disponíveis.

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

Recupera uma lista de nós do server e seus status.

API de cluster de alta disponibilidade

Histórico

Recupera valores históricos coletados pelos processos de monitoramento do Zabbix para apresentação ou processamento adicional.

API de Histórico

Tendências

Recupera valores de tendências calculados pelo Zabbix server para apresentação ou processamento posterior.

Trend API

Eventos

Recupere eventos gerados por triggers, descoberta de rede e outros sistemas Zabbix para um gerenciamento de situações mais flexível ou integração com ferramentas de terceiros.

API de Evento

Problemas

Recupera problemas de acordo com os parâmetros fornecidos.

API de Problemas

Mapas

Configure mapas para criar representações dinâmicas detalhadas de sua infraestrutura de TI.

API de Mapas

Tarefas

Interaja com o gerenciador de tarefas do Zabbix server, criando tarefas e recuperando respostas.

Task API

Serviços A API do Zabbix permite acessar os dados coletados durante o monitoramento de serviços.

Acordo de Nível de Serviço

Defina Objetivos de Nível de Serviço (SLO), recupere informações detalhadas dos Indicadores de Nível de Serviço (SLI) sobre o desempenho do serviço.

API de SLA

Serviços

Gerencie serviços para monitoramento de nível de serviço e recupere informações detalhadas de SLA sobre qualquer serviço.

API de Serviço

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.

Host API | Host group API | Host interface API | User macro API | Value map API | Maintenance API

Items

Defina os items a serem monitorados.

API de item

Triggers

Configure triggers para notificá-lo sobre problemas em seu sistema. Gerencie dependências de triggers.

API de trigger

Gráficos

Edite gráficos ou itens de gráfico separados para melhor apresentação dos dados coletados.

API de gráfico | API de item de gráfico

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

Configure regras de descoberta de baixo nível, bem como protótipos de item, trigger e gráfico para monitorar entidades dinâmicas.

API de regra de LLD | API de protótipo de item | API de protótipo de trigger | API de protótipo de gráfico | API de protótipo de host

Correlação de eventos

Crie regras personalizadas de correlação de eventos.

API de correlação

Descoberta de rede

Gerencie regras de descoberta em nível de rede para encontrar e monitorar automaticamente novos hosts. Tenha acesso total às informações sobre serviços e hosts descobertos.

API de regra de descoberta | API de verificação de descoberta | API de host descoberto | API de serviço descoberto

Exportar e importar

Exporte e importe dados de configuração do Zabbix para backups de configuração, migração ou atualizações de configuração em larga escala.

API de configuração

Monitoramento web

Configure cenários web para monitorar suas aplicações e serviços web.

API de cenário web

Alertas A API do Zabbix permite que você gerencie as ações e alertas do seu sistema de monitoramento.

Ações e alertas

Defina ações e operações para notificar usuários sobre determinados eventos ou executar comandos remotos automaticamente. Obtenha acesso a informações sobre alertas gerados e seus destinatários.

API de Ação | API de Alerta

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 ajudar nas tarefas diárias.

Script API

Usuários A API do Zabbix permite gerenciar 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

Configurar housekeeping.

Housekeeping API

Proxies e grupos de proxies

Gerencie os proxies usados em sua configuração de monitoramento distribuído.

Proxy API | Proxy group API

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 - cria novas ações
- action.delete - exclui ações
- action.get - recupera ações
- action.update - atualiza ações

Objeto action

Os seguintes objetos estão diretamente relacionados à API `action`.

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.

Propriedade	Tipo	Descrição
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. Ele 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, será usado o período de escalonamento padrão da ação. 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”
evaltype	integer	Método de avaliação da condição da operação. Valores possíveis: 0 - *(padrão)* AND / OR; 1 - AND; 2 - OR.

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 `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
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”

Propriedade	Tipo	Descrição
opinventory	object	Modo de inventário definido para o 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*

Propriedade	Tipo	Descrição
value	string	Valor a ser comparado. 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”
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

Propriedade	Tipo	Descrição
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 da 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 realizada 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*
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 `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 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*

Propriedade	Tipo	Descrição
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 de filtro de ação

O objeto de condição de 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 Zabbix agent, 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.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 do 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 na propriedade `actionids`. A ordem dos IDs retornados corresponde à 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.

Requisição:

```
{
    "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. Uma vez que um trigger com severidade maior ou igual a ”Warning” do host ”10084” ou do host ”10106” entre em estado de PROBLEM, 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 autorregistro de agent

Crie uma ação de autorregistro 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 adiciona um host ao grupo de hosts "2" e adiciona 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() em *ui/include/classes/api/services/CAction.php*.

action.delete

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 na propriedade `actionids`.

Exemplos

Excluir várias ações

Exclua duas ações.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

Origem

CAction::delete() em *ui/include/classes/api/services/CAction.php*.

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	Retorna apenas ações com os IDs fornecidos.
groupids	ID/array	Retorna apenas ações que usam os grupos de hosts fornecidos nas condições da ação.
hostids	ID/array	Retorna apenas ações que usam os hosts fornecidos nas condições da ação.
triggerids	ID/array	Retorna apenas ações que usam os triggers fornecidos nas condições da ação.
mediatypeids	ID/array	Retorna apenas ações que usam os tipos de mídia fornecidos para enviar mensagens.
usrgrpids	ID/array	Retorna apenas ações que estão configuradas para enviar mensagens para os grupos de usuários fornecidos.
userids	ID/array	Retorna apenas ações que estão configuradas para enviar mensagens para os usuários fornecidos.
scriptids	ID/array	Retorna apenas ações que estão configuradas para executar os scripts fornecidos.
selectFilter	query	Retorna uma propriedade `filter` com o filtro de condição da ação.
selectOperations	query	Retorna uma propriedade `operations` com as operações da ação.
selectRecoveryOperations	query	Retorna uma propriedade `recovery_operations` com as operações de recuperação da ação.
selectUpdateOperations	query	Retorna uma propriedade `update_operations` com as operações de atualização da ação.
sortfield	string/array	Ordena 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:

- um 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() em *ui/include/classes/api/services/CAction.php*.

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"

Parâmetro	Tipo	Descrição
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 na propriedade `actionids`.

Exemplos

Desabilitar ação

Desabilita uma ação, ou seja, define 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

Origem

CAction::update() em *ui/include/classes/api/services/CAction.php*.

Alert

Esta classe foi projetada para trabalhar com alertas.

Referências de objetos:

- Alert

Métodos disponíveis:

- alert.get - recuperar alertas

Objeto alert

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 de 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.
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.

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	Retornar apenas alertas com os IDs fornecidos.
actionids	ID/array	Retornar apenas alertas gerados pelas ações fornecidas.
eventids	ID/array	Retornar apenas alertas gerados pelos eventos fornecidos.
groupids	ID/array	Retornar apenas alertas gerados por objetos dos grupos de host fornecidos.
hostids	ID/array	Retornar apenas alertas gerados por objetos dos hosts fornecidos.
mediatypeids	ID/array	Retornar apenas alertas de mensagem que usaram os tipos de mídia fornecidos.
objectids	ID/array	Retornar apenas alertas gerados pelos objetos fornecidos
userids	ID/array	Retornar apenas alertas de mensagem que foram enviados para os usuários fornecidos.
eventobject	integer	Retornar 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	Retornar 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	Retornar apenas alertas que foram gerados após o tempo fornecido.
time_till	timestamp	Retornar apenas alertas que foram gerados antes do tempo fornecido.
selectHosts	query	Retornar uma propriedade `hosts` com dados dos hosts que dispararam a operação da ação.
selectMediatypes	query	Retornar 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	Retornar 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	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperar alertas por ID de ação

Recuperar 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*.

Autenticação

Esta classe foi projetada para trabalhar com configurações de autenticação.

Referências de objetos:

- Autenticação

Métodos disponíveis:

- authentication.get - recuperar configurações de autenticação
- authentication.update - atualizar configurações de autenticação

Objeto de autenticação

Os seguintes objetos estão diretamente relacionados à API de `authentication`.

Autenticação

O objeto de autenticação possui as seguintes propriedades.

Propriedade	Tipo	Descrição
authentication_type	integer	Autenticação padrão. Valores possíveis: 0 - *(padrão)* Interna; 1 - LDAP.
http_auth_enabled	integer	Autenticação HTTP. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada. Comportamento da propriedade: - *suportada* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
http_login_form	integer	Formulário de login padrão. Valores possíveis: 0 - *(padrão)* Formulário de login do Zabbix; 1 - Formulário de login HTTP. Comportamento da propriedade: - *suportada* se `$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: - *suportada* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
http_case_sensitive	integer	Login HTTP com diferenciação de maiúsculas e minúsculas. Valores possíveis: 0 - Desligado; 1 - *(padrão)* Ligado. Comportamento da propriedade: - *suportada* se `$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.

Propriedade	Tipo	Descrição
ldap_userdirectoryid	ID	ID do diretório de usuários 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.
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 o 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 para atribuir ao 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"

Propriedade	Tipo	Descrição
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"

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. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Propriedades de autenticação a serem atualizadas.

Valores de retorno

`(array)` Retorna um array com os nomes dos parâmetros atualizados.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "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*.

Autoregistration

Esta classe foi projetada para trabalhar com autoregistration.

Referências de objeto:

- Autoregistration

Métodos disponíveis:

- autoregistration.get - recuperar autoregistration
- autoregistration.update - atualizar autoregistration

Objeto de autorregistro

Os seguintes objetos estão diretamente relacionados à API de `autoregistration`.

Autoregistro

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 emparelhada 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*
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *somente escrita*

autoregistration.get

Descrição

`object autoregistration.get(object parameters)`

O método permite recuperar o objeto de autorregistro 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 autorregistro.

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() em *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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

(object) Propriedades de autoregistro a serem atualizadas.

Valores de retorno

(boolean) Retorna verdadeiro como resultado em caso de 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 de 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 user agent 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.
delay	string	Intervalo de execução do cenário web. Aceita segundos ou unidade de tempo com sufixo (ex.: 30s, 1m, 2h, 1d), ou uma macro de usuário. Padrão: 1m.
headers	array	Cabeçalhos HTTP que serão enviados ao realizar uma requisição.
http_password	string	Senha usada para autenticação HTTP básica ou NTLM.

Propriedade	Tipo	Descrição
http_proxy	string	Proxy que será usado pelo cenário web, fornecido como *http://[usuário[:senha]@]proxy.exemplo.com[:porta]*.
http_user	string	Nome de usuário usado para autenticação HTTP básica ou NTLM.
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.
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 par; 1 - verificar par.
uuid	string	Identificador global único, usado para vincular cenários web importados a 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 de cenário web. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da etapa do cenário. Comportamento da propriedade: - *obrigatório*
no	integer	Número de sequência da etapa em um cenário web. Comportamento da propriedade: - *obrigatório*
url	string	URL a ser verificada. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
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.
retrieve_mode	integer	Parte da resposta HTTP que a etapa do cenário deve recuperar. Valores possíveis: 0 - *(padrão)* apenas o 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. 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 de usuário. Veja 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	Passos do cenário. Comportamento do parâmetro: - *obrigatório*
tags	array	Tags do cenário web.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos cenários web criados na propriedade `httptestids`. A ordem dos IDs retornados corresponde à ordem dos cenários 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 garantir que retornem 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() em *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 web excluídos na propriedade `httptestids`.

Exemplos

Excluindo múltiplos 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() em *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 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
groupids	ID/array	Retorna apenas cenários web que pertencem aos grupos de host fornecidos.
hostids	ID/array	Retorna apenas cenários web que pertencem aos hosts fornecidos.
httptestids	ID/array	Retorna apenas cenários web com os IDs fornecidos.
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 fornecidos.
expandName	flag	Expande macros no nome do cenário web.
expandStepName	flag	Expande macros nos nomes dos passos do cenário.
evaltype	integer	Regras para busca por tags. Valores possíveis: 0 - (padrão) E/Ou 2 - Ou
tags	array	Retorna apenas cenários web com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os cenários web. Tipos de operadores possíveis: 0 - (padrão) Contém (correspondência por substring, não sensível a maiúsculas/minúsculas) 1 - Igual (sensível a maiúsculas/minúsculas) 2 - Não contém (correspondência por substring, não sensível a maiúsculas/minúsculas) 3 - Diferente (sensível a maiúsculas/minúsculas) 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 os passos 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 fornecidas. Valores possíveis: `httptestid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	

Parâmetro	Tipo	Descrição
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

Origem

CHttpTest::get() em *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. Consulte 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 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

Origem

CHttpTest::update() em *ui/include/classes/api/services/CHttpTest.php*.

Configuração

Esta classe foi projetada para exportar e importar dados de configuração do Zabbix.

Métodos disponíveis:

- configuration.export - exportar dados de configuração
- configuration.import - importar dados de configuração
- configuration.importcompare - comparar arquivo de importação com os elementos atuais do sistema

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 em que 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 hosts 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 templates 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>8.0</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.) podem ser importados apenas 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`.

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`.
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 templates. 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 templates. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos dashboards de templates serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, dashboards de templates existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, dashboards de templates não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.

Parâmetro	Tipo	Descrição
triggers	object	Regras sobre como importar triggers. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas triggers serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, triggers existentes serão atualizadas; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, triggers não presentes nos dados importados serão excluídas do banco de dados; padrão: `false`.
valueMaps	object	Regras sobre como importar mapas de valores de host ou template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas de valores serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas de valores existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, mapas de valores 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>8.0</version><templ
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Fonte

CConfiguration::import() em *ui/include/classes/api/services/CConfiguration.php*.

configuration.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 para 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 para 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 para 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 para 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 valores de host ou template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas de valores serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas de valores existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, mapas de valores 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>8.0</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 administrativas comuns.

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 busca 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 sejam Super administradores e para usuários que não façam 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 em Últimos dados, Web e widgets de painel 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: Não classificado.
severity_name_1	string	Nome para o nível de severidade ”Informação”. Padrão: Informação.
severity_name_2	string	Nome para o nível de severidade ”Aviso”. Padrão: Aviso.

Propriedade	Tipo	Descrição
severity_name_3	string	Nome para o nível de severidade ”Média”. Padrão: Média.
severity_name_4	string	Nome para o nível de severidade ”Alta”. Padrão: Alta.
severity_name_5	string	Nome para o nível de severidade ”Desastre”. Padrão: Desastre.
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	Exibir período de triggers OK. Aceita segundos e unidade de tempo com sufixo. Padrão: 5m.
blink_period	string	Período de piscar de triggers ao mudar 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 RESOLVIDO não reconhecidos como código hexadecimal. Padrão: 009900.
ok_ack_color	string	Cor para eventos RESOLVIDO reconhecidos como código hexadecimal. Padrão: 009900.
problem_unack_style	integer	Piscar para eventos PROBLEM não reconhecidos. Valores possíveis: 0 - Não exibir piscar; 1 - *(padrão)* Exibir piscar.
problem_ack_style	integer	Piscar para eventos PROBLEM reconhecidos. Valores possíveis: 0 - Não exibir piscar; 1 - *(padrão)* Exibir piscar.
ok_unack_style	integer	Piscar para eventos RESOLVIDO não reconhecidos. Valores possíveis: 0 - Não exibir piscar; 1 - *(padrão)* Exibir piscar.
ok_ack_style	integer	Piscar para eventos RESOLVIDO reconhecidos. Valores possíveis: 0 - Não exibir piscar; 1 - *(padrão)* Exibir piscar.
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)* Desativado; 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	Fornecedor 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	Fornecedor de Vault. Valores possíveis: 0 - *(padrão)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Resolver valores de macro secreta por: Valores possíveis: 0 - *(padrão)* apenas server; 1 - server e proxies independentemente.
timeout_zabbix_agent	string	Gastar no máximo `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*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

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() em *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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Propriedades das 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() em *ui/include/classes/api/services/CSettings.php*.

Connector

Esta classe foi projetada para trabalhar com connectors.

Referências de objetos:

- Connector
- Tag filter

Métodos disponíveis:

- connector.create - criar novos connectors
- connector.delete - excluir connectors
- connector.get - recuperar connectors
- connector.update - atualizar 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 - Caracter; 4 - Log; 8 - Numérico (sem sinal); 10 - Texto; 20 - Binário. 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 se o certificado do host é autêntico. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
verify_host	integer	Se deve validar se 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 de tags. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.

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 de condição. Valores possíveis: 0 - *(padrão)* Igual; 1 - Diferente; 2 - Contém; 3 - Não contém; 12 - Existe; 1 - Não existe.
value	string	Valor da tag.

connector.create

Descrição

`object connector.create(object/array connectors)`

Este método permite criar novos objetos de conector.

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

`(objeto/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 na propriedade `connectorids`. A ordem dos IDs retornados corresponde à ordem dos conectores passados.

Exemplos

Criando um conector

Crie um conector para exportar eventos de trigger com um filtro de tag. A autenticação HTTP será realizada usando o token Bearer.

Requisição:

```
{
    "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() em *ui/include/classes/api/services/CConnector.php*.

connector.delete

Descrição

`object connector.delete(array connectorids)`

Este método permite excluir entradas de conector.

> **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 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
}
```

Origem

CConnector::delete() em *ui/include/classes/api/services/CConnector.php*.

connector.get

Descrição

`integer/array connector.get(object parameters)`

O método permite recuperar objetos de conector 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
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	

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

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*.

connector.update

Descrição

`object connector.update(object/array connectors)`

Este método permite atualizar conectores 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 conector a serem atualizadas.

A propriedade `connectorid` deve ser definida para cada conector, todas as outras propriedades são opcionais. Apenas 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 de retorno

`(object)` Retorna um objeto contendo os IDs dos conectores atualizados na 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
}
```

Atualizando o filtro de tag

Altere o filtro de tag para o conector com ID ”5”.

Requisição:

```
{
    "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
}
```

Origem

CConnector::update() em *ui/include/classes/api/services/CConnector.php*.

Correlação

Esta classe foi projetada para trabalhar com correlações.

Referências de objeto:

- 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 correlation

Os seguintes objetos 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 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 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 (antiga ou nova). Comportamento da propriedade: - *obrigatório* se `type` for 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` for definido como ”grupo de hosts do novo evento”
oldtag	string	Tag do evento antigo. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”par de tags de evento”
newtag	string	Tag do evento antigo. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”par de tags de evento”
value	string	Valor da tag do evento (antiga ou nova). Comportamento da propriedade: - *obrigatório* se `type` for 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` for 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

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 de usuário. Consulte 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 das correlações criadas na propriedade `correlationids`. A ordem dos IDs retornados corresponde 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 habilitada.

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

Origem

CCorrelation::create() em *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Descrição

`object correlation.delete(array correlationids)`

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. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das correlações a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das correlações excluídas na propriedade `correlationids`.

Exemplo

Excluir múltiplas correlações

Exclua duas correlações.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CCorrelation::delete() em *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

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 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar correlações

Recupera 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() em *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 de 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 das correlações atualizadas na 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

Origem

CCorrelation::update() 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 dashboard

Os seguintes objetos 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 da 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 (será usado 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 ação; clock - Relógio; discovery - Status de descoberta; favgraphs - Gráficos favoritos; favmaps - Mapas favoritos; gauge - Medidor; geomap - Geomapa; graph - Gráfico (clássico); graphprototype - Protótipo de gráfico; honeycomb - Favo de mel; hostavail - Disponibilidade do host; hostcard - Cartão do host; hostnavigator - Navegador de host; itemcard - Cartão do item; 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; topitems - Principais itens; toptriggers - Principais triggers; trigover - Visão geral de 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.

Propriedade	Tipo	Descrição
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 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-escrita. Comportamento da propriedade: - *obrigatório*

dashboard.create

Descrição

`object dashboard.create(object/array dashboards)`

Este método permite criar novos 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

`(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 em que forem especificadas. 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 dos dashboards criados na propriedade dashboardids. A ordem dos IDs retornados corresponde à ordem dos dashboards passados.

Exemplos

Criando um dashboard

Crie um dashboard chamado "Meu 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*.

dashboard.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 dashboards a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos dashboards excluídos na propriedade `dashboardids`.

Exemplos

Excluindo múltiplos dashboards

Exclua dois dashboards.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
```

```
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Origem

CDashboard::delete() em *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Descrição

`integer/array dashboard.get(object parameters)`

O método permite recuperar dashboards 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 com os IDs fornecidos.
selectPages	query	Retorna uma propriedade `pages` com as páginas do dashboard, corretamente ordenadas.
selectUsers	query	Retorna uma propriedade `users` com os usuários com quem o dashboard é compartilhado.
selectUserGroups	query	Retorna uma propriedade `userGroups` com os grupos de usuários com quem o dashboard é compartilhado.
sortfield	string/array	Ordena 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando um dashboard por 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

Descrição

`object dashboard.update(object/array dashboards)`

Este método permite atualizar dashboards 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 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 dos dashboards atualizados na 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 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 os parâmetros dos widgets do dashboard e possíveis valores de propriedades para os respectivos objetos dashboard widget field.

Para ver os parâmetros e valores de propriedades para 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
- Cartão do host
- Navegador de hosts
- Cartão de item

- Histórico do item
- Navegador de itens
- Valor do item
- Mapa
- Árvore de navegação de mapas
- Gráfico de pizza
- Hosts com problemas
- Problemas
- Relatório de SLA
- Informações do sistema
- Problemas por severidade
- Top hosts
- Top itens
- Top triggers
- Visão geral dos triggers
- URL
- Monitoramento web

1 Log de ações

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Log de ações* 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 *Log de ações*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Log de ações*.

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.
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	type	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 do dashboard 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
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
Ordenar entradas por	0	sort_triggers	3 - Hora (ascendente); 4 - *(padrão)* Hora (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 visam apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Action log*. Para mais informações sobre a configuração de um dashboard, 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 foram bem-sucedidas.

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

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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Relógio*.

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 server; 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. 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.

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 exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Clock*. Para obter mais informações sobre como configurar um dashboard, consulte `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 de widget de dashboard

- dashboard.create
- dashboard.update

3 Status da 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 *Status da descoberta*.

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 dashboard para o widget *Discovery status*. Para obter mais informações sobre como configurar um dashboard, 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`

4 Gráficos favoritos

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do 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	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 *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`

5 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 descrever apenas 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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`

6 Gauge

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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. Comportamento do parâmetro: - *obrigatório* se *Item (Widget)* não estiver definido

Parâmetro	tipo	nome	valor
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 fonte de dados para os items. Comportamento do parâmetro: - *obrigatório* se *Item* não estiver definido
Mín	1	min	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).
Exibir	0	show.0	1 - Descrição; 2 - Valor; 3 - Ponteiro; 4 - Escala; 5 - Arco de valor. 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, 4, 5. Os valores "Ponteiro" e "Escala" não são suportados se ambos: - nenhum objeto de campo de widget de dashboard para *Exibir* com o valor "Arco de valor" estiver definido; - o parâmetro de configuração avançada *Exibir arco* estiver definido como "Desabilitado". Os parâmetros de configuração avançada para opções de *Exibir* não são suportados se nenhum objeto de campo de widget de dashboard com os respectivos valores estiver definido.
Substituir host	1	override_hostid._reference	`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”
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”

Parâmetro	tipo	nome	valor
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.
Ponteiro			
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”
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”

Parâmetro	tipo	nome	valor
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 painel para o widget *Gauge*. Para obter mais informações sobre como configurar um painel, 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 de widget do dashboard
- `dashboard.create`
- `dashboard.update`

7 Geomapa

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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 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 *Geomap*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Geomap*.

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 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.

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 pertençam. 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 de host do dashboard 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.
Tags			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se estiver configurando o widget 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 estiver configurando o widget em um dashboard de template.
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* Este parâmetro não é suportado se estiver configurando o widget 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* Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
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`.
Agrupamento			
Modo de agrupamento	0	clustering_mode	0 - *(padrão)* Automático; 1 - Nível de zoom.
Nível de zoom para agrupamento	0	clustering_zoom_level	Valores possíveis variam de 0-30. Padrão: 0. Comportamento do parâmetro: - *suportado* se o *Modo de agrupamento* estiver definido como "Nível de zoom"
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 dashboard para o widget *Geomap*. Para obter mais informações sobre como configurar um dashboard, 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`

8 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 descritos 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 exclusivo 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. 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.
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"

Parâmetro	tipo	nome	valor
Desenhar	0	ds.0.type	0 - *(padrão)* Linha; 1 - Pontos; 2 - Degrau; 3 - Barra.
Empilhado	0	ds.0.stacked	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha”, ”Degrau” ou ”Barra”
Largura	0	ds.0.width	Valores possíveis variam de 1-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha” ou ”Degrau”
Tamanho do ponto	0	ds.0.pointsize	Valores possíveis variam de 1-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Pontos”
Transparência	0	ds.0.transparency	Valores possíveis variam de 1-10. Padrão: 5.
Preenchimento	0	ds.0.fill	Valores possíveis variam de 1-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha” ou ”Degrau”
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 ”Degrau”
Sobrescrever host	1	ds.0.override_hostid.ABCDE_hostid	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.
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).
Função de agre-gação	0	ds.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
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.
Agregação	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”
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”

Parâmetro	tipo	nome	valor
Linha de percentil (direita)			
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._referenceDASHBOARD.	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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 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
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

Eixos

Os seguintes parâmetros são suportados para configurar *Eixos*.

Parâmetro	tipo	nome	valor
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"

Parâmetro	tipo	nome	valor
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"
Escala	0	lefty_scale righty_scale	0 - *(padrão)* Linear; 1 - Logarítmica.
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
Exibir legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Exibir min/méd/máx	0	legend_statistic	Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado" 0 - *(padrão)* Desabilitado; 1 - Habilitado.
Exibir função de agregação	0	legend_aggregation	Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado" 0 - *(padrão)* Desabilitado; 1 - Habilitado.
Linhas	0	legend_lines_mode	Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado" 0 - *(padrão)* Fixo; 1 - Variável.
Número de linhas/ Número máximo de linhas	0	legend_lines	Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado" 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 min/méd/máx* 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 com problemas	1	problemhosts.0	Nome do host. Nota: O número no nome da propriedade faz referência ao host configurado. Para configurar múltiplos 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.
Severidade	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.
Problema	1	problem_name	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).
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	tipo	nome	valor
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*

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, `*: Número de *valores processados por segundo`). Ao configurar o widget em um dashboard de template, apenas os padrões para 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 a 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 a 10.
Preenchimento	0	or.0.fill	Os valores possíveis variam de 1 a 10.
Tamanho do ponto	0	or.0.pointsize	Os valores possíveis variam de 1 a 10.
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.

Parâmetro	tipo	nome	valor
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 de dashboard para o widget *Gráfico*. Para mais informações sobre a configuração de 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 ”Lista de items” 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 ”Padrões de items”, 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.
- 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.

Requisição:

```
{
    "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.1",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "FF0000"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23269
```

```
            },
            {
                "type": 1,
                "name": "ds.0.color.2",
                "value": "BF00FF"
            },
            {
                "type": 4,
                "name": "ds.0.itemids.3",
                "value": 23257
            },
            {
                "type": 1,
                "name": "ds.0.color.3",
                "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": 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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`

9 Gráfico (clássico)

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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.
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"

Parâmetro	type	name	value
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._referenc	`DASHBOARD._timeperiod` - defina o seletor de período de tempo do dashboard 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 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
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
Exibir legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Substituir host	1	override_hostid._refere	`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.
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 *Gráfico (clássico)*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Graph (classic)*

Configure um widget *Graph (classic)* 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 do dashboard
- `dashboard.create`
- `dashboard.update`

10 Protótipo de gráfico

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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 dos parâmetros descritos 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"
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._referenda	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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*.

Parâmetro	tipo	nome	valor
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
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
Exibir legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
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 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 a 24. Padrão: 2.
Linhas	0	rows	Os valores possíveis variam de 1 a 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 de widget do dashboard
- `dashboard.create`
- `dashboard.update`

11 Honeycomb

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 descrito 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.
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 de host do dashboard 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			

Parâmetro	tipo	nome	valor
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*
Operador	0	host_tags.0.operator	Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template. 0 - Contém; 1 - Igual a; 2 - Não contém; 3 - Diferente de; 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*
Valor da tag	1	host_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 de host*
Padrões de item	1	items.0	Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template. 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*

Parâmetro	tipo	nome	valor
Operador	0	item_tags.0.operator	0 - Contém; 1 - Igual a; 2 - Não contém; 3 - Diferente de; 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*
Exibir hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Exibir	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 com 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 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
Rótulo primário			
Tipo	0	primary_label_type	0 - *(padrão)* Texto; 1 - Valor.

Parâmetro	tipo	nome	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 do 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.
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"

Parâmetro	tipo	nome	valor
Casas decimais	0	secondary_label_decimals	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 do 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 do objeto User: `D9E7ED` para ”blue-theme”; `3D5059` para ”dark-theme”; `AAD7E9` para ”hc-light”; `335463` para ”hc-dark”.
Limiares			
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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`

12 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 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 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.
Tipo de interface	0	interface_type.0	0 - Nenhum; 1 - Zabbix agent (verificações passivas); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix agent (verificações ativas). Padrão: 1, 2, 3, 4, 5 (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.
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 *Disponibilidade do host*

Configure um widget *Disponibilidade do host* 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 do widget do dashboard
- `dashboard.create`
- `dashboard.update`

13 Cartão de host

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Host card* 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 *Host card*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Host card*.

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.
Host	3	hostid.0	ID do Host. Comportamento do parâmetro: - *obrigatório* se *Host (Widget/Dashboard)* não estiver definido Este parâmetro não é suportado ao configurar o widget em um dashboard de template.
Host (Widget/Dashboard)	1	hostid._reference	Em vez do ID do Host: `DASHBOARD.hostid` - define o seletor de host do dashboard como a fonte de dados para o host; `ABCDE._hostid` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para o host. Comportamento do parâmetro: - *obrigatório* se *Host* não estiver definido Este parâmetro não é suportado ao configurar o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar	0	sections.0	0 - Grupos de host; 1 - Descrição; 2 - Monitoramento; 3 - Disponibilidade; 4 - Monitorado por; 5 - Templates; 6 - Inventário; 7 - Tags. Nota: O número no nome da propriedade referencia a ordem da seção na lista de seções. Para configurar várias seções, crie um objeto de campo de widget de dashboard para cada seção com um número incrementado no nome da propriedade.

Os seguintes parâmetros são suportados se *Mostrar* estiver definido como ”Inventário”.

Parâmetro	tipo	nome	valor
Campos de inventário	0	inventory.0	ID do Inventário. Nota: Para configurar vários campos de inventário, crie um objeto de campo de widget de dashboard para cada campo de inventário com um número incrementado no nome da propriedade.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Host card*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Host card*

Configure um widget *Host card* que exibe estas seções: ”Monitoring”, ”Availability”, ”Monitored by”, ”Inventory” e ”Tags”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
```

```
                            "name": "hostid.0",
                            "value": 10084
                        },
                        {
                            "type": 0,
                            "name": "show_suppressed",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "sections.0",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "sections.1",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "sections.2",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "sections.3",
                            "value": 6
                        },
                        {
                            "type": 0,
                            "name": "sections.4",
                            "value": 7
                        },
                        {
                            "type": 0,
                            "name": "inventory.0",
                            "value": 25
                        },
                        {
                            "type": 0,
                            "name": "inventory.1",
                            "value": 26
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"id": 1
```

```
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

14 Navegador de hosts

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 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 descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Navegador de 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.
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
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.
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.

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 múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Exibir hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
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 - 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 com 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 *Navegador de hosts*

Configure um widget *Navegador de hosts* que exibe hosts agrupados por seu grupo de hosts e, em seguida, pelo valor da tag "city".

Requisição:

```
{
    "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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`

15 Cartão 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 *Item card* 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 *Item card*, consulte o comportamento do parâmetro descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Item card*.

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 o item. Comportamento do parâmetro: - *obrigatório* se *Item* não estiver definido

Parâmetro	tipo	nome	valor
Mostrar	0	sections.0	0 - Descrição; 1 - Texto de erro; 2 - Métricas; 3 - Últimos dados; 4 - Tipo de informação; 5 - Triggers; 6 - Interface do host; 7 - Tipo; 8 - Inventário do host; 9 - Tags. Nota: O número no nome da propriedade faz referência à ordem da seção na lista de seções. Para configurar várias seções, crie um objeto de campo de widget de dashboard para cada seção com um número incrementado no nome da propriedade.
Substituir 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. Padrão: `""` (vazio) Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Sparkline

Os seguintes parâmetros são suportados se *Mostrar* estiver definido como "Últimos dados".

Parâmetro	tipo	nome	valor
Largura	0	sparkline.width	Os valores possíveis variam de 0 a 10. Padrão: 1.
Preenchimento	0	sparkline.fill	Os valores possíveis variam de 0 a 10. Padrão: 3.
Cor	1	sparkline.color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `42A5F5`.
Período de tempo	1	sparkline.time_period._reference	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro `reference` igual a `ABCDE`) como fonte de dados. Padrão: `""` (vazio) Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	sparkline.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.). Padrão: now-1h. Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido

Parâmetro	tipo	nome	valor
Até	1	sparkline.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.). Padrão: now. Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido
Dados históricos	0	sparkline.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.

Exemplos

Os exemplos a seguir têm como objetivo descrever apenas a configuração dos objetos de campo do widget do painel para o widget *Item card*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Item card*

Configure um widget *Item card* que exibe estas seções: ”Descrição”, ”Últimos dados”, ”Triggers” e ”Tags”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
```

```
                                "name": "sections.3",
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

16 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 descritos 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)			
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 de host do dashboard como fonte de dados para hosts. Este parâmetro não é suportado ao configurar o widget 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 possuem 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) 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*
Item	4	columns.0.itemid	ID do item. Ao configurar o widget em um dashboard de template, apenas 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.
Mín	1	columns.0.min	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibição* estiver definido como ”Barra” ou ”Indicadores”
Máx	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	type	name	value
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	tipo	nome	valor
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.
Mostrar cabeçalho da coluna	0	show_column_header	0 - Desligado; 1 - Horizontal; 2 - *(padrão)* Vertical.
Período de tempo	1	time_period._referenceDASHBOARD	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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
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

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Histórico do item*. Para mais informações sobre a configuração de um dashboard, 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 separada para o timestamp.

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 de widget do dashboard
- `dashboard.create`
- `dashboard.update`

17 Navegador de item

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 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	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 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.hostid` - defina o seletor de host do dashboard 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	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.
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.
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 - 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	tipo	nome	valor
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 único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo descrever apenas a configuração dos objetos de campo do widget do painel para o widget *Navegador de item*. Para mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Navegador de item*

Configure um widget *Navegador de item* 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`

18 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 fonte de dados para os itens. Comportamento do parâmetro: - *obrigatório* se *Item* não estiver definido
Mostrar	0	show.0	1 - Descrição; 2 - Valor; 3 - Hora; 4 - Indicador de alteração; 5 - Sparkline. Padrão: 1, 2, 3, 4. 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.
Substituir host	1	override_hostid._reference	`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 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 agregação	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 do dashboard 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	time_period.from	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na 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”
Até	1	time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na 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”
Dados de histórico	0	history	0 - *(padrão)* Auto; 1 - History; 2 - Trends.

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 1-100. Padrão: 15.
Negrito	0	desc_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	desc_color	Código de cor hexadecimal (ex.: 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. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Posição vertical	0	value_v_pos	0 - Superior; 1 - *(padrão)* Meio; 2 - Inferior. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Tamanho	0	value_size	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 1-100. 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).

Sparkline

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Sparkline”.

Parâmetro	tipo	nome	valor
Largura	0	sparkline.width	Os valores possíveis variam de 0 a 10. Padrão: 1.

Parâmetro	tipo	nome	valor
Preenchimento	0	sparkline.fill	Os valores possíveis variam de 0 a 10. Padrão: 3.
Cor	1	sparkline.color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: 42A5F5.
Período de tempo	1	sparkline.time_period.from sparkline.time_period.to	DASHBOARD._timeperiod - define o seletor de período de tempo do dashboard 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*.
De	1	sparkline.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.).
Até	1	sparkline.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.).
Dados de histórico	0	sparkline.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Valor do item*. Para mais informações sobre a configuração de um dashboard, 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`

19 Mapa

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 em 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 descrever apenas a configuração dos objetos de campo do widget de dashboard para o widget *Mapa*. Para obter mais informações sobre a configuração de um dashboard, 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

20 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 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.
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 descrever apenas 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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`
- Mapa

21 Gráfico de pizza

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico de pizza*.

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.

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 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. 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 de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *Tipo de conjunto de dados* estiver definido como "Lista de items"

Parâmetro	tipo	nome	valor
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 *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.
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 de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *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.
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 *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. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Rosquinha”
Exibir 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 *Exibir valor total* estiver definido como ”Habilitado”
Tamanho (valor para tamanho personalizado)	0	value_size	Os valores possíveis variam de 1-100. Padrão: 20. Comportamento do parâmetro: - *suportado* se *Exibir 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 *Exibir valor total* estiver definido como ”Habilitado”
Unidades (checkbox)	0	units_show	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir valor total* estiver definido como ”Habilitado”
Unidades (valor)	1	units	Qualquer valor de string. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como ”Habilitado”
Negrito	0	value_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir valor total* estiver definido como ”Habilitado”
Cor	1	value_color	Código de cor hexadecimal (ex: FF0000). Comportamento do parâmetro: - *suportado* se *Exibir valor total* estiver definido como ”Habilitado”

Parâmetro	tipo	nome	valor
Espaço entre setores	0	space	Os valores possíveis variam de 0-10. Padrão: 1.
Mesclar setores menores que N% (checkbox)	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%* (checkbox) estiver definido como "Habilitado"
Mesclar setores menores que N% (cor)	1	merge_color	Código de cor hexadecimal (ex: FF0000). Comportamento do parâmetro: - *suportado* se *Mesclar setores menores que N%* (checkbox) 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._referenc	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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
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

Legenda

Os seguintes parâmetros são suportados para configurar a *Legenda*.

Parâmetro	tipo	nome	valor
Mostrar legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Mostrar valor	0	legend_value	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado”
Mostrar função de agregação	0	legend_aggregation	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado”
Linhas	0	legend_lines_mode	0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *suportado* se *Mostrar 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 *Mostrar legenda* estiver definido como ”Habilitado”
Número de colunas	0	legend_columns	Os valores possíveis variam de 1-4. Padrão: 4. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado” e *Mostrar 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 que são 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.1",
                        "value": 23264
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.1",
                        "value": "FF0000"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.type.1",
                        "value": 0
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.2",
                        "value": 23269
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.2",
                        "value": "BF00FF"
                    },
                    {
                      "type": 0,
                      "name": "ds.0.type.2",
                      "value": 0
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.3",
                        "value": 23257
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.3",
                        "value": "0040FF"
                    },
                    {
                      "type": 0,
                      "name": "ds.0.type.3",
                      "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": 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 do widget do dashboard
- `dashboard.create`
- `dashboard.update`

22 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	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 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.
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.
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 de host do dashboard 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).
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.

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 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*
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 do 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
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

23 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 dos parâmetros descritos 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.

Parâmetro	tipo	nome	valor
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.
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.
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 de host do dashboard 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).
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.

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 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 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.

Parâmetro	tipo	nome	valor
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 o widget estiver sendo configurado em um dashboard de template.
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)”
Mostrar linhas	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 mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Problemas*

Configure um widget *Problemas* 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: ”Aviso”, ”Média”, ”Alta”, ”Desastre”.

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`

24 Problemas por gravidade

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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 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.
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.
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.

Parâmetro	tipo	nome	valor
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor de host do dashboard 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).
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.
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*
Exibir	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 é definido como "Totais".

Parâmetro	tipo	nome	valor
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Totais”
Exibir dados operacionais	0	show_opdata	0 - *(padrão)* Nenhum; 1 - Separadamente; 2 - Com o nome 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. Comportamento do parâmetro: - *suportado* se *Exibir* 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.
Exibir 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 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 dashboard para o widget *Problemas por severidade*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Problemas por gravidade*

Configure um widget *Problemas por gravidade* que exibe o total 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`

25 Relatório de SLA

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do 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	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.
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 1-100. 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 descrever apenas 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
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

26 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 têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Informações do sistema*. Para obter mais informações sobre como configurar um dashboard, 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`

27 Principais hosts

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do 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.
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.
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 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 de host do dashboard 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.
Tags de host			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade referencia a 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 estiver configurando o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
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 referencia a 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 estiver configurando o widget em um dashboard de template.
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade referencia a 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 estiver configurando o widget 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 1-1000. Padrão: 10. Este parâmetro não é suportado se estiver configurando o widget 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 válido do item. Ao configurar o widget em um dashboard de template, apenas os itens configurados no template devem ser definidos.
Exibir valor do item como	0	columns.0.display_value_as	0 - *(padrão)* Numérico; 1 - Texto; 2 - Binário.
Exibir	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores; 6 - Sparkline. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como ”Numérico”
Mínimo	1	columns.0.min	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Máximo	1	columns.0.max	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Casas dec-i-mais	0	columns.0.decimal_places	Os valores possíveis variam de 0-10. Padrão: 2. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como ”Numérico”
Sparkline			

Parâmetro	tipo	nome	valor
Largura	0	columns.0.sparkline.width	Os valores possíveis variam de 0-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Preenchimento	0	columns.0.sparkline.fill	Os valores possíveis variam de 0-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Cor	1	columns.0.sparkline.color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `42A5F5`. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Período de tempo	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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 *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
De	1	columns.0.sparkline.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 *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Até	1	columns.0.sparkline.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 *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Dados históricos	0	columns.0.sparkline.history	0 *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibir* estiver definido como "Sparkline"
Thresholds			
Cor	1	columnsthresholds.0.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico"

Parâmetro	tipo	nome	valor
Threshold	1	columns.0.thresholds.0.threshold	Qualquer valor de string. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico"
Destaques			
Cor	1	columns.0.highlights.0.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Texto"
Padrão	1	columns.0.highlights.0.pattern	Qualquer valor de string. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Texto"
Mostrar miniatura	0	columns.0.show_thumbnail	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Binário"
Função de agregação	0	columns.0.aggregate_function	0 - *(padrão)* não usada; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Período de tempo	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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"
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"

Parâmetro	tipo	nome	valor
Dados históricos	0	columns.0.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *suportado* se *Exibir valor do item como* estiver definido como "Numérico"
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 *Data* 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 *Data* 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 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 %", "1m avg", "5m avg", "15m avg", "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 de widget do dashboard
- `dashboard.create`
- `dashboard.update`

28 Principais items

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Itens principais* 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 *Itens principais*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Top items*.

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 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.

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 pertençam. 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 de host do dashboard 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.
Tags de host			
Tipo de avaliação	0	host_tags_evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se estiver configurando o widget 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 estiver configurando o widget 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 estiver configurando o widget 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 estiver configurando o widget em um dashboard de template.

Parâmetro	type	name	value
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical.
Mostrar problemas	0	show_problems	0 - Todos; 1 - *(padrão)* Não suprimidos; 2 - Nenhum.
Items (veja abaixo)			

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados para o widget *Top items*.

Parâmetro	tipo	nome	valor
Ordenação de hosts			
Ordenar por	0	host_ordering_order_by	0 - *(padrão)* Nome do host; 3 - Valor do item.
Padrões de item	1	host_ordering_item.0	Nome ou padrão do item (por exemplo, "*: Número de valores processados por segundo"). 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. Ao configurar o widget em um dashboard de template, apenas os padrões para itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se *Ordenar por* estiver definido como "Valor do item"
Ordem	0	host_ordering_order	2 - *(padrão)* Top N; 3 - Bottom N.
Limite	0	host_ordering_limit	Os valores possíveis variam de 1-100. Padrão: 10.
Ordenação de itens			
Ordenar por	0	item_ordering_order_by	1 - Host; 2 - Nome do item; 3 - *(padrão)* Valor do item.
Padrões de host	1	item_ordering_host.0	Nome ou padrão do host. Nota: Para configurar múltiplos padrões de host, crie um objeto de campo de widget de dashboard para cada padrão de host com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Ordenar por* estiver definido como "Host"
Ordem	0	item_ordering_order	2 - *(padrão)* Top N; 3 - Bottom N.
Limite	0	item_ordering_limit	Os valores possíveis variam de 1-100. Padrão: 10.

Colunas

Note:
O primeiro número no nome da propriedade (por exemplo, columns.0.items.0, columns.0.item_tags_evaltype) representa a coluna específica, enquanto o segundo número, se presente, representa a entidade configurada (por exemplo, padrão do item, tag).

Parâmetro	tipo	nome	valor
Padrões de item	1	columns.0.items.0	Nome ou padrão do item (por exemplo, "*: Número de valores processados por segundo"). Nota: Para configurar vários padrões de item, crie um objeto de campo de widget de dashboard para cada padrão de item com um número incremental na segunda posição do nome da propriedade. 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*
Tags do item			
Tipo de avaliação	0	columns.0.item_tags_evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	columns.0.item_tags.0.tag	Qualquer valor de string. Nota: O segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se configurar *Tags do item*
Operador	0	columns.0.item_tags.0.operator	0 - Contém; 1 - Igual a; 2 - Não contém; 3 - Diferente de; 4 - Existe; 5 - Não existe. Nota: O segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se configurar *Tags do item*
Valor da tag	1	columns.0.item_tags.0.value	Qualquer valor de string. Nota: O segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se configurar *Tags do item*
Cor base	1	columns.0.base_color	Código de cor hexadecimal (por exemplo, FF0000).
Exibir valor como	0	columns.0.display_value_as	1 - *(padrão)* Numérico; 2 - Texto.
Exibir	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores; 6 - Sparkline.

Parâmetro	tipo	nome	valor
Mínimo	1	columns.0.min	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Numérico” e *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Máximo	1	columns.0.max	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Numérico” e *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Sparkline			
Largura	0	columns.0.sparkline.width	Os valores possíveis variam de 0-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Sparkline”
Preenchimento	0	columns.0.sparkline.fill	Os valores possíveis variam de 0-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Sparkline”
Cor	1	columns.0.sparkline.color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `42A5F5`. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Sparkline”
Período de tempo	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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 *Exibir* estiver definido como ”Sparkline”
De	1	columns.0.sparkline.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 *Exibir* estiver definido como ”Sparkline”
Até	1	columns.0.sparkline.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 *Exibir* estiver definido como ”Sparkline”
Dados de histórico	0	columns.0.sparkline.history	0 *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Sparkline”

Parâmetro	tipo	nome	valor
Limiares			
Cor	1	columns.0.thresholds.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Numérico”
Limiar	1	columns.0.thresholds.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Numérico”
Destaques			
Destaque	1	columns.0.highlights.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Texto”
Padrão	1	columns.0.highlights.pattern	Qualquer expressão regular. Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Texto”
Casas decimais	0	columns.0.decimal_places	Os valores possíveis variam de 0-10. Padrão: 2.
Configuração avançada			
Função de agregação	0	columns.0.aggregate_function	Se *Exibir valor como* estiver definido como ”Numérico”: 0 - *(padrão)* não usado; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. Se *Exibir valor como* estiver definido como ”Texto”: 0 - *(padrão)* não usado; 4 - count; 6 - first; 7 - last.
Período de tempo	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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”

Parâmetro	tipo	nome	valor
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”
Dados de histórico	0	columns.0.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *suportado* se *Exibir valor como* estiver definido como ”Numérico”

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Top items*. Para mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Top items*

Configure um widget *Top items* que exiba dados para o host ”10084” e apenas para os items para os quais a tag com o nome ”component” contenha o valor ”cpu”. Além disso, exiba os dados com os hosts localizados no topo e use uma barra de medidor colorida para a representação da célula.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
                        "name": "Top items",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
```

```
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

29 Principais triggers

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de 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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Top triggers*.

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.
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 pertencem. 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 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.

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.
Operador	0	tags.0.operator	Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema* 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.
Valor da tag	1	tags.0.value	Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema* Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags.
Período de tempo	1	time_period._referenceDASHBOARD	Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema* `DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard 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
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
Limite de trig-gers	0	show_lines	Os valores possíveis variam de 1-100. Padrão: 10.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Top triggers*. Para 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 gravidades "Aviso", "Média", "Alta" ou "Desastre", 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 de widget de dashboard
- `dashboard.create`
- `dashboard.update`

30 Visão geral do trigger

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.

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 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 de host do dashboard 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 - 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*
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Visão geral dos triggers*. Para mais informações sobre a configuração de um dashboard, consulte `dashboard.create`.

Configurando um widget *Visão geral de triggers*

Configure um widget *Visão geral de triggers* que exibe os estados das triggers para todos os grupos de hosts que possuem triggers com uma tag que tem o nome "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`

31 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 de 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 descrever apenas a configuração dos objetos de campo do widget de dashboard para o widget *URL*. Para mais informações sobre a configuração de 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/8.0/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 de widget do dashboard
- `dashboard.create`
- `dashboard.update`

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 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.
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.
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.

Parâmetro	tipo	nome	valor
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor de host do dashboard 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 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 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 visam 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 *Web monitoring*

Configure um widget *Web monitoring* 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 do 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 com o 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*

Propriedade	Tipo	Descrição
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*
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 do diretório 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 atributos de mídia do usuário LDAP/SAML (por exemplo, e-mail) para 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 personalizado 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 name ID SAML deve ser criptografado. Valores possíveis: 0 - *(padrão)* Não criptografar name ID; 1 - Criptografar name ID. 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 name ID 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”

Propriedade	Tipo	Descrição
idp_certificate	string	Conteúdo do certificado do Provedor de Serviços (SP) para configuração do serviço de logon único (SSO) SAML. Exemplos: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente gravação*
sp_private_key	string	Conteúdo da Chave Privada do Provedor de Serviços (SP) para configuração do serviço de logon único (SSO) SAML. Fornece autenticação segura e troca de dados com o Provedor de Identidade (IdP). Exemplos: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente gravação*
sp_certificate	string	Conteúdo do certificado do Provedor de Serviços (SP) para configuração do serviço de logon único (SSO) SAML. Exemplos: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente gravação*
idp_certificate_hash	string	O hash md5 do valor de idp_certificate. Retornará uma string vazia para idp_certificate vazio. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente leitura*
sp_private_key_hash	string	O hash md5 do valor de sp_private_key. Retornará uma string vazia para sp_private_key vazio. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente leitura*

Propriedade	Tipo	Descrição
sp_certificate_hash	string	O hash md5 do valor de sp_certificate. Retornará uma string vazia para sp_certificate vazio. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` em zabbix.conf.php - *somente leitura*

Mapeamentos de tipos de mídia

O objeto de mapeamentos de tipos 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 tipos 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 habilitado") 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 de 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)* habilitado; 1 - desabilitado.
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 habilitado"). Suporta o caractere curinga "*". Único em todos os mapeamentos de grupos de provisionamento. Comportamento da propriedade: - *obrigatório*
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 de diretório de usuário.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos diretórios de usuário criados na propriedade `userdirectoryids`. A ordem dos IDs retornados corresponde à ordem dos diretórios de usuário 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() em *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 está sendo usado diretamente por pelo menos um grupo de usuários.
O diretório de usuários LDAP padrão não pode ser excluído quando `authentication.ldap_configured` está definido como 1 ou quando ainda 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 múltiplos 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
}
```

Origem

CUserDirectory::delete() em *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 usuários. Suporta `count`.
selectProvisionMedia	query	Retorna uma propriedade `provision_media` com mapeamentos de tipos de mídia associados a um diretório de usuários.
selectProvisionGroups	query	Retorna uma propriedade `provision_groups` com mapeamentos de grupos de provisionamento associados a um diretório de usuários.
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 busca `LIKE "%…%"`. Propriedades suportadas: `name`, `description`. O diretório de usuários do tipo SAML terá um 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` tiver sido usado

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 solicitação 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 usuário.

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 usuário, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
test_username	string	Nome de usuário para testar no diretório de usuário.
test_password	string	Senha associada ao nome de usuário para testar no diretório de usuário.

Valores de retorno

`(bool)` Retorna verdadeiro em caso de sucesso.

Exemplos

Testar o diretório de usuários para um 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

Testa 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
}
```

Origem

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. Somente 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

Atualizar 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
}
```

Origem

CUserDirectory::update() em *ui/include/classes/api/services/CUserDirectory.php*.

Evento

Esta classe foi projetada para trabalhar com eventos.

Referências de objetos:

- Evento
- Tag de evento
- URL do tipo de mídia

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 ativo do agent; 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 ativo do agent": 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	Severidade 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 causador.
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 URL do tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da URL definida no tipo de mídia.
url	string	Valor da URL definida no tipo de mídia.

Os resultados conterão entradas apenas para tipos de mídia ativos com a 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.
- Desmarcar 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.

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 de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Parâmetros contendo os IDs dos eventos e as operações de atualização que devem ser realizadas.

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 - remover supressão do 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 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 na 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": "Manutenção necessária para corrigir.",
        "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 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
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 de 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 de 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 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 eventos com os IDs dos usuários que realizaram as ações de atualização de evento.
suppressed	boolean	`true` - retorna apenas eventos suprimidos; `false` - retorna eventos no estado normal.
symptom	boolean	`true` - retorna apenas eventos de sintoma; `false` - retorna apenas eventos de causa.
severities	integer/array	Retorna apenas eventos com as severidades fornecidas. Aplica-se apenas se o objeto for trigger.
trigger_severities	integer/array	Retorna apenas eventos com as severidades de trigger fornecidas. Aplica-se apenas se o objeto for trigger.
evaltype	integer	Regras para busca por tag. Valores possíveis: 0 - (padrão) And/Or; 2 - Or.

Parâmetro	Tipo	Descrição
tags	array	Retorna apenas eventos com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os eventos. Tipos de operador possíveis: 0 - (padrão) Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 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 que foram criados após ou no horário fornecido.
time_till	timestamp	Retorna apenas eventos que foram criados antes ou no horário fornecido.
problem_time_from	timestamp	Retorna apenas eventos que estavam no estado de problema a partir de `problem_time_from`, independentemente do 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 do 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 do 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 realizada, consulte `event.acknowledge`; `old_severity` - `(integer)` severidade do evento antes desta ação de atualização; `new_severity` - `(integer)` severidade 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.

Parâmetro	Tipo	Descrição
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 correspondência. Não suporta propriedades do tipo de dado `text` data type.
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	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado, mas o parâmetro `groupBy` não foi utilizado;
- um array de objetos com resultados de agregação, se o parâmetro `groupBy` foi utilizado.

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 especificado

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 principais triggers 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 - Cadeia de caracteres incluída; 1 - Qualquer cadeia de caracteres incluída; 2 - Cadeia 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 cadeia 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. Consulte 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 na 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
}
```

Origem

CRegexp::create() em *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 de 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 na 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
}
```

Origem

CRegexp::delete() em *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Descrição

`integer/array regexp.get(parâmetros do objeto)`

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

`(objeto)` 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	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 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() em *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 de usuário. Consulte 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. Apenas 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() em *ui/include/classes/api/services/CRegexp.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 graph

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
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 convertido de protótipo. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
graphtype	integer	Tipo de layout do gráfico. Valores possíveis: 0 - *(padrão)* normal; 1 - empilhado; 2 - pizza; 3 - explodido.
percent_left	float	Percentil esquerdo. Padrão: 0.
percent_right	float	Percentil direito. Padrão: 0.
show_3d	integer	Se deve mostrar gráficos de pizza e explodidos 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 tempo de trabalho no gráfico. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
show_triggers	integer	Se deve mostrar a linha de 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.
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”.

Propriedade	Tipo	Descrição
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 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. Consulte 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 na 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. Consulte 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 na 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
}
```

Origem

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. Consulte 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.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta de gráficos. O objeto de descoberta de gráficos vincula um gráfico descoberto a um protótipo de gráfico do qual foi descoberto. Possui as seguintes propriedades: `parent_graphid` - `(ID)` ID do protótipo de gráfico a partir do qual o gráfico foi criado; `status` - `(int)` status da descoberta do gráfico: 0 - *(padrão)* gráfico é descoberto, 1 - gráfico não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um gráfico que não é mais descoberto será excluído.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra de descoberta de baixo nível que criou o 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.
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.

Parâmetro	Tipo	Descrição
selectItems	query	Retorna uma propriedade `items` com os items usados no gráfico.
selectGraphItems	query	Retorna uma propriedade `gitems` com os items usados no 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	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	
selectGraphDiscovery	query	Retorna uma propriedade `graphDiscovery` com o objeto de descoberta de gráficos. Os objetos de descoberta de gráficos vinculam o gráfico a um protótipo de gráfico a partir do qual foi criado. Esta consulta está **obsoleta**, por favor, use `selectDiscoveryData` em seu lugar.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi 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 de 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. Consulte 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 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 dos gráficos atualizados na 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
}
```

Origem

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 a 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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Grupos de host a serem criados.

O método aceita grupos de host com as propriedades padrão do grupo de host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de hosts criados na 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:

- contiver hosts que pertençam apenas a este grupo;
- estiver marcado como interno;
- for usado por um protótipo de host;
- for usado em um script global;
- 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 dos grupos de host excluídos na propriedade `groupids`.

Exemplos

Excluindo múltiplos 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
}
```

Origem

CHostGroup::delete() em *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 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
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	flag	Retorna apenas grupos de hosts que contenham hosts com gráficos.
with_graph_prototypes	flag	Retorna apenas grupos de hosts que contenham hosts com protótipos de gráficos.
with_hosts	flag	Retorna apenas grupos de hosts que contenham hosts.
with_httptests	flag	Retorna apenas grupos de hosts que contenham hosts com verificações web. Sobrescreve o parâmetro `with_monitored_httptests`.
with_items	flag	Retorna apenas grupos de hosts que contenham hosts com items. Sobrescreve os parâmetros `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	flag	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	flag	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	flag	Retorna apenas grupos de hosts que contenham hosts com verificações web habilitadas.
with_monitored_hosts	flag	Retorna apenas grupos de hosts que contenham hosts monitorados.
with_monitored_items	flag	Retorna apenas grupos de hosts que contenham hosts com items habilitados. Sobrescreve o parâmetro `with_simple_graph_items`.
with_monitored_triggers	flag	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	flag	Retorna apenas grupos de hosts que contenham hosts com items numéricos.

Parâmetro	Tipo	Descrição
with_triggers	flag	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 de LLD que descobriram o grupo de hosts.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` 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; `ts_delete` - `(timestamp)` momento 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 ordenados por `host`.
sortfield	string/array	Ordena 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	
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. Esta consulta está **obsoleta**, por favor, use `selectDiscoveryData` em seu lugar.

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",
            "internal": "0"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "internal": "0"
        }
    ],
    "id": 1
}
```

Veja também

- Host

Origem

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 de usuário. Consulte 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() em *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 do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de host a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
groupids	ID/array	IDs dos grupos de host a serem atualizados. Comportamento do parâmetro: - *obrigatório*
hostids	ID/array	IDs dos hosts a serem removidos de todos os grupos de host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de host atualizados na propriedade `groupids`.

Exemplos

Removendo hosts de grupos de hosts

Remove 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() em *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 host a serem atualizados e os objetos que devem ser atualizados.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de host a serem atualizados. Os grupos de host devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
hosts	object/array	Hosts para substituir os hosts atuais nos grupos de host fornecidos. Todos os outros hosts, exceto os mencionados, serão excluídos dos grupos de host. 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 dos grupos de hosts atualizados na 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 host 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 de usuário. Veja 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. Apenas as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de host atualizados na 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 - cria novos grupos de proxy
- proxygroup.delete - exclui grupos de proxy
- proxygroup.get - recupera grupos de proxy
- proxygroup.update - atualiza 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 do 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 de usuário. Consulte 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
}
```

Origem

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	
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. Somente 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
}
```

Origem

CProxyGroup::update() em *ui/include/classes/api/services/CProxyGroup.php*.

Grupo de templates

Esta classe foi projetada para trabalhar com grupos de templates.

Referências de objetos:

- Grupo de templates

Métodos disponíveis:

- templategroup.create - criar novos grupos de templates
- templategroup.delete - excluir grupos de templates
- templategroup.get - recuperar grupos de templates
- templategroup.massadd - adicionar objetos relacionados a grupos de templates
- templategroup.massremove - remover objetos relacionados de grupos de templates
- templategroup.massupdate - substituir ou remover objetos relacionados de grupos de templates
- templategroup.propagate - propagar permissões para subgrupos de grupos de templates
- templategroup.update - atualizar grupos de templates

Objeto de grupo de template

Os seguintes objetos estão diretamente relacionados à API `templategroup`.

Grupo de template

O objeto 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 único universal, usado para vincular grupos de template importados aos 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 do grupo de template.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template criados sob a 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 pertençam apenas a esse 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. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos grupos de templates 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	flag	Retorna apenas grupos de templates que contenham templates com gráficos.
with_graph_prototypes	flag	Retorna apenas grupos de templates que contenham templates com protótipos de gráficos.
with_httptests	flag	Retorna apenas grupos de templates que contenham templates com verificações web.
with_items	flag	Retorna apenas grupos de templates que contenham templates com items. Sobrescreve os parâmetros `with_simple_graph_items`.
with_item_prototypes	flag	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	flag	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	flag	Retorna apenas grupos de templates que contenham templates com items numéricos.
with_templates	flag	Retorna apenas grupos de templates que contenham templates.
with_triggers	flag	Retorna apenas grupos de templates que contenham templates com triggers.

Parâmetro	Tipo	Descrição
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` tiver sido usado.

Exemplos

Recuperando dados pelo nome

Recupere todos os dados sobre dois grupos de templates 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

Origem

CTemplateGroup::get() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Descrição

`object templategroup.massadd(object parameters)`

Este método permite adicionar simultaneamente múltiplos 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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de template a serem atualizados e os objetos a serem adicionados a todos os grupos de template.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de template a serem atualizados. Os grupos de template devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
templates	object/array	Templates a serem adicionados a todos os grupos de template. 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 de 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 de 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 templates.

> **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 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 sob a 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

Origem

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 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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do grupo de templates a serem atualizadas.

A propriedade `groupid` deve ser definida para cada grupo de templates, 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

Renomear 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
}
```

Origem

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 de 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.

Propriedade	Tipo	Descrição
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).
mfaid	ID	Método de MFA usado para os usuários do grupo. Comportamento da propriedade: - *suportado* para usuários do tipo *Super admin* - *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* - *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 ao qual 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 de usuário. Veja 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.

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 de usuário. Consulte 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 na 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
}
```

Origem

CUserGroup::delete() em *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 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
mfaids	ID/array	Retorna apenas grupos de usuários com os métodos MFA fornecidos. Comportamento do parâmetro: - *suportado* para usuários do tipo *Super admin*
mfa_status	integer	Retorna apenas grupos de usuários com o status MFA fornecido. Consulte a página do grupo de usuários para uma lista de status suportados.
status	integer	Retorna apenas grupos de usuários com o status fornecido. Consulte a página do grupo de usuários para 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 permissões do grupo de usuários para grupos de hosts na propriedade permissões em `hostgroup_rights`. Consulte a página do grupo de usuários para uma lista de níveis de acesso aos grupos de hosts.
selectTemplateGroupRights	query	Retorna permissões do grupo de usuários para grupos de templates na propriedade permissões em `templategroup_rights`. Consulte a página do grupo de usuários para uma lista de níveis de acesso aos grupos de templates.
limitSelects	integer	Limita o número de registros retornados por subconsultas.

Parâmetro	Tipo	Descrição
output	query	Propriedades do objeto grupo de usuários a serem retornadas. Usuários dos tipos *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	
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 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

- User

Fonte

CUserGroup::get() em *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 de grupo de hosts para substituir as permissões atuais atribuídas ao grupo de usuários.
templategroup_rights	object/array	Permissões de 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.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de usuários atualizados na 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

Origem

CUserGroup::update() em *ui/include/classes/api/services/CUserGroup.php*.

Histórico

Esta classe foi projetada para trabalhar com dados de histórico.

Referências de objeto:

- Histórico de float
- Histórico de inteiro
- Histórico de string
- Histórico de texto
- Histórico de log

Métodos disponíveis:

- history.clear - limpa dados de histórico
- history.get - recupera 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 de 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.

history.clear

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. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos items a serem limpos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos items limpos 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*.

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 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
history	integer	Tipos de objeto de histórico a serem retornados. Valores possíveis: 0 - numérico float; 1 - caractere; 2 - log; 3 - *(padrão)* numérico sem sinal; 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.

Parâmetro	Tipo	Descrição
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:

- um array 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 de 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 obter 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 históricos do item

Enviar dados históricos do 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*.

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 convertido a partir de um protótipo. 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 emparelhada 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 gravação* - *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 gravação* - *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 a items.

ID	Propriedade	Tipo	Descrição	Comprimento máximo
4	alias	string	Apelido.	128 caracteres
11	asset_tag	string	Etiqueta de patrimônio.	64 caracteres
28	chassis	string	Chassi.	64 caracteres
23	contact	string	Pessoa de contato.	65535 caracteres para bancos de dados SQL
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.	65535 caracteres para bancos de dados SQL
39	host_netmask	string	Máscara de sub-rede do host.	39 caracteres
38	host_networks	string	Redes do host.	65535 caracteres para bancos de dados SQL
40	host_router	string	Roteador do host.	39 caracteres
30	hw_arch	string	Arquitetura do hardware.	32 caracteres
33	installer_name	string	Nome do instalador.	64 caracteres
24	location	string	Localização.	65535 caracteres para bancos de dados SQL
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

ID	Propriedade	Tipo	Descrição	Comprimento máximo
3	name	string	Nome.	128 caracteres
27	notes	string	Notas.	65535 caracteres para bancos de dados SQL
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 SO.	128 caracteres
6	os_full	string	Nome detalhado do SO.	255 caracteres
7	os_short	string	Nome curto do SO.	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.	65535 caracteres para bancos de dados SQL
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.	65535 caracteres para bancos de dados SQL
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.	65535 caracteres para bancos de dados SQL
55	site_rack	string	Localização do rack do site.	128 caracteres
52	site_state	string	Estado do site.	64 caracteres
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.	65535 caracteres para bancos de dados SQL
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	Fabricante.	64 caracteres

Tag do host

O objeto de tag do 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 do 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 adicionar o host. 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 na 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.

Requisição:

```
{
    "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 do host
- Inventário do host
- Tag do 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 dos hosts a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts excluídos na 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 os web checks informados.
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.

Parâmetro	Tipo	Descrição
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 web checks. Sobrescreve o parâmetro `with_monitored_httptests`.
with_monitored_httptests	flag	Retorna apenas hosts que possuem web checks habilitados.
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	Retorna hosts que possuem problemas suprimidos. Valores possíveis: `null` - *(padrão)* todos os hosts; `true` - apenas hosts com problemas suprimidos; `false` - apenas hosts com problemas não suprimidos.
evaltype	integer	Regras para busca por tags. Valores possíveis: 0 - *(padrão)* And/Or; 2 - Or.
severities	integer/array	Retorna hosts que possuem apenas problemas com as severidades informadas. Aplica-se apenas se o objeto de problema for trigger.
tags	object/array	Retorna apenas hosts com as tags informadas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os hosts. Valores possíveis para o operador: 0 - *(padrão)* Contém (correspondência por substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência por substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 4 - Existe; 5 - Não existe.
inheritedTags	boolean	Retorna hosts que possuem as `tags` informadas também em todos os seus templates vinculados. Padrão: Valores possíveis: `true` - os templates vinculados também devem possuir as `tags` informadas; `false` - *(padrão)* as tags dos templates vinculados são ignoradas.

Parâmetro	Tipo	Descrição
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta de host. O objeto de descoberta de host vincula um host descoberto a um protótipo de host do qual ele foi descoberto. Possui as seguintes propriedades: `host` - `(string)` ID do host; `parent_hostid` - `(string)` ID do protótipo de host do qual o host foi criado; `status` - `(int)` status da descoberta do host: 0 - *(padrão)* host foi descoberto, 1 - host não é mais descoberto; `ts_delete` - `(timestamp)` momento em que um host que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` momento 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.
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).
selectDiscoveryRules	query	Retorna uma propriedade `discoveryRules` com as regras LLD do host. Suporta `count`.
selectGraphs	query	Retorna uma propriedade `graphs` com os gráficos do host. Suporta `count`.
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`.
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.

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. 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`; `selectDiscoveryRules` - ordenados por `name`; `selectTriggers` - ordenados por `description`; `selectGraphs` - ordenados por `name`; `selectDashboards` - ordenados por `name`.
search	object	Retorna resultados que correspondam ao padrão informado (não sensível a maiúsculas/minúsculas). 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. Suporta propriedades adicionais: propriedades de interface de host.
searchInventory	object	Retorna hosts que possuem dados de inventário que correspondam ao padrão informado (não sensível a maiúsculas/minúsculas). 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.
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	
selectDiscoveries	query	Retorna uma propriedade `discoveries` com as regras de descoberta de baixo nível do host. Suporta `count`. Esta consulta está **obsoleta**, por favor, utilize `selectDiscoveryRules` em seu lugar.

Parâmetro	Tipo	Descrição
selectHostDiscovery	query	Retorna uma propriedade `hostDiscovery` com os dados do objeto de descoberta de host. Esta consulta está **obsoleta**, por favor, utilize `selectDiscoveryData` em seu lugar.

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 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 by 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 de 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
}
```

Pesquisando 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
}
```

Pesquisando host com tags e tags de template

Recupere 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

Recupere 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
}
```

Recupere 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
}
```

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 na 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 remover os hosts fornecidos.
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 na 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

Origem

CHost::massRemove() em *ui/include/classes/api/services/CHost.php*.

host.massupdate

Descrição

`object host.massupdate(object parameters)`

Este método permite substituir ou remover simultaneamente objetos relacionados 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 na 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 ou estiver vazio) 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 de host para substituir as tags de host atuais. Todas as tags que não estiverem listadas na solicitação serão removidas.
inventory	object	Propriedades do Inventário de 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.

Note:
Ao contrário do frontend do Zabbix, quando `name` (nome visível do host) é igual a `host` (nome técnico do host), atualizar `host` via API não atualizará automaticamente `name`. Ambas as propriedades precisam ser atualizadas explicitamente.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados na 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

Desvincular e limpar dois templates de um 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 do 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 descobertos

Converta a macro criada pela regra de descoberta de ”automática” 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 proxy

Origem

CHost::update() em *ui/include/classes/api/services/CHost.php*.

Host descoberto

Esta classe foi projetada para trabalhar com hosts descobertos.

Referências de objetos:

- Host descoberto

Métodos disponíveis:

- dhost.get - recupera hosts descobertos

Objeto de 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

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.

Parâmetro	Tipo	Descrição
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.
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 classificados por `dserviceid`.
sortfield	string/array	Classifica 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperar hosts descobertos por regra de descoberta

Recupera todos os hosts e os serviços descobertos que estão em execução 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.

Propriedade	Tipo	Descrição
hk_events_trigger	string	Período de armazenamento dos dados de trigger. Aceita segundos e unidade de tempo com sufixo. Padrão: 365d.
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 autoregistro. 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: 31d.
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.

Propriedade	Tipo	Descrição
hk_trends_mode	integer	Habilitar housekeeping interno para tendências. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_trends_global	integer	Sobrescrever o período de tendência 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 compressão 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 compressão TimescaleDB para histórico e tendências. Valores possíveis: 0 - *(padrão)* Desligado; 1 - Ligado.
compress_older	string	Comprimir 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. 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 housekeeping.

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

CHousekeeping ::get() em *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. Consulte 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
}
```

Origem

CHousekeeping::update() em *ui/include/classes/api/services/CHousekeeping.php*.

Imagem

Esta classe foi projetada para trabalhar com imagens.

Referências de objetos:

- Imagem

Métodos disponíveis:

- image.create - criar novas imagens
- image.delete - excluir imagens
- image.get - recuperar imagens
- image.update - atualizar imagens

Objeto de imagem

Os seguintes objetos estão diretamente relacionados à API de `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. 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 na propriedade `imageids`. A ordem dos IDs retornados corresponde à ordem das imagens passadas.

Exemplos

Criar uma imagem

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
}
```

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 na 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	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

Recuperar uma imagem

Recupere todos os dados da imagem com 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*.

image.update

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. Consulte 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 na 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*.

Informações da API

Esta classe foi projetada para recuperar metainformações sobre a API.

Métodos disponíveis:

- apiinfo.version - recupera a versão da API do Zabbix

apiinfo.version

Descrição

`string apiinfo.version(array)`

Este método permite recuperar a versão da API do Zabbix.

> **Attention:**
> Este método está disponível apenas para usuários não autenticados.

Parâmetros

`(array)` O método aceita um array vazio.

Valores de retorno

`(string)` Retorna a versão da API do Zabbix.

> **Note:**
> A partir do Zabbix 2.0.4, a versão da API corresponde à versão do Zabbix.

Exemplos

Recuperando a versão da API

Recupere a versão da API do Zabbix.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Fonte

CAPIInfo::version() em *ui/include/classes/api/services/CAPIInfo.php*.

Interface de host

Esta classe foi projetada para trabalhar com interfaces de host.

Referências de objeto:

- 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 como uma string vazia.

Propriedade	Tipo	Descrição
interfaceid	ID	ID da interface. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
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

Propriedade	Tipo	Descrição
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 determinado 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”
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.

Propriedade	Tipo	Descrição
community	string	Comunidade SNMP. Usada 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. Usada apenas por interfaces SNMPv3.
privpassphrase	string	Senha de privacidade SNMPv3. Usada apenas por interfaces SNMPv3.
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 de interface de host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das interfaces de host criadas na propriedade `interfaceids`. A ordem dos IDs retornados corresponde à ordem das interfaces de host passadas.

Exemplos

Criar uma nova interface

Crie uma interface de agent secundária 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

Origem

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. Consulte 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 das interfaces de host excluídas na 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 subseleções. Aplica-se às seguintes subseleções: `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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperar interfaces de host

Recupera 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 de 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 das interfaces de host criadas na 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 de 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 das interfaces de host excluídas na propriedade `interfaceids`.

Exemplos

Removendo interfaces

Remova 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. Consulte 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 das interfaces de host criadas na 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

Origem

CHostInterface::replaceHostInterfaces() em *ui/include/classes/api/services/CHostInterface.php*.

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 do 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 das interfaces de host atualizadas na 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 objetos descobertos
name	string	Nome do item. Suporta macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou objetos 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 objetos descobertos
url	string	String de URL. Suporta macros de usuário, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou objetos descobertos
value_type	integer	Tipo de informação do 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 ou objetos 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. 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 convertido de protótipo. 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 objetos descobertos
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 ou objetos 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 objetos 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 objetos descobertos
jmx_endpoint	string	String de conexão personalizada do Java gateway. 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 objetos descobertos

Propriedade	Tipo	Descrição
master_itemid	ID	ID do item mestre. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Dependent item” - *somente leitura* para objetos herdados ou objetos 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 objetos 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 objetos descobertos
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”) 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 objetos 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 objetos 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 da 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 da 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 executar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou objetos 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 objetos 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 objetos descobertos
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP agent” - *somente leitura* para objetos herdados ou objetos descobertos
ssl_cert_file	string	Caminho do arquivo da chave pública SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou objetos descobertos

Propriedade	Tipo	Descrição
ssl_key_file	string	Caminho do arquivo da chave privada SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou objetos 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 objetos 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 objetos 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 de valor. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float” ou ”numeric unsigned” - *somente leitura* para objetos herdados ou objetos 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` 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”) ou objetos descobertos
uuid	string	Identificador único universal, 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 objetos 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 objetos 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 objetos 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. 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 - 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), ”Mudança simples” (9), ”Mudança 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	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	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 se 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
[7] padrão Prometheus seguindo a sintaxe: `<nome da métrica>{<nome do label>="<valor do label>", ...} == <valor>`. Cada componente do padrão Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser macro de usuário.
[8] saída Prometheus seguindo a sintaxe: `<nome do label>` (pode ser macro de usuário) se `label` for selecionado como segundo parâmetro.
[9] Uma das funções de agregação: `sum`, `min`, `max`, `avg`, `count` se `function` for selecionado como 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 do 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": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
            {
                "tag": "equipment",
                "value": "workstation"
            }
        ],
        "delay": "30s"
    },
    "id": 1
}
```

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": "11312",
        "type": 4,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "units": "uptime",
        "interfaceid": "1156",
        "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. 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.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": "Script example",
        "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() em *ui/include/classes/api/services/CItem.php*.

item.delete

Descrição

`object item.delete(array itemIds)`

Este método permite excluir items.

Note:
Items da 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 do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos items a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos items excluídos na 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
}
```

Origem

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	Regras para busca de tags. Valores possíveis: 0 - (padrão) E/OU; 2 - OU.

Parâmetro	Tipo	Descrição
tags	array	Retorna apenas os items com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os items. Tipos de operadores possíveis: 0 - (padrão) Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 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 ao qual 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`.
selectGraphs	query	Retorna uma propriedade `graphs` com os gráficos que contêm o item. Suporta `count`.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta de item. O objeto de descoberta de item vincula um item descoberto a um protótipo de item do qual foi descoberto. Possui as seguintes propriedades: `parent_itemid` - `(string)` ID do protótipo de item do qual o item foi criado; `key_` - `(string)` chave do protótipo de item; `status` - `(int)` status da descoberta do item: 0 - *(padrão)* item foi 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.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD que criou o item.
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.

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 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	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	
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 do qual foi criado. Esta consulta está **obsoleta**, por favor use `selectDiscoveryData` em seu lugar.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Encontrando items por chave

Recupere todos os items usados em triggers para um 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 items dependentes pela chave

Recupere todos os items 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 HTTP agent

Encontre o item HTTP agent 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 do 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 de 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 items atualizados na 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": "Nome atualizado do item dependente",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Atualizar item HTTP agent

Habilitar o trapping de valor 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 ”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 eram usados pelo script anterior.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.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

CItem::update() em *ui/include/classes/api/services/CItem.php*.

Item do gráfico

Esta classe foi projetada para trabalhar com items de gráficos.

Referências de objetos:

- Item do gráfico

Métodos disponíveis:

- graphitem.get - recupera items de gráficos

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 do item de gráfico como um código hexadecimal. Comportamento da propriedade: - *obrigatório* para operações de criação
itemid	ID	ID do item. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
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 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 em ”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 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 item de gráfico suportados.
selectGraphs	query	Retorna uma propriedade `graphs` com um array de gráficos aos quais o item pertence.

Parâmetro	Tipo	Descrição
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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando os itens do 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

Origem

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 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 do usuário 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.
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.

Propriedade	Tipo	Descrição
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 ícones; 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; 54 - Autenticação multifator; 55 - Grupo de proxies; 56 - Protótipo de regra LLD.
resourceid	ID	Identificador do recurso da entrada do log de auditoria.
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.

Propriedade	Tipo	Descrição
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 do 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 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
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 utilizado.

Exemplos

Recuperar log de auditoria

Recuperar 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",
            "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",
            "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 objeto:

- Macro global
- Macro de host
  - Configuração de macro

Métodos disponíveis:

- usermacro.create - cria novas macros de host
- usermacro.createglobal - cria novas macros globais
- usermacro.delete - exclui macros de host
- usermacro.deleteglobal - exclui macros globais
- usermacro.get - recupera macros de host e globais
- usermacro.update - atualiza macros de host
- usermacro.updateglobal - atualiza macros globais

Objeto 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
description	string	Descrição da macro.
macro	string	String da macro. Comportamento da propriedade: - *obrigatório* para operações de criação
type	integer	Tipo de macro. Valores possíveis: 0 - *(padrão)* Macro de texto; 1 - Macro secreta; 2 - Segredo do Vault.
value	string	Valor da macro. Comportamento da propriedade: - *somente escrita* se `type` estiver definido como ”Macro secreta” - *obrigatório* para operações de criação

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. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização

Propriedade	Tipo	Descrição
automatic	integer	Define se a macro é controlada por uma regra de descoberta. Valores possíveis: 0 - *(padrão)* Macro gerenciada pelo usuário; 1 - Macro gerenciada por regra de descoberta. O usuário não tem permissão para criar macro automática. Para atualizar uma macro automática, ela deve ser convertida para manual.
config	object/array	Configuração da macro responsável por como a macro é exibida no Assistente de Host.
description	string	Descrição da macro.
hostid	ID	ID do host, protótipo de host ou template 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
type	integer	Tipo de macro. Valores possíveis: 0 - *(padrão)* Macro de texto; 1 - Macro secreta; 2 - Segredo do Vault.
value	string	Valor da macro. Comportamento da propriedade: - *somente escrita* se `type` estiver definido como "Macro secreta" - *obrigatório* para operações de criação

Configuração de macro

O objeto de configuração de macro define como uma macro é exibida no Assistente de host.

Propriedade	Tipo	Descrição
type	integer	Tipo de campo de entrada da macro. Valores possíveis: 0 - A macro não é usada no Assistente de host; 1 - Caixa de texto; 2 - Lista; 3 - Caixa de seleção. Comportamento da propriedade: - *obrigatório*
label	string	Rótulo para o campo de entrada da macro. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como "Caixa de texto", "Lista" ou "Caixa de seleção"
description	text	Texto de ajuda exibido ao lado do campo de entrada da macro. Suporta formatação Markdown.
priority	integer	Posição do campo de entrada da macro na lista de macros. Se `priority` não for definido, a macro será adicionada ao final da lista de macros não agrupadas.

Propriedade	Tipo	Descrição
required	integer	Marca a macro como obrigatória. Valores possíveis: 0 - Não obrigatória; 1 - Obrigatória. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Caixa de texto” ou ”Lista”
regex	string	Expressão regular para validar a entrada do usuário em um campo de caixa de texto. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Caixa de texto”
section_name	string	Rótulo da seção recolhível onde a macro é agrupada. Se `section_name` não for definido, a macro não será agrupada. Macros não agrupadas são exibidas primeiro; macros agrupadas são exibidas abaixo, ordenadas por sua `priority` dentro de cada seção recolhível.
options	text	String JSON definindo itens de lista ou valores de caixa de seleção. Para listas: um array de objetos com as propriedades `value` e `text`. Exemplo: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Para caixas de seleção: um objeto com as propriedades `checked` e `unchecked`. Exemplo: `{"checked": true, "unchecked": false}` Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Lista” ou ”Caixa de seleção”

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 na 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() em *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 de 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 na 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() em *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 do host a serem excluídas.

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() em *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 de usuário. Veja 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
}
```

Origem

CUserMacro::deleteGlobal() em *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	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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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 de host para um template

Recupere as macros de host definidas para o template ”10265” que contenham ”STATUS” no nome da macro.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "O nome do host ou endereço IP do host da página de status do Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
```

```
                "priority": "1",
                "section_name": "",
                "label": "Host da página de status do Apache",
                "description": "Nome do host ou endereço IP do host da página de status do Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "O caminho da URL da página de status do Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Caminho da página de status do Apache",
                "description": "Caminho da URL da página de status do Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "A porta da página de status do Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Porta da página de status do Apache",
                "description": "No intervalo de 1 a 65535 inclusive.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "O esquema de requisição, que pode ser HTTP ou HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Esquema de requisição",
                "description": "Esquema de requisição, que pode ser HTTP ou HTTPS.",
                "required": "0",
```

```
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "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() em *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 de 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. Somente 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 sob a 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

Converter a macro ”automática” criada pela regra de descoberta para ”manual” e alterar 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
}
```

Origem

CUserMacro::update() em *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() em *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 de `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 (inclusive). 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 (exclusive). 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.
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. Padrão: 3600.

Propriedade	Tipo	Descrição
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` estiver 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` estiver definido como ”diariamente”, ”semanalmente” ou ”mensalmente”
every	integer	Para períodos diários e semanais, `every` define os intervalos de dia ou semana nos quais a manutenção deve entrar em vigor. Valor padrão se `timeperiod_type` estiver 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` estiver 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` estiver 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` estiver definido como ”diariamente”, ”semanalmente” ou ”mensalmente”

Propriedade	Tipo	Descrição
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 de bitmap é aceitável (por exemplo, 21 para Segunda, Quarta e Sexta-feira). Comportamento da propriedade: - *obrigatório* se `timeperiod_type` estiver definido como ”semanalmente” ou se `timeperiod_type` estiver 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` estiver definido como ”mensalmente” e `dayofweek` não estiver definido
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 de bitmap é aceitável (por exemplo, 585 para Janeiro, Abril, Julho e Outubro). Comportamento da propriedade: - *obrigatório* se `timeperiod_type` estiver 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*

Propriedade	Tipo	Descrição
operator	integer	Operador de 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"

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das manutenções criadas na propriedade `maintenanceids`. A ordem dos IDs retornados corresponde à ordem das manutenções passadas.

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 22.01.2013 até 22.01.2014, entrar em vigor todo domingo às 18:00 e durar por uma hora.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1358844540,
        "active_till": 1390466940,
        "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

Origem

CMaintenance::create() em *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. Consulte 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
}
```

Origem

CMaintenance::delete() em *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.

Parâmetro	Tipo	Descrição
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`.
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 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. Consulte 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
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"

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() em *ui/include/classes/api/services/CMaintenance.php*.

Map

Esta classe foi projetada para trabalhar com mapas.

Referências de objetos:

- Map
- Elemento do mapa
  - Elemento do mapa Host
  - Elemento do mapa Grupo de hosts
  - Elemento do mapa Mapa
  - Elemento do mapa Trigger
  - Elemento do mapa Tag
  - Elemento do mapa URL
- Link do mapa
  - Trigger do link do mapa
  - Indicadores do link do mapa
- URL do mapa
- Usuário do mapa
- Grupo de usuários do mapa
- Formas do mapa
- Linhas do mapa

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. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
background_scale	integer	Se deve habilitar o redimensionamento proporcional da imagem de fundo. Valores possíveis: 0 - desabilitar redimensionamento; 1 - *(padrão)* habilitar redimensionamento da imagem.
expand_macros	integer	Se deve expandir macros nos rótulos ao configurar o mapa. Valores possíveis: 0 - *(padrão)* não expandir macros; 1 - expandir macros.
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 deve habilitar o alinhamento à grade. Valores possíveis: 0 - desabilitar alinhamento à grade; 1 - *(padrão)* habilitar alinhamento à grade.
grid_show	integer	Se deve mostrar a grade no mapa. Valores possíveis: 0 - não mostrar a grade; 1 - *(padrão)* mostrar 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 deve habilitar rótulos avançados. 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”

Propriedade	Tipo	Descrição
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”
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.

Propriedade	Tipo	Descrição
markelements	integer	Se deve destacar elementos do mapa que mudaram recentemente seu status. 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 das severidades de trigger suportadas.
show_element_label	integer	Como exibir os rótulos dos elementos por padrão. Valores possíveis: 0 - sempre exibir; 1 - *(padrão)* ocultar automaticamente.
show_link_label	integer	Como exibir os rótulos dos links por padrão. Valores possíveis: 0 - sempre exibir; 1 - *(padrão)* ocultar automaticamente.
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 Elemento. Comportamento da propriedade: - *obrigatório* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”

Propriedade	Tipo	Descrição
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.
evaltype	integer	Método de avaliação da condição de filtragem de tags do elemento do mapa. Valores possíveis: 0 - *(padrão)* AND / OR; 2 - OR.
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.

Propriedade	Tipo	Descrição
permission	integer	Tipo de nível de permissão. Valores possíveis: -1 - nenhum; 2 - somente leitura; 3 - leitura-escrita.
show_label	integer	Como exibir o rótulo do elemento. Valores possíveis: -1 - *(padrão)* padrão do mapa; 0 - sempre exibir; 1 - ocultar automaticamente.
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.
y	integer	Coordenadas Y do elemento em pixels. Padrão: 0.
zindex	integer	Valor usado para ordenar os elementos do mapa (z-index). 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.

Elemento de mapa Map

O objeto de elemento de mapa Map 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 de elemento de mapa

O objeto Tag de elemento de mapa possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do elemento de mapa. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador de condição da tag do elemento de mapa. Valores possíveis: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
value	string	Valor da tag do elemento de 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*
sysmapid	ID	ID do mapa ao qual o link pertence.
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*
drawtype	integer	Estilo de desenho da linha do link. Valores possíveis: 0 - *(padrão)* linha; 2 - linha em negrito; 3 - linha pontilhada; 4 - linha tracejada.
color	string	Cor da linha como um código de cor hexadecimal. Padrão: 000000.
label	string	Rótulo do link.
show_label	integer	Como exibir o rótulo do link. Valores possíveis: -1 - *(padrão)* padrão do mapa; 0 - sempre exibir; 1 - ocultar automaticamente.
indicator_type	integer	Seleciona o tipo de indicador do link. Valores possíveis: 0 - *(padrão)* link estático; 1 - trigger; 2 - valor do item.
linktriggers	array	Triggers do link do mapa a serem usadas como indicadores de status do link. O objeto de trigger de link de mapa é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* se `indicator_type` estiver definido como ”trigger” (1).
itemid	ID	ID do item. Comportamento da propriedade: - *obrigatório* se `indicator_type` estiver definido como ”valor do item” (2).
highlights	array	Destaques do link do mapa a serem usados como indicadores de status do link. O objeto de indicadores de link de mapa é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* se `indicator_type` estiver definido como ”valor do item” (2).

Propriedade	Tipo	Descrição
thresholds	array	Limiares do link do mapa a serem usados como indicadores de status do link. O objeto de indicadores de link de mapa é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* se `indicator_type` estiver definido como "valor do item" (2).
permission	integer	Tipo de nível de permissão. Valores possíveis: -1 - nenhum; 2 - somente leitura; 3 - leitura e gravação.

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
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.

Indicadores de links de mapa

O objeto de indicadores de links de mapa define um indicador de status de link de mapa com base no valor do item. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
pattern	string	Expressão regular para comparar. Disponível apenas para destaques.
sortorder	integer	Usado para definir a ordem de classificação dos destaques. Disponível apenas para destaques.
threshold	string	Valor numérico para comparar. Disponível apenas para limites.
drawtype	integer	Estilo de desenho do indicador. Valores possíveis: 0 - *(padrão)* linha; 2 - linha em negrito; 3 - linha pontilhada; 4 - linha tracejada.
color	string	Cor do indicador como um código de cor hexadecimal. Padrão: DD0000. Comportamento da propriedade: - *obrigatório*

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*
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*

Propriedade	Tipo	Descrição
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.
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.

Propriedade	Tipo	Descrição
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 - inferior. 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.
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.

Propriedade	Tipo	Descrição
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 hosts

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 trigger

Crie um mapa com um elemento de trigger, que contém dois triggers.

Requisição:

```
{
    "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() em *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() em *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.

Parâmetro	Tipo	Descrição
expandUrls	flag	Adiciona URLs globais do mapa aos elementos de mapa correspondentes 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 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	Ordena 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi 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",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "0",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": []
                },
                {
```

```
                "selementid": "12",
                "sysmapid": "3",
                "elementtype": "0",
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
```

```
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
        "grid_show": "1",
        "grid_align": "1",
        "label_format": "0",
        "label_type_host": "2",
        "label_type_hostgroup": "2",
        "label_type_trigger": "2",
        "label_type_map": "2",
        "label_type_image": "2",
        "label_string_host": "",
        "label_string_hostgroup": "",
        "label_string_trigger": "",
        "label_string_map": "",
        "label_string_image": "",
        "iconmapid": "0",
        "expand_macros": "0",
```

```
            "severity_min": "0",
            "userid": "1",
            "private": "1",
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```

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

Origem

CMap::get() em *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 usuário do mapa para substituir os elementos existentes.
userGroups	array	Compartilhamentos de grupo 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 proprietário do mapa

Disponível apenas para administradores e super administradores.

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() em *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 valor

Os seguintes objetos estão diretamente relacionados à API `valuemap`.

Mapa de valor

O objeto mapa de valor possui as seguintes propriedades.

Propriedade	Tipo	Descrição
valuemapid	ID	ID do mapa de valor. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
hostid	ID	ID do host ou template ao qual o mapa de valor pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
name	string	Nome do mapa de valor. Comportamento da propriedade: - *obrigatório* para operações de criação
mappings	array	Mapeamentos de valor para o mapa de valor 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 valor importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o mapa de valor 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 (intervalos são inclusivos; múltiplos 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 de 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
}
```

Origem

CValueMap::create() em *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 mapas de valores a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de valores excluídos na propriedade `valuemapids`.

Exemplos

Excluindo múltiplos 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
}
```

Origem

CValueMap::delete() em *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	Retorna apenas mapas de valores com os IDs fornecidos.
selectMappings	query	Retorna os mapeamentos de valores para o mapa de valores atual na propriedade `mappings`. Suporta `count`.
sortfield	string/array	Ordena 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` tiver sido usado.

Exemplos

Recuperando mapas de valores

Recupera 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
}
```

Recupera 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() em *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. Veja 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

(object) Retorna um objeto contendo os IDs dos mapas de valores atualizados na 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
}
```

Origem

CValueMap::update() em *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 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. Consulte 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 na propriedade `iconmapids`. A ordem dos IDs retornados corresponde à 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 de 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 na 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
}
```

Origem

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. Consulte 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	Retorna apenas mapas de ícones com os IDs fornecidos.
sysmapids	ID/array	Retorna apenas mapas de ícones que são usados nos mapas fornecidos.
selectMappings	query	Retorna uma propriedade `mappings` com os mapeamentos de ícones usados.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `iconmapid`, `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

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. Consulte 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 na 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

Origem

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 dos códigos TOTP. Valores possíveis: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512.
code_length	integer	Comprimento do código de verificação. Comportamento da propriedade: - *obrigatório* se `type` for definido como "TOTP" Valores possíveis: 6 - 6 dígitos; 8 - 8 dígitos.
api_hostname	string	Hostname da API fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` for definido como "TOTP"
clientid	string	ID do cliente fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` for 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` for 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 de 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 MFA

Crie um método 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 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
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 pelo nome

Recupere todos os métodos MFA com ”Zabbix” em seu 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 Métodos MFA)`

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. Apenas 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 integrados: 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 integrados; `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 de 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 módulos com os IDs fornecidos.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `moduleid`, `relative_path`.
countOutput	boolean	Esses 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 de 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

Origem

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 objeto:

- 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*. Consulte 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi 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 nodes 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 nodes 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 é projetada para trabalhar com problemas.

Referências de objeto:

- 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:**
> Os 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.

Propriedade	Tipo	Descrição
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.
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 causador.
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 - média; 4 - alta; 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 tipo de mídia ativas.

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.

Propriedade	Tipo	Descrição
url	string	Valor da URL definida pelo tipo de mídia.

Os resultados conterão entradas apenas para tipos de mídia ativos com a 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 que é considerado "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.

Parâmetro	Tipo	Descrição
acknowledged	boolean	`true` - retorna apenas problemas reconhecidos; `false` - apenas não reconhecidos.
action	integer	Retorna apenas problemas para os quais as ações de atualização de evento fornecidas foram realizadas. Para múltiplas 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 problemas com os IDs dos usuários que realizaram as ações de atualização de evento de problema.
suppressed	boolean	`true` - retorna apenas problemas suprimidos; `false` - retorna problemas em estado normal.
symptom	boolean	`true` - retorna apenas eventos de problema do tipo sintoma; `false` - retorna apenas eventos de problema do tipo causa.
severities	integer/array	Retorna apenas problemas com as severidades de evento fornecidas. Aplica-se apenas se o objeto for trigger.
evaltype	integer	Regras para busca por tag. Valores possíveis: 0 - (padrão) And/Or; 2 - Or.
tags	array	Retorna apenas problemas com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os problemas. Tipos de operador possíveis: 0 - (padrão) Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 4 - Existe; 5 - Não existe.
recent	boolean	`true` - retorna problemas em PROBLEMA e problemas recentemente RESOLVIDOS (depende de Exibir triggers OK por N segundos) Padrão: `false` - apenas problemas NÃO RESOLVIDOS
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.

Parâmetro	Tipo	Descrição
selectAcknowledges	query	Retorna uma propriedade `acknowledges` com as atualizações do problema. As atualizações do problema são classificadas em ordem cronológica reversa. 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)` severidade do evento antes desta ação de atualização; `new_severity` - `(integer)` severidade 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 está suprimido.
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.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `eventid`.
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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperando eventos de problemas do 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 especificado

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() em *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 - criar novos protótipos de gráficos
- graphprototype.delete - excluir protótipos de gráficos
- graphprototype.get - recuperar protótipos de gráficos
- graphprototype.update - atualizar 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
flags	integer	Origem do protótipo de gráfico. Valores possíveis: 2 - um protótipo de gráfico; 6 - um protótipo de gráfico descoberto Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
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 mostrar gráficos de pizza e pizza explodida descobertos em 3D. Valores possíveis: 0 - *(padrão)* mostrar em 2D; 1 - mostrar em 3D.
show_legend	integer	Se deve mostrar a legenda no gráfico descoberto. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
show_work_period	integer	Se deve mostrar o período de trabalho no gráfico descoberto. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
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.
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.

Propriedade	Tipo	Descrição
uuid	string	Identificador único universal, usado para vincular protótipos de gráficos importados a 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. Consulte Funções de usuário para obter 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 dos protótipos de gráficos criados na propriedade `graphids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de gráficos enviados.

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() em *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 gráficos a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de gráficos excluídos na 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
}
```

Origem

CGraphPrototype::delete() em *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.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta do protótipo de gráfico. O objeto de descoberta do protótipo de gráfico vincula um protótipo de gráfico descoberto a um protótipo de gráfico do qual foi descoberto. Possui as seguintes propriedades: `parent_graphid` - `(ID)` ID do protótipo de gráfico a partir do qual o gráfico foi criado; `status` - `(int)` status da descoberta do gráfico: 0 - *(padrão)* protótipo de gráfico é descoberto, 1 - protótipo de gráfico não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um protótipo de gráfico que não é mais descoberto será excluído.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD à qual o protótipo de gráfico pertence.
selectDiscoveryRulePrototype	query	Retorna uma propriedade `discoveryRulePrototype` com o protótipo de regra LLD pai ao qual o protótipo de gráfico pertence.
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.

Parâmetro	Tipo	Descrição
selectHosts	query	Retorna uma propriedade `hosts` com os hosts aos quais o protótipo de gráfico pertence.
selectItems	query	Retorna uma propriedade `items` com os itens 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`. 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	

Valores de retorno

`(integer/array)` Retorna:

- um array 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",
            "flags": "2"
            "discover": "0"
        }
    ],
    "id": 1
}
```

Veja também

- Regra de descoberta
- Item de gráfico
- Item
- Host
- Grupo de hosts
- Template
- Grupo de templates

Fonte

CGraphPrototype::get() em *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 de 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 dos protótipos de gráficos atualizados na 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.
flags	integer	Origem do protótipo de host. Valores possíveis: 2 - um protótipo de host; 6 - um protótipo de 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.
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)* herdar interfaces do host pai; 1 - usar 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 aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de host pertencer a um template

Link de grupo

O objeto de link 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 algum 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 solicitações SNMP bulk. Valores possíveis: 0 - não usar solicitações bulk; 1 - *(padrão)* - usar solicitações 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 solicitações 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 do usuário. Consulte 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 dos protótipos de host criados na 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

Origem

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. Consulte 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 dos protótipos de host excluídos na 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
}
```

Origem

CHostPrototype::delete() em *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

`(objeto)` 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 items herdados de um template.

Parâmetro	Tipo	Descrição
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta do protótipo de host. O objeto de descoberta do protótipo de host vincula um protótipo de host descoberto a um protótipo de host do qual ele foi descoberto. Possui as seguintes propriedades: `host` - `(string)` ID do host; `parent_hostid` - `(string)` ID do protótipo de host a partir do qual o protótipo de host foi criado; `status` - `(int)` status da descoberta do protótipo de host: 0 - *(padrão)* protótipo de host foi descoberto, 1 - protótipo de host não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um protótipo de host que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um protótipo de host que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador se o protótipo de host foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD à qual o protótipo de host pertence.
selectDiscoveryRulePrototype	query	Retorna uma propriedade `discoveryRulePrototype` com o protótipo de regra LLD pai ao qual o protótipo de host pertence.
selectInterfaces	query	Retorna uma propriedade `interfaces` com as interfaces personalizadas do protótipo de host.
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	Ordena 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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",
            "flags": "2",
            "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() em *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. Veja 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.

Parâmetro	Tipo	Descrição
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 customizadas do protótipo de host para substituir as interfaces atuais. O objeto de interface customizada 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 customizadas de 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 dos protótipos de host atualizados na 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

Propriedade	Tipo	Descrição
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. 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
name	string	Nome do protótipo de item. Suporta macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
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}, {HOST.PORT}, {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. 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.
flags	integer	Origem do protótipo de item. Valores possíveis: 2 - um protótipo de item; 6 - um protótipo de 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
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
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

Propriedade	Tipo	Descrição
jmx_endpoint	string	String de conexão personalizada do Java gateway. 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
master_itemid	ID	ID do item mestre. 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` 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”)

Propriedade	Tipo	Descrição
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
privatekey	string	Nome do arquivo da 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 da 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 realizar 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

Propriedade	Tipo	Descrição
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo da 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 da 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
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 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 e macros LLD. 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
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
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` 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”)
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 uma 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

Propriedade	Tipo	Descrição
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 - 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), ”Alteração simples” (9), ”Alteração 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	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	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 se 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 label>="<valor do label>", ...} == <valor>`. Cada componente do padrão Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser macro de usuário ou macro LLD.
[8] saída Prometheus seguindo a sintaxe: `<nome do label>` (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 múltiplos 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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Protótipo 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 dos protótipos de item criados na propriedade `itemids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de item 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": "Espaço livre em disco em {#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 de agente HTTP

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
}
```

Origem

CItemPrototype::create() em *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 item a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de item excluídos na 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 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() em *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	Retorna apenas protótipos de item que pertencem às regras LLD fornecidas.
graphids	ID/array	Retorna apenas protótipos de item que são usados nos protótipos de gráfico fornecidos.
hostids	ID/array	Retorna apenas protótipos de item que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas protótipos de item herdados de um template.
itemids	ID/array	Retorna apenas protótipos de item com os IDs fornecidos.
monitored	boolean	Se definido como `true`, retorna apenas protótipos de item habilitados que pertencem a hosts monitorados.
templated	boolean	Se definido como `true`, retorna apenas protótipos de item que pertencem a templates.
templateids	ID/array	Retorna apenas protótipos de item que pertencem aos templates fornecidos.
triggerids	ID/array	Retorna apenas protótipos de item que são usados nos protótipos de trigger fornecidos.

Parâmetro	Tipo	Descrição
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta do protótipo de item. O objeto de descoberta do protótipo de item vincula um protótipo de item descoberto a um protótipo de item do qual foi descoberto. Possui as seguintes propriedades: `parent_itemid` - `(string)` ID do protótipo de item a partir do qual o protótipo de item foi criado; `key_` - `(string)` chave do protótipo de item; `status` - `(int)` status de descoberta do protótipo de item: 0 - *(padrão)* protótipo de item é descoberto, 1 - protótipo de item não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um protótipo de item que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um protótipo de item que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador se o protótipo de item foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra de descoberta de baixo nível à qual o protótipo de item pertence.
selectDiscoveryRulePrototype	query	Retorna uma propriedade `discoveryRulePrototype` com o protótipo de regra LLD pai ao qual o protótipo de item pertence.
selectGraphs	query	Retorna uma propriedade `graphs` com os protótipos de gráfico em que o protótipo de item é usado. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts aos quais o protótipo de item pertence.
selectTags	query	Retorna as tags do protótipo de item na propriedade `tags`.
selectTriggers	query	Retorna uma propriedade `triggers` com os protótipos de trigger em que o protótipo de item é usado. Suporta `count`.
selectPreprocessing	query	Retorna uma propriedade `preprocessing` com as opções de pré-processamento do protótipo de item.
selectValueMap	query	Retorna uma propriedade `valuemap` com o mapa de valores do protótipo de 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 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	Estes parâmetros são descritos no comentário de referência.

Parâmetro	Tipo	Descrição
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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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": "",
```

```
            "flags": "2",
            "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": "",
            "flags": "2",
            "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": "",
            "flags": "0",
            "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() em *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 de 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 item 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

Atualize 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”.

Requisição:

```
{
    "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() em *ui/include/classes/api/services/CItemPrototype.php*.

Protótipo de regra LLD

Esta classe foi projetada para trabalhar com protótipos de regras de descoberta de baixo nível.

Referências de objetos:

- Protótipo de regra LLD
  - Cabeçalho HTTP
  - Campo de consulta HTTP
- Filtro de protótipo de regra LLD

    - Condição de filtro de protótipo de regra LLD
- Caminho da macro do protótipo LLD
- Pré-processamento do protótipo de regra LLD
- Substituições do protótipo de regra LLD
    - Filtro de substituição do protótipo de regra LLD
        * Condição de filtro de substituição do protótipo de regra LLD
    - Operação de substituição do protótipo de regra LLD
        * Status da operação de substituição do protótipo de regra LLD
        * Descoberta da operação de substituição do protótipo de regra LLD
        * Período da operação de substituição do protótipo de regra LLD
        * Histórico da operação de substituição do protótipo de regra LLD
        * Tendências da operação de substituição do protótipo de regra LLD
        * Gravidade da operação de substituição do protótipo de regra LLD
        * Tag da operação de substituição do protótipo de regra LLD
        * Template da operação de substituição do protótipo de regra LLD
        * Inventário da operação de substituição do protótipo de regra LLD

Métodos disponíveis:

- discoveryruleprototype.create - criar novos protótipos de regras LLD
- discoveryruleprototype.delete - excluir protótipos de regras LLD
- discoveryruleprototype.get - recuperar protótipos de regras LLD
- discoveryruleprototype.update - atualizar protótipos de regras LLD

Objeto de protótipo de regra de LLD

Os seguintes objetos estão diretamente relacionados à API `discoveryruleprototype`.

Protótipo de regra LLD

O objeto protótipo de regra de descoberta de baixo nível possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do protótipo de regra LLD. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
ruleid	ID	ID da regra LLD/rule LLD protótipo pai. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
delay	string	Intervalo de atualização do protótipo de regra 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. 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 o protótipo de regra LLD pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
flags	integer	Origem do protótipo de regra LLD. Valores possíveis: 3 - um protótipo de regra de descoberta de baixo nível; 7 - um protótipo de regra de descoberta de baixo nível descoberto. Comportamento da propriedade: - *somente leitura*
interfaceid	ID	ID da interface do host do protótipo de regra LLD. Comportamento da propriedade: - *obrigatório* se o protótipo de regra LLD pertencer ao host e `type` for definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent” ou ”SNMP agent” - *suportado* se o protótipo de regra LLD 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 regra LLD. Pelo menos uma macro LLD é obrigatória. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
name	string	Nome do protótipo de regra LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
type	integer	Tipo do protótipo de regra 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; 23 - Nested. 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}, {HOST.PORT}, {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. 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 regra LLD.
error	string	Texto de erro se houver problemas ao atualizar o valor do protótipo de regra 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 solicitação 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 Java gateway. 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 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 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. 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 do protótipo de regra LLD: - script executado para regras LLD SSH e Telnet; - consulta SQL para regras LLD de monitoramento de banco de dados; - fórmula para regras LLD calculadas; - o script para regras 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` estiver 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 post armazenado 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 solicitação 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 solicitação HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo de método de solicitação. 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	OID SNMP. 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
state	integer	Estado do protótipo de regra LLD. Valores possíveis: 0 - *(padrão)* normal; 1 - não suportado. Comportamento da propriedade: - *somente leitura*
status	integer	Status do protótipo de regra LLD. Valores possíveis: 0 - *(padrão)* protótipo de regra LLD habilitado; 1 - protótipo de regra LLD 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
templateid	ID	ID do protótipo de regra LLD do template pai. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da solicitação 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 possível: 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 ”Permitir aceitar dados recebidos”
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”)

Propriedade	Tipo	Descrição
uuid	string	Identificador único universal, usado para vincular protótipos de regra LLD importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de regra LLD pertencer a um template
verify_host	integer	Se deve validar se 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 se 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 são usados para especificar um parâmetro de URL. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Filtro de protótipo de regra LLD

O objeto de filtro de protótipo 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 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 de filtro de protótipo de regra LLD

O objeto de condição de filtro de protótipo 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*

Propriedade	Tipo	Descrição
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 protótipo 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.

Note:
Para entender melhor como usar filtros com vários tipos de expressões, veja exemplos nas páginas dos métodos discoveryruleprototype.get e discoveryruleprototype.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 protótipo de regra LLD

O objeto de pré-processamento de protótipo 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 múltiplos registros "Nome do campo,Prefixo OID,Formato" delimitados por um caractere de nova linha.

Substituições de protótipos de regras LLD

O objeto de substituições de protótipos de regras LLD define um conjunto de regras (filtros, condições e operações) que são usadas para substituir propriedades de diferentes objetos de protótipos. Ele 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 de protótipo de regra LLD

O objeto de filtro de substituição de protótipo de 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*
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 fazem referência a condições específicas do filtro de substituição 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 dos filtros de substituição com uma expressão personalizada. A expressão deve conter IDs que fazem referência a condições específicas do filtro de substituição por 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 sobrescrição de protótipo de regra LLD

O objeto de condição de filtro de sobrescrição de protótipo 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 único 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 sobrescrição de protótipo de regra LLD estiver definido como ”expressão personalizada”

Propriedade	Tipo	Descrição
operator	integer	Operador da condição. Valores possíveis: 8 - *(padrão)* corresponde à expressão regular; 9 - não corresponde à expressão regular; 12 - existe; 13 - não existe.

Operação de sobrescrição de protótipo de regra LLD

A operação de sobrescrição de protótipo 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 executar a ação. Valores possíveis: 0 - Item prototype; 1 - Trigger prototype; 2 - Graph prototype; 3 - Host prototype. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição de sobrescrição. 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 item, trigger, gráfico ou protótipo de host, dependendo do objeto selecionado.
opstatus	object	Objeto de status da operação de sobrescrição para objetos de item, trigger e protótipo de host.
opdiscover	object	Objeto de status de descoberta da operação de sobrescrição (todos os tipos de objeto).
opperiod	object	Objeto de período da operação de sobrescrição (intervalo de atualização) para objeto de protótipo de item.
ophistory	object	Objeto de histórico da operação de sobrescrição para objeto de protótipo de item.
optrends	object	Objeto de tendências da operação de sobrescrição para objeto de protótipo de item.
opseverity	object	Objeto de severidade da operação de sobrescrição para objeto de protótipo de trigger.
optag	object/array	Objeto de tag da operação de sobrescrição para objetos de protótipo de trigger e host.
optemplate	object/array	Objeto de template da operação de sobrescrição para objeto de protótipo de host.
opinventory	object	Objeto de inventário da operação de sobrescrição para objeto de protótipo de host.

Status da operação de sobrescrição do protótipo da regra LLD

Status da operação de sobrescrição do protótipo da regra LLD que é definido para o objeto descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
status	integer	Sobrescreve o status para o objeto selecionado. Valores possíveis: 0 - Criar habilitado; 1 - Criar desabilitado. Comportamento da propriedade: - *obrigatório*

Operação de descoberta de protótipo de regra LLD de substituição

Status de descoberta da operação de substituição de protótipo 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 do protótipo de regra LLD

O período de operação de substituição do protótipo de 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 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*

Histórico de operação de sobrescrição de protótipo de regra LLD

Valor do histórico de operação de sobrescrição de protótipo de regra LLD que é definido para o item descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
history	string	Sobrescreve 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 protótipo de regra LLD

Valor de tendências de operação de substituição de protótipo 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*

Severidade da operação de substituição do protótipo da regra LLD

Valor de severidade da operação de substituição do protótipo da regra LLD que é definido para o trigger descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
severity	integer	Substitui a severidade 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 protótipo de regra LLD

O objeto de tag de operação de substituição de protótipo de regra 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 substituição de protótipo de regra LLD

Objeto de template de operação de substituição de protótipo de regra LLD que está vinculado ao host descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
templateid	ID	Substitua o template dos templates vinculados ao protótipo de host. Comportamento da propriedade: - *obrigatório*

Inventário da operação de substituição do protótipo da regra LLD

O valor do modo de inventário da operação de substituição do protótipo 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*

discoveryruleprototype.create

Descrição

`object discoveryruleprototype.create(object/array lldRules)`

Este método permite criar novas regras de descoberta de baixo nível (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)` Protótipos de regras de LLD a serem criados.

Além das propriedades padrão do protótipo de regra de LLD, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Filtro do protótipo de regra de LLD para a regra de LLD.
preprocessing	object/array	Opções de pré-processamento do protótipo de regra de LLD.
lld_macro_paths	object/array	Opções de lld_macro_path do protótipo de regra de LLD.
overrides	object/array	Opções de substituições do protótipo de regra de LLD.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das protótipos de regras LLD criadas na propriedade `itemids`. A ordem dos IDs retornados corresponde à ordem das protótipos de regras LLD passadas.

Exemplos

Criando um protótipo de regra LLD

Crie um protótipo de regra LLD (Tipo: Aninhado) para descobrir tablespaces em uma instância de banco de dados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
```

```
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

Veja também

- Filtro de protótipo de regra de LLD
- Caminhos de macro de LLD
- Pré-processamento de protótipo de regra de LLD

Fonte

CDiscoveryRulePrototype::create() em *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Descrição

`object discoveryruleprototype.delete(array lldRuleIds)`

Este método permite excluir protótipos de regras 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

`(array)` IDs das regras de descoberta a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das regras LLD excluídas na propriedade `ruleids`.

Exemplos

Excluindo múltiplos protótipos de regras LLD

Exclua dois protótipos de regras LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

Origem

CDiscoveryRulePrototype::delete() em *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Descrição

`integer/array discoveryruleprototype.get(object parameters)`

O método permite recuperar protótipos de 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. Consulte 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 regras LLD que pertencem às regras LLD ou protótipos de regras LLD fornecidos.
itemids	ID/array	Retorna apenas protótipos de regras LLD com os IDs fornecidos.
groupids	ID/array	Retorna apenas protótipos de regras LLD que pertencem aos hosts dos grupos fornecidos.
hostids	ID/array	Retorna apenas protótipos de regras LLD que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas protótipos de regras LLD herdados de um template.
interfaceids	ID/array	Retorna apenas protótipos de regras LLD que usam as interfaces de host fornecidas.
monitored	boolean	Se definido como `true`, retorna apenas protótipos de regras LLD habilitados que pertencem a hosts monitorados.
templated	boolean	Se definido como `true`, retorna apenas protótipos de regras LLD que pertencem a templates.
templateids	ID/array	Retorna apenas protótipos de regras LLD que pertencem aos templates fornecidos.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD pai do protótipo de regra LLD. Retornará uma resposta vazia se o pai for um protótipo de regra de descoberta.
selectDiscoveryRulePrototype	query	Retorna uma propriedade `discoveryRulePrototype` com o protótipo de regra LLD pai do protótipo de regra LLD. Retornará uma resposta vazia se o pai for uma regra de descoberta (não protótipo).
selectDiscoveryRulePrototypes	query	Retorna uma propriedade `discoveryRulePrototypes` com protótipos de regras LLD filhas que pertencem ao protótipo de regra LLD. Suporta `count`.

Parâmetro	Tipo	Descrição
selectFilter	query	Retorna uma propriedade `filter` com dados do filtro usado pelo protótipo de regra LLD.
selectGraphs	query	Retorna uma propriedade `graphs` com protótipos de gráficos que pertencem ao protótipo de regra LLD. Suporta `count`.
selectHostPrototypes	query	Retorna uma propriedade `hostPrototypes` com protótipos de host que pertencem ao protótipo de regra LLD. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts ao qual o protótipo de regra LLD pertence.
selectItems	query	Retorna uma propriedade `items` com protótipos de item que pertencem ao protótipo de regra LLD. Suporta `count`.
selectTriggers	query	Retorna uma propriedade `triggers` com protótipos de trigger que pertencem ao protótipo de 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 do protótipo de 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 o protótipo de 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:

- um array de objetos;

- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando protótipos de regras de descoberta de um host

Recupere todos os protótipos de regras de descoberta para um ID de host específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Discover tablespaces for {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
```

```
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Veja também

- Protótipo de gráfico
- Host
- Protótipo de item
- Filtro de protótipo de regra LLD
- Protótipo de trigger

Fonte

CDiscoveryRulePrototype::get() em *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Descrição

`object discoveryruleprototype.update(object/array lldRules)`

Este método permite atualizar protótipos de regras LLD existentes.

Observe que a atualização de protótipos já descobertos é limitada.

> **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 protótipo de regra LLD a serem atualizadas.

A propriedade `itemid` deve ser definida para cada protótipo de regra LLD, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do protótipo de regra LLD, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Filtro do protótipo de regra LLD para substituir o filtro existente.
preprocessing	object/array	Opções de pré-processamento do protótipo de 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 do protótipo de 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 do protótipo de 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 de descoberta de baixo nível atualizadas na propriedade `itemids`.

Exemplos

Atualizando opções de pré-processamento de protótipo de regra LLD

Atualize um protótipo de regra LLD com uma regra de pré-processamento JSONPath. Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Origem

CDiscoveryRulePrototype::update() em *ui/include/classes/api/services/CDiscoveryRulePrototype.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 dos proxies Zabbix ativos.

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 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 escrita* - *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 escrita* - *obrigatório* se `tls_connect` estiver definido como ”PSK”, ou `tls_accept` contiver o bit ”PSK”
custom_timeouts	integer	Se deve sobrescrever os timeouts globais de item no nível do proxy. Valores possíveis: 0 - *(padrão)* usar configurações globais; 1 - sobrescrever timeouts.
timeout_zabbix_agent	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: ””. 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 comparada à versão do Zabbix server. Valores possíveis: 0 - Indefinido; 1 - Versão atual (proxy e server possuem a mesma versão principal); 2 - Versão desatualizada (a versão do proxy é mais antiga que a 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 de 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 ”Active proxy” 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() em *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 mais informações.

Parâmetros

`(array)` IDs dos 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
}
```

Origem

CProxy::delete() em *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 de 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

Recuperar 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() em *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 de usuário. Consulte 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 proxies

Atualize o proxy com ID ”5” e adicione-o ao grupo de proxies 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() em *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 - cria novas regras de descoberta de rede
- drule.delete - exclui regras de descoberta de rede
- drule.get - recupera regras de descoberta de rede
- drule.update - atualiza 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. 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 se houver 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 de usuário. Veja 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 criadas na propriedade `druleids`. A ordem dos IDs retornados corresponde 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 de 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

- Verificação de descoberta

Origem

CDRule::create() em *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 de 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 das regras de descoberta excluídas na 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() em *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	Retornar apenas regras de descoberta que criaram os hosts descobertos fornecidos.
druleids	ID/array	Retornar apenas regras de descoberta com os IDs fornecidos.
dserviceids	ID/array	Retornar apenas regras de descoberta que criaram os serviços descobertos fornecidos.
selectDChecks	query	Retornar uma propriedade `dchecks` com as verificações de descoberta usadas pela regra de descoberta. Suporta `count`.
selectDHosts	query	Retornar 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	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

Recuperar todas as regras de descoberta

Recupera todas as regras de descoberta configuradas e as verificações de descoberta que elas utilizam.

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 de usuário. Consulte 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 das regras de descoberta atualizadas na 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

- Verificação de descoberta

Origem

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 da macro de LLD
- Pré-processamento da regra de LLD
- Substituições da regra de LLD
  - Filtro de substituição da regra de LLD
    - Condição de filtro de substituição da regra de LLD
  - Operação de substituição da regra de LLD
    - Status da operação de substituição da regra de LLD
    - Descoberta da operação de substituição da regra de LLD
    - Período da operação de substituição da regra de LLD
    - Histórico da operação de substituição da regra de LLD
    - Tendências da operação de substituição da regra de LLD
    - Gravidade da operação de substituição da regra de LLD
    - Tag da operação de substituição da regra de LLD
    - Template da operação de substituição da regra de LLD
    - Inventário da operação de substituição da regra de LLD

Métodos disponíveis:

- discoveryrule.create - cria novas regras de LLD
- discoveryrule.delete - exclui regras de LLD
- discoveryrule.get - recupera regras de LLD
- discoveryrule.update - atualiza regras de LLD

Objeto de regra de LLD

Os seguintes objetos estão diretamente relacionados à API `discoveryrule`.

Regra LLD

O objeto de regra de descoberta de baixo nível possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID da regra LLD. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
delay	string	Intervalo de atualização da regra 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. Várias 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 LLD pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
flags	integer	Origem da regra de descoberta. Valores possíveis: 1 - uma regra de descoberta de baixo nível; 5 - uma regra de descoberta de baixo nível convertida de protótipo. Comportamento da propriedade: - *somente leitura*
interfaceid	ID	ID da interface do host da regra LLD. Comportamento da propriedade: - *obrigatório* se a regra LLD pertencer ao host e `type` for definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent” ou ”SNMP agent” - *suportado* se a regra 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 LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
name	string	Nome da regra LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
type	integer	Tipo da regra 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; 23 - Nested. Este tipo é permitido apenas se o pai da regra for um template (presume-se que esteja vinculado a um protótipo de host na descoberta) ou host descoberto (que possui uma regra LLD em sua raiz). 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}, {HOST.PORT}, {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. 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”)

Propriedade	Tipo	Descrição
description	string	Descrição da regra LLD.
error	string	Texto de erro se houver problemas ao atualizar o valor da regra LLD. 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
headers	array	Array de cabeçalhos que serão enviados ao executar uma solicitação 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 Java gateway. 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 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 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. 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 LLD: - script executado para regras LLD SSH e Telnet; - consulta SQL para regras LLD de monitoramento de banco de dados; - fórmula para regras LLD calculadas; - o script para regras 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` estiver 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
posts	string	Dados do corpo da solicitação 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 solicitação HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo de método de solicitação. 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	OID SNMP. 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
state	integer	Estado da regra LLD. Valores possíveis: 0 - *(padrão)* normal; 1 - não suportado. Comportamento da propriedade: - *somente leitura*
status	integer	Status da regra LLD. Valores possíveis: 0 - *(padrão)* regra LLD habilitada; 1 - regra 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 LLD do template pai. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da solicitação 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 possível: 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”
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”)

Propriedade	Tipo	Descrição
uuid	string	Identificador único universal, usado para vincular regras LLD importadas às já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se a regra LLD pertencer a um 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*

Propriedade	Tipo	Descrição
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. 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 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 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. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
macro	string	Macro LLD para realizar a verificação. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
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 LLD

O objeto de substituições de regras 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*
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 sobrescrição de regra LLD

O objeto de condição de filtro de sobrescrição 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 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. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto de filtro de sobrescrição de regra LLD estiver definido como ”expressão personalizada”

Propriedade	Tipo	Descrição
operator	integer	Operador da condição. Valores possíveis: 8 - *(padrão)* corresponde à expressão regular; 9 - não corresponde à expressão regular; 12 - existe; 13 - não existe.

Operação de sobrescrição de regra LLD

A operação de sobrescrição 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; 4 - Protótipo de descoberta. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição de sobrescrição. 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 sobrescrição para objetos protótipo de item, trigger e host.
opdiscover	object	Objeto de status de descoberta da operação de sobrescrição (todos os tipos de objeto).
opperiod	object	Objeto de período da operação de sobrescrição (intervalo de atualização) para objeto protótipo de item.
ophistory	object	Objeto de histórico da operação de sobrescrição para objeto protótipo de item.
optrends	object	Objeto de tendências da operação de sobrescrição para objeto protótipo de item.
opseverity	object	Objeto de severidade da operação de sobrescrição para objeto protótipo de trigger.
optag	object/array	Objeto de tag da operação de sobrescrição para objetos protótipo de trigger e host.
optemplate	object/array	Objeto de template da operação de sobrescrição para objeto protótipo de host.
opinventory	object	Objeto de inventário da operação de sobrescrição para objeto protótipo de host.

Status da operação de sobrescrição da regra LLD

Status da operação de sobrescrição da regra LLD que é definida para o objeto descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
status	integer	Sobrescreve o status para o objeto selecionado. Valores possíveis: 0 - Criar habilitado; 1 - Criar desabilitado. Comportamento da propriedade: - *obrigatório*

Operação de descoberta de sobrescrição de regra LLD

Status de descoberta da operação de sobrescrição de regra LLD que é definido para o objeto descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
discover	integer	Sobrescreve 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. Intervalos personalizados podem ser uma mistura de intervalos flexíveis e de agendamento. Aceita macros de usuário ou 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*

Histórico de operação de sobrescrição de regra LLD

Valor do histórico de operação de sobrescrição de regra LLD que é definido para o item descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
history	string	Sobrescreve 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 sobrescrição de regra de LLD

O objeto de tag de operação de sobrescrição de regra de LLD contém o nome e o valor da tag que são definidos para o objeto descoberto. 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. 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.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 de usuário. Veja 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 de 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 ”e”.

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 de LLD com um filtro que usará uma expressão personalizada para avaliar as condições. A regra de 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 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 de 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": "Database host",
                        "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": "Database host",
                        "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 de LLD com período de tempo especificado para desabilitação e sem exclusão

Crie uma regra de LLD com período de tempo personalizado para desabilitar a entidade após ela não ser mais descoberta, com a configuração de que ela 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 LLD
- Caminhos de macro LLD
- Pré-processamento de regra LLD

Origem

CDiscoveryRule::create() em *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Descrição

`object discoveryrule.delete(array lldRuleIds)`

Este método permite excluir regras 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

`(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() em *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Descrição

`integer/array discoveryrule.get(object parameters)`

O método permite recuperar regras de 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. 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 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.
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.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta da regra LLD. O objeto de descoberta da regra LLD vincula uma regra LLD descoberta a um protótipo de regra LLD do qual foi descoberta.
selectDiscoveryRulePrototypes	query	Retorna uma propriedade `discoveryRulePrototypes` com protótipos de regras LLD que pertencem à regra LLD. Suporta `count`.
selectFilter	query	Retorna uma propriedade `filter` com 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.

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. Suporta propriedades adicionais: `host` - nome técnico do host ao qual a regra LLD pertence.
limitSelects	integer	Limita o número de registros retornados por subseleções. Aplica-se às seguintes subseleções: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
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:

- um array 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": "",
        "flags": "1",
        "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",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "state": "0",
        "error": "",
        "parameters": []
    },
    {
        "itemid": "27426",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10202",
        "name": "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": "",
            "flags": "1",
            "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",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


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": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
                    "value": "api"
                },
                {
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
```

```
            "parameters": []
        }
    ],
    "id": 1
}
```

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() em *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 de 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
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() em *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.
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 (armazenada como timestamp do início do dia (00:00:00)).

Propriedade	Tipo	Descrição
active_till	string	Data de término. Valores possíveis: string vazia - *(padrão)* não especificado (armazenado como 0); data específica no formato AAAA-MM-DD (armazenada 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 foi processado; 1 - relatório foi gerado e enviado com sucesso para todos os destinatários; 2 - falha ao gerar o relatório; ”info” contém informações sobre o erro; 3 - relatório foi gerado, mas o envio para alguns (ou todos) destinatários falhou; ”info” contém informações sobre o 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 o usuário será excluído da lista de envio. 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 para criar.

Além das propriedades padrão de relatórios agendados, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
users	object/array	Usuários para enviar o relatório. Comportamento do parâmetro: - *obrigatório* se `user_groups` não estiver definido
user_groups	object/array	Grupos de usuários para enviar o relatório. Comportamento do parâmetro: - *obrigatório* se `users` não estiver definido

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos relatórios agendados criados na propriedade `reportids`. A ordem dos IDs retornados corresponde à ordem dos relatórios agendados 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

Origem

CReport::create() em *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 agendados a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos relatórios agendados 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() em *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; se `false`, apenas relatórios agendados ativos.
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	

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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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() em *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. Veja 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 agendados 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

Origem

CReport::update() em *ui/include/classes/api/services/CReport.php*.

Role

Esta classe foi projetada para trabalhar com funções de usuário.

Referências de objeto:

- Role

- 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`.

Função

O objeto função possui as seguintes propriedades:

Propriedade	Tipo	Descrição
roleid	ID	ID da função. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da função. Comportamento da propriedade: - *obrigatório* para operações de criação
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 função 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 Elemento da interface.
ui.default_access	integer	Se o acesso a novos elementos da interface está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Propriedade	Tipo	Descrição
services.read.mode	integer	Acesso somente leitura aos serviços. Valores possíveis: 0 - Acesso somente leitura aos serviços especificados pela propriedade `services.read.list` ou correspondentes pela propriedade `services.read.tag`; 1 - *(padrão)* Acesso somente leitura a todos os serviços.
services.read.list	array	Array de objetos 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 escrita aos serviços. Comportamento da propriedade: - *suportado* se `services.read.mode` estiver definido como "0"
services.read.tag	object	Array de objetos 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 escrita aos serviços. Comportamento da propriedade: - *suportado* se `services.read.mode` estiver definido como "0"
services.write.mode	integer	Acesso de leitura e escrita aos serviços. Valores possíveis: 0 - *(padrão)* Acesso de leitura e escrita aos serviços especificados pela propriedade `services.write.list` ou correspondentes pela propriedade `services.write.tag`; 1 - Acesso de leitura e escrita a todos os serviços.
services.write.list	array	Array de objetos Serviço. Os serviços especificados, incluindo serviços filhos, terão acesso de leitura e escrita concedido à função de usuário. O acesso de leitura e escrita 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 Tag de serviço. Os serviços correspondentes à tag, incluindo serviços filhos, terão acesso de leitura e escrita concedido à função de usuário. O acesso de leitura e escrita substituirá o acesso somente leitura aos serviços. Comportamento da propriedade: - *suportado* se `services.write.mode` estiver definido como "0"
modules	array	Array de objetos 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.

Propriedade	Tipo	Descrição
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 Ação.
actions.default_access	integer	Se o acesso a novas ações está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Elemento da interface

O objeto de elemento da interface possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do elemento da interface. 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 → Últimos dados*; `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 auto-registro*; `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 está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Service

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; `edit_own_media` - Permitir criar/editar sua própria mídia. 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. Valores possíveis se o `type` do objeto Role estiver definido como "Super admin": `edit_user_media` - Permitir criar/editar mídia para usuários. 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 de usuário. Consulte 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 das funções criadas 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.

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

Origem

CRole::create() em *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. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos papéis a serem excluídos.

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() em *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 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
roleids	ID/array	Retorna apenas funções com os IDs fornecidos.
selectRules	query	Retorna uma propriedade `rules` com as regras da função.
selectUsers	query	Retorna uma propriedade `users` com os usuários aos quais a função está atribuída. Veja `user.get` para restrições baseadas no tipo de usuário.

Parâmetro	Tipo	Descrição
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `roleid`, `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 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": [],
    "modules.default_access": "1",
    "api.access": "1",
    "api.mode": "0",
    "api": [],
    "actions": [
        {
```

```
            "name": "edit_dashboards",
            "status": "1"
        },
        {
            "name": "edit_maps",
            "status": "1"
        },
        {
            "name": "acknowledge_problems",
            "status": "1"
        },
        {
            "name": "suppress_problems",
            "status": "1"
        },
        {
            "name": "close_problems",
            "status": "1"
        },
        {
            "name": "change_severity",
            "status": "1"
        },
        {
            "name": "add_problem_comments",
            "status": "1"
        },
        {
            "name": "execute_scripts",
            "status": "1"
        },
        {
            "name": "manage_api_tokens",
            "status": "1"
        },
        {
            "name": "edit_maintenance",
            "status": "1"
        },
        {
            "name": "manage_scheduled_reports",
            "status": "1"
        },
        {
            "name": "manage_sla",
            "status": "1"
        },
        {
            "name": "invoke_execute_now",
            "status": "1"
        },
        {
            "name": "change_problem_ranking",
            "status": "1"
        },
        {
            "name": "edit_own_media",
            "status": "1"
        },
        {
            "name": "edit_user_media",
            "status": "1"
```

```
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```

Veja também

- Regras de função
- Usuário

Fonte

CRole::get() em *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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da função a serem atualizadas.

A propriedade `roleid` deve ser definida para cada função, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da função, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
rules	array	Regras de acesso a serem atualizadas para a função.

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 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() em *ui/include/classes/api/services/CRole.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
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

Propriedade	Tipo	Descrição
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 em 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”
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.

Propriedade	Tipo	Descrição
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. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”
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”
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”

Propriedade	Tipo	Descrição
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 de entrada possíveis separados por vírgula.
manualinput_default_value	string	Valor padrão para preenchimento automático da entrada do usuário. Comportamento da propriedade: - *suportado* se `manualinput` estiver definido como ”Habilitado”
		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. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Scripts a serem criados.

O método aceita scripts com as propriedades padrão de script.

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

Criar um script de webhook

Crie um script de webhook que envie uma solicitação HTTP para um serviço externo.

Solicitação:

```
{
    "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
        "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": "Script de URL com entrada manual",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Selecione uma página para abrir:",
        "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() em *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 de 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 múltiplos 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() em *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.

Request:

```
{
    "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
}
```


Executar um script personalizado

Executar 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
}
```


Execute um script personalizado com entrada manual

Execute 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
}
```


Origem

CScript::execute() em *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. Consulte Funções de usuário para mais informações.

Parâmetros

`(objeto)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas scripts que podem ser executados nos grupos de hosts fornecidos.
hostids	ID/array	Retorna apenas scripts que podem ser executados nos hosts fornecidos.
scriptids	ID/array	Retorna apenas scripts com os IDs fornecidos.
usrgrpids	ID/array	Retorna apenas scripts que podem ser executados por usuários nos grupos de usuários fornecidos.
selectHostGroups	query	Retorna uma propriedade `hostgroups` com grupos de hosts nos quais o script pode ser executado.
selectHosts	query	Retorna uma propriedade `hosts` com hosts nos quais o script pode ser executado.
selectActions	query	Retorna uma propriedade `actions` com ações às quais o script está associado.
sortfield	string/array	Ordena 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	

Valores de retorno

(integer/array) Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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

Origem

CScript::get() em *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Descrição

`object script.getscriptsbyevents(object parameters)`

Este método permite recuperar todos os scripts disponíveis para o 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 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 de evento como propriedades e arrays de scripts disponíveis como valores. Se o ID do script for fornecido, o valor associado é um array contendo o script específico.

> **Note:**
> O método irá expandir automaticamente 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 evento

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.

Recupere o 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 valor da 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 é um array contendo o script específico.

> **Note:**
> O método irá expandir automaticamente 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
}
```

Origem

CScript::getScriptsByHosts() em *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. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do script a serem atualizadas.

A propriedade `scriptid` deve ser definida para cada script, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas. Uma exceção é a alteração 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 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() em *ui/include/classes/api/services/CScript.php*.

Service

Esta classe foi projetada para trabalhar com serviços de infraestrutura de TI/negócios.

Referências de objetos:

- 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 de serviço

Os seguintes objetos estão diretamente relacionados à API de `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

Propriedade	Tipo	Descrição
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. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do serviço. Comportamento da propriedade: - *obrigatório* para operações de criação
sortorder	integer	Posição do serviço usada para ordenação. Valores possíveis: 0-999. Comportamento da propriedade: - *obrigatório* para operações de criação
weight	integer	Peso do serviço. Valores possíveis: 0-1000000. Padrão: 0.
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` for ”0” ou ”3”: 0 - Não classificado. Valores possíveis se `propagation_rule` for ”1” ou ”2”: 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Valores possíveis se `propagation_rule` for ”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

Propriedade	Tipo	Descrição
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	boolean	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*

Propriedade	Tipo	Descrição
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*
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 alterado 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() em *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
}
```

Origem

CService::delete() em *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	Regras para busca de tags. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
tags	object/array	Retorna apenas os serviços com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 4 - Existe; 5 - Não existe.

Parâmetro	Tipo	Descrição
problem_tags	object/array	Retorna apenas os serviços com as tags de problema fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 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 pais. Suporta `count`.
selectTags	query	Retorna uma propriedade `tags` com as tags do serviço. Suporta `count`.
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 do evento resolvido. 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	

Parâmetro	Tipo	Descrição
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 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": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "My Service - 0001",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "My Service Description 0001.",
            "uuid": "dfa4daeaea754e3a95c04d6029182681",
            "created_at": "946684800",
            "readonly": false,
            "parents": [],
            "children": []
        },
        {
            "serviceid": "2",
            "name": "My Service - 0002",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
```

```
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "My Service Description 0002.",
            "uuid": "20ea0d85212841219130abeaca28c065",
            "created_at": "946684800",
            "readonly": false,
            "parents": [],
            "children": []
        }
    ],
    "id": 1
}
```

Origem

CService::get() em *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() em *ui/include/classes/api/services/CService.php*.

Serviço descoberto

Esta classe foi 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:**
> Os 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

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.

Parâmetro	Tipo	Descrição
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:

- um array 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

Propriedade	Tipo	Descrição
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 (inclusive). 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. Consulte 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 criação de uma entrada de SLA para: * rastrear o tempo de atividade dos serviços relacionados ao SQL-engine; * agendamento personalizado para 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
}
```

Origem

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. Consulte 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 as janelas de indisponibilidade excluídas 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	
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 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
}
```

Origem

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	Data de início (inclusiva) para relatar o SLI. Valores possíveis: timestamp.
period_to	timestamp	Data de término (exclusiva) para relatar o SLI. Valores possíveis: timestamp.
periods	array	Número preferencial de períodos para relatar. 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 em combinações de parâmetros.

> **Note:**
> Os períodos retornados não precederão o primeiro período disponível com base na data efetiva do SLA e não excederão o período atual.

Parâmetros			Descrição
period_from	**period_to**	**periods**	
-	-	-	Retorna os últimos 20 períodos.
-	-	especificado	Retorna os últimos períodos especificados pelo parâmetro **periods**.
-	especificado	-	Retorna os últimos 20 períodos antes da data **period_to** especificada.
-	especificado	especificado	Retorna os últimos períodos especificados pelo parâmetro **periods** antes da data **period_to** especificada.
especificado	-	-	Retorna os primeiros 20 períodos a partir da data **period_from** especificada.
especificado	-	especificado	Retorna os primeiros períodos especificados pelo parâmetro **periods** a partir da data **period_from** especificada.
especificado	especificado	-	Retorna até 100 períodos dentro do intervalo de datas especificado.

Parâmetros			Descrição
especificado	especificado	especificado	Retorna os períodos especificados pelo parâmetro **periods** dentro do intervalo de datas 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` - Data de início do período reportado (timestamp). - `period_to` - Data de término do período reportado (timestamp). Os períodos são classificados pelo campo `period_from` em ordem crescente.
serviceids	array	Lista de IDs de serviços nos períodos reportados. A ordem de classificação da lista não é 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 reportado e serviço. 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 de SLI

Os dados de 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.
excluded_downtimes	array	Array de períodos de inatividade excluídos neste período de relatório. Cada objeto conterá os seguintes parâmetros: - `name` - Nome do período de inatividade excluído. - `period_from` - Data e hora de início (inclusive) do período de inatividade excluído. - `period_to` - Data e hora de término (exclusive) do período de inatividade excluído. Os períodos de inatividade excluídos são classificados pelo campo period_from em ordem crescente.

Exemplos

Calculando SLI

Recupere os dados de SLI dos serviços com IDs ”50”, ”60” e ”70” que estão vinculados ao SLA com ID ”5”. Recupere dados para 3 períodos a partir de 01 de novembro de 2021.

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": "Excluded Downtime - 1",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
```

```
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Excluded Downtime - 1",
                        "period_from": 1637836212,
                        "period_to": 1638316800
                    }
                ]
            },
            {
                "uptime": 1186212,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Excluded Downtime - 1",
                        "period_from": 1637836212,
                        "period_to": 1638316800
                    }
                ]
            }
        ],
        [
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Excluded Downtime - 1",
                        "period_from": 1638439200,
                        "period_to": 1639109652
                    }
                ]
            },
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Excluded Downtime - 1",
                        "period_from": 1638439200,
                        "period_to": 1639109652
                    }
                ]
            },
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Excluded Downtime - 1",
                        "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
}
```


Origem

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 ao 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
}
```

Origem

CSla::update() em *ui/include/classes/api/services/CSla.php*.

Task

Esta classe foi projetada para trabalhar com tarefas (como checagem de items ou regras de descoberta de baixo nível sem recarregar a configuração).

Referências de objetos:

- Task
  - Objeto de requisição ’Executar agora’
  - Objeto de requisição ’Atualizar configuração do proxy’
  - Objeto de requisição ’Informação de diagnóstico’
    - Objeto de requisição de estatística
    - Objeto de resultado de estatística

Métodos disponíveis:

- task.create - cria novas tarefas
- task.get - recupera 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 - Informações de diagnóstico; 2 - Atualizar configuração do proxy; 6 - Executar agora. Usuários do tipo *Admin* e *User* podem criar apenas tasks do tipo 'Executar agora'. Note que `task.get` sempre retorna "7" (Resumo da execução da task). Comportamento da propriedade: - *obrigatório*
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 as estatísticas de informações de diagnóstico são coletadas. Comportamento da propriedade: - *suportado* se `type` estiver definido como "Informações de diagnóstico" ou "Atualizar configuração do proxy"

Propriedade	Tipo	Descrição
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 'Informações de diagnóstico' é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório*
result	object	Objeto de resultado da task de informações 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 regras de descoberta de baixo nível. Usuários do tipo *Admin* e *User* podem 'Executar agora' itens em hosts para os quais possuem permissão de *leitura-escrita* permissão, ou permissão de *leitura* e a ação `invoke_execute_now` ação 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 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 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 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 classificar 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 - o 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 como string 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. 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 de 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 'Execute agora'
- Objeto de solicitação 'Atualizar configuração do proxy'
- Objeto de solicitação 'Informações de diagnóstico'

Fonte

CTask::create() em *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 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
taskids	ID/array	Retorna 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

Recupera 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() em *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
description	texto	Descrição do template.
host	string	Nome técnico do template. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
name	string	Nome visível do template. Padrão: valor da propriedade `host`.
readme	texto	Instruções de configuração específicas do template para exibir no Assistente de Host. Suporta formatação Markdown.
uuid	string	Identificador universal único, usado para vincular templates importados aos já existentes. Gerado automaticamente, se não 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.
wizard_ready	integer	Se o template está disponível para seleção no Assistente de Host. Valores possíveis: 0 - *(padrão)* Não disponível. 1 - Disponível.

Tag de template

O objeto de tag de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de template. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag de 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 de usuário. Consulte 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 templates 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() em *ui/include/classes/api/services/CTemplate.php*.

template.delete

Descrição

`object template.delete(array templateIds)`

Este método permite excluir templates.

A exclusão de 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 próprio template, 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 excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos templates excluídos 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
}
```

Origem

CTemplate::delete() em *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	Regras para busca por tags. Valores possíveis: 0 - (padrão) E/Ou; 2 - Ou.
tags	object/array	Retorna apenas templates com as tags fornecidas. Correspondência exata por tag e busca sensível ou não a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os templates. Valores possíveis para o operador: 0 - (padrão) Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 4 - Existe; 5 - Não existe.
selectTags	query	Retorna as tags do template na propriedade `tags`.
selectDiscoveryRules	query	Retorna uma propriedade `discoveryRules` com regras LLD do template. Suporta `count`.
selectHosts	query	Retorna os hosts que estão vinculados ao template na propriedade `hosts`. Suporta `count`.
selectTags	query	Retorna as tags do template na propriedade `tags`.
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`.

Parâmetro	Tipo	Descrição
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`.
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`; `selectDiscoveryRules` - 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	
selectDiscoveries	query	Retorna descobertas de baixo nível do template na propriedade `discoveries`. Suporta `count`. Esta consulta está **obsoleta**, por favor use `selectDiscoveryRules` em seu lugar.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` 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 by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": "Este é um template oficial do Linux. Ele requer o Zabbix agent 8.0 ou mais rec
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Visão geral\r\n\r\nEste é um template oficial do Linux. Ele requer o Zabbix agen
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
```

```
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": "Este é um template oficial do Windows. Ele requer o Zabbix agent 8.0 ou mais r
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Visão geral\r\n\r\nEste é um template oficial do Windows. Ele requer o Zabbix ag
        }
    ],
    "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
}
```

Pesquisa por tags de template

Recupere templates que tenham 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() em *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 a serem vinculados aos 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

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() em *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 do 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*
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 templates atualizados na propriedade `templateids`.

Exemplos

Removendo templates de um grupo

Remover 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

Origem

CTemplate::massRemove() em *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 múltiplos 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 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 templates para substituir os grupos de templates atuais aos quais os templates pertencem. Os grupos de templates 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 templates 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() em *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 são passados são apenas desvinculados. Os templates devem ter apenas a propriedade `templateid` definida.
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.

Requisição:

```
{
    "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 templates vinculados ao template

Desvincule (sem limpar) todos os templates do template fornecido e vincule 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
}
```

Origem

CTemplate::update() em *ui/include/classes/api/services/CTemplate.php*.

Template dashboard

Esta classe foi projetada para trabalhar com dashboards de template.

Referências de objeto:

- 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 slideshow automaticamente. Valores possíveis: 0 - não iniciar slideshow automaticamente; 1 - *(padrão)* iniciar slideshow automaticamente.
uuid	string	Identificador universal único, usado para vincular dashboards de template importados a dashboards já existentes. Gerado automaticamente, se não fornecido.

Página do dashboard do template

O objeto da 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 (será usado o período de exibição padrão da página).
widgets	array	Array de objetos widget do dashboard do template.

Widget de dashboard de template

O objeto de 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; discovery - Status de descoberta; favgraphs - Gráficos favoritos; favmaps - Mapas favoritos; gauge - Medidor; graph - Gráfico (clássico); graphprototype - Protótipo de gráfico; honeycomb - Colmeia; hostavail - Disponibilidade do host; hostcard - Cartão do host; hostnavigator - Navegador de host; itemcard - Cartão de item; 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; topitems - Principais itens; toptriggers - Principais triggers; trigover - Visão geral de 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.
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 a host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade de 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*
value	mixed	Valor do campo do widget dependendo do tipo. Valores possíveis: veja Campos de widget de dashboard. Observe que alguns parâmetros relacionados a host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade de 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 de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Dashboards de template a serem criados.

Além das propriedades padrão de dashboard de template, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas de 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 dashboards de template criados na propriedade `dashboardids`. A ordem dos IDs retornados corresponde à ordem dos dashboards de template passados.

Exemplos

Criando um dashboard de template

Crie um dashboard de template chamado “Graphs” com um widget de Gráfico em uma única página de 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() em *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 do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos painéis de template a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos dashboards de template excluídos na propriedade `dashboardids`.

Exemplos

Excluindo múltiplos dashboards de template

Exclua dois dashboards de template.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Origem

CTemplateDashboard::delete() em *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

`(objeto)` 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	
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 dashboards de template

Recupere 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 de dashboard de template
- Widget de dashboard de template
- Campo de widget de dashboard de template

Fonte

CTemplateDashboard::get() em *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 de usuário. Consulte Funções de usuário para obter 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 necessário para a propriedade `pages`.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos dashboards de template 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 do dashboard do 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

Fonte

CTemplateDashboard::update() em *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 - cria novos tipos de mídia
- mediatype.delete - exclui tipos de mídia
- mediatype.get - recupera tipos de mídia
- mediatype.update - atualiza 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
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 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”

Propriedade	Tipo	Descrição
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 utilizado. 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”
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 utilizado. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - Senha normal; 2 - Token OAuth. A autenticação OAuth não é permitida para o provedor de e-mail *Office365 relay*. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
redirection_url	string	URL do frontend do Zabbix para redirecionar de volta a autorização OAuth. Padrão: Valor da propriedade de configurações da API `url` com a parte `zabbix.php?action=oauth.authorize` Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”
client_id	string	O identificador do cliente registrado no servidor de autorização OAuth. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”

Propriedade	Tipo	Descrição
client_secret	string	O segredo do cliente registrado no servidor de autorização OAuth. Acessível apenas para usuário do tipo Super Admin. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”
authorization_url	string	URL OAuth, com parâmetros, para obter tokens de acesso e atualização. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”
token_url	string	URL OAuth para trocar o token de autorização por tokens de acesso e atualização. Esta URL também é usada pelo server para atualizar o token de acesso inválido. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”
tokens_status	integer	Máscara de bits no status dos tokens. Valores possíveis: 0 - (padrão) Ambos os tokens contêm valor inválido 1 - O token de acesso contém valor válido 2 - O token de atualização contém valor válido 3 - Ambos os tokens contêm valor válido. Comportamento da propriedade: - *suportado* se `smtp_authentication` estiver definido como ”Token OAuth”
access_token	string	Valor do token de acesso OAuth. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”
access_token_updated	timestamp	Timestamp da última modificação do `access_token` feita pelo server ao atualizar com `refresh_token` ou pela API em alterações de token. Comportamento da propriedade: - *suportado* se `smtp_authentication` estiver definido como ”Token OAuth”
access_expires_in	integer	Tempo em segundos quando o `access_token` se tornará desatualizado e exigirá uma solicitação para `refresh_url`. É definido pelo Zabbix server na atualização do `access_token` ou pela API em alterações de token. O timestamp é calculado somando o valor de `access_token_updated`. Comportamento da propriedade: - *suportado* se `smtp_authentication` estiver definido como ”Token OAuth”
refresh_token	string	Valor do token de atualização OAuth. Comportamento da propriedade: - *obrigatório* se `smtp_authentication` estiver definido como ”Token OAuth”

Propriedade	Tipo	Descrição
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 de envio de 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.
message_format	integer	Formato da mensagem. Valores possíveis: 0 - Texto simples; 1 - *(padrão)* HTML.
script	text	Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail” Corpo do script do webhook (JavaScript).
timeout	string	Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Webhook” Timeout do script do webhook. Aceita segundos e unidade de tempo com sufixo. Valores possíveis: 1-60s. Padrão: 30s.
process_tags	integer	Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook” Processar valores de propriedade 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”

Propriedade	Tipo	Descrição
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`.
event_menu_url	string	Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook” URL usada na entrada do menu de evento e na propriedade `urls` retornada por problem.get e event.get.
event_menu_name	string	Comportamento da propriedade: - *suportado* se `show_event_menu` estiver definido como ”Incluir entrada de menu de evento e propriedade `urls`” Nome usado para a entrada do menu de evento e na propriedade `urls` retornada por problem.get e event.get.
parameters	array	Comportamento da propriedade: - *suportado* se `show_event_menu` estiver definido como ”Incluir entrada de menu de evento e propriedade `urls`” Parâmetros do webhook ou parâmetros do script.
description	text	Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook” ou ”Script” 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 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 de 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
```

```
}
```

Origem

CMediaType::create() em *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. Consulte 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() em *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. 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.

Note:
Ao solicitar informações relacionadas ao usuário de tipos de mídia, usuários do tipo *Admin* e *User* podem recuperar dados apenas 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*
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.
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. Possíveis propriedades do objeto Media type para usuários do tipo *Admin* e *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`.
output	query	Propriedades do objeto Media type a serem retornadas. Usuários do tipo *Admin* e *User* podem recuperar apenas as seguintes propriedades: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Para um exemplo, consulte Recuperando tipos de mídia como *Admin*.
search	object	Padrão: `extend`. 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* e *User*: `name`, `description`.

Parâmetro	Tipo	Descrição
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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido 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 email com um usuário (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() em *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
}
```

Origem

CMediaType::update() em *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*

Propriedade	Tipo	Descrição
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

`(object/array)` Tokens a serem criados.

O método aceita tokens com as propriedades padrão do token.

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 irá autenticar 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
}
```

Origem

CToken::create() em *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 múltiplos tokens

Exclua dois tokens.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
```

```
        "188",
        "192"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Origem

CToken::delete() em *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* permissão é necessária para a função de usuário gerenciar tokens para 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
}
```

Origem

CToken::generate() em *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* pode visualizar tokens de outros usuários.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
tokenids	ID/array	Retorna apenas tokens com os IDs fornecidos.
userids	ID/array	Retorna apenas tokens criados para os usuários fornecidos.
token	string	Retorna apenas tokens criados para o *Auth token* fornecido.
valid_at	timestamp	Retorna apenas tokens que são válidos (não expirados) na data e hora fornecidas.
expired_at	timestamp	Retorna apenas tokens que estão expirados (não válidos) na data e hora fornecidas.
sortfield	string/array	Ordena 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar um token

Recupere todos os dados do token com 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
}
```

Origem

CToken::get() em *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* é necessária para a função de usuário gerenciar tokens de outros usuários. Veja permissão.

Parâmetros

`(object/array)` Propriedades do token a serem atualizadas.

A propriedade `tokenid` deve ser definida para cada token, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

O método aceita tokens com as propriedades padrão do token.

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
}
```

Origem

CToken::update() em *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.

Propriedade	Tipo	Descrição
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 tendências 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperando dados de tendências de 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() em *ui/include/classes/api/services/CTrend.php*.

Trigger

Esta classe foi projetada para trabalhar com triggers.

Referências de objetos:

- Trigger
- Trigger tag

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 reduzida do 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 se houver algum problema ao atualizar o estado do trigger. Comportamento da propriedade: - *somente leitura*
flags	integer	Origem do trigger. Valores possíveis: 0 - *(padrão)* um trigger simples; 4 - um trigger convertido de um protótipo. 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*

Propriedade	Tipo	Descrição
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.
recovery_expression	string	Expressão reduzida de recuperação do 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. Consulte 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.

Attention:
A expressão do trigger deve ser fornecida em sua forma expandida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos triggers criados na propriedade `triggerids`. A ordem dos IDs retornados corresponde à ordem dos triggers passados.

Exemplos

Criando um trigger

Crie um trigger com uma única dependência de 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 status",
            "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() em *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 dos triggers a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos triggers excluídos 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
}
```

Origem

CTrigger::delete() em *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 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
triggerids	ID/array	Retorna apenas triggers com os IDs fornecidos.
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 items fornecidos.
functions	string/array	Retorna apenas triggers que usam as funções fornecidas. Consulte a página de funções suportadas para 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 items 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 items desabilitados ou hosts de items desabilitados.
lastChangeSince	timestamp	Retorna apenas triggers que mudaram de estado após o tempo fornecido.
lastChangeTill	timestamp	Retorna apenas triggers que mudaram de estado antes do tempo fornecido.

Parâmetro	Tipo	Descrição
only_true	flag	Retorna apenas triggers que estiveram recentemente em estado de problema.
min_severity	integer	Retorna apenas triggers com severidade maior ou igual à severidade fornecida.
evaltype	integer	Regras para busca por tags. Valores possíveis: 0 - (padrão) E/Ou; 2 - Ou.
tags	array	Retorna apenas triggers com as tags fornecidas. Correspondência exata por tag e busca sensível ou não sensível a maiúsculas/minúsculas pelo valor da tag, dependendo do valor do operador. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os triggers. Tipos de operadores possíveis: 0 - (padrão) Contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 1 - Igual (sensível a maiúsculas/minúsculas); 2 - Não contém (correspondência de substring, não sensível a maiúsculas/minúsculas); 3 - Diferente (sensível a maiúsculas/minúsculas); 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.
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 items 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`.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta de trigger. O objeto de descoberta de trigger vincula um trigger descoberto a um protótipo de trigger do qual foi descoberto. 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 se o trigger foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.

Parâmetro	Tipo	Descrição
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`.
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 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	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
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	
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 foi criado. Esta consulta está **obsoleta**, por favor, use `selectDiscoveryData` em seu lugar.

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 severidade de todos os triggers em estado de problema e classifique-os por severidade 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() em *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.

> **Attention:**
> A expressão do trigger deve ser fornecida em sua forma expandida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos 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
}
```

Origem

CTrigger::update() em *ui/include/classes/api/services/CTrigger.php*.

Trigger prototype

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 reduzida do 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.
flags	integer	Origem do protótipo de trigger. Valores possíveis: 2 - um protótipo de trigger; 6 - um protótipo de trigger descoberto. Comportamento da propriedade: - *somente leitura*
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 de evento OK. Valores possíveis: 0 - *(padrão)* Expressão; 1 - Expressão de recuperação; 2 - Nenhum.
recovery_expression	string	Expressão reduzida de recuperação do 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.
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.

Propriedade	Tipo	Descrição
uuid	string	Identificador único universal, usado para vincular protótipos de trigger importados a 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 novas 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)` Prototipos 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 prototipos de trigger dos quais o protótipo de trigger depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags de protótipo de trigger.

> **Attention:**
> A expressão do trigger deve ser fornecida em sua forma expandida e deve conter pelo menos um protótipo de item.

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 trigger 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
}
```

Origem

CTriggerPrototype::create() em *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 do usuário. Consulte 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() em *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.

Parâmetro	Tipo	Descrição
monitored	flag	Retorna apenas protótipos de trigger habilitados que pertencem a hosts monitorados e contêm apenas itens 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`.
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta do protótipo de trigger. O objeto de descoberta do protótipo de trigger vincula um protótipo de trigger descoberto a um protótipo de trigger do qual foi descoberto. Possui as seguintes propriedades: `parent_triggerid` - `(ID)` ID do protótipo de trigger do qual o protótipo de trigger foi criado; `status` - `(int)` status de descoberta do protótipo de trigger: 0 - *(padrão)* protótipo de trigger foi descoberto, 1 - protótipo de trigger não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um protótipo de trigger que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um protótipo de trigger que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador se o protótipo de trigger foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
selectDiscoveryRule	query	Retorna a regra LLD à qual o protótipo de trigger pertence na propriedade `discoveryRule`.
selectDiscoveryRulePrototype	query	Retorna uma propriedade `discoveryRulePrototype` com o protótipo da regra LLD pai ao qual o protótipo de trigger pertence.
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 itens 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`.

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 correspondência. 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	

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

Recupera 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 da 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 da trigger prototype, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers e trigger prototypes das quais a trigger prototype depende. As triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags da trigger prototype.

> **Attention:**
> A expressão da trigger deve ser fornecida em sua forma expandida e deve conter pelo menos uma item prototype.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de trigger 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
}
```

Origem

CTriggerPrototype::update() em *ui/include/classes/api/services/CTriggerPrototype.php*.

Usuário

Esta classe foi projetada para trabalhar com usuários.

Referências de objetos:

- User
- Media

Métodos disponíveis:

- user.checkauthentication - verifica e prolonga sessões de usuários
- 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 de usuário

Os seguintes objetos estão diretamente relacionados à API de `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 do papel do usuário. Observe que usuários sem papel 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	Horário 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*
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.

Propriedade	Tipo	Descrição
provisioned	integer	Se o usuário foi provisionado. Valores possíveis: 0 - não provisionado; 1 - provisionado. 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.
ts_provisioned	timestamp	Horário em que a última operação de provisionamento foi realizada. Comportamento da propriedade: - *somente leitura* - *suportado* para usuários do tipo *Super admin*
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*
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*

Propriedade	Tipo	Descrição
mediatypeid	ID	ID do tipo de mídia usada 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 uma 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 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.
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*

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 deve prolongar a sessão do usuário. Valor padrão: ”true”. Definir o valor como ”false” permite verificar a sessão do usuário sem prolongá-la. Comportamento do parâmetro: - *suportado* se `sessionid` estiver definido
sessionid	string	Token de autenticação do usuário. Comportamento do parâmetro: - *obrigatório* se `token` não estiver definido
secret	string	String aleatória de 32 caracteres. É gerada no login do usuário.
token	string	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.
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",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "1",
        "deprovisioned": false,
        "auth_type": 0
    },
    "id": 1
}
```


Origem

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 adicionar o usuário. 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() em *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 dos 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() em *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 de 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 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 usam a mídia fornecida.
mediatypeids	ID/array	Retorna apenas usuários que usam os tipos de mídia fornecidos.
userids	ID/array	Retorna apenas usuários com os IDs fornecidos.
usrgrpids	ID/array	Retorna apenas usuários que pertencem aos grupos de usuários fornecidos.
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 usadas pelo usuário na propriedade `medias`.
selectMediatypes	query	Retorna os tipos de mídia usados 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`.
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. Propriedades possíveis do objeto Usuário para usuários do tipo *Admin* e *User* ao solicitar dados de usuários em seu grupo de usuários: `userid`, `name`, `surname`, `username`.
output	query	Propriedades do objeto Usuário a serem retornadas. 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 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 busca `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Propriedades possíveis do objeto Usuário para usuários do tipo *Admin* e *User* ao solicitar dados de usuários em seu grupo de usuários: `name`, `surname`, `username`.

Parâmetro	Tipo	Descrição
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `userid`, `username`.
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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

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() em *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 ativada.

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 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.
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 desabilitado 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.

Propriedade	Tipo	Descrição
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 que é necessário para a autenticação.

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.

Requisição:

```
{
    "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

Origem

CUser::login() em *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

Efetua 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

Origem

CUser::login() em *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 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

Provisione dois usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Origem

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 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 envia 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

Redefinir 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

Origem

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 de 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() em *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 do 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 Usuário estiver definido e o usuário alterar sua própria senha
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() em *ui/include/classes/api/services/CUser.php*.

Verificação de descoberta

Esta classe é 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”

Propriedade	Tipo	Descrição
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”
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”

Propriedade	Tipo	Descrição
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”
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.

Propriedade	Tipo	Descrição
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

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	Retornar apenas verificações de descoberta com os IDs fornecidos.
druleids	ID/array	Retornar apenas verificações de descoberta que pertencem às regras de descoberta fornecidas.
dserviceids	ID/array	Retornar apenas verificações de descoberta que detectaram os serviços descobertos fornecidos.
selectDRules	query	Retornar regras de descoberta relacionadas às verificações de descoberta.
sortfield	string/array	Classificar 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:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar verificações de descoberta para uma regra de descoberta

Recupera 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 `true` se passado e diferente de `null`; caso contrário, o valor é considerado `false`.
integer	Um número inteiro.
float	Um número de ponto flutuante.
string	Uma string de texto.

Tipo	Descrição
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 precisa ser definido, mas pode ser definido em algumas condições específicas (por exemplo, *suportado* se `type` estiver definido como ”Verificação simples”, ”Verificação externa”, ”agent SSH”, ”agent TELNET” ou ”agent HTTP”); observe, no entanto, que as propriedades *suportadas* ainda podem ser definidas para seus valores padrão, independentemente das condições;
- ***obrigatório*** - o valor da propriedade deve ser definido para todas as operações (exceto operações get) ou em algumas condições específicas (por exemplo, *obrigatório* para operações de criação; *obrigatório* se `operationtype` estiver 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 estão marcadas com rótulos são opcionais.

Comportamento dos parâmetros

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 é obrigatório, 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 parâmetros *suportados* ainda podem ser definidos para seus valores padrão, independentemente das condições;
- ***obrigatório*** - o valor do parâmetro é obrigatório.

Parâmetros que não são marcados com rótulos são opcionais.

Valor de ID reservado ”0” O valor de ID reservado ”0” pode ser usado para filtrar elementos e para 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.

Parâmetro	Tipo	Descrição
editable	boolean	Se definido como `true`, retorna apenas objetos para os quais o usuário tem permissões de gravação. 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`. 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.
output	query	Propriedades do objeto a serem retornadas. Padrão: `extend`.
preservekeys	boolean	Usa os IDs como chaves no array resultante.
search	object	Retorna resultados que correspondem ao padrão fornecido (não diferencia maiúsculas de minúsculas). 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`. 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`, permite 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 obter uma lista de propriedades que podem ser usadas para ordenação. As macros não são expandidas antes da ordenação. Se nenhum valor for especificado, os dados serão retornados sem ordenação.
sortorder	string/array	Ordem de classificação. Se um array for passado, cada valor será associado à propriedade correspondente fornecida no parâmetro `sortfield`. Valores possíveis: `ASC` - *(padrão)* crescente; `DESC` - decrescente.
startSearch	boolean	O parâmetro `search` irá comparar o início dos campos, ou seja, realizará uma pesquisa `LIKE "…%"`. Ignorado se `searchWildcardsEnabled` estiver definido como `true`.

Flags de origem da entidade Os métodos Get retornam uma propriedade `flags` para entidades relacionadas à descoberta de baixo nível (regra LLD/protótipo de regra LLD, item/protótipo de item, etc). Esta propriedade é útil para indicar se a entidade foi descoberta ou não, já que a edição de entidades descobertas é limitada.

A propriedade `flags` retorna um resultado baseado em uma combinação (operação "+") destes valores:

Valor	Descrição
0	Entidade base (item, trigger, gráfico, host)
1	Regra de descoberta de baixo nível
2	Qualquer protótipo (protótipo de item, protótipo de trigger, protótipo de regra LLD, etc)
4	Entidade descoberta (item, trigger, gráfico, host, regra LLD descobertos)

O valor **combinado** retornado pela propriedade `flags` pode ser:

Valor	Combinação de	Descrição
0	0	Entidade simples (item, trigger, gráfico, host).
2	2	Protótipo de entidade (protótipo de item, protótipo de trigger, etc).
6	2+4	Item, trigger, gráfico, host descobertos (convertidos de protótipo).
1	1	Regra de descoberta de baixo nível.
3	1+2	Protótipo de regra de descoberta de baixo nível.
5	1+4	Regra de descoberta de baixo nível descoberta (convertida de protótipo).
7	1+2+4	Protótipo de regra de descoberta de baixo nível descoberta.

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 o id 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 da 7.4 para a 8.0

Mudanças incompatíveis com versões anteriores

Outras alterações e correções de bugs dashboard

ZBXNEXT-8160 Adicionados novos campos de widget de dashboard `clustering_mode` e `clustering_zoom_level` ao widget geomap.

userdirectory

ZBXNEXT-9834 Adicionadas as propriedades `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` e `sp_certificate_hash` ao objeto user directory.

Apêndice 3. Alterações na 8.0

8.0.1 Esta versão ainda não foi lançada.

21 Extensões

Visão geral Embora o Zabbix ofereça uma multiplicidade de recursos, sempre há espaço para funcionalidades adicionais. As extensões são uma maneira conveniente de modificar e aprimorar as capacidades de monitoramento do Zabbix sem alterar seu código-fonte.

Você pode estender a funcionalidade do Zabbix usando opções de extensão integradas (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 para estender o 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 servidor 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 servidor ou proxy Zabbix é possível usando a utilidade Zabbix sender ou o protocolo do Zabbix sender. O envio de dados para o servidor Zabbix também é possível usando o método de API `history.push`.

Verificações externas

Uma verificação externa é um item para executar verificações executando um executável, por exemplo, um script shell ou um binário.

As verificações externas são executadas pelo Zabbix server ou proxy (quando o host é monitorado por proxy), e não requerem um agent em execução no host monitorado.

Parâmetros de 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 onde o agent Zabbix está em execução. Os parâmetros de usuário são úteis para configurar items do agent ou agent 2 que não são predefinidos no Zabbix.

`system.run[]` itens do agent Zabbix

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`).

Attention:
Comandos definidos pelo usuário em items, como verificações externas, parâmetros de usuário e items do agent Zabbix `system.run[]` são executados a partir do usuário do SO que é usado para executar os componentes do Zabbix. Para executar esses comandos, esse usuário deve ter as permissões necessárias.

Itens de agente HTTP

O item agente HTTP é um item para executar solicitações de dados via HTTP/HTTPS. 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. Itens de agente HTTP (com trapping habilitado) também podem funcionar como itens trapper.

Items de script

Um item de script é um item para executar código JavaScript definido pelo usuário que recupera dados via HTTP/HTTPS. Itens de script são úteis quando a funcionalidade fornecida por itens de agente HTTP não é suficiente. Por exemplo, em cenários exigentes de coleta de dados que requerem múltiplas etapas ou lógica complexa, um item de script pode ser configurado para fazer uma chamada HTTP, processar os dados recebidos e, em seguida, passar o valor transformado para uma segunda chamada HTTP.

> **Note:**
> Itens de agente HTTP e itens de script são suportados pelo Zabbix server e proxy, e não requerem um agent em execução no host monitorado.

Extensões avançadas Módulos carregáveis

Módulos carregáveis, escritos em C, são uma opção versátil e orientada para desempenho para estender a funcionalidade dos componentes do Zabbix (server, proxy, agent) em plataformas UNIX. 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.

Os módulos carregáveis têm vários benefícios, incluindo a capacidade de adicionar novas métricas ou implementar qualquer outra lógica (por exemplo, exportação de dados históricos do Zabbix), 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 entregar novas funcionalidades de forma mais fácil e independente do código-fonte principal do Zabbix.

Módulos carregáveis são especialmente úteis em uma configuração de monitoramento complexa. Ao monitorar sistemas embarcados, 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 agent `system.run[]` e verificações externas terão impacto no desempenho. Módulos carregáveis oferecem uma maneira de estender a funcionalidade do Zabbix sem sacrificar o desempenho.

Plugins

Plugins fornecem uma alternativa aos módulos carregáveis (escritos em C). No entanto, plugins são uma forma de estender apenas o agent Zabbix 2.

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 agent Zabbix 2 são suportados:

- Plugins embutidos (suportado desde o Zabbix 4.4.0)
- Plugins carregáveis (suportado desde o Zabbix 6.0.0)

Veja a lista de plugins embutidos.

Para instruções e tutoriais sobre como escrever seus próprios plugins, veja o Centro de desenvolvedores.

Personalização de alerta Webhooks

Um webhook é um tipo de mídia do Zabbix que oferece uma opção para estender as capacidades de alerta do Zabbix para softwares externos, como sistemas de helpdesk, chats ou mensageiros. Semelhante aos 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 tickets de helpdesk criados) que são então exibidos no Zabbix.

Webhooks existentes estão disponíveis no repositório Git do Zabbix. Para desenvolvimento de webhooks personalizados, consulte as Diretrizes de desenvolvimento de webhooks.

Scripts de alerta

Um script de alerta é um tipo de mídia do Zabbix que oferece uma opção para criar uma maneira alternativa (script) de lidar com alertas do Zabbix. Scripts de alerta são úteis se você não estiver satisfeito com os tipos de mídia existentes para envio de alertas no Zabbix.

Personalização do frontend 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 do frontend

Módulos do frontend fornecem uma opção para estender a funcionalidade do frontend do Zabbix adicionando módulos de terceiros ou desenvolvendo os seus próprios. Com os módulos do frontend, você pode adicionar novos itens de menu, suas respectivas visualizações, ações, etc.

Scripts globais Um script global é um conjunto de comandos definidos pelo usuário que podem ser executados 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:

- Operação de ação
- Ação manual de host
- Ação manual de evento

Os scripts globais são úteis em muitos casos. Por exemplo, se configurados para operações de ação ou ações manuais de host, você pode usar scripts globais para executar automaticamente ou manualmente comandos remotos como reiniciar um aplicativo (servidor web, middleware, CRM, etc.) ou liberar espaço em disco (removendo arquivos antigos, limpando `/tmp`, etc). Ou, outro exemplo, se configurados para ações manuais de eventos, você pode usar scripts globais para gerenciar tickets de problemas em sistemas externos.

Os scripts globais podem ser executados pelo Zabbix server, proxy ou agent.

> **Attention:**
> Os comandos definidos pelo usuário são executados a partir do usuário do SO que é usado para executar os componentes do Zabbix. Para executar esses comandos, esse usuário deve ter as permissões necessárias.

Zabbix API Zabbix API é uma API baseada em HTTP que faz parte do frontend do Zabbix. Com a Zabbix API, 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 softwares de terceiros.
- Automatizar tarefas rotineiras.

A Zabbix API consiste em uma multiplicidade de métodos que são agrupados nominalmente em APIs separadas. Cada método executa uma tarefa específica. Para os métodos disponíveis, bem como uma visão geral das funções fornecidas pela Zabbix API, consulte a Referência de métodos da Zabbix API.

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 `system.run[]` itens de agent do Zabbix. Esses métodos 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 isso não é um grande problema, no entanto, pode ser um problema sério ao monitorar sistemas embarcados, ter um grande número de parâmetros monitorados ou scripts pesados com lógica complexa ou tempo de inicialização longo.

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 com ele.

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 e a distribuição de módulos em formato binário 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 de módulos 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 do Zabbix é iniciado com a flag ”-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 de 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 (sem sinal), 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 fontes 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 precedente for absoluto (começar com '/'), então LoadModulePath será ignorado. É permitido incluir múltiplos 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 irá carregar 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 irá iniciar se um módulo estiver faltando, 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 (ativo)”. Se o módulo for carregado no server ou proxy, então o tipo de item deve ser ”Verificação simples”.

A exportação de histórico através de módulos do Zabbix não necessita 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
**
** Este programa é um software livre; você pode redistribuí-lo e/ou modificá-lo
** sob os termos da Licença Pública Geral GNU conforme publicada pela
** Free Software Foundation; tanto a versão 2 da Licença, ou
** (a seu critério) qualquer versão posterior.
**
** Este programa é distribuído na esperança de que seja útil,
** mas SEM QUALQUER GARANTIA; sem mesmo a garantia implícita de
** COMERCIALIZAÇÃO ou ADEQUAÇÃO A UM DETERMINADO FIM. Veja a
** Licença Pública Geral GNU para mais detalhes.
**
** Você deve ter recebido uma cópia da Licença Pública Geral GNU
** junto com este programa; se não, escreva para a 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"

/* a variável mantém a configuração de timeout para o processamento do item */
static int  item_timeout = 0;

/* o módulo DEVE definir funções internas como static e usar um padrão de nomenclatura diferente dos símbo
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[] =
/*  CHAVE         FLAG        FUNÇÃO      PARÂMETROS DE TESTE */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
```

```
 *                                                                            *
 * Função: zbx_module_api_version                                             *
 *                                                                            *
 * Objetivo: retorna o número da versão da interface do módulo                *
 *                                                                            *
 * Valor de retorno: ZBX_MODULE_API_VERSION - versão do module.h com o qual o *
 *                   módulo foi compilado, para carregar o módulo com sucesso *
 *                   o Zabbix DEVE ser compilado com a mesma versão deste     *
 *                   arquivo de cabeçalho                                     *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Função: zbx_module_item_timeout                                            *
 *                                                                            *
 * Objetivo: define o valor de timeout para o processamento dos items         *
 *                                                                            *
 * Parâmetros: timeout - timeout em segundos, 0 - sem timeout definido        *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Função: zbx_module_item_list                                               *
 *                                                                            *
 * Objetivo: retorna a lista de chaves de item suportadas pelo módulo         *
 *                                                                            *
 * Valor de retorno: lista de chaves de item                                  *
 *                                                                            *
 ******************************************************************************/
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)
    {
        /* define mensagem de erro opcional */
        SET_MSG_RESULT(result, strdup("Número inválido de parâmetros."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);
```

```
    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Função: dummy_random                                                       *
 *                                                                            *
 * Objetivo: ponto de entrada principal para o processamento de um item       *
 *                                                                            *
 * Parâmetros: request - estrutura que contém a chave do item e parâmetros    *
 *              request->key - chave do item sem parâmetros                   *
 *              request->nparam - número de parâmetros                        *
 *              request->params[N-1] - ponteiros para os parâmetros da chave  *
 *              request->types[N-1] - tipos dos parâmetros da chave:          *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (parâmetro vazio)        *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (string entre aspas ou não) *
 *                                                                            *
 *             result - estrutura que conterá o resultado                     *
 *                                                                            *
 * Valor de retorno: SYSINFO_RET_FAIL - função falhou, o item será marcado    *
 *                                 como não suportado pelo zabbix             *
 *               SYSINFO_RET_OK - sucesso                                     *
 *                                                                            *
 * Comentário: get_rparam(request, N-1) pode ser usado para obter um ponteiro *
 *          para o N-ésimo parâmetro a partir de 0 (primeiro parâmetro).      *
 *          Certifique-se de que ele existe verificando o valor de            *
 *          request->nparam. Da mesma forma, get_rparam_type(request, N-1)    *
 *          pode ser usado para obter o tipo do parâmetro.                    *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* define mensagem de erro opcional */
        SET_MSG_RESULT(result, strdup("Número inválido de parâmetros."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* não há validação estrita de parâmetros e tipos para simplificar */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Intervalo especificado inválido."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

```c
/******************************************************************************
 *                                                                            *
 * Função: zbx_module_init                                                    *
 *                                                                            *
 * Objetivo: a função é chamada na inicialização do agent                     *
 *           Deve ser usada para chamar rotinas de inicialização              *
 *                                                                            *
 * Valor de retorno: ZBX_MODULE_OK - sucesso                                  *
 *                ZBX_MODULE_FAIL - falha na inicialização do módulo          *
 *                                                                            *
 * Comentário: o módulo não será carregado em caso de ZBX_MODULE_FAIL         *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* inicialização para dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Função: zbx_module_uninit                                                  *
 *                                                                            *
 * Objetivo: a função é chamada no desligamento do agent                      *
 *           Deve ser usada para liberar recursos utilizados, se houver       *
 *                                                                            *
 * Valor de retorno: ZBX_MODULE_OK - sucesso                                  *
 *                ZBX_MODULE_FAIL - função falhou                             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Funções: dummy_history_float_cb                                            *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Objetivo: funções de callback para armazenar dados históricos dos tipos    *
 *           float, integer, string, text e log, respectivamente, em          *
 *           armazenamento externo                                            *
 *                                                                            *
 * Parâmetros: history     - array de dados históricos                        *
 *             history_num - número de elementos no array history             *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* faça algo com history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}
```

```c
static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* faça algo com 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++)
    {
        /* faça algo com 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++)
    {
        /* faça algo com 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++)
    {
        /* faça algo com history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Função: zbx_module_history_write_cbs                                       *
 *                                                                            *
 * Objetivo: retorna um conjunto de funções do módulo que o Zabbix chamará    *
 *           para exportar diferentes tipos de dados históricos               *
 *                                                                            *
 * Valor de retorno: estrutura com ponteiros para funções de callback (pode   *
 *                ser NULL se o módulo não estiver interessado em dados de    *
 *                certos tipos)                                               *
 *                                                                            *
 ******************************************************************************/
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 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 fornecem uma opção para estender as capacidades de monitoramento do Zabbix. Plugins são escritos na linguagem de programação Go e são suportados apenas pelo Zabbix agent 2. Plugins fornecem uma alternativa aos módulos carregáveis (escritos em C) e outros métodos para estender a funcionalidade do Zabbix.

As seguintes funcionalidades são específicas do agent 2 e seus plugins:

- suporte a intervalos agendados e flexíveis para verificações passivas e ativas
- gerenciamento de fila de tarefas com respeito ao agendamento e concorrência de tarefas
- timeouts em nível de plugin
- verificação de compatibilidade do Zabbix agent 2 e seus plugins na inicialização

Desde o Zabbix 6.0, os plugins não precisam ser integrados diretamente ao agent 2 e podem ser adicionados como plugins carregáveis, facilitando assim o processo de criação de plugins adicionais para coletar novas métricas de monitoramento.

Esta página lista os plugins nativos e carregáveis do Zabbix e descreve os princípios de configuração de plugins do ponto de vista do usuário.

> **Note:**
> Para instruções sobre como escrever seus próprios plugins, consulte o Centro de desenvolvedores e o Exemplo de plugin para o Zabbix agent 2.
>
> Para detalhes sobre o processo de comunicação entre o Zabbix agent 2 e um plugin carregável, bem como o processo de coleta de métricas, consulte o Diagrama de conexão.

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 via 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 o intervalo de atualização *Scheduling* 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 `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 na chave do item ou nos parâmetros da sessão nomeada, o plugin usará o valor padrão. Se um parâmetro padrão também não estiver definido, valores padrão internos serão usados.

> **Note:**
> Se uma chave de item não tiver nenhum parâmetro, o Zabbix agent 2 tentará coletar a métrica usando os valores definidos na seção de parâmetros padrão.

Sessões nomeadas

As 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.

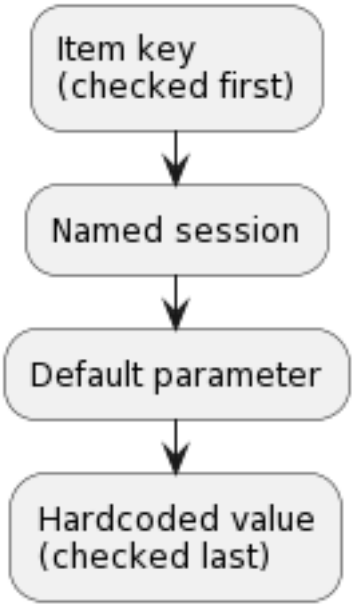
A lista de parâmetros de sessão nomeada disponíveis depende do plugin.

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 os 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; nesse 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 dessa 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 recomendado 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

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

Integrados

Os seguintes plugins para o 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.
CPU	Monitoramento da CPU do sistema (número de CPUs/núcleos, CPUs descobertas, percentual de uso).	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.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
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 em tópicos MQTT.	mqtt.get	Para configurar a conexão criptografada com o broker MQTT, especifique os parâmetros TLS no arquivo de configuração do agent como sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros da chave do 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 a conexão criptografada com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros da chave do item.
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 uso 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	

Nome do plugin	Descrição	Chaves de item suportadas	Comentários

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.
			São necessários direitos de acesso sudo/root ao smartctl 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.
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.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
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.
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.

Carregáveis

Plugins carregáveis para o Zabbix agent 2 não estão disponíveis por padrão e precisam ser instalados separadamente:

- Para Linux, você pode usar pacotes ou compilar plugins.
- Para Windows, você pode instalar via MSI ou compilar plugins.

Clique no nome do plugin para ir ao repositório do plugin com informações adicionais.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
Ceph	Monitoramento do Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Veja também parâmetros de configuração do plugin Ceph.
Ember+	Monitoramento do Ember+.	ember.get	Atualmente disponível apenas para compilação a partir do código-fonte (para Unix e Windows). Veja também parâmetros de configuração do plugin Ember+.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
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.
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.
NVIDIA GPU	Monitoramento de GPU NVIDIA.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	Veja também parâmetros de configuração do plugin NVIDIA GPU.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
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.
 - `-t, --test <item key>` — executa o plugin para teste (configuração do plugin ignorada).

1 Compilando plugins carregáveis

Visão geral

Esta página fornece as etapas necessárias para construir um binário de plugin carregável a partir dos fontes.

Se o tarball de fontes for baixado, é possível compilar o plugin offline, ou seja, sem conexão com a internet.

O plugin PostgreSQL é usado como exemplo. Outros plugins carregáveis podem ser compilados de forma semelhante.

Etapas

1. Baixe os fontes do plugin em Zabbix Cloud Images and Appliances. A página oficial de download estará disponível em breve.

2. Transfira o arquivo compactado para a máquina onde você irá compilar o plugin.

3. Extraia o arquivo tarball, por exemplo:

```
tar xvf zabbix-agent2-plugin-postgresql-1.0.0.tar.gz
```

Certifique-se de substituir "zabbix-agent2-plugin-postgresql-1.0.0.tar.gz" pelo nome do arquivo baixado.

4. Entre no diretório extraído:

```
cd <caminho para o diretório>
```

5. Execute:

```
make
```

6. O executável do plugin pode ser colocado em qualquer lugar, desde que seja carregável pelo agent Zabbix 2. 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
```

7. O caminho para o arquivo de configuração do plugin deve ser especificado no parâmetro Include do arquivo de configuração do agent Zabbix 2:

```
Include=/caminho/para/o/arquivo/de/configuracao/do/plugin/postgresql.conf
```

Alvos do Makefile

Os plugins carregáveis fornecidos pelo Zabbix possuem makefiles simples com os seguintes alvos:

Alvo	Descrição
make	Compilar o plugin.
make clean	Excluir todos os arquivos que normalmente são criados ao compilar o plugin.
make check	Executar auto-testes. Um banco de dados PostgreSQL real é necessário.
make style	Verificar o estilo do código Go com 'golangci-lint'.
make format	Formatar o código Go com 'go fmt'.
make dist	Criar um arquivo contendo os fontes do plugin e os fontes de todos os pacotes necessários para compilar o plugin e seus auto-testes.

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 informando 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

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 baixou o 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 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 execução (+rx) à pasta do módulo e a todos os subdiretórios, e acesso de leitura (+r) a todos os arquivos dentro dela.

Problema	Solução
Frontend travou	O código do módulo não é compatível com a versão atual do Zabbix ou com a configuração do servidor. 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 em *Procurar diretório* novamente para remover módulos inexistentes do banco de dados.
Mensagem de erro sobre namespace, ID ou ações idênticas	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	Reporte os erros ao desenvolvedor do módulo.

Desenvolvendo módulos

Para obter informações sobre o desenvolvimento de módulos personalizados, consulte o Centro do desenvolvedor.

22 Apêndices

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

1 Instalação e configuração

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 Zabbix server ou proxy.

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

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 collation do banco de dados Zabbix. Veja também Limites de filtragem com collations 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. 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/mysql
mysql -uzabbix -p<senha> zabbix < schema.sql
#### pare aqui se estiver criando o banco de dados para o proxy do Zabbix
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 no banco de dados.

Se você está instalando a partir de **pacotes** do Zabbix, siga as instruções para sua plataforma.

Se você está instalando o Zabbix a partir do código-fonte:

- Crie um usuário de banco de dados.

O seguinte comando shell irá criar 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 irá criar 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 schema 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 você 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 qualquer 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.

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 a collation do banco de dados do 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' passará a ser uma referência para 'utf8mb4'. Desde o Zabbix 6.0, 'utf8mb4' é suportado. Para evitar problemas futuros, é altamente recomendado usar 'utf8mb4'. Outra vantagem de migrar 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 schema do banco de dados para 6.0.x ou superior 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 <seu nome do BD> 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 <seu nome do BD> < utf8mb4_convert.sql
```

6. Execute o script:

```
                SET @ZABBIX_DATABASE = '<seu nome do BD>';
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 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 suficiente em disco; 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 de tabelas para chaves primárias também atualiza as tabelas para usar tipos de dados de precisão dupla.

 Se você possui o Zabbix 7.0 (ou posterior), as tabelas já usam precisão dupla. No entanto, as instruções desta página ainda podem ser seguidas para atualizar as tabelas para chaves primárias sem afetar as tabelas que já usam precisão dupla.

 Se você possui o Zabbix 6.4 (ou anterior), considere atualizar as tabelas para precisão dupla primeiro. Para mais informações, consulte Atualizando para valores numéricos de faixa estendida na documentação do Zabbix 7.0.

As instruções estão disponíveis para:

- MySQL
- PostgreSQL
- TimescaleDB

Notas importantes

Para executar a atualização do banco de dados:

1. Pare o Zabbix server.

Parar o Zabbix server durante a atualização é fortemente recomendado. No entanto, se absolutamente necessário, você pode executar a atualização enquanto o server está em execução (apenas para MySQL, MariaDB e PostgreSQL sem TimescaleDB).

2. Faça backup do seu banco de dados.
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 8.4+ com mysqlsh

Este método pode ser usado com um Zabbix server em execução, mas é recomendável parar o server durante o tempo da atualização. O MySQL Shell (*mysqlsh*) deve estar instalado e 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 8.4+ 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.
- 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 server 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 server 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 para os arquivos nas queries conforme necessário ou defina o valor de *secure_file_priv* como uma string vazia durante a 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, ignore 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 e, 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 server em execução

- Exporte o histórico atual, importe-o para a tabela temporária e, 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 Zabbix server deve estar parado 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 de hypertable do TimescaleDB de acordo com as 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 de INSERT.

- Siga as instruções pós-migração para remover as tabelas antigas.

Pós-migração

Para todos os bancos de dados, uma vez que a migração esteja concluída, faça o seguinte:

- Verifique se tudo está funcionando conforme o 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 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 futuros patches de atualização, 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 colocar o Zabbix frontend em modo de manutenção durante a alteração. No entanto, se absolutamente necessário, existe uma maneira de alterar a chave primária sem tempo de inatividade (veja abaixo).

O particionamento da tabela `auditlog` pode melhorar, por exemplo, a limpeza em ambientes 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 manualmente os índices. 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 reconstrução de índices).

```
OPTIMIZE TABLE auditlog;
```

Alterando a chave primária sem tempo de inatividade

O método manual de alteração da chave primária está descrito aqui. Como alternativa, você pode usar o pt-online-schema-change da Percona. Essa ferramenta executa as seguintes ações automaticamente, minimizando também 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 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 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 (múltiplas instruções `INSERT INTO` com cláusulas `WHERE clock` conforme necessário) para evitar 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	Frontend do Zabbix, server do Zabbix, proxy do Zabbix
PostgreSQL	Frontend do Zabbix, server do Zabbix, proxy do Zabbix

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: replicação em grupo, etc. servidores de banco de dados.
- 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 frequentemente são fornecidos com versões desatualizadas do software de banco de dados para as quais o suporte à criptografia não está implementado, por exemplo, sistemas baseados em RHEL 7 e PostgreSQL 9.2, MariaDB 5.5 sem suporte à criptografia.

Terminologia

Definir esta opção força 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 *Criptografia TLS do banco de dados* na etapa Configurar conexão com o BD para habilitar a criptografia de transporte.
- Marque a caixa *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 *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 *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 estiverem marcadas):

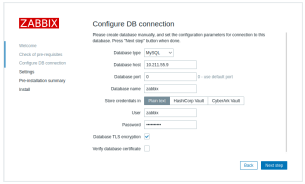
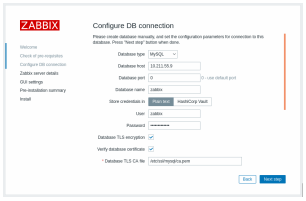
Parâmetro	Descrição
Arquivo CA TLS do banco de dados	Especifique o caminho completo para um arquivo de autoridade certificadora (CA) TLS válido.
Arquivo de chave TLS do banco de dados	Especifique o caminho completo para um arquivo de chave TLS válido.
Arquivo de certificado TLS do banco de dados	Especifique o caminho completo para um arquivo de certificado TLS válido.
Verificação do host do banco de dados	Marque esta caixa 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 do 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 da 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 da 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 Zabbix server/proxy

Conexões seguras com o banco de dados podem ser configuradas com os respectivos parâmetros no arquivo de configuração do Zabbix server e/ou proxy.

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 da 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 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 8.4.0 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 (o Performance Schema deve estar ativado):

```
SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, processli
FROM performance_schema.status_by_thread  AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

Criptografia apenas de transporte

Configuração do MySQL

Versões modernas do banco de dados já estão prontas para o modo de criptografia `required` por padrão. Um certificado do lado do servidor será criado após a configuração e inicialização.

Crie usuários e funções para os principais componentes:

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

Observe que o protocolo X.509 não é usado para verificar a identidade, mas o usuário está configurado para usar apenas conexões criptografadas. Veja 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 os conjuntos de cifras disponíveis:

```
mysql> status
--------------
mysql Ver 8.4.0 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384


mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Frontend

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 de transporte para 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

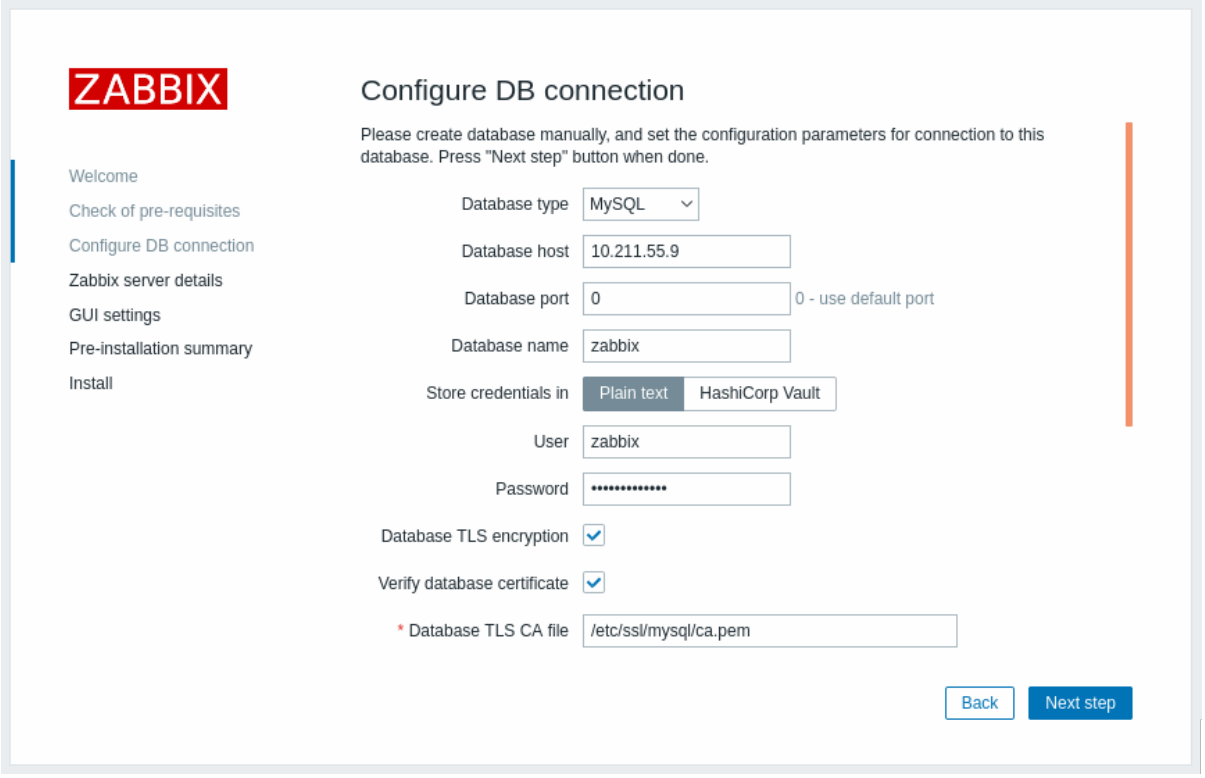
Copie o CA do MySQL necessário para o servidor frontend do Zabbix, atribua as permissões adequadas para permitir que o servidor web leia este arquivo.

> **Note:**
> Este modo não funciona no 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 para 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 MySQL ou Criando certificados e chaves SSL usando 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:

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

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
```

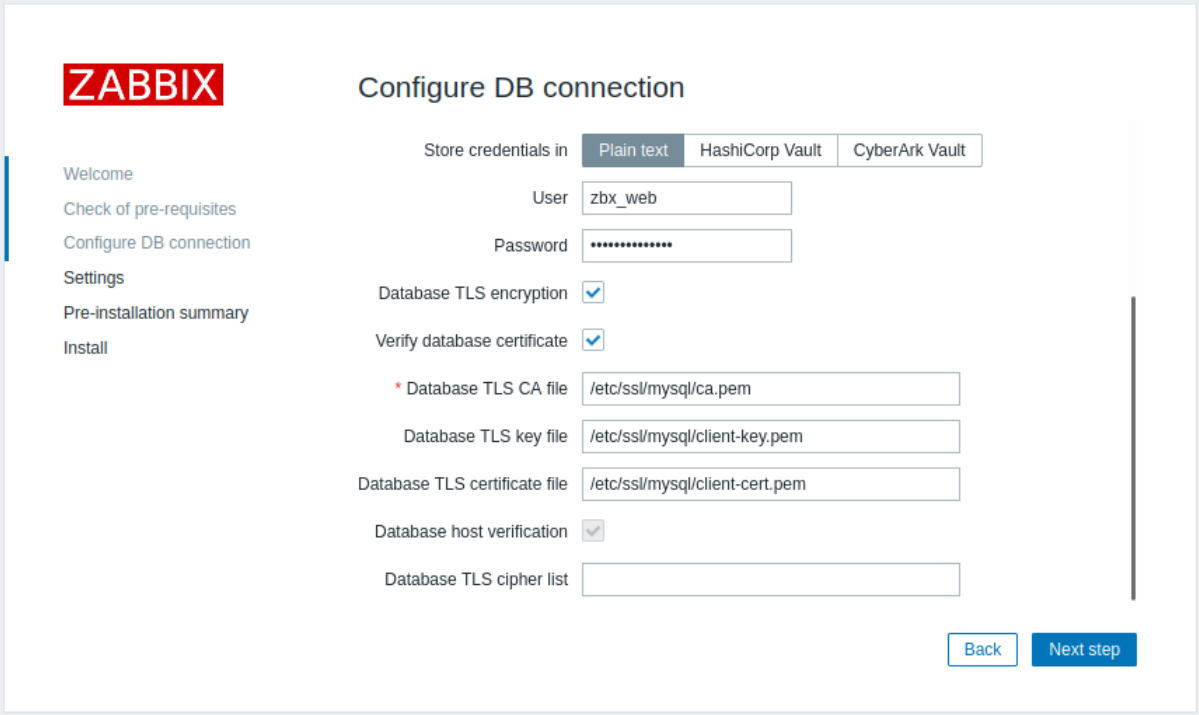
Frontend

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 pulada 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 permitir conexões remotas de rede, permita escutar 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

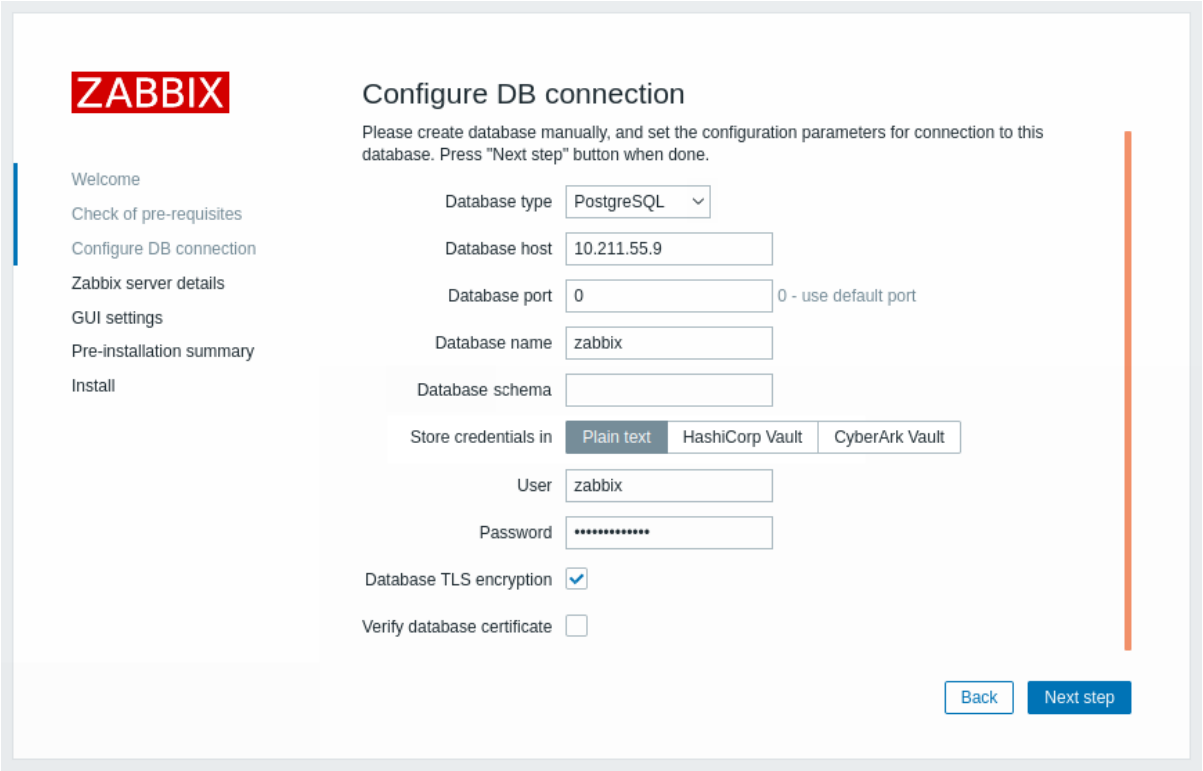
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

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



Servidor

Para habilitar a criptografia apenas de transporte para conexões entre o servidor e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

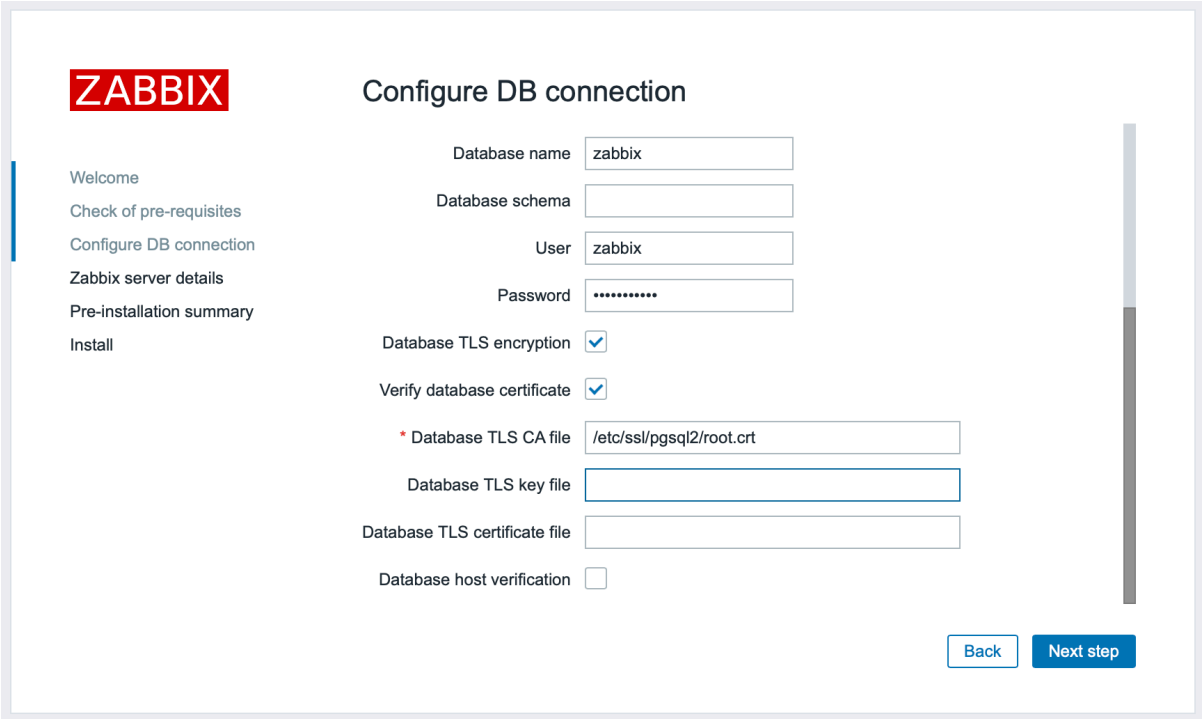
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Criptografia com verificação de autoridade certificadora

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:*

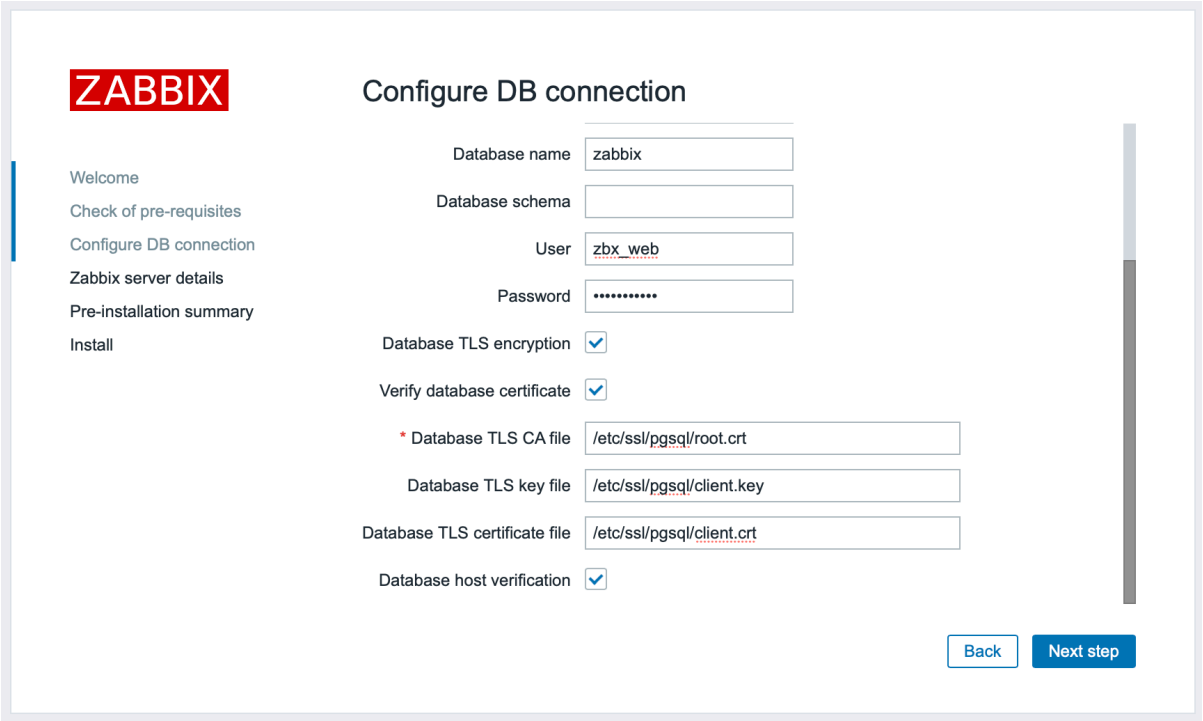
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

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 Conexão segura com o frontend

Visão geral

Esta seção fornece etapas de configuração do Zabbix e exemplos de configuração para conexões TLS seguras entre o frontend do Zabbix e o server do Zabbix.

Configuração

Por padrão, a comunicação entre o frontend do Zabbix e o server do Zabbix não é criptografada. Para maior segurança, habilite o TLS em ambos os lados. Abaixo está um exemplo da maneira mais simples de fazer isso.

1. Gere certificados e chaves.

Crie um diretório de trabalho:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Crie um certificado CA (ajuste o valor `MyZabbixCA` para corresponder ao nome comum real):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Gere uma chave privada e um certificado para o server do Zabbix (ajuste o valor `zabbix-server.example.com` para corresponder ao nome comum real):

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Gere uma chave privada e um certificado para o frontend do Zabbix (ajuste o valor `zabbix-frontend-node` para corresponder ao nome comum real):

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. Defina as permissões adequadas.

Para o server do Zabbix (ajuste o proprietário/grupo de acordo com o usuário do daemon do server do Zabbix da sua distribuição):

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

Para o frontend (ajuste o proprietário/grupo de acordo com o usuário do servidor web da sua distribuição):

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Configure o server do Zabbix.

No `zabbix_server.conf` adicione:

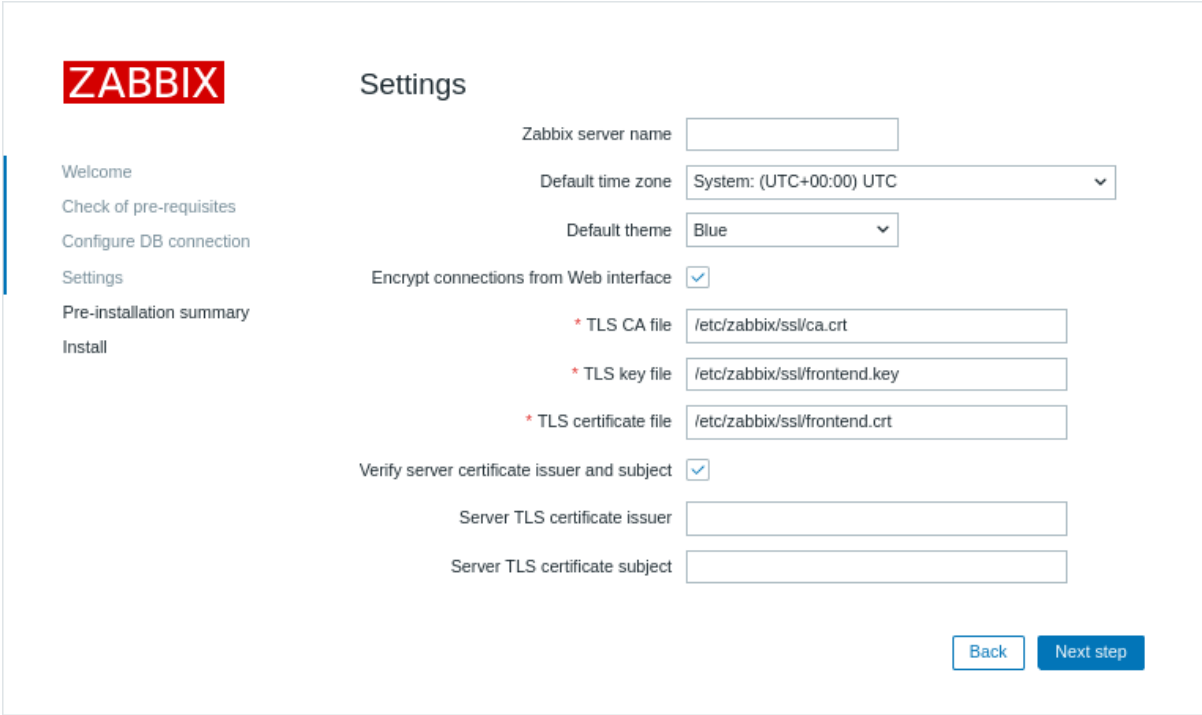
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Opcionalmente:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Em seguida, reinicie o server:

```
sudo systemctl restart zabbix-server
```

4. Configure o frontend do Zabbix.

Durante a instalação da interface web, habilite a opção *Criptografar conexões da interface web* (e a opção *Verificar emissor e assunto do certificado do servidor*, se necessário) e preencha os campos *Arquivo CA TLS*, *Arquivo de chave TLS*, *Arquivo de certificado TLS* (e os campos *Emissor do certificado TLS do servidor* e *Assunto do certificado TLS do servidor*, se necessário):



Parâmetro	Descrição
Arquivo CA TLS	Especifique o caminho completo para o arquivo do certificado da Autoridade Certificadora (CA) usado para verificar o certificado do server.
Arquivo de chave TLS	Especifique o caminho completo para o arquivo da chave privada do cliente correspondente ao certificado do cliente.
Arquivo de certificado TLS	Especifique o caminho completo para o arquivo do certificado do cliente se a autenticação mútua TLS for necessária.
Emissor do certificado TLS do servidor	Especifique um Nome Distinto (DN) do emissor para corresponder ao certificado do server.
Assunto do certificado TLS do servidor	Especifique um Nome Distinto (DN) do assunto para corresponder ao certificado do server.

Em instalações existentes, edite os seguintes campos em `zabbix.conf.php`:

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Opcionalmente:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Verifique a criptografia confirmando que não há mensagens de erro no frontend do Zabbix ou no arquivo de log do server do Zabbix:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 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 suportar um desempenho mais rápido em grande 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 (veja Configuração).
- Atualizar um esquema de banco de dados TimescaleDB existente ao atualizar o Zabbix (veja 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.

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 mensagens de aviso informando que as melhores práticas não estão sendo seguidas ao executar o script `schema.sql` em TimescaleDB versão 2.9.0 ou superior. 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 servidor e o frontend do Zabbix 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 tendências do item

Para usar a limpeza particionada para histórico e tendências, ambas as opções devem ser habilitadas. Também é possível habilitar a substituição individualmente, apenas para histórico ou apenas para tendências.

Para PostgreSQL e TimescaleDB, o script `postgresql/timescaledb/schema.sql` define dois parâmetros adicionais:

- Habilitar compressão
- Comprimir registros mais antigos que 7 dias

Para remover com sucesso os dados compactados pelo housekeeper, as opções *Substituir período de histórico do item* e *Substituir período de tendências do item* devem estar ativadas. Se a substituição estiver desativada e as tabelas tiverem blocos compactados, o housekeeper não removerá 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* → *Housekeeping* 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 servidor Zabbix não configura automaticamente esses hypertables (por exemplo, ao atualizar do Zabbix 6.4 para o 8.0, pois as versões 7.0.0 e 7.0.2 introduziram novos hypertables).

Para configurar os novos hypertables do TimescaleDB, siga estas etapas:

1. Inicie o servidor Zabbix; isso atualizará o banco de dados existente.
2. Verifique no arquivo de log do servidor se a atualização do banco de dados foi concluída; quando concluída, pare o servidor Zabbix. Observe que o servidor 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, o local e o nome do script mudaram 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` em versões do TimescaleDB 2.9.0 ou superiores. Independentemente desse aviso, a configuração será concluída com sucesso.

Compressão do 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 grandes tabelas 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 algumas 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* na seção *Administração → Limpeza* 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` (configurado 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 dessa configuração é aderir às melhores práticas do TimescaleDB e garantir que o tamanho do chunk permaneça dentro dos recursos disponíveis do sistema.

8 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 usado, 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 que os parâmetros do arquivo de configuração do server e do arquivo de configuração do frontend estejam devidamente configurados.

Zabbix server e frontend

Rascunho do arquivo de configuração do Zabbix server com parâmetros a serem atualizados:

```
### Option: HistoryStorageURL
# URL HTTP[S] de armazenamento do histórico.
#
# Obrigatório: não
# Padrão:
# HistoryStorageURL=
### Option: HistoryStorageTypes
# Lista separada por vírgulas dos tipos de valores a serem enviados para o armazenamento do histórico.
#
# Obrigatório: não
# Padrão:
# HistoryStorageTypes=uint,dbl,str,log,text
```

Exemplo de valores de parâmetros para preencher o arquivo de configuração do Zabbix server:

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,text
```

Esta configuração força o Zabbix Server a armazenar valores históricos de tipos numéricos no banco de dados correspondente e dados históricos textuais no Elasticsearch.

O Elasticsearch suporta os seguintes tipos de item:

`uint,dbl,str,log,text`

Explicação dos tipos de item suportados:

Tipo de valor do item	Tabela do banco de dados	Tipo do Elasticsearch
Numérico (sem sinal)	history_uint	uint
Numérico (float)	history	dbl
Caractere	history_str	str
Log	history_log	log
Texto	history_text	text

Rascunho do arquivo de configuração do Zabbix frontend (`conf/zabbix.conf.php`) com parâmetros a serem atualizados:

```
// URL do Elasticsearch (pode ser string se a mesma URL for usada para todos os tipos).
$HISTORY['url']   = [
      'uint' => 'http://localhost:9200',
      'text' => 'http://localhost:9200'
];
// Tipos de valores armazenados no Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Exemplo de valores de parâmetros para preencher o arquivo de configuração do Zabbix frontend:

```
$HISTORY['url']   = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

Esta configuração força o armazenamento de valores históricos de `Texto`, `Caractere` e `Log` no Elasticsearch.

Também é necessário tornar `$HISTORY` global em `conf/zabbix.conf.php` para garantir que tudo esteja funcionando corretamente (veja `conf/zabbix.conf.php.example` para saber como fazer isso):

```
// Arquivo de configuração da interface web do Zabbix.
global $DB, $HISTORY;
```

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:
Mapping 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 conforme as instruções.

Para criar o mapeamento para o tipo `text`, envie a seguinte requisição 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 requisição 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 históricos em vários índices baseados em data

Esta seção descreve as etapas adicionais necessárias para trabalhar com pipelines e ingest nodes.

Para começar, você deve criar templates para os índices.

O exemplo a seguir mostra uma solicitação para criar um template uint:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/uint_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "uint*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "type": "long"
         }
      }
   }
}'
```

Para criar outros templates, o usuário deve alterar a URL (a última parte é o nome do template), alterar o campo "`index_patterns`" para corresponder ao nome do índice e definir o mapeamento válido, que pode ser obtido em `database/elasticsearch/elasticsearch.`

Por exemplo, o comando a seguir pode ser usado para criar um template para o índice de texto:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "text*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
```

```
            "type": "date"
         },
         "value": {
            "fields": {
               "analyzed": {
                  "index": true,
                  "type": "text",
                  "analyzer": "standard"
               }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Isso é necessário para permitir que o Elasticsearch defina o mapeamento válido para índices criados automaticamente. Em seguida, é necessário criar a definição do pipeline. O pipeline é uma espécie de pré-processamento dos dados antes de armazená-los nos índices. O comando a seguir pode ser usado para criar um pipeline para o índice uint:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily uint index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "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 alterar a URL (a última parte é o nome do pipeline) e alterar o campo ”index_name_prefix” para corresponder ao nome do índice.

Veja também a documentação do Elasticsearch.

Além disso, o armazenamento de dados históricos em vários índices baseados em data também deve ser habilitado no novo parâmetro na configuração do Zabbix server:

```
### Option: HistoryStorageDateIndex
# Habilita o pré-processamento dos valores históricos no armazenamento histórico para armazenar valores em
# 0 - desabilita
# 1 - habilita
#
# Obrigatório: não
# Padrão:
# 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 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.)

9 Notas específicas da distribuição sobre a configuração do Nginx para o Zabbix

RHEL

O Nginx está disponível apenas no EPEL:

```
dnf -y install epel-release
```

SLES 15

No SUSE Linux Enterprise Server 15, você precisa configurar o `php-fpm` (o caminho para o arquivo de configuração pode variar ligeiramente 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
```

10 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 via 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 reabilita 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 para o 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 como o qual ele será executado. Nenhuma modificação adicional é necessária.

11 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 checagens ativas 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 checagens ativas 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/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 o 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/minúsculas) do host Windows como o hostname
   HostnameItem=system.hostname[shorthost,lower] - para que o agent Zabbix use apenas a parte do hostname antes do primeiro ponto, convertida para minúsculas.
   HostnameItem=system.hostname[fqdn] - para que o agent Zabbix use o Fully Qualified Domain Name como o 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, uma versão de 64 bits do agent Zabbix é necessária 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 da seguinte forma:

```
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 agent

Para iniciar o serviço do agent, você pode usar o Painel de Controle ou fazer isso pela linha de comando.

Para iniciar uma única instância do agent Zabbix com o arquivo de configuração padrão:

```
zabbix_agentd.exe --start
```

Para iniciar uma única instância do agent Zabbix com outro arquivo de configuração:

```
zabbix_agentd.exe --config <seu_arquivo_de_configuração> --start
```

Para iniciar uma das múltiplas instâncias do agent Zabbix:

```
zabbix_agentd.exe --config <arquivo_de_configuração_para_esta_instância> --start --multiple-agents
```

Parando o agent

Para parar o serviço do agent, você pode usar o Painel de Controle ou fazer isso pela linha de comando.

Para parar uma única instância do agent Zabbix iniciada com o arquivo de configuração padrão:

```
zabbix_agentd.exe --stop
```

Para parar uma única instância do agent Zabbix iniciada com outro arquivo de configuração:

```
zabbix_agentd.exe --config <seu_arquivo_de_configuração> --stop
```

Para parar uma das múltiplas instâncias do agent Zabbix:

```
zabbix_agentd.exe --config <arquivo_de_configuração_para_esta_instância> --stop --multiple-agents
```

Desinstalando o serviço do agent no Windows

Para desinstalar uma única instância do agent do Zabbix usando o arquivo de configuração padrão:

```
zabbix_agentd.exe --uninstall
```

Para desinstalar uma única instância do agent do Zabbix usando um arquivo de configuração não padrão:

```
zabbix_agentd.exe --config <seu_arquivo_de_configuração> --uninstall
```

Para desinstalar múltiplas instâncias do agent do Zabbix dos serviços do Windows:

```
zabbix_agentd.exe --config <arquivo_de_configuração_da_instância_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <arquivo_de_configuração_da_instância_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <arquivo_de_configuração_da_instância_N> --uninstall --multiple-agents
```

Limitações

O agent Zabbix para Windows não oferece suporte a 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.

12 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.

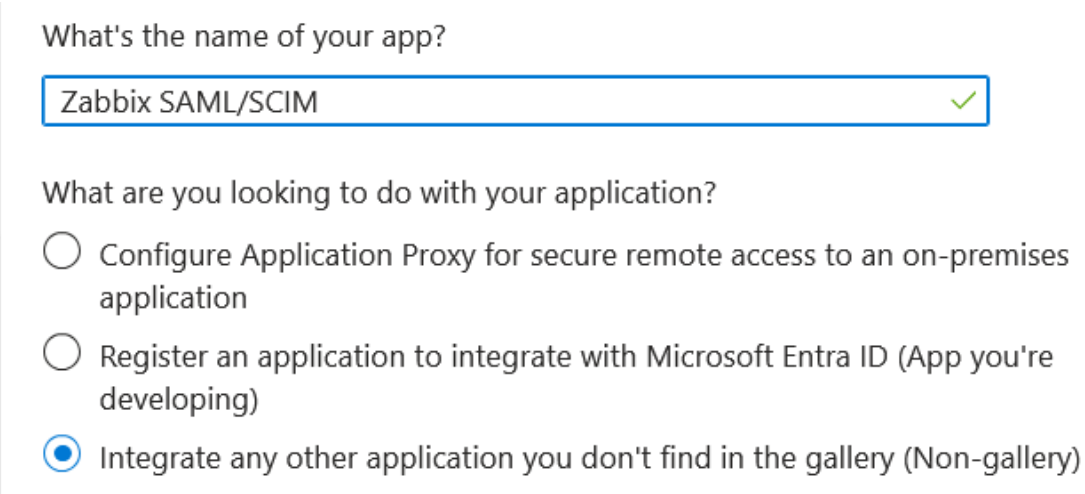
Configuração do Microsoft Entra ID

Criando 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*.

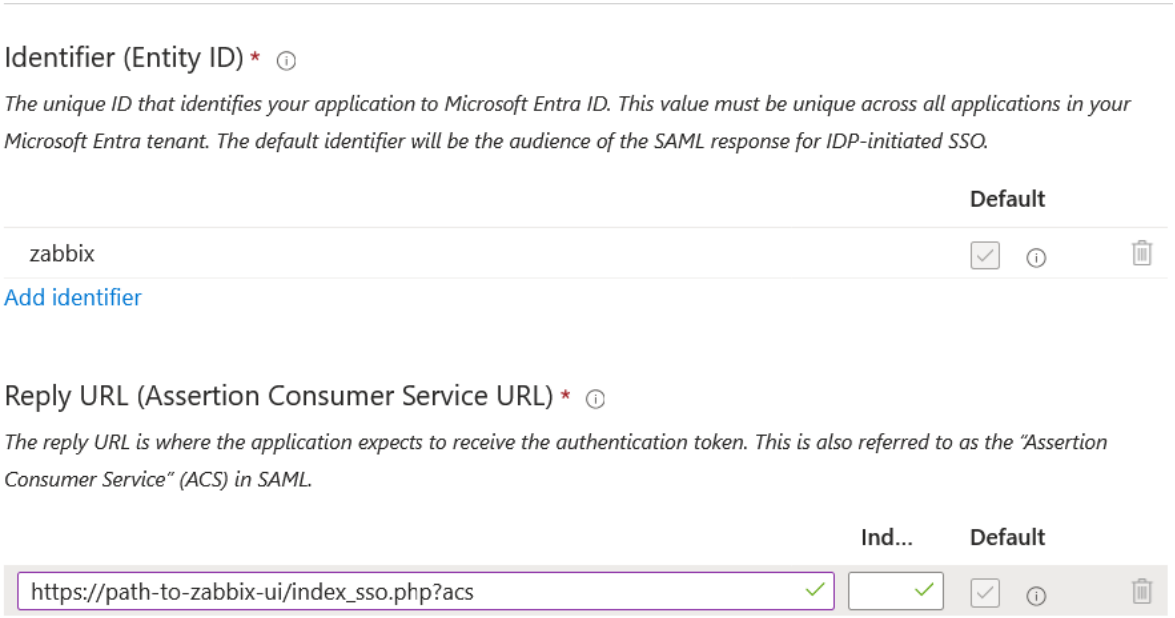


Configurando o single sign-on

1. Na página do seu aplicativo, vá para *Configurar single sign on* e clique em *Começar*. Em seguida, selecione *SAML*.

2. Edite a *Configuração SAML básica*:

- 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`:



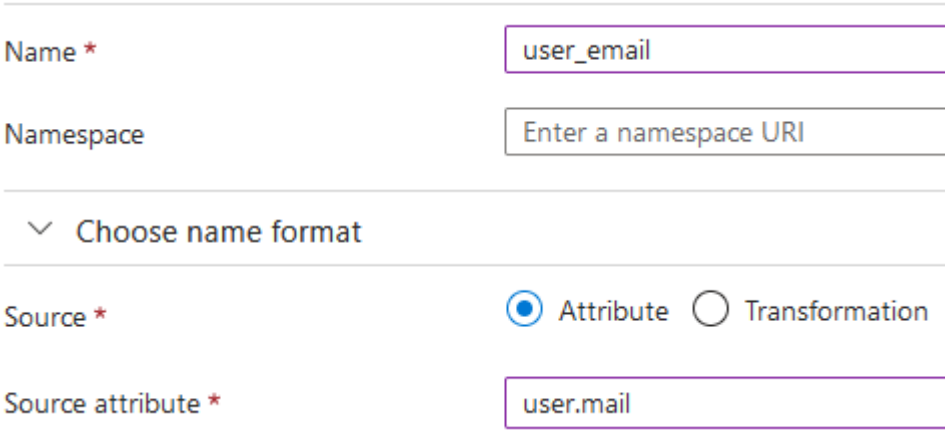
Observe que "https" é obrigatório. Para que isso funcione com o Zabbix, é necessário adicionar ao `conf/zabbix.conf.php` a seguinte linha:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

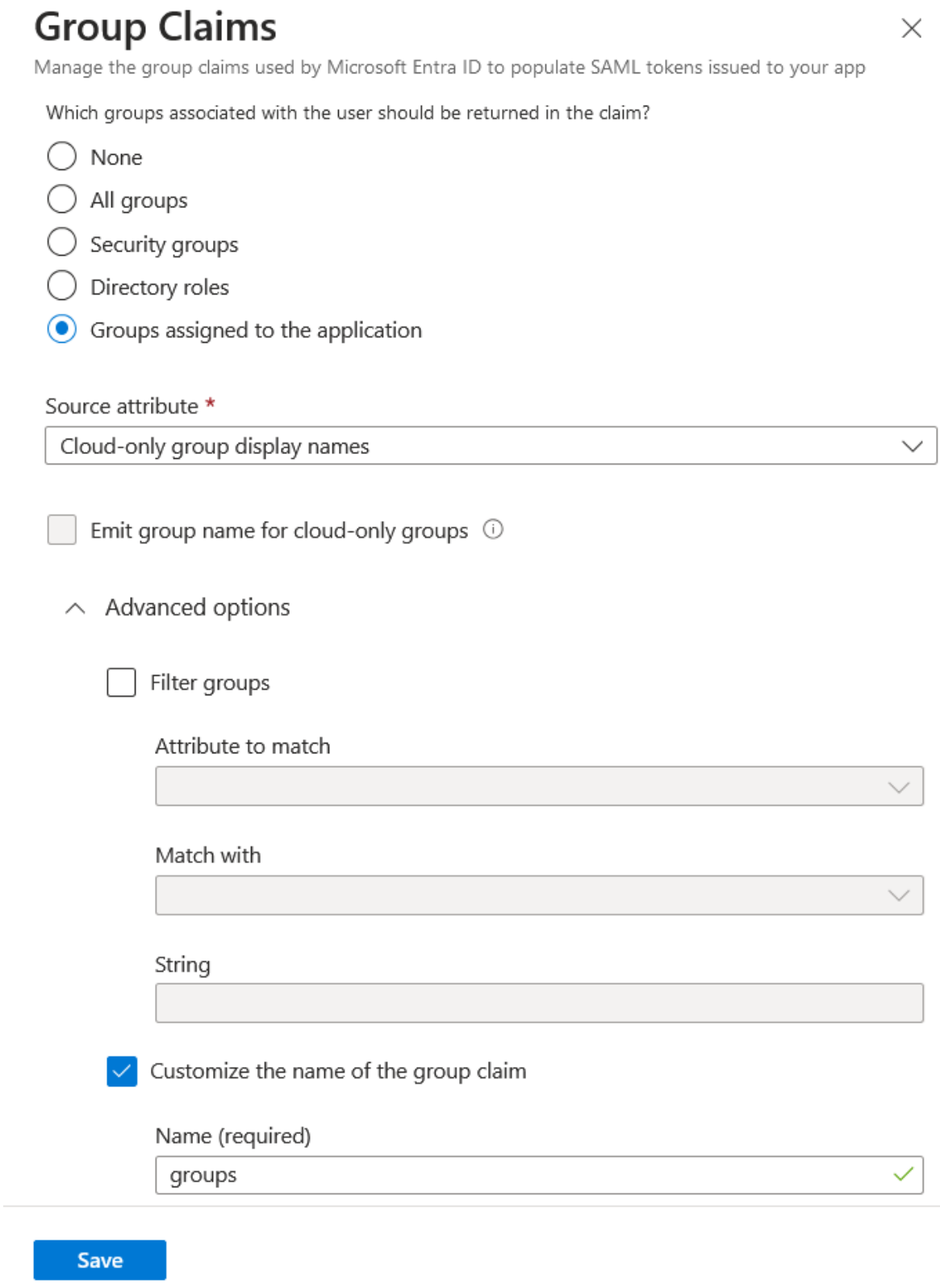
3. Edite *Atributos e declarações*. Você deve adicionar todos os atributos que deseja passar para o Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Os nomes dos atributos são arbitrários. Podem ser usados nomes de atributos diferentes, porém, é necessário que eles 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:



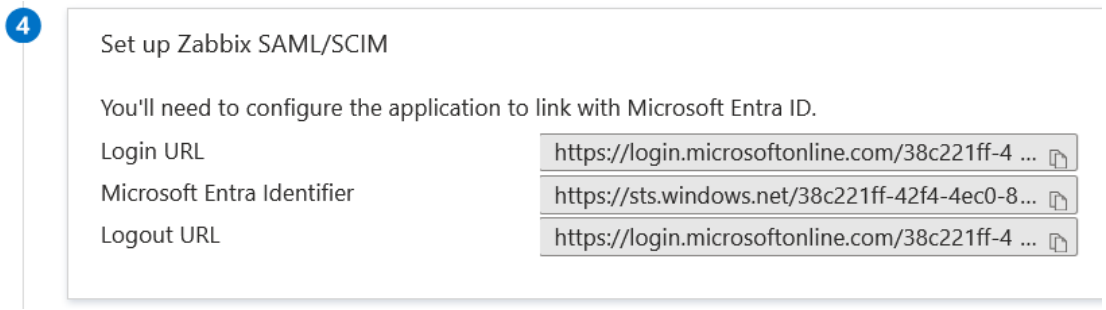
É importante nesta declaração que os nomes dos grupos (e não os IDs dos grupos) sejam passados para o Zabbix pelo *Atributo de origem* selecionado. Caso contrário, o provisionamento de usuários JIT não funcionará corretamente.

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:

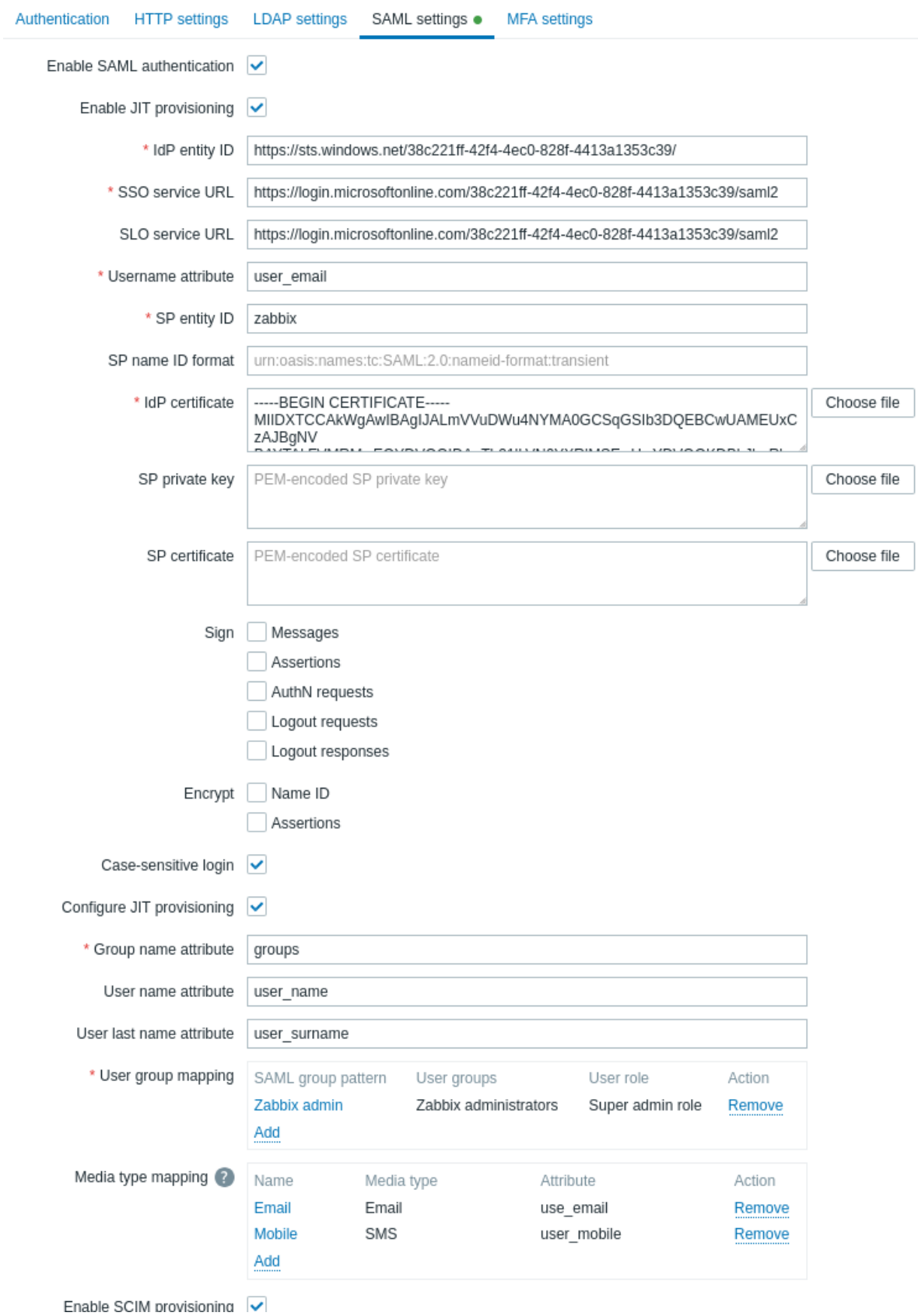
```
chmod 644 entra.cer
```

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, acesse 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`

2. Adicione o certificado Base64 fornecido pelo Entra ID.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido no zabbix.conf.php, você pode colar o texto do certificado ou fazer upload do arquivo do certificado no frontend durante a configuração do SAML — nenhum arquivo é necessário no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no zabbix.conf.php, o certificado deve estar disponível no sistema de arquivos (por padrão em ui/conf/certs ou no caminho configurado em zabbix.conf.php) e o frontend não armazenará certificados no banco de dados. Observe que se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será assumido e os certificados serão lidos de ui/conf/certs.

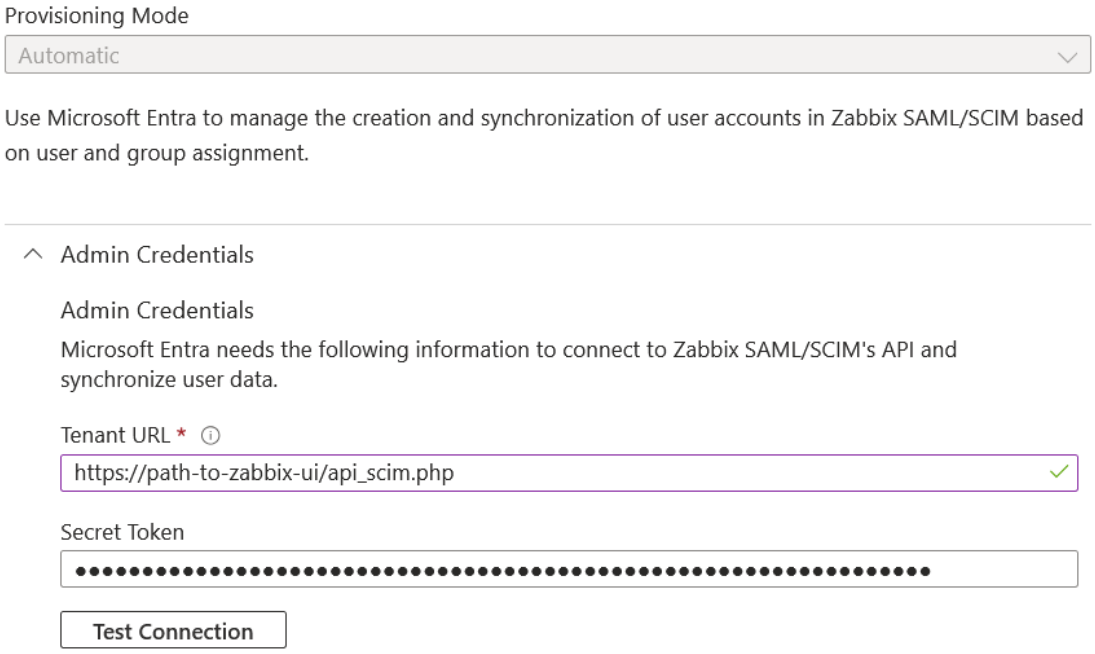
Também é necessário configurar o mapeamento de grupos de usuários. O mapeamento de mídia é opcional.

3. Pressione o botão *Atualizar* para salvar essas configurações.

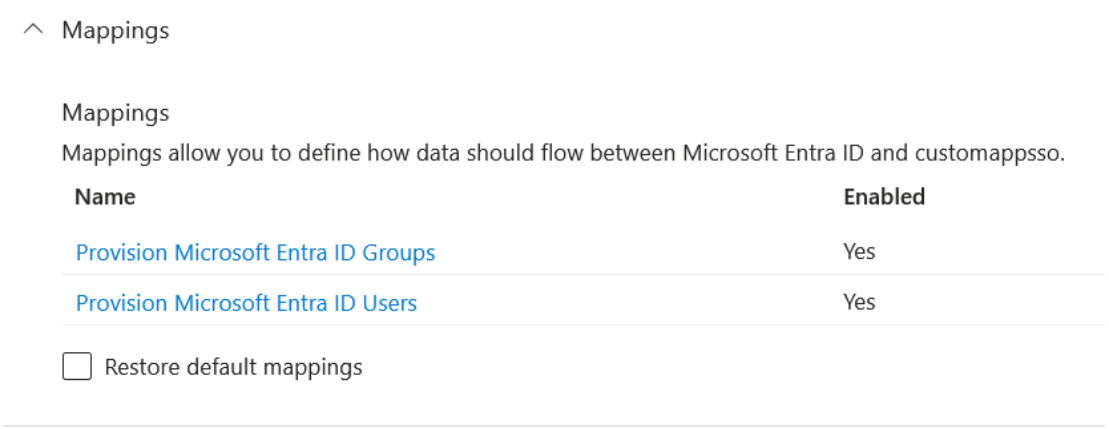
Provisionamento de usuários SCIM

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://<caminho-para-o-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.

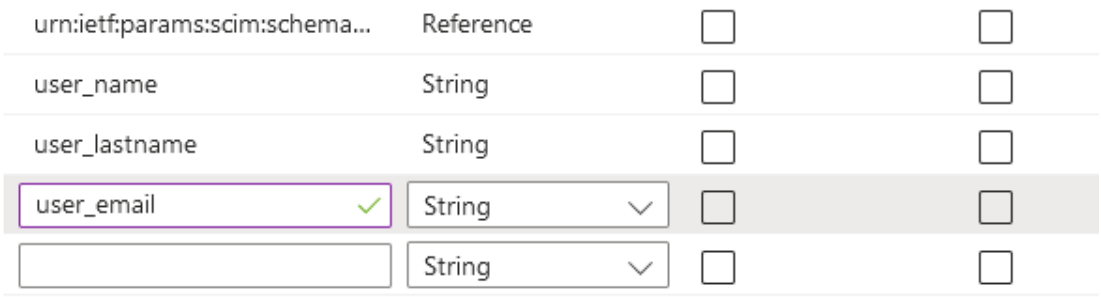


2. Agora você pode adicionar todos os atributos que serão passados com o SCIM para o Zabbix. Para isso, clique em *Mapeamentos* e depois em *Provisionar usuários do Microsoft Entra ID*.



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 ID, você pode ir ao menu *Usuários e grupos* do seu aplicativo e adicioná-los ao app.

6. Vá ao 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 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*).

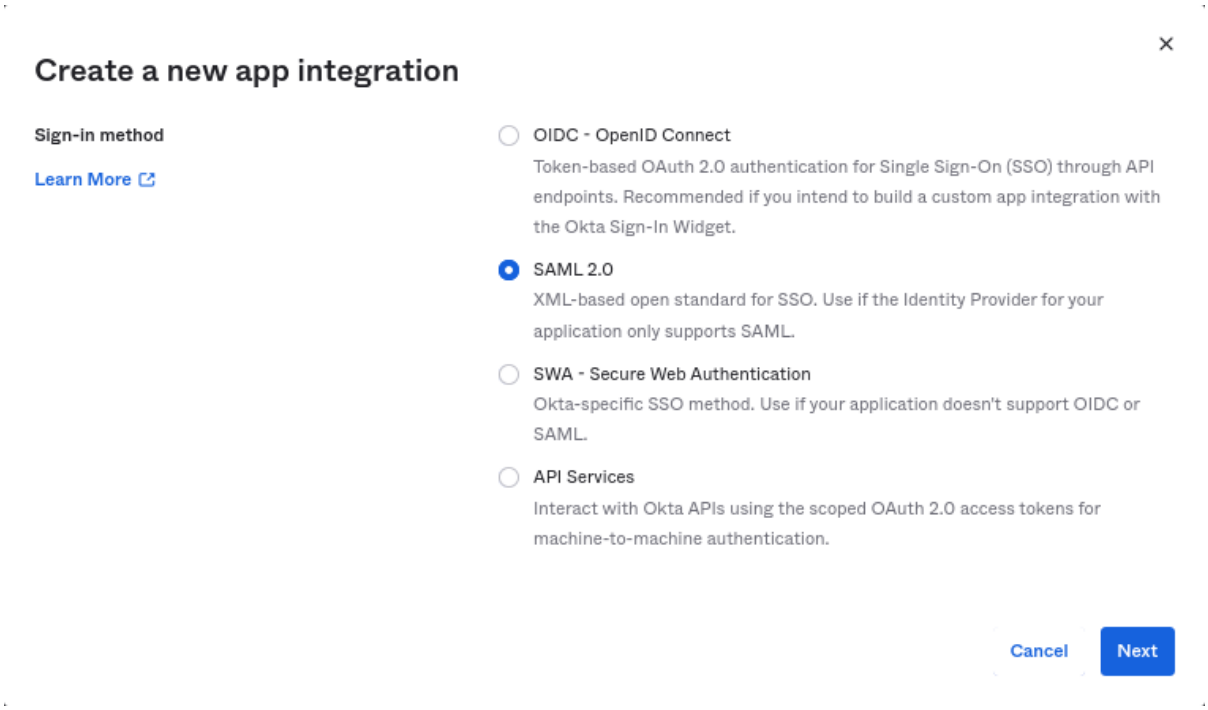
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 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.

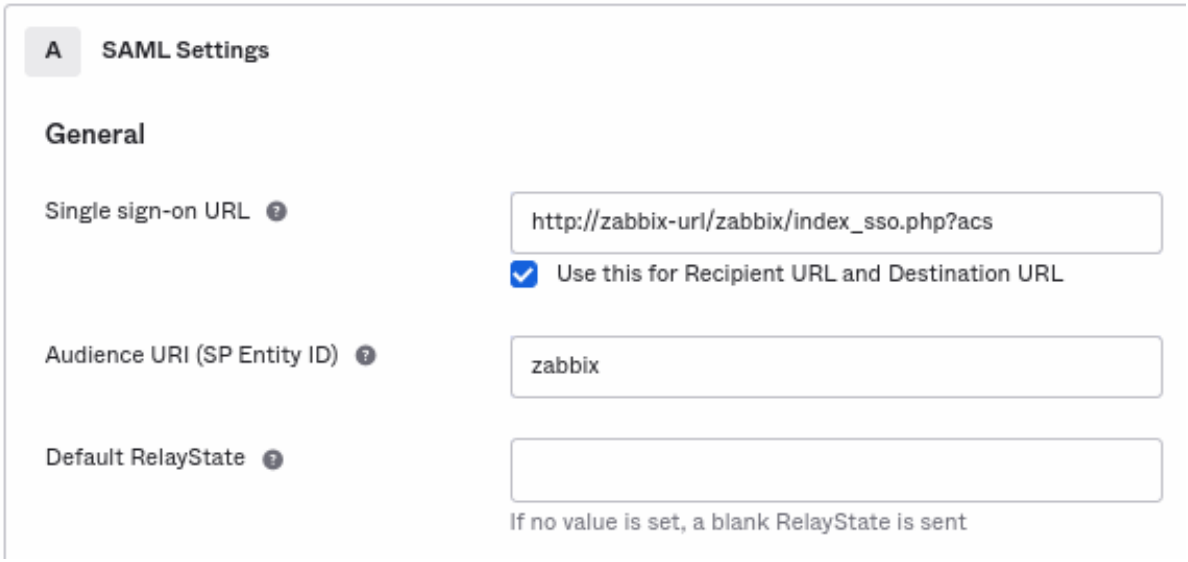
Configuração do Okta

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*.

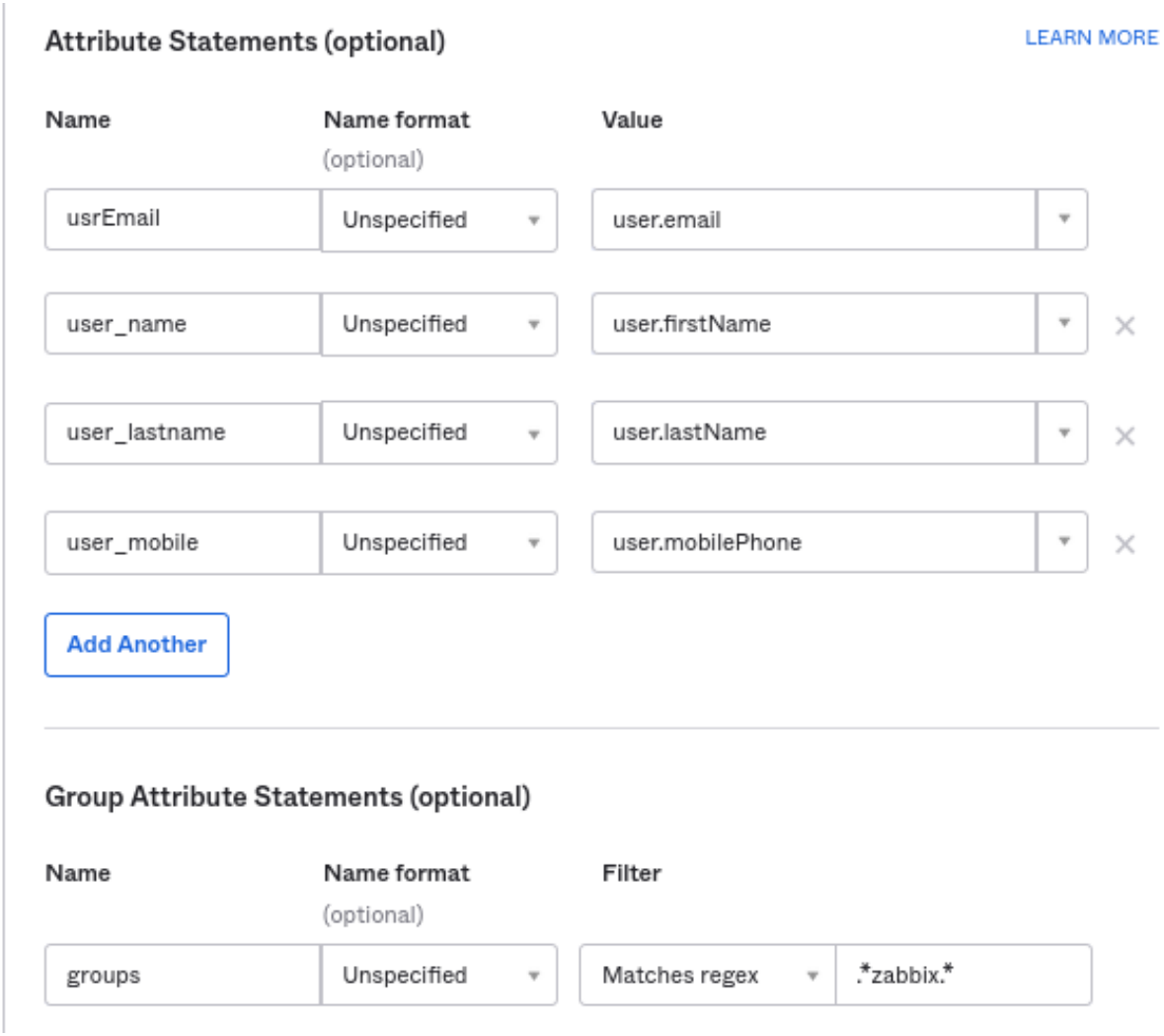


Selecione "SAML 2.0" como método de login e clique em *Next*.

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://<seu-zabbix-url>/zabbix/index_sso.php?acs`
    Observe o uso de "http", e não "https", para que o parâmetro `acs` não seja removido da requisição. A caixa *Use this for Recipient URL and Destination URL* também deve ser marcada.
  - *Audience URI (SP Entity ID)*: `zabbix`
    Observe que este valor será usado dentro da asserção SAML como um identificador único 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 nas configurações *Usuários* > *Usuários*.
  - 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.

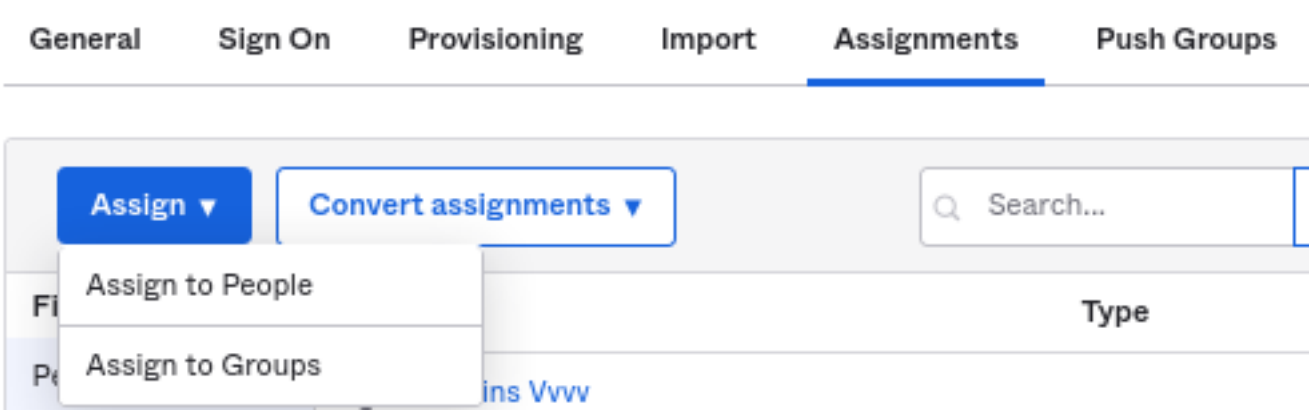
Os nomes dos atributos usados aqui são exemplos arbitrários. Você pode usar nomes de atributos diferentes, no entanto, é necessário que eles correspondam ao valor do campo respectivo nas configurações SAML do Zabbix.

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 upload do certificado público no 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 clique em *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 popup que aparece, atribua o aplicativo às pessoas que usarão o SAML 2.0 para autenticar no Zabbix e clique em *Save and go back*.

9. Baixe o certificado IdP. Defina as permissões 644 para ele executando:

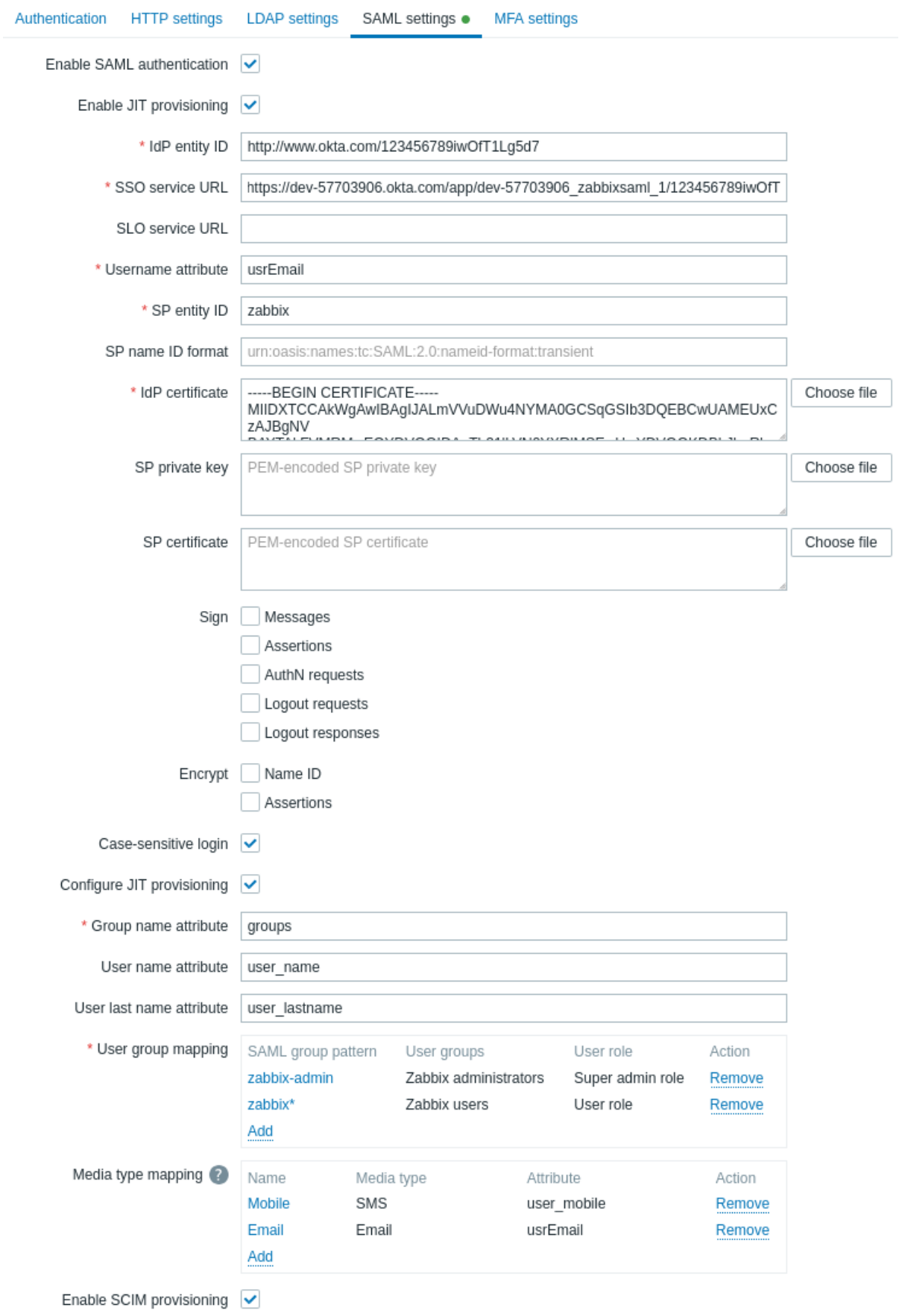
```
chmod 644 idp.crt
```

10. 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. Adicione o certificado Base64 fornecido nas instruções de configuração SAML do Okta.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido no zabbix.conf.php, você pode colar o texto do certificado ou fazer upload do arquivo do certificado no frontend durante a configuração do SAML — nenhum arquivo é necessário no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no zabbix.conf.php, o certificado deve estar disponível no sistema de arquivos (por padrão em ui/conf/certs ou no caminho configurado no zabbix.conf.php) e o frontend não armazenará certificados no banco de dados. Observe que se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será assumido e os certificados serão lidos de ui/conf/certs.

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.

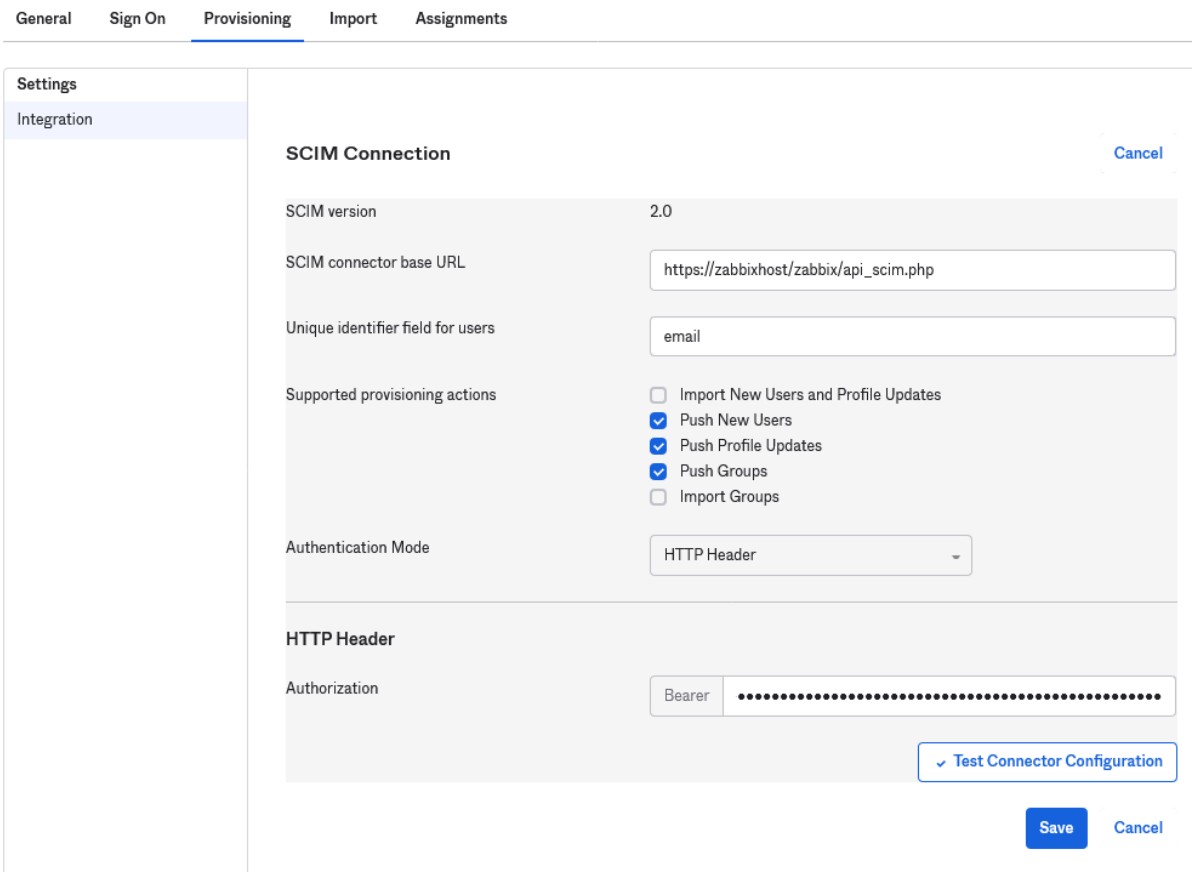
Provisionamento SCIM

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.

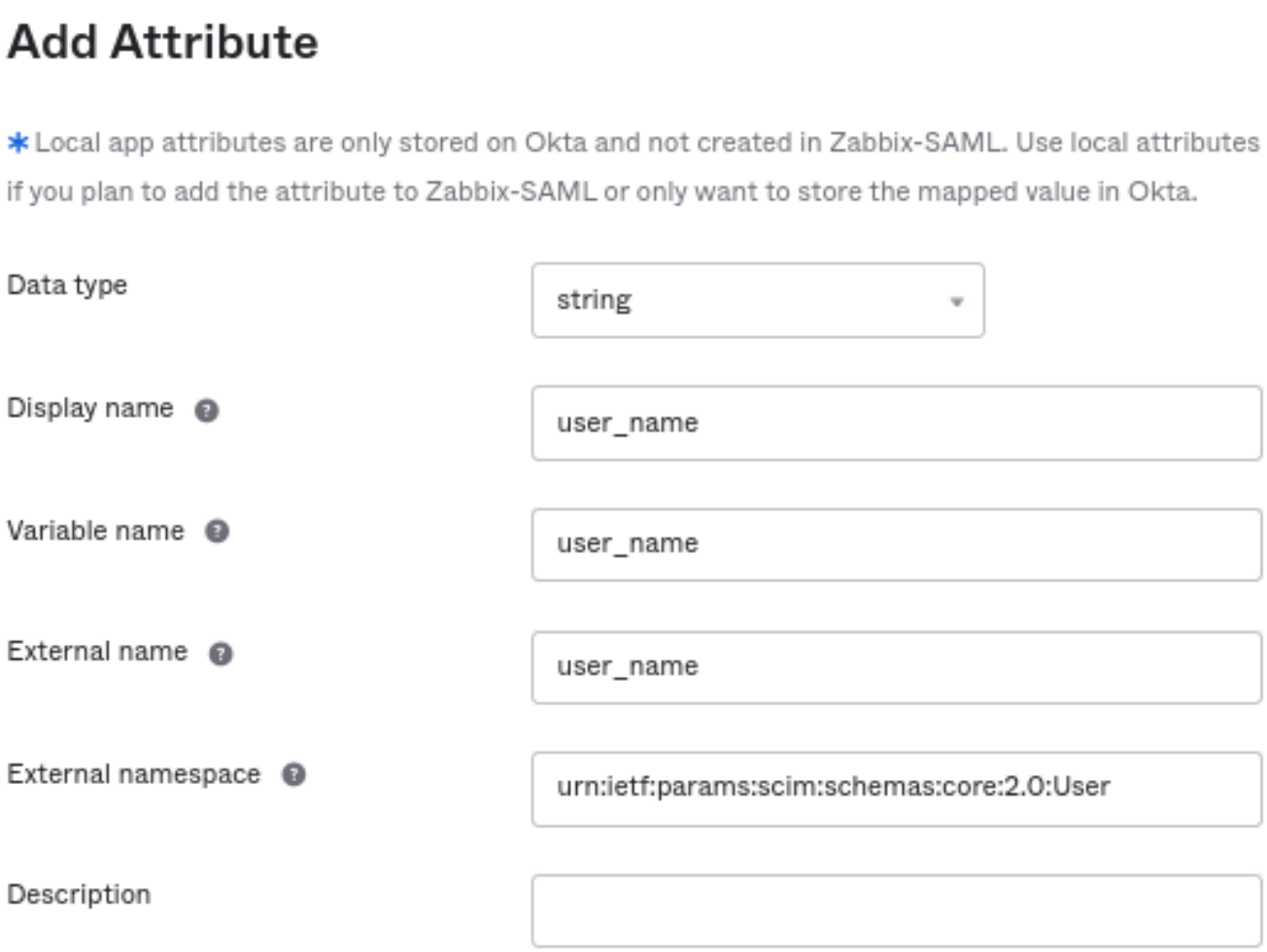
4. Em ”Provisioning” -> ”To App”, certifique-se de marcar as seguintes caixas:

- 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*.

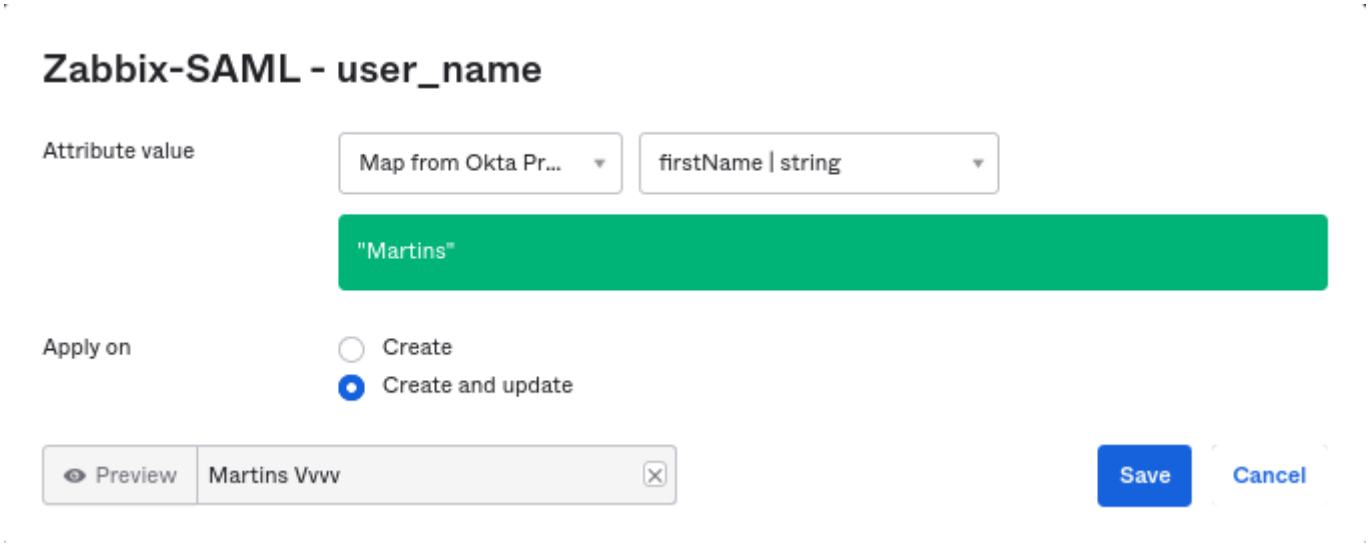
Clique em *Add Attribute*. Preencha os valores para *Display name*, *Variable name*, *External name* com o nome do atributo SAML, por exemplo, `user_name`.



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.

14 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)*.

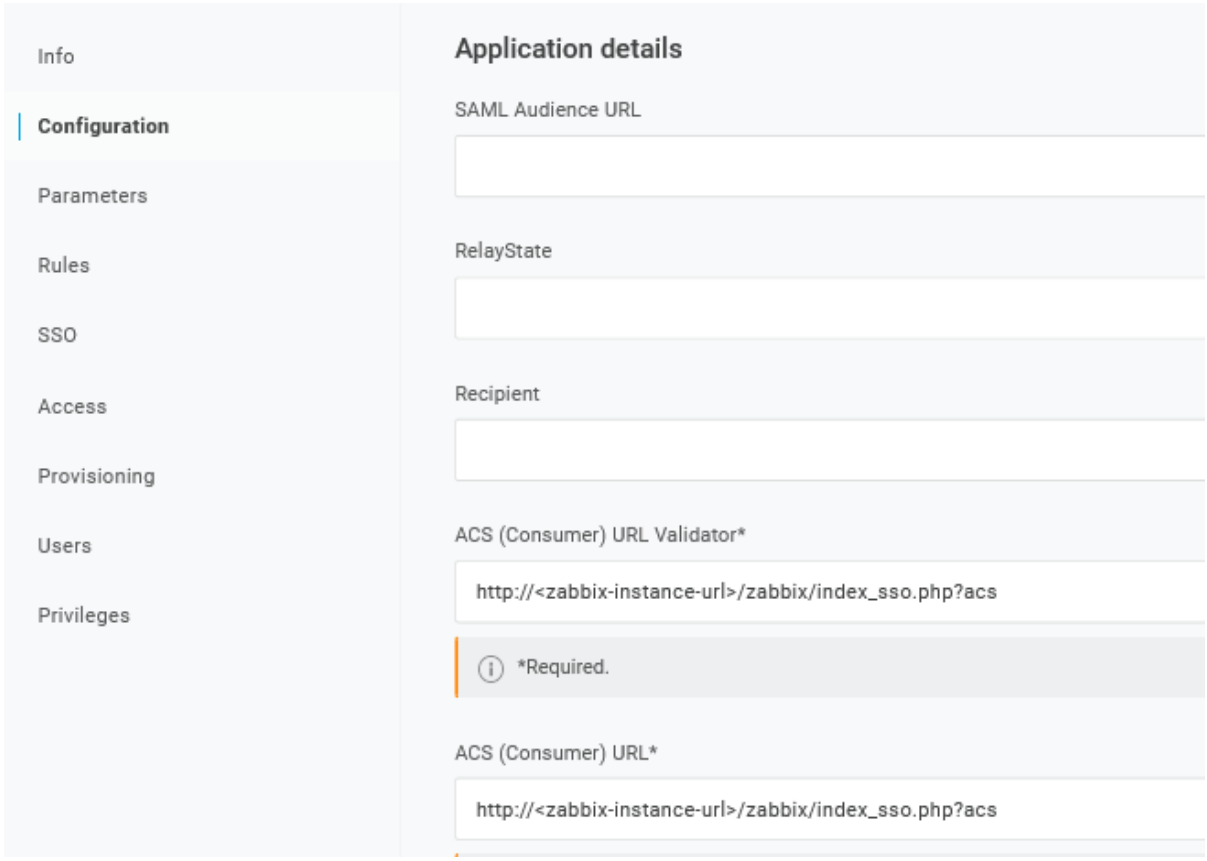
4. Para começar, você pode querer personalizar o nome de exibição do seu aplicativo. Você também pode adicionar o ícone e os detalhes do aplicativo. Depois disso, clique em *Save*.

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 destes 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`:

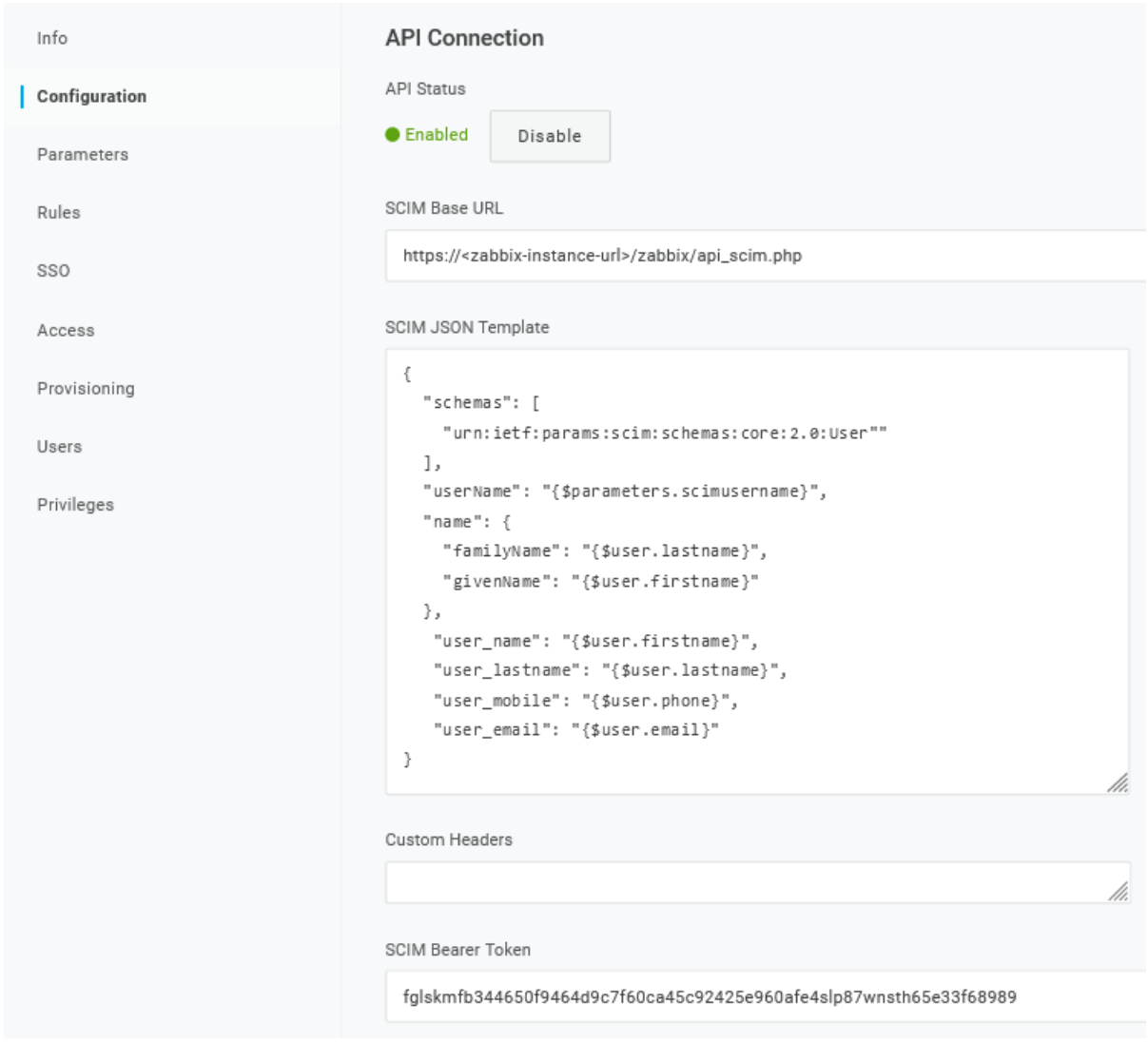
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$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.

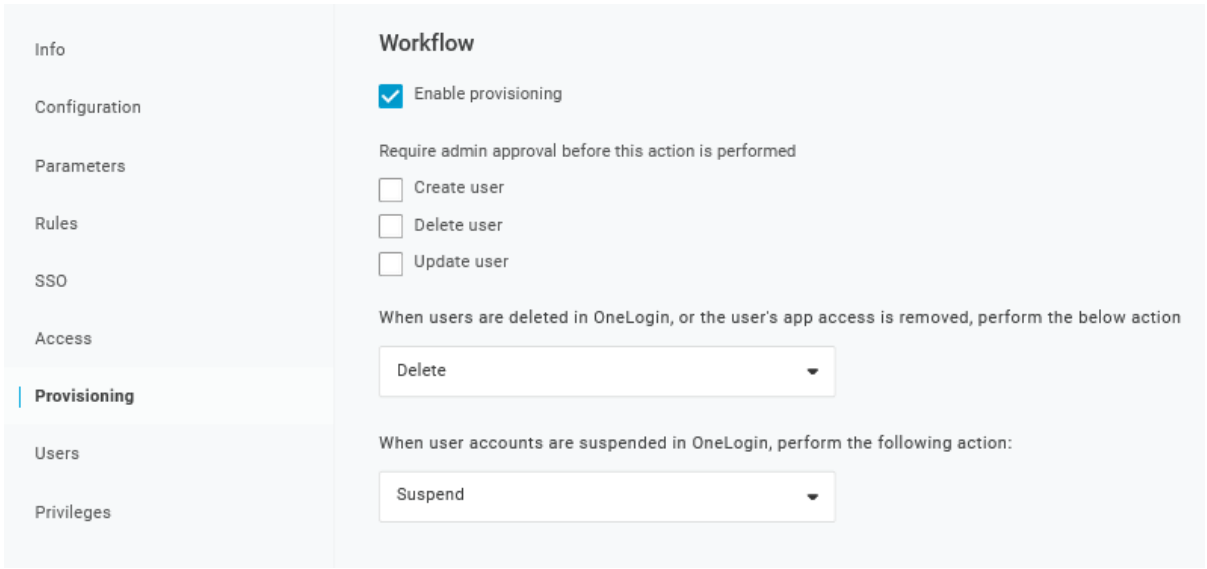
Observe que, para o provisionamento de usuários funcionar, o OneLogin precisa receber em resposta um atributo 'name' com 'givenName' e 'familyName', mesmo que não seja exigido pelo provedor de serviço. Assim, é necessário especificar isso no schema na parte de configuração do aplicativo.

- *SCIM Bearer Token*: insira um token de API do Zabbix com permissões de Super admin.

Clique em *Enable* para ativar a conexão.

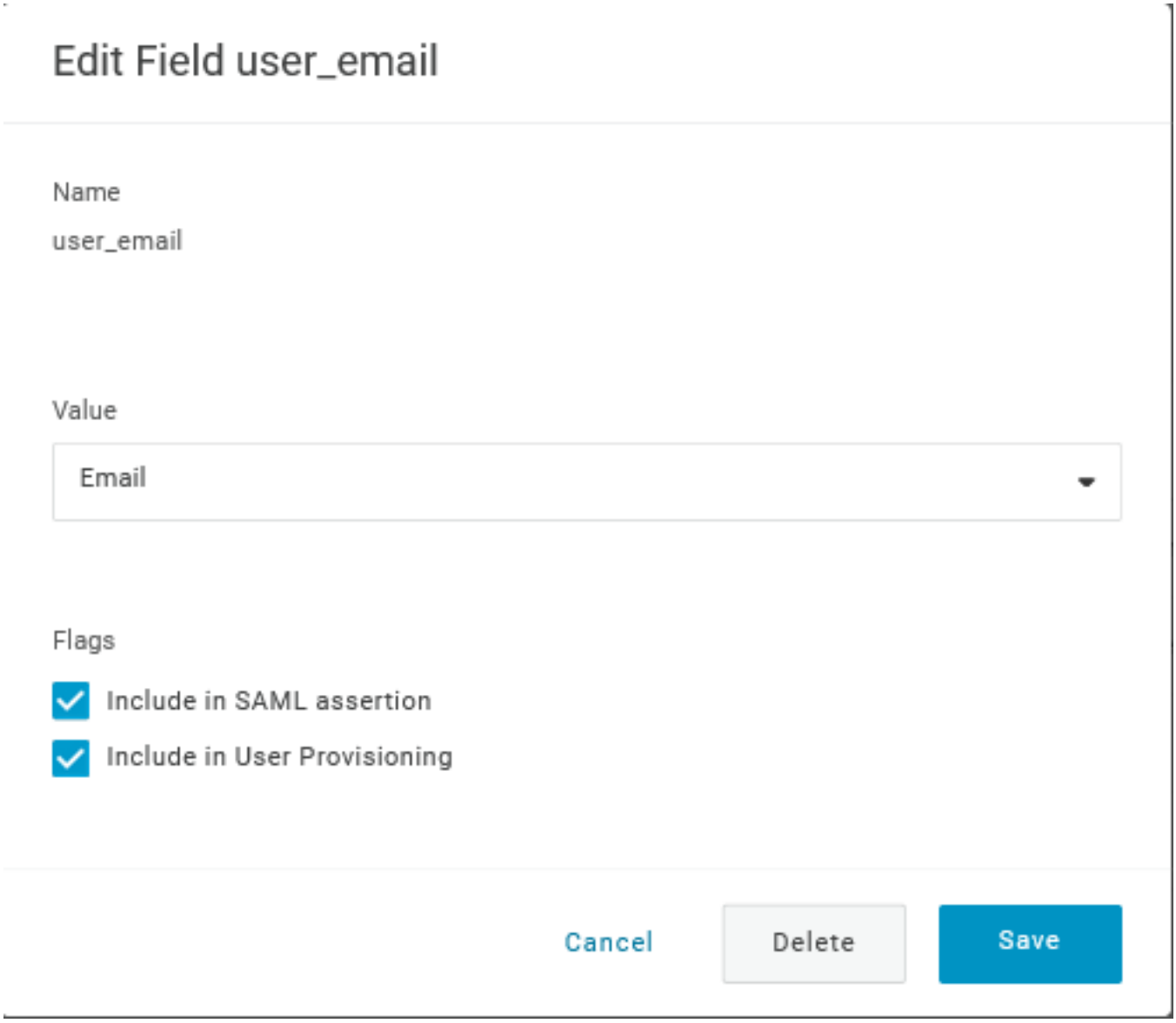


3. Na página *Provisioning*, habilite a opção Provisioning:



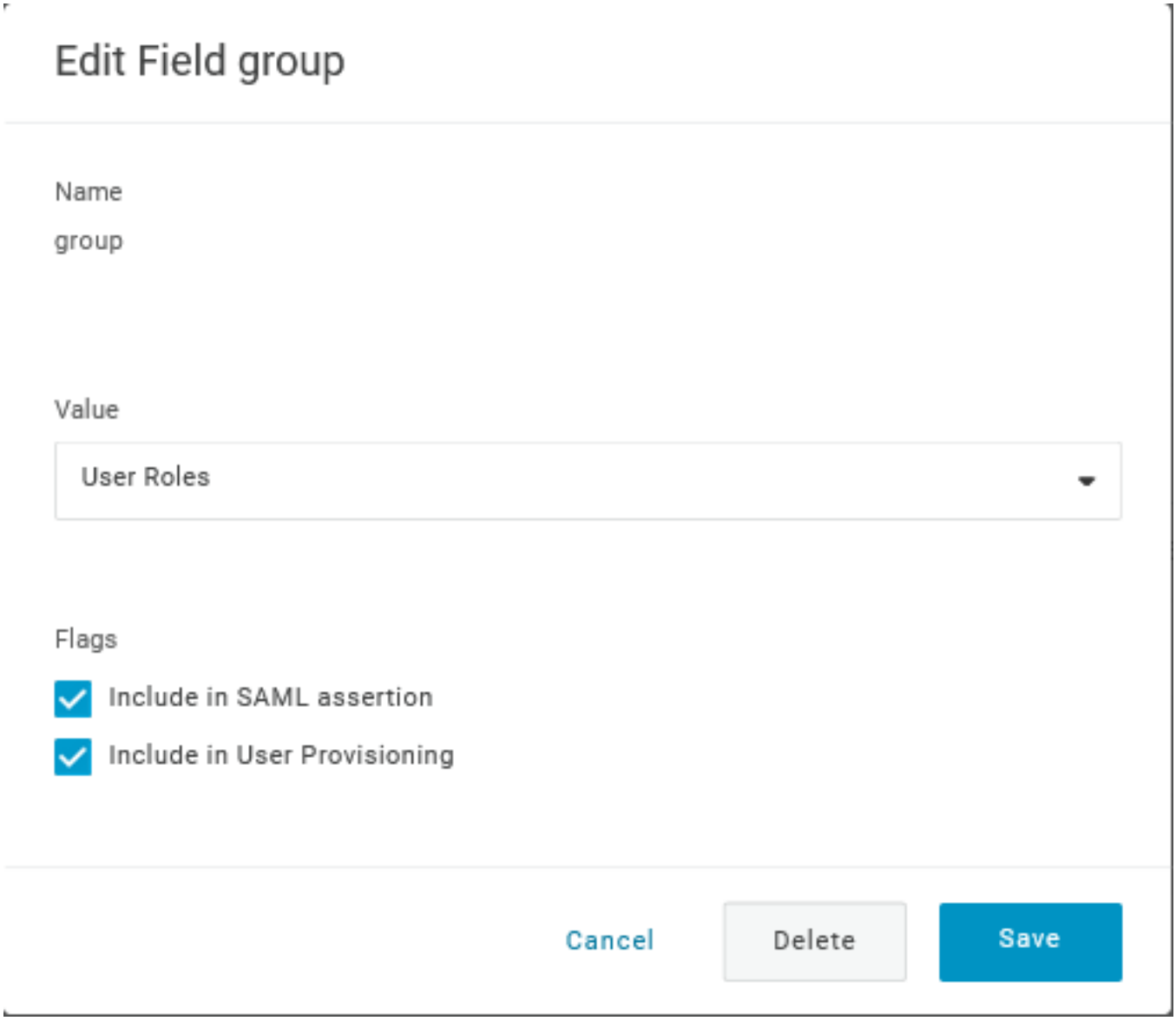
4. A página *Parameters* contém uma lista de parâmetros padrão:

- 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`:

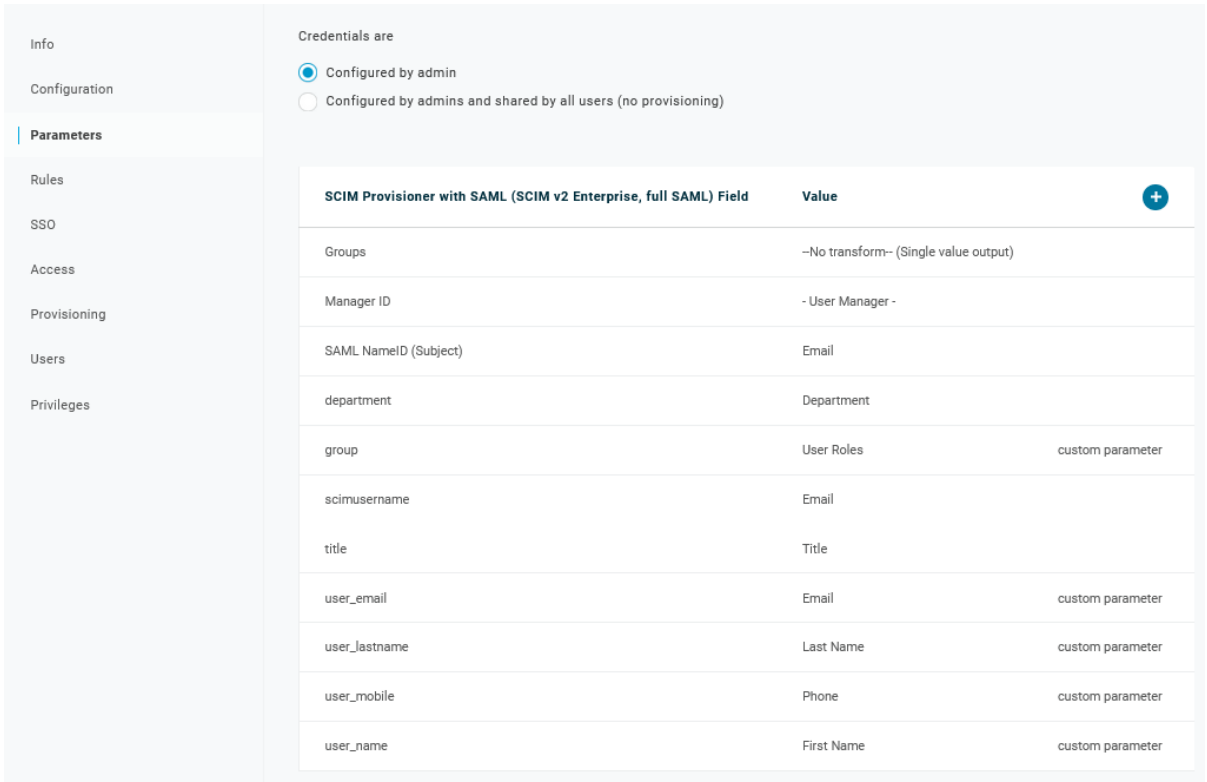


Ao adicionar um parâmetro, certifique-se de marcar as opções *Include in SAML assertion* e *Include in User Provisioning*.

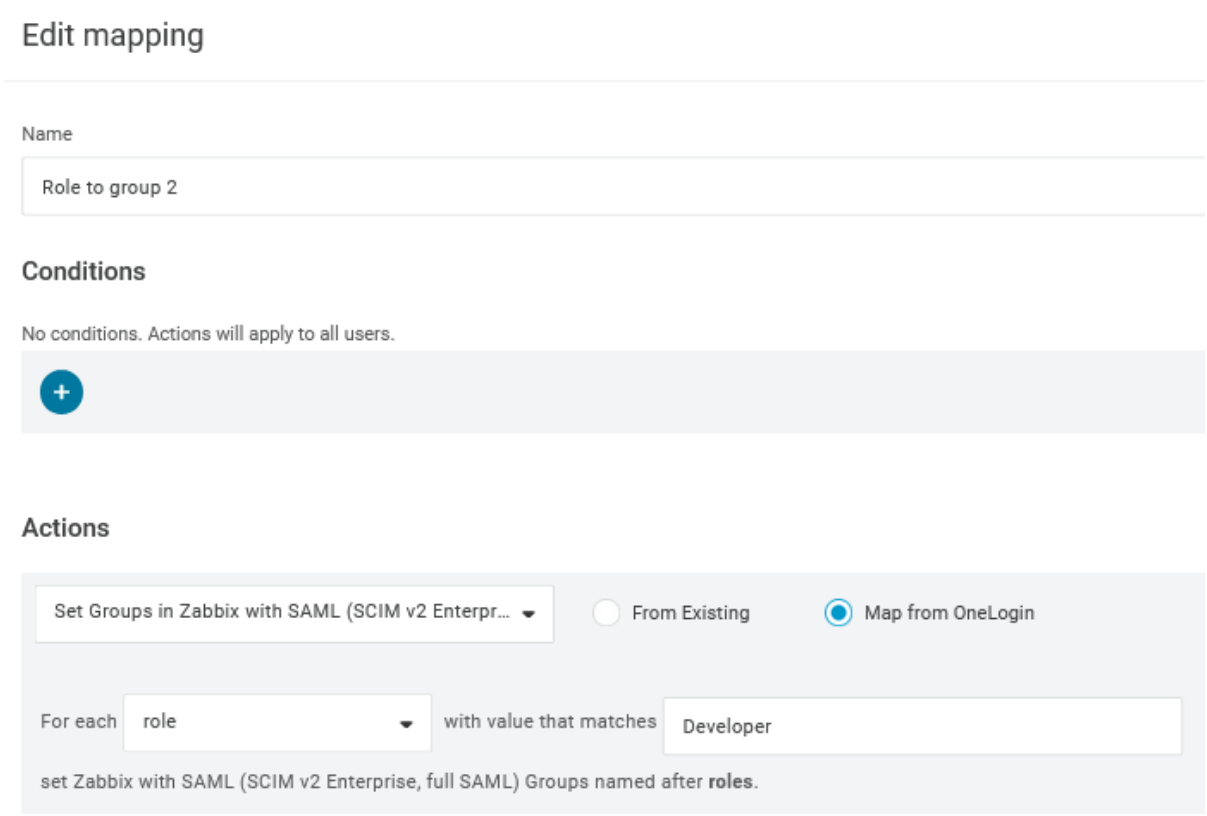
- Adicione um parâmetro 'group' que corresponda aos papéis de usuário no OneLogin. Os papéis de usuário serão passados como uma string, separados por ponto e vírgula `;`. Os papéis de usuário do OneLogin serão usados para criar grupos de usuários no Zabbix:



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.

6. Baixe o certificado IdP. Defina as permissões 644 para ele executando:

```
chmod 644 idp.crt
```

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. Adicione o certificado Base64 fornecido pelo OneLogin.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido em zabbix.conf.php, você pode colar o texto do certificado ou fazer upload do arquivo do certificado no frontend durante a configuração do SAML — nenhum arquivo é necessário no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido em zabbix.conf.php, o certificado deve estar disponível no sistema de arquivos (por padrão em ui/conf/certs ou no caminho configurado em zabbix.conf.php) e o frontend não armazenará certificados no banco de dados. Observe que se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo é assumido e os certificados são lidos de ui/conf/certs.

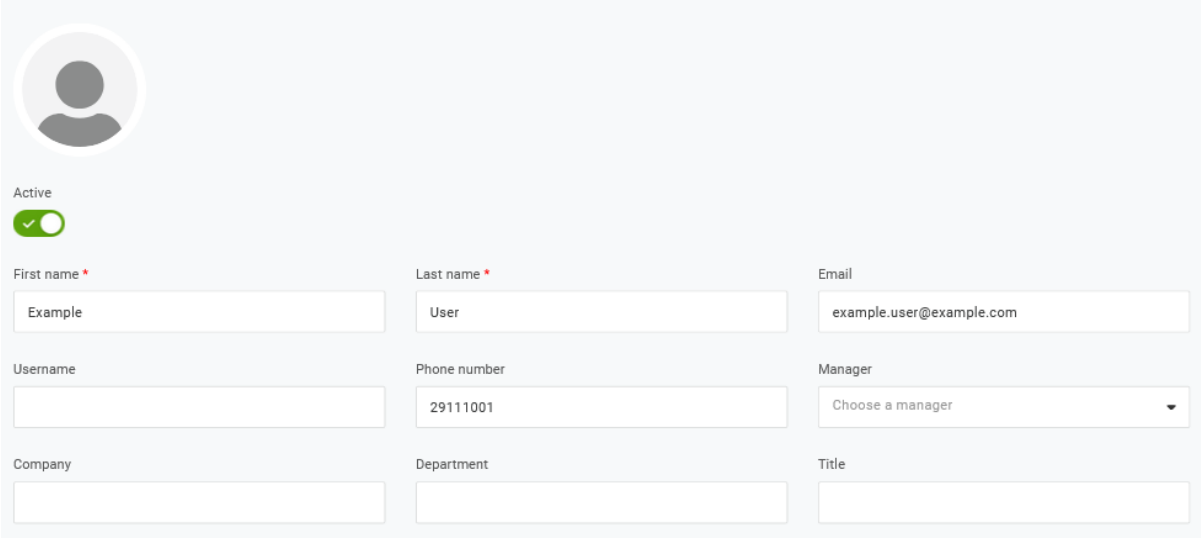
Você pode acessar o download do certificado no OneLogin em *Applications* -> *SSO* -> clique em *View details* abaixo do certificado atual.

3. Pressione o botão *Atualizar* para salvar essas configurações.

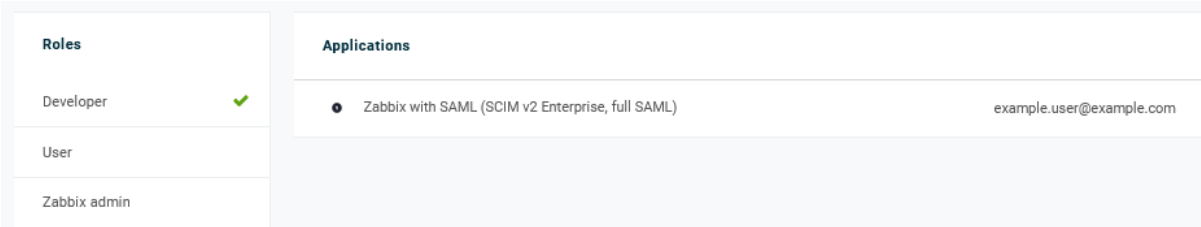
Provisionamento de usuários SCIM

Com o provisionamento de usuários ativado, agora é possível adicionar/atualizar usuários e seus papéis no OneLogin e tê-los imediatamente provisionados no Zabbix.

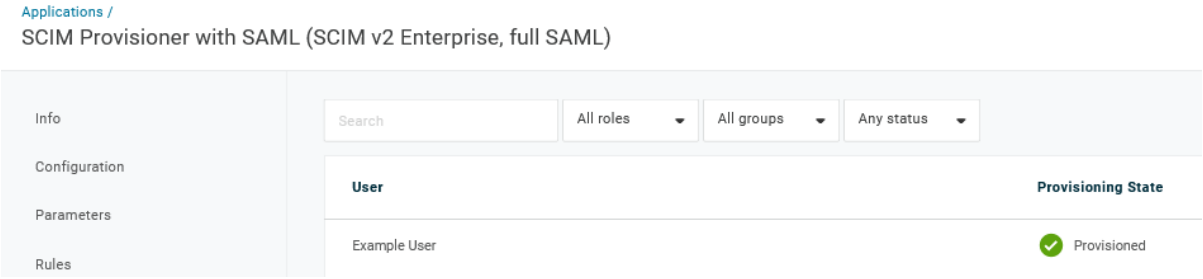
Por exemplo, você pode criar um novo usuário:



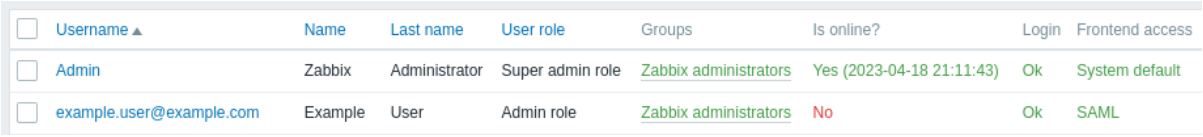
Adicione-o a um papel de usuário e ao aplicativo que irá provisionar o usuário:



Ao salvar o usuário, ele será provisionado para o 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

15 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 de 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 servidor Zabbix 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 `<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 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 o menu do frontend *Administração → Geral → Outros parâmetros*
- Especifique a URL completa da interface web do Zabbix no parâmetro *Frontend URL*

frontend_url.png

Note:
Após concluir o procedimento de configuração, você pode configurar e enviar um relatório de teste para garantir que tudo está funcionando corretamente.

16 Idiomas adicionais do frontend

Visão geral

Para usar qualquer outro idioma além do inglês na interface web do Zabbix, o locale correspondente deve estar instalado no servidor web. Além disso, a extensão PHP gettext é necessária para que as traduções funcionem.

Instalando locais

Para listar todos os idiomas instalados, execute:

```
locale -a
```

Se alguns idiomas necessários não estiverem listados, abra o arquivo */etc/locale.gen* e descomente os locais necessários. Como o Zabbix usa codificação UTF-8, você precisa selecionar locais com o charset UTF-8.

Agora execute:

```
locale-gen
```

Reinicie o servidor web.

Os locais agora devem estar instalados. Pode ser necessário recarregar a página do frontend do Zabbix no navegador usando Ctrl + F5 para que os novos idiomas apareçam.

Instalando o Zabbix

Se estiver instalando o Zabbix diretamente do repositório git do Zabbix, os arquivos de tradução devem ser gerados manualmente. Para gerar os arquivos de tradução, execute:

```
make gettext
locale/make_mo.sh
```

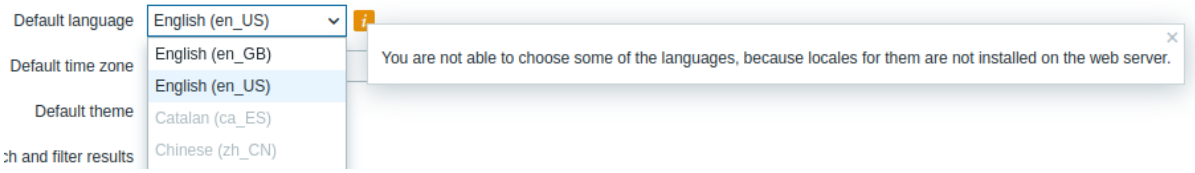
Esta etapa não é necessária ao instalar o Zabbix a partir de pacotes ou arquivos tar.gz do código-fonte.

Selecionando um idioma

Existem várias maneiras de selecionar um idioma na interface web do Zabbix:

- Durante a instalação da 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.
- 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 de processos

Use a barra lateral para acessar o conteúdo desta seção.

1 Zabbix server

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 frontend do Zabbix 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 irá atualizar o 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 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).
ExternalScripts	O local dos scripts externos.
Fping6Location	O local do fping6.
FpingLocation	O local do fping.
FrontendAllowedIP	Uma lista separada por vírgulas de endereços IP ou intervalos CIDR que têm permissão para se conectar a partir do frontend.
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.

Parâmetro	Descrição
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 na qual 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 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 de 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 para 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 irá sincronizar 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 SNMP trap 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-iniciadas de agent pollers assíncronos do Zabbix.
StartAlerters	O número de instâncias pré-iniciadas de alerters.
StartBrowserPollers	O número de instâncias pré-iniciadas de browser item pollers.
StartConnectors	O número de instâncias pré-iniciadas de workers de conector.
StartDBSyncers	O número de instâncias pré-iniciadas de history syncers.
StartDiscoverers	O número de instâncias pré-iniciadas de discovery workers.
StartEscalators	O número de instâncias pré-iniciadas de escalators.
StartHistoryPollers	O número de instâncias pré-iniciadas de history pollers.
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.
StartLLDProcessors	O número de instâncias pré-iniciadas de workers de descoberta de baixo nível (LLD).
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 workers de pré-processamento.
StartProxyPollers	O número de instâncias pré-iniciadas de pollers para proxies passivos.
StartReportWriters	O número de instâncias pré-iniciadas de report writers.
StartSNMPPollers	O número de instâncias pré-iniciadas de SNMP pollers assíncronos.
StartSNMPTrapper	Se definido como 1, um processo SNMP trapper será iniciado.
StartTimers	O número de instâncias pré-iniciadas de timers.

Parâmetro	Descrição
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 em 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]`) e item `icmpping[*]`.
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.
TLSFrontendAccept	Quais conexões de entrada aceitar do frontend.
TLSFrontendCertIssuer	Emissor de certificado do frontend permitido.
TLSFrontendCertSubject	Assunto do certificado do frontend permitido.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do server, usado para comunicações criptografadas entre componentes do Zabbix.
TLSListen	Controla o TLS no socket do trapper.
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	Reduz 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.
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.

Parâmetro	Descrição
WebDriverURL	URL HTTP[S] da interface WebDriver.
WebServiceURL	URL HTTP[S] para o serviço web do Zabbix 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;
- Os valores suportam variáveis de ambiente;
- 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

Permitir 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

Permitir que o server trabalhe com versões de banco de dados não suportadas.

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, apenas string vazia resulta em tentativa de usar socket.

Padrão: `localhost`

DBName

O nome do banco de dados.

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 local.[3]

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 esta opção para os seguintes valores força o uso de uma conexão TLS com o banco de dados:
required - conectar usando TLS
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 especificada por DBHost corresponde ao seu certificado

Com MySQL, a partir da versão 5.7.11, e PostgreSQL, os seguintes valores são suportados: `required`, `verify_ca`, `verify_full`.
Com MariaDB, a partir da versão 10.2.6, os valores `required` e `verify_full` são suportados.
Por padrão, não é definido para nenhuma opção e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação de certificado de banco de dados.

Obrigatório: não (sim, se DBTLSConnect definido como *verify_ca* ou *verify_full*)

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 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

Habilitar 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, habilita 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.
Nota que 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 apenas para exportação de events:

```
ExportType=events
```

ExternalScripts

O local dos scripts externos (depende da variável de instalação `datadir` definida em tempo de compilação).

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 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 de root e o sinalizador SUID definido.

Padrão: `/usr/sbin/fping`

FrontendAllowedIP

Uma lista separada por vírgulas de endereços IP ou intervalos CIDR que têm permissão para se conectar a partir do frontend. A conexão do frontend será aceita apenas dos endereços listados aqui se este parâmetro for definido. Por padrão, todas as conexões são aceitas para solicitações do frontend. 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:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

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-16G

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 em cache.

Padrão: 4M
 Intervalo: 128K-16G

HistoryStorageDateIndex

Habilitar o pré-processamento dos valores do histórico no armazenamento do histórico para armazenar valores em índices diferentes com base na data.

Padrão: 0
 Valores: 0 - desabilitar; 1 - habilitar

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 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. Só é necessário se os Java pollers forem iniciados.

JavaGatewayPort

A porta que o Zabbix Java gateway escuta.

Padrão: 10052
 Intervalo: 1024-32767

ListenBacklog

O número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante definida no código, 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 separados por vírgulas nos quais o trapper deve escutar.
O trapper irá escutar em todas as interfaces de rede se este parâmetro estiver ausente.

Padrão: 0.0.0.0

ListenPort

A porta de escuta para 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 log.

Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo do arquivo de log em MB.
0 - desabilita a rotação automática do 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 é truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024
 Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogSlowQueries

Determina quanto tempo uma consulta ao banco de dados pode levar antes de ser registrada em milissegundos.
0 - não registra consultas lentas.
Esta opção é ativada a partir de DebugLevel=3.

Padrão: 0
 Intervalo: 0-3600000

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`

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 histórico e tendências de itens 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: <endereço>[:<porta>]

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 sobrescrita 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 com que frequência o Zabbix irá excluir problemas para triggers excluídos, em segundos.

Padrão: 60
 Intervalo: 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 de histórico de um Zabbix proxy em segundos. Usado apenas para proxies no modo passivo.

Padrão: 1
 Intervalo: 1-3600

ServiceManagerSyncFrequency

Determina com que frequência o Zabbix irá sincronizar a configuração de um gerenciador de serviços em segundos.

Padrão: 60
 Intervalo: 1-3600

SMSDevices

Uma lista de arquivos de modem separados por vírgulas permitidos para uso pelo Zabbix server.
O envio de SMS não é possível se este parâmetro não for definido.

Exemplo:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Arquivo temporário usado para passar dados do daemon SNMP trap para o server.
Deve ser o mesmo que está em zabbix_trap_receiver.pl ou no arquivo de configuração do SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

Diretório para armazenar sockets IPC usados pelos serviços internos do Zabbix.

Padrão: `/tmp`

SourceIP

Endereço IP de origem para:

- conexões de saída para o Zabbix proxy e Zabbix agent;
- 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 SSH.

SSLCertLocation

Localização dos arquivos de certificado de cliente SSL para autenticação de cliente.
Este parâmetro é usado apenas no monitoramento web.

SSLKeyLocation

Localização dos arquivos de chave privada SSL para autenticação do cliente.
Este parâmetro é usado apenas no monitoramento 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 do 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. Aproximadamente, 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-forked 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: 0-100

StartODBCPollers

O número de instâncias pré-forked de ODBC pollers[1].

Padrão: 1
 Intervalo: 0-1000

StartPingers

O número de instâncias pré-forked 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[1] 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
 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 desativada.
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-criadas de timers.
Timers processam períodos de manutenção.

Padrão: 1
 Intervalo: 1-1000

StartTrappers

O número de instâncias pré-forked de trappers[1].
Trappers aceitam conexões recebidas do Zabbix sender, agents ativos e proxies ativos.

Padrão: 5
 Intervalo: 1-1000

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 proxy, agent, serviço web, e para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`) e item `icmpping[*]`.
 Este parâmetro define a duração para várias operações de comunicação:

- execução de comandos remotos no Zabbix agent;
- execução de comandos SSH / Telnet;
- requisições ao Zabbix Web Service;
- timeout de comunicação para requisição de teste de tipo de mídia e método `history.push`;
- reagendamento de itens quando a interface IPMI se torna indisponível;
- envio de resposta ao Zabbix proxy ao falhar na troca de dados devido a problemas de direitos ou criptografia;
- ICMP pinger;
- deadline para sockets IPC assíncronos e opções de controle de tempo de execução;
- conexões JMX;

- obtenção de estatísticas de um Zabbix proxy ou server remoto;
- envio de respostas para o Zabbix frontend;
- requisições DNS do poller assíncrono;
- resposta para heartbeat de checagem ativa;
- obtenção de dados (valores) do Zabbix agent de agents ativos;
- obtenção de dados do Zabbix sender;
- resposta quando o trapper falha ao processar uma requisição;
- envio da lista de checagens ativas para o Zabbix agent.

Padrão: 3
 Intervalo: 1-30

TLSCAFile

O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação de certificado de peer, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O caminho completo de um arquivo contendo o certificado do servidor ou 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 no 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 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 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.

TLSFrontendAccept

Quais conexões de entrada aceitar do frontend.

Vários valores podem ser especificados, separados por vírgula:

- unencrypted - aceitar conexões sem criptografia.
- cert - aceitar conexões protegidas com TLS e um certificado.

Padrão: `unencrypted`

TLSFrontendCertIssuer

Emissor de certificado frontend permitido.

TLSFrontendCertSubject

Assunto do certificado frontend permitido.

TLSKeyFile

O caminho completo de um arquivo contendo a chave privada do server, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSListen

Controla o TLS no socket trapper.

Valores suportados:

- required - aceita apenas conexões TLS

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 servidor 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-16G

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 um host é tratado como indisponível.

Padrão: 45
 Intervalo: 1-3600

Usuário

Reduz os privilégios para um usuário específico e existente no sistema.
Só tem efeito se executado como 'root' e AllowRoot estiver desabilitado.

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.
Definir como 0 desativa o cache de valores (não recomendado).
Quando o cache de valores fica sem memória compartilhada, uma mensagem de aviso é gravada no log do server a cada 5 minutos.

Padrão: 8M
 Intervalo: 0,128K-64G

Vault

Especifica o provedor de vault:
HashiCorp - HashiCorp KV Secrets Engine versão 2
CyberArk - CyberArk Central Credential Provider
Deve corresponder ao provedor de vault 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 o 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 o 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 de 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 HashiCorp Vault que deve ter sido gerado exclusivamente para o Zabbix server com permissão somente leitura para os caminhos especificados em macros do Vault e permissão somente leitura para o caminho especificado no parâmetro de configuração opcional VaultDBPath.
É um erro se VaultToken e a variável de ambiente VAULT_TOKEN estiverem definidos ao mesmo tempo.

Obrigatório: Sim, se 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 Verificações internas).
Observe que a memória compartilhada não é alocada se não houver instâncias do coletor vmware configuradas para iniciar.

Padrão: 8M
 Intervalo: 256K-2G

VMwareFrequency

O atraso em segundos entre a coleta de dados de um único serviço VMware.
Esse atraso deve ser definido para o menor intervalo de atualização de qualquer item de monitoramento 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 (vCenter ou hipervisor ESX).

Padrão: 10
 Intervalo: 1-300

WebServiceURL

A URL HTTP[S] para o serviço web do Zabbix no formato `<host:port>/report`.

Exemplo:

```
WebServiceURL=http://localhost:10053/report
```

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 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 históricos que esses items possuíam. Por exemplo, se precisarmos remover apenas 1 item prototype do template, mas esse template está vinculado a 50 hosts e para cada host o prototype 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) para os 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 Proxy Zabbix

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'.
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 para um arquivo contendo o(s) certificado(s) da(s) CA(s) de nível superior para verificação do certificado do banco de dados.
DBTLSCertFile	O caminho completo para um arquivo contendo o certificado do Zabbix proxy para autenticação no banco de dados.
DBTLSKeyFile	O caminho completo para 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 limpeza 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 na qual 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.

Parâmetro	Descrição
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 autorregistro (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 autorregistro 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.
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 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-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 inalcançáveis (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 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]`) e item `icmpping[*]`.
TLSAccept	Quais conexões de entrada aceitar do Zabbix server.
TLSCAFile	O caminho completo para um arquivo contendo o(s) certificado(s) 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 para 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.

Parâmetro	Descrição
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 para um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo para um arquivo contendo a chave privada do proxy, usado para comunicações criptografadas entre componentes do Zabbix.
TLSListen	Controla o TLS no socket do trapper.
TLSPSKFile	O caminho completo para um arquivo contendo a chave pré-compartilhada do proxy, 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 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 de cliente.
VaultTLSKeyFile	O nome do arquivo de chave privada SSL usado para autenticação de 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 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	A URL HTTP[S] da interface WebDriver.

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.
- Os valores suportam variáveis de ambiente.
- 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

AllowRoot

Permitir 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

Permitir que o proxy trabalhe com versões de banco de dados não suportadas.

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 comandos remotos.
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 string vazia resulta no uso de um socket. Com PostgreSQL, apenas uma string vazia resulta em tentativa de usar o socket.

Padrão: `localhost`

DBName

O nome do banco de dados ou o caminho para o arquivo do banco de dados para SQLite3 (a arquitetura de multiprocessos do Zabbix não permite o uso de banco de dados em memória, por exemplo, `:memory:`, `file::memory:?cache=shared` ou `file:memdb1?mode=memory&cache=shared`).
Aviso: Não tente usar o mesmo banco de dados que o Zabbix server está usando.

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.

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

Definir esta opção obriga o uso de conexão TLS com o banco de dados:
required - conectar usando TLS
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 especificada por DBHost corresponde ao seu certificado
No MySQL a partir da versão 5.7.11 e no PostgreSQL, os seguintes valores são suportados: ”required”, ”verify”, ”verify_full”.
No MariaDB a partir da versão 10.2.6, os valores ”required” e ”verify_full” são suportados.
Por padrão, não está definido para nenhuma opção e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação de certificado do banco de dados.

Obrigatório: não (sim, se DBTLSConnect definido como *verify_ca* ou *verify_full*)

DBTLSCertFile

O caminho completo de um arquivo contendo o certificado do proxy Zabbix 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 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

Especifica o nível de debug:
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

EnableRemoteCommands

Se comandos remotos do Zabbix server são permitidos.

Padrão: 0
 Valores: 0 - não permitido; 1 - permitido

ExternalScripts

O local dos scripts externos (depende da variável de instalação `datadir` em tempo de compilação).

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 o sinalizador SUID definido.

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-16G

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-16G

Hostname

Um nome de proxy exclusivo e sensível a maiúsculas e minúsculas. Certifique-se de que o nome do proxy seja conhecido pelo server.
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 o 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 posteriormente.
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*.

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 Java pollers forem iniciados.

JavaGatewayPort

A porta na qual o Zabbix Java gateway 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 separados por vírgula nos quais o trapper deve escutar.
O trapper irá escutar em todas as interfaces de rede se este parâmetro estiver ausente.

Padrão: 0.0.0.0

ListenPort

A porta de escuta para 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 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

LogRemoteCommands

Habilita o registro de comandos shell executados como avisos.

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 syslog;
console - grava o log na saída padrão.

Padrão: `file`

LogSlowQueries

Por quanto tempo uma consulta ao banco de dados pode demorar antes de ser registrada (em milissegundos).
0 - não registrar consultas lentas.
Esta 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 de HTTP agent, poller de agent ou poller de SNMP. 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 autorregistro: *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. No desligamento, 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. No desligamento, 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 dados de configuração do servidor Zabbix 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 server.
Este parâmetro pode ser usado se os dados locais forem usados por aplicativos de terceiros.

Padrão: 0
 Intervalo: 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 alternar 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 Zabbix server.
Dados mais antigos serão perdidos.

Padrão: 1
 Intervalo: 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 de entrada 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 SNMP trap para o proxy.
Deve ser o mesmo que no zabbix_trap_receiver.pl ou no arquivo de configuração do SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

O diretório para armazenar sockets 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 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 SSH e ações.

SSLCertLocation

O local dos arquivos de certificado de cliente SSL para autenticação de cliente.
Este parâmetro é usado apenas no monitoramento web.

SSLKeyLocation

O local dos arquivos de chave privada SSL para autenticação do cliente.
Este parâmetro é usado apenas no monitoramento 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 dos pollers de agent do Zabbix. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartBrowserPollers

O número de instâncias pré-forked de pollers de item de navegador.

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 inalcançáveis (incluindo IPMI e Java). 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.

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
 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 para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`) e item `icmpping[*]`. Este parâmetro define a duração para várias operações de comunicação:

- Execução de comando remoto no Zabbix agent
- Execução de comando SSH / Telnet
- Reagendamento de itens quando a interface IPMI se torna indisponível
- ICMP pinger
- 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 verificação ativa de heartbeat
- 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

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 caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação de certificado de peer, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O caminho completo para um arquivo contendo o certificado do proxy ou 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 no 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
```

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 caminho completo de um arquivo contendo a chave privada do proxy, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSListen

Controla o TLS no socket trapper.

Valores suportados:

- required - aceita apenas conexões TLS

TLSPSKFile

O caminho completo de um arquivo contendo a chave pré-compartilhada do proxy, 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 permitido.

TLSServerCertSubject

O assunto do certificado do servidor permitido.

TmpDir

O diretório temporário.

Padrão: `/tmp`

TrapperTimeout

Especifica o tempo limite em segundos para:
 - obtenção 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 para um usuário específico e existente no sistema.
Só tem efeito se executado como 'root' e AllowRoot estiver desabilitado.

Padrão: `zabbix`

Vault

O provedor do vault:
HashiCorp - HashiCorp KV Secrets Engine versão 2
CyberArk - CyberArk Central Credential Provider
Deve corresponder ao provedor do vault 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.
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 HashiCorp vault que deve ter sido gerado exclusivamente para o Zabbix proxy com 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 estiverem definidos ao mesmo tempo.

Obrigatório: Sim, se 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 Verificações internas).
Observe que a memória compartilhada não é alocada se não houver instâncias do coletor vmware configuradas para iniciar.

Padrão: `8M`
 Intervalo: 256K-2G

VMwareFrequency

O atraso em segundos entre a coleta de dados de um único serviço VMware.
Esse atraso deve ser definido para o menor intervalo de atualização de qualquer item de monitoramento 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 (vCenter ou hipervisor 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 Agent 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 nesta porta 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.

Parâmetro	Descrição
Server	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS de 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 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 prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de suíte de cifras 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 suíte de cifras 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 suíte de cifras 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 suíte de cifras 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 suíte de cifras 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 suíte de cifras 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.
- Os valores suportam variáveis de ambiente.
- 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 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

Permitir 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.
Múltiplas regras de correspondência de chave podem ser definidas em combinação com DenyKey. Os parâmetros são processados um por um de acordo com sua ordem de aparição. Veja também: Restringindo verificações do agent.

AllowRoot

Permitir 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 sob um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

BufferSend

Não mantenha os dados por 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

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 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 caracteres.
Várias regras de correspondência de chaves 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 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 (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.

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. Obrigatório 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 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 definida no código, 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 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

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 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 do 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

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 syslog;
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' 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 em 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. 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 de cluster é um ou mais endereços de server separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. 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 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 quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o proxy ou server Zabbix.
 Este parâmetro define a duração de várias operações, incluindo:

- aguardar uma resposta do server Zabbix;
- enviar solicitações ao server Zabbix, incluindo solicitações de configuração de checagens ativas e dados de item;
- 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 checagens `vfs.*`;
- sendo usado pelos módulos do agent Zabbix;
- servindo como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia checagens sem timeouts.

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 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 contendo o certificado do agent ou 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 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 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 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
```

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 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 servidor (proxy) permitido.

UnsafeUserParameters

Permitir que todos os caracteres sejam passados em 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 para um usuário específico e existente no sistema.
Só tem efeito se executado como 'root' e AllowRoot estiver desabilitado.

Padrão: `zabbix`

UserParameter

Um parâmetro definido pelo usuário para monitoramento. Podem 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 irá alterar 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 verificações ativas e passivas a partir da versão 2.0.0

4 Agent Zabbix 2 (UNIX)

Visão geral

O agent 2 do Zabbix é uma nova geração do agent do Zabbix e pode ser usado no lugar do agent do Zabbix.

Os parâmetros suportados pelo arquivo de configuração do agent 2 do Zabbix (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 mantenha 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'.
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 agent do 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 do 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.
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 agent 2 do Zabbix 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 server ou proxy do Zabbix ao processar verificações ativas 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Habilita o registro dos comandos de shell executados como avisos.
PluginSocket	O caminho para o socket UNIX para comunicações de plugin 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 na notação CIDR, ou nomes DNS de servers e proxies do Zabbix.
ServerActive	O endereço do server/proxy do Zabbix 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 proxy ou server do Zabbix.
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.

Parâmetro	Descrição
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 server ou proxy do Zabbix.
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 server do Zabbix.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o server do Zabbix.
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 em 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.
- Os valores suportam variáveis de ambiente.
- 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.
Múltiplas regras de correspondência de chave podem ser definidas em combinação com DenyKey. Os parâmetros são processados um por um de acordo com sua ordem de 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 antes.

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 socket de controle, usado para enviar comandos em tempo de execução com a opção ’-R’.

Padrão: `/tmp/agent.sock`

DebugLevel

Especifica o nível de debug:
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 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 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. HostMetadata permite distinguir entre hosts além do hostname.

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.

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 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.

Hostname

Uma lista de nomes de host exclusivos, sensíveis a maiúsculas e minúsculas, separados por vírgulas. Obrigatório 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 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 o tempo de 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=/caminho/absoluto/para/arquivos/de/configuracao/*.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 conectando a ele, para recuperar a lista de verificações ativas.

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.

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 - 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 é 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`

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 pelo 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 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 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. 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 socket UNIX para comunicações de plugin carregáveis.

Padrão: `/tmp/agent.plugin.sock`

PluginTimeout

O tempo limite para conexões com plugins 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 servers 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 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 de cluster é um ou mais endereços de server separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. 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 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 agent irá escutar nesta porta por requisições 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 proxy ou server Zabbix.
 Este parâmetro define a duração de várias operações, incluindo:

- aguardar uma resposta do server Zabbix;
- enviar solicitações ao server Zabbix, incluindo solicitações de configuração de checagens ativas e dados de item;
- 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 checagens `vfs.*`;
- servir como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia checagens sem timeouts.

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.

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
```

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
```

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 em vez disso.

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.

TLSCipherPSK

A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
kECDHEPSK+AES128:kPSK+AES128
```

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
```

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 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 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. 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,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 Agent Zabbix (Windows)

Visão geral

Os parâmetros suportados pelo arquivo de configuração do Zabbix agent 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 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.
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 performance 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 performance 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 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.
StartAgents	O número de instâncias pré-iniciadas 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.
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, 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 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 daemon, não os valores nos arquivos de configuração fornecidos.
- Os valores suportam variáveis de ambiente.
- 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 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 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 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.
Múltiplas regras de correspondência de chaves podem ser definidas em combinação com DenyKey. Os parâmetros são processados um por 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 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).

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 caracteres.
Várias regras de correspondência de chaves 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 seu lugar.
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 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 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 a 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 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 multilinha 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 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`

Incluir

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 silenciosamente truncados 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 neste porto 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 é 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 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 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. 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 server separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. Mais de um Zabbix proxy não deve ser especificado para cada Zabbix server/cluster. Se o 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 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 desabilitadas 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 é usado para definir a duração de várias operações:

- aguardando uma resposta do Zabbix server;
- enviando solicitações para o Zabbix server, incluindo solicitações de configuração de checagens ativas e dados de item;
- recuperando dados de log por meio do monitoramento de arquivos de log ou do log de eventos do Windows;
- enviando mensagens de heartbeat;
- duração máxima das checagens `vfs.*`;
- usado pelos módulos do agent Zabbix;
- também usado como fallback em cenários onde server/proxy mais antigos que a versão 7.0 estão enviando checagens sem timeouts.

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 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 contendo 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 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 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 que contém 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 server (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. 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
```

> **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 Agent Zabbix 2 (Windows)

Visão geral

O agent 2 do Zabbix é uma nova geração do agent do Zabbix e pode ser usado no lugar do agent do Zabbix.

Os parâmetros suportados pelo arquivo de configuração do agent 2 do Zabbix 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 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'.
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 agent do 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 do Zabbix usado para obter o nome do host.

Parâmetro	Descrição
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 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 agent 2 do Zabbix 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>.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 server ou proxy do Zabbix ao processar verificações ativas 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Habilita o registro dos comandos de 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 verificações ativas é atualizada.
Server	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS de servers e proxies do Zabbix.
ServerActive	O endereço do server/proxy do Zabbix ou configuração de cluster para obter verificações ativas.
SourceIP	O endereço IP de origem.
StatusPort	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 proxy ou server do Zabbix.
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.
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 server ou proxy do Zabbix.
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 server do Zabbix.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o server do Zabbix.
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;
- Os valores suportam variáveis de ambiente;
- 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 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 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.

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 antes.

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 socket de controle, usado para enviar comandos em tempo de execução com a opção '-R'.

Padrão: `\\.\pipe\agent.sock`

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 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 (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.

Metadados de múltiplas 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 a 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.

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. Obrigatório para verificações ativas e deve corresponder aos nomes de host configurados no server. O valor é obtido de HostnameItem se não 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`

Incluir

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 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. 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 estiver conectando a ele, para recuperar a lista de verificações ativas.

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_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 - desabilita a rotação automática do 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;
console - grava o log 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 habilitado (*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 as 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 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 socket UNIX para comunicações de plugin carregáveis.

Padrão: \\.\pipe\agent.plugin.sock

PluginTimeout

O tempo limite para conexões com plugins 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 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 de cluster é um ou mais endereços de server separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. Mais de um Zabbix proxy não deve ser especificado para 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 desabilitadas.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

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 agent irá escutar nesta porta por requisições 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 proxy ou server Zabbix.
 Este parâmetro define a duração de várias operações, incluindo:

- aguardar uma resposta do server Zabbix;
- enviar solicitações ao server Zabbix, incluindo solicitações de configuração de checagens ativas e dados de item;
- 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 checagens `vfs.*`;
- servir como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia checagens sem timeouts.

Padrão: 3
 Intervalo: 1-30

TLSAccept

As conexões de entrada a serem aceitas. Usado para checagens 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 estiverem 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 contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas com 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
```

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.

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
```

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 caminho completo do arquivo que contém 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 server (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=<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,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

O caminho de pesquisa padrão para comandos UserParameter. Se usado, o agent irá alterar 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 agent 2 do Zabbix

Visão geral

Esta seção contém descrições dos parâmetros do arquivo de configuração para os plugins do Zabbix agent 2. Utilize a barra lateral para acessar informações sobre o plugin específico.

1 Plugin Ceph

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Ceph.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `ceph.conf`) e, em seguida, usar a diretiva Include para adicionar esse arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.Ceph.Default.ApiKey	não			Chave de API padrão para conectar ao Ceph; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Ceph.Default.Mode	não	native / restful	restful (obsoleto)	Modo padrão para conectar ao Ceph. O modo nativo é suportado apenas em Linux e a partir do Ceph 16. O modo restful não funcionará com o Ceph versão 20 ou mais recente. Observe que o conjunto de credenciais do usuário difere para cada modo e não são compatíveis entre si.
Plugins.Ceph.Default.User	não			Nome de usuário padrão para conectar ao Ceph; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Ceph.Default.Uri	não		https://localhost:8003	URI padrão para conectar ao 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`
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>** - defina o nome de uma sessão para usar em chaves de item.
Plugins.Ceph.Sessions.<SessionName>.Mode	não	Native / restful	restful (obsoleto)	Modo de sessão nomeada para conectar ao Ceph. O modo nativo é suportado apenas em Linux e a partir do Ceph 16. O modo restful não funcionará com o Ceph versão 20 ou mais recente. Observe que o conjunto de credenciais do usuário difere para cada modo e não são compatíveis entre si.
Plugins.Ceph.Sessions.<SessionName>.User	não			Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome de uma sessão para usar em chaves de item.
Plugins.Ceph.Sessions.<SessionName>.Uri	não			String de conexão de uma sessão nomeada. **<SessionName>** - defina o nome de uma sessão para usar 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

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Docker.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `docker.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;

- Os valores suportam variáveis de ambiente;
- O Zabbix suporta arquivos de configuração apenas em codificação UTF-8 sem BOM;
- Comentários iniciados 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

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Ember+.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `emberplus.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

O plugin Ember+ é um plugin carregável e está disponível e totalmente descrito no repositório do plugin Ember+.

Este plugin está atualmente 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.
- Os valores suportam variáveis de ambiente;
- 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	Exibe a versão do plugin e informações de licença.
-h --help	Exibe informações de ajuda (forma abreviada).
-t, --test <item key>	Executa o plugin para teste (configuração do plugin ignorada).

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/libexec/zabbix/zabbix-`

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
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

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Estes parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Memcached.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `memcached.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- O Zabbix suporta arquivos de configuração apenas em codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Memcached.Default.Password	não			Senha padrão para conexão ao Memcached; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.

Plugins.Memcached.Default.Uri	não		tcp://localhost:11211	URI padrão para conexão 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 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 conexão 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 uso nas chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição

Plugins.Memcached.Sessions.<SessionName>.Uri	não			String de conexão da sessão nomeada.
				<SessionName> - defina o nome da sessão para uso nas chaves de item.
				
				Não deve incluir credenciais embutidas (serão ignoradas).
				Deve corresponder ao formato 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 uso nas chaves de item.
Plugins.Memcached.Timeout	não	1-30	timeout global	Timeout de execução da requisição (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

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Estes parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Modbus.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `modbus.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.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 timeout de execução da requisição (quanto tempo esperar para uma requisição ser concluída antes de encerrá-la), use o formulário de configuração 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

6 Plugin MongoDB

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Estes parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin MongoDB.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mongo.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

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.
- Os valores suportam variáveis de ambiente;
- 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 (abreviação).
-t, --test <item key>	Executa o plugin para teste (configuração do plugin ignorada).

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.
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 de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Não (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 par, 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 de item.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Sim, se Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile for especificado			Caminho completo para um arquivo contendo o certificado do Client 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 de item.
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 de item. 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 (key) 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 de item.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Não			String de conexão de uma sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves de item. 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.Sessions.<SessionName>.User	Não			Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves de item.
Plugins.MongoDB.System.Path	Não			Caminho para o executável do plugin MongoDB. Exemplo de uso: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout	Não	1-30	timeout global	Timeout de execução da requisição (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 configuração de plugins

7 Plugin MQTT

Visão geral

O arquivo de configuração do agent Zabbix 2 é usado para configurar plugins. Esses parâmetros de configuração do agent Zabbix 2 são suportados para operar o plugin MQTT.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mqtt.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do agent Zabbix 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.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 o(s) certificado(s) 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`

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 da sessão para uso nas chaves de item.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	não			Caminho completo para um arquivo contendo o(s) certificado(s) 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 da sessão para uso nas chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				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 da sessão para uso nas chaves de item.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				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 da sessão para uso nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.Topic				Tópico da sessão nomeada para assinatura MQTT.
				<SessionName> - defina o nome da sessão para uso nas chaves de item.
				
				O tópico pode conter curingas (”+”,”#”)
				Exemplos: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Sessions.<SessionName>.Url				String de conexão de uma sessão nomeada.
				<SessionName> - defina o nome da sessão para uso 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				Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves de item.

Se você precisar definir o tempo limite de execução da requisição (quanto tempo esperar para que uma requisição seja concluída antes de encerrá-la), utilize 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

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin MSSQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mssql.conf`) e, em seguida, usar a diretiva Include para adicionar esse arquivo à configuração do Zabbix agent 2.

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.
- Os valores suportam variáveis de ambiente;
- 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	Exibe a versão do plugin e informações de licença.
-h --help	Exibe informações de ajuda (forma abreviada).
-t, --test <item key>	Executa o plugin para teste (configuração do plugin ignorada).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MSSQL.CustomQueriesDir	Não		`/usr/local/share/zabbix/custom-queries/mssql` para sistemas Unix `*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` para sistemas Windows, onde `*` é o nome da unidade da variável de ambiente `ProgramFiles`	Especifica o caminho do 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.CustomQueriesEnabled	Não		false	Se definido, habilita a execução da chave de item `mssql.custom.query`. Se desabilitado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.
Plugins.MSSQL.Default.CACertPath	Não			O caminho padrão do arquivo para o certificado de 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 do banco de dados padrão 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.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Não			O caminho do arquivo para o certificado de 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.
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 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. Aplicada a todas as conexões. Exemplo de uso: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
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

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin MySQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mysql.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.Mysql.CallTimeout	não	1-30	timeout global	Tempo máximo em segundos para aguardar a conclusão de uma solicitação.
Plugins.Mysql.CustomQueriesPath	não		`/usr/local/share/zabbix/custom-queries/mysql` para sistemas Unix `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` para sistemas Windows, onde `*` é o nome da unidade da variável de ambiente `ProgramFiles`	Caminho para o diretório com arquivos `.sql` com consultas personalizadas.
Plugins.Mysql.CustomQueriesEnabled	não		false	Se definido, habilita a execução da chave de item `mysql.custom.query`. Se desabilitado, nenhuma consulta será carregada do diretório de consultas personalizadas.
Plugins.Mysql.Default.Password	não			Senha padrão para conexão com o MySQL; usada se nenhum valor for especificado na chave de item ou sessão nomeada.
Plugins.Mysql.Default.TLSCAFile	não (sim, se Plugins.Mysql.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo do 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 do 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.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Mysql.Default.TLSConnect	não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e 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 do 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 conexão com o MySQL; usado se nenhum valor for especificado na chave de item ou 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 conexão com o MySQL; usado se nenhum valor for especificado na chave de item ou 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 da sessão para uso nas chaves de item.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	não (sim, se Plugins.Mysql.Sessions.<SessionName>.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			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 o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - define o nome da sessão para uso nas chaves de item.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	sim, se Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile for especificado			Caminho completo do 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 da sessão para uso nas chaves de item.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e bancos de dados monitorados. **<SessionName>** - define o nome da sessão para uso nas chaves de item. Valores suportados: *required* - requer conexão TLS; *verify_ca* - verifica certificados; *verify_full* - verifica certificados e endereço IP.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	não			Caminho completo do 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 da sessão para uso nas chaves de item.

Plugins.Mysql.Sessions.<SessionName>.Uri	não			String de conexão da sessão nomeada.
				<SessionName> - define o nome da 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 da 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 NVIDIA GPU

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin NVIDIA GPU.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `nvidiagpu.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

O plugin NVIDIA GPU é um plugin carregável e está disponível e totalmente descrito no repositório do plugin NVIDIA GPU.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.NVIDIA.System.Path	sim			Caminho para o executável do plugin NVIDIA GPU. Exemplo de uso: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout	não	1-30	timeout global	Especifica o tempo máximo (em segundos) para aguardar uma resposta do server durante as tentativas de conexão e operações subsequentes na sessão. O timeout global do tipo de item ou o timeout individual do item substituirão este valor se forem maiores.

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 Oracle

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Estes parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Oracle.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `oracle.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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		`/usr/local/share/zabbix/oracle/custom_queries` para sistemas Unix `*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle` para sistemas Windows, onde `*` é o nome da unidade da variável de ambiente `ProgramFiles`	Caminho do diretório contendo arquivos `.sql` com consultas personalizadas. Exemplo: `/etc/zabbix/oracle/sql`
Plugins.Oracle.CustomQueriesEnabled	não		false	Se definido, habilita a execução da chave de item `oracle.custom.query`. Se desabilitado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.
Plugins.Oracle.Default.Password	não			Senha padrão para conectar ao 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 conectar ao Oracle (SID não é suportado); usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição

Plugins.Oracle.Default.Uri	não		tcp://localhost:1521	URI padrão para conectar ao Oracle; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
				
				Não deve incluir credenciais embutidas (serão ignoradas).
				Deve corresponder ao formato URI.
				Apenas o esquema `tcp` é suportado; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=1521).
				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 conectar ao 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) para conectar ao servidor Oracle. Se definido como false, o esquema e a porta padrão serão adicionados à ConnString (se ausentes). Se a opção for definida como true, o esquema e a porta padrão serão omitidos (a menos que explicitamente especificados em ConnString), e ConnString será passada para o cliente Oracle como está. Se o cliente Oracle encontrar ConnString no arquivo tnsnames.ora, a descrição da conexão encontrada será usada para conectar ao servidor Oracle.
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.

Plugins.Oracle.Sessions.<SessionName>.Uri	não			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 URI.
				Apenas o esquema `tcp` é suportado; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=1521).
				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	não			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

12 Plugin do PostgreSQL

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin do PostgreSQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `postgresql.conf`) e, em seguida, usar a diretiva Include para adicionar esse arquivo à configuração do Zabbix agent 2.

O plugin do 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.
- Os valores suportam variáveis de ambiente;
- 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	Exibe a versão do plugin e informações de licença.
-h --help	Exibe informações de ajuda (forma abreviada).
-t, --test <item key>	Executa o plugin para teste (configuração do plugin ignorada).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.Default.CacheMode	Não		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	Não	1-30	timeout global	Tempo máximo de espera (em segundos) para que uma solicitação seja concluída.
Plugins.PostgreSQL.CustomQueriesPath	Não		`/usr/local/share/zabbix/custom-queries/postgresql` para sistemas Unix `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` para sistemas Windows, onde * é o nome da unidade da variável de ambiente `ProgramFiles`	Caminho para o diretório contendo arquivos `.sql` com consultas personalizadas.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.CustomQueriesEnabled	Não		false	Se definido, habilita a execução da chave de item `postgresql.custom.query`. Se desabilitado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.
Plugins.PostgreSQL.Default.Database	Não			Banco de dados padrão para conexão com o PostgreSQL; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.PostgreSQL.Default.Password	Não			Senha padrão para conexão com o 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 conexão com o PostgreSQL; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
				
				Não deve incluir credenciais incorporadas (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 conexão com o 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>.CachePolicy	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 (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 PostgreSQL. Exemplo de uso: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.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 configuração de plugins

13 Plugin Redis

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Estes parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin Redis.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `redis.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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.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 embutidas (serão ignoradas).
				Deve corresponder ao formato de 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			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>.Uri	não			String de conexão de uma sessão nomeada.
				<SessionName> - defina o nome da sessão a ser usado 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=6379).
				Exemplos: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User	não			Usuário para enviar ao servidor Redis protegido. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens.
Plugins.Redis.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

14 Plugin SMART

Visão geral

O arquivo de configuração do Zabbix agent 2 é usado para configurar plugins. Esses parâmetros de configuração do Zabbix agent 2 são suportados para operar o plugin SMART.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `smart.conf`) e, em seguida, usar a diretiva Include para adicionar este arquivo à configuração do Zabbix agent 2.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- Os valores suportam variáveis de ambiente;
- 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 carregam 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 o `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 um aplicativo 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.

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.
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 na formatação do relatório em PDF de um dashboard.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo para 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 para um arquivo contendo o certificado do serviço ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.

Parâmetro	Descrição
TLSKeyFile	O caminho completo para um arquivo contendo a chave privada do serviço, usado para comunicações criptografadas entre componentes do Zabbix.

Todos os parâmetros são opcionais, 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;
- Os valores suportam variáveis de ambiente;
- 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 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.

Exemplo:

`127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com`

Obrigatório: sim

DebugLevel

Especifica o nível de debug:
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

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 do 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 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 serviço ou cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSKeyFile

O caminho completo do arquivo contendo a chave privada do serviço, usada para comunicações criptografadas entre os componentes do Zabbix.

10 Variáveis de ambiente

Visão geral

As variáveis de ambiente permitem configurar os componentes do Zabbix sem codificar valores nos arquivos de configuração. Isso facilita o gerenciamento de configurações em ambientes dinâmicos, como o Docker, onde as variáveis podem ser passadas em tempo de execução para se adaptar a diferentes configurações.

No caso mais simples, ao definir o valor do parâmetro de configuração DebugLevel do Zabbix server para uma variável de ambiente, você pode usá-la para configurar o server na inicialização:

```
#### Arquivo de configuração do Zabbix server:
DebugLevel=${NEW_DEBUG_LEVEL}

#### Iniciando o Zabbix server:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

As variáveis de ambiente são suportadas pelos seguintes componentes do Zabbix:

- Server
- Proxy
- Agent (UNIX ou Windows)
- Agent 2 (UNIX ou Windows), incluindo plugins
- Web service
- Zabbix sender (ao usar a opção -c, --config)

Notas importantes

- Quando um parâmetro de configuração é definido como uma variável de ambiente, que não é especificada ao executar o componente, o valor padrão do parâmetro é usado.
- Ao usar comandos em tempo de execução (por exemplo, para aumentar o nível de log do agent), quaisquer variáveis de ambiente usadas anteriormente devem ser especificadas. Isso ocorre porque os componentes do Zabbix usam seu arquivo de configuração para executar comandos em tempo de execução; se as variáveis de ambiente forem omitidas, os valores padrão dos parâmetros de configuração serão usados. Veja Exemplos.
- O comando em tempo de execução `userparameter_reload` runtime command não suporta recarregar variáveis de ambiente. Durante o recarregamento, as variáveis são ignoradas e apenas os parâmetros com valores regulares são recarregados.
- As variáveis de ambiente atuais do processo, que foram usadas nos arquivos de configuração, são limpas após o componente Zabbix ser iniciado. Isso garante que processos filhos (por exemplo, scripts remotos executados pelo Zabbix) não possam acessar essas variáveis. No entanto, observe que as variáveis iniciais do processo ainda podem ser recuperadas (por exemplo, via arquivo `/proc/<PID>/environ`).

Sintaxe

As variáveis de ambiente devem usar a seguinte sintaxe: `${alfanuméricos/underscores}`.

O nome da variável pode incluir apenas letras (a-z, A-Z), underscores (_) e dígitos (0-9), e não deve começar com um dígito.

Variáveis que não correspondam à sintaxe exigida ou que sejam combinadas com um valor regular serão tratadas como valores regulares, o que pode produzir erros.

Sintaxe correta de variável:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Sintaxe incorreta de variável:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
No Windows, os nomes das variáveis de ambiente não diferenciam maiúsculas de minúsculas.

Exemplos

Os exemplos a seguir mostram como configurar e usar variáveis de ambiente com os componentes do Zabbix.

Exemplo 1: Configurando e testando o agent Zabbix

1. Defina as variáveis de ambiente no arquivo de configuração do agent:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Teste o arquivo de configuração:

```
ZBX_HOSTNAME="Novo agent Zabbix" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agen
```

3. Inicie o agent com variáveis de ambiente:

```
ZBX_HOSTNAME="Novo agent Zabbix" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agen
```

Ao usar comandos em tempo de execução (por exemplo, para aumentar o nível de log do agent), quaisquer variáveis de ambiente usadas anteriormente devem ser especificadas:

```
ZBX_HOSTNAME="Novo agent Zabbix" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agen
```

Isso ocorre porque o agent usa seu arquivo de configuração para executar comandos em tempo de execução; se as variáveis de ambiente forem omitidas, os valores padrão dos parâmetros de configuração serão usados.

Como alternativa, após definir as variáveis de ambiente no arquivo de configuração do agent, você pode disponibilizá-las para os processos (por exemplo, usando o comando `export`). Isso reduz o risco de comportamento inesperado devido a variáveis ausentes ou configuradas incorretamente.

```
export ZBX_HOSTNAME="Novo agent Zabbix"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

Exemplo 2: Configurando o agent Zabbix para um container

Se você estiver criando e configurando sua própria imagem personalizada para componentes do Zabbix (por exemplo, agent Zabbix), poderá definir parâmetros de configuração usando variáveis de ambiente e, em seguida, iniciar o container com essas variáveis.

1. Ao preparar a imagem, defina as variáveis de ambiente no arquivo de configuração do agent:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Após criar a imagem do container, inicie o container do agent (por exemplo, Docker) com variáveis de ambiente:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Ao usar comandos em tempo de execução (por exemplo, para aumentar o nível de log do agent), acesse o shell do container e execute o comando em tempo de execução:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
O comando em tempo de execução `userparameter_reload` não oferece suporte ao recarregamento de variáveis de ambiente. Durante o recarregamento, as variáveis são ignoradas e apenas os parâmetros com valores regulares são recarregados.

11 Inclusão

Visão geral

Arquivos ou diretórios adicionais podem ser incluídos na configuração do server/proxy/agent 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 server-proxy

Visão geral

A troca de dados entre server e proxy é baseada no formato JSON.

As mensagens de solicitação e resposta devem começar com cabeçalho e comprimento dos dados.

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 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 de <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).
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": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
```

```
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
```

```
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
```

```
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Solicitação de dados

A solicitação `proxy data` é usada para obter a disponibilidade da interface do host, dados históricos, de descoberta e de auto-registro 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'
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 crescente, 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.

nome	tipo de valor	descrição
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.
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 da 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 UP **1**, *DOBJECT_STATUS_DOWN* - Serviço DOWN
auto registration	*array*	*(opcional)* Array de objetos de dados de auto-registro.
clock	*number*	Timestamp dos dados de auto-registro.
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

nome	tipo de valor	descrição
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.
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":"8.0.0"
}
```


server→proxy:


```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
      }
  ]
}
```


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.
server→proxy:		
fullsync	*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*	Informação de macro secreta, ausente se não houver alterações em macros do cofre (opcional).
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":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
```

```
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
```

```
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
```

```
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


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 de histórico.
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 crescente, 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.

nome	tipo de valor	descrição
dcheckid	*number*	Identificador da verificação de descoberta ou nulo para dados de 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.
status	*number*	*(opcional)* Status do serviço: **0**, *DOBJECT_STATUS_UP* - Serviço UP **1**, *DOBJECT_STATUS_DOWN* - Serviço DOWN
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 (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 de histórico 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 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*	Alvo 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.
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": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
```

```
}]
}
```

2 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 Protocolo de plugin do Zabbix agent 2

O protocolo do Zabbix agent 2 é baseado em código, tamanho e modelo de dados.

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 de 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, ela contém apenas os parâmetros de dados comuns.

Exemplo:

```
{"id":3,"type":4}
```

Solicitação de encerramento

Uma solicitação enviada pelo agent para desligar um plugin.

direction	agent → plugin
response	não

A solicitação não possui parâmetros específicos, apenas contém os parâmetros de dados comuns.

Exemplo:

```
{"id":3,"type":5}
```

Solicitação de exportação

Uma solicitação para executar a função Export da interface Exporter.

direção	agent → plugin
resposta	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":"error message"}
```

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	objeto JSON (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 o formato de pacote grande possa ser usado para enviar qualquer dado, atualmente apenas o sincronizador de configuração do proxy do Zabbix consegue 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	Comprimento dos dados.
<RESERVED>	4	8	Quando a compressão é usada (flag 0x02) - o comprimento 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 Novo protocolo de exportação JSON delimitado por nova linha

Esta seção apresenta detalhes do protocolo de exportação no formato JSON delimitado por nova linha, utilizado 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 possuem 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":"Either Zabbix agent is unreachable on Host B or polle
```

Recuperação:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problema (geração de múltiplos eventos de problema):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

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	object	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).
-	object	
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).
ns	number	Número de nanossegundos a ser adicionado a `clock` para obter o tempo preciso de coleta do valor.
timestamp (*Log* apenas)	number	0 se não disponível.
source (*Log* apenas)	string	String vazia se não disponível.
severity (*Log* apenas)	number	0 se não disponível.
eventid (*Log* apenas)	number	0 se não disponível.
value	number (para itens numéricos) ou string (para itens de texto)	Valor do item coletado.
type	number	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	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).
count	número	Número de valores coletados para uma determinada hora.
min	número	Valor mínimo do item para uma determinada hora.
avg	número	Valor médio do item para uma determinada hora.
max	número	Valor máximo do item para uma determinada hora.
type	número	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 do 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 recarregada 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 que nem todo o cache de página pode ser liberado e o low watermark ser usado 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 alguma informação útil.
* Ao mesmo tempo, memórias inativas, cached e free são consideradas disponíveis, porque esses tipos de memória podem ser instantaneamente fornecidos a processos que solicitam mais memória.

Portanto, a memória inativa é simultaneamente usada e disponível. 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 o cálculo de memória em diferentes SO

2 Verificações passivas e ativas do agent

Visão geral

Esta seção fornece detalhes sobre as verificações passivas e ativas realizadas pelo agent Zabbix e 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 a partir do 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 de volta o resultado para o 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	*string*	sim	Timeout do item.

Resposta do agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

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”:”8.0.0”,”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”:”8.0.0”,”variant”:2,”data”:[{”error”:”Unsupporte 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 7.2, 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 posterior) 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 estão 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 (valor fixo).

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": "8.0.0",
  "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": "1234456akdsjhfoui",
  "host": "Zabbix server",
  "version": "8.0.0",
  "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 em caso de problemas de rede).
itemid	*string*	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 estado 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 o 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 server 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). Este ID é usado para descartar valores duplicados que podem ser enviados em ambientes com conectividade ruim.

Resposta do server

A resposta de dados do agent é enviada pelo server 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 server (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 server processa os dados e envia o status de volta
4. A conexão TCP é fechada

Observe como, no exemplo acima, o status não suportado para vfs.fs.size[/nono] é indicado pelo valor ”state” igual a 1 e pela mensagem de erro na propriedade ”value”.

Attention:
A mensagem de erro será limitada a 2048 símbolos no lado do server.

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": "8.0.0",
  "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 do 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 outra instância de proxy ou 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 mais antigo

Note:
O Zabbix aceitará até 16 MB de dados XML codificados em Base64, mas um único valor decodificado não deve ter mais de 64 KB, caso contrário, será truncado para 64 KB durante a decodificação.

3 Nível mínimo de permissão para itens do agent do Windows

Visão geral

Ao monitorar sistemas usando um agent, uma boa prática é obter métricas do host no qual o agent está instalado. Para usar o princípio do menor privilégio, é necessário determinar quais métricas são obtidas do agent.

A tabela neste documento permite selecionar os direitos mínimos para garantir o funcionamento correto do Zabbix agent.

Se um usuário diferente for selecionado para o agent funcionar, em vez de 'LocalSystem', então, para a operação do agent como um serviço do Windows, o novo usuário deve ter os direitos "Log on as a service" em "Política local→Atribuição de direitos do usuário" e o direito de criar, gravar e excluir o arquivo de log do Zabbix agent. Um usuário do Active Directory deve ser adicionado ao grupo *Performance Monitor Users*.

Note:
Ao trabalhar com os direitos de um agent com base no grupo ”mínimo tecnicamente aceitável”, é necessário fornecer previamente os direitos aos objetos para monitoramento.

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
system.cpu.util	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
system.hostname	Convidados	Convidados

Chave do item	Grupo de usuários	
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.size	Administradores	Convidados
vfs.fs.get	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

Chaves de item 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
perf_counter	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
proc_info	Administradores	Convidados
service.discovery	Convidados	Convidados
service.info	Convidados	Convidados
services	Convidados	Convidados
wmi.get	Administradores	Convidados

Chave do item	Grupo de usuários	
vm.vmemory.size	Convidados	Convidados

4 Codificação dos valores retornados

O Zabbix server espera que todo valor de texto retornado esteja na codificação UTF8. Isso se aplica a qualquer tipo de verificação: Zabbix agent, SSH, Telnet, etc.

Diferentes sistemas/dispositivos monitorados e verificações podem retornar caracteres não-ASCII no valor. Para esses casos, quase todas as possíveis chaves do **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 os seguintes 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 levará a uma falha na consulta SQL (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 sistemas Solaris de 32 bits devido à sua incompatibilidade com procfs e swapctl.

6 Sensor

Cada chip de sensor recebe seu próprio diretório na árvore sysfs /sys/devices. Para encontrar todos os chips de sensor, é mais fácil seguir os links simbólicos de dispositivo 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 sensor 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 **sensor-detect** e **sensors** (pacote lm-sensors: http://lm-sensors.org/). **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 sensor. A rotulagem das leituras dos sensores, usada por este programa, pode ser diferente do esquema comum de nomenclatura (<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 **sensor**, os sensores disponíveis são separados pelo tipo de barramento (adaptador ISA, adaptador PCI, adaptador SPI, dispositivo virtual, interface ACPI, adaptador HID).

No Linux 2.4:

(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]

No Linux 2.6+:

(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, device e sensor são tratados literalmente).

Exemplo de chave:

sensor[k8temp-pci-00c3,temp,max] ou sensor[0000:00:18.3,temp1]

sensor[smsc47b397-isa-0880,in,avg] ou sensor[smsc47b397.2176,in1]

Obtendo nomes de sensores

Os rótulos dos sensores, conforme impressos pelo comando *sensors*, nem sempre podem ser usados diretamente porque a nomeação dos rótulos pode ser diferente para cada fornecedor de chip de sensor. 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 rótulo pode ser usado diretamente:

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Tentar usar outros rótulos (como *Vcore* ou *+12V*) não funcionará.

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Para descobrir o nome real do sensor, que pode ser usado pelo Zabbix para recuperar as leituras do sensor, 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
```

...

Portanto, *Vcore* deve ser consultado como *in1* e *+12V* deve ser consultado como *in4*. De acordo com a especificação, estas 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 a voltagem (in), mas também leituras de corrente (curr), temperatura (temp) e velocidade do ventilador (fan) podem ser recuperadas pelo Zabbix.

7 Notas sobre o parâmetro memtype em itens proc.mem

Visão geral

O parâmetro **memtype** é suportado nas plataformas Linux, AIX, FreeBSD e Solaris.

Três valores comuns de 'memtype' são suportados em todas essas plataformas: `pmem`, `rss` e `vsize`. Além disso, valores de 'memtype' específicos da plataforma são suportados em algumas plataformas.

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 do Zabbix.
3. Vamos passar pelo exemplo do 'topasrec' para mostrar como o agent do 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[] possui argumentos:

```
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]
```

O primeiro critério é um nome de processo (argumento <name>). Em nosso exemplo, o agent do Zabbix o verá como 'topasrec'. Para corresponder, você precisa especificar 'topasrec' ou deixá-lo em branco. O segundo critério é um nome de usuário (argumento <user>). Para corresponder, você precisa especificar 'root' ou deixá-lo 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 deixá-la 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 do 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

Valor suportado	Descrição	Fonte na estrutura psinfo	Tenta ser compatível com
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 da 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 do 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/ZBXNEXT-1078/zabbix_age
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/ZBXNEXT-1078/zabbix_
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 do 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 conseguindo corresponder ao nome do processo do arquivo status, o agent do 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/ZBXNEXT-1078/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 do 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 do 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 do Linux

Threads não podem ser selecionados 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 do Linux são contadas pelo item proc.num[], mas não relatam memória no item proc.mem[]. Por exemplo:

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

Mas o que acontece se houver um processo de usuário com o mesmo nome de uma thread do kernel? Então poderia ser assim:

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num[] contou tanto a thread do kernel quanto o processo de usuário. proc.mem[] relata a memória apenas para o processo de usuário e conta a memória da 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 nome do programa coincidir com o de uma thread.

Antes de colocar parâmetros nos itens proc.mem[] e proc.num[], você pode querer testar os parâmetros usando o item proc.num[] e o comando ps.

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 ", depois 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", depois 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”, depois 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”, depois 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. Depois 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, em seguida, 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 de 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.

Parâmetros do item net.udp.service

ntp

Envia um pacote SNTP via UDP e valida a resposta de acordo com RFC 4330, seção 5. A porta padrão 123 é usada se não for especificada.

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
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: total de segundos de CPU (usuário)	pin: tamanho das páginas fixadas
rss: tamanho do conjunto residente	cputime_system: total de segundos de CPU (sistema)	pte: tamanho das entradas da tabela de páginas

mode=process	mode=thread	mode=summary
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: total de segundos de CPU (usuário)
lib: tamanho das bibliotecas compartilhadas		cputime_system: total de segundos de CPU (sistema)
peak: pico de 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: tamanho proporcional do conjunto residente
stk: tamanho do segmento de pilha		
swap: tamanho do espaço de swap usado		
cputime_user: total de segundos de CPU (usuário)		
cputime_system: total de segundos de CPU (sistema)		
state: estado do processo (recuperado transparentemente do procfs, forma longa)		
ctx_switches: número de trocas de contexto		
threads: número de threads		
page_faults: número de falhas de página		
pss: tamanho 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: total de segundos de CPU (usuário)
vsize: tamanho da memória virtual	tname: nome da thread	cputime_system: total de segundos de CPU (sistema)
pmem: porcentagem de memória real (apenas FreeBSD)	cputime_user: total de segundos de CPU (usuário)	ctx_switches: número de trocas de contexto

mode=process	mode=thread	mode=summary
rss: tamanho do conjunto residente	cputime_system: total de segundos de CPU (sistema)	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 utilizado
cputime_user: total de segundos de CPU (usuário)		io_read_op: número de vezes que o sistema teve que realizar entrada
cputime_system: total de segundos de CPU (sistema)		io_write_op: número de vezes que o sistema teve que realizar saída
state: estado do processo (espera de disco/executando/dormindo/parada de rastreamento/zumbi/outro)		
ctx_switches: número de trocas de contexto		
threads: número de threads (não suportado para NetBSD)		
page_faults: número de falhas de página		
fds: número de descritores de arquivos (apenas OpenBSD)		
swap: tamanho do espaço de swap utilizado		
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	pid: PID	name: nome do processo
ppid: PID pai	ppid: PID pai	processes: número de processos
name: nome do processo	name: nome do processo	vmsize: tamanho da memória virtual
user: usuário sob o qual o processo é executado	user: usuário sob o qual o processo é executado	wkset: tamanho do conjunto de trabalho do processo
sid: SID do usuário	sid: SID do usuário	cputime_user: total de segundos de CPU (usuário)
vmsize: tamanho da memória virtual	tid: ID do thread	cputime_system: total de segundos de CPU (sistema)
wkset: tamanho do conjunto de trabalho do processo		threads: número de threads
cputime_user: total de segundos de CPU (usuário)		page_faults: número de falhas de página
cputime_system: total de segundos de CPU (sistema)		handles: número de handles
threads: número de threads		io_read_b: bytes de IO lidos
page_faults: número de falhas de página		io_write_b: bytes de IO gravados
handles: número de handles		io_read_op: operações de leitura de IO
io_read_b: bytes de IO lidos		io_write_op: operações de gravação de IO

mode=process	mode=thread	mode=summary
io_write_b: bytes de IO gravados		io_other_b: bytes de IO transferidos, exceto operações de leitura e gravação
io_read_op: operações de leitura de IO		io_other_op: operações de IO, exceto operações de leitura e gravação
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

Vários parâmetros de configuração definem como o Zabbix server deve se comportar quando uma verificação de agent (Zabbix, SNMP, IPMI, JMX) falha e uma interface de host se torna inacessível.

Interface inalcançável

Uma interface de host é tratada como inalcançável 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 inalcançável, o parâmetro **UnreachableDelay** define com que frequência ela será rechecada usando um dos items (incluindo regras LLD). Essas rechecagens são realizadas por pollers de interfaces inalcançáveis (ou pollers IPMI para verificações IPMI). Por padrão, o intervalo entre verificações consecutivas de disponibilidade é de 15 segundos.

> **Attention:**
> As verificações realizadas por pollers assíncronos não são movidas para pollers de interfaces inalcançáveis.

No log do server Zabbix, a indisponibilidade 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 o item exato que falhou e seu tipo (agent Zabbix).

> **Note:**
> O parâmetro *Timeout* também afetará o quão cedo uma interface será rechecada durante a indisponibilidade. 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 indisponibilidade. 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 indisponibilidade.

Alternando a interface de volta para disponível

Quando o período de inacessibilidade termina, a interface é verificada novamente, diminuindo a prioridade para o item que tornou a interface inacessível. Se a interface inacessível reaparecer, o monitoramento retorna ao normal automaticamente:

```
retomando verificações do agent Zabbix no host "Novo host": conexão restaurada
```

Note:
Assim que a interface se torna disponível, o host não verifica todos os seus items imediatamente por dois motivos:

- Isso pode sobrecarregar o host.
- O tempo de restauração da interface nem sempre coincide com o tempo planejado de verificação do item.

Portanto, após a interface se tornar disponível, os items não são verificados imediatamente, mas são reagendados para a próxima rodada de verificação.

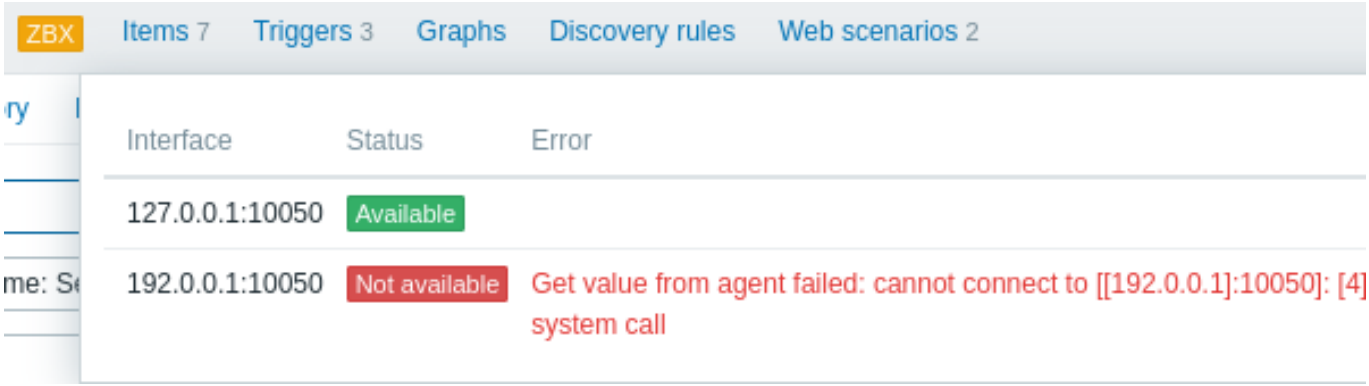
Interface indisponível

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.

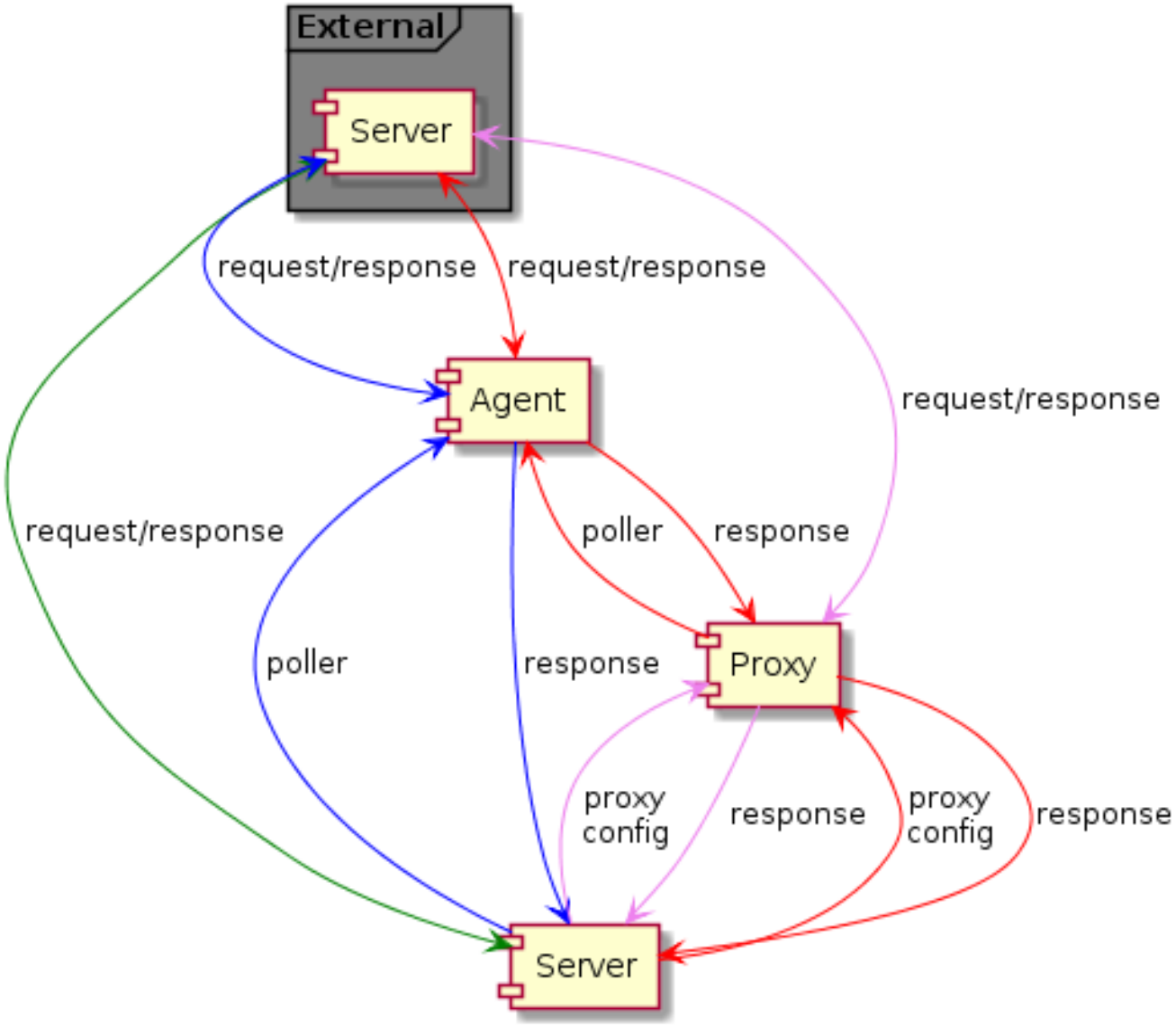
Items

Para configurar a consulta de estatísticas internas em outra instância do Zabbix, você pode usar dois items:

- item interno `zabbix[stats,<ip>,<port>]` - para consultas remotas diretas do Zabbix server/proxy. <ip> e <port> são usados para identificar a instância de destino.
- item de agent `zabbix.stats[<ip>,<port>]` - para consultas remotas baseadas em agent do Zabbix server/proxy. <ip> e <port> são usados para identificar a instância de destino.

Veja também: Itens internos, Itens de agent Zabbix

O diagrama a seguir ilustra o uso de cada item dependendo do contexto.



- - Server → instância Zabbix externa (`zabbix[stats,<ip>,<port>]`)
- - Server → proxy → instância Zabbix externa (`zabbix[stats,<ip>,<port>]`)
- - Server → agent → instância Zabbix externa (`zabbix.stats[<ip>,<port>]`)
- - Server → proxy → agent → instância Zabbix externa (`zabbix.stats[<ip>,<port>]`)

Para garantir que a instância de destino permita consultas por uma instância externa, liste o endereço da instância externa no parâmetro 'StatsAllowedIP' na instância de destino.

Métricas expostas

Os items de estatísticas coletam as estatísticas em lote e retornam um JSON, que é a base para os items dependentes obterem seus dados. As seguintes métricas internas são retornadas por qualquer um dos dois items:

- `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 das 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 o 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 de fila interna

Existem também outros dois itens que permitem consultar remotamente as estatísticas da fila interna em outra instância do

Zabbix:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` item interno - para consultas diretas da fila interna ao server/proxy Zabbix remoto
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` item agent - para consultas da fila interna baseadas em agent ao server/proxy Zabbix remoto

Veja também: Itens internos, Itens agent Zabbix

13 Configurando o Kerberos com o Zabbix

Visão geral

A autenticação Kerberos pode ser usada no monitoramento web e em items HTTP no Zabbix.

Esta seção descreve um exemplo de configuração do Kerberos com o Zabbix server para realizar o monitoramento web de `www.example.com` com o usuário 'zabbix'.

Etapas

Passo 1

Instale o pacote Kerberos.

Para Debian/Ubuntu:

```
apt install krb5-user
```

Para RHEL:

```
dnf install krb5-workstation
```

Passo 2

Configure o arquivo de configuração do Kerberos (consulte a documentação do MIT para obter detalhes)

```
cat /etc/krb5.conf
[libdefaults]
    default_realm = EXAMPLE.COM

#### As seguintes variáveis krb5.conf são apenas para o MIT Kerberos.
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true

[realms]
    EXAMPLE.COM = {
    }

[domain_realm]
    .example.com=EXAMPLE.COM
    example.com=EXAMPLE.COM
```

Passo 3

Crie um ticket Kerberos para o usuário *zabbix*. Execute o seguinte comando como usuário *zabbix*:

```
kinit zabbix
```

> **Attention:**
> É importante executar o comando acima como usuário *zabbix*. Se você executá-lo como *root*, a autenticação não irá funcionar.

Passo 4

Crie um cenário web ou um item HTTP agent com o tipo de autenticação Kerberos.

Opcionalmente, pode ser testado com o seguinte comando curl:

```
curl -v --negotiate -u : http://example.com
```

Observe que, para monitoramento web prolongado, é necessário renovar o ticket Kerberos. O tempo padrão de expiração do ticket é de 10h.

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 ao 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 de 'type' sequenciais que serão lidos 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

Parâmetro	Descrição	Padrão	Exemplo
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
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*

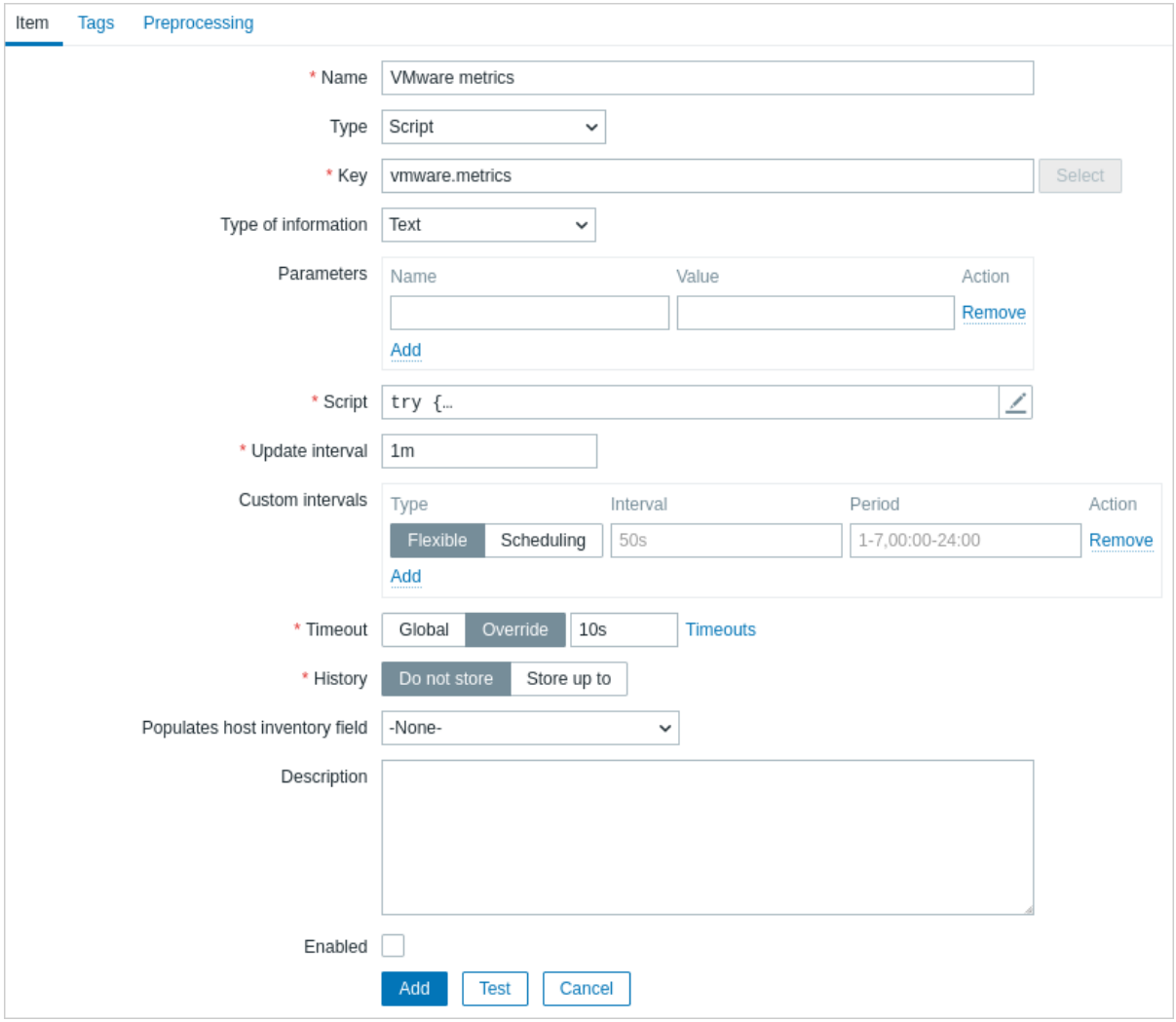
Assim, o exemplo acima resultaria no seguinte caminho do contador: `cpu/usagemhz[average]`

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*: Métricas do VMware
- *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;
```

Uma vez que o item esteja 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
grupo	//groupInfo[../key=6]/key	cpu
contador	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

O caminho resultante do contador de desempenho, neste caso, é: `cpu/usagemhz[average]`

16 Valores de retorno para system.sw.packages.get

Visão geral

Esta seção fornece detalhes sobre os valores 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 estes dados (`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 do Zabbix agent 2 `net.dns.get`.

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 items 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 relatados 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, firstclock, fuzzytime, last, lastclock, logeventid, logseverity, logsource, logtimestamp, 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 pred-iti-vas	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
- Itens calculados
- Macros de expressão

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 items 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 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)<: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 ser 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 levar a implicações de desempenho, enquanto usar **exists_foreach()**, que funciona apenas com dados de configuração, não terá tal efeito.
- Os parâmetros opcionais *operador* ou *padrão* não podem ser deixados vazios após uma vírgula, apenas totalmente omitidos.
- Com *bitand* como o 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 o 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 itens net.if.in que receberam dados na última hora até
count(last_foreach(/*/vfs.fs.dependent.size[*,pused]),"gt",95) #o número de sistemas de arquivos com mais
```

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(filtro de item)

A contagem de items existentes na configuração que correspondem aos critérios do filtro.
 Tipo de valor suportado: *Integer*.

Parâmetro:

- **filtro de item** - critérios para seleção de item, permite referência por grupo de hosts, host, chave do item e tags. Caracteres curinga são suportados. Veja filtro de item 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 "curtose" da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido. Veja também: Curtose.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
kurtosis(/host/key,1h) #curtose 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 valor 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 valor 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 valor 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 para a 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: Variância.
 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.

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, permitindo que os itens necessários sejam 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 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 ":" nele.

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 maior comodidade, 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(filtro de item,período de tempo,<número do parâmetro>)
```

onde <número do parâmetro> 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 <número do parâmetro> é ’1’.

count_foreach

Terceiro e quarto parâmetros opcionais são suportados pela função *count_foreach*:

`count_foreach(filtro de item,período de tempo,<operador>,<padrão>)`

Onde:

- **operador** é o operador condicional para valores de item (deve ser colocado 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 (diferencia maiúsculas de minúsculas)
bitand - AND bit a bit
regexp - correspondência sensível a maiúsculas/minúsculas da expressão regular fornecida em `padrão`
iregexp - correspondência insensível a maiúsculas/minúsculas da expressão regular fornecida em `padrão`
- **padrão** é o padrão necessário (argumentos de string devem estar entre aspas duplas); suportado se *operador* for especificado no terceiro parâmetro.

Comentários:

- Os parâmetros opcionais *operador* ou *padrão* não podem ser deixados em branco 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_foreach() 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 diferenciação de maiúsculas/minúsculas é herdada das configurações globais de expressão regular. 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 das funções foreach.

Comportamento dependendo da disponibilidade

A tabela a seguir ilustra como cada função se comporta em casos de disponibilidade limitada do host/item e dos 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 de função opcionais (ou partes de parâmetros) são indicados por < >

bitand(value,mask)

O valor de ”E bit a bit” de um valor de item e máscara.
 Tipos de valor 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 com 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(valor, bits para deslocar)

O deslocamento à esquerda em nível de bit de um valor de item.
 Tipos de valor suportados: *Integer*.

Parâmetros:

- **valor** - o valor a ser verificado;
- **bits para deslocar** (obrigatório) - o número de bits a serem deslocados.

Embora a comparação seja feita em nível de 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 valor 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(valor, bits para deslocar)

O deslocamento de bits para a direita de um valor de item.
 Tipos de valores suportados: *Inteiro*.

Parâmetros:

- **valor** - o valor a ser verificado;
- **bits para deslocar** (obrigatório) - o número de bits a deslocar.

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 valor 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
- Items calculados
- Macros de expressão

> **Attention:**
> 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 AAAAMMDD.

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, 1º 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.
firstclock	O timestamp do 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.
lastclock	O timestamp do N-ésimo valor mais recente dentro do período de avaliação definido.
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.
logtimestamp	O timestamp da mensagem de log do N-ésimo valor de item de log mais recente.
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 (porcentagem) é especificado pelo terceiro parâmetro.
rate	A taxa média por segundo do aumento em um contador monotonicamente crescente 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 uma cerquilha)
  - **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

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 com estes 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/key)>10
```

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 apenas as diminuições; *inc* - conta apenas os aumentos

Para tipos de valor não numéricos, o parâmetro *mode* é ignorado.

Exemplos:

```
changecount(/host/key,1w) #o número de alterações de valor na última semana até agora
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 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**. 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/key,10m) #os valores dos últimos 10 minutos até agora
count(/host/key,10m,"like","error") #o número de valores dos últimos 10 minutos até agora que contêm 'erro
count(/host/key,10m,,12) #o número de valores dos últimos 10 minutos até agora que são iguais a '12'
count(/host/key,10m,"gt",12) #o número de valores dos últimos 10 minutos até agora que são maiores que '12
count(/host/key,#10,"gt",12) #o número de valores dentro dos últimos 10 valores até agora que são maiores
```

```
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 dos 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 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**. 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>)

Procura uma correspondência de valor dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.
 Retorna: 1 - encontrado; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns;
- **sec** ou **#num** (opcional) - por padrão, utiliza 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 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 *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.

Exemplo:

```
find(/host/key,10m,"like","error") #encontra um valor que contenha 'error' nos últimos 10 minutos até agor
```

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
```

firstclock(/host/key,sec<:time shift>)

O carimbo de data/hora do valor mais antigo dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns.

A função falha com um erro se nenhum dado tiver sido coletado no período fornecido.

Veja também lastclock().

Exemplo:

```
firstclock(/host/key,1h) #recupera o carimbo de data/hora do valor mais antigo dentro da última hora até a
firstclock(/host/key,1h:now-24h) #recupera o carimbo de data/hora do valor mais antigo dentro da última ho
```

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 tempo local está sincronizado com o tempo local do Zabbix server. *Observe* que 'system.localtime' deve ser configurado como um cheque passivo.
- 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 itens 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`.

Exemplo:

```
fuzzytime(/host/key,60s)=0 #detecta um problema se a diferença de tempo for superior a 60 segundos<br><br>
```

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 em 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().

Exemplo:

```
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 diferem
```

lastclock(/host/key,<#num<:time shift»)

O carimbo de data/hora do N-ésimo valor mais recente dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns;
- **#num** (opcional) - o N-ésimo valor mais recente.

A função falha com um erro se nenhum dado tiver sido coletado no período fornecido ou se nenhum valor N-ésimo tiver sido coletado.

Veja também firstclock().

Exemplo:

```
lastclock(/host/key) #recupera o carimbo de data/hora do valor mais recente
lastclock(/host/key,#2) #recupera o carimbo de data/hora do 2º valor mais recente
lastclock(/host/key,#3:now-1d) #recupera o carimbo de data/hora do 3º valor mais recente há um dia atrás
```

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 valores 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).

logseverity(/host/key,<#num<:time shift»)

Severidade do log da última entrada de log.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - severidade padrão; N - severidade (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 - Verbose).

Parâmetros:

- Veja parâmetros comuns;
- **#num** (opcional) - o N-ésimo valor mais recente.

O Zabbix obtém a severidade do log a partir do campo **Information** do log de eventos do Windows.

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 Expressão Regular Compatível com Perl (PCRE) (argumentos de string devem estar entre aspas duplas).

Normalmente usado para logs de eventos do Windows.

Exemplo:

```
logsource(/host/key,,"VMware Server")
```

logtimestamp(/host/key,<#num<:time shift»)

O timestamp da mensagem de log do N-ésimo valor de item de log mais recente.
 Tipos de valor suportados: *Log*.

Parâmetros:

- Veja parâmetros comuns;
- **#num** (opcional) - o N-ésimo valor mais recente.

Comentários:

- O cálculo do deslocamento de tempo é baseado no clock do valor do item, não no timestamp da mensagem de log;
- A função falha com erro se:
  - um item de tipo diferente de log for recebido;
  - nenhum dado tiver sido coletado no período especificado;
  - nenhum valor N-ésimo tiver sido coletado;
  - a mensagem de log não contiver um valor de timestamp.

Exemplo:

```
logtimestamp(/host/key) #recupera o timestamp da última mensagem de log
logtimestamp(/host/key,#2) #recupera o timestamp da 2ª última mensagem de log
logtimestamp(/host/key,#3:now-1d) #recupera o timestamp da 3ª última mensagem de log de um dia atrás
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se houve uma diminuição monótona nos valores.
 Tipos de valores suportados: *Inteiro*.
 Retorna: 1 - se todos os elementos no período de tempo diminuíram continuamente; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns;
- **mode** (deve ser colocado entre aspas duplas) - *weak* (cada valor é menor ou igual ao anterior; padrão) ou *strict* (cada valor diminuiu).

Exemplo:

```
monodec(/Host1/system.swap.size[all,free],60s) + monodec(/Host2/system.swap.size[all,free],60s) + monodec(
```

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 estar entre aspas duplas) - *weak* (cada valor é maior ou igual ao anterior; padrão) ou *strict* (cada valor aumentou).

Exemplo:

```
monoinc(/Host1/system.localtime,#3,"strict")=0 #verifica se o tempo local do sistema está aumentando consi
```

nodata(/host/key,sec,<mode>)

Verifica se não há dados recebidos.
 Tipos de valor 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 se tornar indisponível, os triggers 'nodata' não serão disparados imediatamente após a restauração da conexão, mas pularão os dados pelo período atrasado. 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 reabilitado
- 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;
- a função nodata() não pode ser usada sozinha na expressão; pelo menos uma função de outro grupo, referenciando o item do host, deve ser incluída na expressão (exceto funções de data e hora). Para informações detalhadas sobre como a função nodata() funciona dentro das expressões, veja Tempo de cálculo.

percentile(/host/key,(sec|#num)<:time shift>,percentage)

O percentil P de um período, onde P (porcentagem) 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.

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.

Exemplo:

```
rate(/host/key,30s) #se o aumento monotônico em 30 segundos for 20, esta função retornará 0,67.
```

Veja todas as funções suportadas.

5 Funções de tendência

Funções de tendência, em contraste com as funções de histórico, usam dados de tendência para cálculos.

Tendências armazenam valores agregados por hora. Funções de tendência usam essas médias horárias e, portanto, são úteis para análises de longo prazo.

Os resultados das funções de tendência são armazenados em cache, então múltiplas chamadas para a mesma função com os mesmos parâmetros buscam informações do banco de dados apenas uma vez. O cache de funções de tendência é controlado pelo parâmetro do server TrendFunctionCacheSize.

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.

Função	Descrição
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 `((número de valores anômalos)/(número total de valores))`.
trendsum	A soma dos valores de tendência dentro do período de tempo definido.

Parâmetros comuns

- `/host/key` é um primeiro 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 em temporadas anteriores.

Parâmetros:

- Veja parâmetros comuns;
- **data period** - o período de coleta de dados dentro de uma temporada, 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 temporada
- **season unit** - a duração de 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 (população) entre a hora anter
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Calcula a linha de base pela média dos dados do mesmo período em vários períodos de tempo iguais ('temporadas') usando o algoritmo de média móvel ponderada.

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
Time shift - o deslocamento do período de tempo, define o final do período de coleta de dados em temporadas (veja exemplos);
- **season unit** - a duração de 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:

```
baselinewma(/host/key,1h:now/h,"d",3) #calcula a linha de base com base na última hora cheia dentro de um
baselinewma(/host/key,2h:now/h,"d",3) #calcula a linha de base com base nas duas últimas horas dentro de u
baselinewma(/host/key,1d:now/d,"M",4) #calcula a linha de base com base no mesmo dia do mês que 'ontem' no
```

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 (ex: 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 atrás
```

trendcount(/host/key,período de tempo:deslocamento de tempo)

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 atrás
```

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 atrás
```

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 o 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 o 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, o 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 (o 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 atrás
```

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 do item referenciado.
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 a uma potência de um valor.
expm1	O número de Euler elevado a uma 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 do item referenciado.
min	O menor valor dos valores do item referenciado.
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.

Função	Descrição
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 do item referenciado.
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 de função opcionais (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

O valor deve estar entre -1 e 1. 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 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 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),avg(/host2/key2))
```

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 ’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),avg(/host2/key2))
```

min(<value1>,<value2>,...)

O menor valor dos valores de item referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
min(avg(/host/key),avg(/host2/key2))
```

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 de 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 terceira 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 pseudoaleatório 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, onde o valor é um ângulo expresso em radianos.

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 falhará 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),avg(/host2/key2))
```

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 for 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 preditivas

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, máximo, mínimo, delta ou média 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 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 uma cerquilha)
  - **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)<: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 valor 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 `agora + time`; *max*, *min*, *delta* e *avg* investigam a estimativa do valor do item no intervalo entre `agora` e `agora + 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 dados de um d
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 preditivas de trigger.

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
- Items 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.

Função	Descrição
char	Retorna o caractere interpretando o valor como código ASCII.
concat	A string resultante da concatenação dos valores dos items 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 os caracteres especificados do início da string.
mid	Retorna uma substring de N caracteres começando na posição do caractere especificada por 'start'.
repeat	Repete uma string.
replace	Encontra o padrão no valor e substitui pelo valor de substituição.
right	Retorna os caracteres mais à direita do valor.
rtrim	Remove os caracteres especificados do final da string.
trim	Remove os 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 do parâmetro) são indicados por < >

ascii(value)

O código ASCII do caractere mais à esquerda do valor.
 Tipos de valores 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 valor 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: *Inteiro*.

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 float). 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 substituir;
- **replacement** - string de substituição.

Por exemplo, um valor como 'Zabbbix' será substituído por 'Zabbix' se 'bb' (posição inicial 3, posições a substituir 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 ser entre aspas);
- **default** - o valor opcional de fallback se a consulta JSONPath não retornar dados. Observe que em outros erros é 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 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 '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 valores 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 de 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)

Encontra o padrão no valor e substitui pelo valor de substituição. Todas as ocorrências do padrão serão substituídas.
 Tipos de valor 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 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 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 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, a falha é retornada (por exemplo, "expressão inválida").

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 por local

Visão geral

A tabela de macros suportadas por local contém uma referência de todas as macros (integradas, de expressão e definidas pelo usuário) que são suportadas no local específico.

Se todas as macros de um grupo relacionado forem suportadas, apenas o nome do grupo será exibido. Clique em cada nome de macro ou nome de grupo para ver todos os detalhes das macros.

A lista de detalhes das macros contém uma lista detalhada de todas as macros **integradas**, agrupadas por área de aplicação. Observe que, para personalizar os valores das macros (por exemplo, encurtar ou extrair substrings específicas), você pode usar as funções de macro.

Macros suportadas por local

Local	Macros suportadas	Comentários
Ação		
Condição de período de tempo, Duração padrão da etapa da operação, Duração da etapa	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas
Notificações e comandos		
Baseado em trigger	Macros de ação Macros de data/hora {ESC.HISTORY} Macros de evento, macros de evento de causa/sintoma Macros de atualização de evento: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Macro de expressão: {?EXPRESSION} {FUNCTION.VALUE} Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de host de destino (apenas comandos) Macros de inventário de host Macros de item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros de proxy Macros de trigger: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Macros definidas pelo usuário	Veja também: Usando macros em mensagens

Local	Macros suportadas	Comentários
Atualização de problema	Macros de ação Macros de data/hora {ESC.HISTORY} Macros de evento, macros de evento de causa/sintoma, macros de evento de recuperação Macros de atualização de evento: todas (exceto [{EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro de expressão: {?EXPRESSION} Macros de função Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de host de destino (apenas comandos) Macros de inventário de host Macros de item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros de proxy Macros de trigger: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} {USER.FULLNAME} Macros definidas pelo usuário	
Recuperação de problema	Macros de evento de recuperação {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Descoberta	Macros de ação Macros de data/hora Macros de descoberta Macros de evento: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macros de host de destino (apenas comandos) Macros de proxy	
Autoregistro	Macros de ação Macros de data/hora Macros de evento: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macros de host: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Macros de host de destino (apenas comandos) Macros de proxy	
Baseado em serviço	Macros de ação Macros de data/hora {ESC.HISTORY} Macros de evento: todas exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA} Macros de serviço Macros definidas pelo usuário	
Recuperação de serviço	Macros de evento (exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macros de evento de recuperação	

Local	Macros suportadas	Comentários
Atualização de serviço	Macros de ação Macros de data/hora {ESC.HISTORY} Macros de evento (exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Macros de atualização de evento: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.MESSAGE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Macros de serviço Macros definidas pelo usuário	
Notificações internas	Macros de ação Macros de data/hora {ESC.HISTORY} Macros de evento (exceto {EVENT.NSEVERITY}, {EVENT.SEVERITY}), macros de evento de recuperação Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros de item: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Macros de proxy	
Baseado em item	Macros de item: {ITEM.STATE}, {ITEM.STATE.ERROR}	
Baseado em regra LLD	Macros de notificação de descoberta de baixo nível	
Baseado em trigger	Macros de trigger: {TRIGGER.DESCRIPTION}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATE}, {TRIGGER.STATE.ERROR}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME} Macros definidas pelo usuário	
Parâmetros de script de alerta	Macros de alerta Macros definidas pelo usuário (apenas globais)	
Conector		
Campos URL, Nome de usuário, Senha, Token Bearer, Proxy HTTP, Certificado SSL, Arquivo de chave SSL, Senha da chave SSL	Macros definidas pelo usuário (apenas globais)	
Campo Timeout	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.

Local	Macros suportadas	Comentários
Nome do gráfico	Macro de expressão: {?EXPRESSION}	Apenas uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (sufixos de tempo podem ser usados) é suportada nesta macro neste local. A macro {HOST.HOST<1-9>} pode ser usada como host dentro da macro (veja também macros indexadas). 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)}
Host, protótipo de host		
IP/DNS da interface	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Macros definidas pelo usuário [1]	[1] Para endereço IP, uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas
Porta da interface	Macros definidas pelo usuário	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas
Comunidade SNMP *Nome do contexto SNMPv3, Nome de segurança, Senha de autenticação, Senha de privacidade* *Nome de usuário IPMI, senha*	Macros definidas pelo usuário	
Item, protótipo de item, regra LLD		
Nome	Macros definidas pelo usuário	Não suportado em nomes de regras LLD.
Descrição	Macros definidas pelo usuário	

Local	Macros suportadas	Comentários
Parâmetros da chave	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	As macros {HOST.*} suportadas serão resolvidas para a interface selecionada para o item. Quando usadas em itens sem interfaces, elas serão resolvidas para a interface Zabbix agent, SNMP, JMX ou IPMI do host nesta ordem de prioridade ou para 'UNKNOWN' se o host não tiver nenhuma interface.
Campos Intervalo de atualização, Intervalos personalizados, Timeout (disponível para tipos de item suportados), Armazenar até (para histórico/tendências)	Macros definidas pelo usuário	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Campos Excluir recursos perdidos, Desabilitar recursos perdidos	Macros definidas pelo usuário	Esses campos são suportados apenas para regras LLD. Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Parâmetros de etapa de pré-processamento (incluindo scripts personalizados), *parâmetros personalizados de tratamento de erro* (campos *Definir valor para* e *Definir erro para*)	Macros definidas pelo usuário	
Expressões regulares de filtro	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	Esses campos são suportados apenas para regras LLD.
Expressões regulares de filtro de sobrescrita	Macros definidas pelo usuário	Esses campos são suportados apenas para regras LLD.
Operações de sobrescrita: *Intervalo de atualização*, *Período de armazenamento do histórico*, *Período de armazenamento de tendências* (para protótipos de item)	Macros definidas pelo usuário	Esses campos são suportados apenas para regras LLD. Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Tipos de item		

Local	Macros suportadas	Comentários
Item **Browser** *nomes e valores de parâmetros*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros definidas pelo usuário	
Item **Script** *nomes e valores de parâmetros*		
Fórmula de item **calculado/agregado** (*constantes de expressão*, *parâmetros de função*, *parâmetros de chave de item*, (apenas item agregado) *condições de filtro* (nome do grupo de hosts e nome da tag))	Macros definidas pelo usuário	
Item **Database monitor** Campos *Nome de usuário, Senha, Consulta SQL*	Macros definidas pelo usuário	
Item **HTTP agent** Campos *URL, Campos de consulta, Corpo da requisição, Cabeçalhos (nomes e valores), Arquivo de certificado SSL, Arquivo de chave SSL*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros definidas pelo usuário	URLs que contêm uma macro secreta não funcionarão, pois a macro será resolvida como "******"
Item **HTTP agent** Campo *HTTP Proxy*	Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros definidas pelo usuário	
Item **HTTP agent** Campos *Códigos de status requeridos, Nome de usuário/senha de autenticação HTTP, Senha da chave SSL*	Macros definidas pelo usuário	
Item **HTTP agent** Campo *Hosts permitidos*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	
Item **JMX agent** Campo *Endpoint*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Macros definidas pelo usuário	
Item **Trapper** Campo *Hosts permitidos*		
Item **SNMP agent** Campo *SNMP OID*	Macros definidas pelo usuário	
Item **SSH e Telnet** *script*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	
Item **SSH agent** Campos *Nome de usuário, Arquivo de chave pública, Arquivo de chave privada, Senha*	Macros definidas pelo usuário	

Local	Macros suportadas	Comentários
Item **Telnet agent** Campos *Nome de usuário, Senha*	Macros definidas pelo usuário	
Mapa		
URL, nome da URL	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Macros de inventário de host Macros de mapa {TRIGGER.ID}	
Rótulo do elemento	Macro de expressão: {?EXPRESSION} [1] Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Macros de inventário de host Macros de mapa Macros de trigger: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] Apenas uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (sufixos de tempo podem ser usados) é suportada nesta macro neste local. A macro {HOST.HOST<1-9>} pode ser usada como host dentro da macro. 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)}
Rótulo do link	Macro de expressão: {?EXPRESSION} [1]	
Rótulo da forma	Macro de expressão: {?EXPRESSION} [1]	
Campo de texto em formas do mapa	{MAP.NAME}	
Descoberta de rede		
Campo Intervalo de atualização	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Campos Comunidade SNMP, SNMP OID *Campos Nome do contexto SNMPv3, Nome de segurança, Senha de autenticação, Senha de privacidade*	Macros definidas pelo usuário (apenas globais)	
Proxy		
Campos Endereço para porta de agentes ativos Para modo proxy passivo: campos *Endereço* e *Porta* da interface Campos *Timeouts para tipos de item* no modo de sobrescrita	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.

Local	Macros suportadas	Comentários
Grupo de proxy		
Campos Período de failover, Número mínimo de proxies	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Scripts		
Comandos, texto de confirmação (Tipo: script, **ação de host** manual)	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host [2] {MANUALINPUT} Macros de nome de usuário Macros definidas pelo usuário	[1] Apenas texto de confirmação [2] Suportado para Zabbix server e Zabbix proxy
Comandos (Tipo: script, **ação de evento** manual)	Macros de data/hora Macros de evento, macros de evento de causa/sintoma, macros de evento de recuperação (se a recuperação ocorreu) Macros de atualização de evento: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros de item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Macros de proxy Macros de trigger: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Macros de nome de usuário Macros definidas pelo usuário	
Texto de confirmação (Tipo: script, **ação de evento** manual)	Macros de evento: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host {MANUALINPUT} Macros de nome de usuário Macros definidas pelo usuário	
URL, texto de confirmação (Tipo: URL, **ação de host/evento** manual)		
Nome e valor da tag	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de trigger: {TRIGGER.ID} (apenas valor da tag de trigger) Macros definidas pelo usuário	Macros de nome e valor de tag são resolvidas apenas durante o processo de geração do evento.

Local	Macros suportadas	Comentários
Trigger		
Nome	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros/referências posicionais: $1...$9 Macros definidas pelo usuário	
Nome do evento	Macros de data/hora: {TIME}, {TIMESTAMP} Macro de expressão: {?EXPRESSION} Macros de função Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de trigger: {TRIGGER.EXPRESSION.EXPLAIN}	
Dados operacionais	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros definidas pelo usuário	
Expressão	{TRIGGER.VALUE} Macros definidas pelo usuário [1]	[1] Apenas em constantes e parâmetros de função; macros secretas não são suportadas
Campos Tag para correspondência, Nome da entrada do menu, URL da entrada do menu	Macros definidas pelo usuário	URLs que contêm uma macro secreta não funcionarão, pois a macro será resolvida como "******".
Descrição	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros definidas pelo usuário {EVENT.ID}	
URL	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Comandos de parâmetro de usuário	Macros/referências posicionais: $1...$9	
Cenário web		
Campos Nome, Agent, HTTP proxy	Macros definidas pelo usuário	
Campo Intervalo de atualização	Macros definidas pelo usuário	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Campos Variáveis, Cabeçalhos, Certificado SSL, Arquivo de chave SSL	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	

Local	Macros suportadas	Comentários
Campos Usuário, Senha, Senha da chave SSL	Macros definidas pelo usuário	
Etapa do cenário web Campos *Nome, Variáveis (apenas valores)*	Macros definidas pelo usuário	
Etapa do cenário web Campos *Timeout, Códigos de status requeridos*	Macros definidas pelo usuário	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Etapa do cenário web Campos *URL, Cabeçalhos (nomes e valores), String requerida*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	
Etapa do cenário web Campo *Post*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Tipo de mídia webhook		
Valores de parâmetro	Macros de alerta Todas as macros suportadas em notificações de problema baseadas em trigger (veja acima)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Script JavaScript, nome do parâmetro do script JavaScript	Macros definidas pelo usuário (apenas globais)	
Nome da entrada do menu, URL da entrada do menu	{EVENT.TAGS.<tag name>}	
Widgets		
Widget **Gauge** Campo *Descrição*	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros de item (exceto {ITEM.STATE.ERROR}) Macros definidas pelo usuário (apenas globais)	
Widget **Honeycomb** Campo *Rótulo primário/secundário* Widget **Valor do item** Campo *Descrição*		
Widget **Top hosts** Coluna *Dados de texto*	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros de inventário de host Macros definidas pelo usuário (apenas globais)	
Widget **URL** (Dinâmico) Campo *URL*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros definidas pelo usuário	
Outros locais		

Local	Macros suportadas	Comentários
Campo Horário de trabalho	Macros definidas pelo usuário (apenas globais)	Uma única macro deve preencher todo o campo; múltiplas macros/mistura com texto não são suportadas.
Campo Timeouts para tipos de item (configuração global) Mídia de usuário Campo *Quando ativo*		
Tipo de mídia Email Campos *Nome de usuário, Senha*	Macros definidas pelo usuário (apenas globais)	Macros secretas são recomendadas.
Relatórios agendados Campos *Assunto, Mensagem*	Macros de data/hora: {TIME}, {TIMESTAMP}	

Detalhes da macro

Esta lista contém detalhes para cada macro interna agrupada por área de aplicação:

- Macros de ação
- Macros de alerta
- Macros de data/hora
- Macros de descoberta
- Macros de evento
  - Macros de atualização de evento
  - Macros de evento de causa/sintoma
  - Macros de evento de recuperação
- Macros de escalonamento
- Macros de função
- Macros de host
  - Macros de host de destino
- Macros de grupo de host
- Macros de inventário de host
- Macros de item
- Macros de notificação de descoberta de baixo nível
- Macros de mapa
- Macros de proxy
- Macros de script
- Macros de serviço
- Macros de trigger
- Macros de nome de usuário

Macros de ação

{ACTION.ID}

O ID numérico da ação disparada.

{ACTION.NAME}

O nome da ação disparada.

Macros de alerta

{ALERT.MESSAGE}

O valor de *Mensagem padrão* da configuração da ação.

{ALERT.SENDTO}

O valor *Enviar para* da configuração de mídia do usuário.

{ALERT.SUBJECT}

O valor de *Assunto padrão* da configuração da ação.

Macros de data e hora

{DATE}

A data atual no formato `aaaa.mm.dd.`.

{TIME}

A hora atual no formato `hh:mm:ss`.

{TIMESTAMP}

A hora atual no formato de timestamp UNIX.

Macros de descoberta

{DISCOVERY.DEVICE.IPADDRESS}

O endereço IP do dispositivo descoberto.
 Sempre disponível, não depende do host ser adicionado.

{DISCOVERY.DEVICE.DNS}

O nome DNS do dispositivo descoberto.
 Sempre disponível, não depende do host ser adicionado.

{DISCOVERY.DEVICE.STATUS}

O status do dispositivo descoberto (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

O tempo desde a última alteração do status de descoberta para um determinado dispositivo.
 Por exemplo: 1h 29m 01s.
 Para dispositivos com status DOWN, este é o período de seu tempo de inatividade.

{DISCOVERY.RULE.NAME}

O nome da regra de descoberta que detectou a presença ou ausência do dispositivo/serviço.

{DISCOVERY.SERVICE.NAME}

O nome do serviço descoberto.
 Por exemplo: HTTP.

{DISCOVERY.SERVICE.PORT}

A porta do serviço descoberto.
 Por exemplo: 80.

{DISCOVERY.SERVICE.STATUS}

O status do serviço descoberto (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

O tempo desde a última alteração do status de descoberta para um serviço específico.
 Por exemplo: 1h 29m 01s.
 Para serviços com status DOWN, este é o período de indisponibilidade.

Macros de escalonamento

{ESC.HISTORY}

O log de notificações enviadas anteriormente, seu passo de escalonamento e status (*enviado*, *em andamento* ou *falhou*).

Macros de evento

{EVENT.ACK.STATUS}

O status de reconhecimento do evento (Sim/Não).

{EVENT.AGE}

A idade do evento que acionou uma ação, com precisão de até um segundo.
 Útil em mensagens escalonadas. Por exemplo: 9m 13s.

{EVENT.DATE}

A data do evento que acionou uma ação.
 Por exemplo: 2025.04.14.

{EVENT.DURATION}

A duração do evento (diferença de tempo entre os eventos de problema e de recuperação), com precisão de até um segundo.
 Útil em mensagens de recuperação de problemas.

{EVENT.ID}

O ID numérico do evento que acionou uma ação.

{EVENT.NAME}

O nome do evento que acionou uma ação.

{EVENT.NSEVERITY}

O valor numérico da severidade do evento.
 Valores possíveis: *0* - Não classificado, *1* - Informação, *2* - Aviso, *3* - Médio, *4* - Alto, *5* - Desastre.

{EVENT.OBJECT}

O valor numérico do objeto do evento.
 Valores possíveis: *0* - Trigger, *1* - Host descoberto, *2* - Serviço descoberto, *3* - Autoregistro, *4* - Item, *5* - Regra de descoberta de baixo nível.

{EVENT.OPDATA}

Os dados operacionais do trigger subjacente do problema.

{EVENT.SEVERITY}

O nome da severidade do evento.

{EVENT.SOURCE}

O valor numérico da fonte do evento.
 Valores possíveis: *0* - Trigger, *1* - Descoberta, *2* - Autoregistro, *3* - Interno, *4* - Serviço.

{EVENT.STATUS}

O valor verbal do evento que acionou uma ação.

{EVENT.TAGS}

Uma lista separada por vírgulas de tags de evento.
Expandida para uma string vazia se não existirem tags.

{EVENT.TAGSJSON}

Um array JSON contendo objetos de tags de evento.
Expandido para um array vazio se não existirem tags.

{EVENT.TAGS.<tag name>}

O 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.

{EVENT.TIME}

A hora do evento que acionou uma ação.
 Por exemplo: 12:57:53.

{EVENT.TIMESTAMP}

O timestamp UNIX do evento que acionou uma ação.

{EVENT.VALUE}

O valor numérico do evento que acionou uma ação.
 Valores possíveis: *1* - problema, *0* - recuperação.

Macros de causa/sintoma de evento

{EVENT.CAUSE.ACK.STATUS}

O status de reconhecimento do evento de causa (Sim/Não).
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.AGE}

A idade do evento de causa, com precisão de até um segundo.
 Útil em mensagens de escalonamento.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.DATE}

A data do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.DURATION}

A 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 problemas.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.ID}

O ID numérico do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.NAME}

O nome do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.NSEVERITY}

O valor numérico da severidade do evento de causa.
 Valores possíveis: *0* - Não classificado, *1* - Informação, *2* - Aviso, *3* - Médio, *4* - Alto, *5* - Desastre.
 Usado no contexto de um evento sintoma.

{EVENT.CAUSE.OBJECT}

O 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.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.OPDATA}

Os dados operacionais do trigger subjacente do problema de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.SEVERITY}

O nome da gravidade do evento de causa.
 Valores possíveis: *Não classificado*, *Informação*, *Aviso*, *Média*, *Alta*, *Desastre*.
 Usado no contexto de um evento sintoma.

{EVENT.CAUSE.SOURCE}

O valor numérico da origem do evento de causa.
 Valores possíveis: *0* - Trigger, *1* - Descoberta, *2* - Autoregistro, *3* - Interno.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.STATUS}

O valor verbal do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.TAGS}

Uma lista separada por vírgulas de tags do evento de causa.
Expandido para uma string vazia se não existirem tags.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.TAGSJSON}

Um array JSON contendo objetos de tag de evento de causa objects.
Expandido para um array vazio se não existirem tags.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.TAGS.<tag name>}

O 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.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.TIME}

A hora do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.TIMESTAMP}

O timestamp UNIX do evento de causa.
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.UPDATE.HISTORY}

O log de atualizações do problema de causa (reconhecimentos, etc).
 Usado no contexto de um evento de sintoma.

{EVENT.CAUSE.VALUE}

O valor numérico do evento de causa
 Valores possíveis: *1* - Problema, *0* - Recuperação.
 Usado no contexto de um evento de sintoma.

{EVENT.SYMPTOMS}

A lista de eventos de sintoma. Inclui os seguintes detalhes: nome do host, nome do evento, severidade, idade, tags e valores de serviço.
 Esta macro é usada no contexto do evento de causa e retorna informações sobre eventos de sintoma.

Macros de evento de recuperação

{EVENT.RECOVERY.DATE}

A data do evento de recuperação.

{EVENT.RECOVERY.ID}

O ID numérico do evento de recuperação.

{EVENT.RECOVERY.NAME}

O nome do evento de recuperação.

{EVENT.RECOVERY.STATUS}

O valor verbal do evento de recuperação.

{EVENT.RECOVERY.TAGS}

Uma lista separada por vírgulas de tags de eventos de recuperação. Expandido para uma string vazia se não existirem tags.

{EVENT.RECOVERY.TAGSJSON}

Uma matriz JSON contendo objetos de tag de evento de recuperação objects. Expandido para uma matriz vazia se não existirem tags.

{EVENT.RECOVERY.TIME}

A hora do evento de recuperação.

{EVENT.RECOVERY.TIMESTAMP}

O timestamp UNIX do evento de recuperação.

{EVENT.RECOVERY.VALUE}

O valor numérico do evento de recuperação.

Macros de atualização de evento

{EVENT.UPDATE.ACTION}

Nome legível por humanos da(s) ação(ões) executada(s) durante a atualização do problema.
 É resolvido para os seguintes valores: *reconhecido*, *não reconhecido*, *comentado*, *gravidade alterada de (gravidade original) para (gravidade atualizada)* e *fechado* (dependendo de quantas ações são executadas em uma atualização).

{EVENT.UPDATE.ACTIONJSON}

Uma matriz JSON contendo detalhes da(s) ação(ões) executada(s) durante a atualização do problema.
 Possíveis valores de propriedade JSON:
- true (para as propriedades `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` e `symptom`);
- <string de mensagem> (para a propriedade `message`);
- timestamp (para as propriedades `suppress_until` e `timestamp`) ou 0 (para a propriedade `suppress_until` se suprimido indefinidamente);
- 0, 1, 2, 3, 4, 5 (para as propriedades de gravidade `old` e `new`).

 Por exemplo: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851199,"timestamp":1730822048}`

{EVENT.UPDATE.DATE}

A data da atualização do evento (reconhecimento, etc)
Nome obsoleto: {ACK.DATE}

{EVENT.UPDATE.HISTORY}

O log de atualizações de problemas (reconhecimentos, etc).
 Nome obsoleto: {EVENT.ACK.HISTORY}

{EVENT.UPDATE.MESSAGE}

A mensagem de atualização do problema.
 Nome obsoleto: {ACK.MESSAGE}

{EVENT.UPDATE.NSEVERITY}

O valor numérico da nova severidade do evento definida durante a operação de atualização do problema.

{EVENT.UPDATE.SEVERITY}

O nome da nova severidade do evento definida durante a operação de atualização do problema.

{EVENT.UPDATE.STATUS}

O valor numérico do status de atualização do problema.
 Valores possíveis: *0* - O webhook foi chamado devido a um evento de problema/recuperação, *1* - Operação de atualização.

{EVENT.UPDATE.TIME}

A hora da atualização do evento (reconhecimento, etc)
Nome obsoleto: {ACK.TIME}

{EVENT.UPDATE.TIMESTAMP}

O timestamp UNIX da atualização do evento (reconhecimento, etc)

Macros de função

{FUNCTION.VALUE}

O valor da N-ésima função baseada em item na expressão do trigger no momento do evento.
 Apenas funções com `/host/key` como primeiro parâmetro são contadas.

 Essa 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.

{FUNCTION.RECOVERY.VALUE}

O valor da N-ésima função baseada em item na expressão de recuperação no momento do evento.*
 Apenas funções com `/host/key` como 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.

Macros de host

{HOST.CONN}

O endereço IP ou nome DNS do host, dependendo das configurações do host.
 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 estiver definida, a interface principal SNMP será usada. Se uma interface SNMP também não estiver definida, a interface principal JMX será usada. Se uma interface JMX também não estiver definida, a interface principal IPMI será usada. Se o host não tiver nenhuma interface, a macro será resolvida como 'UNKNOWN'.

 Esta macro pode ser usada 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}

A descrição do host.

 Esta macro pode ser usada com um índice numérico como {HOST.DESCRIPTION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{HOST.DNS}

O nome DNS do host.
 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 estiver definida, a interface principal do SNMP será usada. Se uma interface SNMP também não estiver definida, a interface principal do JMX será usada. Se uma interface JMX também não estiver definida, a interface principal do IPMI será usada. Se o host não tiver nenhuma interface, a macro será resolvida para 'UNKNOWN'.

 Esta macro pode ser usada com um índice numérico como {HOST.DNS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{HOST.HOST}

O nome técnico do host.
 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})`.
 {HOSTNAME<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico como {HOST.HOST<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{HOST.ID}

O ID do host.

 Esta macro pode ser usada 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.

{HOST.IP}

O endereço IP do host.
 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 estiver definida, a interface principal do SNMP será usada. Se uma interface SNMP também não estiver definida, a interface principal do JMX será usada. Se uma interface JMX também não estiver definida, a interface principal do IPMI será usada. Se o host não tiver nenhuma interface, a macro será resolvida para 'UNKNOWN'.
 {IPADDRESS<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico como {HOST.IP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{HOST.METADATA}

A metadata do host.

 Usado apenas para autoregistro de agent ativo.

{HOST.NAME}

O nome visível do host.

 Esta macro pode ser usada com um índice numérico como {HOST.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{HOST.PORT}

A porta do host (agent).
 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 estiver definida, a interface SNMP principal será usada. Se uma interface SNMP também não estiver definida, a interface JMX principal será usada. Se uma interface JMX também não estiver definida,

a interface IPMI principal será usada. Se o host não tiver nenhuma interface, a macro será resolvida como 'UNKNOWN'.

 Esta macro pode ser usada com um índice numérico como {HOST.PORT<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macros do host de destino

{HOST.TARGET.CONN}

O endereço IP ou nome DNS do host de destino.

{HOST.TARGET.DNS}

O nome DNS do host de destino.

{HOST.TARGET.HOST}

O nome técnico do host de destino.

{HOST.TARGET.IP}

O endereço IP do host de destino.

{HOST.TARGET.NAME}

O nome visível do host de destino.

Macros de grupo de hosts

{HOSTGROUP.ID}

O ID do grupo de hosts.

Macros de inventário de host

{INVENTORY.ALIAS}

O campo *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.

{INVENTORY.ASSET.TAG}

O campo *Asset tag* 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}

O campo *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}

O campo *Contato* no inventário do host.
 {PROFILE.CONTACT<1-9>} está obsoleto.

 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.

{INVENTORY.CONTRACT.NUMBER}

O campo *Número do 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}

O campo *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}

O campo *Hardware* no inventário do host.
 {PROFILE.HARDWARE<1-9>} está obsoleto.

 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.

{INVENTORY.HARDWARE.FULL}

O campo *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}

O campo *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}

O campo *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.

{INVENTORY.HOST.ROUTER}

O campo *Host router* 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}

O campo *Hardware architecture* 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}

O campo *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.

{INVENTORY.HW.DATE.EXPIRY}

O campo *Data de expiração da manutenção de 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}

O campo *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}

O campo *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.

{INVENTORY.INSTALLER.NAME}

O campo *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}

O campo *Localização* no inventário do host.
 {PROFILE.LOCATION<1-9>} está obsoleto.

 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.

{INVENTORY.LOCATION.LAT}

O campo *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.

{INVENTORY.LOCATION.LON}

O campo *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}

O campo *Endereço MAC A* no inventário do host.
 {PROFILE.MACADDRESS<1-9>} está obsoleto.

 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.

{INVENTORY.MACADDRESS.B}

O campo *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.

{INVENTORY.MODEL}

O campo *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}

O campo *Nome* no inventário do host.
 {PROFILE.NAME<1-9>} está obsoleto.

 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.

{INVENTORY.NOTES}

O campo *Notas* no inventário do host.
 {PROFILE.NOTES<1-9>} está obsoleto.

 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.

{INVENTORY.OOB.IP}

O campo *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}

O campo *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}

O campo *OOB router* 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.

{INVENTORY.OS}

O campo *SO* no inventário do host.
 {PROFILE.OS<1-9>} está obsoleto.

 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.

{INVENTORY.OS.FULL}

O campo *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}

O campo *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.

{INVENTORY.POC.PRIMARY.CELL}

O campo *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}

O campo *E-mail do POC principal* 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}

O campo *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.

{INVENTORY.POC.PRIMARY.NOTES}

O campo *Notas 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}

O campo *Telefone A do POC primário* no inventário do host.

 Essa 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}

O campo *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.

{INVENTORY.POC.PRIMARY.SCREEN}

O campo *Nome da 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.CELL}

O campo *Celular do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.CELL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.EMAIL}

O campo *E-mail do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.EMAIL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.NAME}

O campo *Nome do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.NOTES}

O campo *Notas do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.NOTES<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.PHONE.A}

O campo *Telefone secundário do POC A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.PHONE.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.PHONE.B}

O campo *Telefone B do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.PHONE.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.POC.SECONDARY.SCREEN}

O campo *Nome da tela do POC secundário* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.SECONDARY.SCREEN<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SERIALNO.A}

O campo *Número de série A* no inventário do host.
 {PROFILE.SERIALNO<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SERIALNO.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SERIALNO.B}

O campo *Número de série B* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SERIALNO.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.ADDRESS.A}

O campo *Endereço do site A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.ADDRESS.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.ADDRESS.B}

O campo *Endereço do local B* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.ADDRESS.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.ADDRESS.C}

O campo *Endereço do local C* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.ADDRESS.C<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.CITY}

O campo *Cidade do local* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.CITY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.COUNTRY}

O campo *País do local* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.COUNTRY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.NOTES}

O campo *Notas do local* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.NOTES<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.RACK}

O campo *Localização do rack do local* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.RACK<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.STATE}

O campo *Estado/província do local* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.STATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SITE.ZIP}

O campo *Site ZIP/postal* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SITE.ZIP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE}

O campo *Software* no inventário do host.
 {PROFILE.SOFTWARE<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.APP.A}

O campo *Aplicativo de software A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.APP.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.APP.B}

O campo *Aplicativo de software B* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.APP.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.APP.C}

O campo *Aplicativo de software C* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.APP.C<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.APP.D}

O campo *Aplicativo de software D* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.APP.D<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.APP.E}

O campo *Aplicativo de software E* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.APP.E<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.SOFTWARE.FULL}

O campo *Software (detalhes completos)* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.SOFTWARE.FULL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.TAG}

O campo *Tag* no inventário do host.
 {PROFILE.TAG<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.TAG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.TYPE}

O campo *Tipo* no inventário do host.
 {PROFILE.DEVICETYPE<1-9>} está obsoleto.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.TYPE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.TYPE.FULL}

O campo *Tipo (Detalhes completos)* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.TYPE.FULL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.URL.A}

O campo *URL A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.URL.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.URL.B}

O campo *URL B* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.URL.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.URL.C}

O campo *URL C* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.URL.C<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{INVENTORY.VENDOR}

O campo *Fornecedor* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.VENDOR<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macros de item

{ITEM.DESCRIPTION}

A descrição do N-ésimo item na expressão do trigger que causou uma notificação.

 Esta macro pode ser usada com um índice numérico como {ITEM.DESCRIPTION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{ITEM.DESCRIPTION.ORIG}

A descrição original (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 como {ITEM.DESCRIPTION.ORIG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{ITEM.ID}

O ID numérico do N-ésimo item na expressão do trigger que causou uma notificação.

 Esta macro pode ser usada com um índice numérico como {ITEM.ID<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{ITEM.KEY}

A chave do N-ésimo item na expressão do trigger que causou uma notificação.
 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}).<br> {TRIGGER.KEY}` está obsoleta.

 Esta macro pode ser usada com um índice numérico como {ITEM.KEY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{ITEM.KEY.ORIG}

A chave original (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 como {ITEM.KEY.ORIG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

{ITEM.LASTVALUE}

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 *Max history display period* (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 no 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}).<br>` 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)}<br><br>` 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.LASTVALUE.AGE}

O tempo decorrido entre a coleta do último valor do item e a avaliação da macro.
 Exibido em formato legível por humanos (por exemplo, 1m 45s).

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LASTVALUE.AGE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.LASTVALUE.DATE}

A data em que o último valor do item foi coletado.
 Exibido no formato `YYYYMMDD`.
 O fuso horário do servidor é usado em nomes de triggers (em Monitoramento -> lista de Problemas), nomes de eventos e nomes e valores de tags. Em todos os outros casos, o fuso horário do usuário é usado.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LASTVALUE.DATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.LASTVALUE.TIME}

A hora em que o último valor do item foi coletado.
 Exibido no formato `HHMMSS`.
 O fuso horário do servidor é usado em nomes de triggers (em Monitoramento -> Lista de problemas), nomes de eventos e nomes e valores de tags. Em todos os outros casos, o fuso horário do usuário é usado.

 Esta macro pode ser usada com um índice numérico, por exemplo,

{ITEM.LASTVALUE.TIME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.LASTVALUE.TIMESTAMP}

O timestamp UNIX quando o último valor do item foi coletado.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LASTVALUE.TIMESTAMP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.LOG.AGE}

A 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.

{ITEM.LOG.DATE}

A 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}

O ID do evento no log de eventos.
 Apenas 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}

A severidade numérica do evento no log de eventos.
 Apenas 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}

A severidade verbal do evento no log de eventos.
 Apenas para monitoramento de log de eventos do Windows.

 Essa 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}

A fonte do evento no log de eventos.
 Apenas 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.

{ITEM.LOG.TIME}

A hora em que a entrada do 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.LOG.TIMESTAMP}

O timestamp UNIX quando a entrada do log foi gravada no log.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.TIMESTAMP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.NAME}

O 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}

O nome original (com macros não 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}

O último estado do N-ésimo item na expressão do trigger que causou uma notificação.
 Valores possíveis: *Não suportado*, *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}

A 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 estar vazio.

{ITEM.VALUE}

Resolvido para:
1) 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.
2) 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 histórico já tiver sido excluído ou nunca tiver sido armazenado.
No segundo caso, e apenas no frontend, será resolvido como *UNKNOWN* se o valor histórico mais recente 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.VALUE.AGE}

O tempo decorrido entre a coleta do valor do item e a avaliação da macro.
 Exibido em um formato legível por humanos (por exemplo, 1m 45s).

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUE.AGE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.VALUE.DATE}

A data em que o valor do item foi coletado.
 Exibido no formato `YYYYMMDD`.
 O fuso horário do server é usado em nomes de triggers (em Monitoramento -> Lista de problemas), nomes de eventos e nomes e valores de tags. Em todos os outros casos, o fuso horário do usuário é usado.

Essa macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUE.DATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.VALUE.TIME}

A hora em que o valor do item foi coletado.
 Exibido no formato `HHMMSS`.
 O fuso horário do server é usado em nomes de triggers (em Monitoramento -> Lista de Problemas), nomes de eventos e nomes e valores de tags. Em todos os outros casos, o fuso horário do usuário é usado.

Essa macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUE.TIME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.VALUE.TIMESTAMP}

O timestamp UNIX quando o valor do item foi coletado.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUE.TIMESTAMP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.VALUETYPE}

O 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.

Macros de notificação de descoberta de baixo nível

{LLDRULE.DESCRIPTION}

A descrição da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.DESCRIPTION.ORIG}

A descrição original (com macros não resolvidas) da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.ID}

O ID numérico da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.KEY}

A chave da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.KEYORIG}

A chave original (com macros não resolvidas) da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.NAME}

O nome da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.NAMEORIG}

O nome original (com macros não resolvidas) da regra de descoberta de baixo nível que causou uma notificação.

{LLDRULE.STATE}

O último estado da regra de descoberta de baixo nível.
 Valores possíveis: *Não suportado*, *Normal*.

 Esta macro pode ser usada com um índice numérico, por exemplo, {LLDRULE.STATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

{LLDRULE.STATE.ERROR}

A mensagem de erro com detalhes do motivo pelo qual a regra de descoberta de baixo nível ficou não suportada.
 Se uma regra de descoberta de baixo nível entrar no estado não suportado e, em seguida, imediatamente voltar a ser suportada, o campo de erro pode ficar vazio.

Macros de mapa

{MAP.ID}

O ID do mapa de rede.

{MAP.NAME}

O nome do mapa de rede.

Macros de proxy

{PROXY.DESCRIPTION}

A descrição do proxy.
 Resolve para:
 1) proxy do N-ésimo item na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui.
 2) proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.DESCRIPTION} aqui, sem indexação.
 3) 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.

{PROXY.NAME}

O nome do proxy.
 Resolve para:
 1) proxy do N-ésimo item na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui.
 2) proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.NAME} aqui, sem indexação.
 3) 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.

Macros de script

{MANUALINPUT}

O valor de entrada manual especificado pelo usuário no momento da execução do script.

Macros de serviço

{SERVICE.DESCRIPTION}

A descrição do serviço com macros resolvidas.

{SERVICE.ID}

O ID numérico do serviço que acionou uma ação.

{SERVICE.NAME}

O nome do serviço com macros resolvidas.

{SERVICE.ROOTCAUSE}

A lista de eventos de problema de trigger que causaram a falha de um serviço, classificados por severidade e nome do host.
 Inclui os seguintes detalhes: nome do host, nome do evento, severidade, idade, tags de serviço e valores.

{SERVICE.TAGS}

Uma lista separada por vírgulas de tags de eventos de serviço.
 As tags de eventos de serviço podem ser definidas na seção de tags de configuração do serviço. Expandido para uma string vazia se não existirem tags.

{SERVICE.TAGSJSON}

Uma matriz JSON contendo objetos de tags de eventos de serviço.
 As tags de eventos de serviço podem ser definidas na seção de tags da configuração do serviço. Expandido para uma matriz vazia se não existirem tags.

{SERVICE.TAGS<tag name>}

O valor da tag do evento de serviço referenciado pelo nome da tag.
 As tags de evento de serviço podem ser definidas na seção de configuração de tags do serviço.
 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.

Macros de trigger

{TRIGGER.DESCRIPTION}

A 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.

{TRIGGER.EVENTS.ACK}

O 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}

O número de eventos de problema reconhecidos para todos os triggers, desconsiderando seu estado.

{TRIGGER.EVENTS.PROBLEM.UNACK}

O número de eventos de problema não reconhecidos para todos os triggers, desconsiderando seu estado.

O número de eventos não reconhecidos para um elemento de mapa em mapas, ou para o trigger que gerou o evento atual em notificações.

{TRIGGER.EXPRESSION}

A expressão do trigger.

{TRIGGER.EXPRESSION.EXPLAIN}

Uma 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 depurar expressões de trigger.

{TRIGGER.EXPRESSION.RECOVERY}

A 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 será retornada.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Uma 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 depurar expressões de recuperação de trigger.

{TRIGGER.HOSTGROUP.NAME}

Uma lista ordenada (por consulta SQL), separada por vírgula e espaço, dos grupos de hosts nos quais o trigger está definido.

{TRIGGER.ID}

O ID numérico do trigger que disparou esta ação.

{TRIGGER.NAME}

O nome do trigger com macros resolvidas.
 Observe que `{EVENT.NAME}` pode ser usado em ações para exibir o nome do evento/problema disparado com macros resolvidas.

{TRIGGER.NAME.ORIG}

O nome original do trigger (com macros não resolvidas).

{TRIGGER.NSEVERITY}

A severidade numérica do trigger.
 Valores possíveis: *0* - Não classificado, *1* - Informação, *2* - Aviso, *3* - Médio, *4* - Alto, *5* - Desastre.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

O número de eventos de problema reconhecidos para triggers em estado de problema.

O número de eventos de problema não reconhecidos para triggers em estado de problema.

{TRIGGER.SEVERITY}

O nome da severidade do trigger.
 Pode ser definido em *Administração* > *Geral* > *Opções de exibição do trigger*.

{TRIGGER.STATE}

O último estado do trigger.
 Valores possíveis: *Unknown*, *Normal*.

{TRIGGER.STATE.ERROR}

A 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}

O valor do trigger no momento da execução da etapa da operação.
 Valores possíveis: *Problem*, *OK*.
 `{STATUS}` está obsoleto.

{TRIGGER.TEMPLATE.NAME}

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}

A URL do trigger.

{TRIGGER.URL.NAME}

O rótulo para a URL do trigger.

{TRIGGER.VALUE}

O valor numérico atual do trigger.
 Valores possíveis: *0* - o trigger está no status OK, *1* - o trigger está no status de Problema.

{TRIGGERS.ACK}

O número de triggers reconhecidos para um elemento do mapa, desconsiderando o estado do trigger.
 Um trigger é considerado reconhecido se todos os seus eventos de problema forem reconhecidos.

{TRIGGERS.PROBLEM.ACK}

O número de triggers de problema reconhecidos para um elemento do mapa.
 Um trigger é considerado reconhecido se todos os seus eventos de problema forem reconhecidos.

{TRIGGERS.PROBLEM.UNACK}

O número de triggers de problema não reconhecidos para um elemento do mapa.
 Um trigger é considerado não reconhecido se pelo menos um de seus eventos de problema não for reconhecido.

{TRIGGERS.UNACK}

O número de triggers não reconhecidos para um elemento do mapa, desconsiderando o estado do trigger.
 Um trigger é considerado não reconhecido se pelo menos um de seus eventos de problema não for reconhecido.

Macros de nome de usuário

{USER.FULLNAME}

O nome, sobrenome e nome de usuário do usuário que adicionou o reconhecimento do evento ou iniciou o script.

{USER.NAME}

O nome do usuário que iniciou o script.

{USER.SURNAME}

O sobrenome do usuário que iniciou o script.

{USER.USERNAME}

O nome de usuário do usuário que iniciou o script.
 `{USER.ALIAS}` está obsoleto.

Macros indexadas

A sintaxe de macro indexada **{MACRO<1-9>}** só pode referenciar o N-ésimo item, função ou host no campo *Expressão* do *trigger*:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** são resolvidos para os endereços IP do primeiro, segundo e terceiro hosts na expressão do *trigger* (se presentes).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** são resolvidos 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 resolvidos para os valores da primeira, segunda e terceira função baseada 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 resolvido 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:
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 as macros definidas 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

Localização		Múltiplas macros/misturar com texto[1]
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 protótipo de item, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/mistura 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 *Item calculado/agregado*	sim

Localização	Múltiplas macros/mistura com texto[1]
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
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

Localização	Múltiplas macros/mistura com texto[1]
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

Localização	Múltiplas macros/misturar com texto[1]
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
Database monitor	
Nome de usuário	sim
Senha	sim
Consulta SQL	sim
JMX agent	
Endpoint JMX	sim
HTTP agent	
URL[3]	sim
Campos de consulta	sim
Corpo da requisição	sim
Cabeçalhos (nomes e valores)	sim
Códigos de status requeridos	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 (se o proxy pertence ao 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 configuração de 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

Localização	Múltiplas macros/mistura com texto[1]
Nome da entrada do menu	sim
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/mistura 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/misturar 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 *Valor do item* 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* → *Mídia*	
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
Token Bearer	sim
Timeout	não
Proxy HTTP	sim
Arquivo de certificado SSL	sim
Arquivo de chave SSL	sim
Senha da chave SSL	sim

Localização	Múltiplas macros/misturar 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 definido 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.

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 proibidas codificada, consistindo em `ms`, `rpm`, `RPM`, `%`. Essa 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 em branco 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.

Exemplos

Horário de trabalho. Segunda a sexta-feira das 9:00 às 18:00:

`1-5,09:00-18:00`

Horário de trabalho mais fim de semana. Segunda a sexta-feira das 9:00 às 18:00 e sábado, domingo 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 em ambas as 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 (em Unix) ou um job (em 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 espera até que o processo filho inicial saia ou ocorra o timeout
7. Se o processo filho inicial saiu e o timeout não foi atingido, o Zabbix verifica o código de saída do processo filho inicial e o compara com 0 (valor diferente de zero é considerado como falha de 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 feito e toda a árvore de processos (ou seja, o grupo de processos ou o job) é terminada

> **Attention:**
> O Zabbix assume que um comando/script terminou o processamento quando o processo filho inicial saiu E nenhum outro processo ainda está mantendo o handle/descritor de arquivo de saída aberto. Quando o processamento é concluído, TODOS os processos criados são terminados.

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 no 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 razões desconhecidas).

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 8.0, o agent Zabbix não deve ser mais antigo que a versão 1.4 e não deve ser mais recente que a 8.0.

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 uso reduzido de memória, utilize o agent mais recente suportado.

Notas para Windows XP

- No Windows XP 32 bits, não utilize agents 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.

Agents 2 suportados

Agents 2 do Zabbix mais antigos, a partir da versão 4.4, são compatíveis com o Zabbix 8.0; o agent 2 do Zabbix não deve ser mais recente que o 8.0.

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 8.0, os proxies devem corresponder à versão principal do server. Apenas proxies Zabbix 8.0.x são totalmente compatíveis com um server Zabbix 8.0.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	Inferior a 6.0; superior a 6.4
7.0	7.0	6.0, 6.2, 6.4	Inferior a 6.0; superior a 7.0
7.2	7.2	7.0	Inferior a 7.0; superior a 7.2
7.4	7.4	7.0	Inferior a 7.0; superior a 7.4
8.0	8.0	7.0, 7.2, 7.4	Inferior a 7.0; superior a 8.0

Funcionalidade suportada por proxies:

Versão do proxy	Atualização de dados	Atualização de configuração	Tarefas
Atual	Sim	Sim	Sim

Versão do proxy	Atualização de dados	Atualização de configuração	Tarefas
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 8.0.

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, os aplicativos 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 aplicativos 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 "Sem criptografia" em *Criptografia* e "Arquivo" em *Empacotamento*. 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 Biblioteca Python para Zabbix API

Visão geral

zabbix_utils é uma biblioteca Python para:

- trabalhar com a API do Zabbix;
- agir como Zabbix sender;
- agir como Zabbix get.

É suportada para o Zabbix 5.0, 6.0, 6.4 e posteriores.

14 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.

15 Outros problemas

Login e systemd

Recomendamos criar um usuário *zabbix* como usuário de sistema, ou seja, sem permissão para login. Alguns usuários ignoram essa recomendação e usam a mesma conta para fazer login (por exemplo, usando SSH) no host que executa o Zabbix. Isso pode causar a queda do daemon do Zabbix ao sair. Nesse caso, você verá algo como o seguinte no log do servidor Zabbix:

```
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 agent 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` configurada em `/etc/systemd/logind.conf`. Quando você sai do sistema, os semáforos criados anteriormente pelo Zabbix são removidos, o que causa a queda.

Uma citação da documentação do systemd:

RemoveIPC=

Controla se os objetos IPC System V e POSIX pertencentes ao usuário devem ser removidos quando o usuário sair completamente. Aceita um argumento booleano. Se ativado, o usuário não poderá consumir recurs após o término da última sessão do usuário. Isso cobre semáforos System V, memória compartilhada e filas de mensagens, bem como memória compartilhada POSIX e filas de mensagens. Observe que os objetos IP usuário root e outros usuários do sistema estão excluídos do efeito desta configuração. O padrão é "yes".

Existem 2 soluções para esse problema:

1. (recomendado) Pare de usar a conta *zabbix* para qualquer coisa além dos processos do Zabbix, crie uma conta dedicada para outras tarefas.
2. (não recomendado) Defina `RemoveIPC=no` em `/etc/systemd/logind.conf` e reinicie o sistema. Observe que `RemoveIPC` é um parâmetro global, alterá-lo afetará todo o sistema.

Usando o frontend do Zabbix atrás de um proxy

Se o frontend do Zabbix estiver sendo executado atrás de um servidor proxy, o caminho do cookie no arquivo de configuração do proxy precisa ser reescrito para corresponder ao caminho do proxy reverso. Veja exemplos abaixo. Se o caminho do cookie não for reescrito, os usuários podem enfrentar problemas de autorização ao tentar fazer login no frontend do Zabbix.

Exemplo de configuração para nginx

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Exemplo de configuração para Apache

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

16 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.
Processos de monitoramento		
Processos	Um processo de verificação 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 verificações net.tcp.service* de HTTPS, LDAP. Adicionalmente, o agent 2 fornece solução de monitoramento nativa do Zabbix para: PostgreSQL, Redis.
Concorrência	Verificações ativas para um único server são executadas sequencialmente.	Verificações de diferentes plugins ou múltiplas verificações 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

17 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;
- Barras invertidas e 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 barra invertida;
- 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.

Instale o agent do Zabbix O agent do Zabbix é o processo responsável pela coleta de dados.

Verifique a versão do seu server Zabbix:

```
zabbix_server -V
```

Instale o agent do 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.

Configurar 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 (e a porta opcional) do seu Zabbix server ao parâmetro server. Por exemplo:

```
Server=192.0.2.0:10051
```

O Zabbix agent usará este endereço para aceitar conexões de entrada apenas dos Zabbix servers ou proxies especificados para recuperar dados.

3. Reinicie o Zabbix agent.

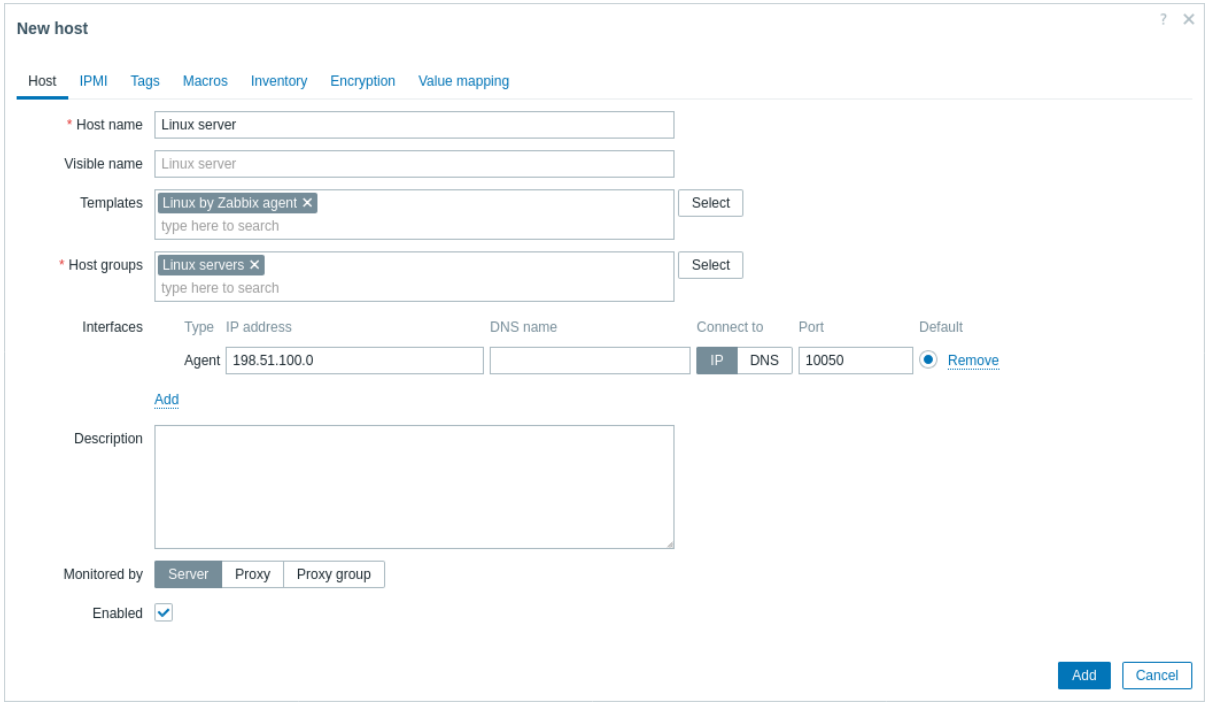
```
systemctl restart zabbix-agent
```

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 para o 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 o 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 frontend do Zabbix. 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.

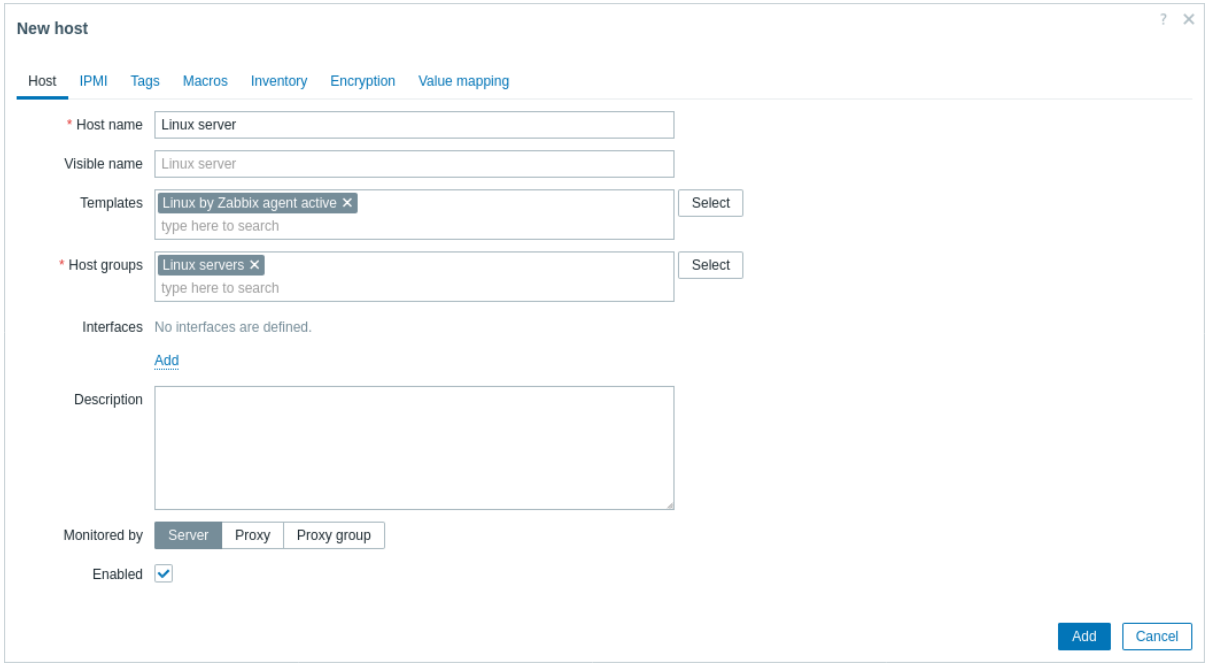
```
systemctl restart zabbix-agent
```

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.



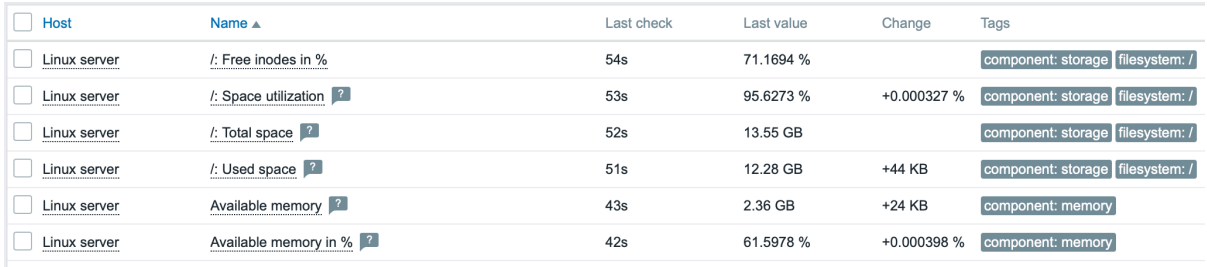
3. Clique em *Adicionar* para adicionar o host. Este host representará a máquina Linux monitorada.

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.



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 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.

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 agent Zabbix, Itens do agent Zabbix para Windows - lista completa de métricas que você pode monitorar usando o agent Zabbix 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 componentes do Zabbix usando pacotes oficiais RPM e DEB para várias distribuições Linux, garantindo acesso aos recursos mais recentes e correções de bugs.

2 Monitorar Windows com o Zabbix agent

Introdução Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de máquinas Windows com o Zabbix.

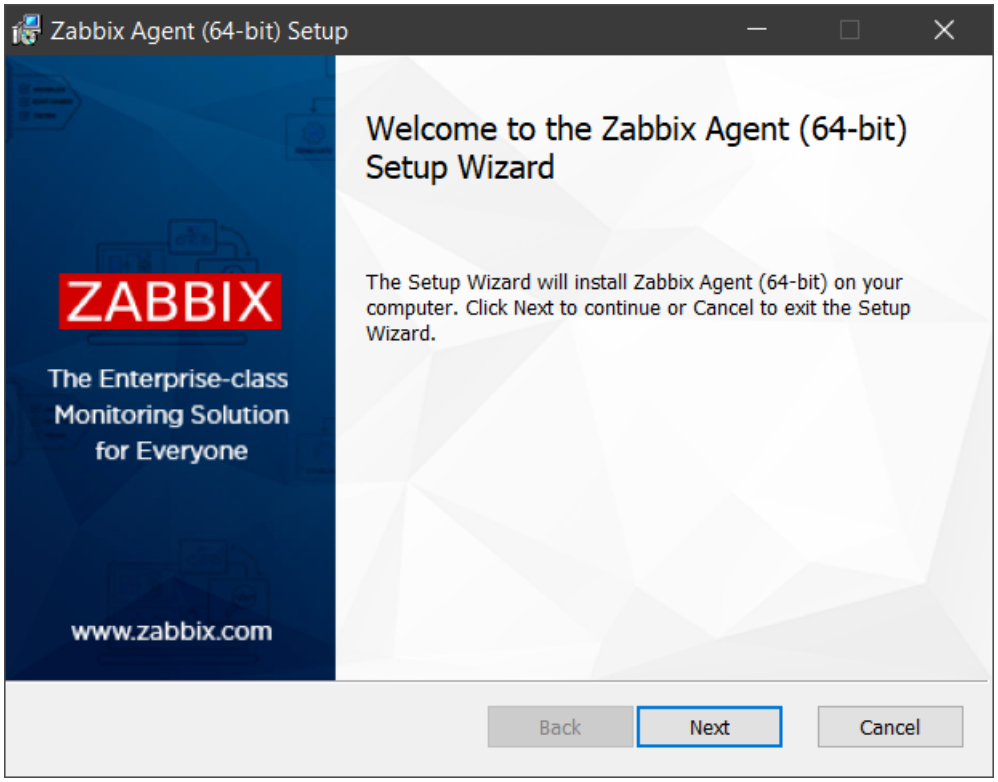
Para quem é este guia

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 Zabbix O agent 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 Zabbix para Windows.



Configurar o Zabbix para monitoramento O agent do Zabbix pode coletar métricas nos modos ativo ou passivo (simultaneamente).

Note:
Uma verificação passiva é uma solicitação simples 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 o agent do Zabbix e um template para o 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 Frontend do Zabbix

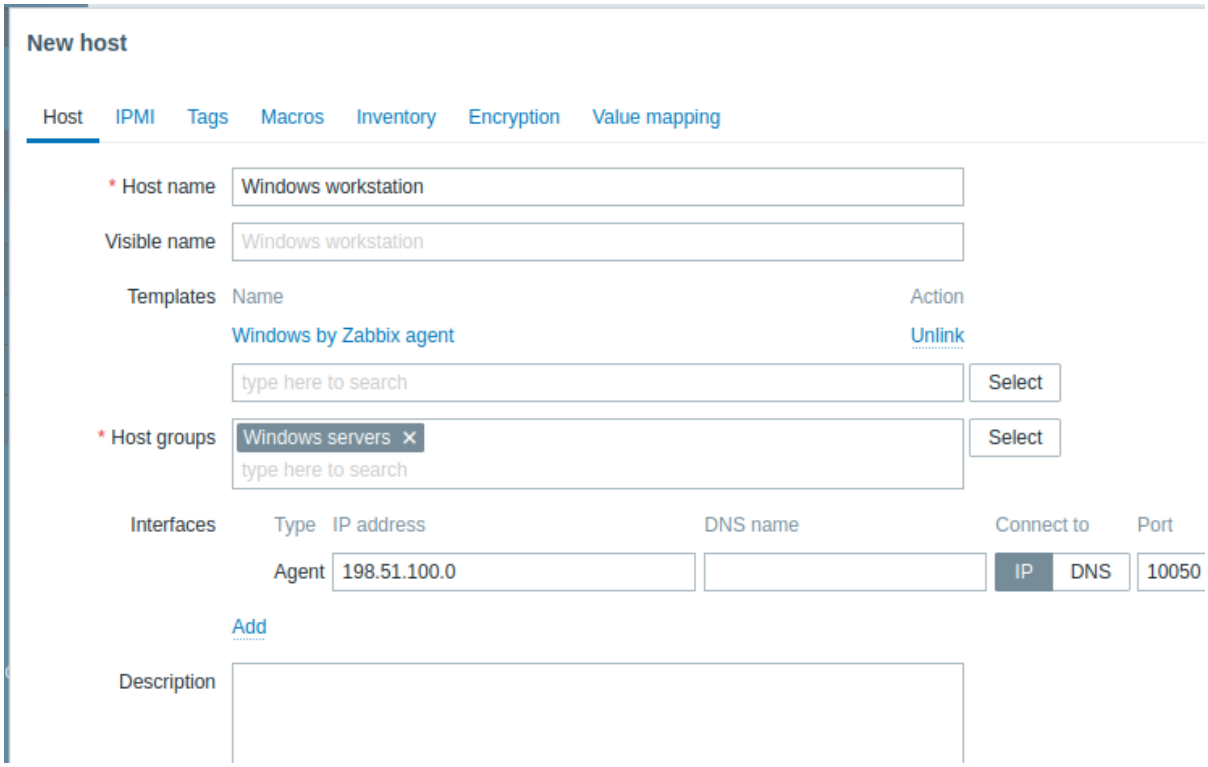
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

Este host representará sua máquina Windows.

3. No parâmetro *Interfaces*, adicione a interface *Agent* e especifique o endereço IP ou o nome DNS da máquina Windows onde o agent está instalado.

4. No parâmetro *Templates*, digite ou selecione *Windows by Zabbix agent*.



Agent Zabbix

Para verificações passivas, o agent Zabbix precisa saber o endereço IP ou o nome DNS do server Zabbix. Se você forneceu as informações corretas durante a instalação do agent, o arquivo de configuração já está atualizado. Caso contrário, você precisa especificá-lo manualmente. Vá para a pasta `C:\Program files\Zabbix Agent`, abra o arquivo *zabbix_agentd.conf* e adicione o IP/DNS do seu server Zabbix ao parâmetro *Server*.

Exemplo:

```
Server=192.0.2.22
```

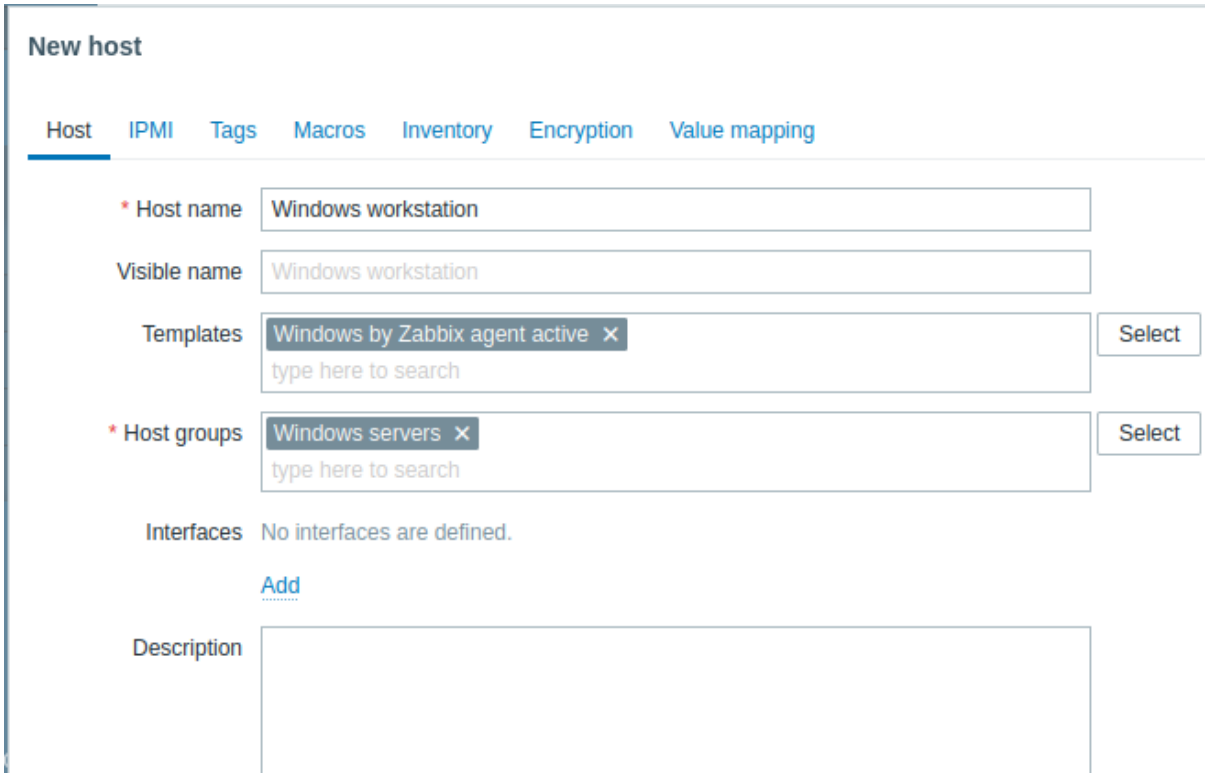
Checagens ativas Frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

Este host irá representar sua máquina Windows.

3. No parâmetro *Templates*, digite ou selecione *Windows by Zabbix agent active*.



Zabbix agent

Na pasta `C:\Program files\Zabbix Agent` abra o arquivo *zabbix_agentd.conf* e adicione:

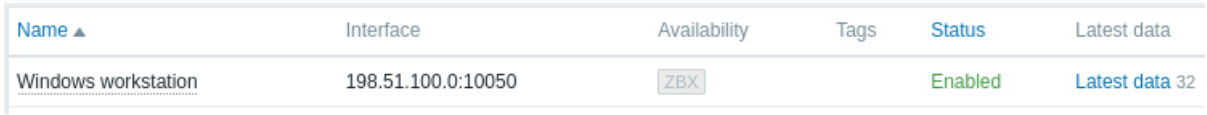
- O nome do host que você criou na interface web do Zabbix ao parâmetro *Hostname*.
- O endereço IP ou nome DNS da máquina do seu Zabbix server ao parâmetro *ServerActive* (pode estar preenchido se você o forneceu durante a instalação do Zabbix agent).

Exemplo:

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

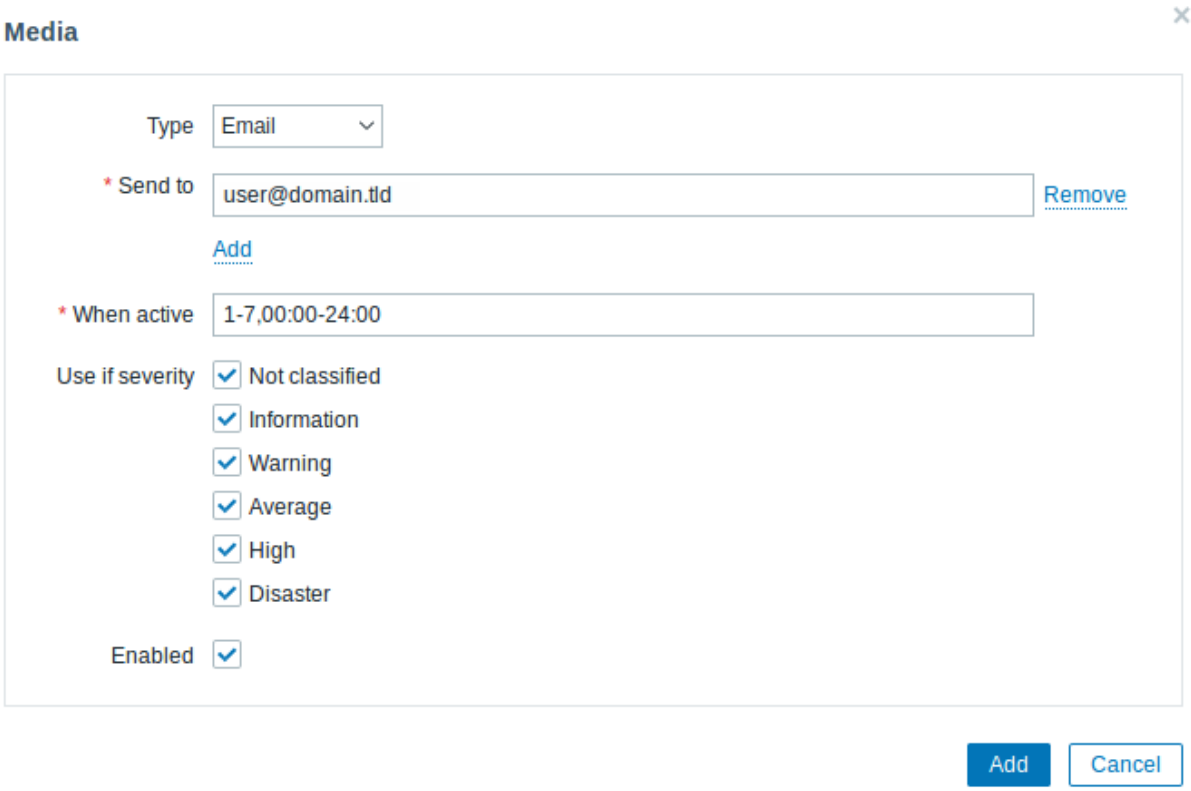
Visualizar métricas coletadas Parabéns! Neste ponto, o Zabbix já está monitorando sua máquina Windows.

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 a ferramenta 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 ao administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem ao 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 avançada, 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
```

Em Debian/Ubuntu, execute:

```
apache2 -v
```

2. Certifique-se de que o Status module está habilitado em sua instância do Apache.

Em sistemas baseados em RHEL, execute:

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

Em 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
```

Em 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
```

Em 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
```

Em 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.

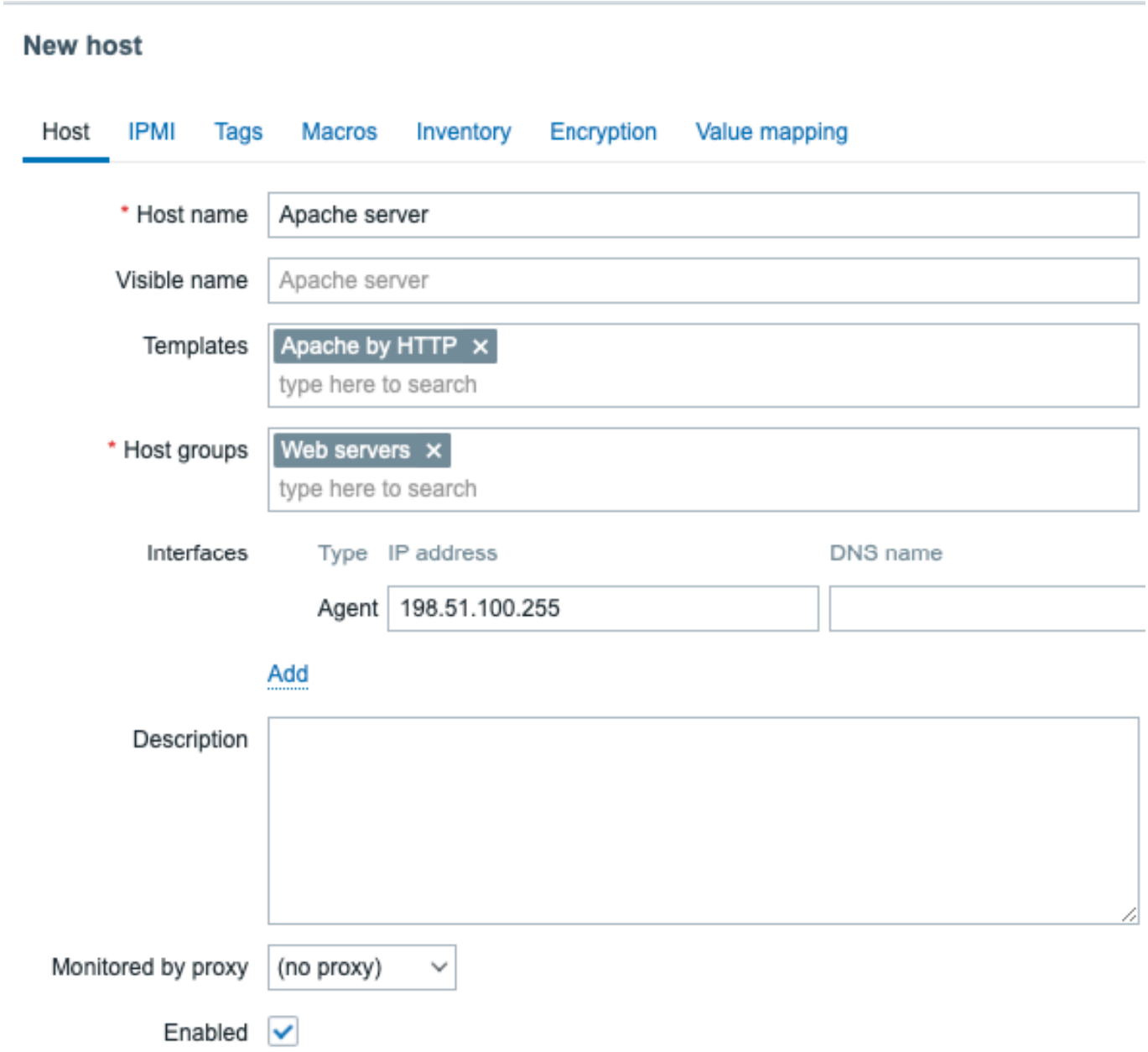
Configurar o Zabbix para monitoramento 1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

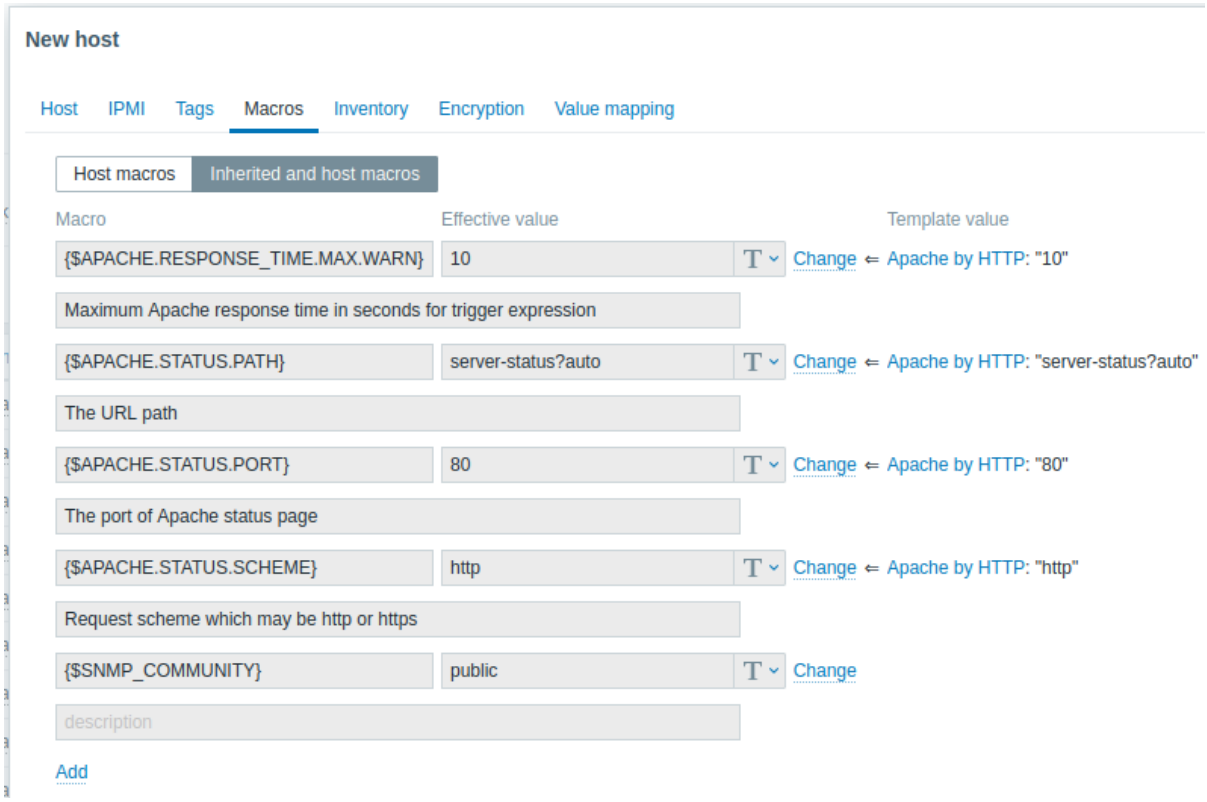
Este host representará seu servidor Apache.

3. No parâmetro *Interfaces*, adicione a interface *Agent* e especifique o endereço IP da sua instância 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 Apache.

4. No parâmetro *Templates*, digite ou selecione *Apache by HTTP*.



5. Mude para a aba **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.

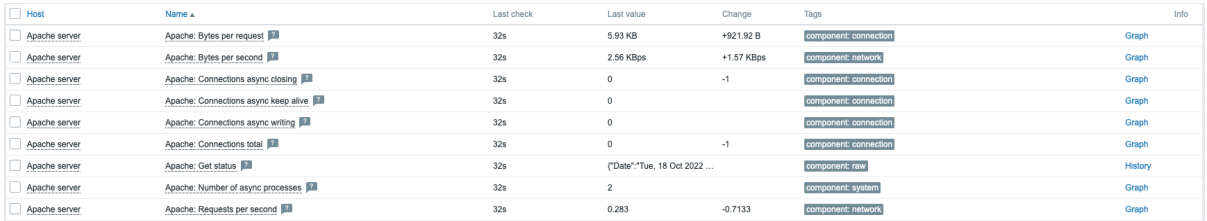
Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento->Hosts* e clique em *Dashboards* ao lado do host.



Esta ação o levará ao dashboard do host com as métricas mais importantes coletadas da página Apache /server-status.

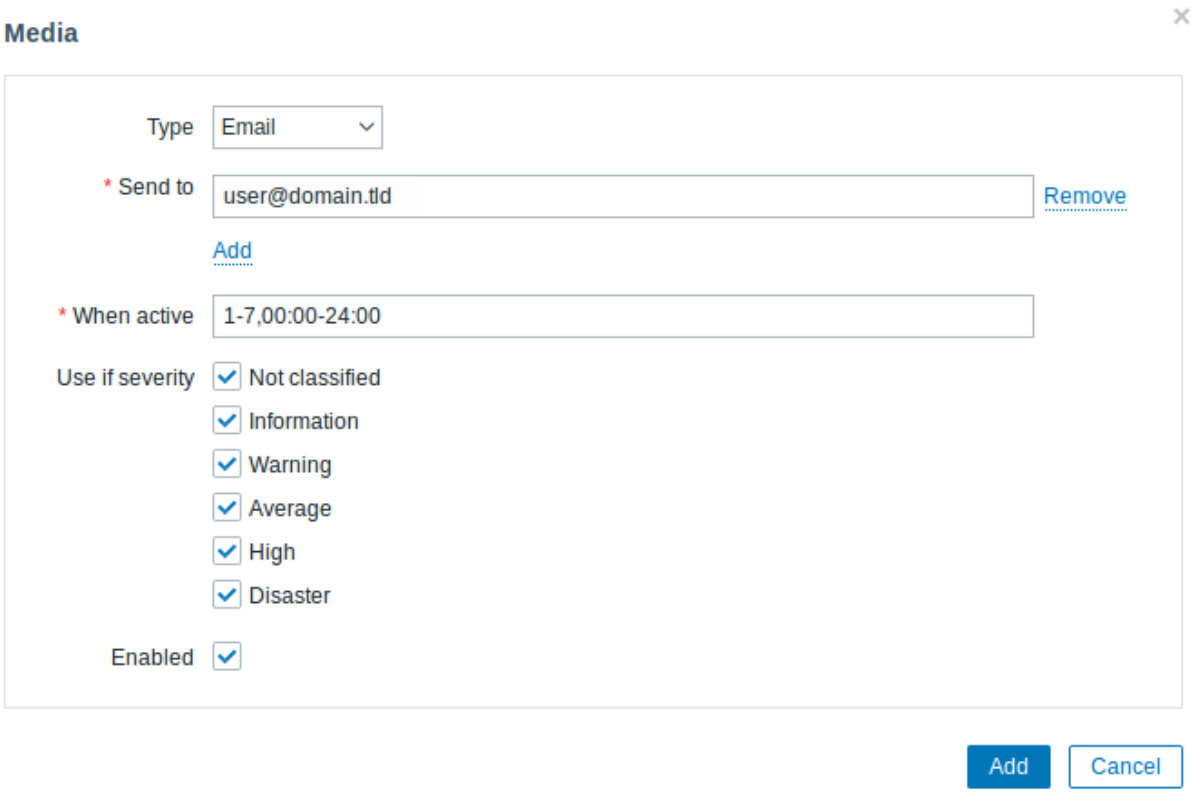


Alternativamente, em *Monitoramento->Hosts*, você pode clicar em *Últimos dados* para visualizar todas as métricas coletadas mais recentes em uma lista.



Configurar alertas de problemas O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando uma variedade de métodos. Este guia fornece as etapas de configuração para o envio de alertas por e-mail.

1. Vá para *Configurações do usuário -> Perfil*, alterne 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 Para simular um problema real e receber um alerta de problema de teste:

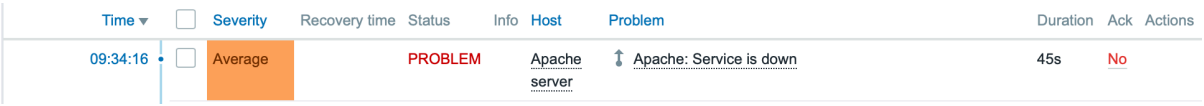
1. Abra a configuração do host *Apache server* no Zabbix.

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:

- Reforço 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 HTTP agent.
- 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).

4 Monitorar o 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, sinta-se 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.

Com base na sua configuração, algumas das etapas deste guia podem ser ligeiramente diferentes. Este guia é baseado na seguinte configuração:

- Versão do Zabbix: Zabbix 7.2 PRE-RELEASE (instalado a partir de pacotes)
- Distribuição do SO: Ubuntu
- Versão do SO: 22.04 (Jammy)
- Componentes do Zabbix: Server, Frontend, Agent 2
- Banco de dados: MySQL
- Servidor web: Apache

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, esse usuário possui mais privilégios do que o necessário para um monitoramento simples (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;
```

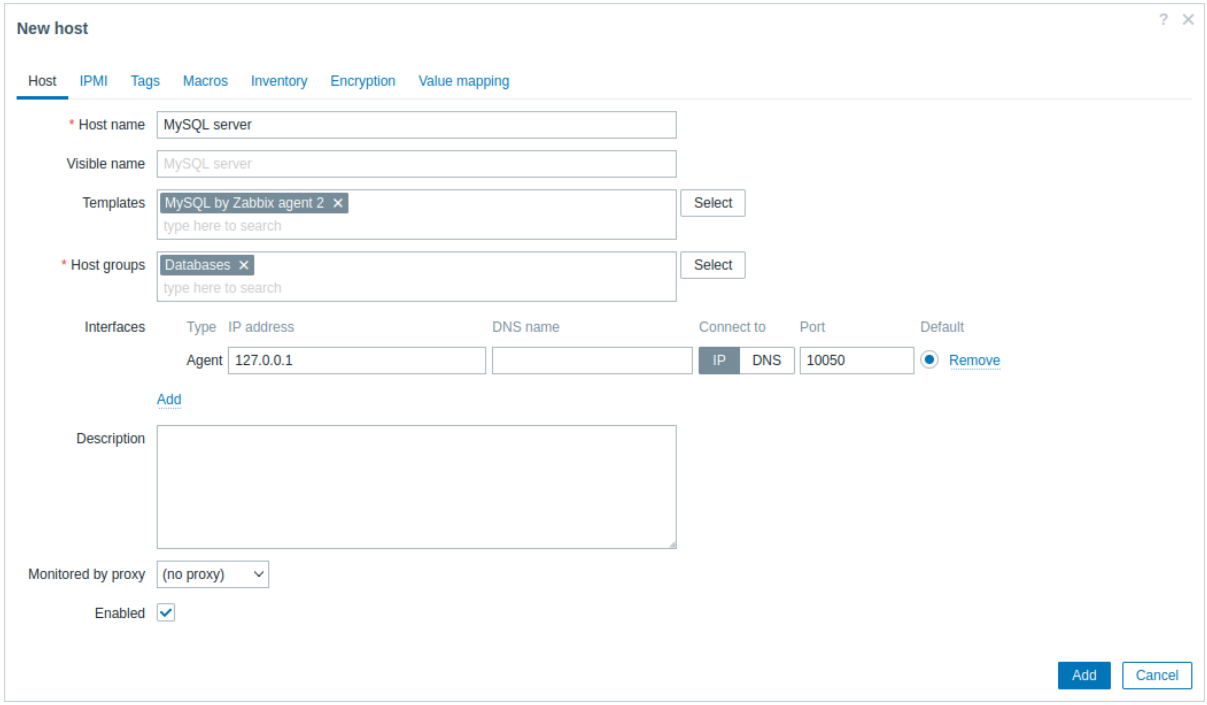
Depois que o usuário for criado, você pode passar para a próxima etapa.

Configure o frontend do Zabbix

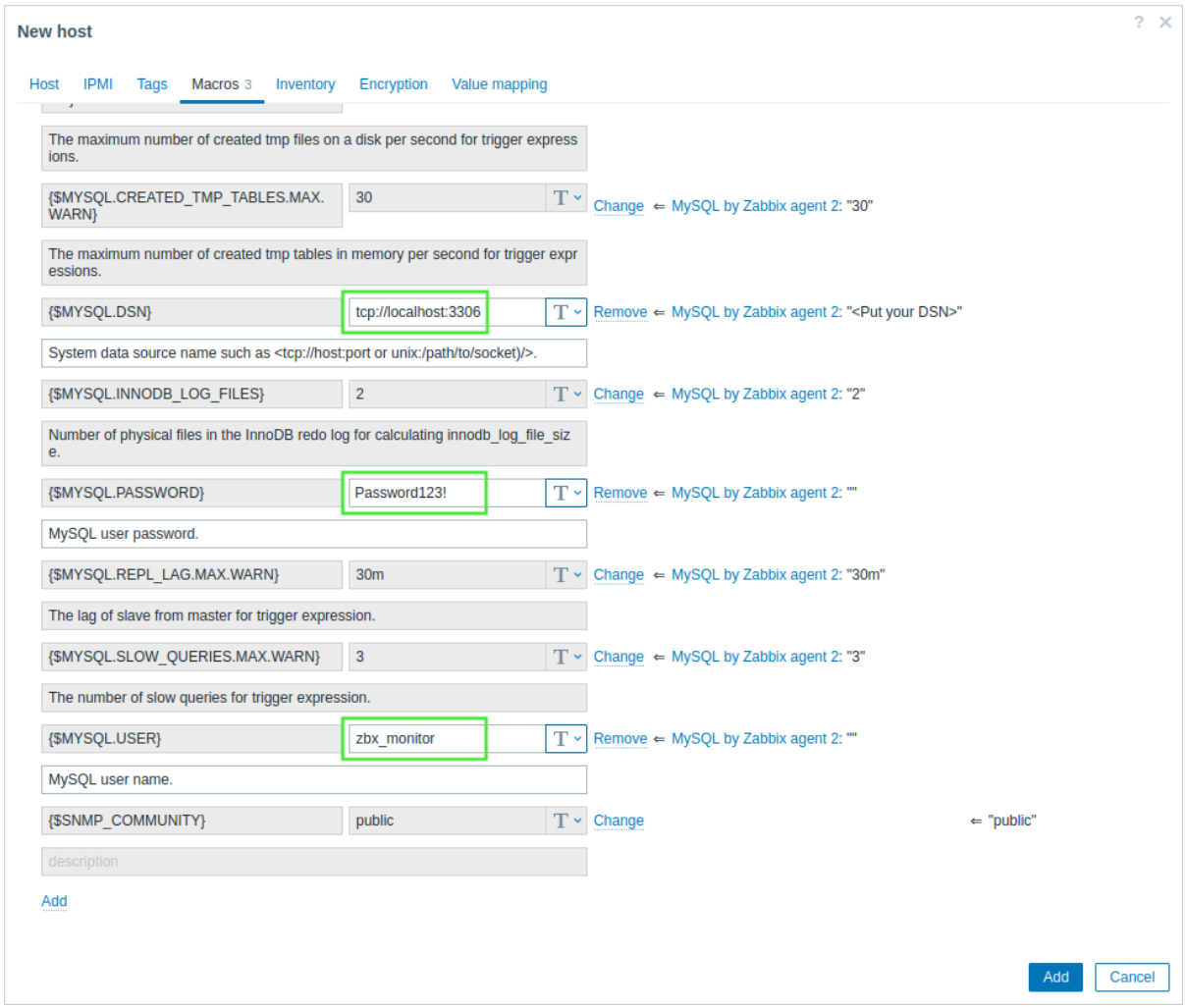
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Nome do host*, insira um nome de host (por exemplo, "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 utiliza "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:
  - {$MYSQL.DSN} - defina a fonte de dados do servidor MySQL (a string de conexão de uma sessão nomeada do arquivo de configuração do plugin MySQL Zabbix agent 2). Este guia utiliza a fonte de dados padrão "tcp://localhost:3306" para monitorar um servidor MySQL que está instalado na mesma máquina que o Zabbix server e o Zabbix agent 2.
  - {$MYSQL.PASSWORD} - defina a senha do usuário MySQL criado anteriormente "zbx_monitor".
  - {$MYSQL.USER} - defina o nome do usuário MySQL criado anteriormente "zbx_monitor".

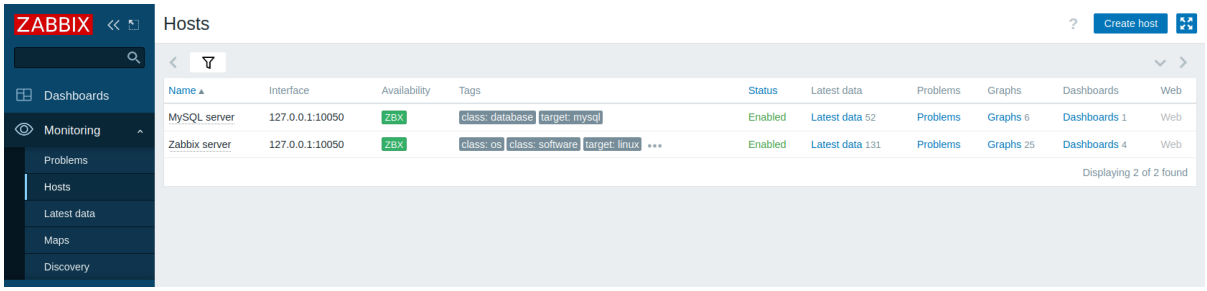


3. Clique em *Adicionar* para adicionar o host. Este host representará seu servidor MySQL.

Visualizar métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando seu servidor MySQL.

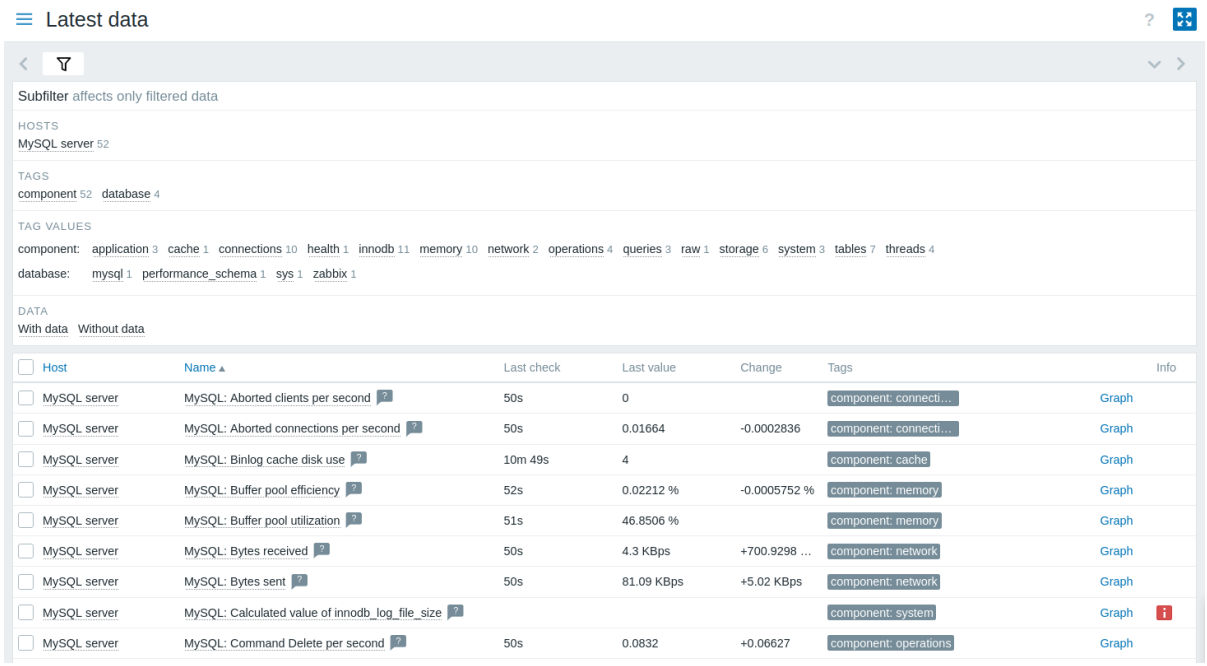
Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento → Hosts* e clique em *Dashboards* ao lado do host.



Esta ação o levará ao dashboard do host (configurado no nível do template) com as métricas mais importantes coletadas do servidor MySQL.



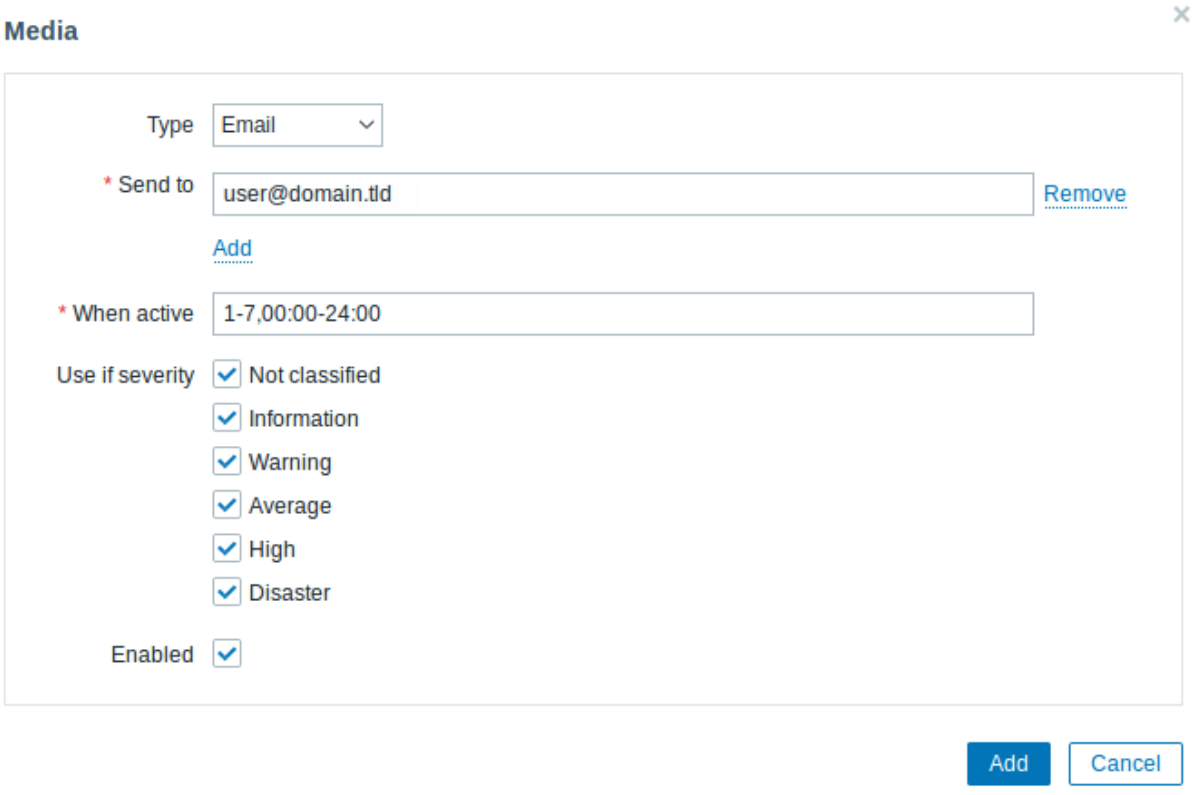
Alternativamente, a partir da seção de menu *Monitoramento → Hosts*, você pode clicar em *Últimos dados* para visualizar todas as métricas coletadas mais recentes em uma lista. Observe que o item *MySQL: Calculated value of innodb_log_file_size* deve estar sem dados, pois o valor será calculado a partir dos dados da última hora.



Configure alertas de problemas

O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando 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.

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} configurado 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 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 Zabbix agent

Para monitorar um servidor MySQL com o Zabbix agent, 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.

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, você precisa configurar o Zabbix agent para que ele possa 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 Zabbix agent as credenciais necessárias para conectar-se ao servidor MySQL como o usuário "zbx_monitor" criado anteriormente.

Configurar 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 os 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

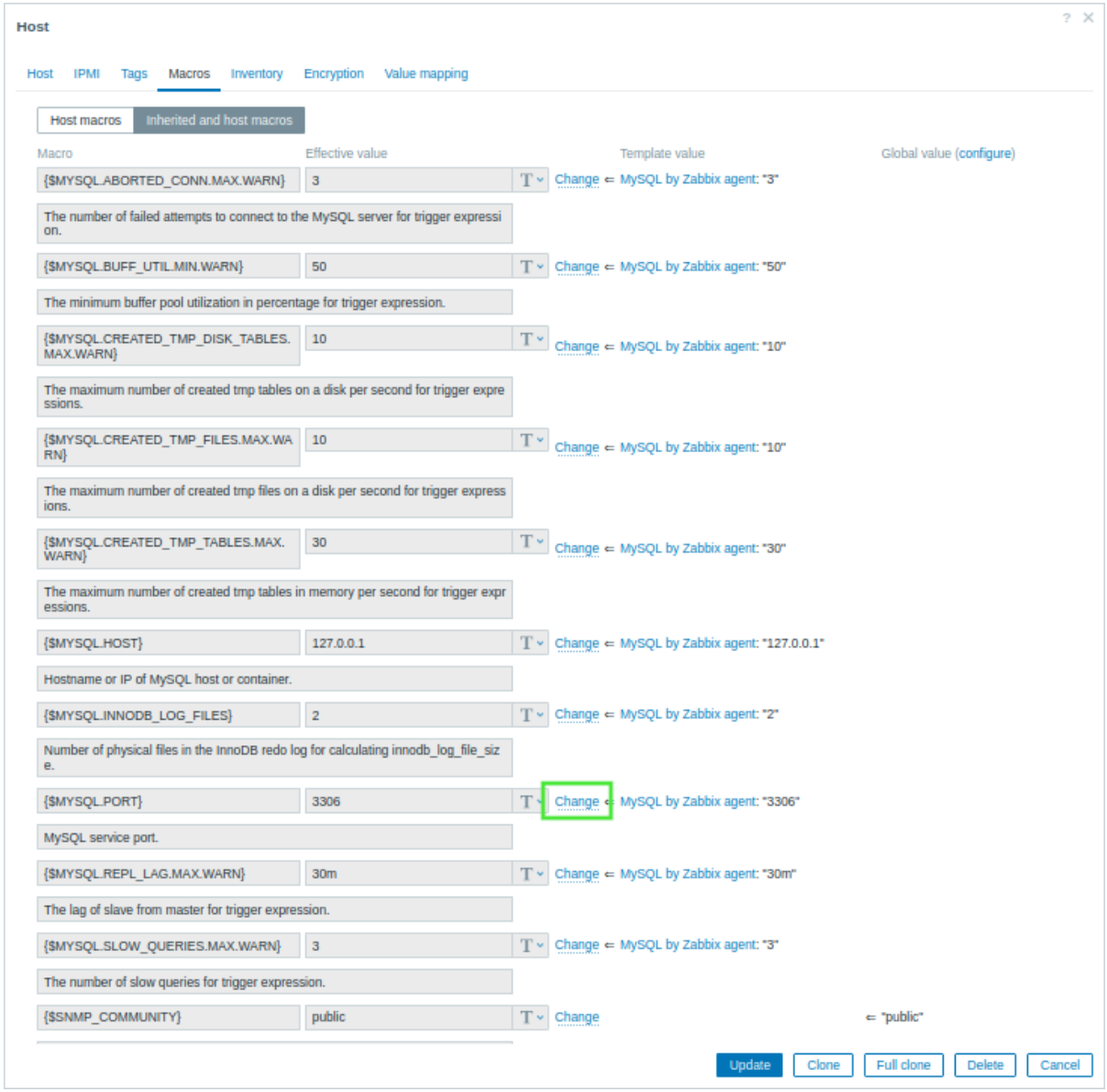
Para configurar o frontend do Zabbix, siga as instruções na seção *Configure o frontend do Zabbix* com os seguintes ajustes:

- 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 o 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 recomendada 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 do 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 a seguir do arquivo *odbcinst.ini*, 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 na 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 do 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.
- 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).
- 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 detalhada, 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.

Este guia é baseado na seguinte configuração:

- Versão do Zabbix: 7.2 PRE-RELEASE (instalado a partir de pacotes)
- Distribuição do SO: Ubuntu
- Versão do SO: 22.04 (Jammy)
- Componentes do Zabbix: Server, Frontend, Agent
- Banco de dados: MySQL
- Servidor web: Apache

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 máquina virtual.

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
```

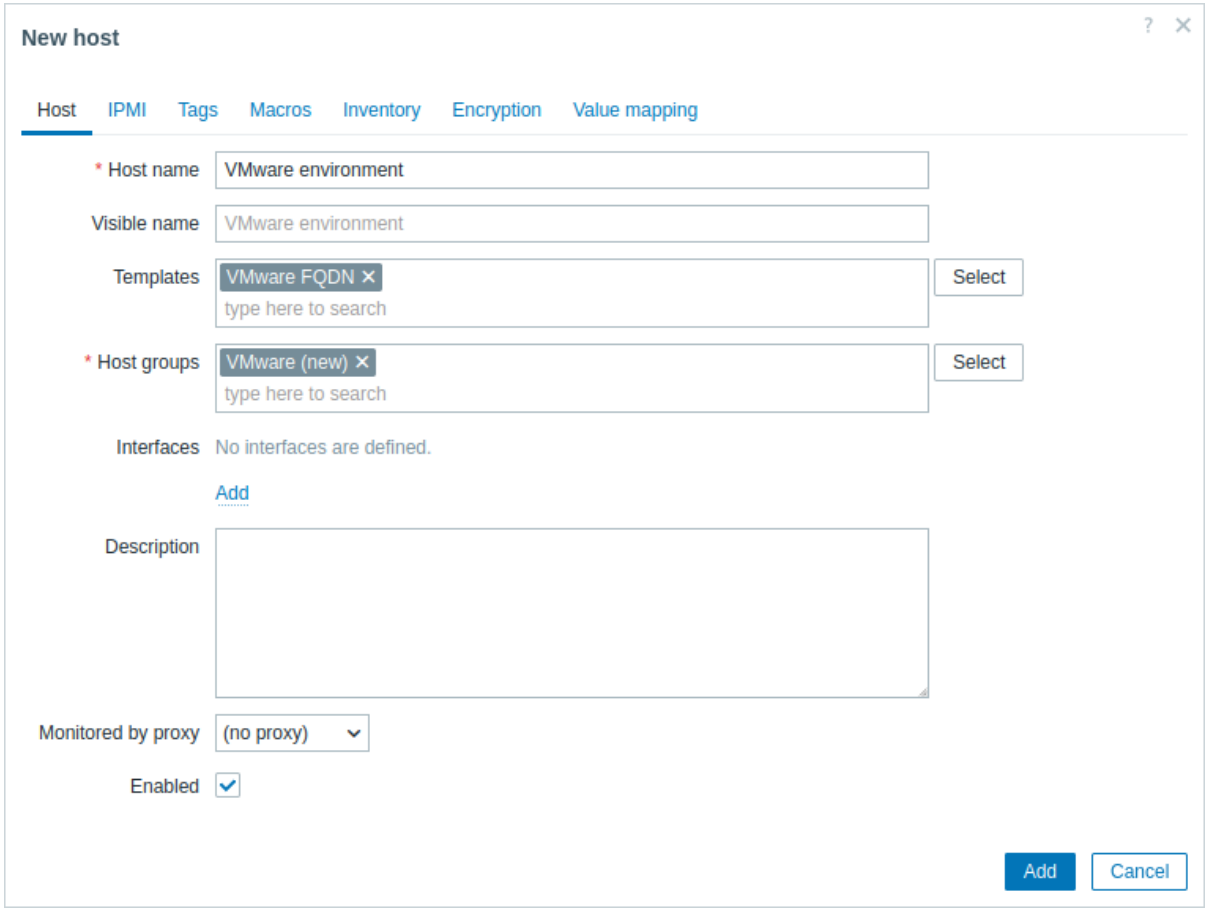
Assim que os processos *vmware collector* forem iniciados, prossiga para o próximo passo.

Configurar o frontend do Zabbix

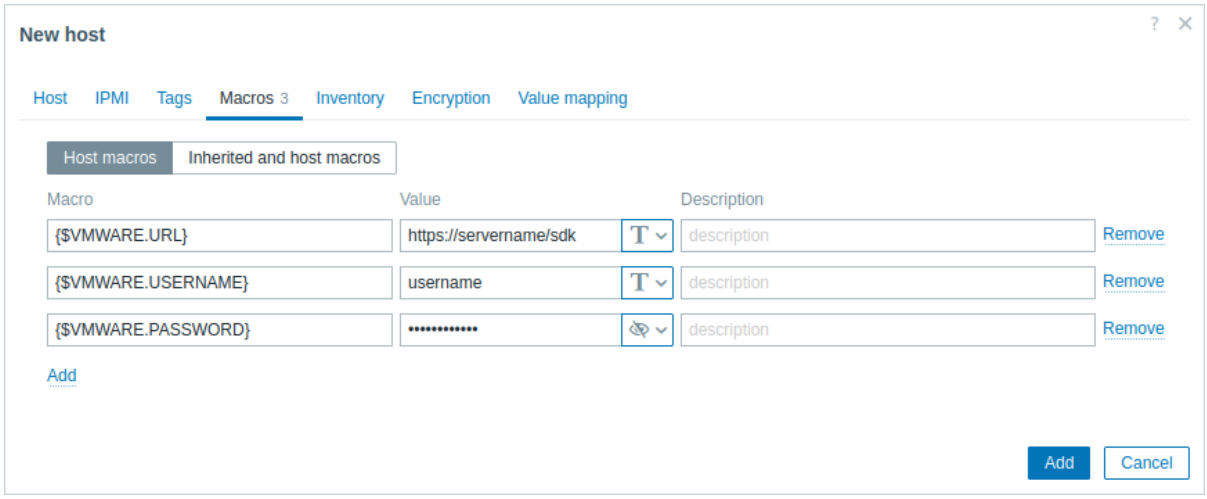
1. Faça login no frontend do Zabbix.

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 obter mais informações sobre esses templates, consulte Monitoramento de máquinas virtuais.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, um novo grupo de hosts ”VMware”).



- Na guia *Macros*, defina as seguintes macros de host:
  - {$VMWARE.URL} - URL do SDK do serviço VMware (vCenter ou hipervisor ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - Nome de usuário do serviço VMware
  - {$VMWARE.PASSWORD} - Senha do usuário {$VMWARE.USERNAME} do serviço VMware



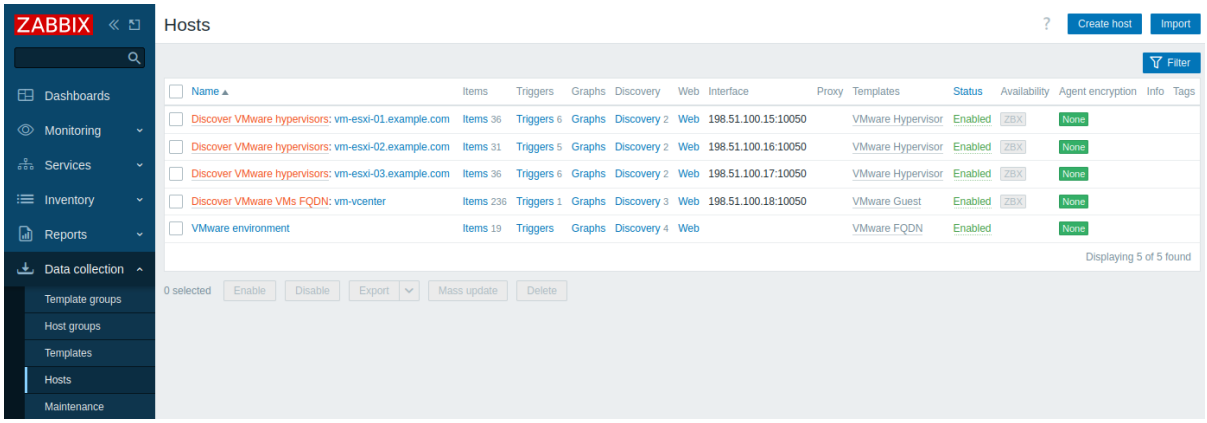
3. Clique no botão *Adicionar* para criar o host. Este host representará seu ambiente VMware.

Visualizar métricas coletadas

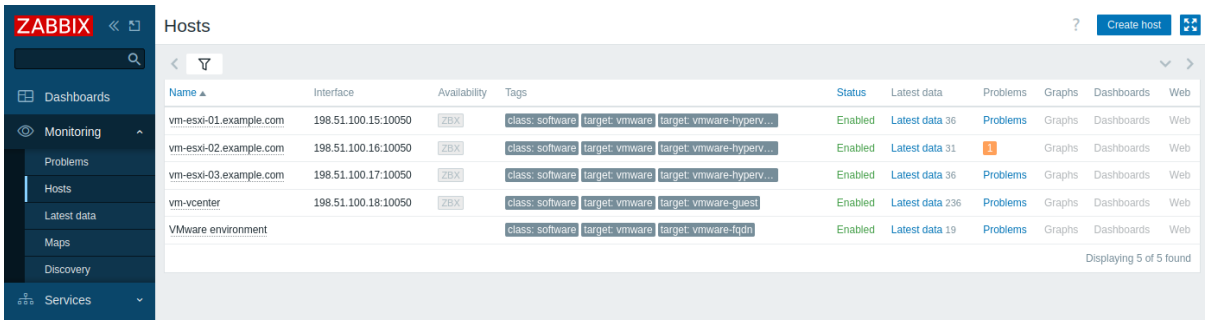
Parabéns! Neste ponto, o Zabbix já está monitorando seu ambiente VMware.

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.

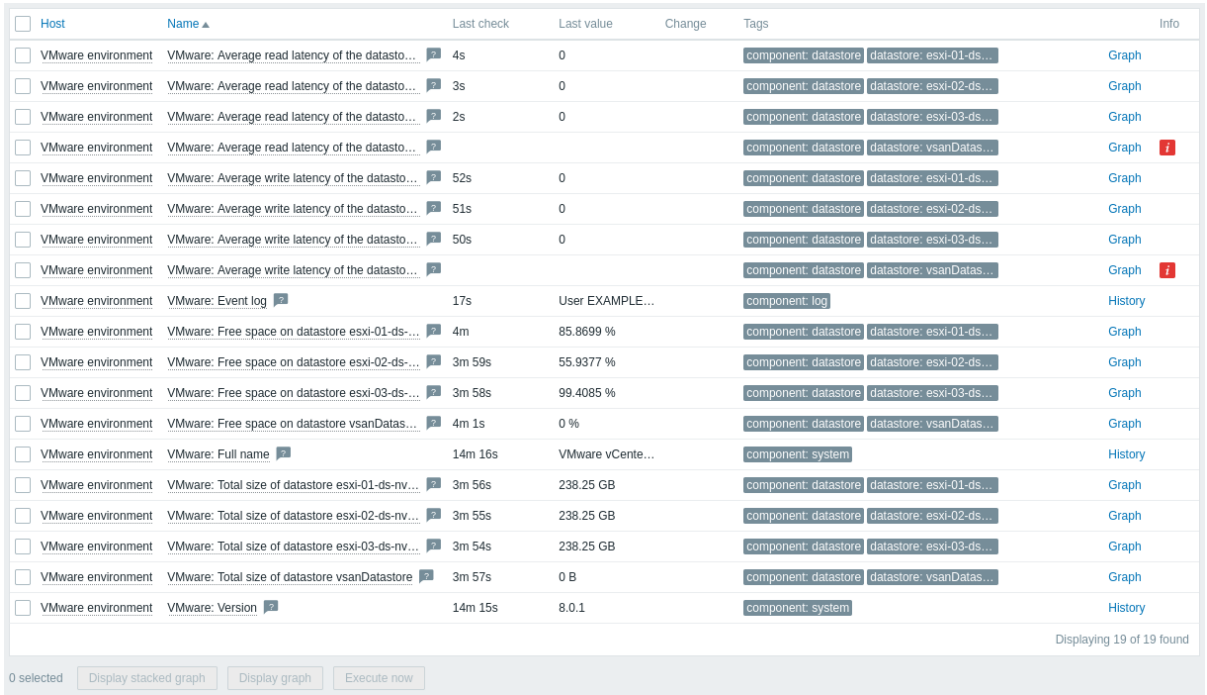
Para visualizar os hosts criados, navegue até a seção de menu *Coleta de dados → Hosts*.



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.

Configurar 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.

Na 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 para o administrador do sistema, depois, se um problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).
- Monitoramento de máquinas virtuais - informações adicionais sobre o monitoramento VMware (processo de coleta de dados, opções de configuração do server, orientações para solução de problemas, etc.).
- Chaves de item de monitoramento VMware - uma lista completa de métricas 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 de 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 do 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
```

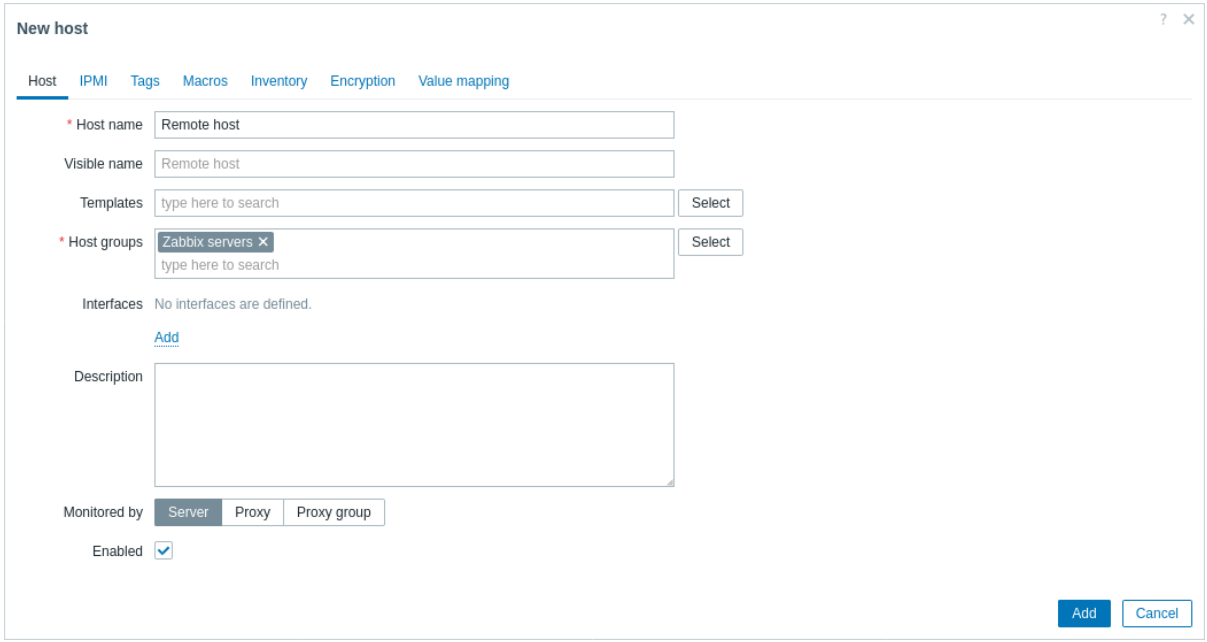
3. Reinicie o agent do Zabbix:

```
systemctl restart zabbix-agent
```

Frontend do Zabbix

1. Faça login no frontend do Zabbix.

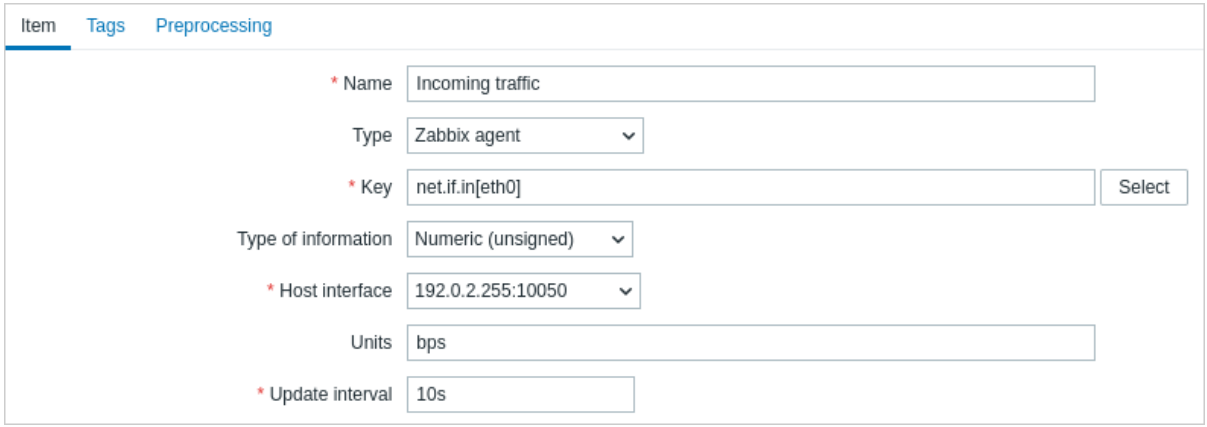
2. Crie um host na interface web do Zabbix, especificando o endereço IP ou o nome DNS da máquina na qual o agent está instalado.



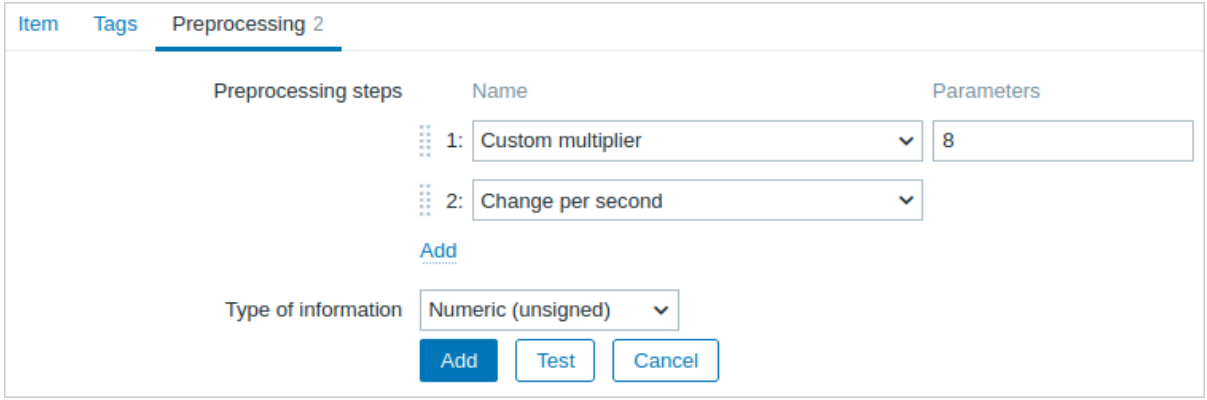
Criar items Siga as instruções sobre criação de um item para adicionar os items para monitoramento de tráfego, a saber:

- Tráfego de entrada
- Tráfego de saída
- Tráfego total

Uma configuração simples para o item de monitoramento de tráfego de entrada seria a seguinte:

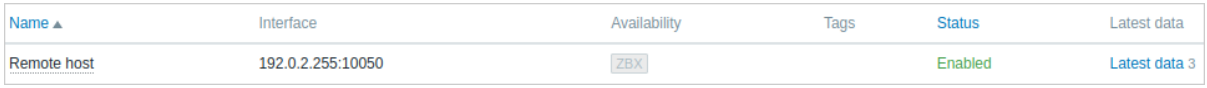


Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os items. No caso apresentado, essas etapas podem ser multiplicação por 8 (para converter bytes em bits) e apresentação como alteração por segundo.

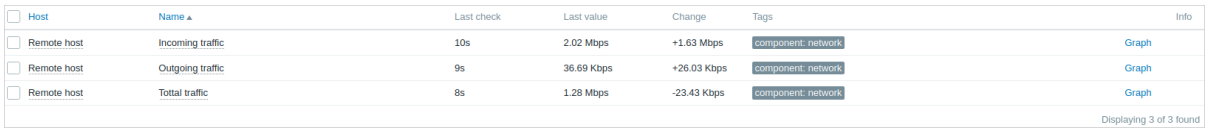


Visualizar dados coletados Parabéns! Neste ponto, o Zabbix já está monitorando o tráfego de rede.

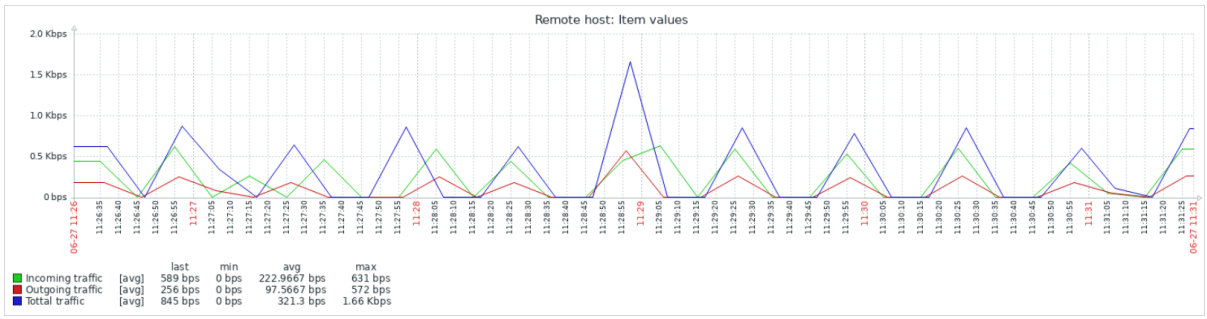
Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento → Hosts* e clique em *Últimos dados* na linha do host.



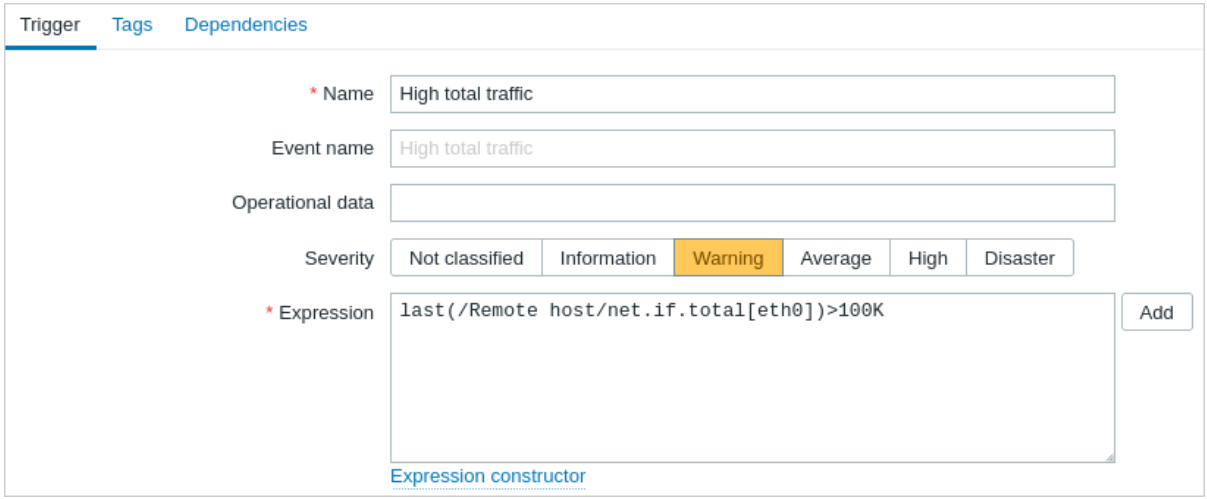
Você verá os dados de tráfego.



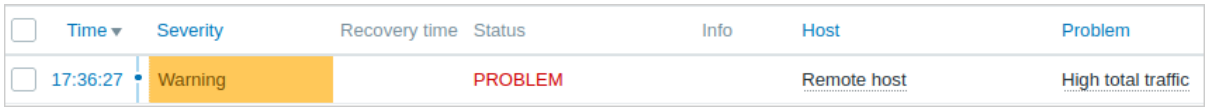
Visualizar gráficos Os dados coletados podem ser exibidos como gráficos. Para visualizá-los, na seção *Últimos dados*, clique em *Gráfico* na linha do item ou selecione os itens desejados e clique em *Exibir gráfico* abaixo.



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 o 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 para a entrega da notificação.

Veja também:

- Escalonamentos de problemas - como criar cenários de alerta em múltiplos passos (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).
- 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.

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 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 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 servidor 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 servidor 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 do 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 servidor. Especificamente, o agent inicia uma conexão com o servidor e se identifica usando o valor de Hostname. O servidor 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
```

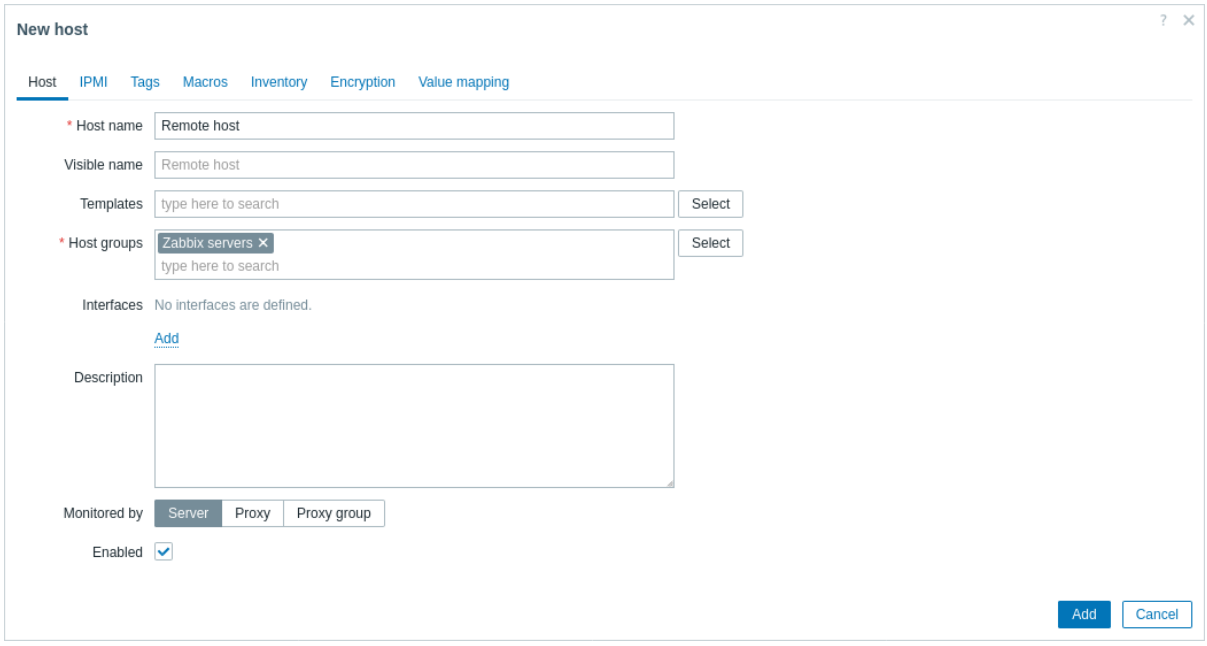
Frontend do Zabbix

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, ”Remote host”) 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.

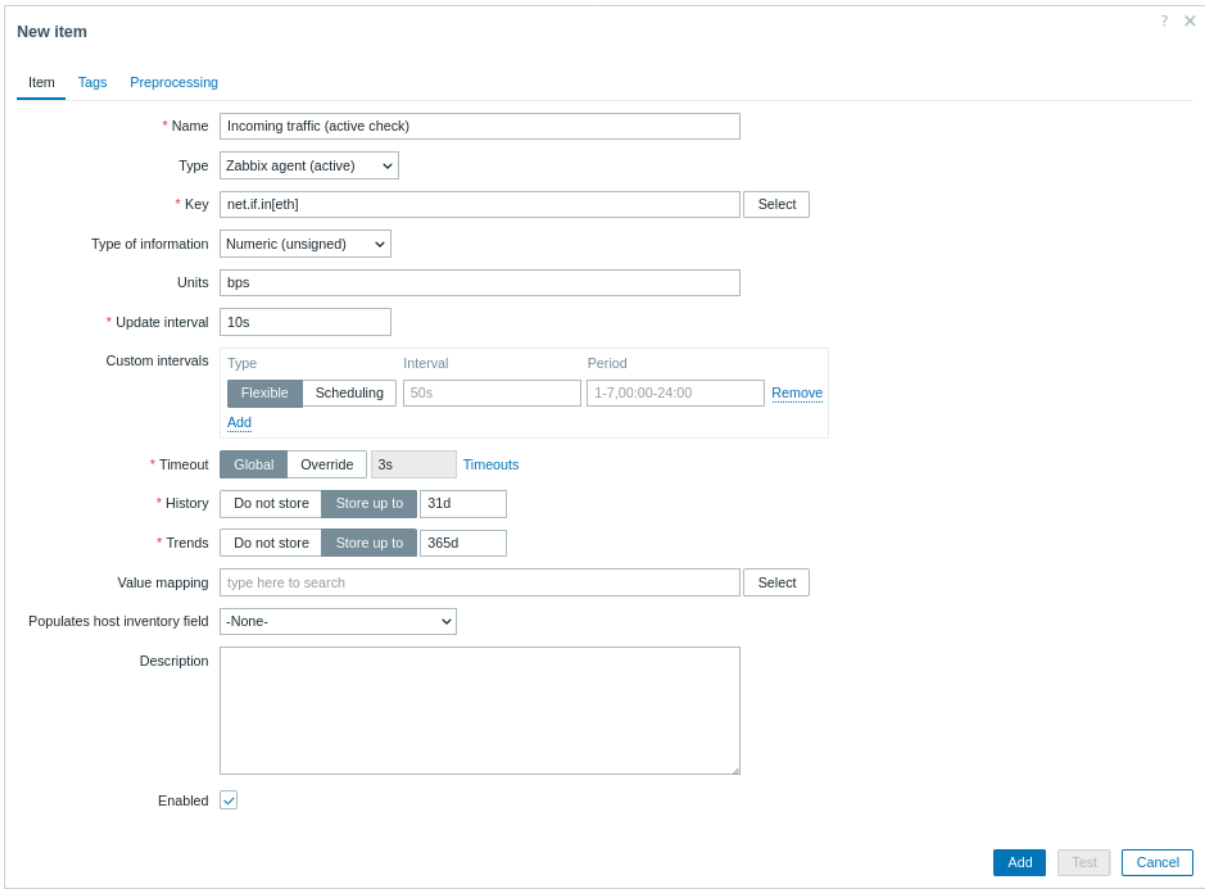


3. Clique em *Adicionar* para adicionar o host. Este host representará a máquina Linux monitorada.

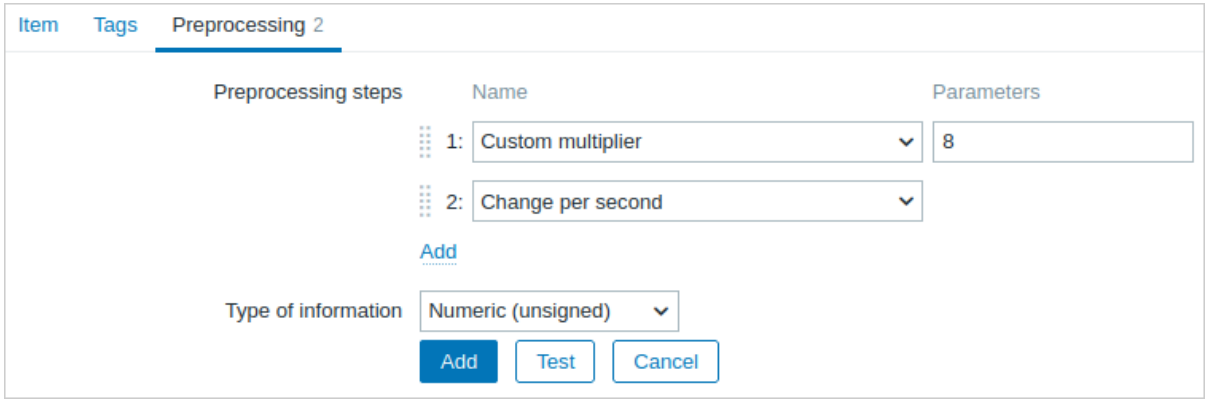
Criar items Siga as instruções em criando um item para adicionar os items para monitoramento de tráfego, a saber:

- 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:

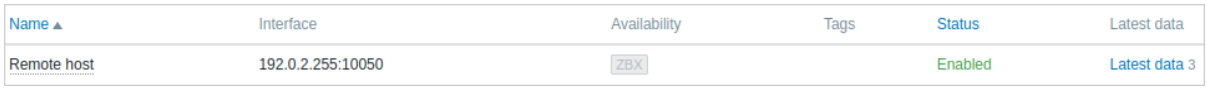


Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os items. No caso dado, essas etapas podem ser multiplicação por 8 (para converter bytes em bits) e apresentação como alteração por segundo.

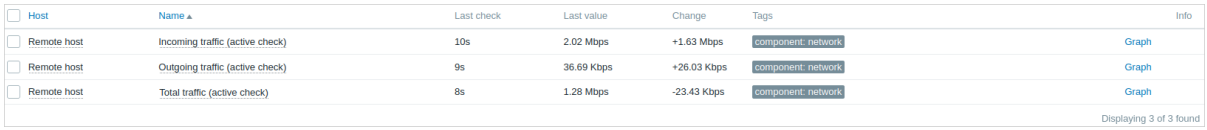


Visualizar dados coletados Parabéns! Neste ponto, o Zabbix já está monitorando o tráfego de rede.

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 como 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 Zabbix agent - 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 do Browser

Introdução

Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de sites com items 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.

Este guia é baseado na seguinte configuração:

- Versão do Zabbix: 7.2 PRE-RELEASE (instalado a partir de pacotes)
- Distribuição do SO: Ubuntu
- Versão do SO: 22.04 (Jammy)
- Componentes do Zabbix: Server, Frontend, Agent
- Banco de dados: MySQL

- Servidor web: Apache

Configurar o WebDriver

Os items 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 aborda 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 Zabbix server

Os itens do navegador são executados e processados pelos processos *browser poller* do Zabbix, que precisam ser habilitados ajustando o parâmetro de configuração do server 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 Zabbix server.

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

2. Localize e defina o parâmetro WebDriverURL no arquivo de configuração do Zabbix server:

```
##### 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 Zabbix server.

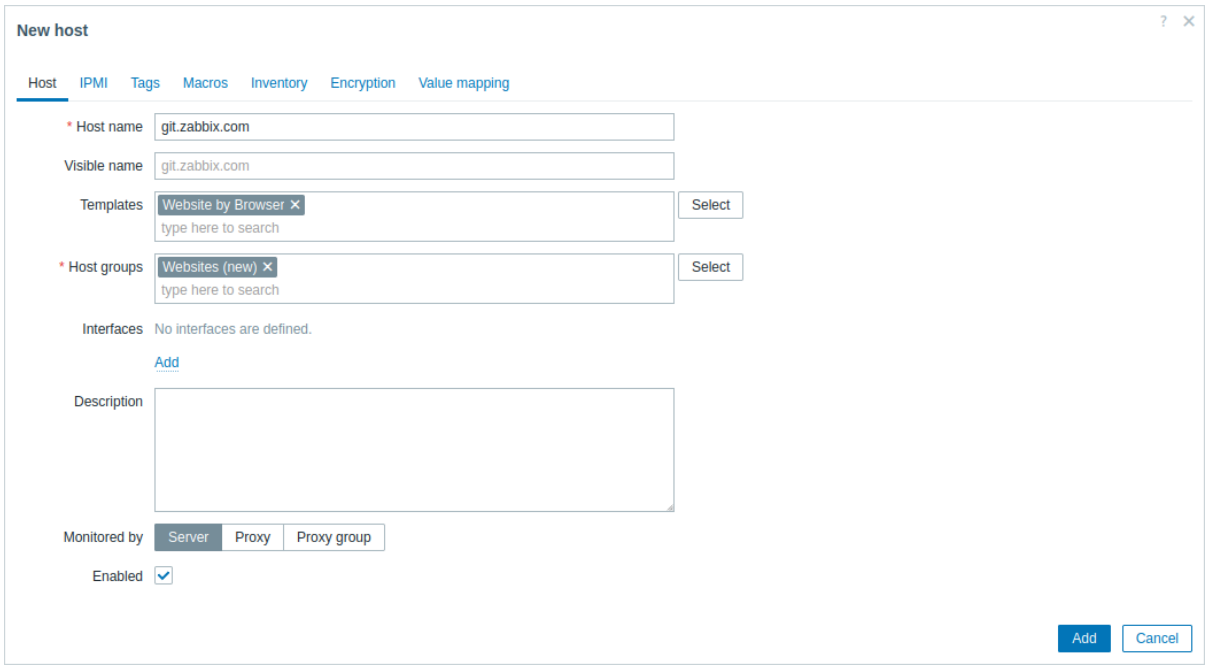
```
systemctl restart zabbix-server
```

Configurar o frontend do Zabbix

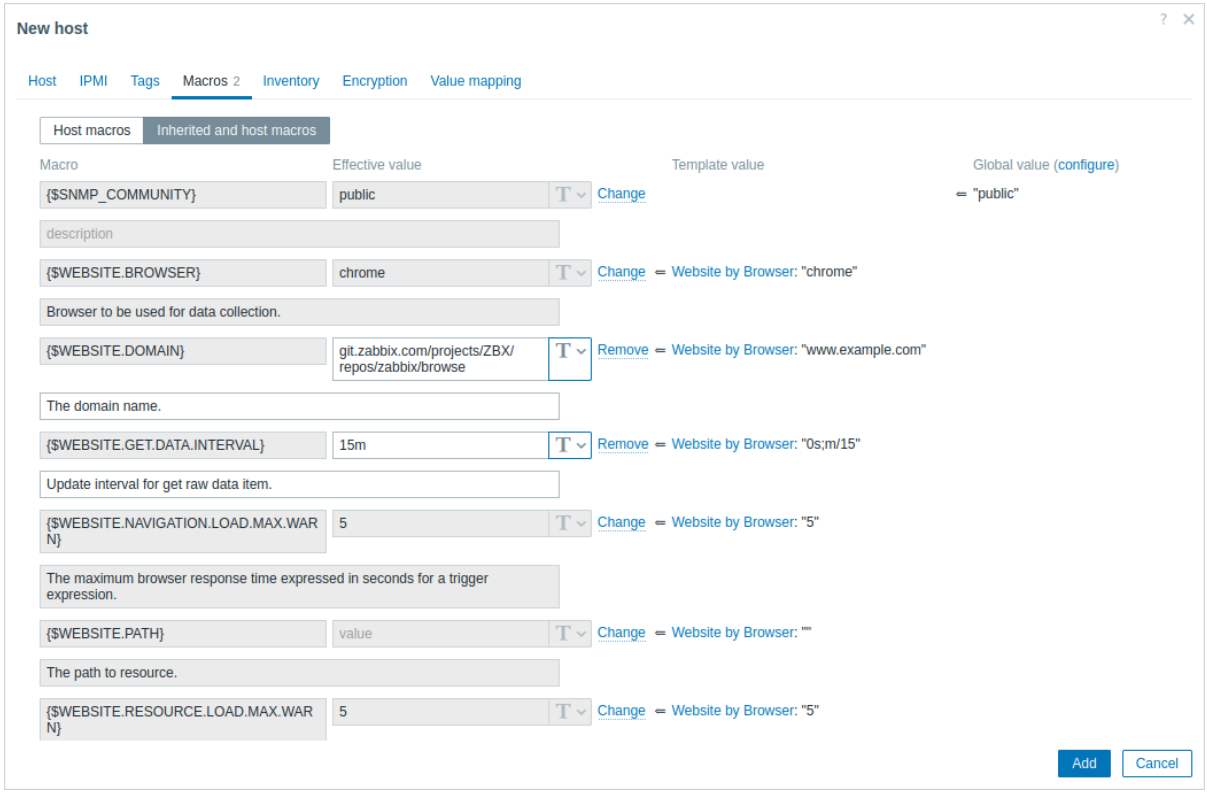
1. Faça login no frontend do Zabbix.

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 aba *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)

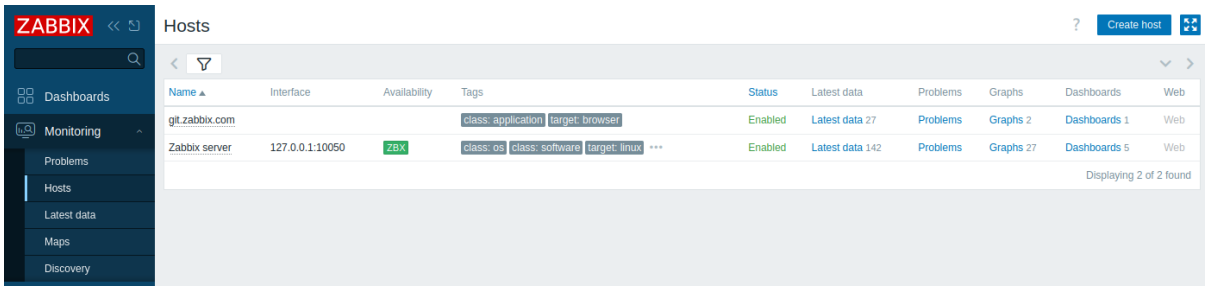


3. Clique no botão *Adicionar* para criar o host. Este host representará o site que você deseja monitorar.

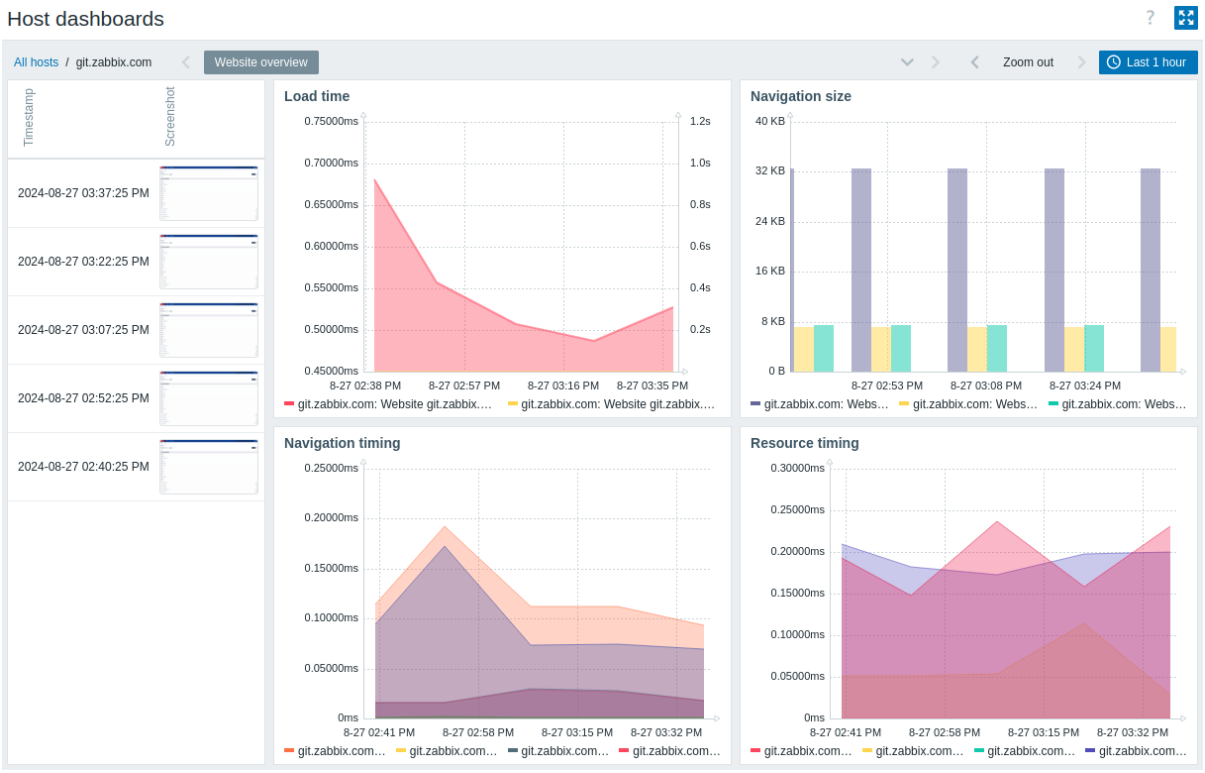
Visualizar métricas coletadas

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 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 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} configurado 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 ao 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 ao administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem ao 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 requer 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 pressupõe que tanto o Zabbix server quanto o agent estão instalados na mesma máquina; portanto, `127.0.0.1` é usado na configuração.

Configurar 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, não é necessária instalação ou configuração separada.

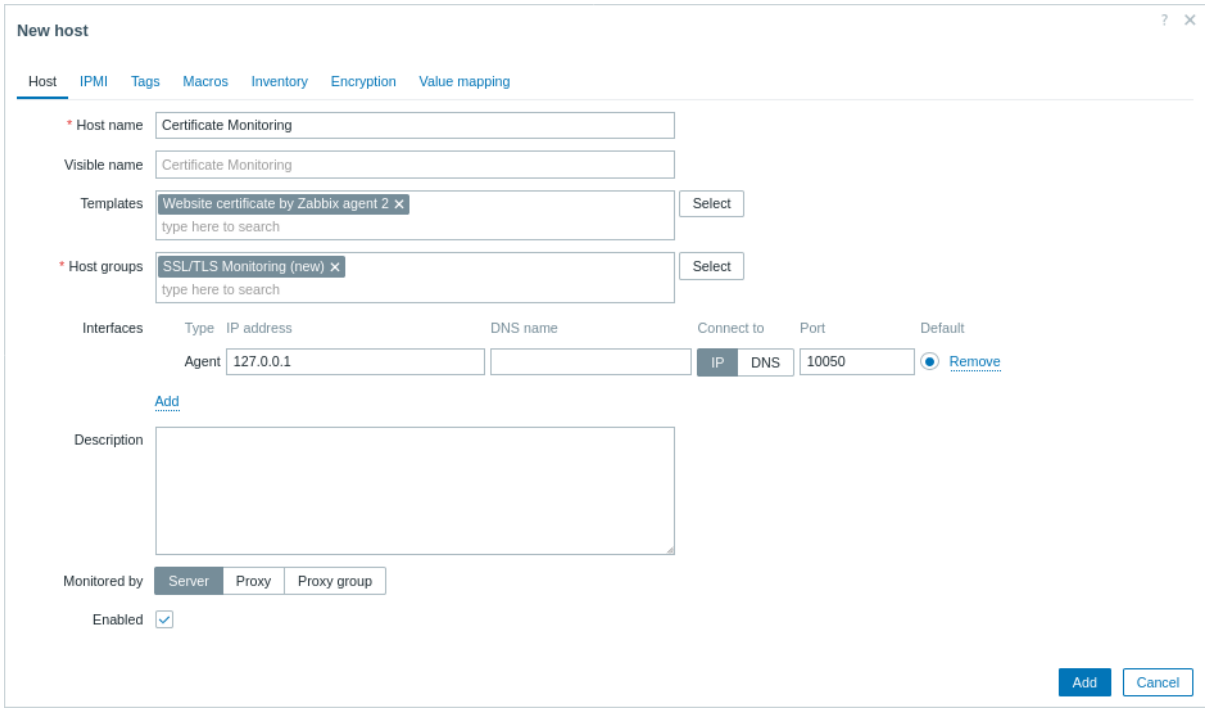
Configurar o frontend do Zabbix

1. Faça login no frontend do Zabbix.

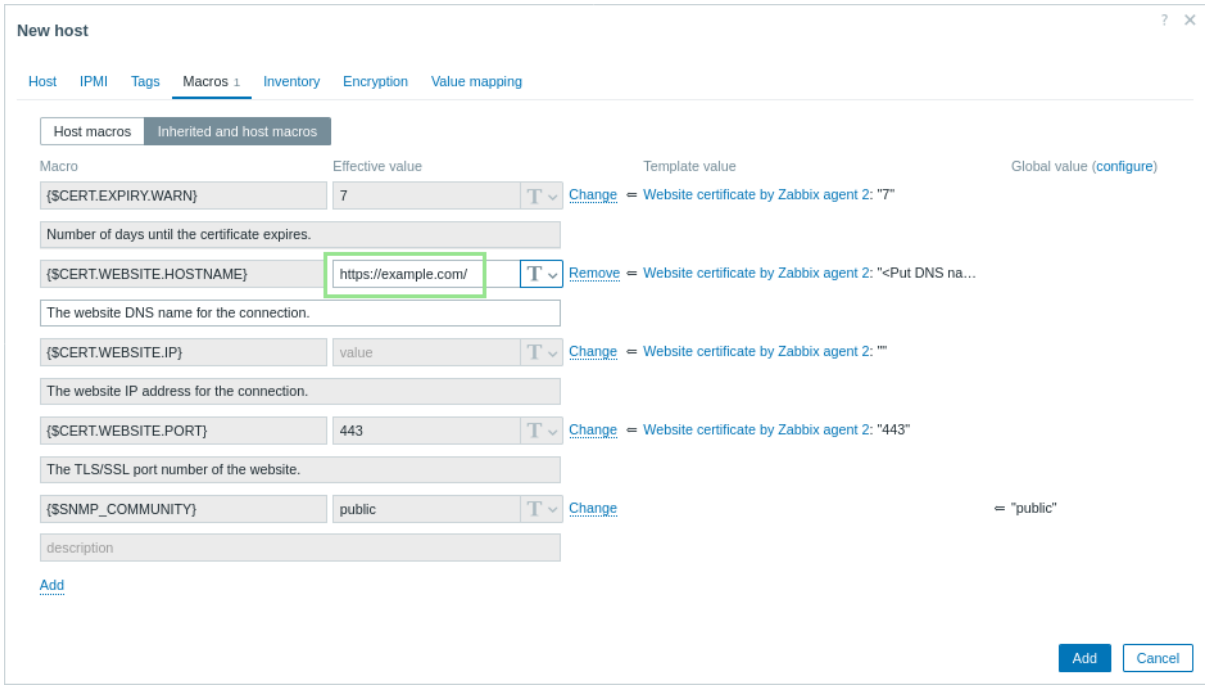
2. Navegue até *Monitoramento* > *Hosts*.

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. Para este 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, você pode especificar uma lista separada por vírgulas de seus nomes de host na macro `{$CERT.WEBSITE.HOSTNAME}`. Opcionalmente, você também pode fornecer uma lista separada por vírgulas de portas na macro `{$CERT.WEBSITE.PORT}`, onde cada porta corresponde ao respectivo nome de host na ordem. Por exemplo, para monitorar example.com e example.org na porta 8443:

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

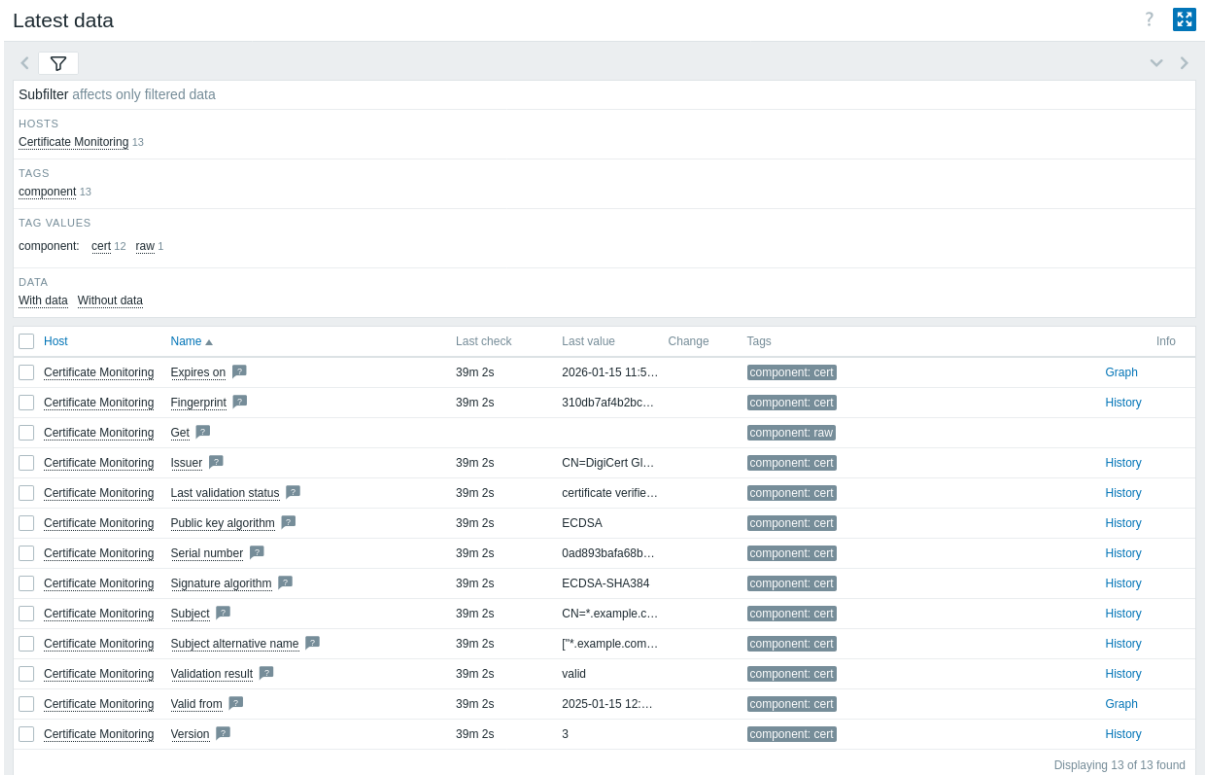
Para cada site especificado na macro `{$CERT.WEBSITE.HOSTNAME}`, o Zabbix criará um conjunto correspondente de items e triggers. Isso permite o monitoramento e alerta individual para o certificado SSL de cada site.

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

Visualizar métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando o certificado web desejado.

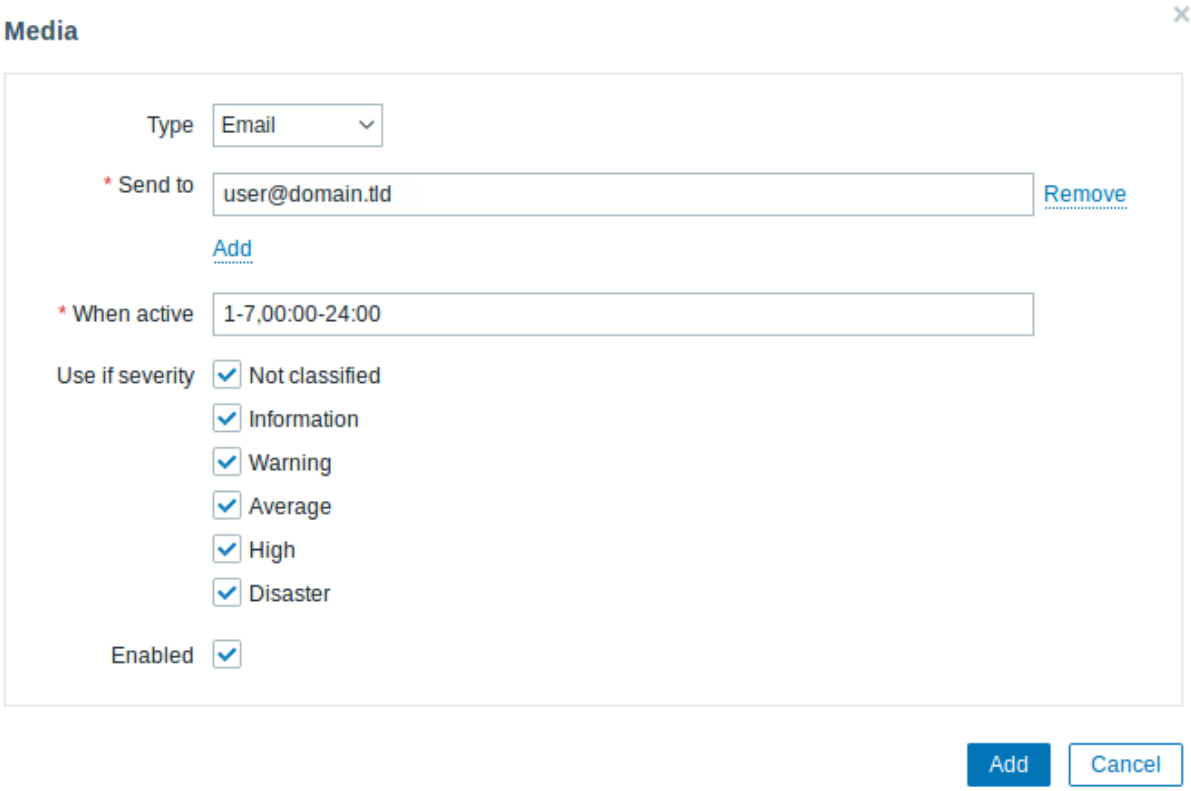
Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento > Hosts* e clique em *Últimos dados* ao lado do host para ver todas as métricas coletadas mais recentes em uma lista, como data de expiração, emissor e sujeito.



Configure alertas de problemas

O Zabbix pode notificá-lo sobre problemas na infraestrutura. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Acesse *Configurações do usuário > Perfil*, vá para a aba *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 (mais de 365 dias deve ser suficiente) para receber um aviso antes do certificado expirar.

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 para o 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 habilitar 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 possui:

- 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).
- Dispositivo habilitado para SNMP: um switch ou roteador de rede (por exemplo, um roteador Cisco) com SNMP habilitado.
- 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, dificultando a configuração de items e a solução de 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.

Este guia é baseado na seguinte configuração:

- Versão do Zabbix: 7.2 (instalado a partir de pacotes)
- Distribuição do SO: Ubuntu
- Versão do SO: 24.04.2+
- Componentes do Zabbix: server, frontend e, opcionalmente, agent (se monitorar métricas de rede locais)
- Banco de dados: MySQL
- Servidor web: Apache
- Dispositivo de rede: Cisco Catalyst 3750V2-24FS

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 de 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. Observação: 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 do 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.

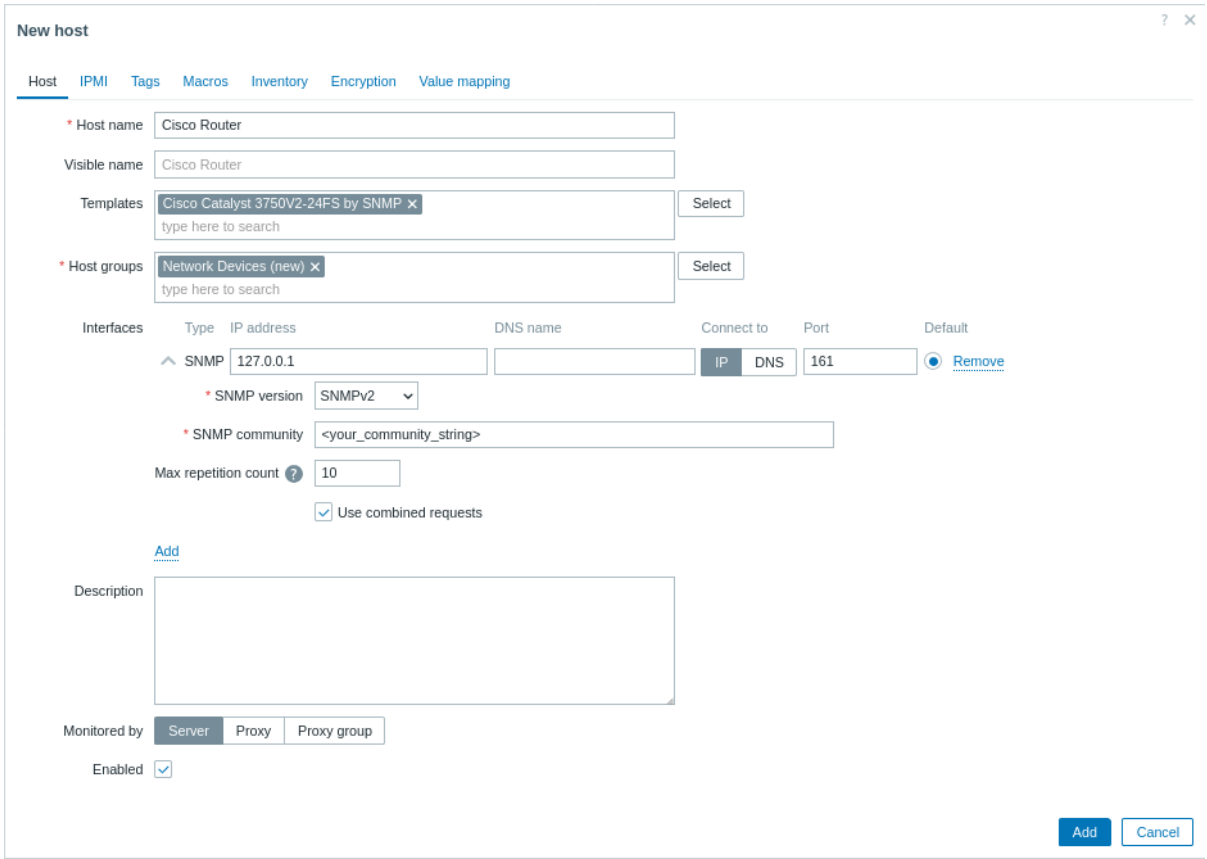
Navegue até *Coleta de dados* > *Hosts* e clique em *Criar host*.

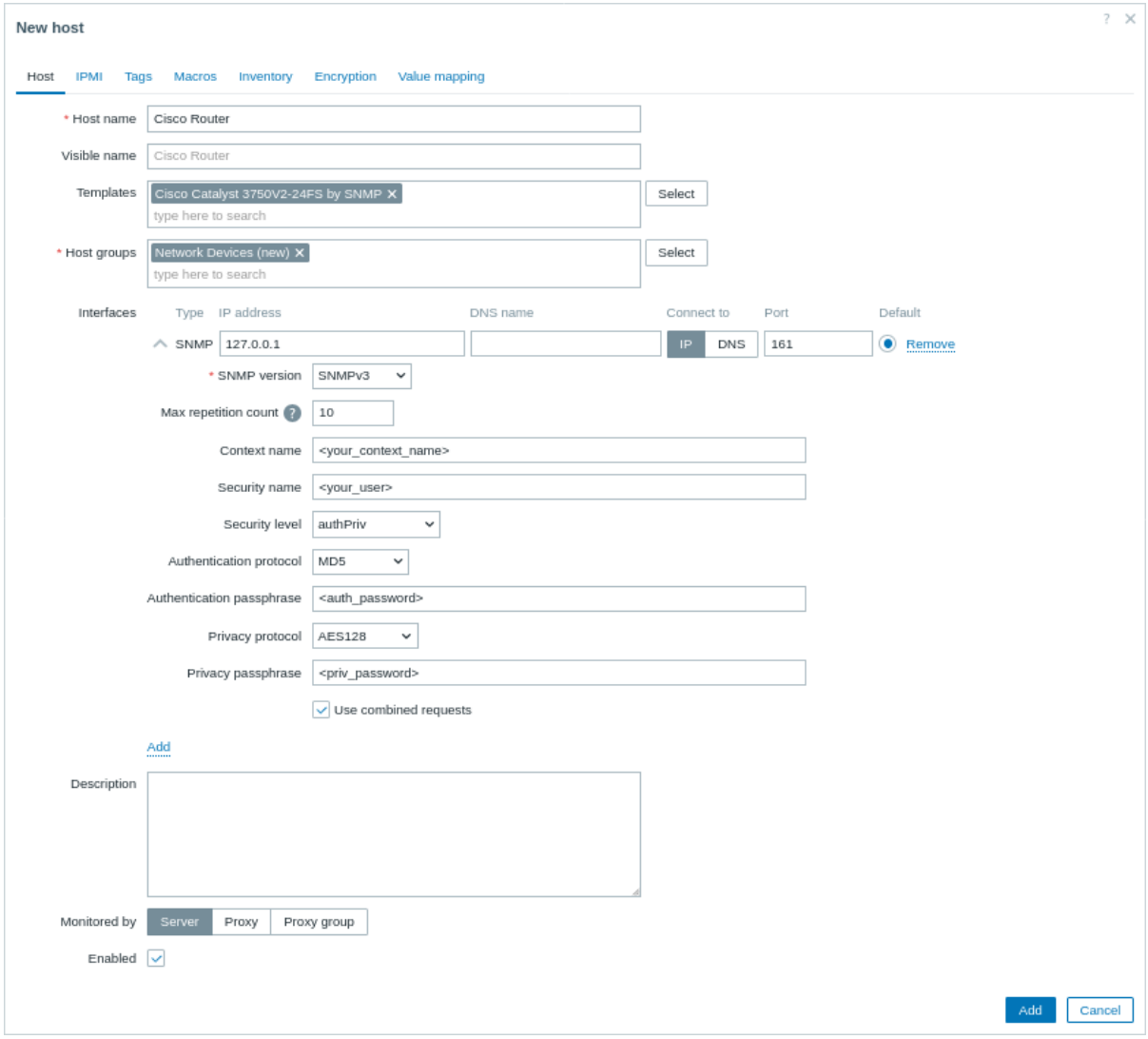
- *Nome do host*: insira um nome para o seu dispositivo (por exemplo, ”Cisco Router”).
- Grupos de hosts: selecione um grupo existente ou crie um novo grupo, como ”Dispositivos de rede”.
- Interfaces:
  - Clique em *Adicionar* em Interfaces.
  - Escolha *SNMP* como o tipo de interface.
  - Insira o endereço IP ou o 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 muitas 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.

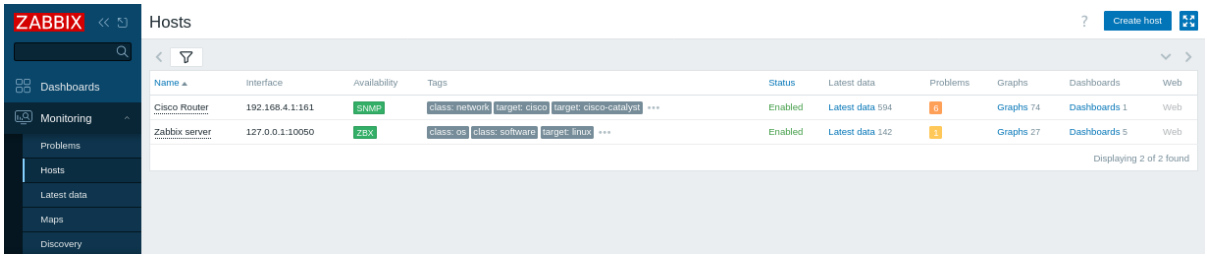




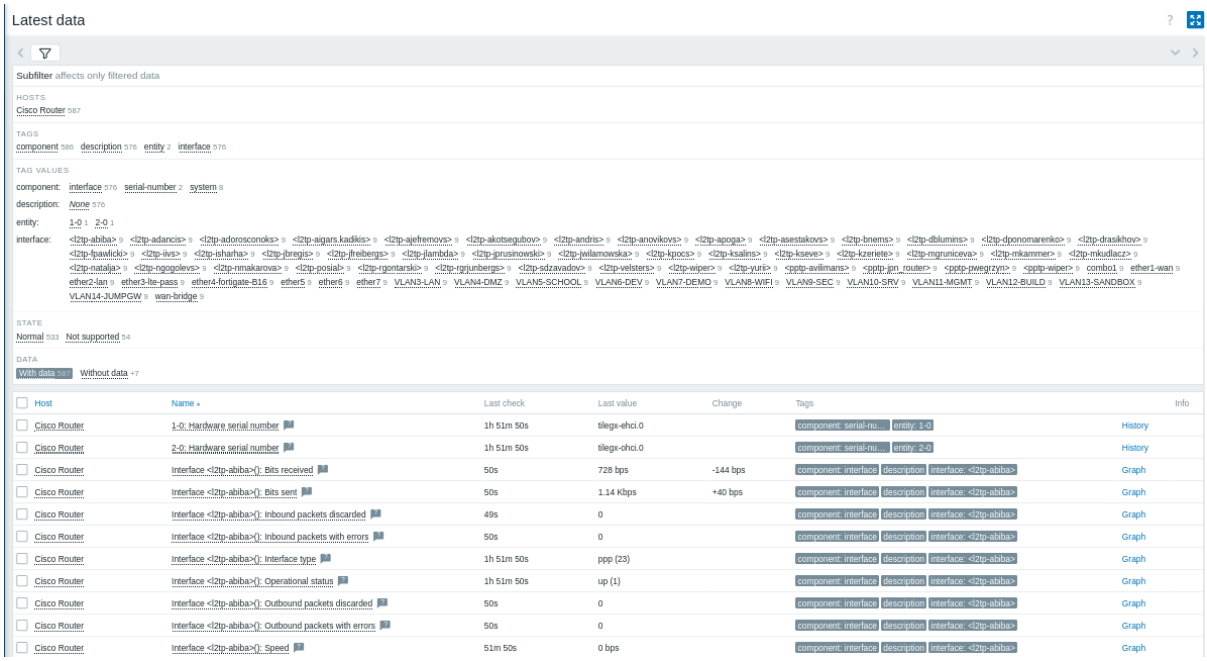
Visualizar métricas coletadas Parabéns! O Zabbix agora está configurado para monitorar seu dispositivo de rede.

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. Observação: 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:

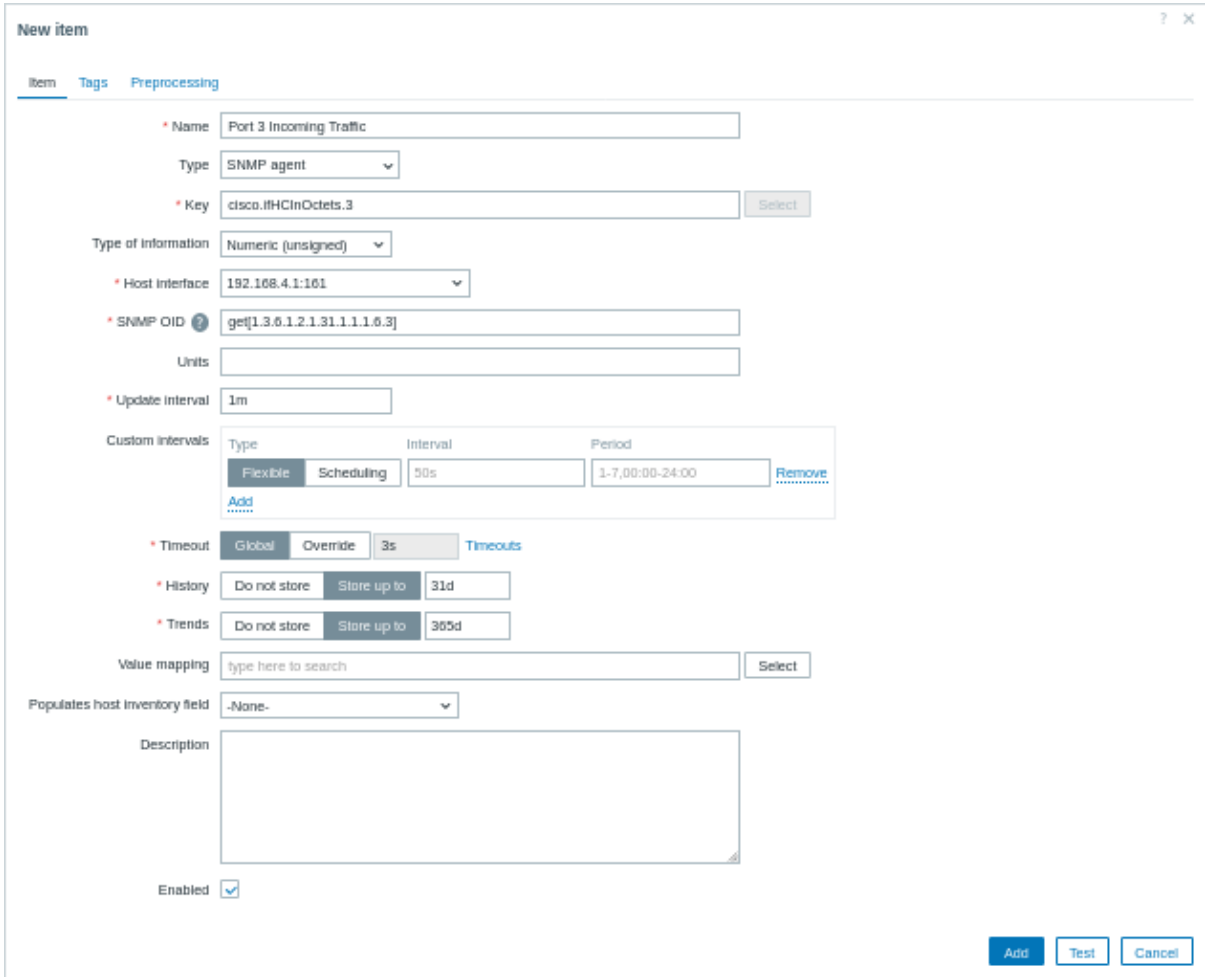
```
snmpwalk -v 2c -c <sua_string_de_comunidade> <IP_do_dispositivo> .
```

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:

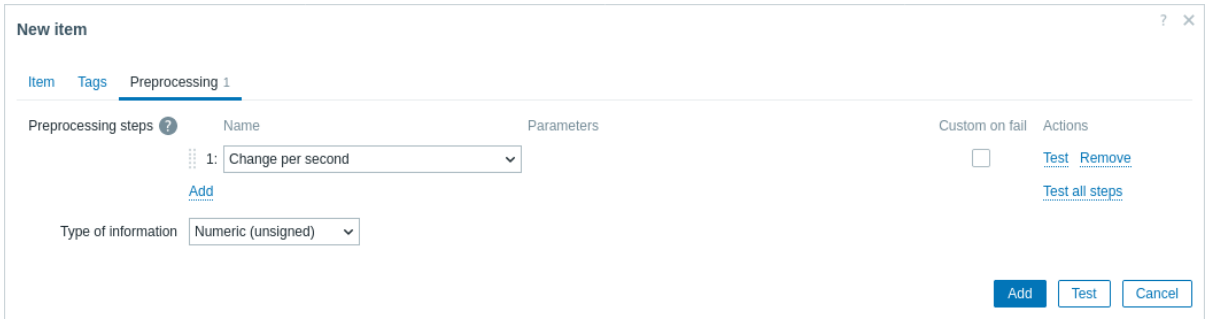
```
snmpget -v 2c -c <sua_string_de_comunidade> -On <IP_do_dispositivo> IF-MIB::ifHCInOctets.3
```

2. Crie um item SNMP:

- Navegue até *Coleta de dados* > *Hosts* e clique na guia *Items* para 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 MIB Ao trabalhar com SNMP, pode ser necessário converter entre OIDs numéricos e seus respectivos nomes MIB. Essa tradução ajuda a identificar e solucionar problemas de métricas com mais facilidade.

- Traduzindo um nome MIB para um OID numérico: use o comando `snmptranslate` com a opção `-On`. Por exemplo, para traduzir o nome 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 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 MIB, execute:

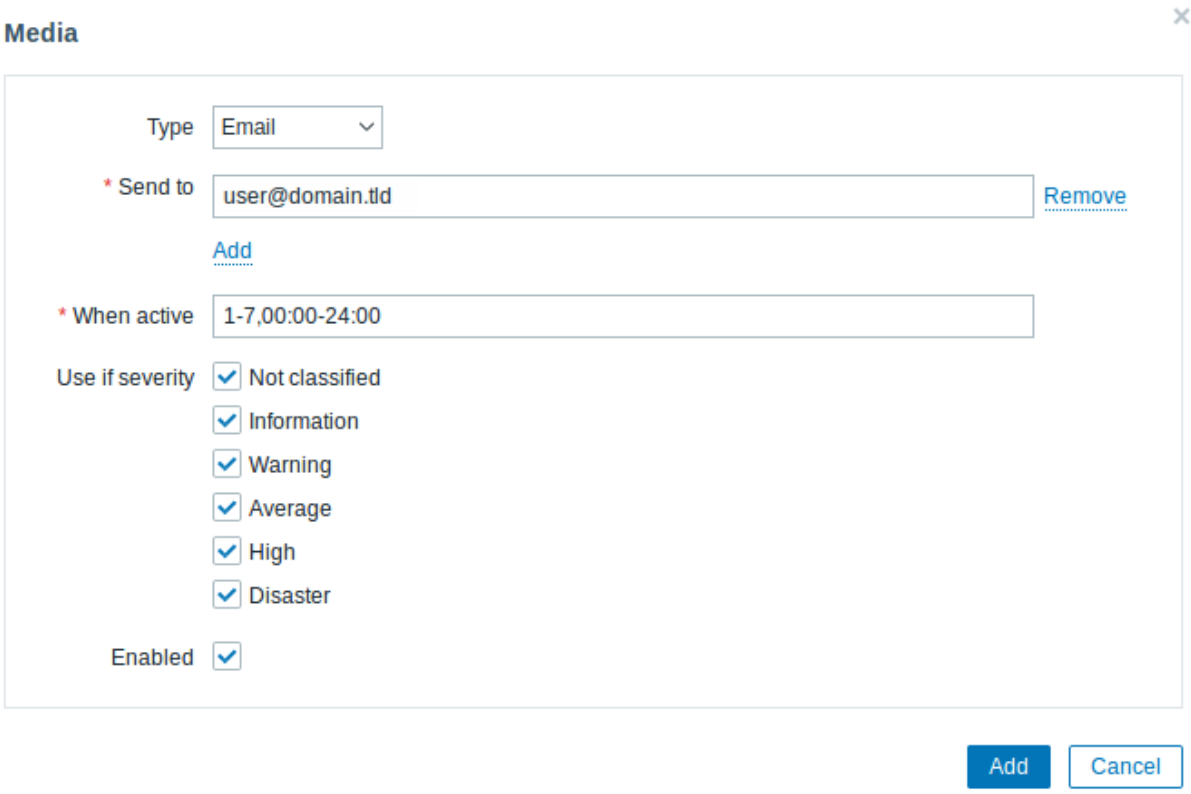
  snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3

Esse comando pode retornar:

`IF-MIB::ifHCInOctets.3`

Configurar alertas de problema 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.

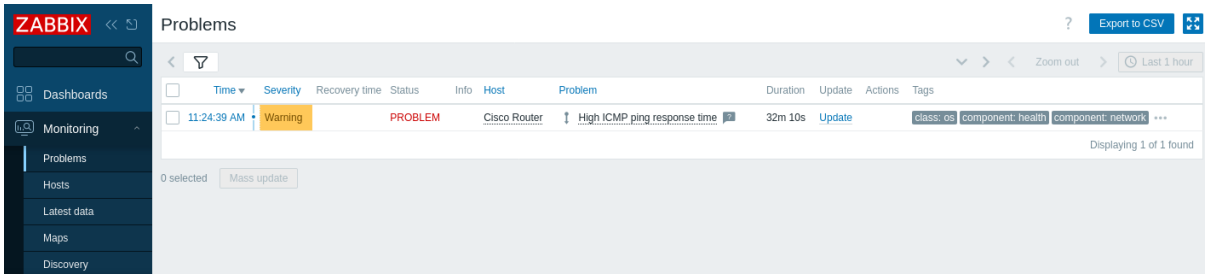


2. Siga o guia para Receber uma notificação de problema.

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 de limite de tempo de resposta semelhante).
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 com segurança.

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.
- Agent SNMP - informações adicionais sobre monitoramento SNMP com o Zabbix.
- Templates padronizados para dispositivos de rede - informações sobre os templates SNMP disponíveis.
- Descoberta de OIDs SNMP - informações adicionais sobre 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 no Zabbix, você pode coletar e analisar eventos críticos (como tentativas de login falhas, erros de 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.

Configurar 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:

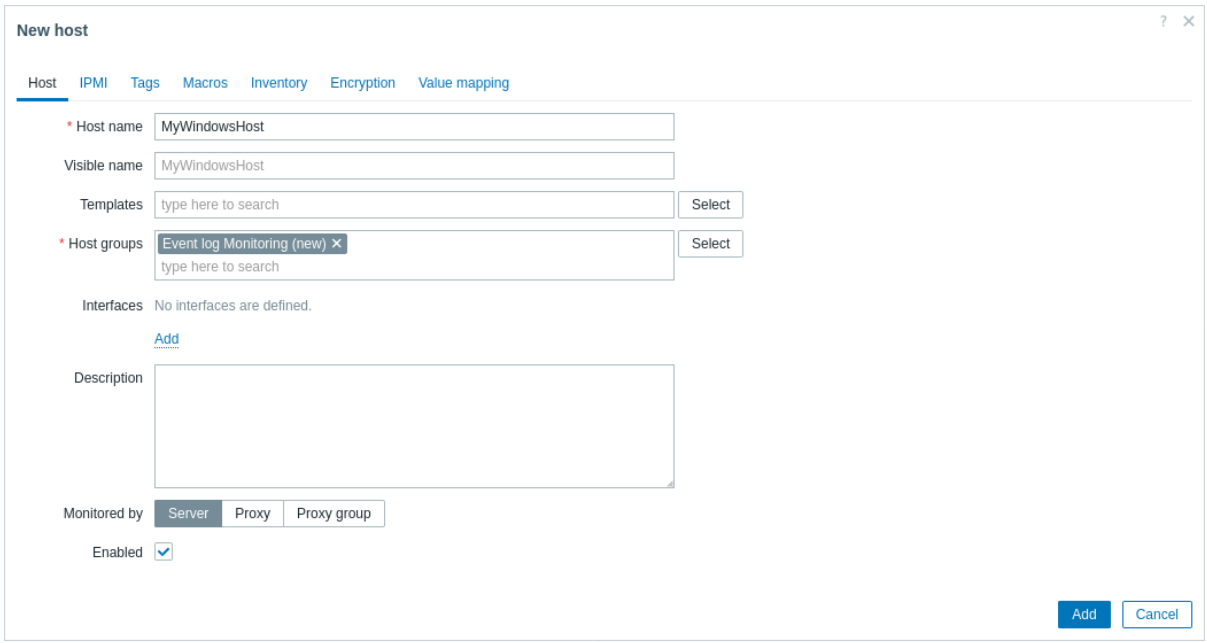
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configure o frontend do Zabbix

1. Navegue até *Coleta de dados* > *Hosts* e crie um host:

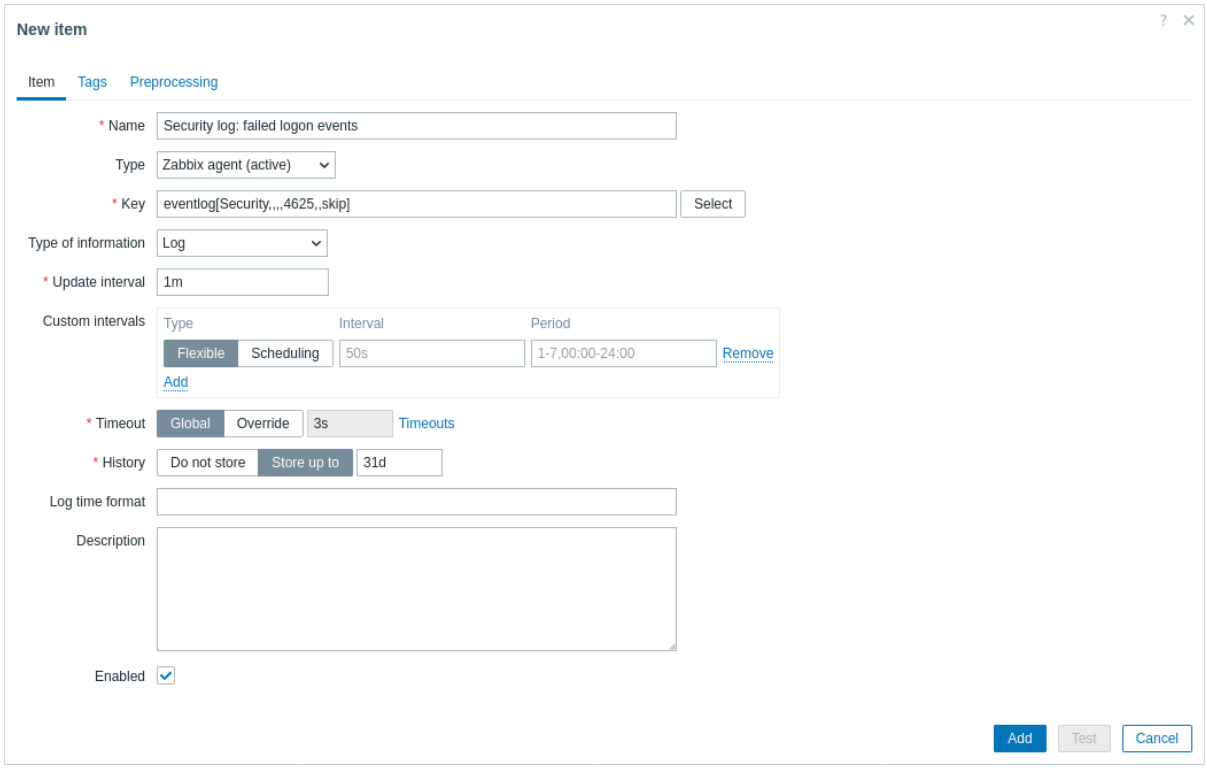
- No campo *Nome do host*, insira um nome para o host (por exemplo, "MeuHostWindows").
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, "Monitoramento de Event log").
- Pressione *Adicionar* para salvar o host configurado.

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 (ativo)” (necessário para monitoramento de Event log).
- No campo *Chave*, use 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”.



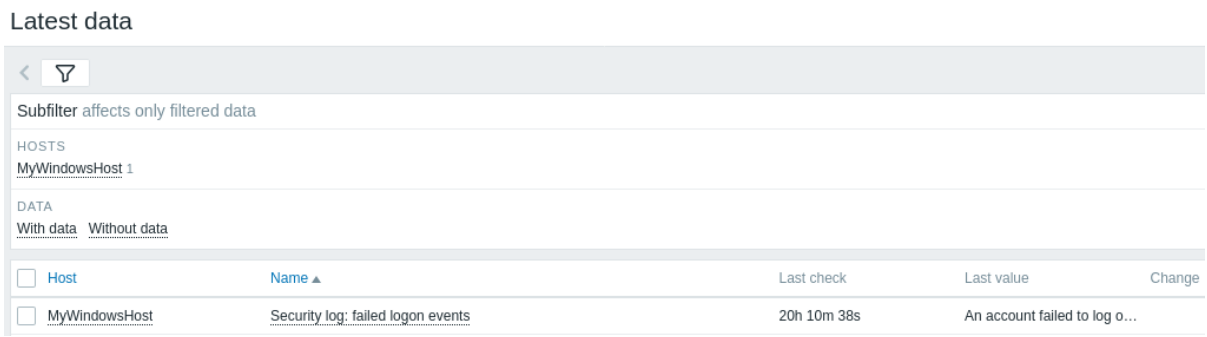
3. Clique em *Adicionar* para salvar o item.

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* > *Dados recentes* 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.

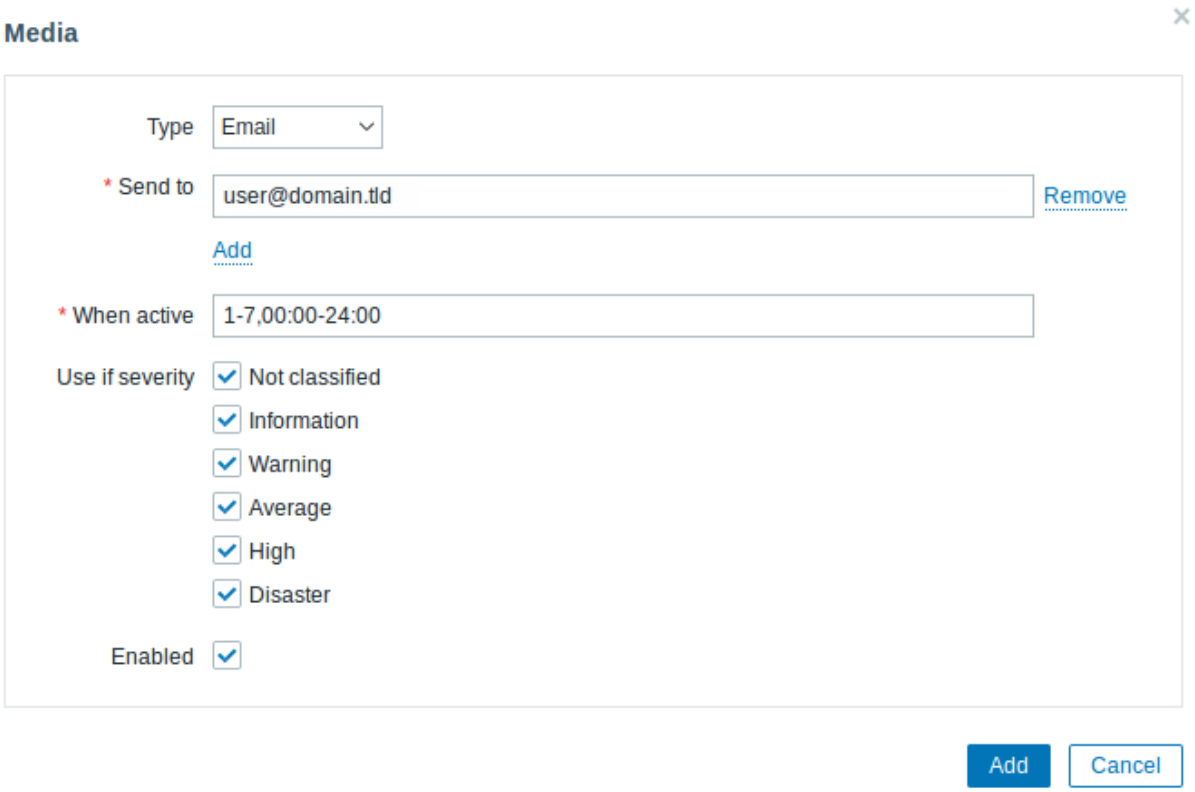
Configurar alertas de problema

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 seu item de log de eventos registrar o padrão desejado. Por exemplo, para capturar tentativas de logon 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, quando 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 o nome exato 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

Esta seção contém tudo o que você precisa saber sobre as opções de configuração e controles de acesso disponíveis para organizações e nós no Zabbix Cloud:

- Implantar o Zabbix com o Zabbix Cloud
- Configuração do nó
- Adicionando usuários a nós e organizações
- Principais diferenças entre o Zabbix Cloud e o Zabbix local

> **Note:**
> O Zabbix Cloud atualmente executa o Zabbix 7.0. Consulte a documentação do Zabbix 7.0 para obter informações relevantes ao ambiente atual do Zabbix Cloud.

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 nenhuma forma. 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 distribuída juntamente com o software no mesmo meio. Qualquer outro uso, como qualquer disseminação 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 que não tenham sido expressamente concedidos acima.

1 Implantar o Zabbix com o Zabbix Cloud

Introdução

Esta página descreve como executar as seguintes ações no Zabbix Cloud:

- Gerenciar organizações
- Criar e gerenciar nós

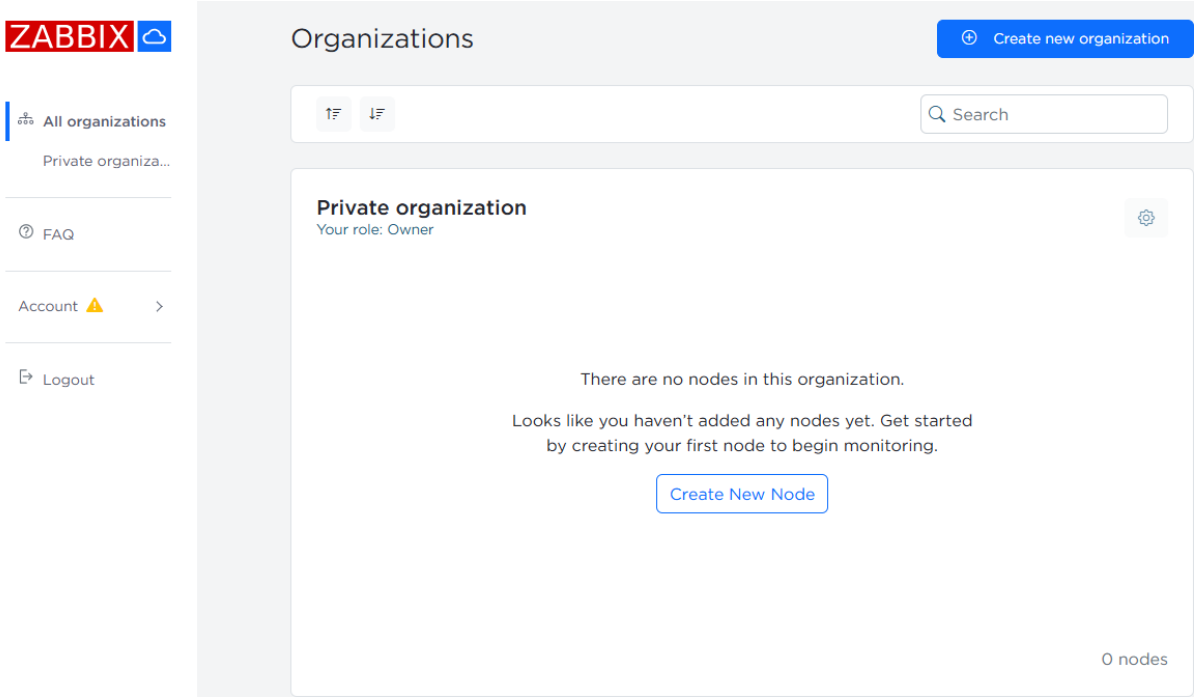
- Acessar o frontend do Zabbix em um nó implantado

Gerenciamento de organizações

Uma organização é um contêiner para um ou mais nós. As organizações também ajudam a estruturar seus recursos e delegar responsabilidades, como:

- Gerenciamento de nós
- Cobrança
- Visualização da atividade do nó

Ao fazer login no Zabbix Cloud, a página *Todas as organizações* é exibida. Cada organização atua como um contêiner para um ou mais nós. 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.

Criando um node

Um node é uma instância Zabbix independente executada em um ambiente de nuvem dedicado. Os nodes fornecem:

- Zabbix server e frontend acessíveis por uma URL exclusiva
- Recursos computacionais dedicados (CPU, memória) e armazenamento em disco
- Isolamento de outros nodes, permitindo testar, monitorar ou atualizar de forma independente
- Implantação baseada em região para soberania de dados e otimização de latência

Para criar um node:

1. Clique em *Criar novo node* na página *Todas as organizações*.

2. Na janela *Criar novo node*, configure o seguinte:

- Nome - insira um nome de exibição para seu node. Após a criação, esse nome aparece como um link clicável na lista de organizações, levando você ao frontend desse node.
- Região - selecione a localização do data center mais próxima dos sistemas que você monitora ou dos usuários finais do seu serviço. Escolher uma região próxima reduz a latência de rede; também considere requisitos de soberania de dados se você coletar métricas sensíveis.

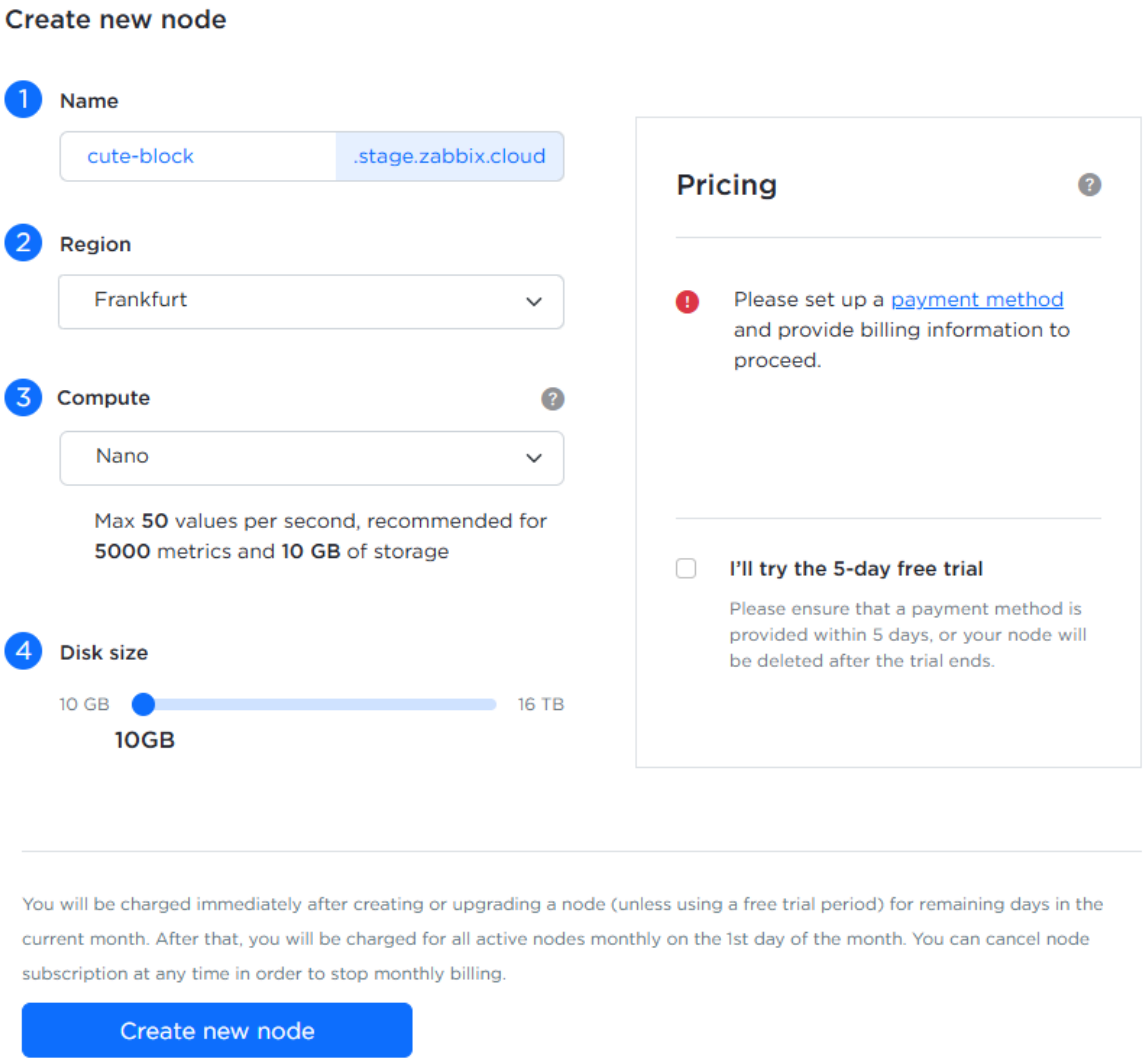
- Nível de recursos - escolha o nível de recursos (CPU, memória) que corresponde à sua carga esperada. Para determinar o nível de recursos necessário, consulte os detalhes de preços. Você sempre pode fazer upgrade posteriormente, se precisar de mais desempenho.

- Tamanho do disco - o controle deslizante é definido por padrão para um mínimo recomendado com base no nível. Aumente se você armazenar grandes volumes de histórico ou tendências. Esteja ciente de que discos maiores geram custos mensais

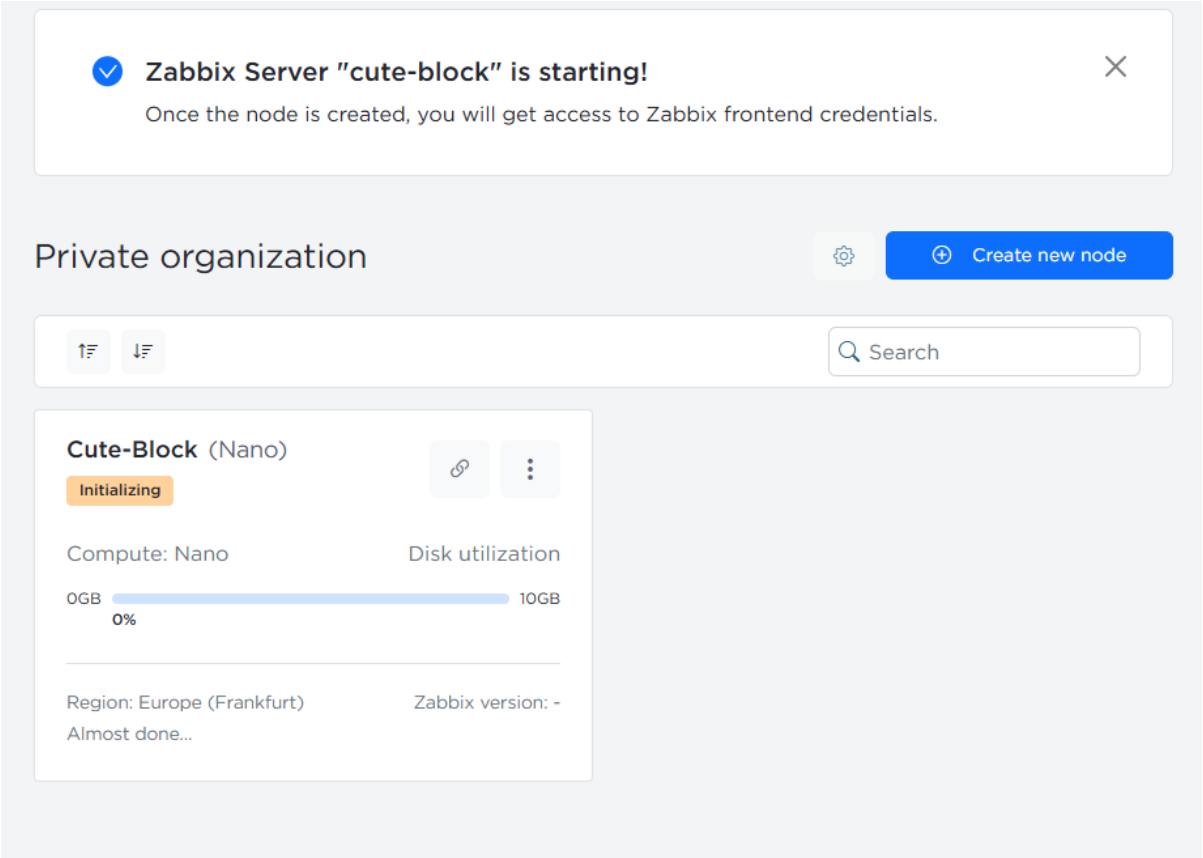
mais altos e que você não pode reduzir o tamanho de um disco por conta própria—será necessário entrar em contato com o suporte para reduzi-lo.

Opcionalmente, ative um teste gratuito de 5 dias antes de criar o node. Ativar o teste permite avaliar o Zabbix Cloud sem custo e não exige informações de cobrança. Você não será cobrado durante o período de teste, mas, após o término, seu node expirará e será removido se não for convertido em um node pago.

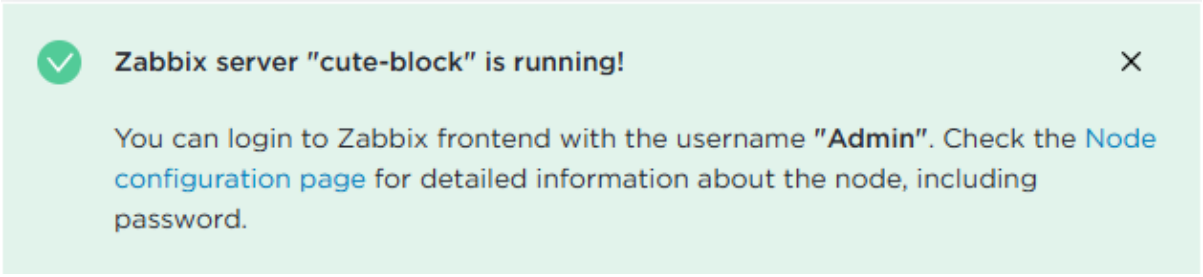
Um método de cobrança será necessário antes de criar um node sem teste. Adicione informações de cobrança pelo menu *Cobrança*.



Após a criação, o node aparecerá na visualização da organização:



Quando seu node estiver em execução, consulte Configuração do node para detalhes sobre o gerenciamento de suas configurações (filtros de acesso, criptografia, backups, etc.).



Após a inicialização do node, seu status muda de *Inicializando* para *Em execução*, e você pode fazer login no frontend do Zabbix em seu node.

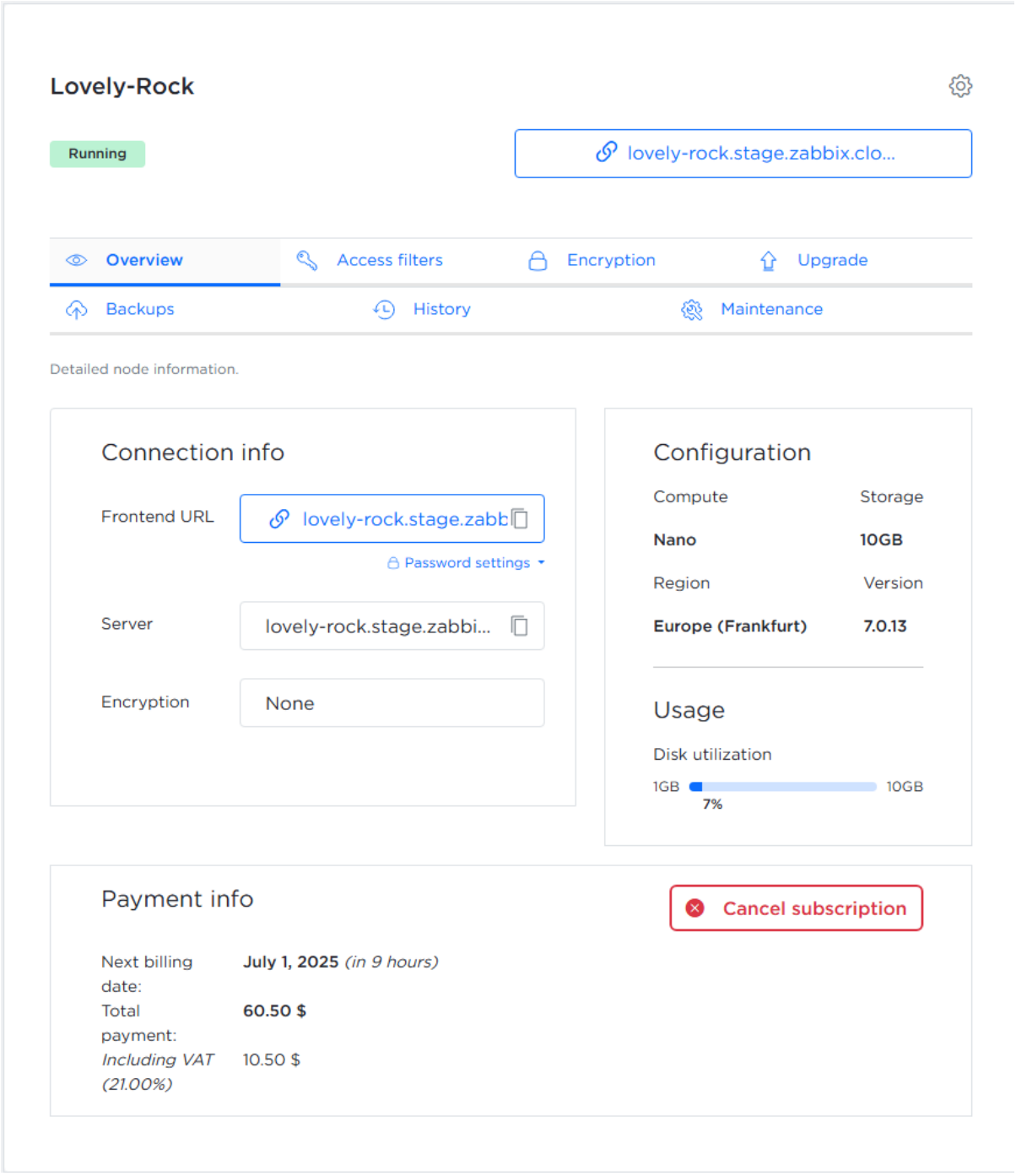
Acessando o frontend do Zabbix

Para acessar o frontend do Zabbix em um nó pela primeira vez:

1. Clique no nó em *Todas as organizações* ou no menu da sua organização para abrir a página de *Configuração do nó*.
2. Na aba *Visão geral*, clique no menu suspenso *Configurações de senha*.

> **Note:**
> Copiar, limpar e redefinir a senha do frontend só pode ser realizado pelo proprietário.

3. Selecione *Copiar senha inicial*.



4. Na página de configuração do nó, clique na URL do frontend (por exemplo, lovely-rock.stage.zabbix.cloud) para abrir o frontend do Zabbix.

Em seguida, faça login usando as seguintes credenciais:

- Usuário - Admin
- Senha - a senha copiada na etapa 4 (por exemplo, HzTG9t7Y)



Veja também:

- Zabbix Cloud FAQ - informações adicionais e respostas para quaisquer dúvidas restantes sobre recursos, implantação, preços e suporte do Zabbix Cloud.

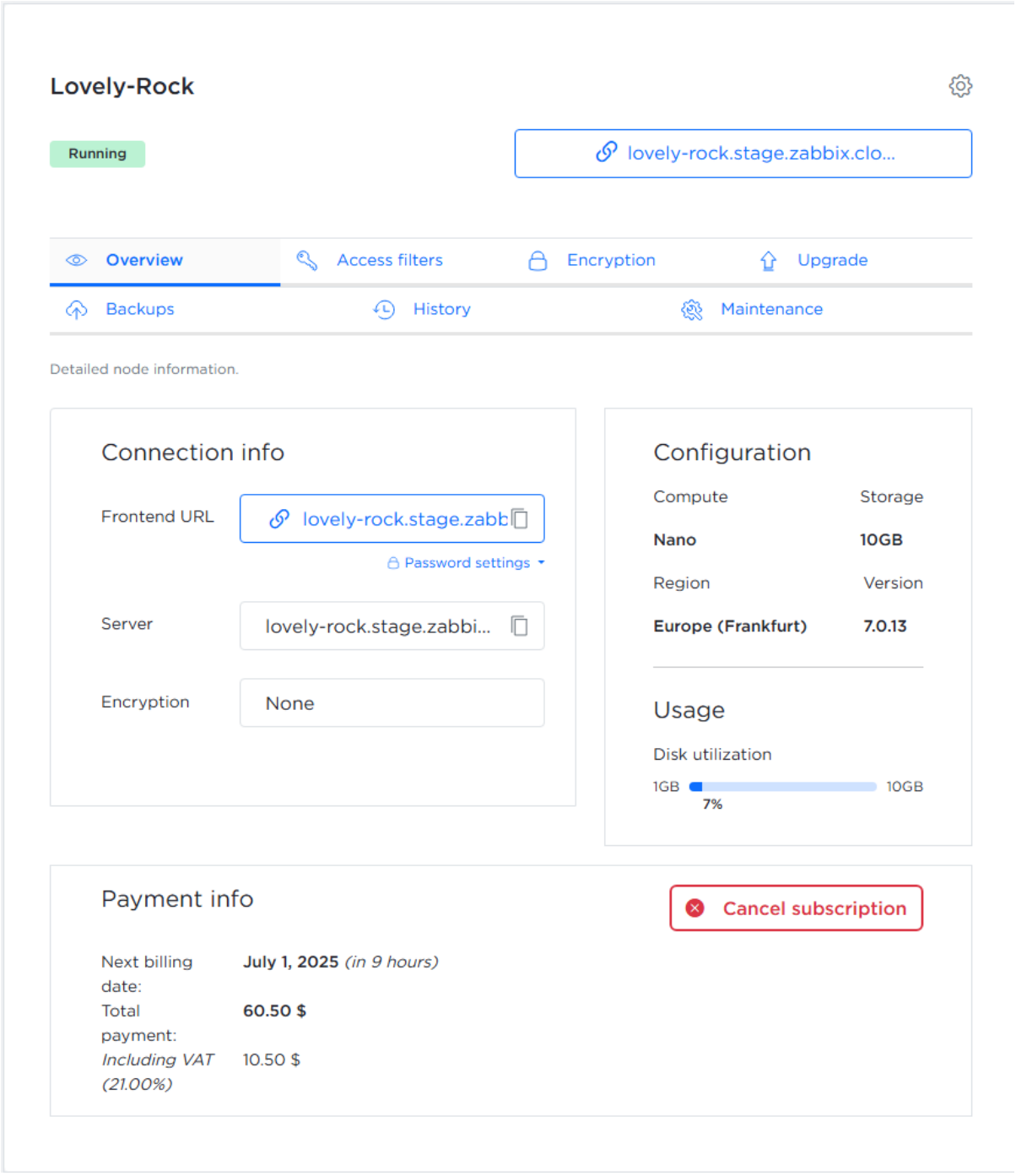
2 Configuração de nó

Visão geral

Para acessar a página de configuração do node, clique no node em *Todas as organizações*, no seu menu de organização, ou usando o link *Página de configuração do node* exibido após um node ser iniciado.

Na visualização de configuração do node, cada aba permite gerenciar um aspecto específico do seu node:

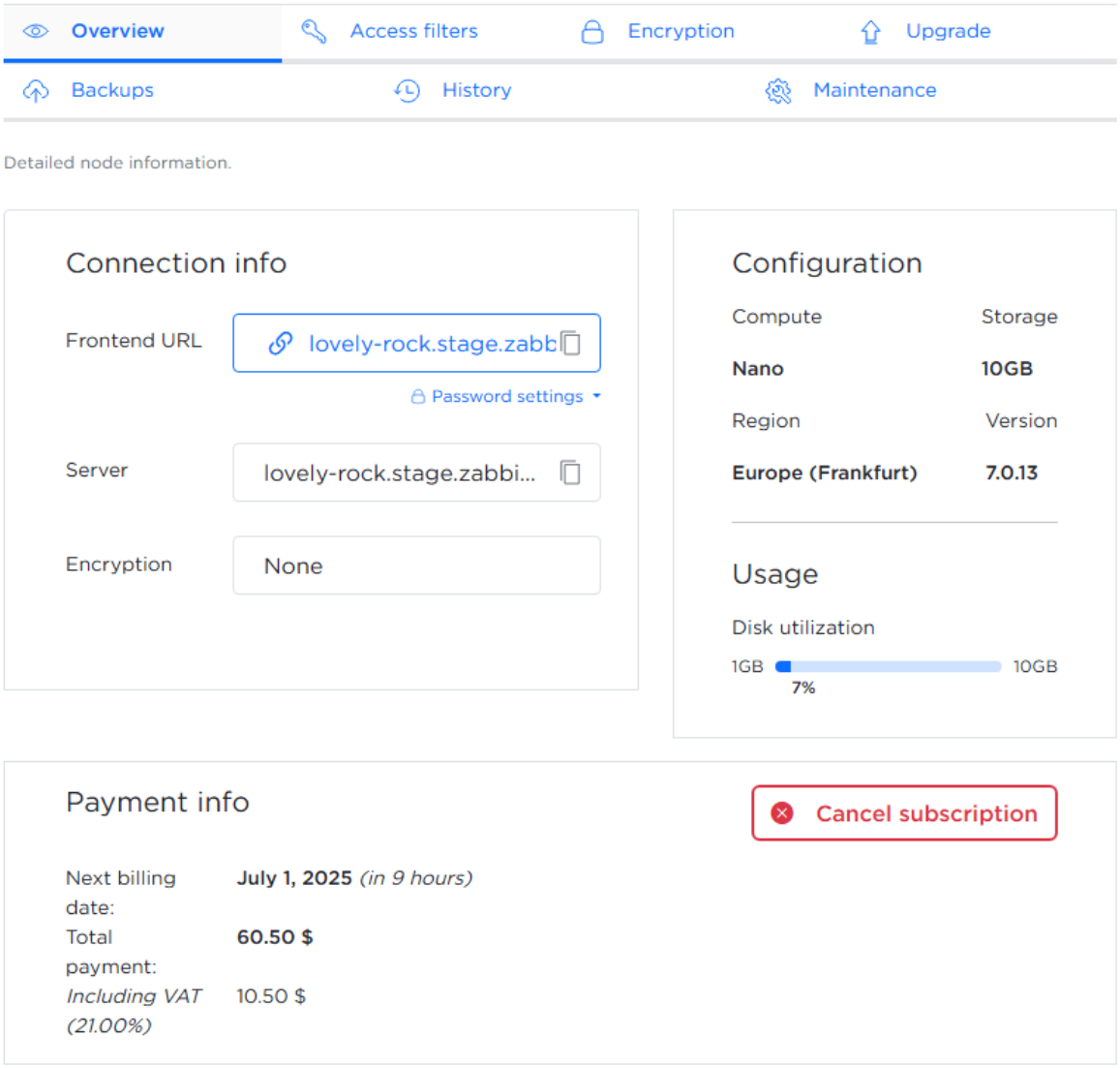
- Visão geral - visualize detalhes do node, abra seu frontend e gerencie usuários do node.
- Filtros de acesso - controle o acesso por endereço IP ou sub-rede.
- Criptografia - configure TLS enviando certificados, chaves e CRLs opcionais.
- Backups - crie, restaure e exclua snapshots do node; visualize o próximo backup agendado.
- Histórico - defina períodos de retenção para logs de auditoria, histórico e tendências, com unidades predefinidas e limite de 3 atualizações/dia.
- Manutenção - defina janelas semanais de manutenção por fuso horário, dia e hora.
- Upgrade - escale recursos de computação/armazenamento ou gerencie seu status de avaliação/pagamento.



Note:
Algumas abas podem não estar disponíveis dependendo do tipo de usuário que visualiza a página e do status do node (trial/não trial).

Aba Visão geral

A aba *Visão geral* fornece informações detalhadas sobre o nó.



Ela é dividida em três seções:

Informações de conexão:

- URL do frontend clicável com opção de copiar para a área de transferência
- Configurações de senha
- Método de criptografia atual

Configuração e uso:

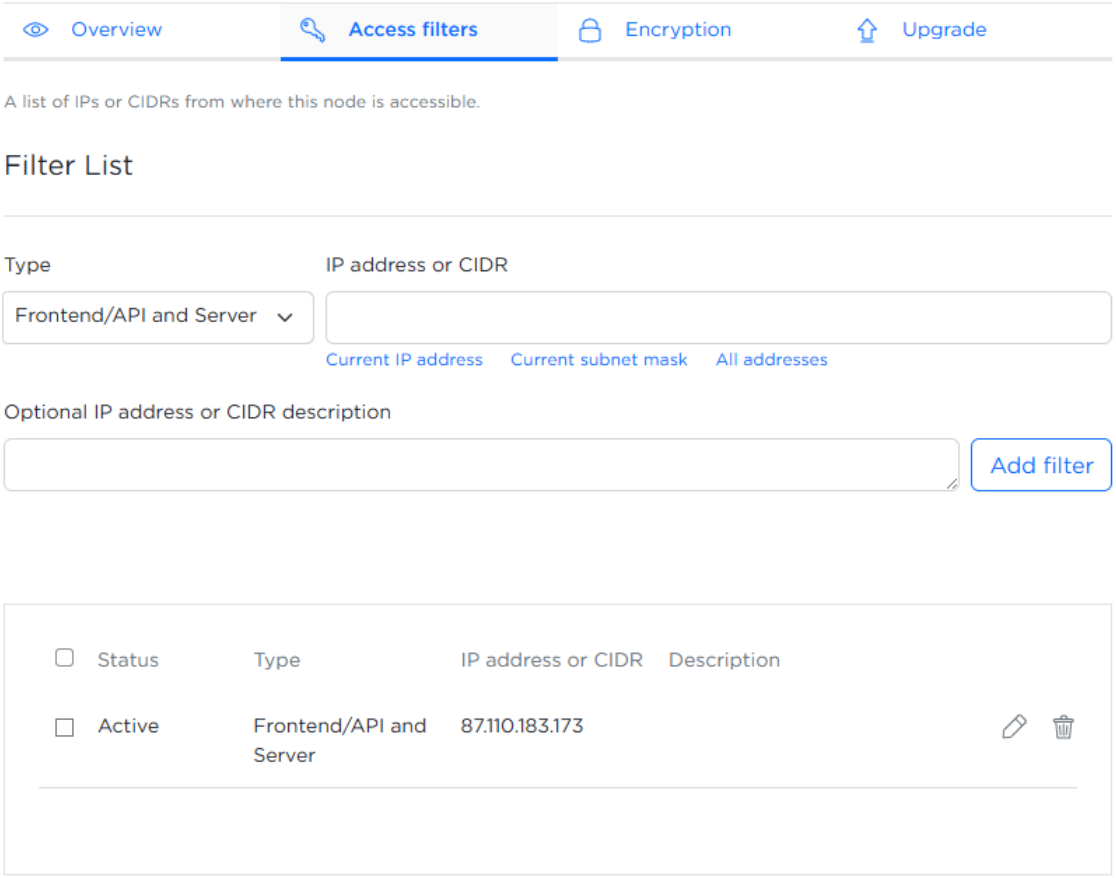
- Nível de computação, armazenamento máximo, região
- Versão dos componentes do Zabbix
- Utilização do disco

Informações do nó/Informações de pagamento:

- Data de expiração (para nós de avaliação e pré-pagos)
- Um botão de contexto específico é exibido dependendo do tipo de nó após a conclusão da inicialização: - *Atualizar para pago* - disponível para nós em avaliação gratuita - *Cancelar assinatura* - disponível para nós pagos com assinatura ativa - *Reassinar para pago* - 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 de 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 do Zabbix limitar: apenas frontend/API, apenas server ou ambos.
- *Endereço IP ou CIDR* - insira o endereço IP ou CIDR manualmente, ou complete o campo automaticamente selecionando a opção relevante.
- *Descrição opcional do endereço IP ou CIDR* - 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 posteriormente você se conectar de um IP ou rede diferente, será necessário adicionar esse endereço aqui ou você será bloqueado.

Você pode:

-  - editar um filtro.
-  - excluir um filtro ou vários filtros usando as caixas de seleção.

Aba Criptografia

A aba *Criptografia* permite configurar a criptografia TLS entre os componentes do Zabbix, fazendo upload dos arquivos de certificado necessários.

Para habilitar a comunicação segura usando certificados, selecione *CERTIFICATE* e faça upload dos seguintes arquivos do seu computador:

- *Certificados Root CA* - usados para verificar a autenticidade dos certificados do par. Clique em *Escolher arquivo* e selecione o arquivo contendo os certificados Root CA confiáveis (normalmente um arquivo `.crt` ou `.pem`, codificado em PEM).

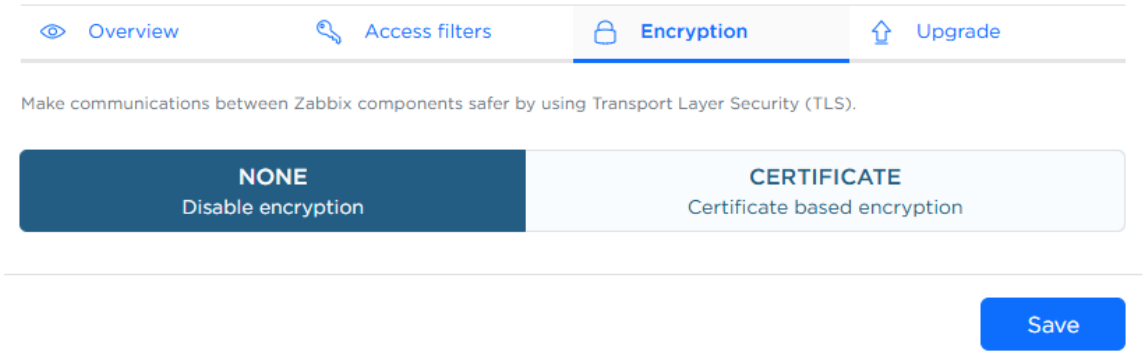
- *Cadeia de certificados* - representa a cadeia de confiança para o seu próprio certificado. Clique em *Escolher arquivo* e selecione o arquivo codificado em PEM (por exemplo, `.crt` ou `.pem`) que inclui seu certificado e quaisquer certificados CA

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 mantiver uma Lista de Certificados Revogados (CRL), clique em *Escolher arquivo* e selecione o arquivo CRL para especificar certificados que não devem mais ser confiáveis.

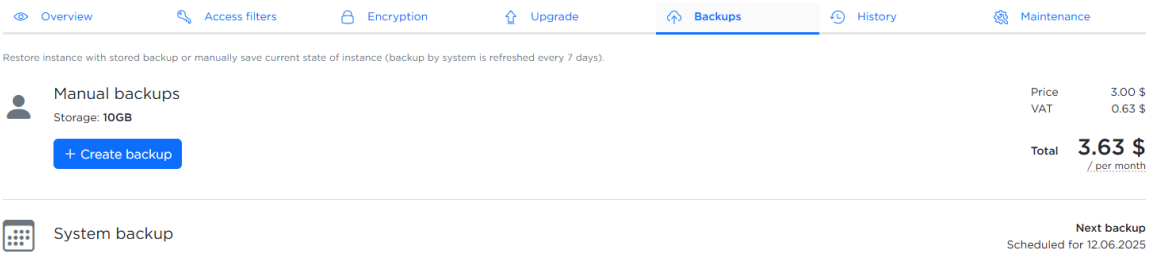
Após selecionar os arquivos, clique em *Salvar*. Os arquivos serão enviados, validados e aplicados para configurar a criptografia TLS entre os componentes do Zabbix.



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.



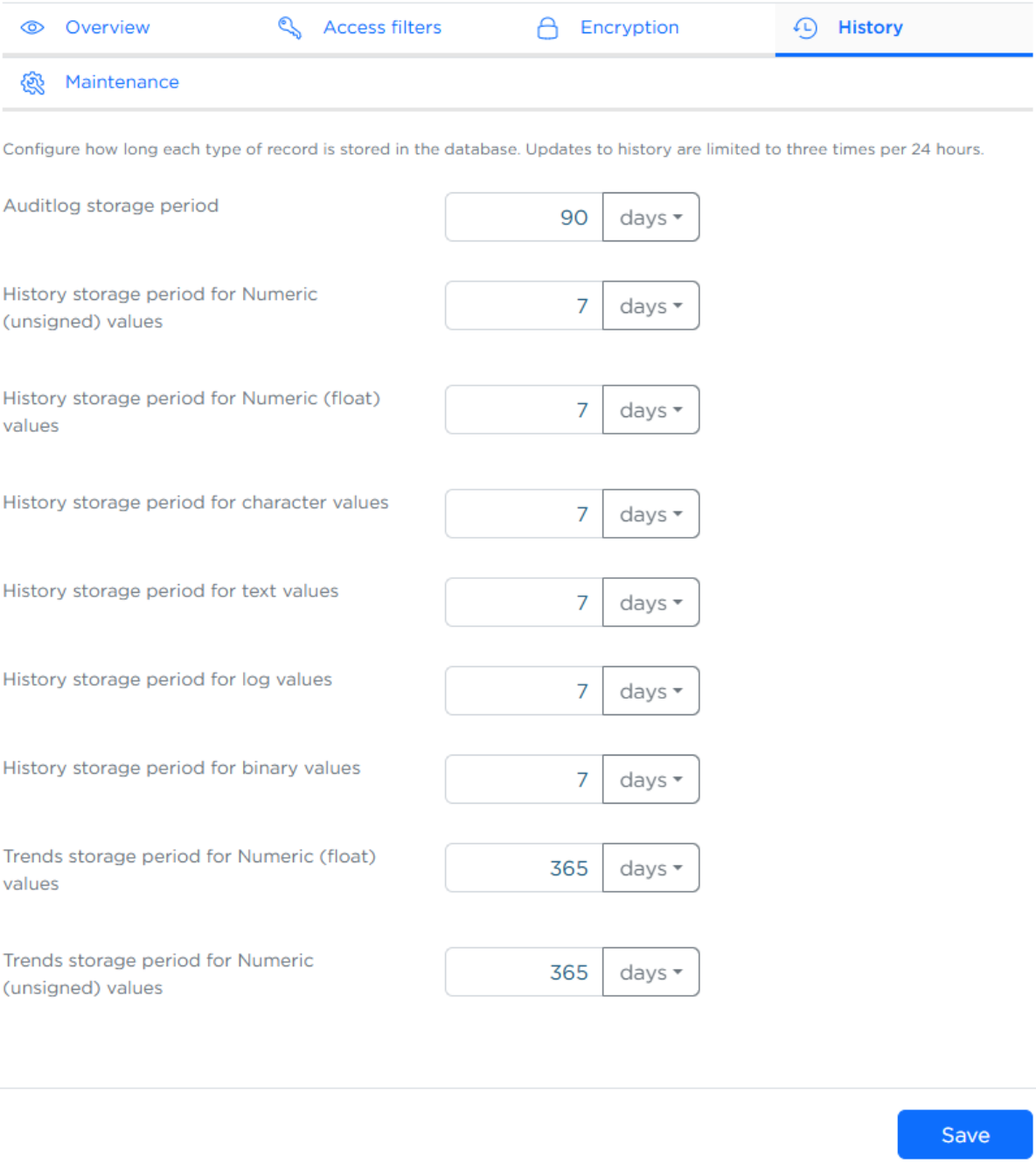
Para criar um backup:

1. Clique no botão *Create backup*.
2. Insira uma descrição opcional na janela pop-up.
3. Clique em *Create* para confirmar.

Warning:
Backups adicionais são cobrados mensalmente com base no uso de 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 de disco e impactar o desempenho. Recomenda-se configurar os períodos de retenção com base nas suas necessidades reais de dados e análise.
 O Zabbix Cloud utiliza particionamento de banco de dados para gerenciar o armazenamento de dados em vez da limpeza tradicional do Zabbix. Isso permite um desempenho mais rápido e um controle de retenção mais previsível.

Aba de 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 um dia e uma janela de horário para manutenção.

Clique em *Salvar* para aplicar as alterações.

Se nenhum dia ou janela de horário for selecionado, um erro será exibido.

Guia de atualização

A guia *Upgrade* permite que você escale os recursos do seu nó ou gerencie sua assinatura.

Você pode aumentar o nível 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 Zabbix para solicitações de redução). Nós de avaliação gratuita podem ser convertidos em assinaturas pagas.

Para aumentar o nível de computação:

1. Clique no menu suspenso *Compute*.
2. Selecione a configuração desejada.
3. Clique no botão *Upgrade*.

Para aumentar o tamanho do disco:

1. Clique no botão de opção *Increase storage size*.
2. Ajuste o controle deslizante *Disk size* para o valor desejado.
3. Clique no botão *Upgrade*.

> **Note:**
> O preço mensal total atualizado é exibido na tabela *Pricing* antes de confirmar as alterações.

3 Adicionando usuários

Introdução

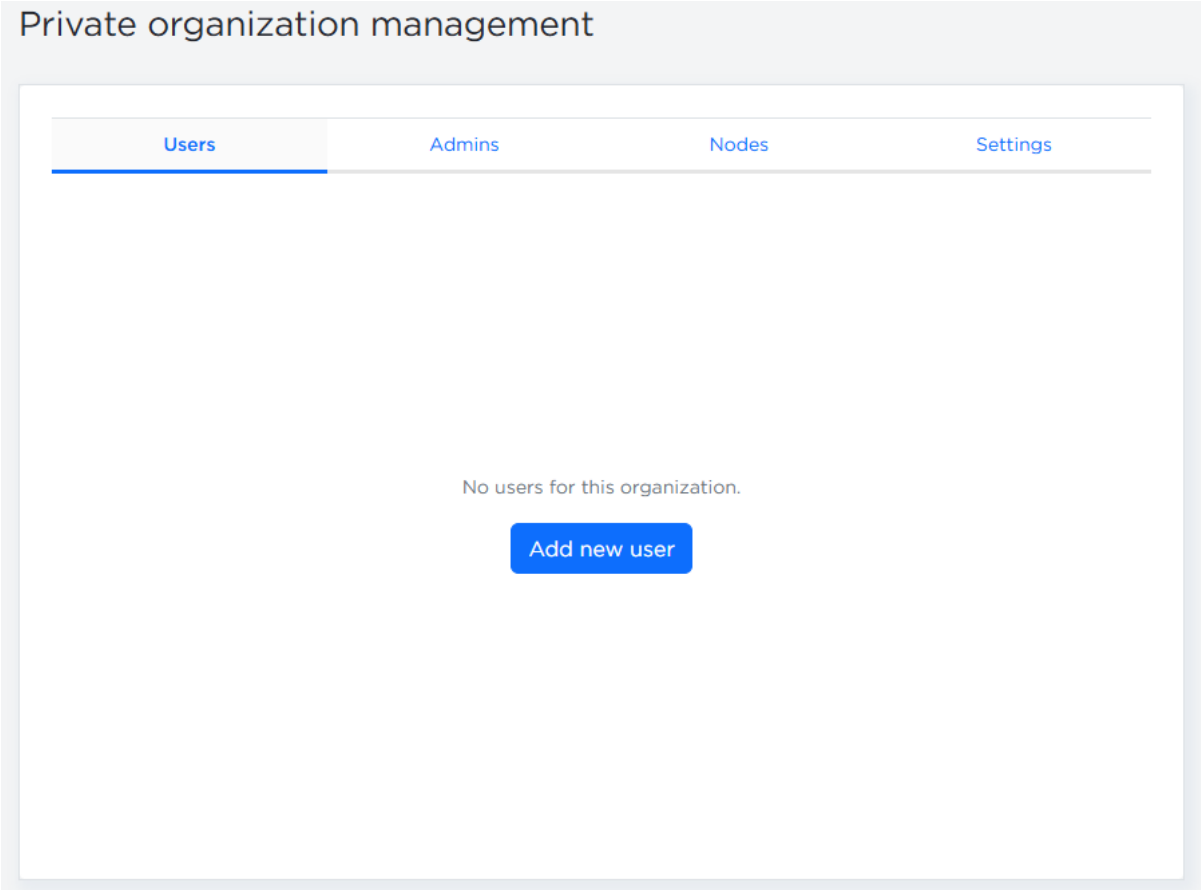
Os exemplos a seguir descrevem como adicionar usuários à sua organização e aos nodes.

Adicionando usuários às organizações

Semelhante aos usuários no Zabbix, você pode criar usuários no Zabbix Cloud e atribuir funções para controlar o que eles podem fazer dentro de uma organização e seus nodes.

Para adicionar usuários à sua organização:

1. Abra a página *Todas as organizações*.
2. Clique no botão de configurações  da organização específica 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 uma função: *Administrador* ou *Usuário*.

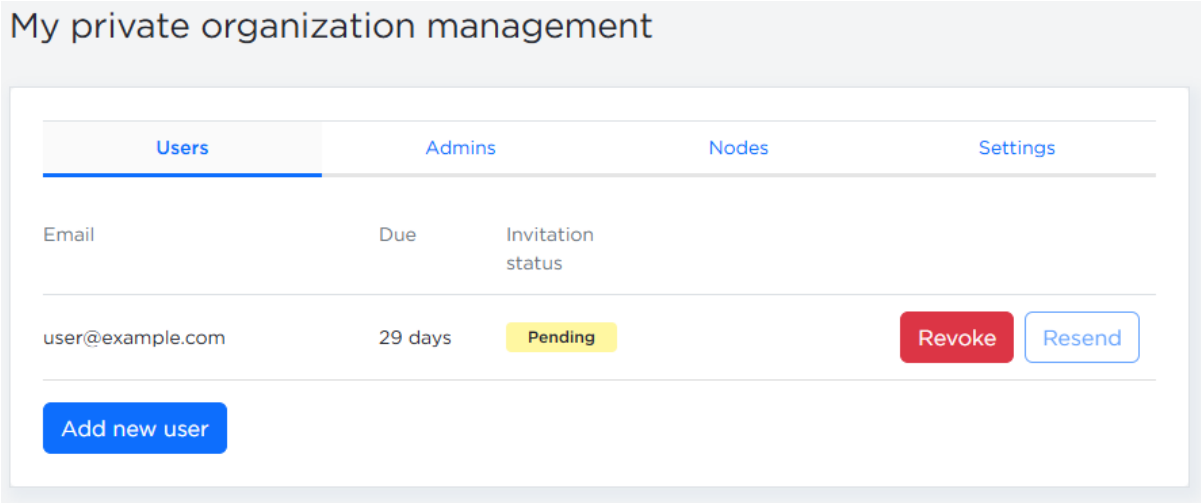
Observe que selecionar a função *Usuário* adiciona o usuário à organização, mas não concede acesso a nenhum node.

Cada função concede um nível diferente de acesso:

- **Usuário** - pode acessar os nodes aos quais está atribuído e pode realizar tarefas de configuração que não geram cobranças, por exemplo:
  - Gerenciar filtros de acesso
  - Configurar criptografia
  - Definir períodos de retenção de histórico/tendência
  - Agendar manutenção
- **Administrador** - possui todas as permissões de Usuário e também pode executar ações que podem gerar cobranças, por exemplo:
  - Criar e restaurar backups
  - Atualizar ou rebaixar recursos do node (nível de computação e tamanho do disco)
  - Cancelar assinaturas
- **Proprietário** - tem controle total sobre a organização. Além de todas as permissões de Administrador, o Proprietário pode:
  - Criar e destruir nodes
  - Adicionar ou remover Usuários e Administradores
  - Renomear ou excluir a organização
  - Gerenciar detalhes de cobrança

Attention:
O método de pagamento do Proprietário é cobrado por todos os serviços. Se for necessária confirmação bancária, o Proprietário deve autorizar a transação, mesmo que tenha sido iniciada por um Administrador.

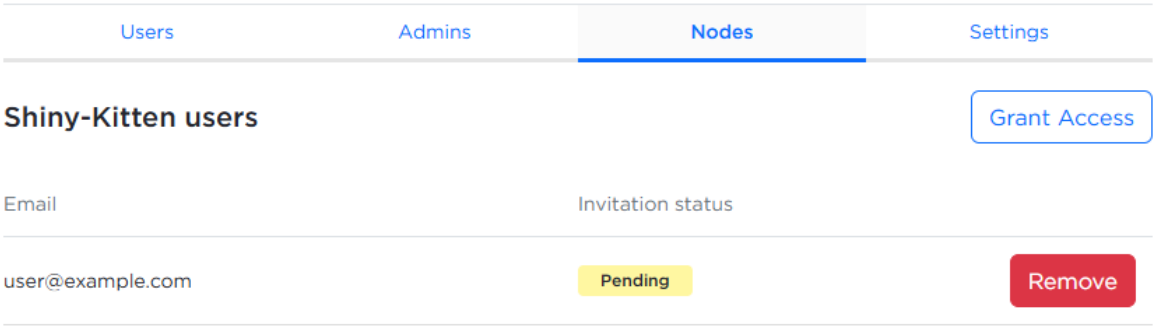
O usuário convidado será adicionado com status *Pendente*. Você pode revogar o convite ou reenviá-lo (o reenvio está disponível uma vez a cada 10 minutos—o botão ficará desabilitado se esse limite for atingido).



Adicionando usuários aos nodes

Para adicionar o usuário ao node:

1. Vá para a aba *Nodes*.
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 a partir da janela de *Configuração do node*:

1. Clique no botão de configurações .
2. Selecione *Adicionar usuário*.
3. Escolha um usuário e clique em *Conceder acesso*.

Os usuários convidados devem aceitar o convite criando uma conta Zabbix Cloud usando o mesmo endereço de e-mail para o qual o convite foi enviado. Se o usuário já possui uma conta Zabbix Cloud com esse e-mail, ele deve fazer login e confirmar o convite. Os convites estão vinculados a um endereço de e-mail específico e não podem ser aceitos a partir de uma conta diferente.

4 Principais diferenças entre Zabbix Cloud e on-premises

Introdução

Esta página destaca as principais limitações e recursos não suportados ao usar o Zabbix Cloud em comparação com uma instalação local (on-premises). Abaixo está uma tabela comparativa resumida; as seções seguintes descrevem itens selecionados em detalhes.

Recurso / Aspecto	Zabbix Cloud	Zabbix on-premises
Correção de vulnerabilidades	Correção automática do SO/Zabbix	Correção manual pelo usuário

Recurso / Aspecto	Zabbix Cloud	Zabbix on-premises
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 máxima velocidade; sem DELETEs longos	Particionamento e limpeza manual (transações DELETE longas possíveis)
Ajuste de desempenho	Ajustado automaticamente pela equipe Cloud (sem changelogs)	Total visibilidade e controle sobre parâmetros de ajuste
Facilidade de uso	Serviço pronto para uso	Requer instalação, configuração e manutenção completas
Controle de firewall	Gerenciado pela equipe 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
Ciclo de lançamento	Mantém-se nas releases LTS	Você escolhe a versão do Zabbix (estável, beta, builds customizadas)
Alta disponibilidade (HA)	Sem HA nativo para aplicação, frontend ou servidor de BD; pode escalar horizontalmente via proxies	HA gerenciado pelo usuário
Chamadas de API	Ilimitadas (sujeito a funções de usuário)	Praticamente ilimitadas (sujeito aos recursos do server)
Traps SNMP	Apenas via Proxy dedicado (sem receptor de traps com HA/balanceamento de carga)	Direto ou via Proxy, HA/balanceamento de carga 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 & 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 síncronas 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 meio de Email (meio de script não suportado)	Suporta ambos meios de script e Email prontos para uso
Zabbix agent 2	Não disponível. Os nós Cloud não executam o Zabbix agent 2	Totalmente suportado; Zabbix agent 2 pode ser instalado e usado localmente
Interface do host para checagens ativas	IP criado automaticamente pode ser externo; limpeza manual necessária para consistência	Interfaces gerenciadas por você; IPs sob seu controle

Diferenças de recursos

Traps SNMP Traps SNMP são suportados 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 exige que cada dispositivo seja exposto via NAT e portas personalizadas, tornando o SNMP polling em larga escala difícil.

Configurações de retenção A configuração dos períodos de retenção de histórico, tendências e auditoria só está disponível pela interface do Zabbix Cloud. Essas configurações não podem ser feitas pelo `zabbix_server.conf` ou pela API.

Substituições manuais de histórico por item não são suportadas (o particionamento é controlado globalmente). Além disso, a URL do nó Cloud não aceita os mesmos parâmetros de consulta que uma instalação local.

Scripts personalizados Os seguintes tipos de scripts personalizados não são suportados no 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 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 é funcional. Se for usado, 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 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 do Cloud ou da API, garantindo que as camadas do sistema operacional e do banco de dados permaneçam isoladas e seguras.

Interface de verificações ativas Verificações ativas na Cloud 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 é suportada, mas o upload de um certificado personalizado ou arquivo de metadados não está disponível no momento.

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.

Zabbix agent 2 Os nós do Zabbix Cloud não executam o Zabbix agent 2, portanto, não é possível usar verificações que o exijam — por exemplo, monitoramento de certificado HTTPS usando o item `web.certificate.get`.

Central do Desenvolvedor

Esta seção contém tudo o que você precisa para começar rapidamente a desenvolver extensões personalizadas para o Zabbix:

- Módulos do frontend
- Widgets do dashboard
- Plugins para o Zabbix agent 2

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 em qualquer 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 similar, desde que a documentação seja distribuída juntamente com o software na mesma mídia. Qualquer outro uso, como qualquer disseminação 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.

Módulos

O que é um módulo 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* da sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).
- Um módulo deve seguir regras simples para garantir o funcionamento correto.
- Um módulo deve ser instalado (descompactado) e habilitado no frontend pelo administrador.

Para que um módulo pode ser usado

- Adicionar novas funcionalidades por meio de seções personalizadas no frontend;
- Criar tipos personalizados de widgets de dashboard (veja módulos de widget);
- Sobrescrever ou estender a funcionalidade existente.

O que um módulo não pode ser usado para

- Registrar um novo método de API ou modificar um existente.

Como os módulos funcionam

- Um módulo habilitado é iniciado em cada solicitação HTTP, antes de executar o código da ação.
- O módulo irá registrar novas ações ou redefinir as existentes.
- O módulo irá adicionar novas seções ao frontend e remover ou redefinir as existentes.
- O módulo irá se conectar a eventos do frontend como onBeforeAction e onTerminate, se necessário.
- A ação solicitada é finalmente executada executando o código da ação - seja o padrão ou o definido pelo módulo.

Onde ir em seguida Se você prefere aprender fazendo ou ler as diretrizes primeiro, estas páginas contêm as informações e etapas necessárias para criar seus próprios módulos:

- Tutoriais passo a passo para escrever seu primeiro módulo
- Estrutura de arquivos do módulo
- Especificidades do módulo de widget
- Exemplos de módulos para reutilizar

Estrutura de arquivos do módulo

Todo o código relacionado a um módulo é armazenado em um único diretório dentro do diretório **modules** da sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).

```
example_module_directory/          (obrigatório)
   manifest.json                   (obrigatório)  Metadados e definição de ação.
   Module.php                                     Inicialização do módulo e manipulação de eventos.
   actions/                                       Arquivos de controladores de ação.
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                         Arquivos de visualização.
       example.something.view.php
       example.something.delete.php
   assets/                                        Quaisquer arquivos adicionais a serem usados nas visualizaç
       js/                                        Arquivos JavaScript usados nas visualizações.
           example.something.view.js.php
       css/                                       Arquivos CSS usados nas visualizações.
           example.something.css
```

```
        image.png                          Imagens usadas nas visualizações.
        example.something.file             Qualquer arquivo para uso nas visualizações.
```

Estrutura de arquivos do módulo

Escrevendo um módulo Um processo de escrita de módulo de exemplo consiste nas seguintes etapas (quando disponível, clique no nome do arquivo ou pasta para ver 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 views do módulo.
4. Crie a pasta *actions* e defina uma ou mais actions do módulo.
5. Crie o arquivo Module.php (ou Widget.php para widgets de dashboard) e defina as regras de inicialização e manipulação 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 views, actions e arquivos de asset necessários no manifest.json.
8. Registre o módulo no frontend do Zabbix e comece a usá-lo.

Note:
Você pode registrar e habilitar um módulo assim que criar o arquivo manifest.json. Assim que o módulo estiver habilitado, você poderá visualizar imediatamente todas as alterações feitas nos arquivos do módulo apenas atualizando o frontend 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": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Parâmetros suportados em **manifest.json** (clique no nome do parâmetro para uma descrição detalhada):

Parâmetro	Descrição	Obrigatório
manifest_version	Versão do manifesto do módulo.	Sim
id	ID único do módulo.	
name	Nome do módulo que será exibido na seção Administração.	
namespace	Namespace PHP para as classes do módulo.	
version	Versão do módulo.	
type	Tipo do módulo. Para widget deve ser definido como *widget*	Sim para widgets, caso contrário não
widget	Configuração do widget. Usado apenas para widgets.	
actions	Ações para registrar com o 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 suportada é **2.0**.

Tipo: Double

Exemplo:

```
"manifest_version": 2.0
```

id

ID do módulo. Deve ser único. Para evitar conflitos de nomes futuros, recomenda-se usar um prefixo para os módulos (nome do autor ou da empresa, ou qualquer outro). Por exemplo, se um módulo for um exemplo para lições e o nome do módulo for ”Meu módulo”, então o ID será ”example_my_module”.

Tipo: String

Exemplo:

```
"id": "example_my_module"
```

name

Nome do módulo que será exibido na seção Administração.

Tipo: String

Exemplo:

```
"name": "Meu módulo"
```

namespace

Namespace PHP para classes de módulo.

Tipo: String

Exemplo:

```
"namespace": "ClockWidget"
```

version

Versão do módulo. A versão será exibida na seção Administração.

Tipo: String

Exemplo:

```
"version": "1.0"
```

type

Tipo do módulo. Obrigatório para widgets e deve ser igual a ”widget”.

Tipo: String

Padrão: ”module”

Exemplo:

```
"type": "widget"
```

actions

Ações a serem registradas com o módulo. Definir a chave do objeto *class* para cada ação é obrigatório, outras chaves de ação são opcionais.

Tipo: Objeto

Chaves de objeto suportadas se *type* for *module*:

- **write.your.action.name** (objeto) - nome da ação, deve ser escrito em 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 suportadas se *type* for *widget*:

- **widget.{id}.view** (objeto) - nome do arquivo e da classe para a visualização do widget. Substitua **{id}** pelo valor id do widget (por exemplo, *widget.example_clock.view*). Suporta as chaves:
  - **class** (string; obrigatório) - nome da classe da ação para o modo de visualização do widget para estender a classe padrão CControllerDashboardWidgetView. O arquivo fonte 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*, que é o esperado por padrão, este parâmetro pode ser omitido. Se estiver usando um nome diferente, especifique-o aqui.
- **widget.{id}.edit** (objeto) - nome do arquivo para a visualização da 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 da ação para o modo de visualização da configuração do widget. O arquivo fonte da classe deve estar localizado no diretório *actions*.
  - **view** (string) - visualização da configuração do widget. Deve estar localizada no diretório *views*. Se o arquivo de visualização for *widget.edit.php*, que é o esperado por padrão, este parâmetro pode 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"
    }
}
```

assets

Estilos CSS e arquivos JavaScript para incluir.

Tipo: Objeto

Chaves de objeto suportadas:

- **css** (array) - arquivos CSS para incluir. Os arquivos devem estar localizados em *assets/css*.
- **js** (array) - arquivos JavaScript para incluir. Os arquivos devem estar localizados em *assets/js*.

Exemplo:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

author

Autor do módulo. O autor será exibido na seção Administração.

Tipo: String

Exemplo:

```
"author": "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 podem ser alterados diretamente no banco de dados.

Tipo: Objeto

Exemplo:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

description

Descrição do módulo.

Tipo: String

Exemplo:

```
"description": "This is a clock widget."
```

widget

Configuração do widget. Usado se *type* estiver definido como *widget*.

Tipo: Objeto

Chaves de objeto suportadas:

- **name** (string) - usado na lista de widgets e como cabeçalho padrão. Se vazio, o parâmetro "name" do módulo será usado.
- **size** (objeto) - dimensões padrão do widget. Suporta as chaves:
  - *width* (inteiro) - largura padrão do widget.
  - *height* (inteiro) - altura padrão do widget.
- **form_class** (string) - classe com o formulário de campos do widget. Deve estar localizada no diretório *includes*. Se a classe for *WidgetForm.php*, que é esperada por padrão, este parâmetro pode ser omitido. Se estiver usando um nome diferente, especifique-o aqui.
- **js_class** (string) - nome de uma classe JavaScript para o modo de visualização do widget para estender a classe padrão CWidget. A classe será carregada com o dashboard. O arquivo fonte da classe deve estar localizado no diretório assets/js. Veja também: assets.
- **use_time_selector** (booleano) - determina se o widget requer o seletor de tempo do dashboard. Valores suportados: *true*, *false* (padrão).
- **refresh_rate** (inteiro) - 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, este link será aberto ao clicar no ícone de ajuda ? na janela *Adicionar widget* ou *Editar widget*. Se **url** não for especificado, ao clicar no ícone de ajuda será aberta a página geral Widgets do dashboard.

Tipo: String

Exemplo:

```
"url": "http://example.com"
```

Ações

As ações são responsáveis pela 'lógica de negócio' do módulo. Uma ação geralmente consiste em um controller e uma action view.

Um módulo pode:

- Chamar ações que já estão definidas no frontend do Zabbix.
- Sobrescrever ações padrão com ações personalizadas.
- Definir ações completamente novas.

Para sobrescrever o comportamento padrão de uma ação com algum comportamento personalizado, defina uma ação com o mesmo nome na configuração do módulo. Quando a ação for chamada, a ação do módulo será executada em vez da ação padrão do Zabbix.

Os arquivos de ação devem ser armazenados na pasta *actions*. As ações precisam ser especificadas no manifest.json.

Controller

Fluxo de trabalho do controller de ação:

1) Verifique se todos os parâmetros passados em uma solicitação HTTP são válidos:

- Chame o método *checkInput()* do controller
- 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 verdadeiro, o Zabbix chama o método *doAction()* do controller.

4) Prepare o array **$data** para a view. Use *CControllerResponseData* e o método *setResponse()* para armazenar a resposta no array **$data**.

Exemplo:

```
/**
 * Valida 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;
}

/**
 * Verifica as permissões do usuário.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Executa a ação e gera o objeto de resposta.
 */
protected function doAction(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
```

```
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
Você pode visualizar a lista completa de classes de controller disponíveis no código-fonte do Zabbix.

Visualizações

O arquivo de visualização recebe os dados de um controlador e, em seguida, prepara a aparência HTML deles.

Note:
Definir visualização(ões) para um módulo do frontend é opcional, a menos que o módulo seja um widget.
Widgets do dashboard precisam de pelo menos duas visualizações: uma para o modo de edição e outra para o modo de visualização (devem ser armazenadas no diretório *views*).

É possível usar classes HTML pré-definidas do Zabbix (de */zabbix/ui/include/classes/html*) na visualização, bem como adicionar novas classes HTML e CSS. Novas classes devem ser armazenadas na pasta *assets* do módulo.

Exemplo:

```
...
(new CColHeader(_('Nome')))
```

Isso adicionará uma nova coluna chamada *Nome* e estilizará a linha superior da tabela como em outras páginas do Zabbix.

Visualização de ação

Este é um arquivo de referência para definir uma visualização de ação.

```
<?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();
```

Assets

A pasta *assets* pode conter quaisquer arquivos e subpastas que não pertençam a outros diretórios. Você pode usá-la para:

- Estilos JavaScript (devem estar dentro de *assets/js*);
- Estilos CSS (devem estar dentro de *assets/css*);
- Imagens;
- Fontes;
- Qualquer outra coisa que você precise incluir.

assets/js

O diretório *assets/js* é reservado e deve conter apenas arquivos JavaScript. Para serem usados pelo widget, especifique esses arquivos no *manifest.json*.

Por exemplo:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

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 estilos diferentes para temas específicos do frontend.

Temas disponíveis e seus valores de atributo:

- **Azul** - [theme='blue-theme']
- **Escuro** - [theme='dark-theme']
- **Alto contraste claro** - [theme='hc-light']
- **Alto contraste escuro** - [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 ao frontend do Zabbix.

Pré-requisitos Antes de prosseguir, certifique-se de que:

- O módulo está localizado dentro do diretório *modules* da sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).
- O módulo possui pelo menos uma versão básica do arquivo manifest.json.
- Você tem acesso à seção de menu Administração no Zabbix (requer o tipo de função de usuário Super admin).

> **Note:**
> O frontend não instalará nem reconhecerá módulos incompatíveis.

Adicionando um módulo Abra a página *Administração→Geral→Módulos* e pressione *Verificar diretório*.

Localize seu módulo na lista e ative-o.

Para ativar um módulo, pressione o hiperlink *Desativado* - o estado do módulo mudará para *Ativado*.

Pressione o nome do módulo para visualizar informações adicionais sobre o módulo, como autor, versão ou uma breve descrição (se definida no manifesto).

Visualização do widget Os módulos de widget, uma vez adicionados, tornam-se imediatamente visíveis na lista de widgets do dashboard.

Você pode abrir um dashboard, alternar para o modo de edição e adicionar o widget ao dashboard normalmente.

Quando você fizer algumas alterações no widget, atualize o dashboard para ver como o widget fica com as atualizações mais recentes.

Widgets

Widgets são módulos do frontend do Zabbix usados para os painéis. Salvo indicação em contrário, todas as diretrizes de módulos também são aplicáveis aos widgets.

No entanto, um widget é notavelmente diferente de um módulo. Para construir um widget:

- especifique o tipo ”widget” no arquivo manifest.json (”type”: ”widget”);
- inclua pelo menos duas views: uma para o modo de apresentação do widget e outra para o modo de configuração do widget (example.widget.view.php e example.widget.edit.php);
- e um controller para apresentação do widget (WidgetView.php);
- use e estenda as classes de widget padrão.

Configuração

Esta página descreve as classes que podem ser usadas para criar uma visualização de configuração de widget com campos de configuração personalizados. A visualização de configuração do widget é a parte do widget que permite ao usuário configurar os parâmetros do widget para apresentação.

Widget

Classe principal do widget, estende a classe base de todos os widgets do dashboard - *CWidget*. Necessário para sobrescrever o comportamento padrão do widget.

A classe *Widget* deve estar localizada no diretório raiz do widget (por exemplo, *zabbix/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;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

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 para lidar com a validação de entrada.

A classe *WidgetForm* deve estar localizada no diretório *includes*. Se a classe tiver um nome diferente, o nome deve 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 {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D'
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

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 visualização de configuração.

A classe *CWidgetFormView* suporta os seguintes métodos:

- *addField()* - recebe uma instância da classe *CWidgetFieldView* como parâmetro; cada classe *CWidgetField* possui uma respectiva classe *CWidgetFieldView* para uso na visualização de configuração do widget.
- *addFieldset()* - recebe uma instância da classe *CWidgetFieldsGroupView* que combina campos em um contêiner recolhível.
- *addFieldsGroup()* - recebe uma instância de *CWidgetFormFieldsetCollapsibleView* que visualmente (com uma borda) combina campos em um grupo.
- *includeJsFile()* - permite adicionar um arquivo JavaScript à visualização de configuração do widget.
- *addJavaScript()* - permite adicionar JavaScript inline 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

/**
 * Minha visualização de formulário de widget personalizada.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
```

```
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

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 singleton da classe é imediatamente criada e armazenada sob o nome *window.my_custom_widget_form*. Assim, ao abrir o formulário pela segunda vez, a instância será recriada.

A classe JavaScript deve estar localizada no diretório *views*.

Exemplo de views/widget.edit.js.php

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

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 do banco de dados	Descrição
CWidgetFieldCheckBox	int32	Caixa de seleção única.
CWidgetFieldCheckBoxList	array de int32	Múltiplas caixas de seleção sob um único campo de configuração.
CWidgetFieldColor	string	Campo de seleção de cor.

Classe CWidgetField	Tipo de campo do banco de dados	Descrição
CWidgetFieldDatePicker	string	Campo de seleção de data.
CWidgetFieldHostPatternSelect	string	Campo de seleção múltipla que permite selecionar um ou mais hosts. Suporta a definição de padrões de nomes de host (todos os hosts correspondentes serão selecionados).
CWidgetFieldIntegerBox	int32	Campo para inserir um número inteiro. Pode ser usado para configurar valores mínimos e máximos.
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 items.
CWidgetFieldMultiSelectItemPattern	ID	Campo de seleção múltipla para selecionar padrões de items.
CWidgetFieldMultiSelectItemPrototype	ID	Campo de seleção múltipla para selecionar protótipos de items.
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 (dashboard 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 decimal.
CWidgetFieldRadioButtonList	int32	Grupo de botões de opção que consiste em um ou mais botões de opção.
CWidgetFieldRangeControl	int32	Controle deslizante para selecionar um valor do tipo inteiro.
CWidgetFieldReference	string	Cria um identificador exclusivo para este widget no dashboard. Ele é usado para referenciar este widget a partir de outros widgets.
CWidgetFieldSelect	int32	Caixa de seleção suspensa.
CWidgetFieldSeverities	array de int32	*CWidgetFieldCheckBoxList* predefinido com severidades de triggers.
CWidgetFieldTags	array de (string, int32, string)	Permite configurar uma ou mais linhas de filtro de tags.
CWidgetFieldTextArea	string	Área de texto para inserir texto de várias linhas.
CWidgetFieldTextBox	string	Caixa de texto para inserir texto de uma única linha.
CWidgetFieldTimePeriod	array de string	Campo de seleção de período de tempo.
CWidgetFieldTimeZone	string	Lista suspensa com fusos horários.
CWidgetFieldThresholds	array de (string, string)	Permite configurar pares de cor e número.
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	array de (múltiplos tipos mistos)	Para o widget *Top hosts*. Cria uma tabela com colunas personalizadas de tipos permitidos.
CWidgetFieldNavTree	string	Para o widget *Árvore de navegação de mapas*. Substitui a visualização do widget no modo de edição pela árvore de seleção de mapas.

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 dashboard em um contêiner.

A visualização de apresentação consiste em três partes:

- Ação do widget
- Visualização do widget
- JavaScript

Ação do widget

A classe de ação do widget (*WidgetView*) contém métodos para operações com widgets no modo de visualização 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 de actions/WidgetView.php (implementado no widget nativo do Zabbix *Informações do sistema*)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Visualização do widget

A classe de visualização do widget (*CWidgetView*) é responsável por construir 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 *actions* do arquivo *manifest.json* (*actions/widget.{id}.view/view*).

Exemplo de views/widget.view.php

```
<?php

/**
 * Minha visualização de widget personalizada.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

A classe JavaScript é responsável por determinar o comportamento do widget, como atualização dos dados do widget, redimensionamento do widget, exibição de elementos do widget, etc.

Todas as operações JavaScript utilizam e/ou estendem a classe base JavaScript de todos os widgets do dashboard - *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: *onInitialize()*, *onStart()*, *onActivate()*, *onDeactivate()*, *onDestroy()*, *onEdit()*.
- Métodos que lidam com a atualização e exibição dos dados do widget: *promiseUpdate()*, *getUpdateRequestData()*, *processUpdateResponse(response)*, *processUpdateErrorResponse(error)*, *setContents(response)*.
- Métodos que modificam a aparência do widget: *onResize()*, *hasPadding()*.

A classe JavaScript deve estar localizada no diretório *assets/js* e especificada no parâmetro *assets* (*assets/js*) no arquivo *manifest.json*.

Métodos do ciclo de vida

Os métodos do ciclo de vida do widget são invocados pelo dashboard, e em diferentes estágios do ciclo de vida do widget durante sua existência dentro do dashboard.

O método ***onInitialize()*** define o estado inicial e/ou valores do widget, sem realizar qualquer manipulação de HTML ou dados. Este método é invocado quando um widget é criado (um objeto widget é instanciado), normalmente ao adicionar o widget a uma página de dashboard ou ao carregar a página do dashboard.

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 qualquer manipulação de dados. Este método é invocado antes da primeira ativação da página do dashboard, ou seja, antes que o dashboard 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 dashboard é ativada, ou seja, quando ela é totalmente exibida na interface do usuário.

Observe que antes do método *onActivate()* ser invocado, 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, escuta eventos, atualiza seu conteúdo periodicamente e pode interagir com outros widgets.

Exemplo:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._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 dashboard é desativada, ou seja, alternada ou excluída, ou quando o widget é excluído da página do dashboard.

Observe que antes do método *onDeactivate()* ser invocado, 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 dashboard, o que pode incluir fechar uma conexão de banco de dados que foi estabelecida durante a inicialização do widget, limpar dados temporários para liberar a memória do sistema e evitar vazamentos de recursos, cancelar o registro de listeners de eventos relacionados a eventos de redimensionamento ou cliques de botões para evitar o tratamento desnecessário de eventos e vazamentos de memória, etc. Este método é invocado quando o widget ou a página do dashboard que o contém é excluído.

Observe que antes que o método *onDestroy()* seja invocado, 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);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

O método ***onEdit()*** define a aparência e o comportamento do widget quando o dashboard entra no modo de edição. Este método é chamado 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();
}
```

Métodos do processo de atualização

Os métodos do processo de atualização do widget são responsáveis por recuperar dados atualizados do Zabbix server 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 ao server para atualizar o widget, reunindo várias propriedades e seus respectivos valores (identificadores do 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)*** trata 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 de 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 responsividade 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, um exemplo da implementação do método *hasPadding()* utilizado 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 frontend do Zabbix. Você pode baixar todos os arquivos deste módulo como um arquivo ZIP: MyAddress.zip.

O que você irá construir

Durante este tutorial, você irá primeiro construir um módulo de frontend que adiciona uma nova seção de menu *Meu Endereço* e depois convertê-lo em um módulo de frontend mais avançado que faz uma requisição HTTP para *https://api.seeip.org* e exibe a resposta - o endereço IP do seu computador - em uma nova página na seção de menu *Meu Endereço* recém-criada. Veja como o módulo finalizado ficará:

Parte I - Nova seção de menu

Adicione um módulo em branco ao frontend do Zabbix

1. Crie um diretório *MyAddress* no diretório *modules* da sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).
2. Crie um arquivo *manifest.json* com os metadados básicos do módulo (consulte a descrição dos parâmetros suportados).

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. No frontend do Zabbix, vá para a seção *Administração → Geral → Módulos* e clique no botão *Procurar diretório*.
4. Localize o novo módulo *My IP Address* 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).

O módulo agora está registrado no frontend. No entanto, ainda não está visível, pois você ainda precisa definir a funcionalidade do módulo. Assim que você adicionar conteúdo ao diretório do módulo, verá imediatamente as alterações no frontend do Zabbix ao atualizar a página.

Crie uma seção de menu

1. Crie um arquivo *Module.php* no diretório *MyAddress*.

Este arquivo implementa uma nova classe *Module* que estende a classe padrão *CModule*. A classe *Module* irá inserir uma nova seção de menu *Meu Endereço* 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 *Perfil do usuário*. Na parte III deste tutorial, você aprenderá como criar uma ação personalizada.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

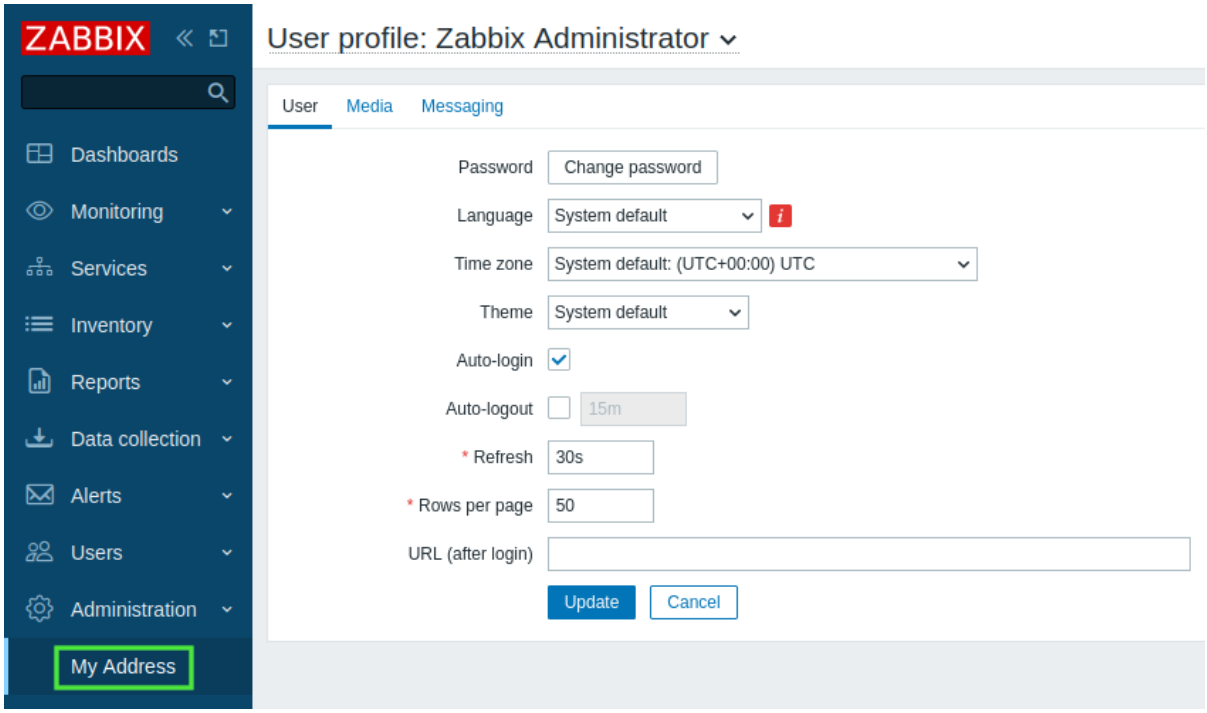
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('Meu Endereço')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Você pode substituir `'userprofile.edit'` por outras ações, por exemplo, `'charts.view'` (abre gráficos personalizados), `'problems.view'` (abre *Monitoramento → Problemas*) ou `'report.status'` (abre o relatório *Informações do sistema*).

3. Atualize o frontend do Zabbix. Agora há uma nova seção *Meu Endereço* na parte inferior do menu principal do Zabbix. Clique em *Meu Endereço* para abrir a página *Perfil do usuário*.



Parte II - Alteração da localização da seção do menu

Nesta parte, você moverá a seção de menu *Meu endereço* para a seção *Monitoramento* e, em seguida, adicionará um menu aninhado a ela. Como resultado, os usuários poderão acessar duas páginas de submenus a partir da seção de menu *Monitoramento* → *Meu endereço*.

1. Abra e edite o arquivo *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

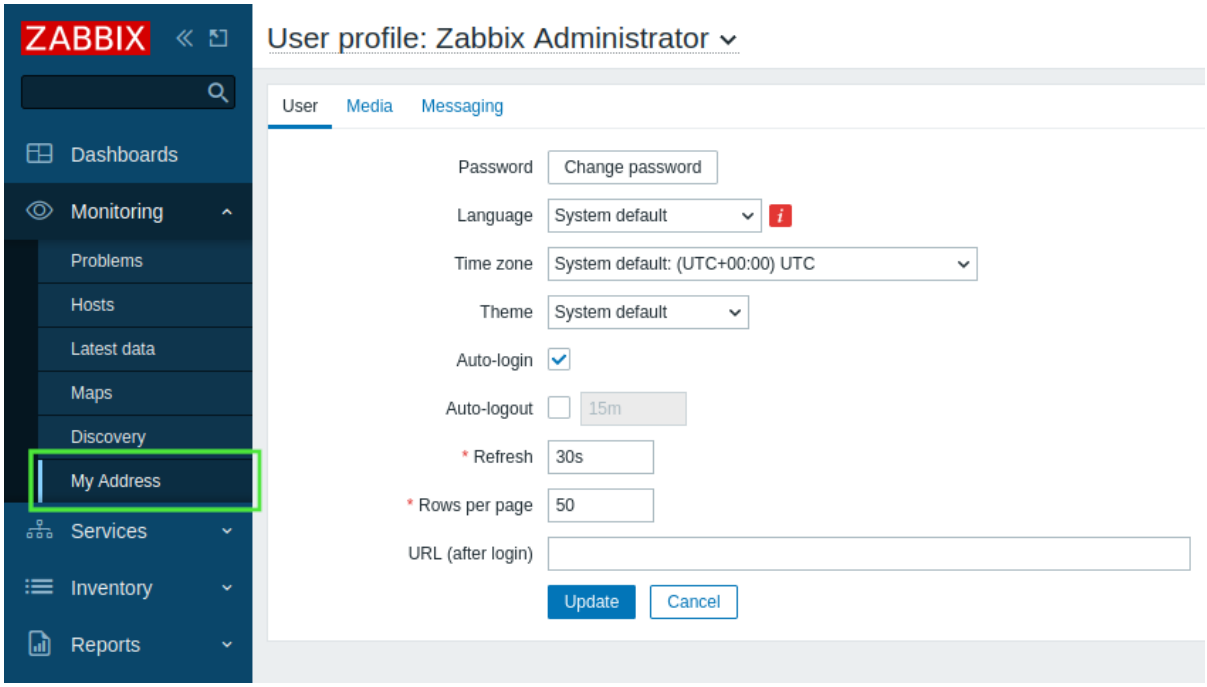
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Atualize o frontend do Zabbix. Expanda a seção de menu *Monitoramento* e observe que a seção *Meu endereço* agora está localizada abaixo da seção *Descoberta*.



3. Para adicionar páginas aninhadas à seção de menu *Meu endereço*, abra e edite novamente o arquivo *Module.php*.

Esta etapa cria duas subseções:

- *IP externo* que executa uma nova ação 'my.address' que será definida nas próximas etapas;
- *Perfil do usuário* que executa a ação predefinida 'userprofile.edit' para abrir a página *Perfil do usuário*.

Observe que, para o menu aninhado, você precisa usar a classe *CMenu* além das classes usadas nas etapas anteriores.

ui/modules/MyAddress/Module.php

```
<?php

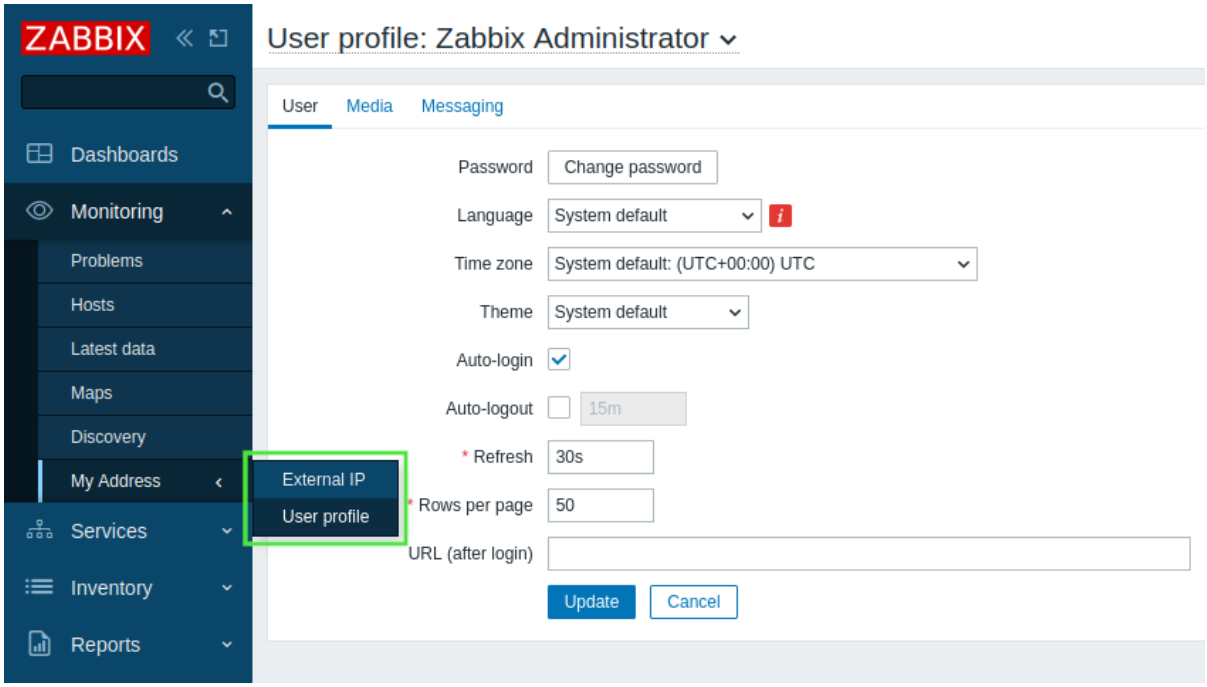
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Atualize o frontend do Zabbix. Observe que a seção de menu *Meu endereço* agora contém um menu de terceiro nível com duas páginas - *IP externo* e *Perfil do usuário*.



Parte III - Ação do módulo

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*. Esta classe de ação implementará quatro funções: *init()*, *checkInput()*, *checkPermissions()* e *doAction()*. O frontend do Zabbix chama a função *doAction()* quando a ação é solicitada. Esta função é responsável pela lógica de negócio do módulo.

Attention:
Os dados devem ser organizados como um array associativo. O array pode ser multidimensional e pode conter quaisquer dados esperados pela visualização.

ui/modules/MyAddress/actions/MyAddress.php

```
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
```

```
        $this->setResponse($response);
    }
}
```

3. 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 ser definida especificamente. O framework passa automaticamente o array associativo para a visualização.

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

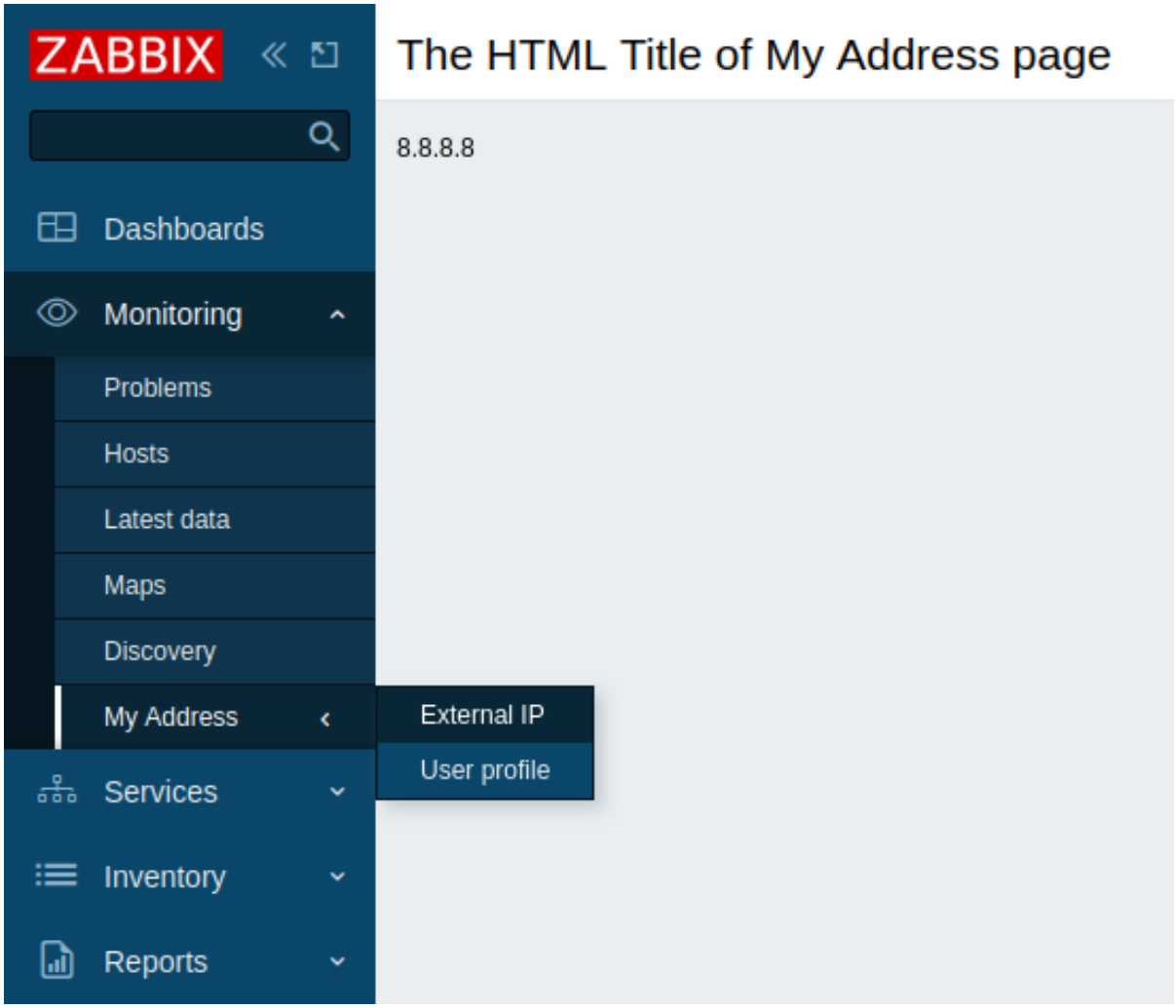
5. A ação do módulo deve ser registrada no arquivo *manifest.json*. Abra o *manifest.json* e adicione um novo objeto `actions` que contém:

- a chave da ação com o nome da ação escrito em minúsculas (a-z) e com palavras separadas por pontos (por exemplo, `my.address`);
- o nome da classe da ação (`MyAddress`) como valor para a chave `class` do objeto `my.address`;
- o nome da visualização da ação (`my.address`) como valor para a chave `view` do objeto `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

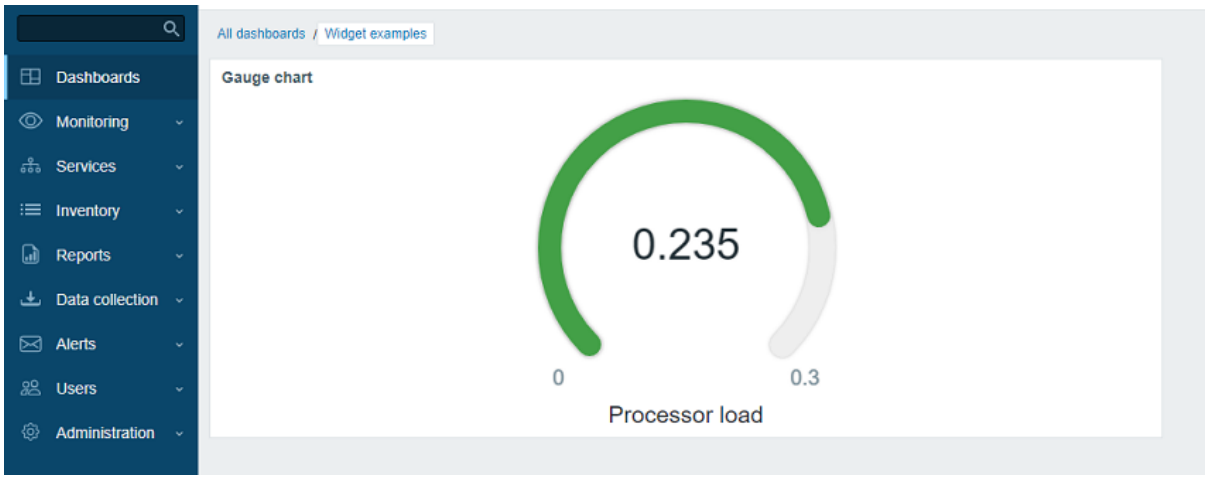
6. Atualize o frontend do Zabbix. Clique em *My address* → *External IP* para ver o endereço IP do seu computador.



Criar um widget (tutorial)

Este é 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.

O que você irá construir Durante este tutorial, você irá primeiro construir um widget básico "Hello, world!" e depois convertê-lo em um widget mais avançado que exibe o valor de um item como um gráfico de gauge. Veja como o widget finalizado ficará:



Parte I - "Olá, mundo!" Nesta seção, você aprenderá como criar os elementos mínimos necessários de um 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"
}
```

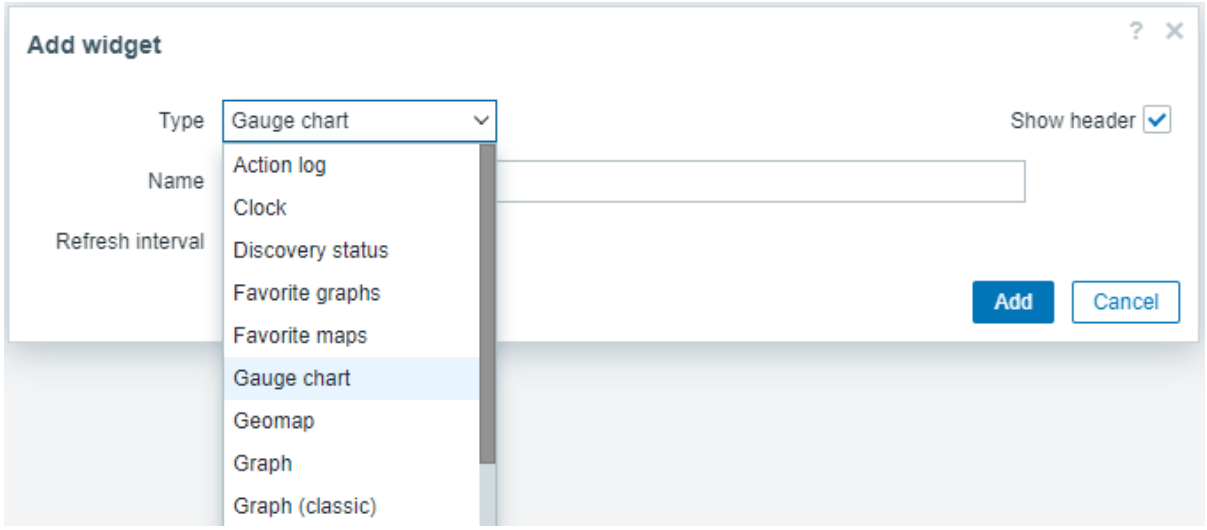
3. No frontend do Zabbix, vá para a seção *Administração* → *Geral* → *Módulos* e clique no botão *Procurar diretório*.



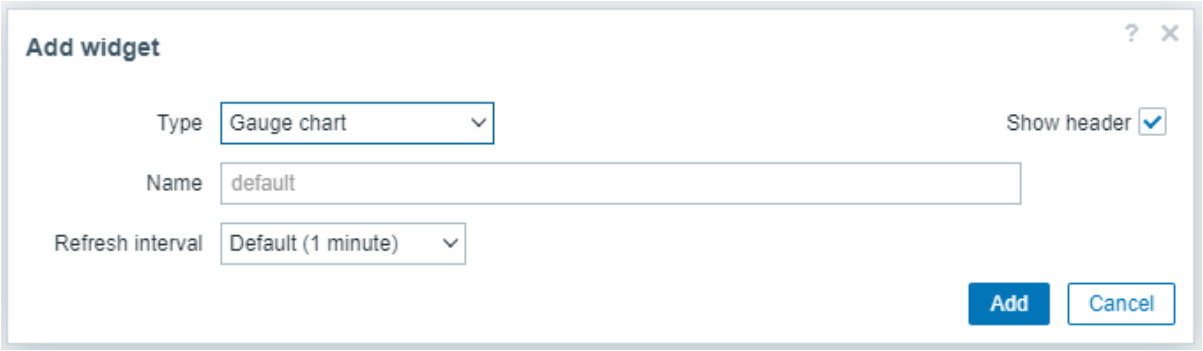
4. Localize o novo módulo *Gauge chart* na lista e clique no hiperlink "Desabilitado" para alterar o status do módulo de "Desabilitado" para "Habilitado" (se o módulo não estiver listado, consulte a seção de solução de problemas).



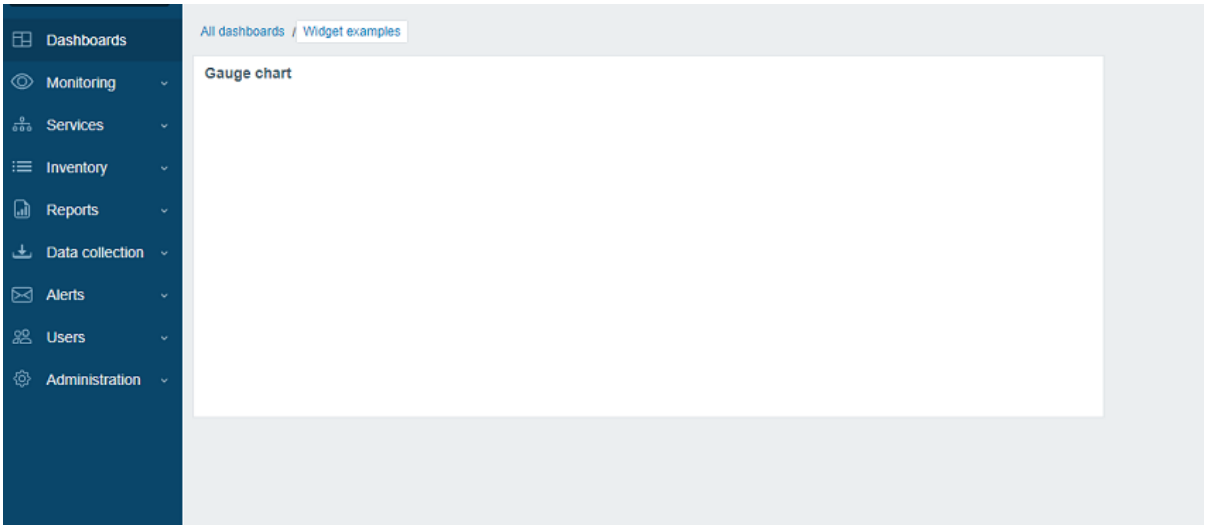
5. Abra um dashboard, alterne para o modo de edição e adicione um novo widget. No campo *Tipo*, selecione "Gauge chart".



6. Neste ponto, a configuração do widget *Gauge chart* contém apenas os campos comuns *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 de **visualização** 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*.

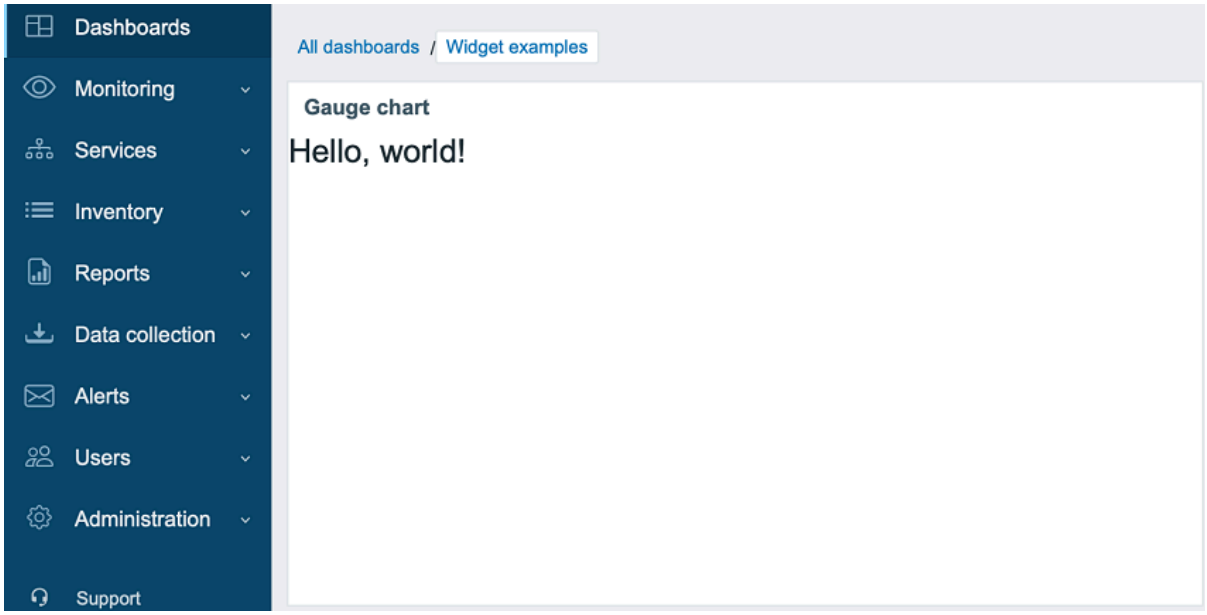
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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 Gauge 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 exibir 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 formato único 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 do widget *widget.edit.php* e adicione uma visualização para o novo campo *Description*. Para a classe de campo *CWidgetFieldTextBox*, a visualização é *CWidgetFieldTextBoxView*.

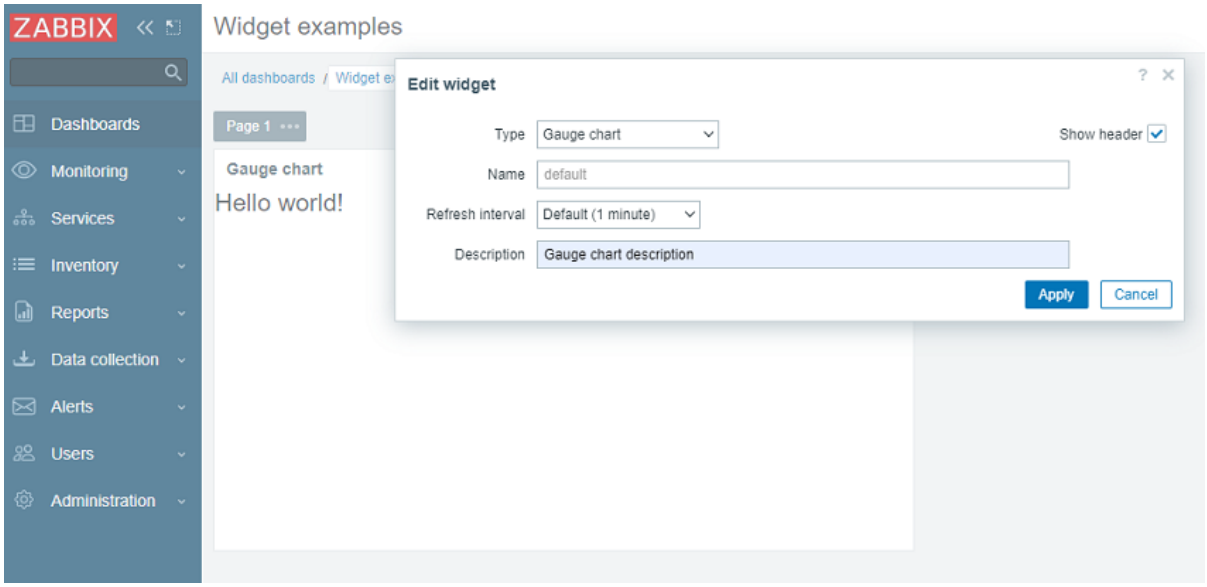
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?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 *Aplicar* no formulário de configuração do widget. Em seguida, clique em *Salvar alterações* 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 description, contido em *$data['description']*, na visualização do widget. Abra *views/widget.view.php* e substitua o texto estático "Hello, world!" por *$data['description']*.

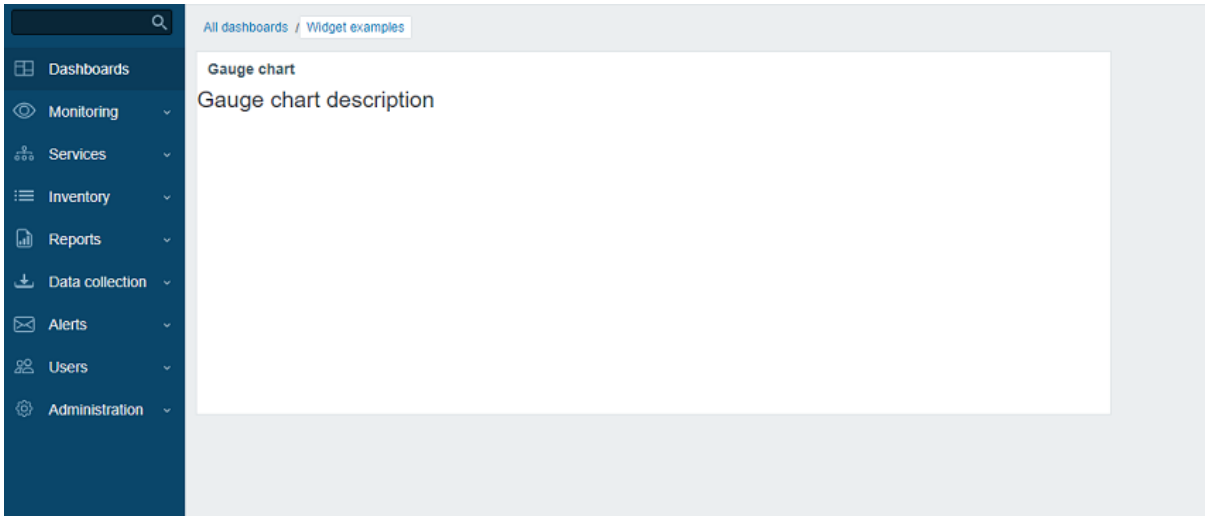
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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.

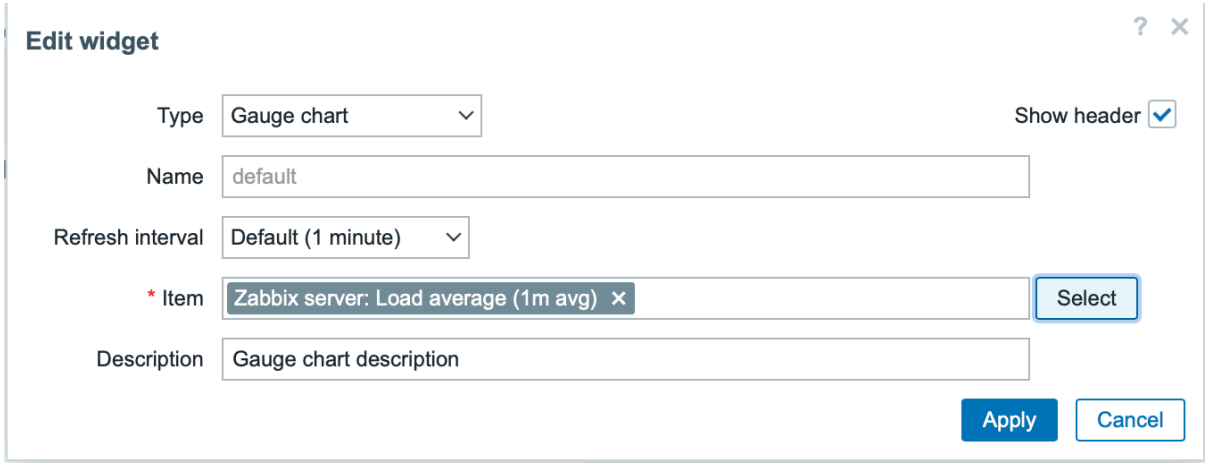
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?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
<?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.

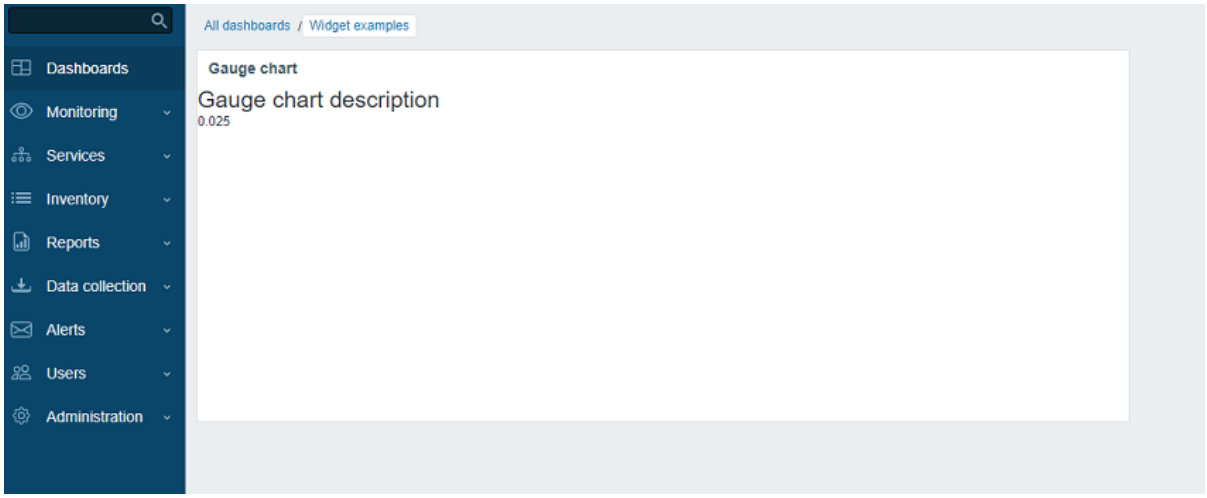
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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 avançadas à visualização de configuração

Nesta seção, você aprenderá como adicionar uma seção expansível/recolhível *Configuração avançada* com parâmetros opcionais, como cor, valores mínimo e máximo, 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* irá estender a classe base *CWidget* para adicionar/sobrescrever as configurações padrão do widget (neste caso - traduções). O JavaScript, fornecido abaixo, exibe a string ”Sem dados” em caso de ausência de dados. A string ”Sem dados” está presente nos arquivos de tradução da interface do Zabbix.

Se houver quaisquer constantes do widget, recomenda-se também especificá-las na classe *Widget*.

ui/modules/lesson_gauge_chart/Widget.php

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. Abra *includes/WidgetForm.php* e adicione os novos campos *Cor* (seletor de cor), *Mín* (campo numérico), *Máx* (campo numérico) e *Unidades* (select), e defina a paleta de cores padrão para o seletor de cor, para que possa ser usada nos próximos passos.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};
```

```
use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Formulário do widget de gráfico de gauge.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Abra *views/widget.edit.php* e adicione os componentes visuais dos campos à visualização de configuração.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Visualização do formulário do widget de gráfico de gauge.
 *
 * @var CView $this
```

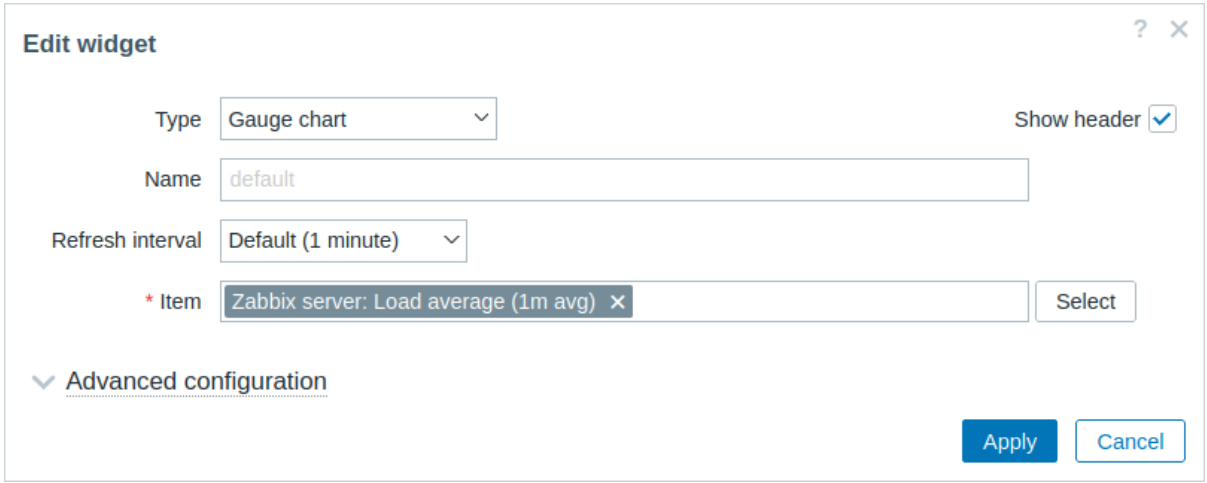
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

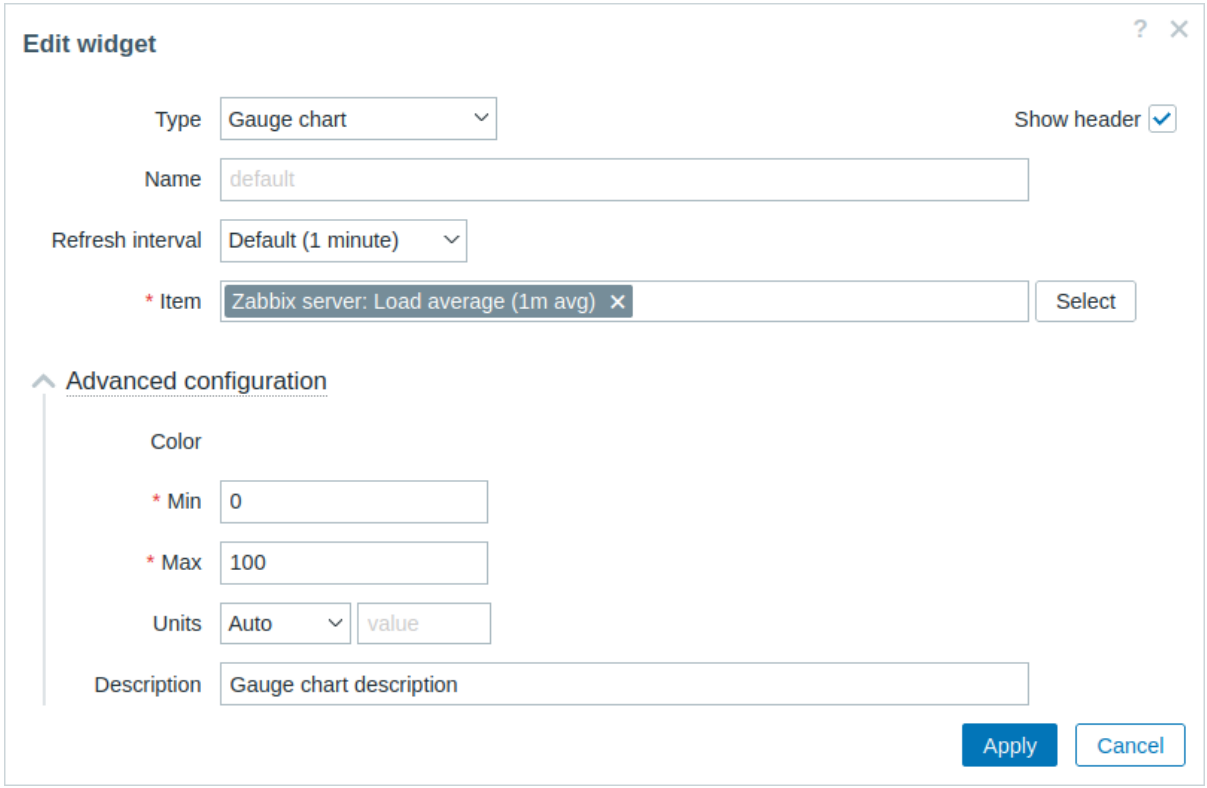
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
O método *addField()* da classe *CWidgetFormView* recebe uma string de classe CSS como segundo parâmetro.

4. Volte ao dashboard, alterne para o modo de edição e clique no ícone de engrenagem no widget para abrir o formulário de configuração do widget. O formulário de configuração do widget agora contém uma nova seção expansível/recolhível *Configuração avançada*.



5. Expanda a seção *Configuração avançada* para ver os campos adicionais de configuração do widget. Observe que o campo *Cor* ainda não possui um seletor de cor. Isso ocorre porque o seletor de cor deve ser inicializado com JavaScript, que será adicionado na próxima seção - *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

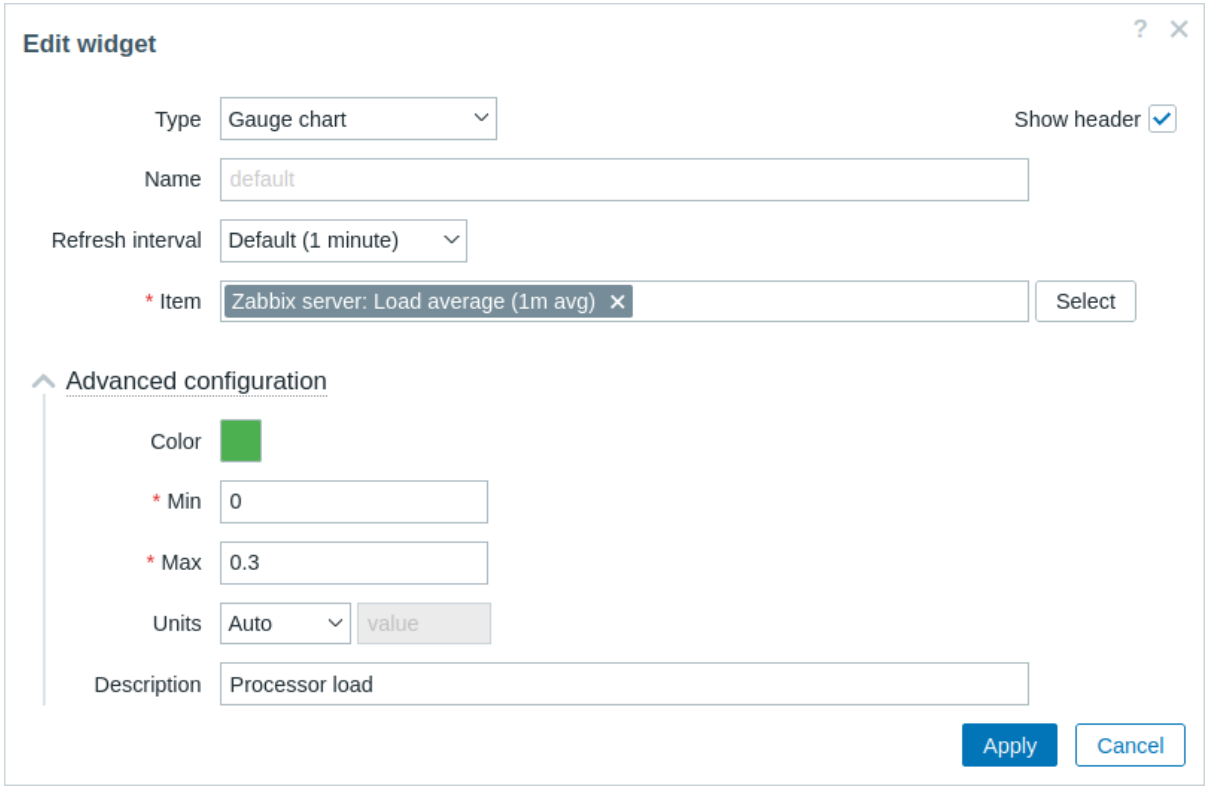
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

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 *Aplicar* no formulário de configuração do widget. Em seguida, clique em *Salvar alterações* 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 o envio da configuração e do 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 através 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 personalizada 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 como adicionar estilos CSS personalizados para tornar o widget mais atraente.

1. Para os estilos do widget, crie um novo diretório *css* no diretório *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;
}
```

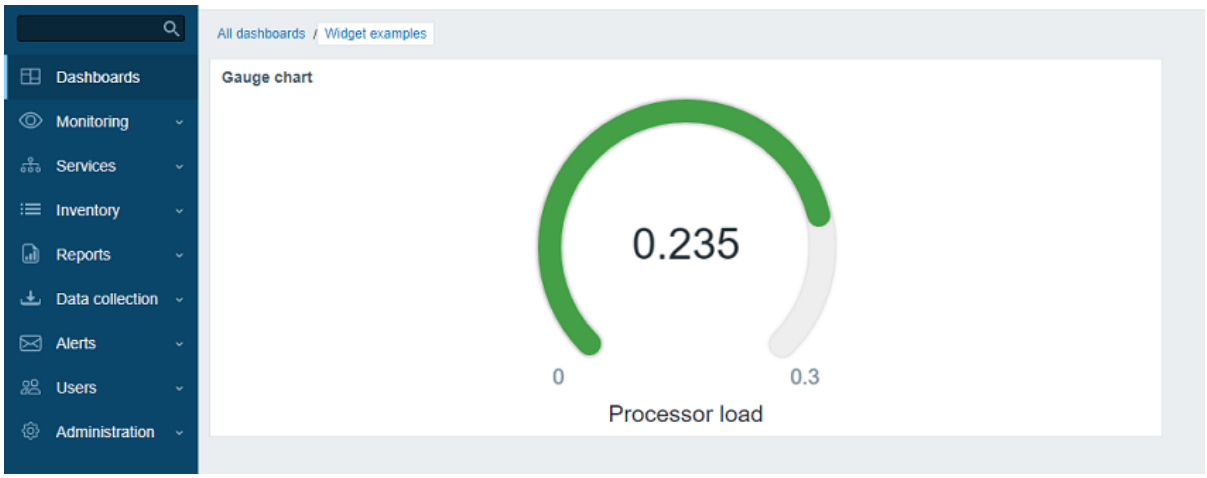
3. Abra o *manifest.json* e adicione o nome do arquivo CSS (*widget.css*) ao array na seção *assets/css*. Isso permitirá que os estilos CSS definidos em *widget.css* sejam carregados junto com a página do dashboard.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
```

```
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Atualize a página do dashboard para ver a versão finalizada do widget.



Exemplos

Esta seção fornece arquivos de módulos e widgets de exemplo, que você pode usar como base para seus módulos personalizados.

Para usar um módulo:

1. Baixe o arquivo ZIP.
2. Descompacte o conteúdo em um diretório separado dentro do diretório *modules* da sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).
3. Registre o módulo no frontend 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 apenas CSS - hello_world_css.zip
- Widget ”Hello, world” usando apenas JavaScript - hello_world_js.zip
- Widget ”Hello, world” usando PHP - hello_world_php.zip

Note:
Você também pode usar widgets nativos do Zabbix como exemplos.

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:

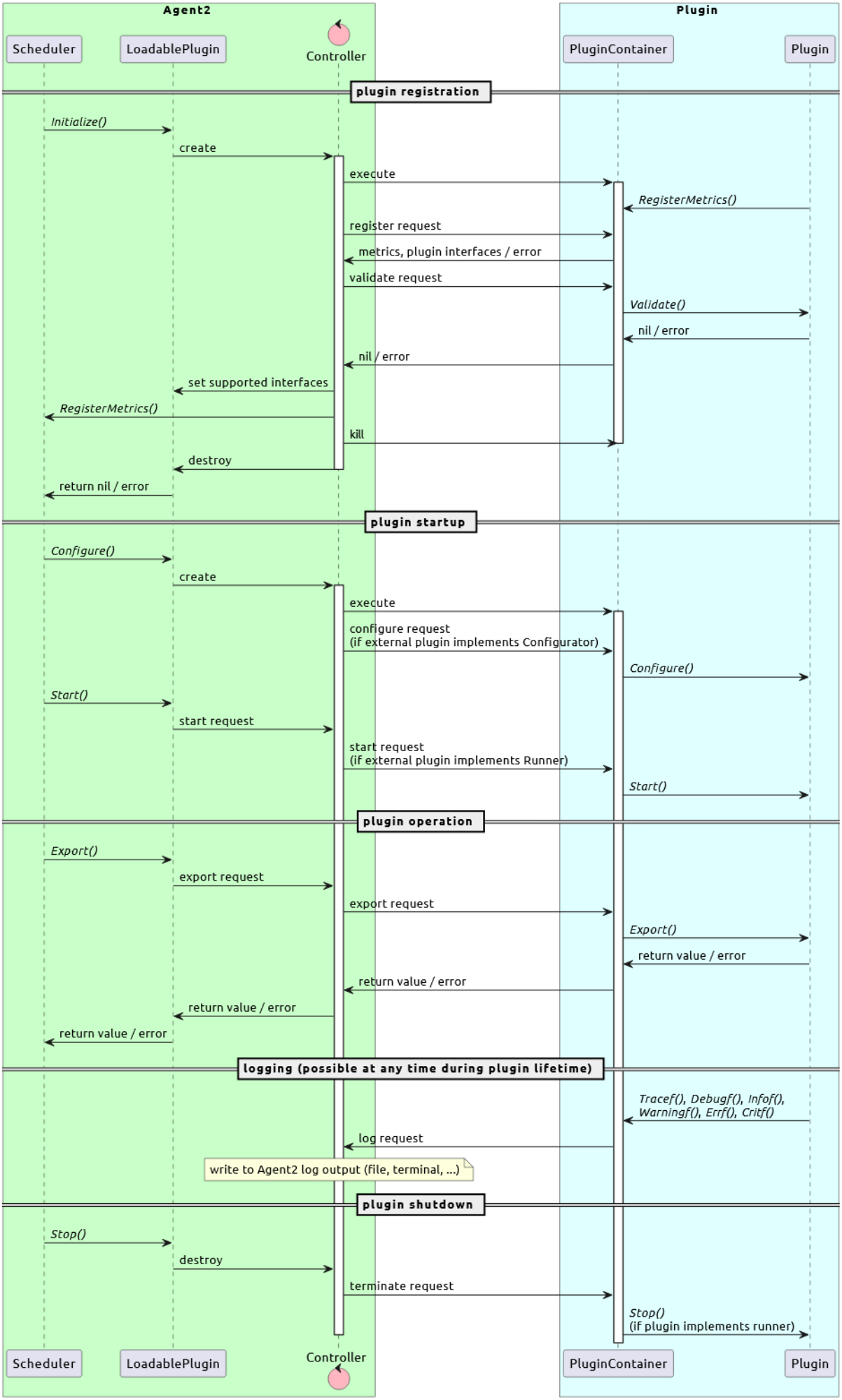
- Escreva seu primeiro plugin
- Interfaces de plugin

Veja também:

- Plugin de exemplo 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 plugins carregáveis existentes fornecidos pelo Zabbix como referência:

- Plugin de exemplo para o Zabbix agent 2
- Plugin MSSQl
- Plugin MongoDB
- Plugin PostgreSQL

Criar um plugin (tutorial)

Este é um tutorial passo a passo sobre como criar um plugin simples carregável para o Zabbix agent 2.

Você também pode usar o repositório de exemplo como um modelo ou guia para criar seus próprios plugins.

O que você irá criar Este tutorial demonstra como criar um novo plugin carregável **MyIP**. O plugin irá implementar uma única chave de item, **myip**, que retorna o endereço IP externo do host onde o Zabbix agent 2 está em execução.

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 path/to/plugins/myip # Acesse o diretório do seu plugin
go mod init myip
```

2. Instale a dependência obrigatória Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Substitua `$LATEST_COMMIT_HASH` pelo hash do commit `HEAD` mais recente do repositório `golang.zabbix.com/sdk` no branch de release apropriado. Por exemplo:

```
go get golang.zabbix.com/sdk@af85407
```

Observe que a versão do `golang.zabbix.com/sdk` atualmente não é suportada, mas isso pode mudar no futuro.

Dependências adicionais podem ser instaladas conforme necessário usando `go get`.

3. Crie um arquivo vazio `main.go` 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 os plugins tenham uma estrutura consistente.

1. Configure o fluxo básico de execução.

Comece definindo o fluxo principal de execução do plugin. Adicione o seguinte código ao `main.go`:

```
package main

func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
return nil
}
```

Isso estabelece o fluxo básico de execução para o plugin. A função run conterá posteriormente a lógica principal do plugin.

2. Explore as interfaces do plugin.

Um plugin do Zabbix agent 2 deve ser representado por uma struct que implementa interfaces do pacote `golang.zabbix.com/sdk/plugin`:

- *Accessor* - define métodos essenciais que todos os plugins devem implementar, como definir o nome do plugin e lidar com timeouts de chave de item.
- Uma ou mais das seguintes interfaces funcionais de plugin:
- *Exporter* - executa uma pesquisa e retorna um valor (ou valores), nada ou um erro; frequentemente usado junto com a interface *Collector*.
- *Collector* - gerencia a coleta periódica de dados.
- *Runner* - define procedimentos de inicialização e desligamento do plugin.
- *Watcher* - permite implementar polling independente de métricas, ignorando o agendador interno do agent; útil para monitoramento baseado em trap ou orientado a eventos.
- *Configurator* - define como o plugin lê e aplica suas configurações.

Você pode implementar essas interfaces por conta própria ou usar as implementações padrão fornecidas pelo Zabbix Go SDK, modificando-as conforme necessário. Este tutorial utiliza as implementações padrão.

3. Crie a struct do plugin.

Agora, importe o pacote *plugin* e crie uma struct `myIP` que incorpora a struct `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
plugin.Base
}
```

A struct myIP atualmente satisfaz a interface Accessor. Um método para implementar uma das interfaces funcionais do plugin, o `Exporter`, será adicionado posteriormente no tutorial.

Etapa 3: Definir as chaves de item Seu plugin precisa das chaves de item para coletar dados e fornecê-los ao Zabbix server ou proxy.

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{}

// Registra a chave de item `myip`.
err := plugin.RegisterMetrics(
p,
"MyIP",                          // Nome do plugin
"myip",                          // Nome da chave de item
"Retorna o endereço IP do host.", // Descrição da chave de 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 handler O handler facilita a comunicação entre o agent e o plugin.

1. Importe o pacote *container*:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Dentro da função `run()`, adicione o código para criar e configurar um handler:

```
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
"Returns the host's IP address.", // Descrição da chave do item
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// Crie um novo handler.
h, err := container.NewHandler("MyIP") // Nome do plugin
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// Configure o log para encaminhar logs do plugin para o agent.
// Disponível via p.Logger.Infof, p.Logger.Debugf, etc.
p.Logger = h

// Inicie a execução do plugin.
// Bloqueia até que uma solicitação de término seja recebida do agent.
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

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,             // O item key a ser coletado.
  params []string,        // Argumentos passados para o item key (`myip[arg1, arg2]`).
  context ContextProvider // Metadados sobre a coleta de dados do item key.
) (any, error)
```

1. Importe os pacotes necessários para requisições HTTP e leitura de resposta:

```
import (
"io"
"net/http"
)
```

2. Implemente o método Export para a struct 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 o `key` fornecido é suportado.
if key != "myip" {
return nil, errs.Errorf("unknown item key %q", key)
}

// O log será encaminhado para o log do agent 2.
p.Infof(
"received request to handle %q key with %d parameters",
```

```
key,
len(params),
)

// Colete os dados e retorne-os.

resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "failed to get IP address")
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

Etapa 6: Compile e configure 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 Zabbix agent 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 (*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 esta:

```
myip                                          [s|192.0.2.0]
```

Com isso, você criou um plugin simples carregável para o Zabbix agent 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
"Returns the host's IP address.", // Descrição da chave do item
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// Cria um novo handler.
h, err := container.NewHandler("MyIP") // Nome do plugin
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// Configura o log para encaminhar logs do plugin para o agent.
// Disponível via p.Logger.Infof, p.Logger.Debugf, etc.
p.Logger = h

// Inicia a execução do plugin.
// Bloqueia até receber uma solicitação de término do agent.
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

return nil
}

func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// O plugin pode usar diferentes lógicas de coleta de dados com base no parâmetro `key`.
    // Esta implementação apenas verifica 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 agent 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, "failed to get IP address")
```

```
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

Código-fonte completo

Interfaces de plugin

Esta seção descreve as interfaces de plugin disponíveis.

plugin.Exporter *Exporter* é a interface mais simples que executa uma coleta e retorna um valor (valores), nada ou erro. Ela aceita uma chave de item pré-analisada, parâmetros e contexto. O acesso a todas as outras interfaces de plugin é exclusivo e nenhum método pode ser chamado se um plugin já estiver executando uma tarefa. Além disso, há um limite de 100 chamadas *Export()* concorrentes por plugin, que pode ser reduzido de acordo com os requisitos de cada plugin.

plugin.Configurator A interface *Configurator* fornece parâmetros de configuração do plugin a partir dos arquivos de configuração do Zabbix agent 2.

plugin.Runner A interface *Runner* fornece os meios para realizar a inicialização quando um plugin é iniciado (ativado) e a desinicialização quando um plugin é interrompido (desativado). Por exemplo, um plugin pode iniciar/parar alguma *goroutine* em segundo plano implementando a interface Runner.

Alterações no desenvolvimento de extensões

Esta página lista todas as alterações (se houver) no desenvolvimento de extensões personalizadas do Zabbix.

Páginas de manual do Zabbix

Estas são as páginas de manual do Zabbix para os processos do Zabbix.

zabbix_agent2

Seção: Comandos de Manutenção (8)
Atualizado: 2019-01-29
Índice Retornar ao Conteúdo Principal

NOME

zabbix_agent2 - Zabbix agent 2

SINOPSE

zabbix_agent2 [**-c** *arquivo-config*]
zabbix_agent2 [**-c** *arquivo-config*] [**-v**] **-p**
zabbix_agent2 [**-c** *arquivo-config*] [**-v**] **-t** *item-key*
zabbix_agent2 [**-c** *arquivo-config*] **-R** *opcao-runtime*
zabbix_agent2 [**-c** *arquivo-config*] **-T**
zabbix_agent2 -h
zabbix_agent2 -V

DESCRIÇÃO

zabbix_agent2 é um aplicativo para monitorar parâmetros de vários serviços.

OPÇÕES

-c, **--config** *config-file*
Use o *config-file* alternativo em vez do padrão.

-R, **--runtime-control** *runtime-option*
Execute funções administrativas de acordo com *runtime-option*.

Opções de controle em tempo de execução: userparameter_reload
Recarregar parâmetros de usuário do arquivo de configuração

log_level_increase
Aumentar o nível de log

log_level_decrease
Diminuir o nível de log

help
Listar opções de controle em tempo de execução disponíveis

metrics
Listar métricas disponíveis

version
Exibir versão

-p, **--print**
Imprimir os items conhecidos e sair. Para cada item, são usados valores padrão genéricos ou valores padrão específicos para teste. Esses valores padrão são listados entre colchetes como parâmetros da chave do item. Os valores retornados são incluídos entre colchetes e precedidos pelo tipo do valor retornado, separados por um caractere pipe. Para parâmetros de usuário, o tipo é sempre **t**, pois o agent não pode determinar todos os possíveis valores de retorno. Os items exibidos como funcionando não têm garantia de funcionar a partir do Zabbix server ou do zabbix_get ao consultar um agent em execução, pois as permissões ou o ambiente podem ser diferentes. Os tipos de valores retornados são:

d
Número com parte decimal.

m
Não suportado. Isso pode ser causado por consultar um item que só funciona no modo ativo, como um item de monitoramento de log ou um item que requer múltiplos 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 sem sinal.

-t, **--test** *item-key*
Testar um único item e sair. Veja **--print** para descrição da saída.

-v, **--verbose**
Habilitar saída detalhada ao testar item ou imprimir items conhecidos.

-T, **--test-config**
Validar o arquivo de configuração e sair.

-h, **--help**
Exibir esta ajuda e sair.

-V, **--version**
Exibir informações da versão e sair.

ARQUIVOS

/usr/local/etc/zabbix_agent2.conf
Localização padrão do arquivo de configuração do Zabbix agent 2 (se não for modificado durante a compilação).

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

Este documento foi criado em: 14:07:57 GMT, 22 de novembro de 2021

zabbix_agentd

Seção: Comandos de Manutenção (8)
Atualizado: 2019-01-29
Índice Retornar ao Conteúdo Principal

NOME

zabbix_agentd - Daemon do agent do Zabbix

SINOPSE

zabbix_agentd [**-c** *arquivo-config*]
zabbix_agentd [**-c** *arquivo-config*] **-p**
zabbix_agentd [**-c** *arquivo-config*] **-t** *item-key*
zabbix_agentd [**-c** *arquivo-config*] **-R** *opcao-runtime*
zabbix_agentd [**-c** *arquivo-config*] **-T**

zabbix_agentd -h
zabbix_agentd -V

DESCRIÇÃO

zabbix_agentd é um daemon para monitorar vários parâmetros do servidor.

OPÇÕES

-c, **--config** *config-file*
Use o *config-file* alternativo em vez do padrão.

-f, **--foreground**
Execute o agent Zabbix em primeiro plano.

-R, **--runtime-control** *runtime-option*
Execute funções administrativas de acordo com *runtime-option*.

Opções de controle em tempo de execução userparameter_reload
Recarregar os parâmetros do usuário a partir do arquivo de configuração

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 (verificações ativas, coletor, listener)

process-type,N
Tipo de processo e número (por exemplo, listener,3)

pid
Identificador do processo, até 65535. Para valores maiores, especifique o alvo como ”process-type,N”

-p, **--print**
Imprime os items conhecidos e sai. Para cada item, são usados os padrões genéricos ou padrões específicos para teste. Esses padrões são listados entre colchetes como parâmetros da chave do item. Os valores retornados são colocados entre colchetes e precedidos pelo tipo do valor retornado, separados por um caractere pipe. Para parâmetros de usuário, o tipo é sempre **t**, pois o agent não pode determinar todos os possíveis valores de retorno. Items exibidos como funcionando não são garantidos de funcionar a partir do Zabbix server ou do zabbix_get ao consultar um agent em execução, pois as permissões ou o ambiente podem ser diferentes. Os tipos de valores retornados são:

d
Número com parte decimal.

m
Não suportado. Isso pode ser causado por consultar um item que só funciona no modo ativo, como um item de monitoramento de log ou um item que requer múltiplos 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 sem sinal.

-t, **--test** *item-key*
Testa um único item e sai. Veja **--print** para descrição da saída.

-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_agentd.conf
Localização padrão do arquivo de configuração do agent Zabbix (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

Este documento foi criado em: 20:50:13 GMT, 22 de novembro de 2021

zabbix_get

Seção: Comandos do Usuário (1)
Atualizado: 2022-01-06
Índice Retornar ao Conteúdo Principal

NOME

zabbix_get - utilitário de obtenção do Zabbix

SINOPSE

zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

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 conecta usando o protocolo JSON, faz fallback e tenta novamente com o protocolo plaintext (padrão)

json conecta usando o protocolo JSON

plaintext conecta usando o protocolo plaintext, 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*
Assunto 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 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 de ciphersuite. Esta opção não está disponível se a versão do OpenSSL for menor que 1.1.1.

--tls-cipher *cipher-string*
String de prioridade GnuTLS (para TLS 1.2 e superior) ou string de cifra OpenSSL (apenas para TLS 1.2). Substitui os critérios padrão de seleção de 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=IT operations,O=Example Corp,DC=example,DC=com” --tls-agent-cert-subject ”CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com” --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

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: 2022-01-06
Índice Retornar ao Conteúdo Principal

NOME

zabbix_js - Utilitário JS do Zabbix

SINOPSE

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

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 script-file.js -p example

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

Este documento foi criado em: 21:23:35 GMT, 18 de março de 2020

zabbix_proxy

Seção: Comandos de Manutenção (8)
Atualizado: 2020-09-04
Índice Retornar ao Sumário

NOME

zabbix_proxy - Daemon do proxy do Zabbix

SINOPSE

zabbix_proxy [**-c** *arquivo-config*]
zabbix_proxy [**-c** *arquivo-config*] **-R** *opcao-runtime*
zabbix_proxy [**-c** *arquivo-config*] **-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 Zabbix server.

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 *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 Zabbix ativo irá se conectar ao server Zabbix e solicitar 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. Observe que o Zabbix armazena em cache os mapeamentos SNMPv3 EngineID→IP e pode reutilizar EngineIDs automaticamente para reduzir a sobrecarga de polling.

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
Localização padrão do arquivo de configuração do Zabbix proxy (se não for modificado durante a 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

Este documento foi criado em: 16:12:22 GMT, 04 de setembro de 2020

zabbix_sender

Seção: Comandos do Usuário (1)
Atualizado: 2021-06-01
Índice Retornar ao Conteúdo Principal

NOME

zabbix_sender - Utilitário de envio do Zabbix

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 uma ferramenta de linha de comando para enviar dados de monitoramento para o Zabbix server ou proxy. No Zabbix server, um item do tipo **Zabbix trapper** deve ser criado com a chave correspondente. Observe que os valores recebidos só serão aceitos de hosts especificados no campo **Allowed hosts** para este item.

OPÇÕES

-c, **--config** *config-file*
Usa *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 enviar dados. Se o envio de dados em lote falhar para um endereço, os lotes seguintes não são enviados para esse endereço.

-z, **--zabbix-server** *server*
Nome do host ou endereço IP do Zabbix server. Se um host for monitorado por um proxy, o hostname 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. Intervalo válido: 1-300 segundos (padrão: 60)

-s, **--host** *host*
Especifica o nome do host ao qual o item pertence (conforme registrado no frontend do Zabbix). O endereço IP do host e o 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**.

Um exemplo de 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 citação.
- Entradas com espaço devem ser citadas.
- Aspas duplas e barras invertidas dentro de uma entrada citada devem ser escapadas com uma barra invertida.
- Escape não é suportado em entradas sem aspas.
- Sequências de escape de quebra de linha (\n) são suportadas em strings citadas.
- 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 de timestamp Unix. 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.

Um exemplo de 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>**.

Um exemplo de 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 (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-server-cert-issuer *cert-issuer*
Emissor de certificado de server permitido.

--tls-server-cert-subject *cert-subject*
Assunto de certificado de server 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 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 de ciphersuite. Esta opção não está disponível se a versão do OpenSSL for menor que 1.1.1.

--tls-cipher *cipher-string*
String de prioridade GnuTLS (para TLS 1.2 e superior) ou string de cifra OpenSSL (apenas para TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite.

-v, **--verbose**
Modo detalhado, **-vv** para mais detalhes.

-h, **--help**
Exibe esta ajuda e sai.

-V, **--version**
Exibe informações da versão e sai.

STATUS DE SAÍDA

O status de saída é 0 se os valores foram enviados e todos foram processados com sucesso pelo server. Se os dados foram enviados, mas o processamento de pelo menos um dos valores falhou, o status de saída é 2. Se o envio dos dados falhar, o status de saída é 1.

EXEMPLOS

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

Envia **342.45** como valor para o item **mysql.queries** do host monitorado. Usa o host monitorado e o Zabbix server definidos no arquivo de configuração do agent.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Monitored Host” -k mysql.queries -o 342.45

Envia **342.45** como valor para o item **mysql.queries** do host **Monitored Host** usando o Zabbix server definido no arquivo de configuração do agent.

zabbix_sender -z 192.168.1.113 -i data_values.txt

Envia valores do arquivo **data_values.txt** para o Zabbix server com o IP **192.168.1.113**. Os nomes dos hosts e as keys são definidos no arquivo.

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

Envia um valor com timestamp da linha de comando para o Zabbix server, especificado no arquivo de configuração do agent. O traço nos dados de entrada indica que o hostname também deve ser usado do mesmo arquivo de configuração.

echo ’”Zabbix server” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Envia um valor vazio de um item para o Zabbix server com endereço IP **192.168.1.113** na porta **10000** a partir da linha de comando. Valores vazios devem ser indicados 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

Envia **342.45** como valor para o item **mysql.queries** no host **Monitored Host** para o server 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

Envia **342.45** como valor para o item **mysql.queries** no host **Monitored Host** para o server 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

STATUS DE SAÍDA

EXEMPLOS

VEJA TAMBÉM

AUTOR

Este documento foi criado em: 08:42:39 GMT, 11 de junho de 2021

zabbix_server

Seção: Comandos de Manutenção (8)
Atualizado: 2020-09-04
Índice Retornar ao Conteúdo Principal

NOME

zabbix_server - Daemon do Zabbix server

SINOPSE

zabbix_server [**-c** *arquivo-config*]
zabbix_server [**-c** *arquivo-config*] **-R** *opcao-runtime*
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** *config-file*
Use o *config-file* alternativo em vez do padrão.

-f, **--foreground**
Execute o Zabbix server em primeiro plano.

-R, **--runtime-control** *runtime-option*
Execute funções administrativas de acordo com *runtime-option*.

-T, **--test-config**
Valide o arquivo de configuração e saia.

-h, **--help**
Exiba esta ajuda e saia.

-V, **--version**
Exiba informações da versão e saia.

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:

config_cache_reload
Recarregar 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 para o processo listado no arquivo PID.

snmp_cache_reload
Recarregar o cache SNMP, limpar as propriedades SNMP (engine time, engine boots, engine id, credenciais) para todos os hosts. Observe que o Zabbix armazena em cache os mapeamentos SNMPv3 EngineID→IP e pode reutilizar automaticamente os EngineIDs para reduzir a sobrecarga de polling.

housekeeper_execute
Executar o housekeeper. Ignorado se o housekeeper estiver sendo executado no momento.

trigger_housekeeper_execute
Executar o trigger housekeeper (remover problemas para triggers excluídos). Ignorado se o trigger housekeeper para serviços estiver sendo executado no momento.

diaginfo[=*section*]
Registrar informações internas de diagnóstico da seção especificada. A seção pode ser *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks*. Por padrão, as informações de diagnóstico de todas as seções são registradas.

ha_status
Registrar 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 os nós ativos/standby não podem ser removidos.

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*] Recarregar o cache de configuração do proxy.

secrets_reload
Recarregar segredos do Vault.

service_cache_reload
Recarregar o cache do gerenciador de serviços.

prof_enable[=*target*]
Habilitar 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.

prof_disable[=*target*]
Desabilitar profiling. Afeta todos os processos se o destino não for especificado.

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 (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 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 (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 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)

ARQUIVOS

/usr/local/etc/zabbix_server.conf
Local 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

Este documento foi criado em: 16:12:14 GMT, 04 de setembro de 2020

zabbix_web_service

Seção: Comandos de Manutenção (8)
Atualizado: 2019-01-29
Índice Retornar ao Conteúdo Principal

NOME

zabbix_web_service - Zabbix web service

SINOPSE

zabbix_web_service [**-c** *arquivo-config*]
zabbix_web_service [**-c** *arquivo-config*] **-T**
zabbix_web_service -h
zabbix_web_service -V

DESCRIÇÃO

zabbix_web_service é um aplicativo para fornecer serviços web para os componentes do Zabbix.

OPÇÕES

-c, **--config** *config-file*
Use o *config-file* alternativo em vez do padrão.

-T, **--test-config**
Valide o arquivo de configuração e saia.

-h, **--help**
Exibe esta ajuda e sai.

-V, **--version**
Exibe informações da versão e sai.

ARQUIVOS

/usr/local/etc/zabbix_web_service.conf
Localização padrão do arquivo de configuração do Zabbix web service (se não for 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)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

Este documento foi criado em: 12:58:30 GMT, 11 de junho de 2021