Documentation 8.0

ZABBIX

29.10.2026

Contents

Manual do usuário **5**

Bem-vindo à Documentação do Zabbix **5**

Primeiros passos com o Zabbix . . . 5
Zabbix Cloud . . . 5
Aplicativo móvel . . . 5
Central do desenvolvedor . . . 5
Comunidade e outros recursos . . . 5
Instalação e primeiros passos . . . 6
Instalação da interface web . . . 6
Guias de início rápido . . . 11
Instalação: guias adicionais . . . 85
Atualização . . . 166
Problemas conhecidos . . . 182
Requisitos . . . 191
Novidades e notas de atualização: 8.0 . . . 201
Templates . . . 206
Tipos de mídia . . . 208
Items . . . 209
Plugins . . . 209
Descoberta e registro automático . . . 210
Descoberta de baixo nível . . . 210
Bancos de dados . . . 210
Processos . . . 211
Segurança . . . 211
Autenticação . . . 212
Widgets . . . 212
Frontend . . . 213
Documentação . . . 215
Notas de atualização . . . 216
Alterações incompatíveis . . . 216
Outros . . . 217
Novidades e notas de atualização: 8.0.x . . . 219
Appliance do Zabbix . . . 220
Processos . . . 223
server . . . 225
proxy . . . 252
agent . . . 298
Agent 2 . . . 329
Java gateway . . . 392
Remetente . . . 397
Obter . . . 399
JS . . . 400
Serviço web . . . 401
Adaptador bridge . . . 404
MCP server . . . 408
Variáveis de ambiente . . . 415
Configuração . . . 417
Hosts e grupos de hosts . . . 428
Itens . . . 445
Detecção de problemas com triggers . . . 671

Visualização . . . 723
Templates e grupos de template . . . 748
Templates prontos para uso . . . 748
Notificações após eventos . . . 762
Macros . . . 812
Usuários e grupos de usuários . . . 828
Relatórios agendados . . . 839
Exportação de dados . . . 843
Exportação/importação de configuração . . . 852
Expressões regulares . . . 918
Eventos e resolução . . . 925
Reconhecimento e supressão de problemas . . . 927
Fechamento manual de problemas . . . 930
Períodos de manutenção . . . 932
Marcação . . . 938
Correlação de eventos . . . 941
Descoberta . . . 947
Descoberta de rede . . . 947
Autoregistro do agent ativo . . . 955
Descoberta de baixo nível . . . 958
Monitoramento de máquina virtual . . . 983
Chaves de item de monitoramento do VMware . . . 984
Campos de chave de descoberta de máquina virtual . . . 1002
Exemplos de JSON para itens VMware . . . 1006
Exemplo de configuração de monitoramento VMware . . . 1011
Monitoramento web . . . 1015
Itens de monitoramento web . . . 1025
Cenário da vida real . . . 1026
Monitoramento de serviços . . . 1033
Árvore de serviços . . . 1034
SLA . . . 1038
Exemplo de configuração . . . 1039
Atualização do monitoramento de serviços . . . 1044
Criptografia e segurança . . . 1044
Criptografia usando certificados . . . 1053
Criptografia usando chaves pré-compartilhadas . . . 1061
Conexão segura com o banco de dados . . . 1064
Conexão segura entre frontend e server . . . 1076
Controle de acesso . . . 1080
Criptografia . . . 1084
Protegendo o web server . . . 1085
Armazenamento de segredos . . . 1088
Confiança no certificado TLS do Google Chrome . . . 1097
Práticas recomendadas de segurança . . . 1098
Interface web . . . 1099
Menu . . . 1099
Seções do frontend . . . 1107
Configurações do usuário . . . 1369
Pesquisa global . . . 1374
Modo de manutenção do frontend . . . 1376
Parâmetros da página . . . 1377
Definições . . . 1380
Idiomas adicionais do frontend . . . 1380
Criando seu próprio tema . . . 1381
Modo de depuração . . . 1382
Cookies usados pelo Zabbix . . . 1383
Fusos horários . . . 1384
Redefinindo a senha . . . 1385
Período de tempo e seletores de host . . . 1386
Parâmetros de configuração: frontend do Zabbix . . . 1387
API . . . 1394
Referência de métodos . . . 1399
Apêndice 1. Comentário de referência . . . 2270

Apêndice 2. Alterações da 7.4 para a 8.0 . . . 2276
Apêndice 3. Alterações na 8.0 . . . 2279
Extensões . . . 2279
Módulos carregáveis . . . 2281
Módulos do frontend . . . 2290
Apêndices . . . 2291
Instalação e configuração . . . 2291
Itens . . . 2338
Funções suportadas . . . 2379
Macros . . . 2379
Compatibilidade de versão . . . 2417
Sintaxe do período de tempo . . . 2418
Símbolos de unidade . . . 2419
Execução de comandos . . . 2420
Exemplos de escape . . . 2421
O que é o Zabbix . . . 2423

Zabbix Cloud **2427**

Zabbix Cloud **2427**
Configurar e gerenciar o Zabbix Cloud . . . 2427
Saiba mais sobre o Zabbix Cloud . . . 2427
Implante o Zabbix na nuvem . . . 2428
Configuração do nó . . . 2437
Visão geral . . . 2438
Aba Visão geral . . . 2439
Guia de filtros de acesso . . . 2442
Aba Encryption . . . 2443
Aba Backups . . . 2443
Aba Histórico . . . 2444
Aba Maintenance . . . 2445
Aba Upgrade . . . 2445
Adicionando usuários . . . 2447
Introdução . . . 2447
Adicionando usuários às organizações . . . 2447
Adicionando usuários aos nodes . . . 2448
Principais diferenças do Zabbix Cloud . . . 2449
Introdução . . . 2449
Log de auditoria . . . 2451
Configurar um proxy do Zabbix Cloud . . . 2453

Aplicativo móvel **2461**

Aplicativo móvel **2461**
Configure o Zabbix para o aplicativo . . . 2461
Saiba mais sobre o aplicativo . . . 2461
Configure o Zabbix para o aplicativo . . . 2462
Visão geral do aplicativo . . . 2463
Acesso móvel seguro . . . 2471

Centro do desenvolvedor **2472**

Central do desenvolvedor **2472**
Desenvolva no Zabbix . . . 2472
Criando um plugin . . . 2472
Interfaces de plugin . . . 2478
Biblioteca Python para Zabbix . . . 2480
Instalação . . . 2480
Guia de início rápido . . . 2480
Usar a API do Zabbix . . . 2484
Coletar dados do agent Zabbix . . . 2487
Enviar dados para o Zabbix server ou proxy . . . 2488
Log de depuração . . . 2492
Módulos . . . 2493
Estrutura de arquivos do módulo . . . 2493

Widgets . . . 2501
Tutoriais . . . 2509
Exemplos . . . 2535
Alterações no desenvolvimento de extensões . . . 2535

Zabbix manpages **2536**
zabbix_agent2 . . . 2536
zabbix_agentd . . . 2537
zabbix_get . . . 2539
zabbix_js . . . 2541
zabbix_proxy . . . 2542
zabbix_sender . . . 2544
zabbix_server . . . 2549
zabbix_web_service . . . 2551

Aviso de direitos autorais **2552**

Manual do usuário

Bem-vindo à Documentação do Zabbix

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

O que há de novo no Zabbix 8.0

Primeiros passos com o Zabbix

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

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

Guias de início rápido Guias concisos e orientados a tarefas que o conduzem pelos conceitos básicos — desde os primeiros passos de configuração até o recebimento do seu primeiro alerta de problema.

Zabbix Cloud

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

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

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

Aplicativo móvel

Obtenha o aplicativo Instale o aplicativo móvel do Zabbix (em breve) em seu dispositivo e comece a monitorar seu ambiente Zabbix de qualquer lugar.

Configure o Zabbix para o aplicativo Instale o adaptador de ponte do Zabbix para permitir que dispositivos móveis se conectem ao Zabbix. Em seguida, adicione seu dispositivo e ative as notificações push.

Visão geral do aplicativo Aprenda os conceitos básicos de como usar o aplicativo móvel do Zabbix.

Central do desenvolvedor

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

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

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

Comunidade e outros recursos

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

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

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

Instalação e primeiros passos

A sequência a seguir descreve o processo de configuração do Zabbix:

1. Verifique se o software necessário está instalado e em execução.
2. Instale o Zabbix.
3. Instale a interface web do Zabbix.
4. Inicie o monitoramento.

Os detalhes de cada etapa são fornecidos abaixo.

Software necessário O Zabbix funciona bem com os seguintes softwares (frequentemente chamados de stack LAMP):

- **Sistema operacional Linux:** Ubuntu, Debian, CentOS ou outros
- **Web server:** Apache ou NGINX
- **Database server:** MySQL/MariaDB ou PostgreSQL
- **PHP**

Na maioria dos sistemas Linux modernos, o PHP pode já estar instalado. Nesse caso, talvez seja necessário apenas configurar o database server e o web server.

Para obter informações mais detalhadas, como as versões de software compatíveis, consulte Software necessário.

Instalar o Zabbix A maneira **mais fácil** de instalar o Zabbix é usando **pacotes da distribuição** (como alternativa, a partir de containers). Os pacotes instalam pequenas dependências de software.

Instruções precisas de instalação, adaptadas ao seu sistema operacional, estão disponíveis no **site do Zabbix**.

Selecione sua plataforma e siga as instruções fornecidas. Recomenda-se instalar um Zabbix server, um Zabbix agent 2 e o frontend (web interface) inicialmente. Você poderá adicionar proxies do Zabbix e mais agents posteriormente.

Parte do processo de instalação consiste em criar um database (para armazenar dados de configuração e histórico) e importar dados iniciais para ele.

Instalar a web interface do Zabbix Conclua as etapas do **assistente de instalação** da web interface.

Será solicitada a senha do database (criada na etapa anterior), para que o frontend possa se comunicar com o database.

Iniciar o monitoramento Faça login na web interface do Zabbix.

Use os **guias de início rápido** para começar a monitorar recursos populares (Linux, Apache, MySQL ou outros). Esses guias rápidos explicam com quais templates você deve trabalhar para começar a monitorar recursos populares.

Você pode começar pelo guia de início rápido para monitorar um server Linux.

Instalação da interface web

Esta seção fornece instruções para instalar a interface web do Zabbix. Siga estas etapas após instalar o backend do Zabbix.

O frontend do Zabbix é escrito em PHP e requer um web server compatível, como Apache ou NGINX, com o PHP configurado. Consulte os requisitos do frontend para obter informações sobre os softwares e as bibliotecas necessários.

Para usar um idioma diferente do inglês na interface web do Zabbix, a respectiva localidade deve estar instalada no web server, juntamente com a extensão PHP gettext. Observe que a alteração do idioma pode alterar o formato de hora/data.

Para configurar o SSL para o frontend do Zabbix, consulte Configuração do SSL para o frontend do Zabbix.

Assistente de instalação

Para instalar a interface web, abra a URL do frontend do Zabbix em um navegador:

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

Tela de boas-vindas

O assistente de instalação do frontend é aberto na tela de boas-vindas.

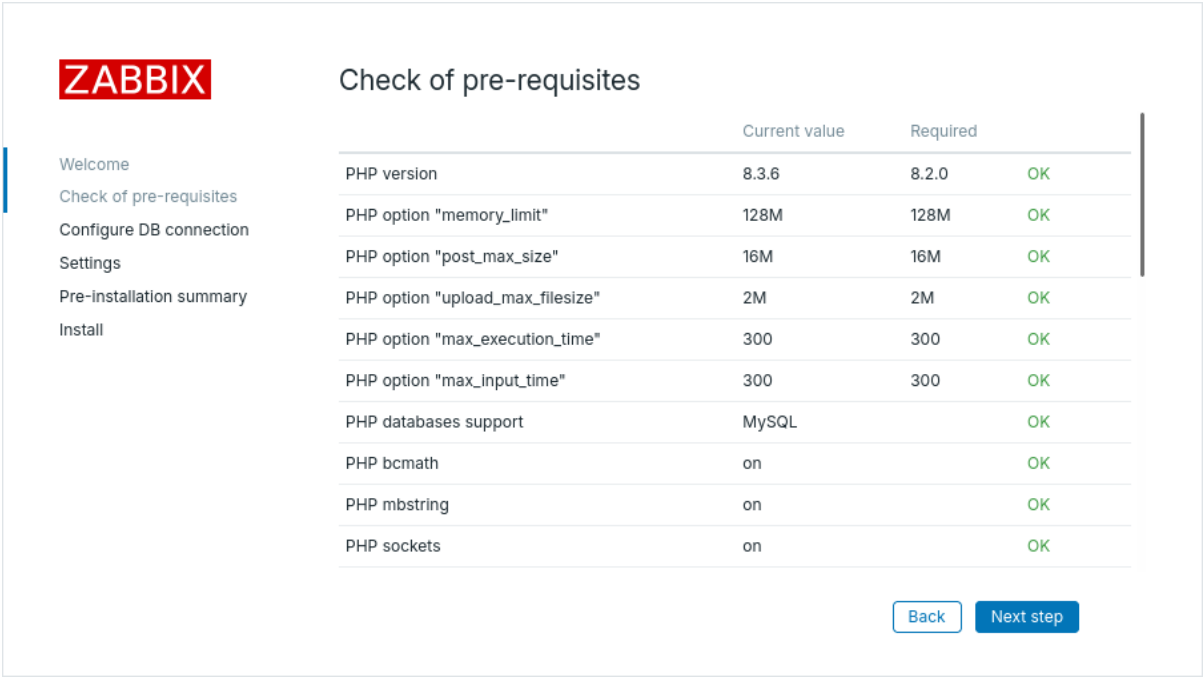


O idioma padrão de todo o sistema é o inglês (en_US), com o formato de hora/data dos EUA. Para alterar o idioma padrão de todo o sistema, selecione o idioma desejado no menu suspenso *Idioma padrão*.

Clique no botão *Próxima etapa* para continuar.

Verificação dos pré-requisitos

Certifique-se de que todos os pré-requisitos obrigatórios do frontend do Zabbix sejam atendidos.



Consulte Requisitos para obter uma lista das extensões PHP obrigatórias e opcionais, a versão necessária do PHP e os bancos de dados compatíveis. Observe que os pré-requisitos opcionais não atendidos são exibidos com o status *Warning*, e o processo de configuração pode continuar mesmo que eles não sejam atendidos.

Valores necessários para as diretivas de configuração do PHP no arquivo `php.ini` usado pelo servidor web:

```
memory_limit = 128M ; Valor mínimo
post_max_size = 16M ; Valor mínimo
upload_max_filesize = 2M ; Valor mínimo
```

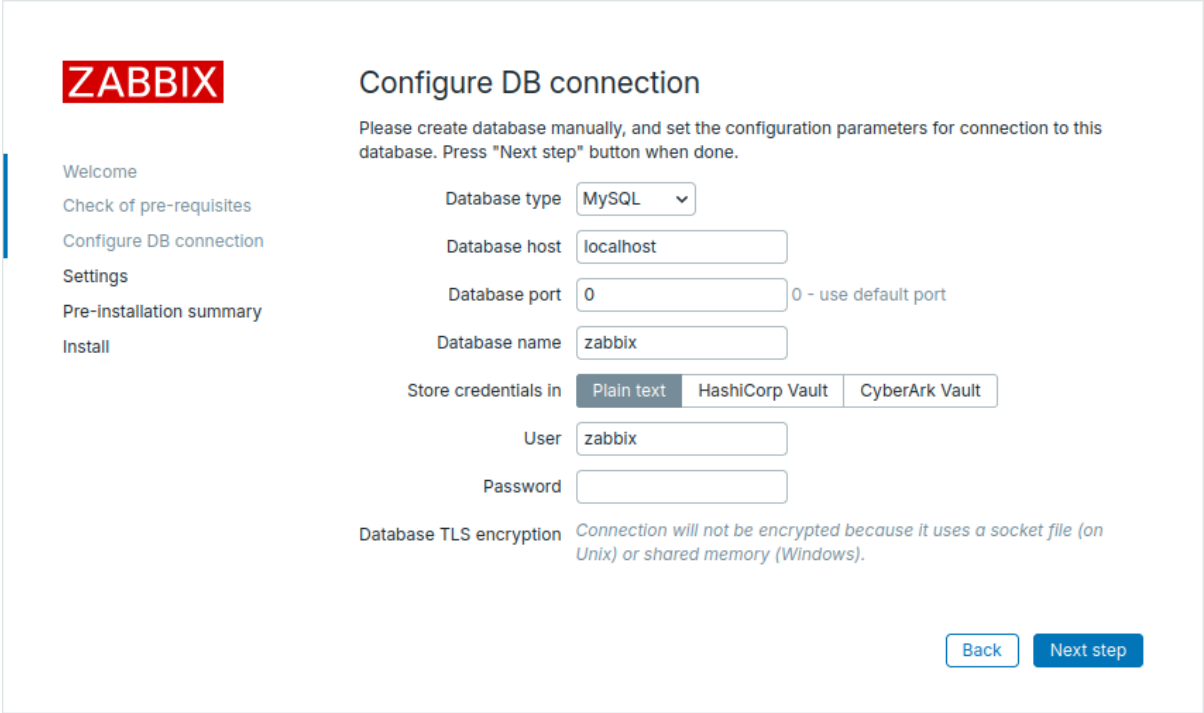
```
max_execution_time = 300 ; Valor mínimo; os valores 0 e -1 também são permitidos
max_input_time = 300 ; Valor mínimo; os valores 0 e -1 também são permitidos
session.auto_start = 0 ; Deve estar desabilitado
arg_separator.output = "&" ; "&amp;" também é permitido
```

Attention:
Se o usuário ou grupo do servidor web precisar ser alterado, verifique se o novo usuário ou grupo tem as permissões necessárias para o diretório de sessões do PHP; caso contrário, a configuração do Zabbix poderá não conseguir continuar.

Se todos os pré-requisitos obrigatórios forem atendidos, prossiga para a próxima etapa.

Configurar a conexão com o banco de dados

Insira os detalhes para conectar-se ao banco de dados.



Se o campo *Host do banco de dados* ficar vazio, o socket de domínio Unix padrão será usado.

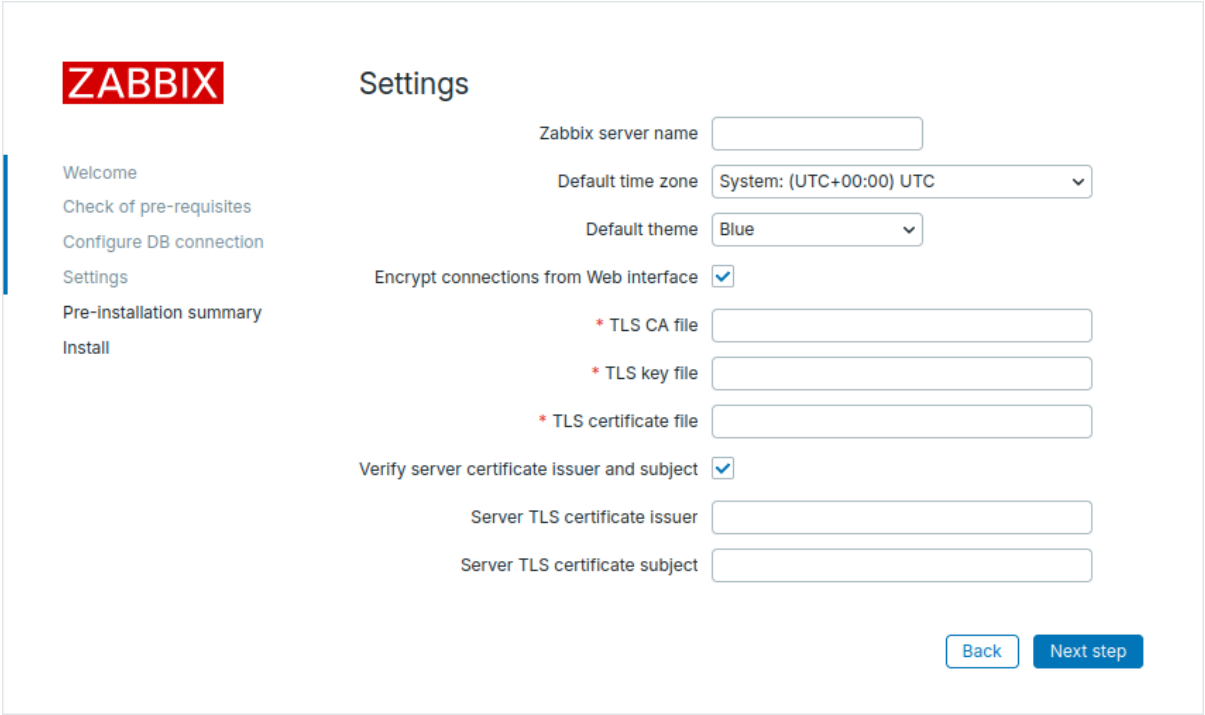
No MySQL, inserir `localhost` no campo *Host do banco de dados* também resulta em uma conexão por meio do socket de domínio Unix padrão. O assistente não fornece uma opção para configurar o socket do banco de dados. Para usar um socket personalizado, configure-o nas configurações do Zabbix server (por exemplo, com o parâmetro DBSocket= no arquivo de configuração do Zabbix server (`zabbix_server.conf`)). Isso mantém o frontend consistente com as configurações de conexão entre o server e o banco de dados.

No PostgreSQL, o campo *Host do banco de dados* pode conter várias entradas `host:port` separadas por vírgulas. A string é dividida em partes e cada porta é validada da mesma forma que o campo *Porta do banco de dados*. Após a validação bem-sucedida, a string é analisada em hosts e portas para construir a string de conexão. Se uma porta não for definida para um host na lista separada por vírgulas, o valor de *Porta do banco de dados* será usado como porta padrão. Se um caminho de socket for especificado (por exemplo, `/var/run/pgbouncer`), esse socket de domínio Unix será usado.

Para configurar uma conexão segura com o banco de dados, selecione a caixa de seleção *Criptografia TLS do banco de dados* e acesse Frontend para o banco de dados para configurar os campos. A caixa de seleção *Criptografia TLS do banco de dados* fica desabilitada nos seguintes casos: - Para MySQL, *Host do banco de dados* está definido como *localhost*. - Para PostgreSQL, o valor do campo *Host do banco de dados* começa com uma barra ou o campo está vazio.

Para configurar o Zabbix para recuperar segredos do HashiCorp Vault, selecione *HashiCorp Vault* em *Armazenar credenciais em* e acesse Configuração do HashiCorp para preencher os campos expandidos. Para configurar o Zabbix para recuperar segredos do CyberArk Vault, selecione *CyberArk Vault* em *Armazenar credenciais em* e acesse Configuração do CyberArk para preencher os campos expandidos.

Configurações



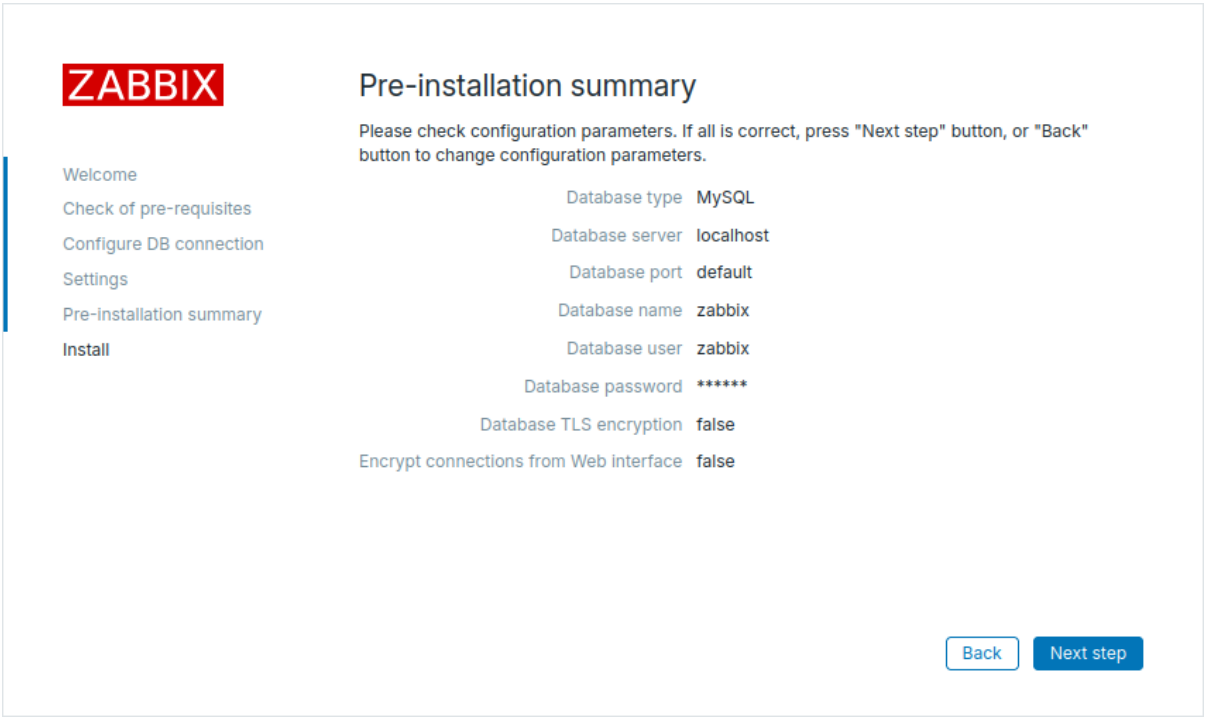
A inserção de um nome para o Zabbix server é opcional; no entanto, se for informado, ele será exibido na barra de menu e nos títulos das páginas.

Defina o fuso horário padrão e o tema do frontend.

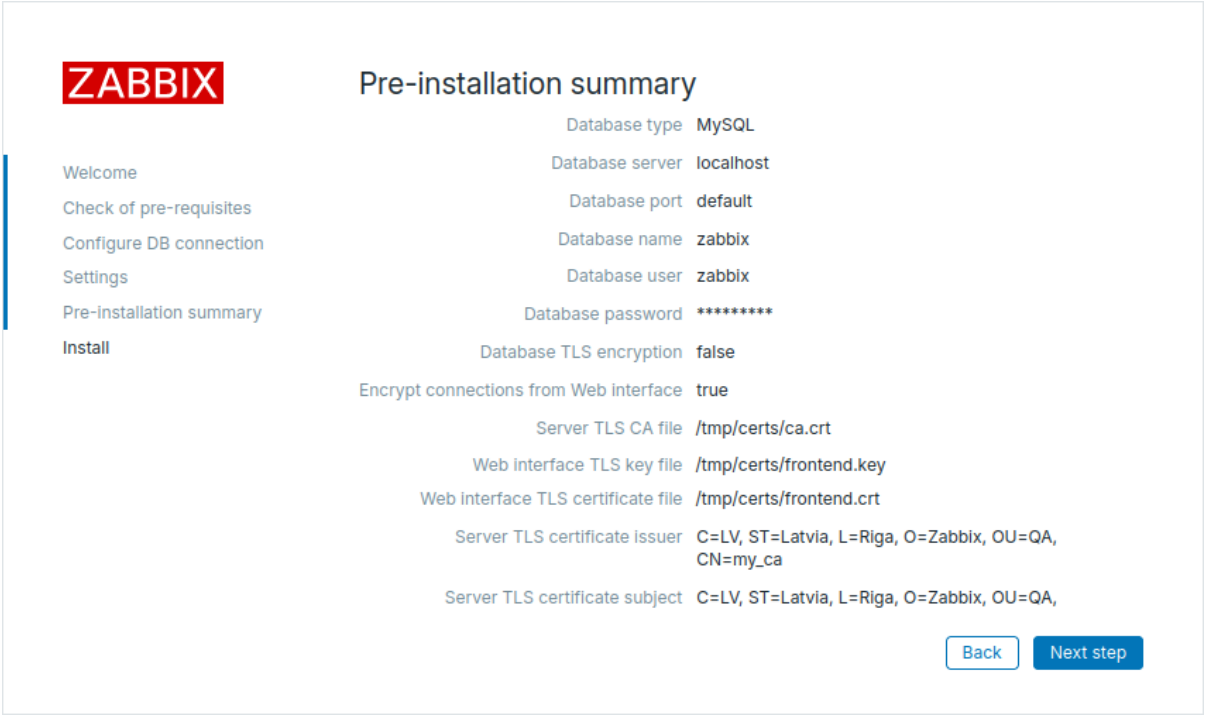
Para configurar conexões TLS entre o frontend do Zabbix e o Zabbix server, selecione a caixa de seleção *Criptografar conexões da interface Web* e prossiga para configurar a conexão TLS para preencher os campos expandidos.

Resumo da pré-instalação

Revise um resumo das configurações.



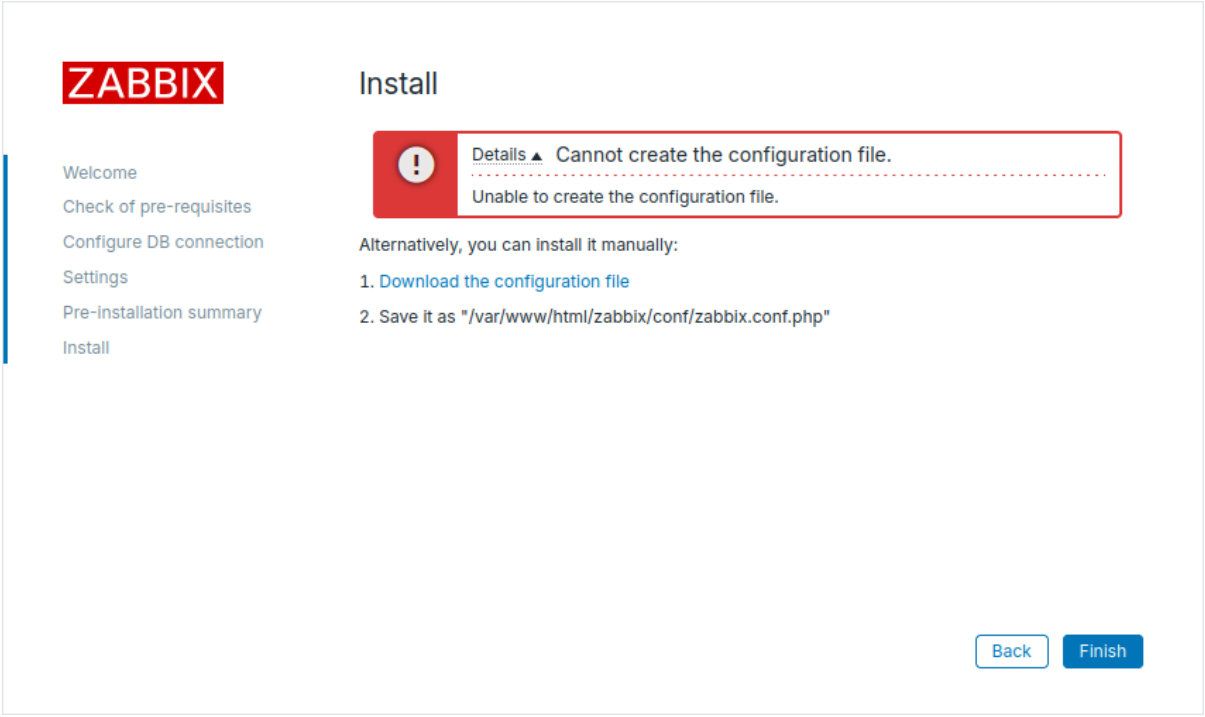
Se o TLS estiver configurado, o assistente exibirá seus detalhes.



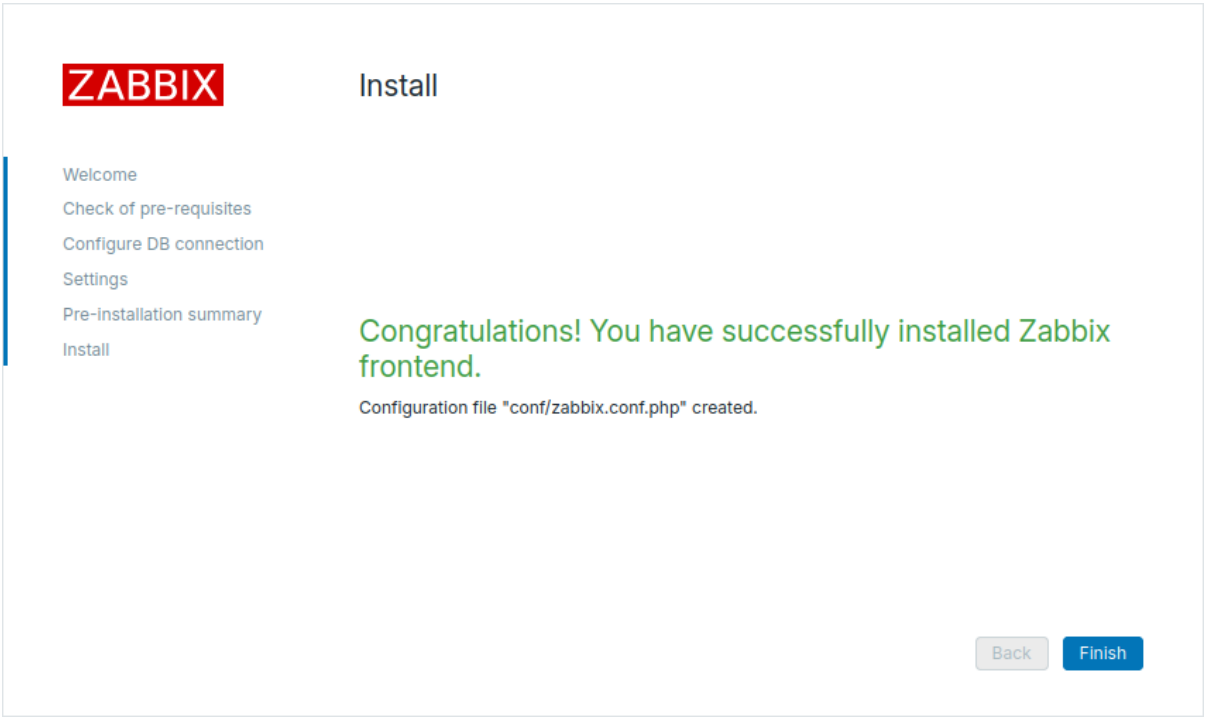
Instalação

Se o usuário do web server tiver acesso de gravação ao diretório `conf/`, o arquivo de configuração será salvo automaticamente, e você poderá prosseguir imediatamente para a próxima etapa.

Se o usuário do web server não tiver acesso de gravação ao diretório `conf/`, será necessário baixar o arquivo de configuração e colocá-lo no diretório `conf/`, dentro do diretório de documentos HTML do web server onde você copiou os arquivos PHP do Zabbix.

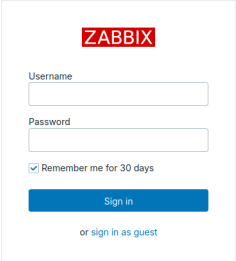


Conclua a instalação.



Fazer login

Faça login usando as credenciais padrão: - Nome de usuário: *Admin* - Senha: *zabbix*



Arquivo de configuração do frontend

Após a instalação, você pode reconfigurar o frontend do Zabbix atualizando os parâmetros no arquivo de configuração do frontend `zabbix.conf.php`.

O caminho padrão para o arquivo de configuração depende do sistema operacional e do método de instalação do Zabbix:

- Para instalação a partir de pacotes: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- Para instalação a partir do código-fonte: `/var/www/html/zabbix/conf/zabbix.conf.php`
- Para appliance: `/etc/zabbix/web/zabbix.conf.php`

Consulte as opções do arquivo de configuração para obter detalhes sobre como configurar o frontend do Zabbix.

Guias de início rápido

Depois de instalar o Zabbix e sua interface web, é hora de começar o monitoramento.

Use estes guias rápidos para começar a monitorar recursos populares:

- Monitoramento de server Linux
- Monitoramento de server web Apache
- Monitoramento de server MySQL

Os guias de início rápido se concentram no uso de templates oficiais do Zabbix para monitoramento.
Um template é um conjunto pré-configurado de elementos de monitoramento.
Um template permite dimensionar o monitoramento: você pode testá-lo em um único alvo de monitoramento (dispositivo, aplicativo etc.), ajustá-lo e, em seguida, aplicá-lo a vários alvos de monitoramento semelhantes.

Embora os templates sejam adequados para o monitoramento pronto para uso, você também pode ter interesse em personalizar seus elementos.

Para entender melhor os elementos básicos que compõem um template, consulte Configuração básica.

Configuração básica

Estes guias apresentam os elementos básicos da configuração do Zabbix:

- Login e configuração de usuário
- Novo host
- Novo item
- Novo trigger
- Recebimento de notificação de problema
- Novo template

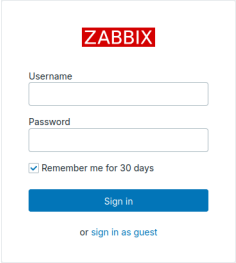
Login e configuração do usuário

Visão geral

Nesta seção, você aprenderá como fazer login e configurar um usuário do sistema no Zabbix.

Login

Abra a URL do frontend do Zabbix em um navegador: - Para Apache: http://<server_ip_or_name>/zabbix - Para NGINX: http://<server_ip_or_name>



Esta é a tela de boas-vindas do Zabbix. Insira 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 conectado 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.
- *Auto-logout* é substituído, mantendo você conectado até o fim desse período.
- Você será conectado automaticamente em visitas futuras dentro de 30 dias, sem precisar inserir novamente as credenciais.

Lembrar por 30 dias desativado:

- Qualquer login automático previamente ativado é removido.
- A sessão expirará de acordo com o intervalo configurado de *Auto-logout*.

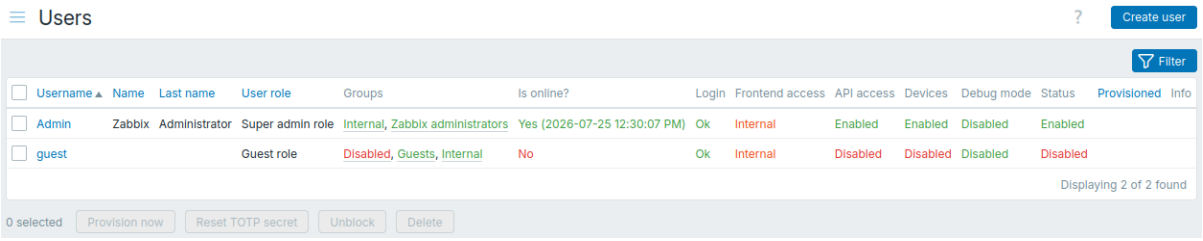
Proteção contra ataques de força bruta

Em caso de cinco tentativas consecutivas de login malsucedidas, a interface do Zabbix fará uma pausa de 30 segundos para evitar ataques de força bruta e ataques de dicionário.

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

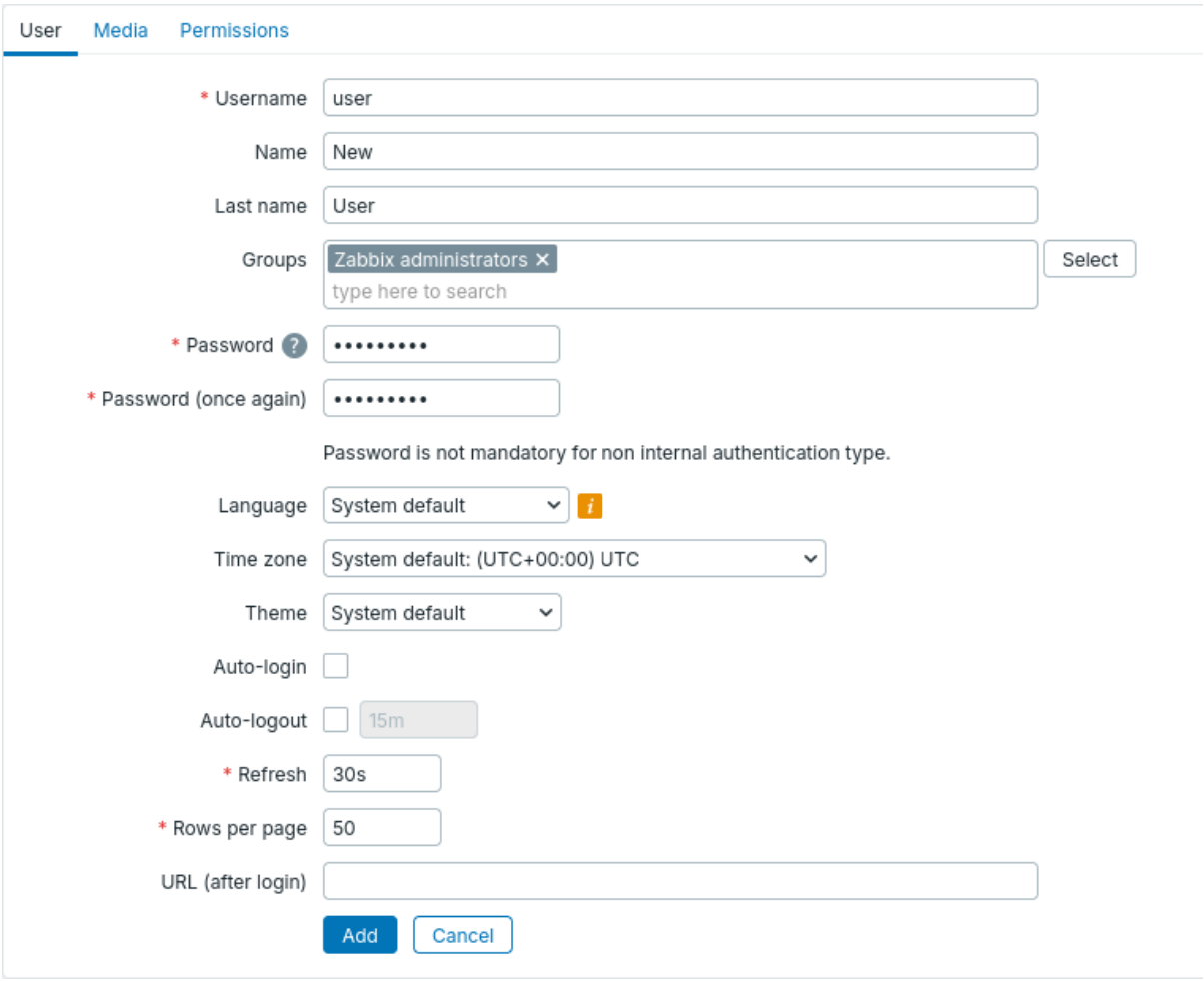
Adicionando usuário

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



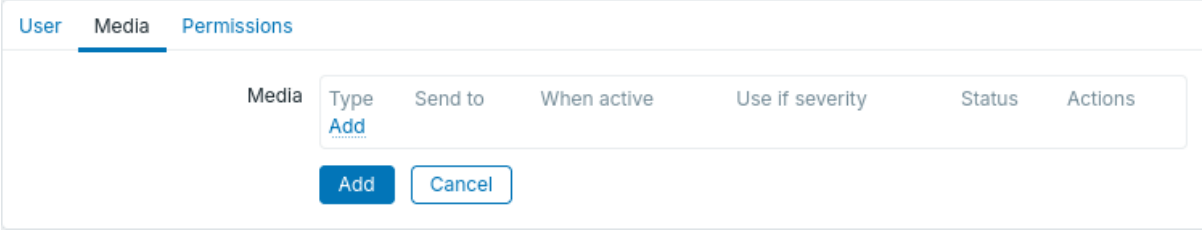
Para adicionar um novo usuário, selecione *Create user* no canto superior direito.

No formulário de novo usuário, certifique-se de adicionar seu usuário a um dos grupos de usuários existentes, por exemplo, 'Zabbix administrators'.



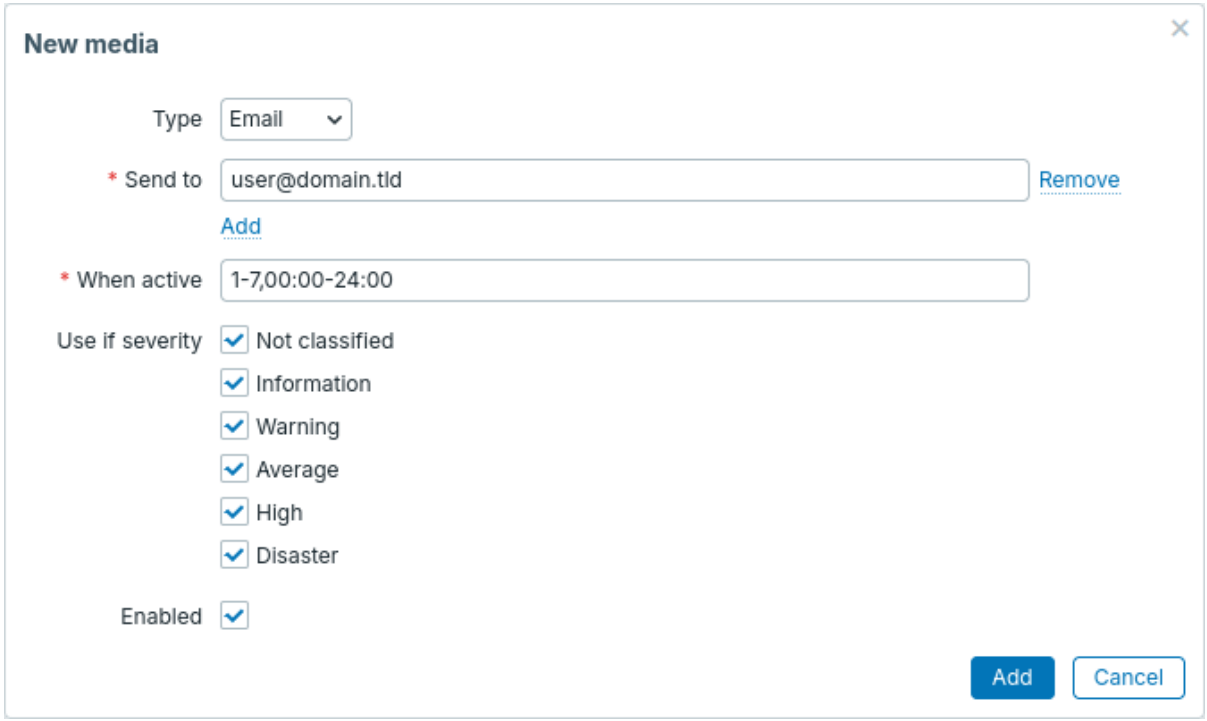
Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho. Para detalhes sobre os campos de entrada neste formulário de configuração, consulte a página User settings.

Por padrão, novos usuários não têm media (métodos de entrega de notificações) definidos para eles. Para criar um, vá para a aba *Media* e clique em *Add*.



No pop-up, informe o endereço de e-mail do usuário.
 Todos os tipos de media estão desativados por padrão e podem ser ativados em *Alerts* > *Media types*.

Você pode especificar um período de tempo em que o meio ficará ativo (consulte a página Time period specification para uma descrição do formato). Por padrão, um meio está sempre ativo. Você também pode personalizar os níveis de trigger severity para os quais o meio ficará ativo, mas deixe todos habilitados por enquanto.



Clique em *Add* para salvar o meio e, em seguida, vá para a aba *Permissions*.

A aba Permissions tem um campo obrigatório *Role*. A função determina quais elementos do frontend o usuário pode visualizar e quais ações ele tem permissão para executar. Clique em *Select* e selecione uma das funções da lista. Por exemplo, selecione *Admin role* para permitir acesso a todas as seções do frontend do Zabbix, exceto *Administration*. Mais tarde, você pode modificar as permissões ou criar mais funções de usuário. Ao selecionar uma função, as permissões aparecerão na mesma aba:

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

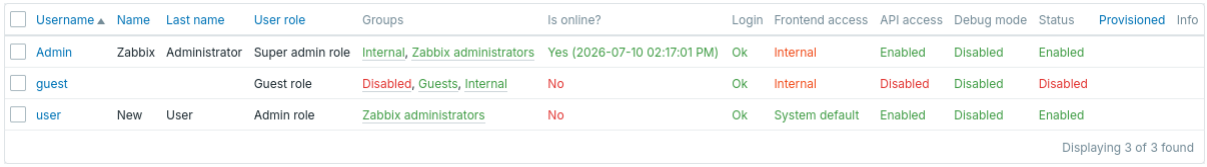
Enabled | Manage own devices

Access to actions

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

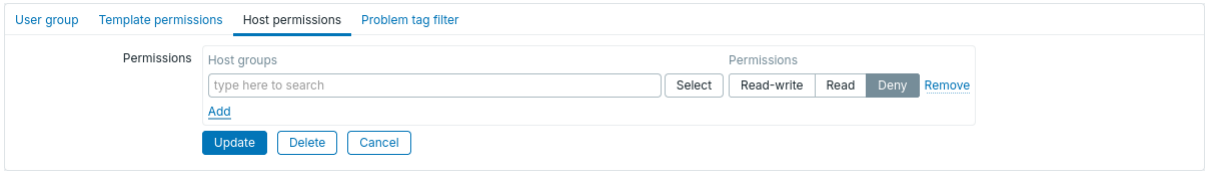
Add | Cancel

Clique em *Add* no formulário de propriedades do usuário para salvar o usuário. O novo usuário aparece na lista de usuários.

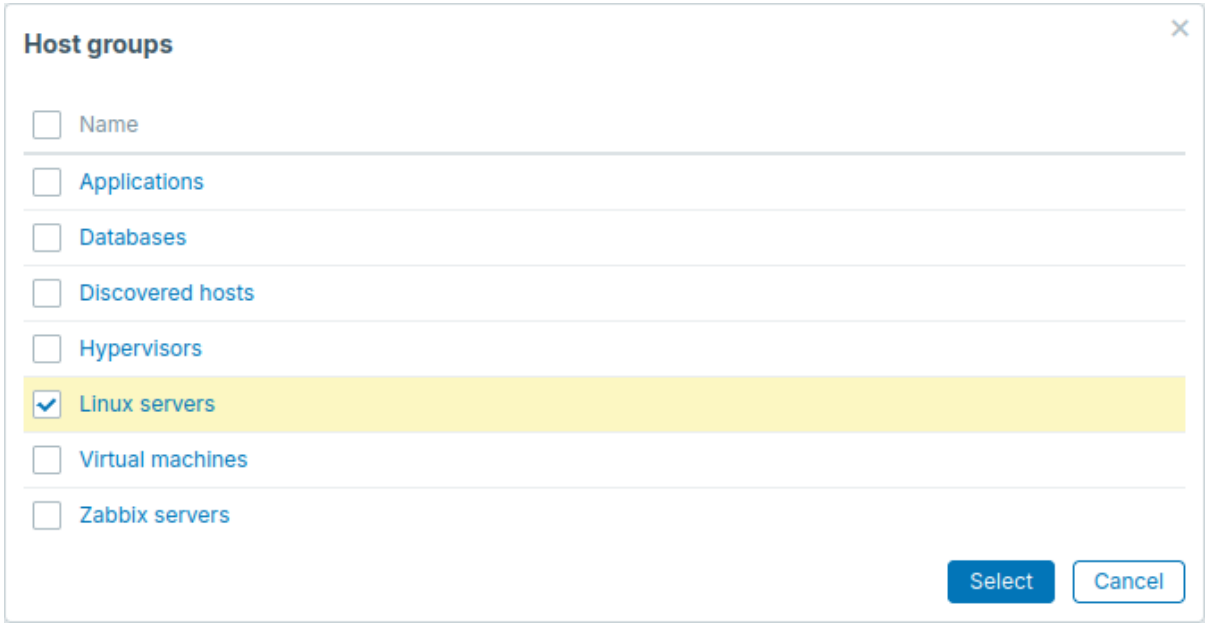


Adicionando permissões

Por padrão, um novo usuário não tem permissões para acessar hosts e templates. Para conceder direitos ao usuário, clique no grupo do usuário na coluna *Groups* (neste caso - *Zabbix administrators*). No formulário de propriedades de *User groups*, vá para a aba *Host permissions* para atribuir permissões aos grupos de hosts. Clique em Add para exibir o campo de seleção do grupo de hosts:



Em seguida, clique em *Select* ao lado do campo para ver a lista dos grupos de hosts. Este usuário deve ter acesso somente leitura ao grupo *Linux servers*, então marque a caixa de seleção apropriada na lista e clique em *Select* para confirmar sua escolha.



Clique no botão *Read* para definir o nível de permissão e depois em *Update* para salvar as alterações feitas na configuração do grupo de usuários.

Para conceder permissões a templates, você precisará mudar para a aba *Template permissions* e especificar grupos de templates. As etapas são idênticas às de atribuição de permissões aos grupos de hosts. Uma visão geral de templates está disponível na seção New template deste Quickstart.

Attention:
No Zabbix, os direitos de acesso a hosts e templates são atribuídos a user groups, e não a usuários individuais.

Pronto! Você pode tentar fazer login usando as credenciais do novo usuário.

Novo host

Visão geral

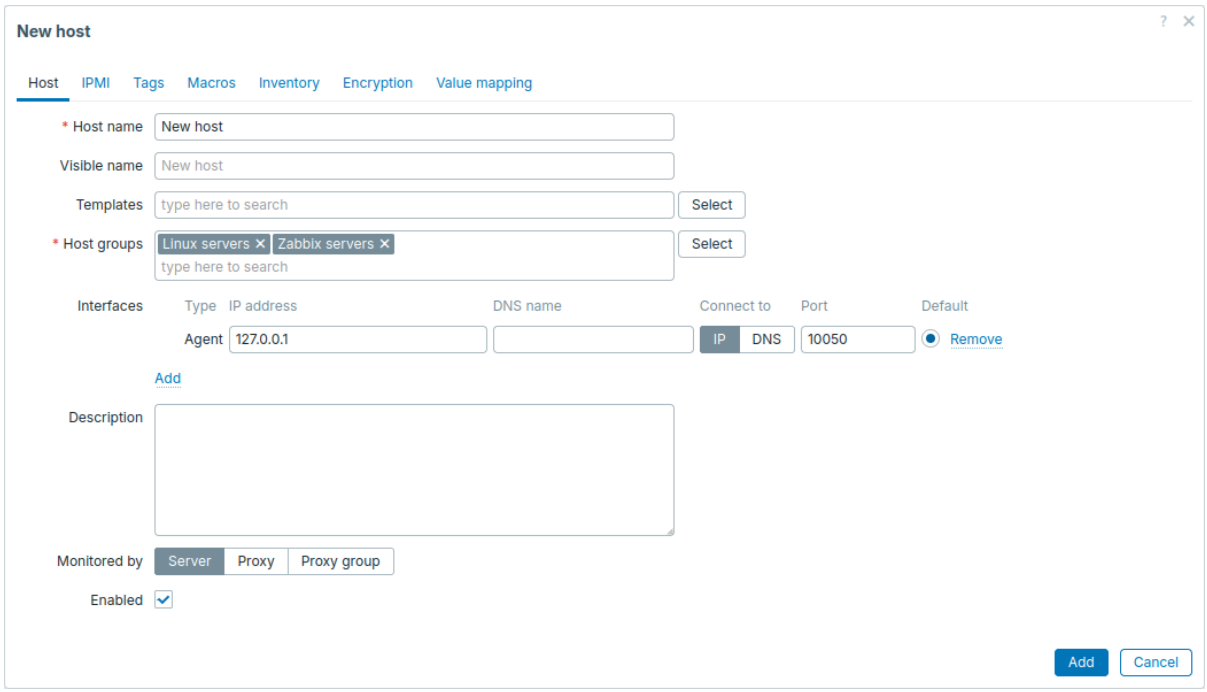
Nesta seção, você aprenderá como configurar um novo host.

Um host no Zabbix é uma entidade em 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 um host

As informações sobre os hosts configurados no Zabbix estão disponíveis na seção de menu *Coleta de dados* > *Hosts*, bem como na seção de menu *Monitoramento* > *Hosts*. Já existe um host predefinido, chamado "Zabbix server", mas queremos aprender a adicionar outro.

Para adicionar um novo host, clique em *Criar host*. Um formulário de configuração do host será aberto.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

O mínimo necessário a ser informado aqui é:

Nome do host

- Informe um nome para o host. São permitidos caracteres alfanuméricos, espaços, pontos, hífens e sublinhados.

Grupos de hosts

- Selecione um ou vários grupos existentes clicando no botão *Selecionar* ou informe o nome de um grupo inexistente para criar um novo grupo.

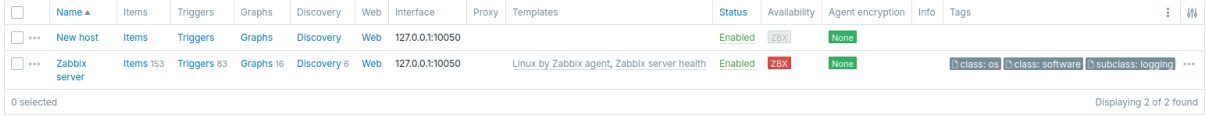
Note:
Todas as permissões de acesso são atribuídas a grupos de hosts, não a hosts individuais. Por isso, um host deve pertencer a pelo menos um grupo.

Interfaces: endereço IP

- Embora tecnicamente não seja um campo obrigatório, uma interface de host é necessária para coletar determinadas métricas. Para usar verificações passivas do Zabbix agent, 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 do Zabbix. Se o Zabbix agent e o Zabbix server estiverem instalados na mesma máquina, será necessário especificar o mesmo IP/DNS nos dois locais.

Outras opções podem permanecer inalteradas, usando seus valores padrão.

Quando terminar, clique em *Adicionar*. Seu novo host deverá estar visível na lista de hosts.



A coluna *Disponibilidade* contém indicadores da disponibilidade do host para cada interface. Quando uma interface do Zabbix agent é definida, podemos usar o ícone de disponibilidade do agent (identificado como *ZBX*) para determinar a disponibilidade do host:

- ZBX - o status do host ainda não foi estabelecido; nenhuma verificação de métrica foi realizada.
- 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 incorretas da interface. Verifique se o Zabbix server está em execução e tente atualizar a página novamente mais tarde.

Novo item

Visão geral

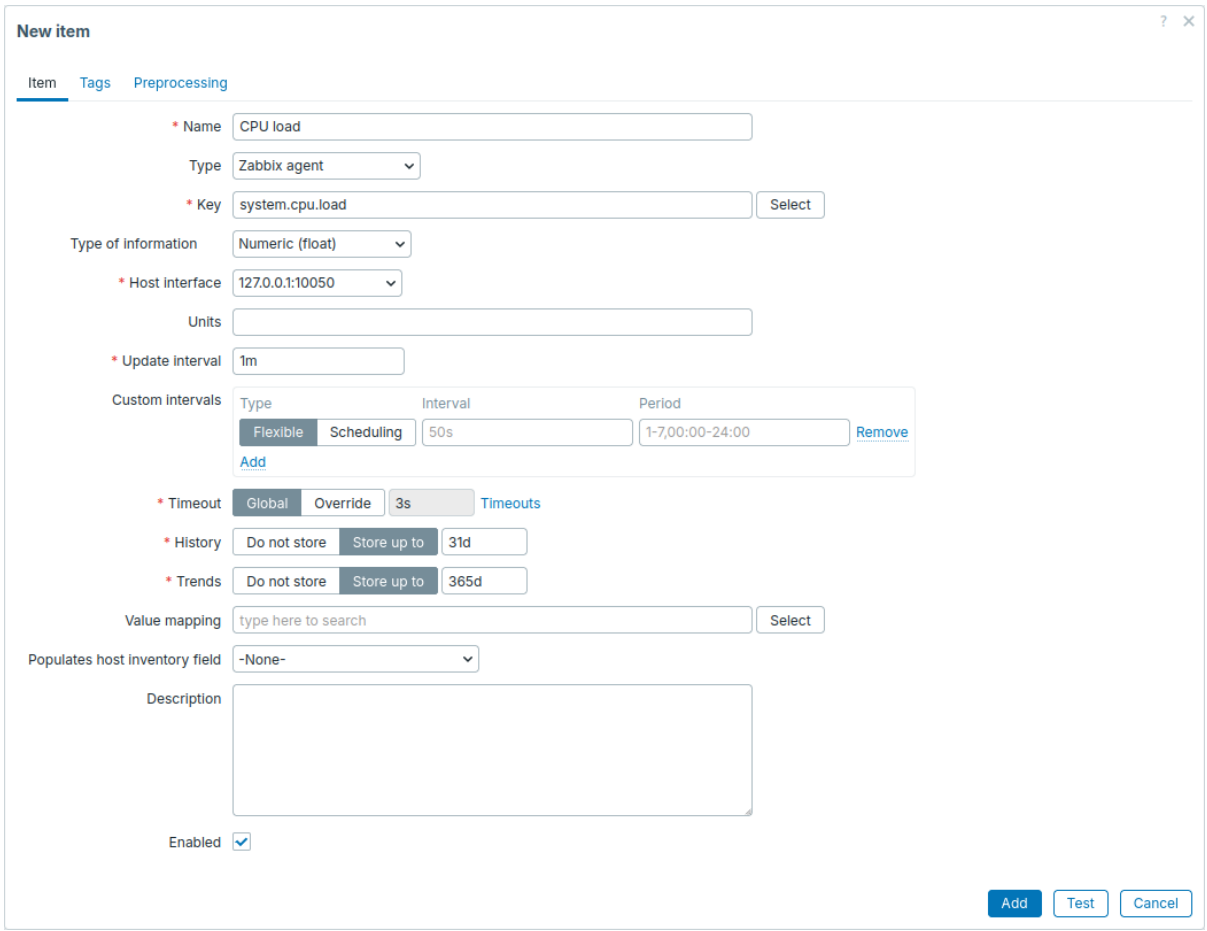
Nesta seção, você aprenderá como configurar um item.

Os items são a base para a coleta de dados no Zabbix. Sem items, não há dados - porque somente um item define uma única métrica ou que tipo de dado coletar de um host.

Adicionando item

Todos os items são agrupados em torno de hosts. Por isso, para configurar um item de exemplo, vamos para *Data collection* > *Hosts* e encontramos o "New host" que criamos.

Clique no link *Items* na linha de "New host" e, em seguida, clique em *Create item*. Um formulário de configuração de item é aberto.



Todos os campos obrigatórios são marcados com um asterisco vermelho.

Para o nosso item de exemplo, as informações essenciais a serem inseridas são:

Name

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

Key

- Insira manualmente *system.cpu.load* como valor. Este é o nome técnico de um item que identifica o tipo de informação que será coletada. A chave específica é apenas uma das chaves predefinidas que acompanham o Zabbix agent.

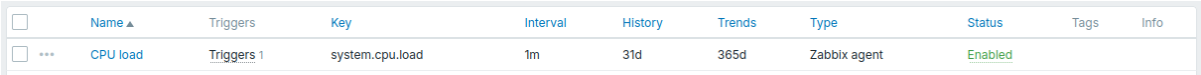
Type of information

- Este atributo define o formato dos dados esperados. Para a chave *system.cpu.load*, este campo será definido automaticamente como *Numeric (float)*.

Note:
Você também pode querer reduzir para 7 ou 14 o número de dias que o histórico do item será mantido. Essa é uma boa prática para aliviar o banco de dados de armazenar muitos valores históricos.

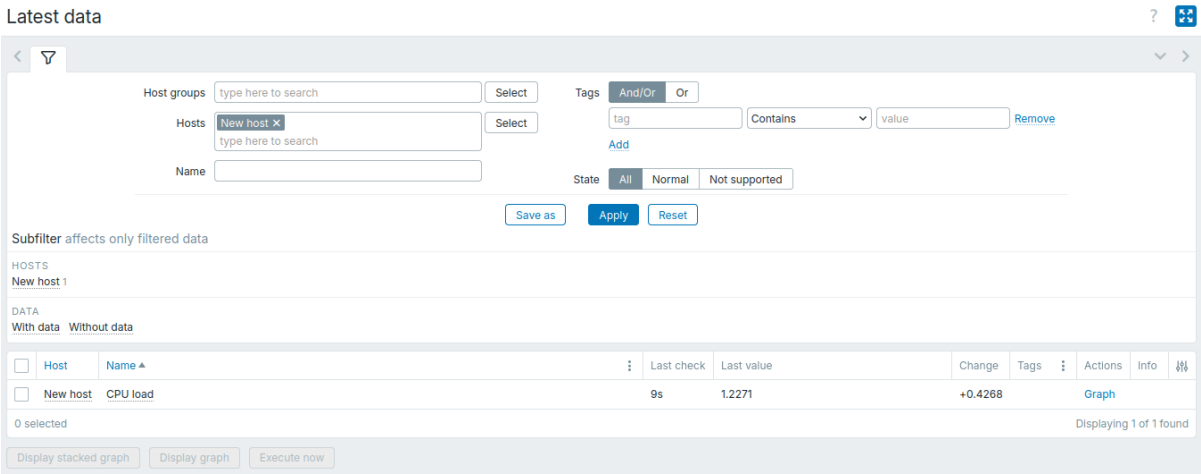
Outras opções podem ser deixadas inalteradas, usando seus valores padrão.

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



Visualizando dados

Com um item definido, você pode ficar curioso para saber se ele está realmente coletando dados. Para isso, vá em *Monitoring* > *Latest data*, selecione 'New host' no filtro e clique em *Apply*.



Dito isso, pode levar até 60 segundos para que os primeiros dados cheguem. Esse é, por padrão, o intervalo com que o server lê as alterações de configuração e identifica novos items para executar.

Se você não vir nenhum valor na coluna 'Change', talvez apenas um valor tenha sido recebido até agora. Aguarde 30 segundos para que outro valor chegue.

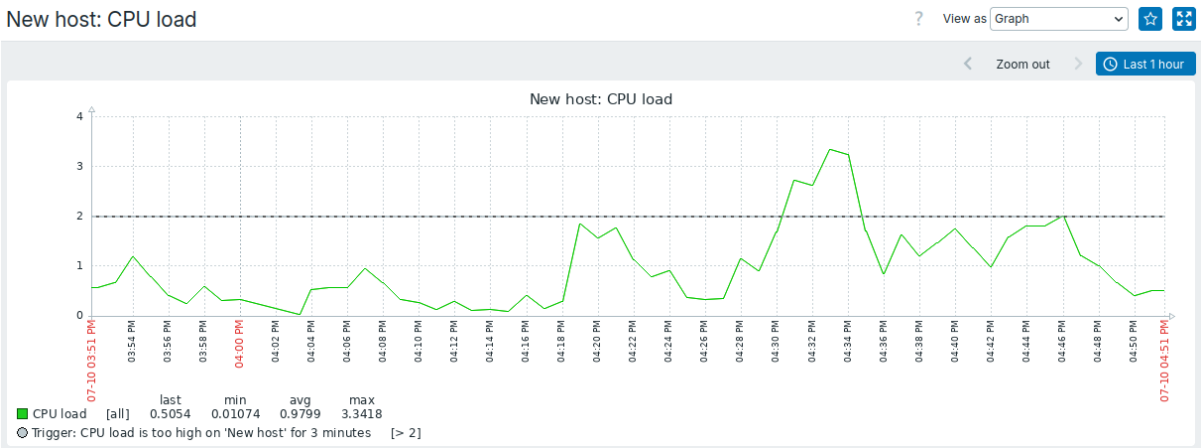
Se você não vir informações sobre o item como na captura de tela, verifique se:

- você preencheu os campos 'Key' e 'Type of information' do item exatamente como na captura de tela;
- tanto o agent quanto o server estão em execução;
- o status do host é 'Monitored' 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 algum tempo, pode ser o momento de visualizar algo mais concreto. 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, acesse *Monitoramento* > *Dados recentes* e clique no link *Gráfico* ao lado do item.



Novo trigger

Visão geral

Nesta seção, você aprenderá como configurar um trigger.

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.

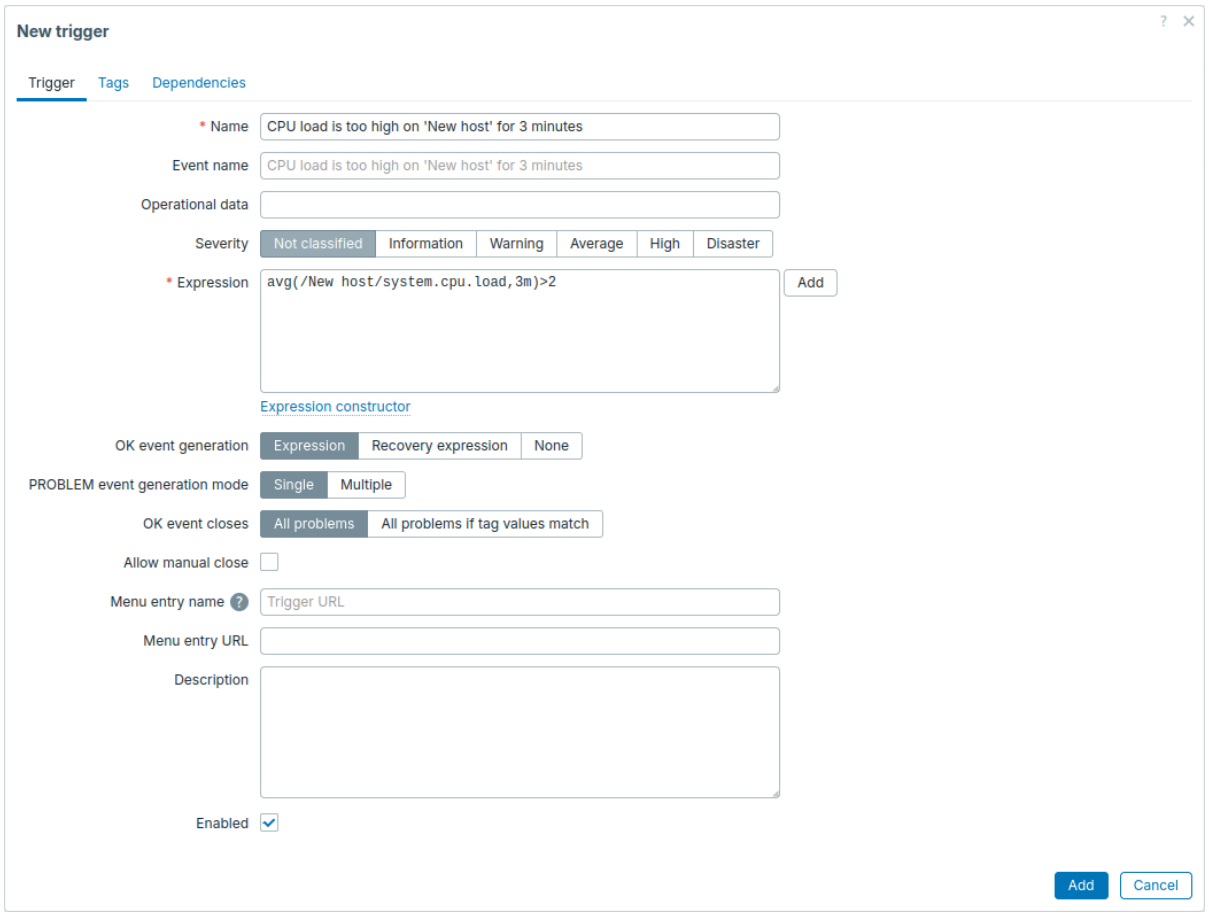
Se esse nível for ultrapassado pelos dados recebidos, um trigger irá "disparar" ou entrar em um estado de 'Problem' - indicando que algo aconteceu e pode exigir atenção. Se o nível voltar a ser aceitável, o trigger retorna ao estado 'Ok'.

Adicionando trigger

Para configurar um trigger para o nosso item:

1. Acesse *Data collection* > *Hosts*.
2. Clique no link *Triggers* na linha de 'New host'.
3. Clique em *Create trigger*.

Um formulário de configuração de trigger é aberto.



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

Name

- Insira *CPU load is too high on 'New host' for 3 minutes* como valor. Este será o nome do trigger exibido em listas e em outros locais.

Expression

- Insira: `avg(/New 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 referenciar o item. Essa expressão, basicamente, diz que o limite de problema é excedido quando o valor médio da carga de CPU por 3 minutos é maior que 2. Você pode saber mais sobre a sintaxe de expressões de trigger.

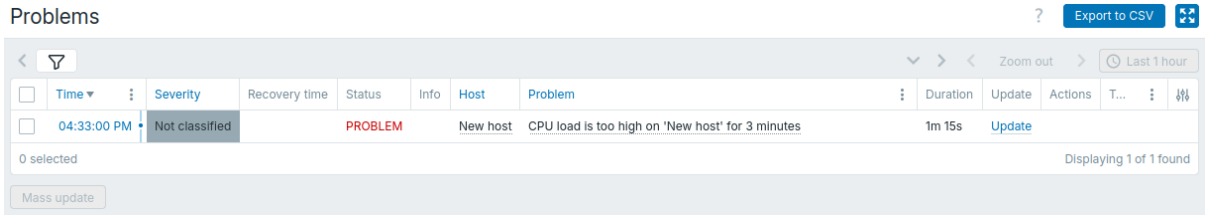
Outras opções podem ser deixadas inalteradas, usando seus valores padrão.

Ao terminar, clique em *Add*. O novo trigger deve aparecer na lista de triggers.

Exibindo o estado do trigger

Com um trigger definido, você pode querer ver seu estado.

Se a carga da CPU tiver excedido o nível de limite que você definiu no trigger, o problema será exibido em *Monitoring* > *Problems*.



A intermitência na coluna de status indica uma alteração recente do status do trigger, ocorrida nos últimos 30 minutos.

Recebendo notificação de problema

Visão geral

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

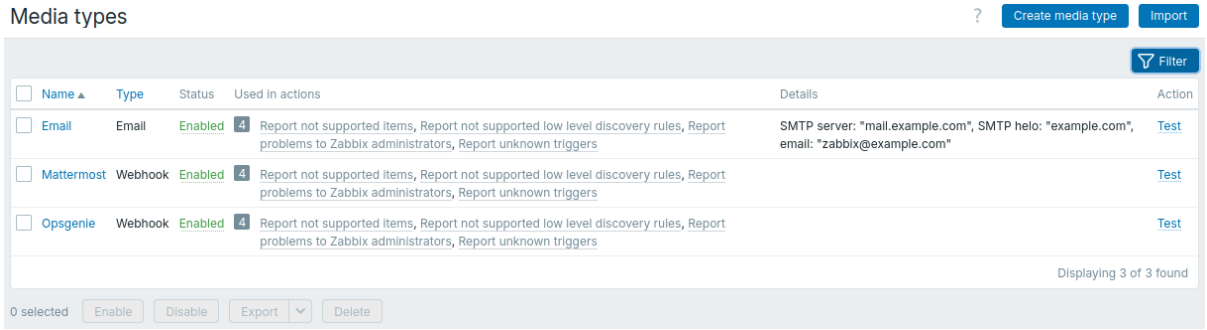
Com os items coletando dados e os triggers projetados para ”disparar” em situações de problema, também seria útil ter algum mecanismo de alerta em vigor para notificar sobre eventos importantes, mesmo quando o frontend do Zabbix não estiver sendo verificado ativamente.

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.

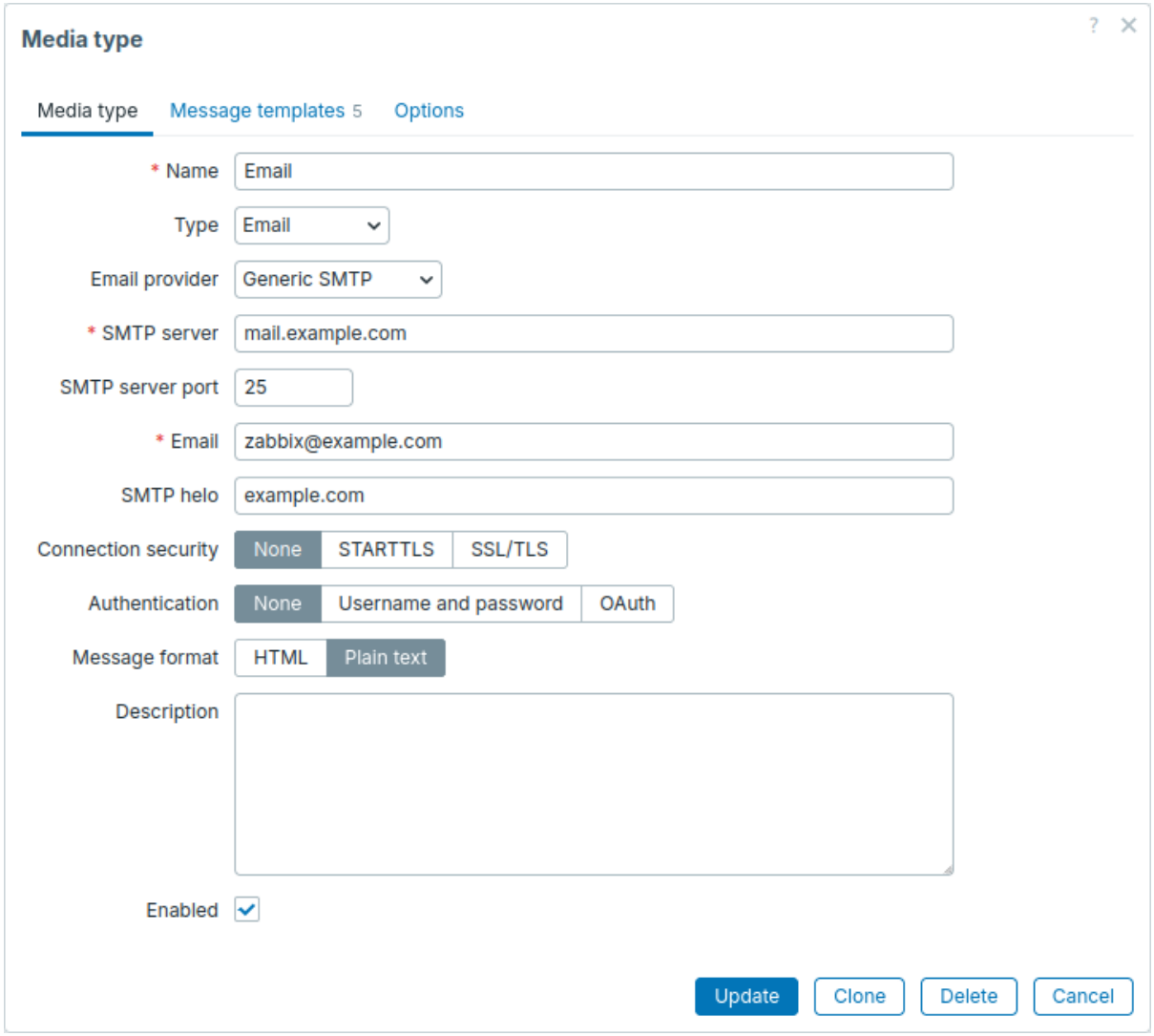
Configurações de email

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

Para configurar as definições de email, vá para *Alerts* > *Media types* e clique em *Email* na lista de tipos de mídia predefinidos.



Um formulário de configuração das definições de email é aberto.



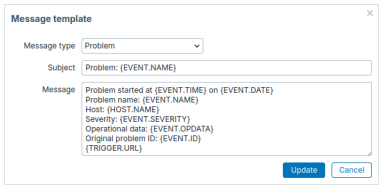
Todos os campos obrigatórios são marcados com um asterisco vermelho.

Na aba *Media type*, defina os valores de *SMTP server*, *SMTP helo* e *Email* de acordo com o seu ambiente.

Note:
O valor do campo *Email* será usado como endereço 'From' 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 *Message templates*.

Clique em *Add* para criar um modelo de mensagem e selecione *Problem* como o tipo de mensagem.

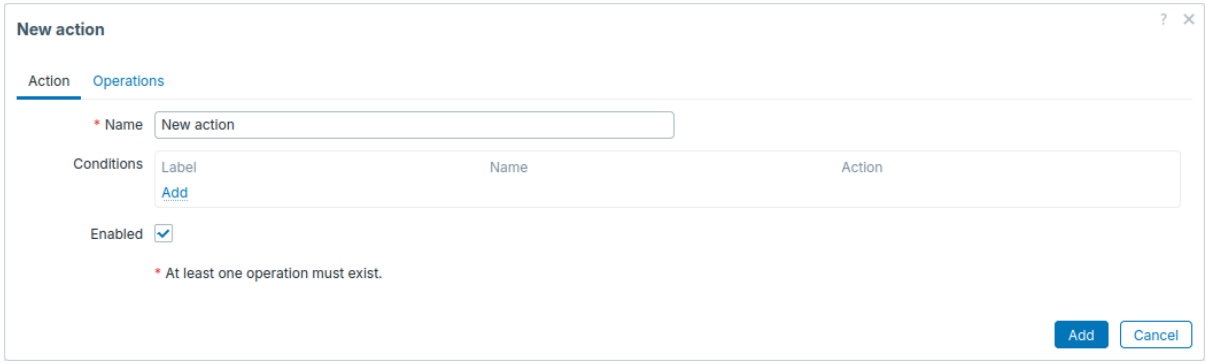


Clique em *Add* quando estiver pronto e salve o formulário.

Agora você configurou *Email* como um tipo de mídia funcional. O tipo de mídia também deve ser vinculado aos usuários, definindo endereços de entrega específicos (como fizemos ao configurar um novo usuário); caso contrário, ele não será usado.

Nova action

O envio de notificações é uma das coisas que actions fazem no Zabbix. Portanto, para configurar uma notificação, vá para *Alerts* > *Actions* > *Trigger actions* e clique em *Create action*.

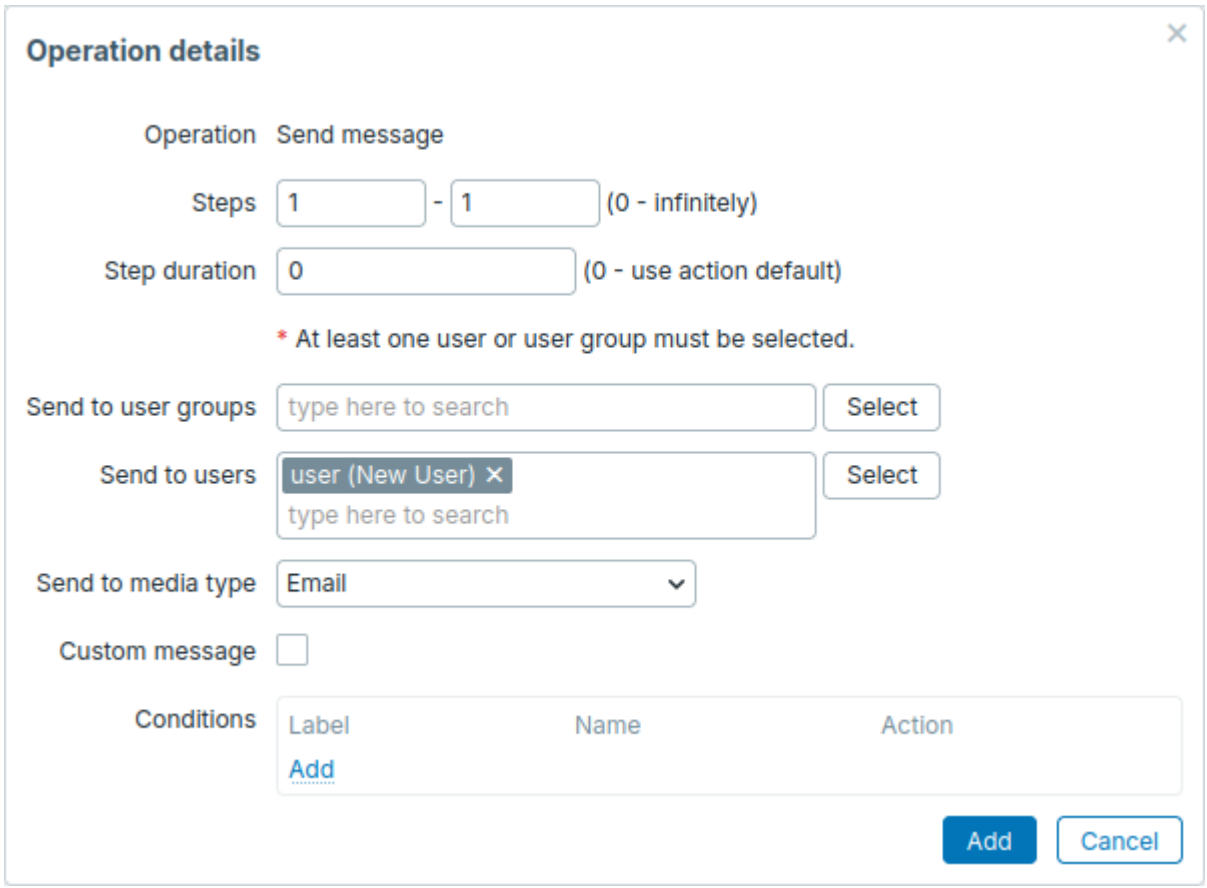


Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Neste formulário, informe um nome para a action.

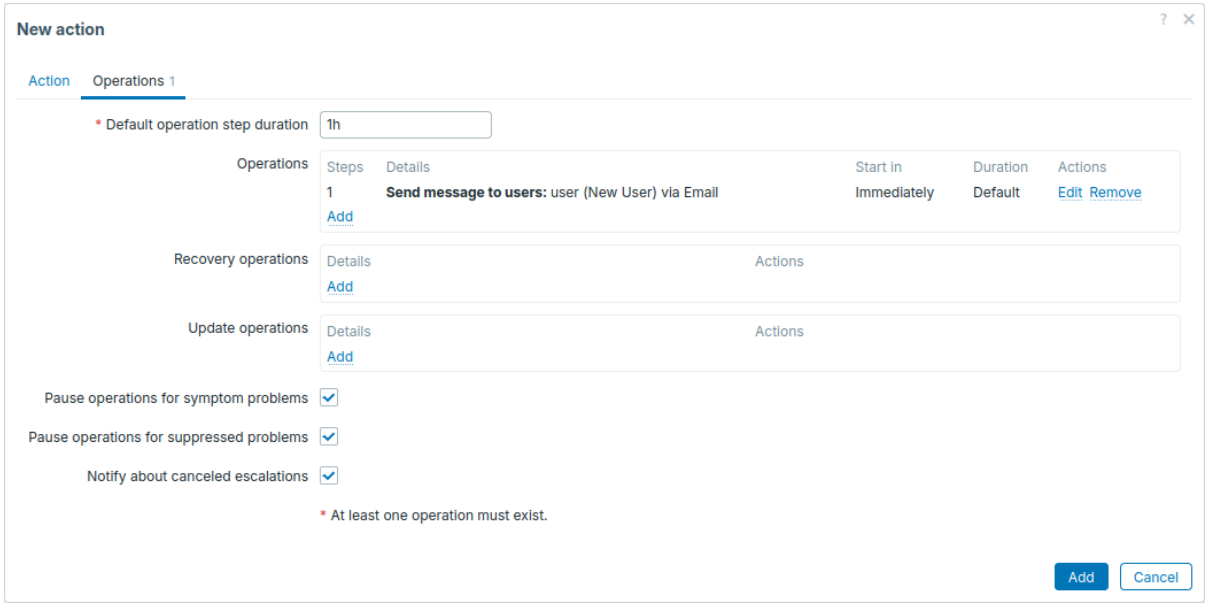
No caso mais simples, se não adicionarmos nenhuma conditions mais específica, a action será executada em qualquer alteração de trigger de 'Ok' para 'Problem'.

Ainda precisamos definir o que a action deve fazer - e isso é feito na aba *Operations*. Clique em *Add* no bloco *Operations*, o que abrirá um novo formulário de operação.



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Aqui, clique em *Select* no bloco *Send to Users* e selecione o usuário ('user') que definimos. Selecione "Email" como o valor de *Send to media type*. Quando terminar, clique em *Add*, e a operação deverá ser adicionada:



Isso é tudo para uma configuração simples de action, então clique em *Add* no formulário da action.

Recebendo notificação

Agora, com o envio de notificações configurado, seria interessante receber uma de fato. Para ajudar nisso, podemos aumentar de propósito a carga no nosso host - para que nosso trigger "dispare" e recebamos uma notificação de problema.

Abra o console no seu host e execute:

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

Você pode executar um ou vários desses processos.

Agora vá para *Monitoring* > *Latest data* e veja como os valores de 'CPU Load' aumentaram. Lembre-se: para o nosso trigger "disparar", o valor de 'CPU Load' precisa ficar acima de '2' por 3 minutos seguidos. Quando isso acontecer:

- em *Monitoring* > *Problems* você deverá ver o trigger com um estado `Problem` piscando;
- você deverá receber uma notificação de problema no seu email.

Attention:
Se as notificações não funcionarem:

- verifique novamente se tanto as configurações de email quanto a action foram configuradas corretamente
- certifique-se de que o usuário que você criou tenha pelo menos permissões de leitura no host que gerou o evento, conforme observado na etapa *Adicionando usuário*. O usuário, por fazer 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 da action acessando *Reports* > *Action log*.

Novo template

Visão geral

Nesta seção, você aprenderá a configurar um template.

Anteriormente, aprendemos a configurar um item e um trigger, além de obter uma notificação de problema para o host.

Embora todas essas etapas ofereçam muita flexibilidade por si só, pode parecer que são muitas etapas a serem executadas quando isso é necessário para, por exemplo, mil hosts. Algum tipo de automação seria útil.

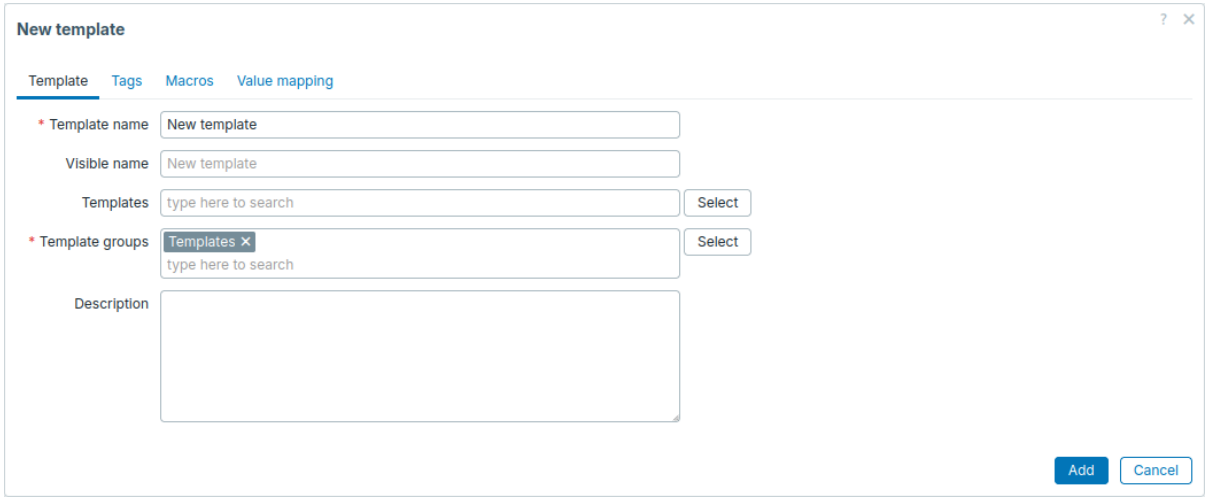
É nesse ponto que os templates ajudam.

Os templates permitem agrupar itens, triggers e outras entidades úteis para que possam ser reutilizados várias vezes, 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 de verificações previamente preparado pode ser aplicado muito rapidamente.

Adicionando template

Para começar a trabalhar com templates, primeiro precisamos criar um. Para isso, em *Data collection* > *Templates*, clique em *Create template*. Isso abrirá um formulário de configuração de template.



Todos os campos obrigatórios são marcados com um asterisco vermelho.

Os parâmetros obrigatórios a serem informados aqui são:

Template name

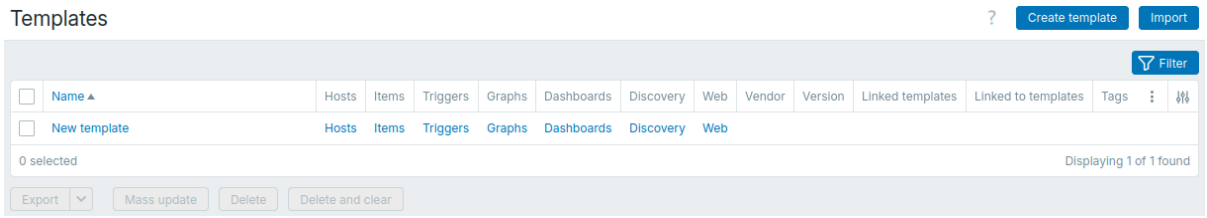
- Informe um nome para o template. São permitidos caracteres alfanuméricos, espaços e sublinhados.

Template groups

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

Note:
As permissões de acesso aos grupos de templates são atribuídas na configuração de user group na aba **Template permissions**, 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 *Add*. Seu novo template deve ficar visível na lista de templates. Você também pode usar o filter para encontrar seu template.

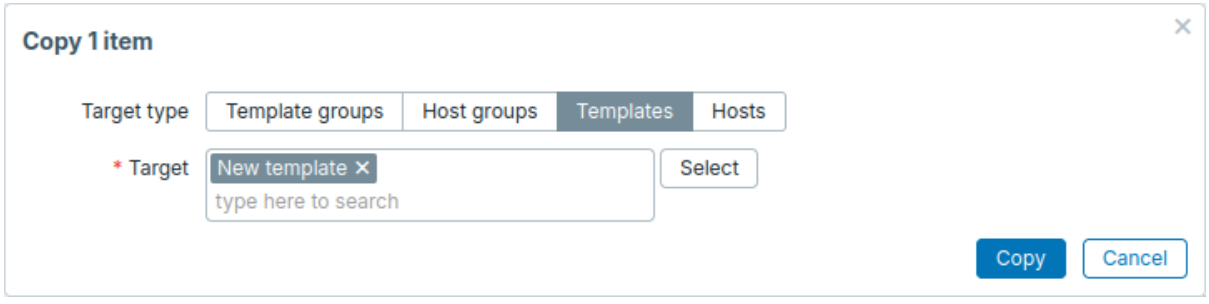


Como você pode ver, o template está lá, mas não contém nada - nenhum item, trigger ou outra entidade.

Adicionando um item ao template

Para adicionar um item ao template:

1. Acesse *Coleta de dados* > *Hosts*.
2. Localize 'New host' na lista e clique no hiperlink *Items*.
3. Marque a caixa de seleção do item 'CPU Load' na lista.
4. Clique em *Copy* abaixo da lista.
5. Selecione a aba *Templates*.
6. Selecione o template para o qual deseja copiar o item.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho. 7. Clique em *Copy*.

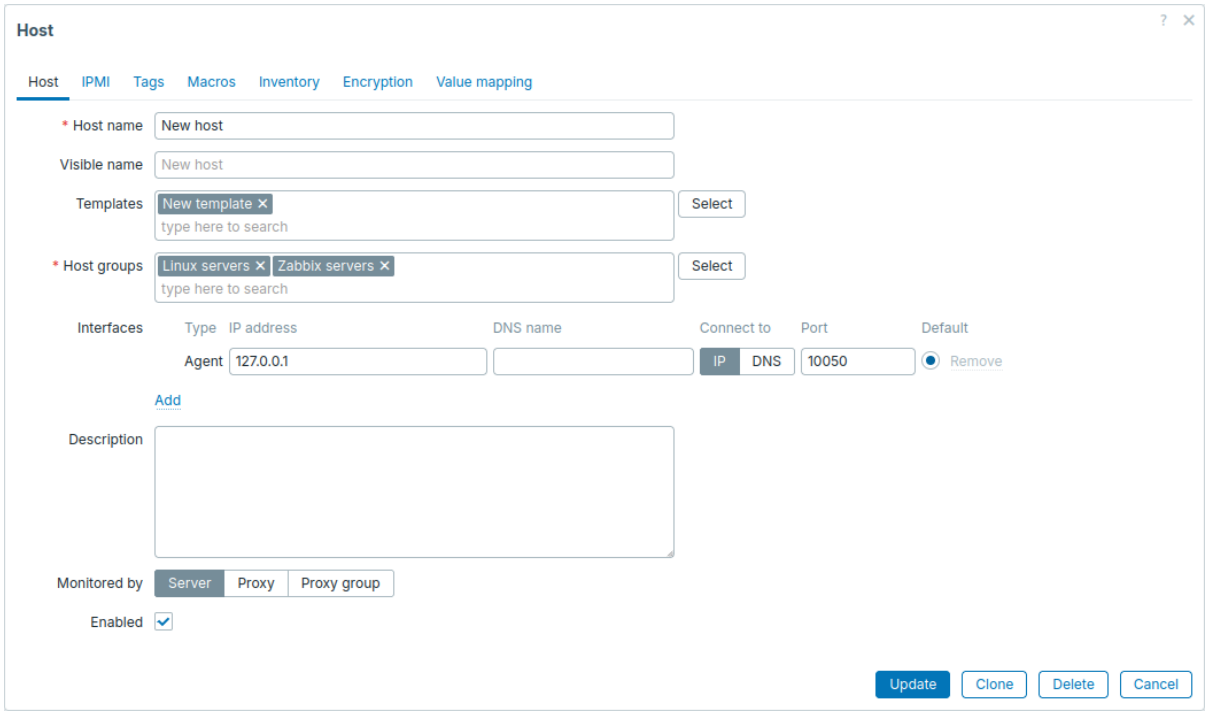
Se você acessar *Coleta de dados* > *Templates* agora, 'New template' deverá conter um novo item.

Por enquanto, vamos nos limitar a um único item, mas, da mesma forma, você pode adicionar quaisquer outros itens, triggers ou outras entidades ao template até que ele contenha um conjunto razoavelmente completo de entidades para uma determinada finalidade (monitoramento do sistema operacional, monitoramento de um único aplicativo).

Vinculando o template ao host

Com o template pronto, basta adicioná-lo a um host. Para fazer isso:

1. Acesse *Coleta de dados* > *Hosts*.
2. Clique em 'Novo host' para abrir o formulário de configuração e localize o campo *Templates*.
3. Comece a digitar *Novo template* no campo *Templates*. O nome do template recém-criado deverá aparecer na lista suspensa.
4. Role a página para baixo para selecioná-lo. Verifique se ele aparece no campo *Templates*.



5. Clique em *Atualizar* no formulário para salvar as alterações.

O template agora está adicionado ao host, com todas as entidades que ele contém.

Dessa forma, ele também pode ser aplicado a qualquer outro host. Quaisquer alterações nos itens, 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 muito rapidamente, você pode vincular o template apropriado a um host, mas tenha em mente 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.

Monitoramento do servidor web Apache

Para começar a monitorar o servidor web Apache, siga este processo:

1. Garanta acesso às estatísticas do Apache.
2. Aplique o template de monitoramento do Apache.
3. Consulte os dados mais recentes.

Para usar este guia, você deve ter o Zabbix server e a interface web instalados. Você deve estar autenticado no Zabbix como o usuário padrão *Admin*.

Acesso às estatísticas do Apache

O objetivo desta etapa é garantir que as estatísticas do servidor web Apache possam ser lidas pelo Zabbix.

No Apache, informações em tempo real sobre o desempenho e a atividade do server são fornecidas pelo módulo interno `mod_status`. Quando habilitado, ele cria uma página acessível pela web (normalmente em `/server-status`) que exibe dados de desempenho e atividade.

Para verificar se o `mod_status` está habilitado para o servidor web Apache, execute:

No Debian/Ubuntu	Em sistemas baseados em RHEL
`apache2ctl -M | grep status`	`httpd -M | grep status`

Se você vir "`status_module (shared)`" na saída, então o módulo de status já está habilitado. Caso contrário, será necessário habilitar o módulo de status.

Aplicar template de monitoramento do Apache

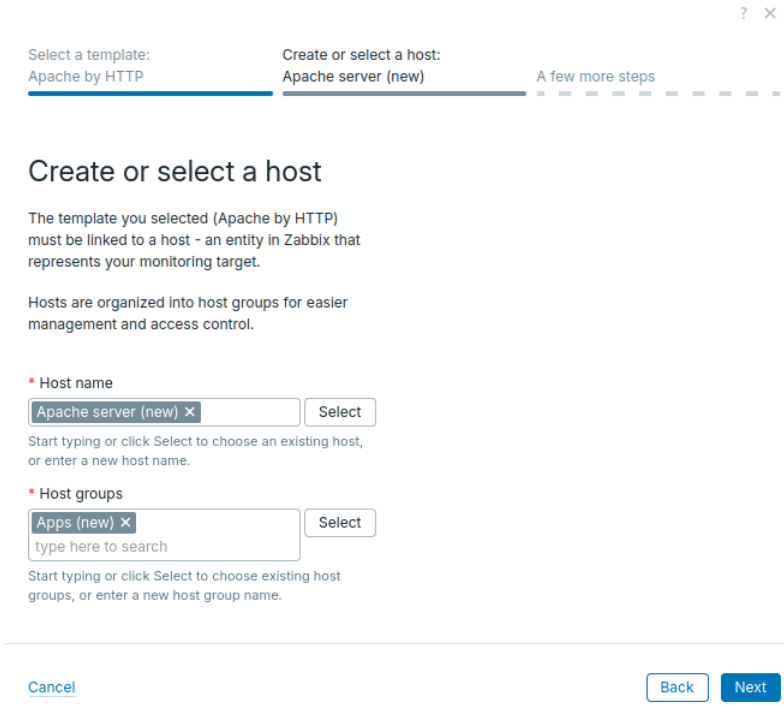
No Zabbix, este template está pronto para uso. Ele contém elementos pré-configurados, prontos para coleta de dados, visualização e análise.

Para aplicar o template:

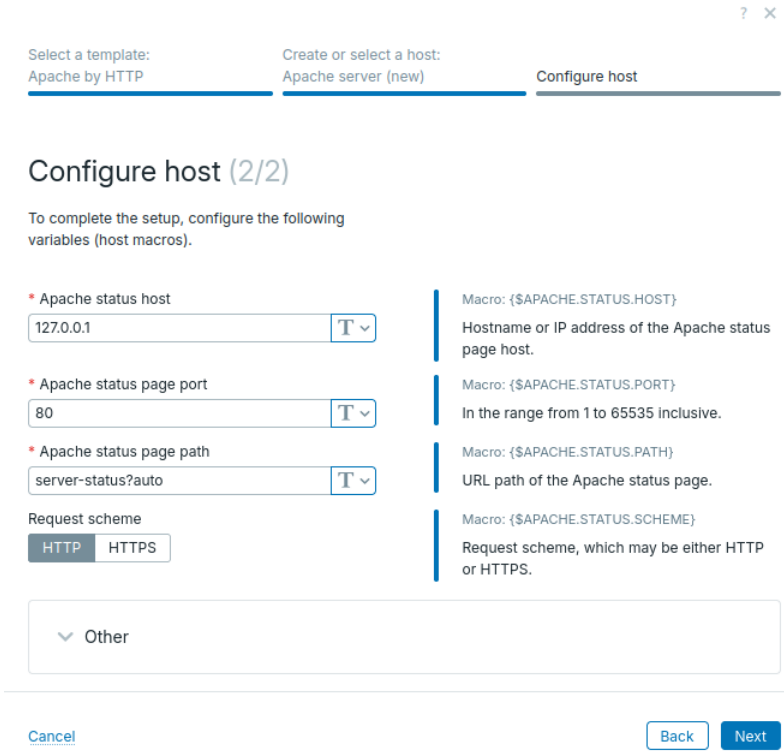
1. Inicie o Host Wizard em *Coleta de dados* > *Hosts*.
2. Selecione o template *Apache by HTTP*.



3. Crie um host Apache e adicione um grupo para ele. Este host virtual conterá dados relacionados ao Apache.



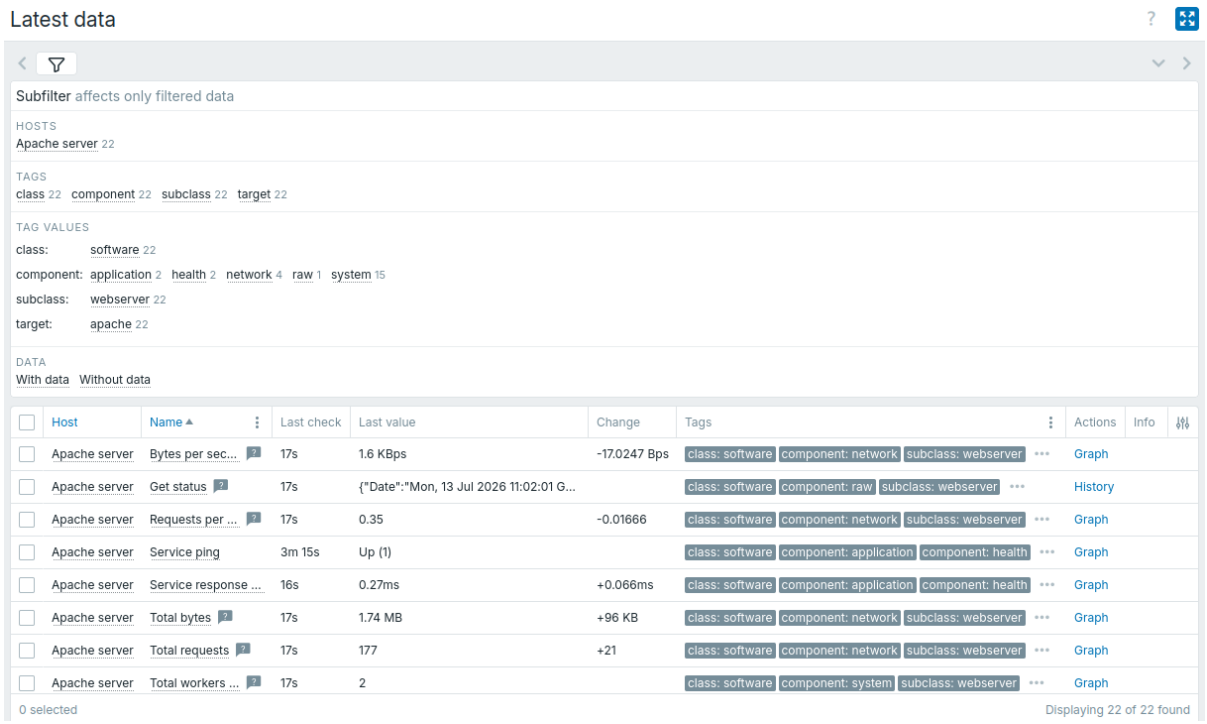
4. Configure o host. Importante: informe o endereço do servidor web Apache (127.0.0.1 em uma máquina local, endereço IP em uma máquina remota).



Você pode deixar as demais configurações com os valores padrão.

5. Para concluir o Host Wizard, clique em *Create* e *Finish*.

Veja os dados mais recentes



Parabéns, o monitoramento do servidor web Apache foi iniciado!

O Zabbix oferece opções de visualização e envia alertas se algum problema surgir. Para configurar alertas por e-mail, consulte Recebendo notificações de problemas.

Monitoramento de servidor Linux

Para começar a monitorar um servidor Linux, siga este processo:

1. Aplique o template de monitoramento Linux.
2. Instale o agent Zabbix no servidor.
3. Verifique os dados mais recentes.

Para começar a monitorar um servidor Linux não local, você precisa saber o endereço IP dele e o endereço IP do server Zabbix.

Um agent Zabbix será usado no servidor Linux para monitorá-lo. As instruções para instalar o agent serão fornecidas no processo.

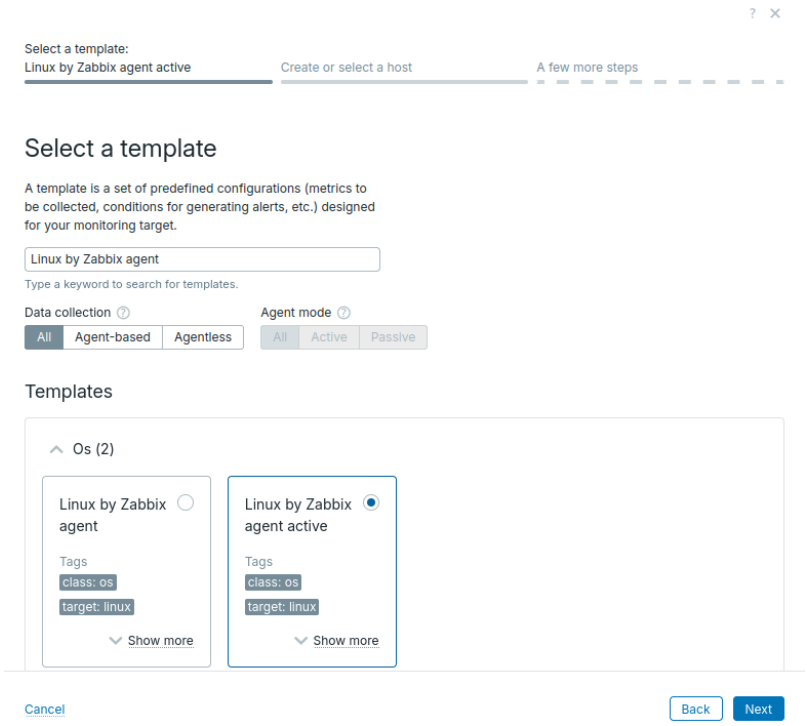
Para usar este guia, você deve ter o server Zabbix e a interface web instalados. Você deve estar autenticado no Zabbix como o usuário padrão *Admin*.

Aplicar template de monitoramento Linux

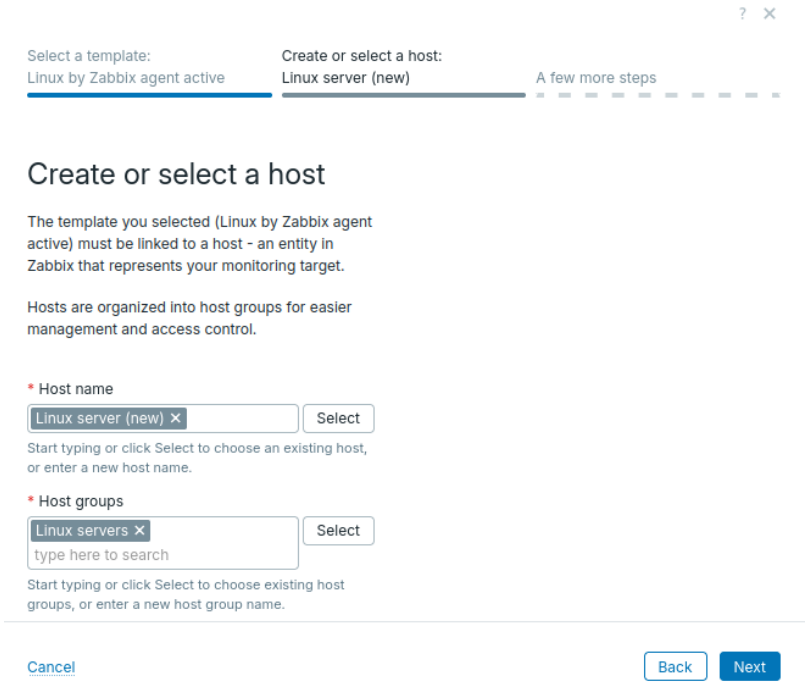
No Zabbix, este template está pronto para uso. Ele contém elementos pré-configurados, prontos para coleta de dados, visualização e análise.

Para aplicar o template:

1. Inicie o Host Wizard em *Data collection* > *Hosts*.
2. Selecione o template *Linux by Zabbix agent active*.



3. Crie um host para o servidor Linux e adicione um grupo para ele. Este host virtual conterá todos os dados relacionados ao monitoramento do servidor Linux.



4. Configure e instale o agent.

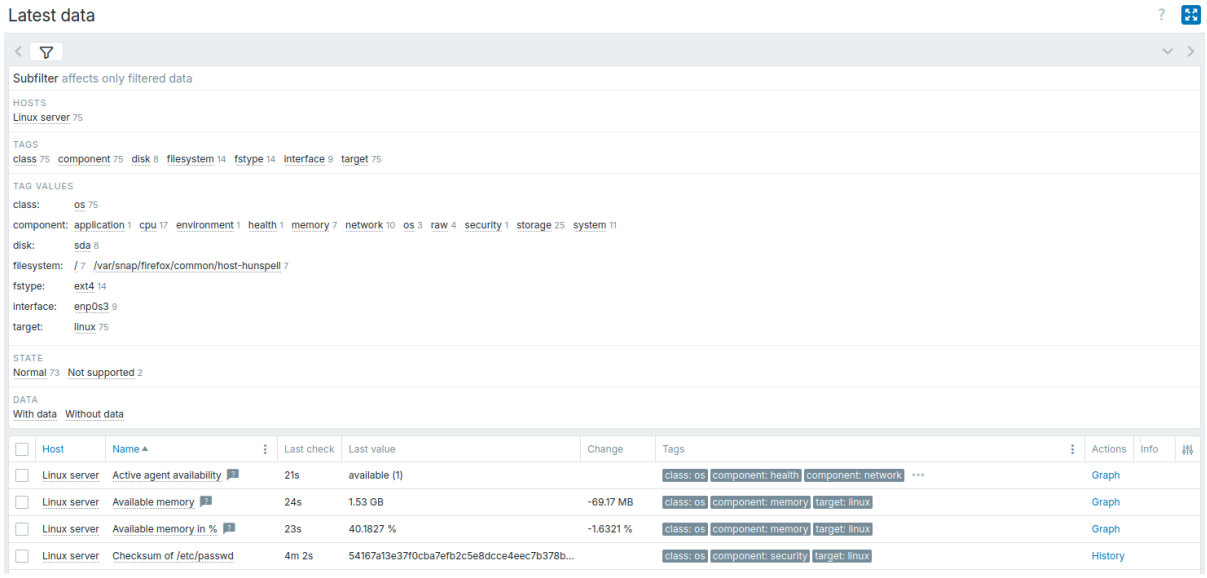


Para instalar o agent, execute o comando fornecido no servidor Linux. Ele instalará o agent e configurará a conexão do agent com o server Zabbix com criptografia.
 Se você já tiver o agent instalado, pode pular esta etapa (clique em *Next*). No entanto, será solicitado que você informe a identidade da chave pré-compartilhada e a chave pré-compartilhada para se conectar ao agent.

Você pode deixar as demais configurações com os valores padrão.

5. Conclua a configuração do host: clique em *Create* e *Finish*.

Veja os dados mais recentes



Parabéns, o monitoramento do server Linux foi iniciado!

O Zabbix oferece opções de visualização e envia alertas se surgirem problemas. Para configurar alertas por email, consulte Receiving problem notification.

Monitoramento de servidor MySQL

Para começar a monitorar um servidor MySQL, siga este processo:

1. Crie um usuário MySQL com privilégios limitados.
2. Aplique o template de monitoramento MySQL.
3. Verifique os dados mais recentes.

Para começar a monitorar um servidor MySQL, você precisa conhecer seu DSN e o endereço IP do server Zabbix.

Um agent Zabbix será usado para monitorar o servidor MySQL. As instruções para instalar o agent serão fornecidas no processo.

Para usar este guia, você deve ter o server Zabbix e a interface web instalados. Você deve estar autenticado no Zabbix como o usuário padrão *Admin*.

Criar usuário MySQL com permissões limitadas

Recomenda-se criar um usuário MySQL com permissões limitadas apenas às necessárias para monitorar o MySQL. Substitua `<password>` por uma senha forte e exclusiva.

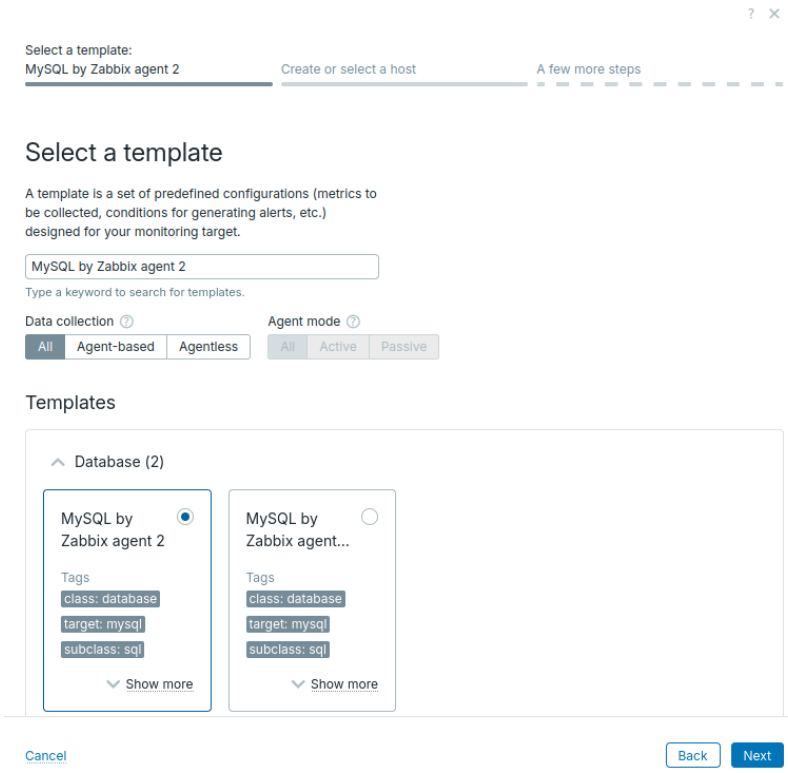
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

Aplicar template de monitoramento MySQL

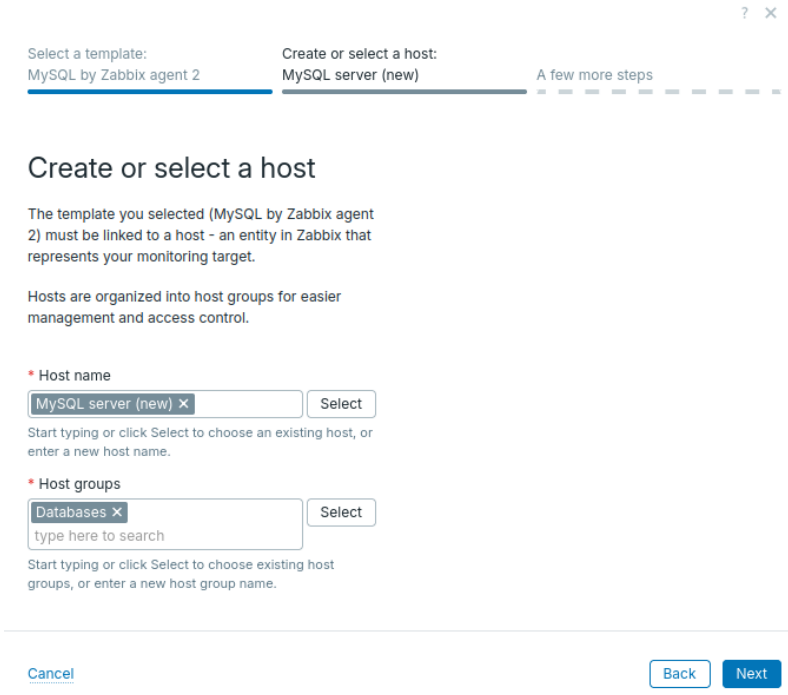
No Zabbix, este template está pronto para uso. Ele contém elementos pré-configurados, prontos para coleta de dados, visualização e análise.

Para aplicar o template:

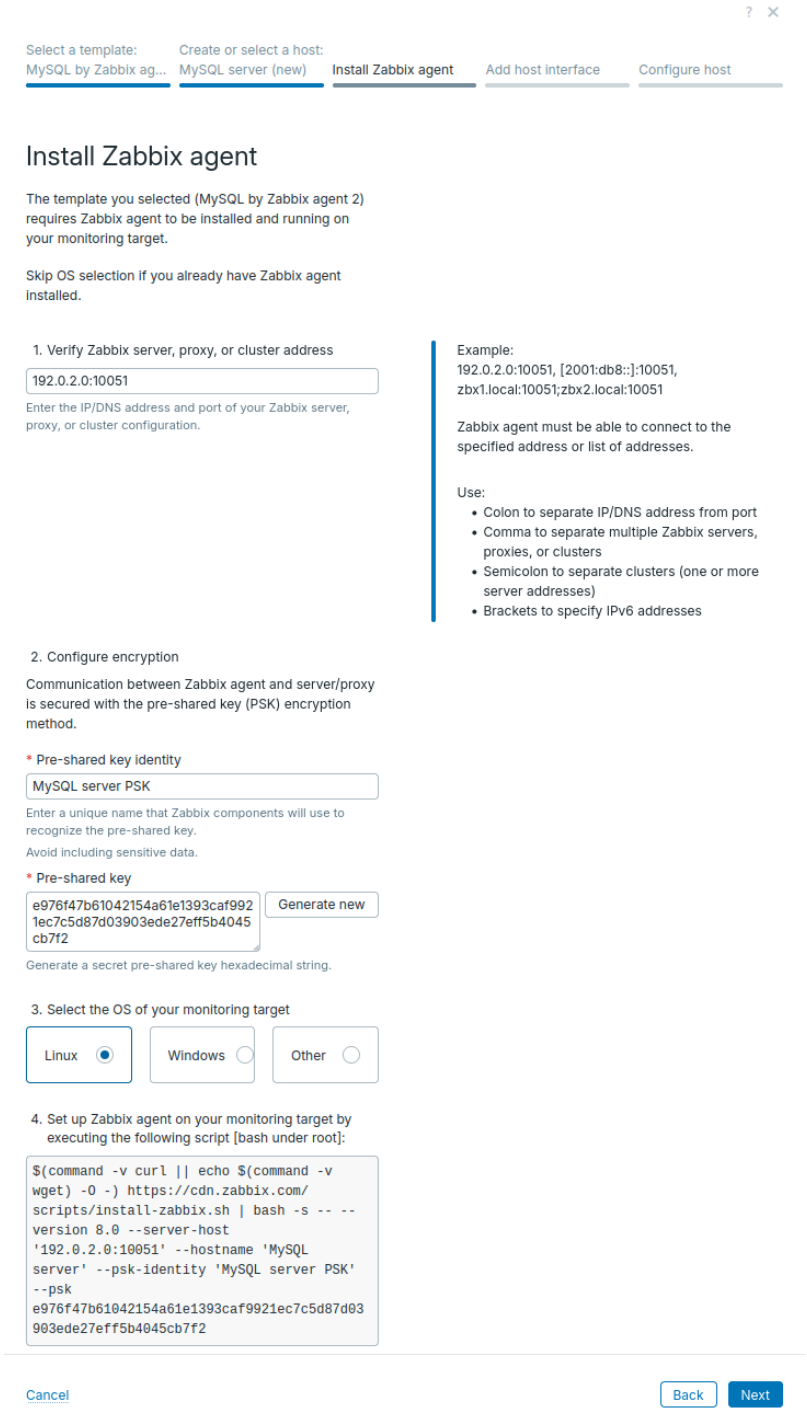
1. Inicie o Host Wizard em *Data collection* > *Hosts*.
2. Selecione o template *MySQL by Zabbix agent 2*.



3. Crie um host de servidor MySQL e adicione um grupo para ele. Esse host virtual conterá todos os dados relacionados ao monitoramento do servidor MySQL.

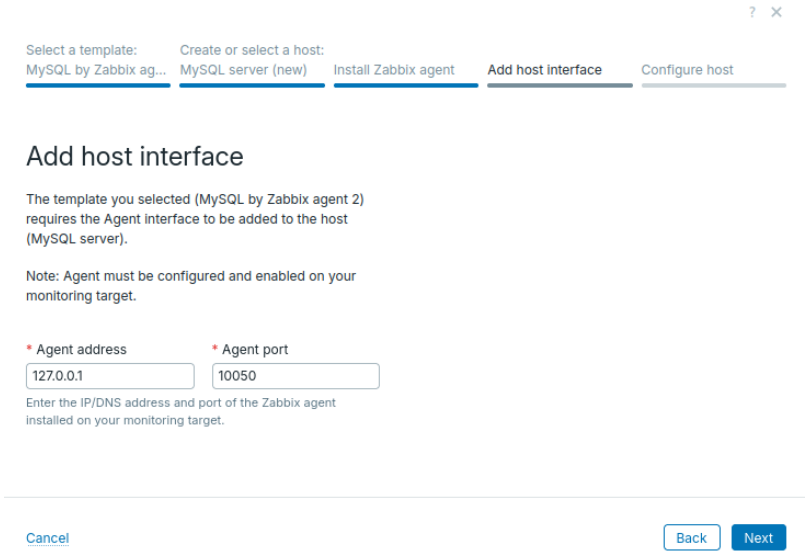


4. Configure e instale o agent 2 (o Zabbix agent 2 fornece monitoramento MySQL pronto para uso).

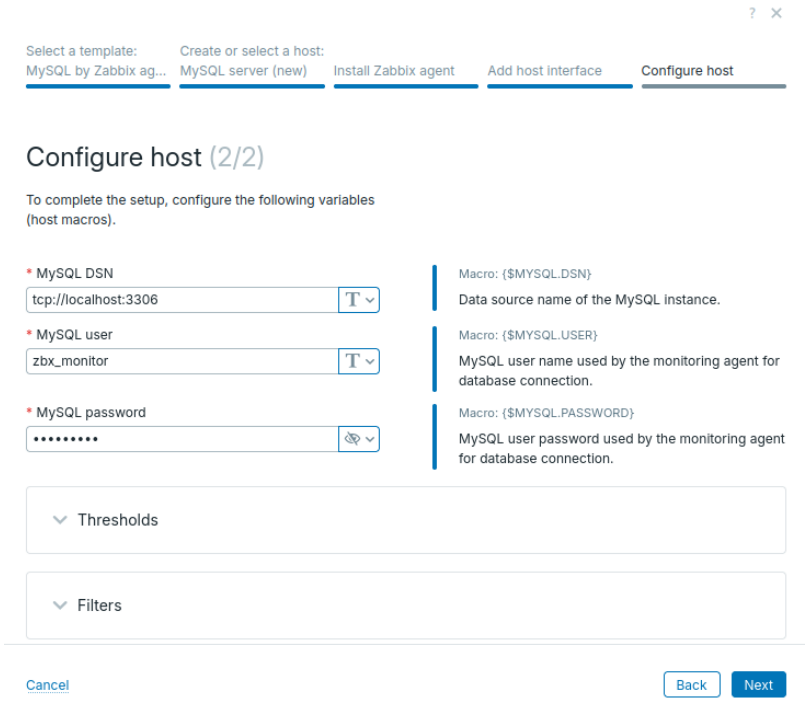


Para instalar o agent, execute o comando fornecido na máquina que hospeda o servidor MySQL. Isso instalará o agent e configurará a conexão do agent com o Zabbix server com criptografia.
 Se você já tiver o agent 2 instalado, pode pular esta etapa (clique em *Next*). No entanto, será solicitado que você informe a identidade da chave pré-compartilhada e a chave pré-compartilhada para conectar ao agent.

5. Informe o endereço IP e a porta do agent (servidor Linux).



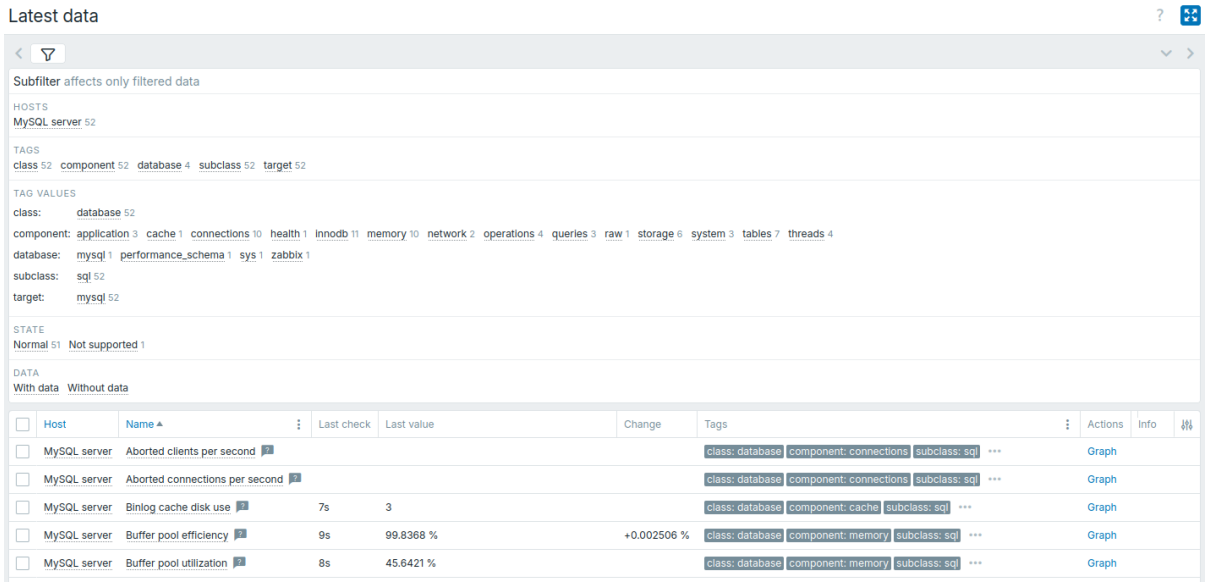
6. Conclua a configuração do host. Importante: será solicitado que você informe o DSN do servidor MySQL, o usuário MySQL e a senha. Forneça o usuário `zbx_monitor` com a senha que você criou.



Você pode deixar as demais configurações com os valores padrão.

6. Para concluir o Host Wizard, clique em *Create* e *Finish*.

Veja os dados mais recentes



Parabéns, o monitoramento do server MySQL foi iniciado!

O Zabbix oferece opções de visualização e envia alertas se surgirem problemas. Para configurar alertas por e-mail, consulte Receiving problem notification.

Mais guias

Esta seção da documentação contém receitas rápidas para configurar o Zabbix para alguns objetivos de monitoramento comumente necessários.

Ela foi projetada pensando no novo usuário do Zabbix e pode ser usada como um guia de navegação por outras seções da documentação que contêm as informações necessárias para resolver a tarefa.

Os seguintes guias de referência rápida estão disponíveis:

- Monitor Linux with Zabbix agent
- Monitor Windows with Zabbix agent
- Monitor Apache via HTTP
- Monitor MySQL with Zabbix agent 2
- Monitor VMware with Zabbix
- Monitor network traffic with Zabbix
- Monitor network traffic with Zabbix using active checks
- Monitor websites with Browser items
- Monitor website certificates with Zabbix agent 2 (passive)
- Monitor a network switch or router with Zabbix
- Monitor Windows event log using active checks

Monitore Linux com o agent Zabbix

Introdução

Esta página orienta você pelas 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 este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico da sua máquina Linux. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais avançada, consulte a seção Configuration do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Instalar o Zabbix agent

O Zabbix agent é o processo responsável por coletar dados.

Verifique a versão do seu Zabbix server:

```
zabbix_server -V
```

Instale o Zabbix agent da mesma versão (recomendado) na máquina Linux que você deseja monitorar. Com base nas suas necessidades de monitoramento, pode ser a mesma máquina em que o Zabbix server está instalado ou uma máquina completamente diferente.

Escolha o método de instalação mais adequado:

- Executar como um contêiner Docker - veja a lista de imagens disponíveis no repositório Docker do Zabbix.
- Instalar a partir dos pacotes do Zabbix (disponíveis para Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Compilar a partir do código-fonte.

Configurar o Zabbix para monitoramento

O agent do Zabbix pode coletar métricas em modo 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, uso de CPU) e o agent do Zabbix envia o resultado de volta ao server. As verificações ativas exigem um processamento mais complexo. Primeiro, o agent deve recuperar do(s) server(s) uma lista de items para processamento independente e, em seguida, enviar os dados em lote de volta. Veja Passive and active agent checks para mais informações.

Os templates de monitoramento fornecidos pelo Zabbix geralmente oferecem duas alternativas - um template para Zabbix agent e um template para Zabbix agent (active). Com a primeira opção, o agent coletará métricas em modo passivo. Esses templates fornecerão resultados de monitoramento idênticos, mas usando protocolos de comunicação diferentes.

A configuração adicional do Zabbix depende de você selecionar um template para verificações de Zabbix agent ativas ou passivas.

Verificações passivas

Zabbix agent

1. Abra o arquivo de configuração do agent na máquina em que 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=192.0.2.1:10051
```

O Zabbix agent usará esse endereço para aceitar conexões de entrada somente dos Zabbix servers ou proxies especificados para recuperar dados.

3. Reinicie o Zabbix agent.

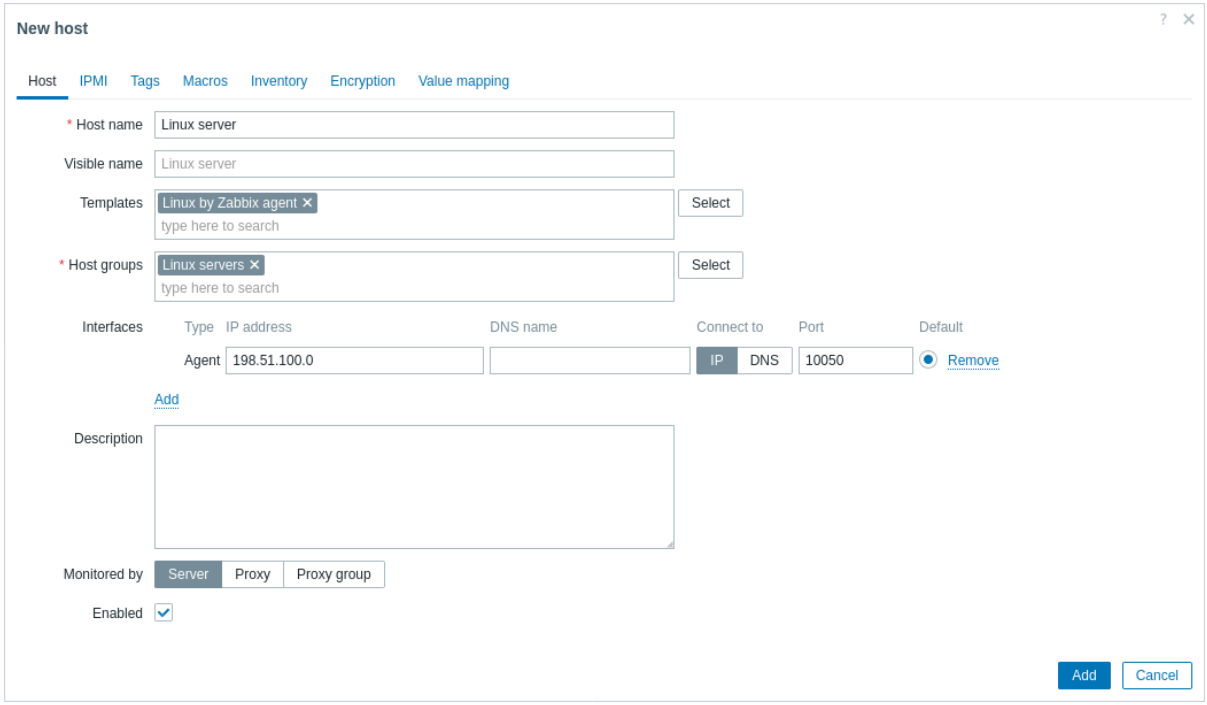
```
systemctl restart zabbix-agent
```

frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

- No campo *Host name*, informe um nome de host (por exemplo, ”Linux server”).
- No campo *Templates*, digite ou selecione o template ”Linux by Zabbix agent” que será vinculado ao host.
- No campo *Host groups*, digite ou selecione um grupo de hosts (por exemplo, ”Linux servers”).
- 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 *Add* para adicionar o host. Esse host representará a máquina Linux monitorada.

Verificações ativas

Zabbix agent

1. Abra o arquivo de configuração do agent na máquina em que 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 ServerActive. Por exemplo:

```
ServerActive=192.0.2.1:10051
```

O Zabbix agent usará esse endereço para se conectar à porta trapper do Zabbix server (padrão: 10051) e solicitar os dados de configuração para verificações ativas.

3. Defina o parâmetro Hostname, que deve corresponder ao nome do host que será definido no Zabbix frontend. Em nosso exemplo, ele 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 no server. Especificamente, o agent inicia uma conexão com o server e se identifica usando o valor de Hostname. Em seguida, o server fornece a configuração de monitoramento desse 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.

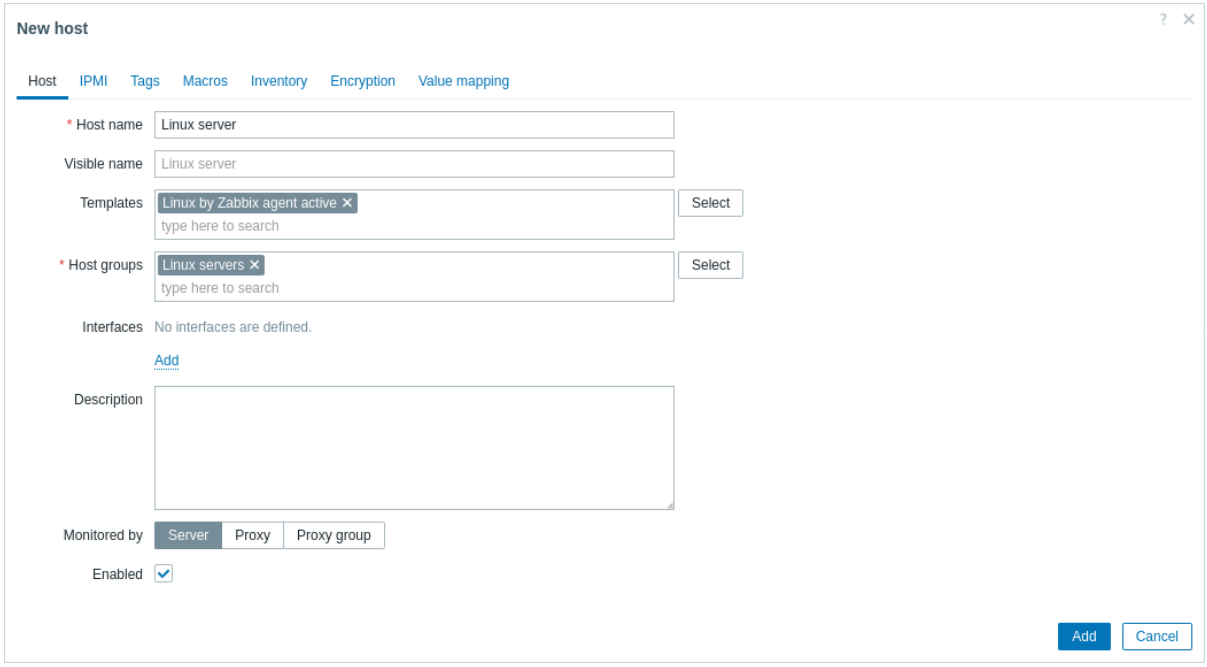
```
systemctl restart zabbix-agent
```

Zabbix frontend

1. Faça login no Zabbix frontend.

2. Crie um host na interface web do Zabbix.

- No campo *Host name*, informe 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 *Host groups*, digite ou selecione um grupo de hosts (por exemplo, ”Linux servers”).
- Deixe *Interfaces* indefinido; uma interface não é necessária para verificações ativas, porque o agent inicia a conexão com o server em vez de aguardar uma conexão do server.

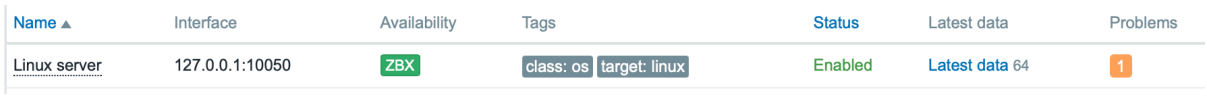


3. Clique em *Add* para adicionar o host. Este host representará a máquina Linux monitorada.

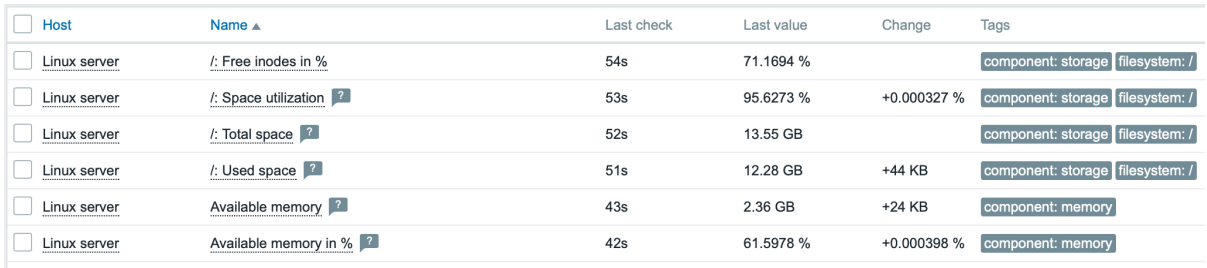
Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando sua máquina Linux.

Para ver as métricas coletadas, abra a seção de menu *Monitoring* > *Hosts* menu section e clique em *Latest data* ao lado do host.



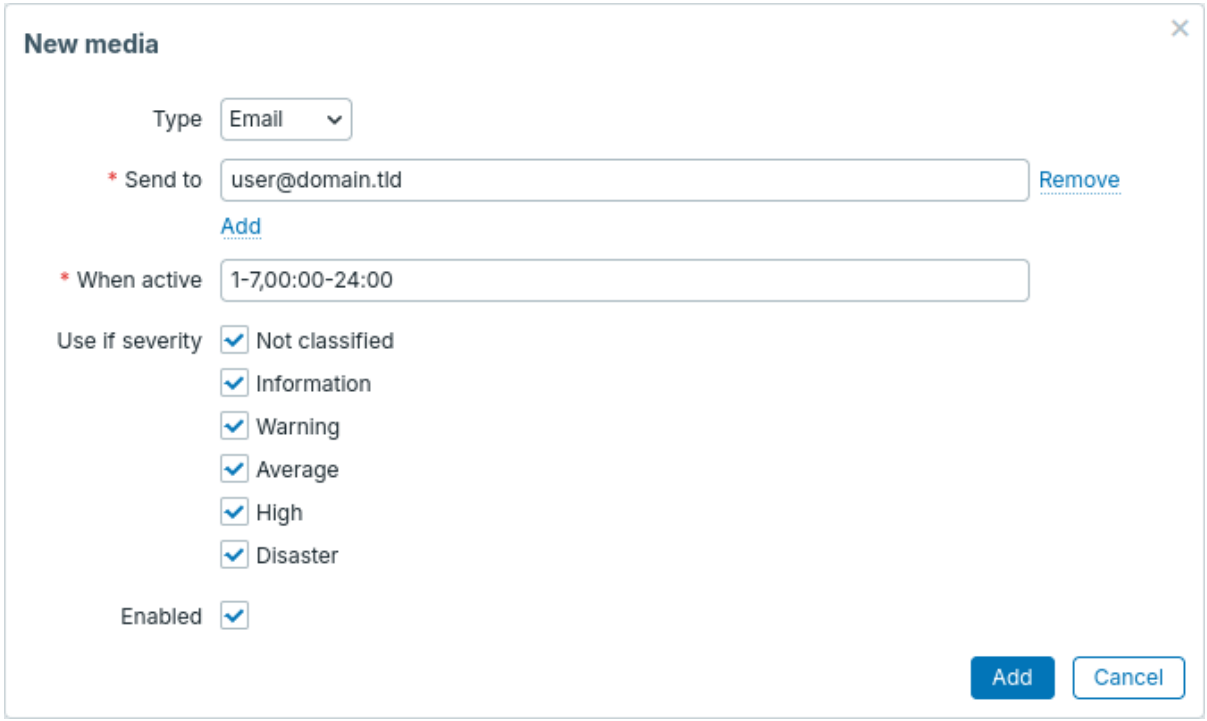
Essa ação abrirá uma lista de todas as métricas mais recentes coletadas do host do server Linux.



Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos.
Este guia fornece etapas de configuração para o envio de alertas por email.

1. Vá para *User settings* > *Profile*, mude para a aba *Media* e
adicione seu email.



2. Siga o guia para Receiving problem notification.

Na próxima vez, quando o Zabbix detectar um problema, você deverá receber um alerta por email.

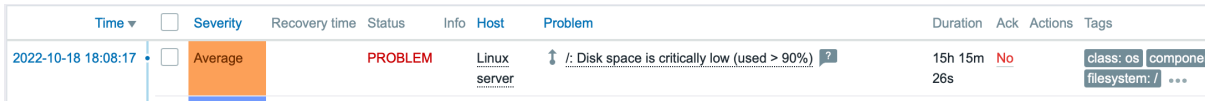
Teste sua configuração

No Linux, você pode simular uma alta carga de CPU e, como resultado, receber um alerta de problema executando:

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

Talvez seja necessário executar vários processos md5sum para que a carga de CPU ultrapasse o limite.

Quando o Zabbix detectar o problema, ele aparecerá na seção *Monitoring* > *Problems*.



Se os alertas estiverem configurados, você também receberá a notificação do problema.

Veja também:

- Criando um item - como começar a monitorar métricas adicionais (monitoramento personalizado sem templates).
- Itens do Zabbix agent, Itens do Zabbix agent para Windows - lista completa de métricas que você pode monitorar usando o Zabbix agent no Windows.
- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, depois, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).
- Instalação a partir de pacotes - como instalar componentes do Zabbix usando os pacotes oficiais RPM e DEB para várias distribuições Linux, garantindo acesso aos recursos mais recentes e às correções de bugs.

Monitore Windows com o agent Zabbix

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico de máquinas Windows com o Zabbix.

Para quem este guia se destina

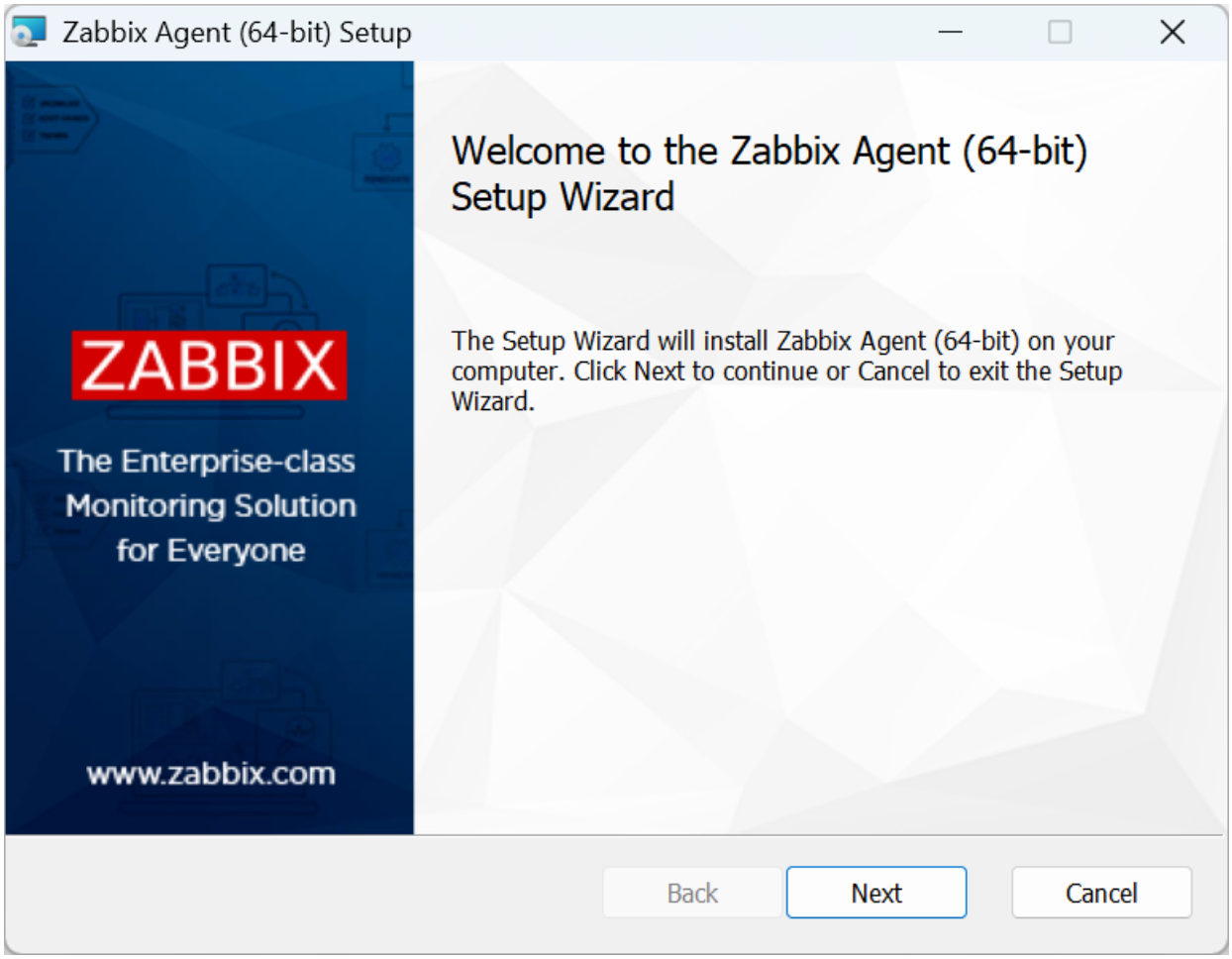
Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico da sua máquina Windows. Se você estiver procurando opções de personalização detalhadas ou precisar de uma configuração mais avançada, consulte a seção Configuration 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 Zabbix agent

O Zabbix agent é o processo responsável por coletar dados. Você precisa instalá-lo na máquina Windows que deseja monitorar. Siga as instruções de instalação do Zabbix agent para Windows.



Configure o Zabbix para monitoramento

O agent do Zabbix pode coletar métricas em modo 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, uso de CPU) e o agent do Zabbix envia o resultado de volta ao server. As verificações ativas exigem um processamento mais complexo. Primeiro, o agent deve recuperar do(s) server(s) uma lista de items para processamento independente e, em seguida, enviar os dados em lote de volta. Veja Passive and active agent checks para mais informações.

Os templates de monitoramento fornecidos pelo Zabbix geralmente oferecem duas alternativas - um template para Zabbix agent e um template para Zabbix agent (active). Com a primeira opção, o agent coletará métricas em modo passivo. Esses templates fornecerão resultados de monitoramento idênticos, mas usando protocolos de comunicação diferentes.

A configuração adicional do Zabbix depende de você selecionar um template para verificações de Zabbix agent ativas ou passivas.

Verificações passivas

frontend do Zabbix

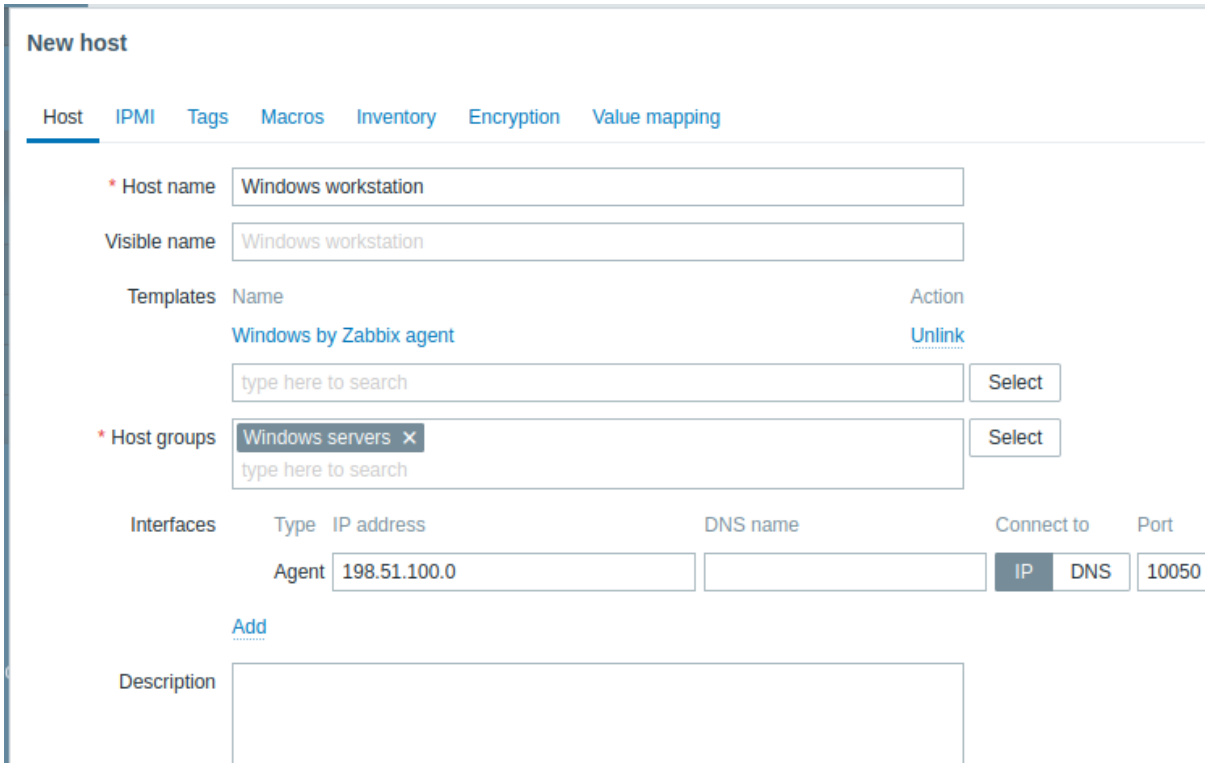
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

Esse 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*.



Zabbix agent

Para verificações passivas, o Zabbix agent precisa saber o endereço IP ou o nome DNS do Zabbix server. Se você forneceu as informações corretas durante a instalação do agent, o arquivo de configuração já foi atualizado. Caso contrário, será necessário especificá-las manualmente. Abra o arquivo de configuração do agent e adicione o IP/DNS do seu Zabbix server ao parâmetro `Server`.

Exemplo:

```
Server=192.0.2.22
```

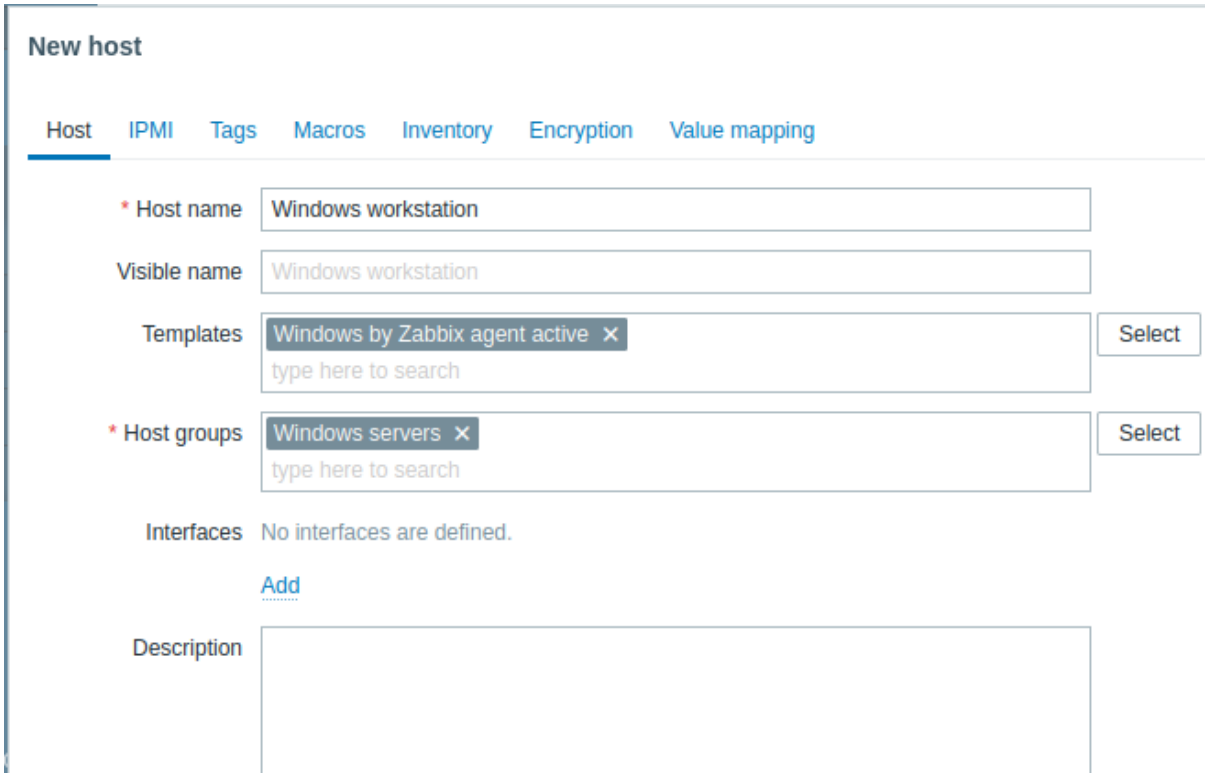
Verificações ativas

frontend do Zabbix

1. Faça login no frontend do Zabbix.
2. Crie um host na interface web do Zabbix.

Esse host representará 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 o nome DNS da máquina do seu Zabbix server ao parâmetro *ServerActive* (pode já estar preenchido se você o tiver informado durante a configuração do Zabbix agent).

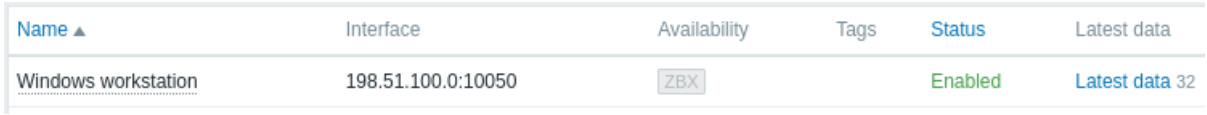
Exemplo:

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

Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando sua máquina Windows.

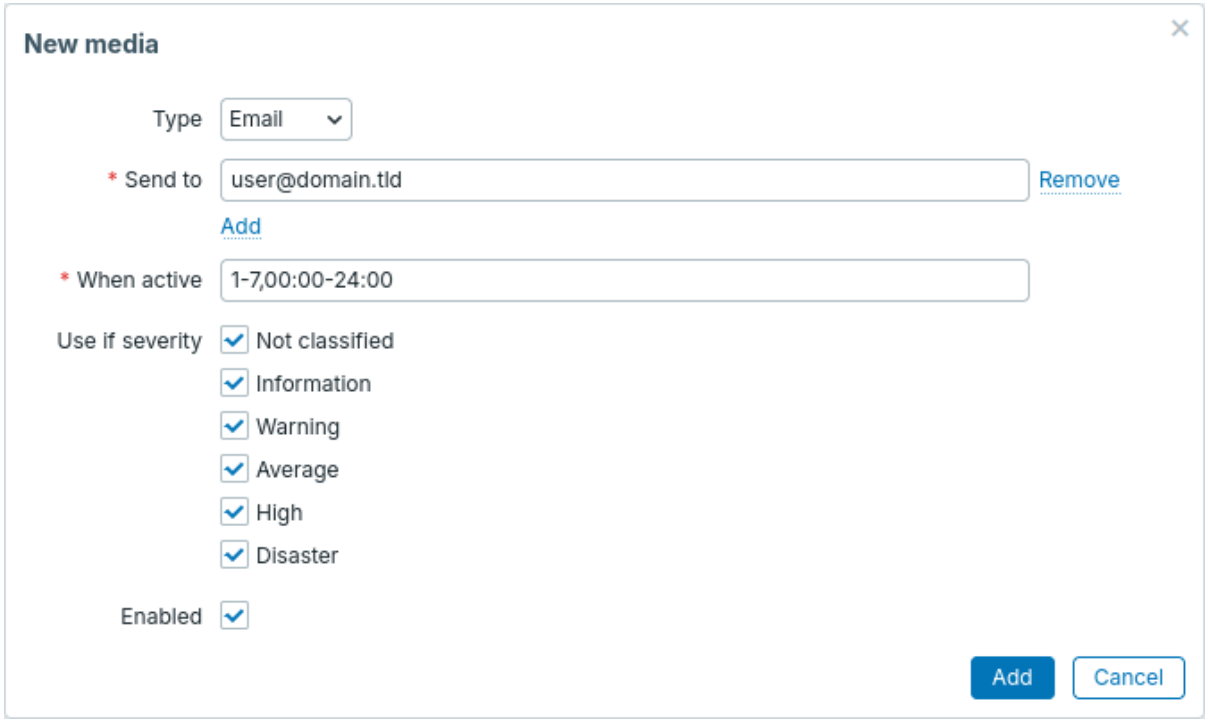
Para visualizar as métricas coletadas, abra a seção de menu *Monitoring* > *Hosts* menu section e clique em *Latest data* ao lado do host.



Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos. Este guia fornece etapas de configuração para o envio de alertas por email.

1. Vá para *User settings* > *Profile*, mude para a aba *Media* e adicione seu email.



2. Siga o guia para Receiving problem notification.

Na próxima vez em que o Zabbix detectar um problema, você deverá receber um alerta por email.

Note:
No Windows, você pode usar o utilitário CpuStres para simular uma alta carga de CPU e, como resultado, receber um alerta de problema.

Veja também:

- Creating an item - como começar a monitorar métricas adicionais (monitoramento personalizado sem templates).
- Zabbix agent items, Zabbix agent items for Windows - lista completa de métricas que você pode monitorar usando o agent do Zabbix no Windows.
- Problem escalations - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, depois, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).

Monitorar 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 este guia é destinado

Este guia foi elaborado 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 estiver procurando opções de personalização avançada ou precisar de uma configuração mais sofisticada, consulte a seção Configuration 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 um sistema baseado em RHEL, execute:

```
httpd -v
```

No Debian/Ubuntu, execute:

```
apache2 -v
```

2. Certifique-se de que o módulo Status esteja habilitado na sua instância do Apache.

Em um sistema baseado em RHEL, execute:

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

No Debian/Ubuntu, execute:

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

Se você não vir status_module na lista, habilite o módulo executando:

Em um sistema baseado em RHEL, execute:

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

No Debian/Ubuntu, execute:

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

3. Edite o arquivo de configuração do Apache para permitir acesso aos relatórios de status a partir do IP do server Zabbix.

Em um sistema baseado em RHEL: `/etc/httpd/conf.modules.d/status.conf`:

```
sudo vi /etc/httpd/conf.modules.d/status.conf
```

No Debian/Ubuntu: `/etc/apache2/mods-enabled/status.conf`:

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

Adicione as seguintes linhas ao arquivo (**substitua 198.51.100.255** pelo endereço IP do seu server Zabbix):

- 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 um sistema baseado em RHEL, execute:

```
sudo systemctl restart httpd
```

No Debian/Ubuntu, execute:

```
sudo systemctl restart apache2
```

5. Para verificar se tudo está configurado corretamente, execute (**substitua 198.51.100.255** pelo endereço IP do seu server Zabbix):

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

A resposta deve conter estatísticas do servidor web Apache.

Configure o Zabbix para monitoramento

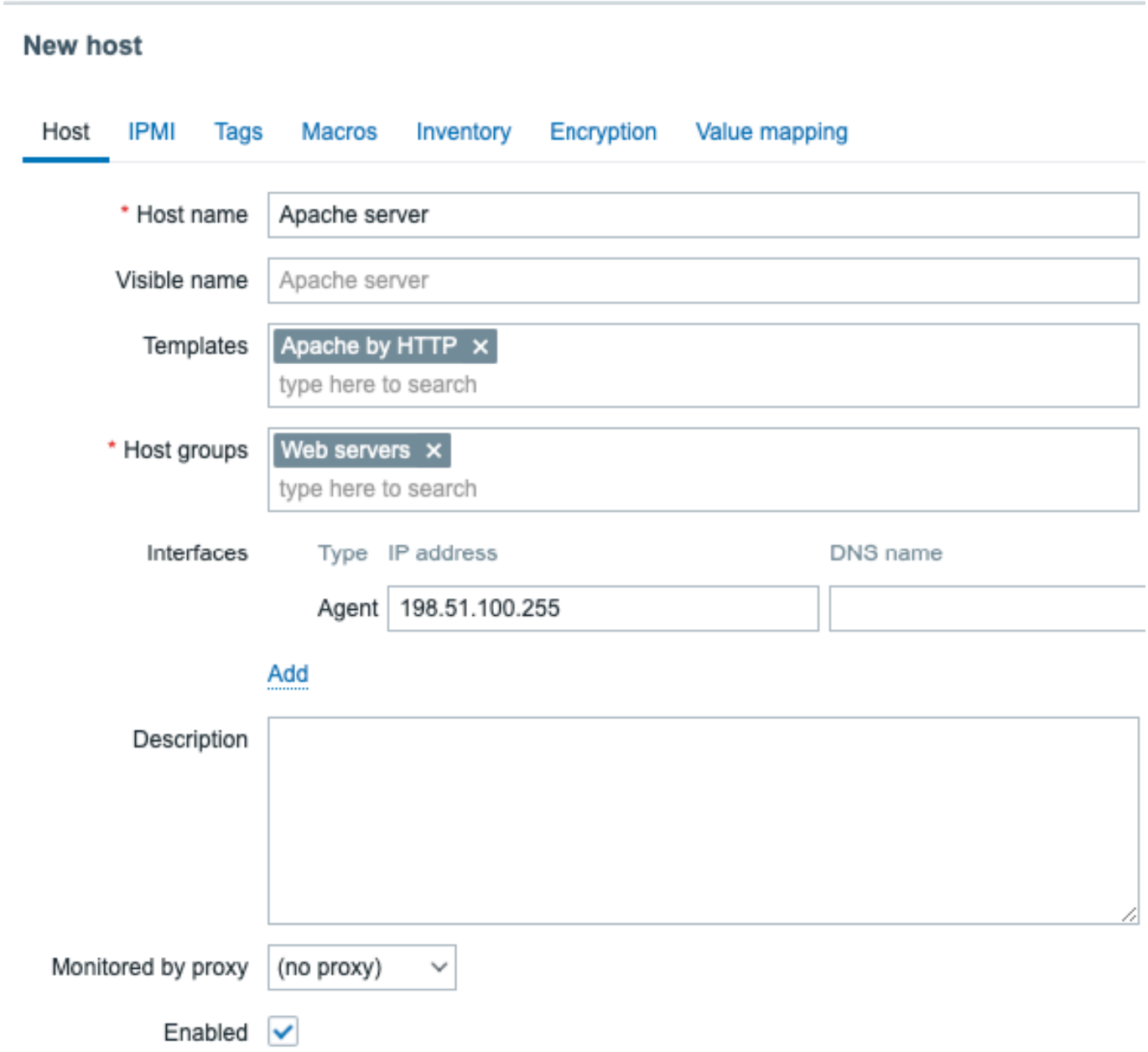
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

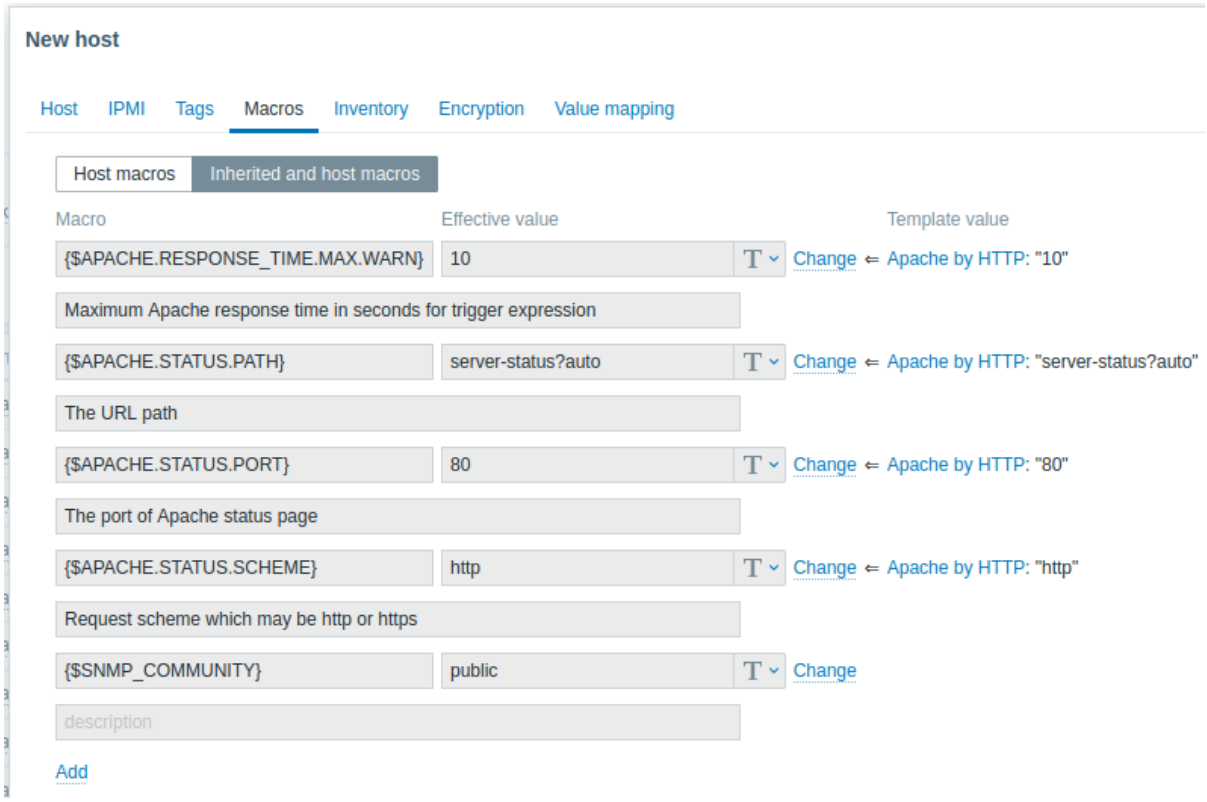
Este host representará seu servidor Apache.

3. No parâmetro *Interfaces*, adicione a interface *Agent* e especifique o endereço IP da sua instância do Apache. **Você não precisa instalar o agent do Zabbix na máquina**, a interface será usada apenas para resolver a macro {HOST.CONN}. Essa macro é usada nos itens do template para localizar a instância do Apache.

4. No parâmetro *Templates*, digite ou selecione *Apache by HTTP*.



5. Acesse a aba **Macros** e selecione o modo *Inherited and host macros*. 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 diferentes.



Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando seu servidor web Apache.

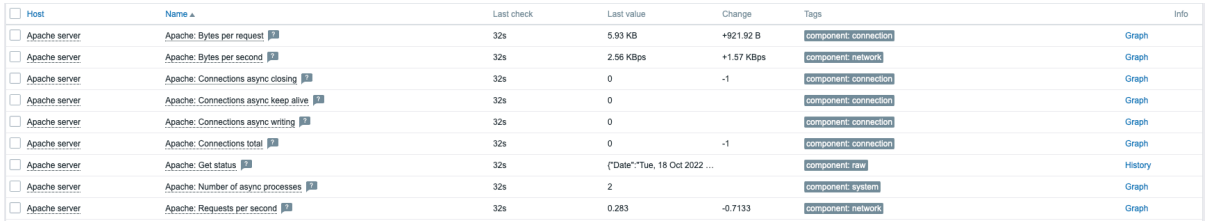
Para ver as métricas coletadas, abra a seção de menu *Monitoring* > *Hosts* menu section e clique em *Dashboards* ao lado do host.



Essa ação levará você ao dashboard do host com as métricas mais importantes coletadas da página /server-status do Apache.



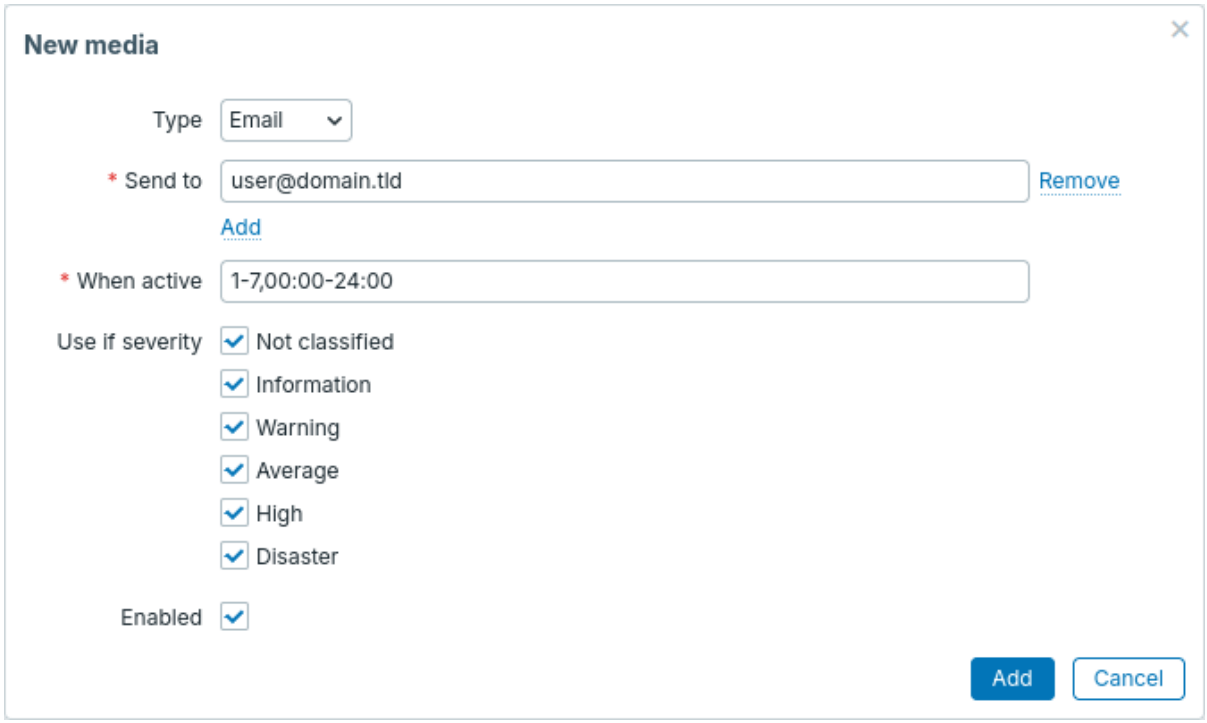
Como alternativa, em *Monitoring* > *Hosts*, você pode clicar em *Latest data* para ver todas as métricas coletadas mais recentes em uma lista.



Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos.
Este guia fornece as etapas de configuração para o envio de alertas por email.

1. Vá para *User settings* > *Profile*, mude para a aba *Media* e adicione seu email.



2. Siga o guia para Receiving problem notification.

Na próxima vez, quando o Zabbix detectar um problema, você deverá receber um alerta por email.

Teste sua configuração

Para simular um problema real e receber um alerta de teste de problema:

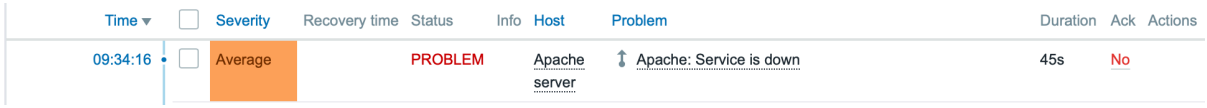
1. Abra a configuração do host *Apache server* no Zabbix.

2. Acesse a aba Macros e selecione *Inherited and host macros*.

3. Clique em *Change* ao lado da macro {$APACHE.STATUS.PORT} e defina uma porta diferente.

4. Clique em *Update* para salvar a configuração do host.

5. Em alguns minutos, o Zabbix detectará o problema *Apache service is down*, porque agora ele não consegue se conectar à instância. Ele aparecerá na seção *Monitoring* > *Problems*.



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 continuar monitorando o Apache.

Veja também:

- Endurecimento do web server - configurações recomendadas para maior segurança do web server.
- Criando um item - como começar a monitorar métricas adicionais.
- Itens HTTP - como monitorar métricas personalizadas usando o HTTP agent.
- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, depois, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).

Monitore MySQL com o Zabbix agent 2

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico de um server MySQL.

Para monitorar um server MySQL, há várias abordagens: Zabbix agent, Zabbix agent 2 ou o padrão Open Database Connectivity (ODBC). O foco principal deste guia é o monitoramento de um server MySQL com o Zabbix agent 2, que é a abordagem **recomendada** devido à sua configuração simples em diferentes cenários. No entanto, esta página também oferece instruções para as outras abordagens, então sinta-se à vontade para escolher a que melhor atende aos seus requisitos.

Para quem este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de um server MySQL. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais avançada, consulte a seção Configuration do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent 2 de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, algumas etapas deste guia podem diferir ligeiramente. Este guia é baseado em um ambiente executando Ubuntu.

Criar usuário MySQL

Para monitorar um server MySQL, o Zabbix requer acesso a ele e aos seus processos. Sua instalação do MySQL já possui um usuário com o nível de acesso necessário (o usuário "zabbix" que foi criado durante a instalação do Zabbix), no entanto, esse usuário tem mais privilégios do que o necessário para o monitoramento simples (privilégios para DROP databases, DELETE entries from tables, etc.). Portanto, é necessário criar um usuário MySQL com a finalidade de *apenas* monitorar o server MySQL.

1. Conecte-se ao cliente MySQL, crie um usuário "zbx_monitor" (substitua <*password*> do usuário "zbx_monitor" por uma senha de sua escolha), e GRANT os privilégios necessários ao usuário:

```
mysql -u root -p
# Enter password:

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

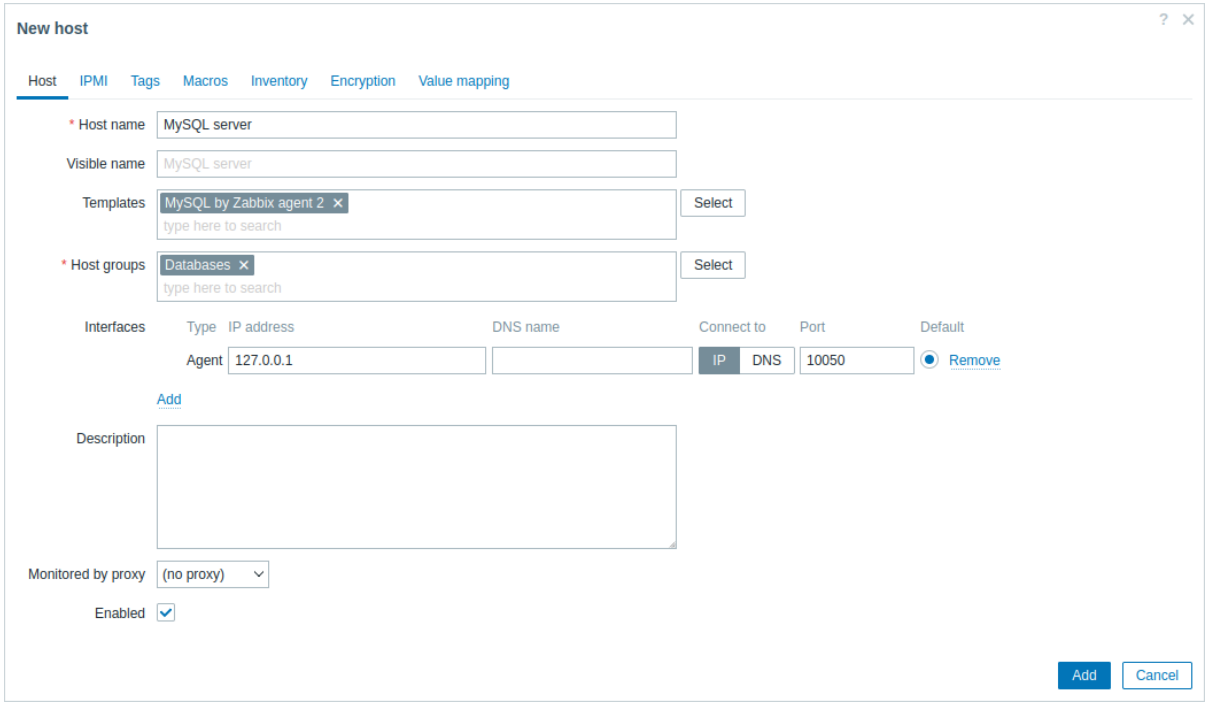
Depois que o usuário for criado, você pode passar para a próxima etapa.

Configurar o frontend do Zabbix

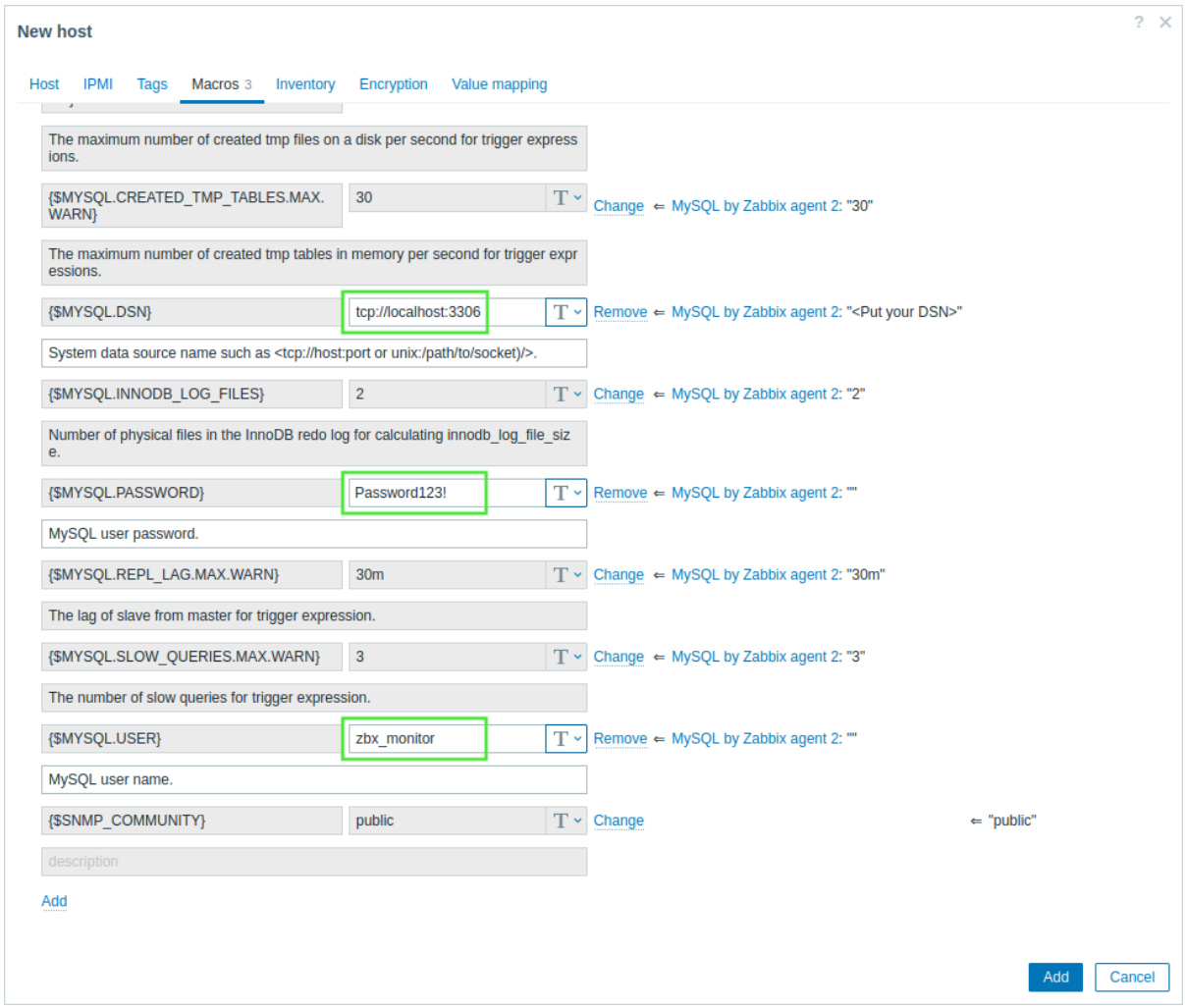
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

   - No campo *Host name*, informe um nome de host (por exemplo, "MySQL server").
   - No campo *Templates*, digite ou selecione o template "MySQL by Zabbix agent 2" que será vinculado ao host.
   - No campo *Host groups*, digite ou selecione um grupo de hosts (por exemplo, "Databases").
   - No campo *Interfaces*, adicione uma interface do tipo "Agent" e especifique o endereço IP do seu servidor MySQL. Este guia usa "127.0.0.1" (localhost) para monitorar um servidor MySQL instalado na mesma máquina que o server Zabbix e o agent 2 do Zabbix.



- Na aba *Macros*, alterne para *Inherited and host macros*, procure as seguintes macros e clique em *Change* ao lado do valor da macro para atualizá-lo:
  - {$MYSQL.DSN} - defina a fonte de dados do servidor MySQL (a string de conexão de uma sessão nomeada do arquivo de configuração do plugin MySQL do agent 2 do Zabbix). Este guia usa a fonte de dados padrão ”tcp://localhost:3306” para monitorar um servidor MySQL instalado na mesma máquina que o server Zabbix e o agent 2 do Zabbix.
  - {$MYSQL.PASSWORD} - defina a senha do usuário MySQL criado anteriormente ”zbx_monitor”.
  - {$MYSQL.USER} - defina o nome do usuário MySQL criado anteriormente ”zbx_monitor”.

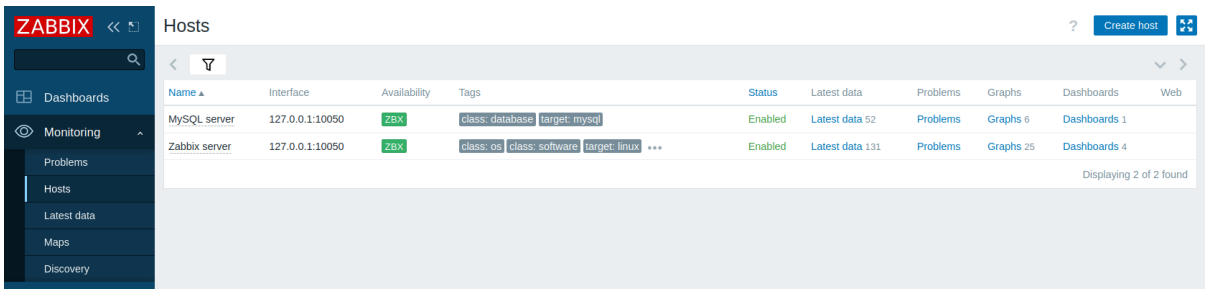


3. Clique em *Add* para adicionar o host. Este host representará o seu servidor MySQL.

Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando seu servidor MySQL.

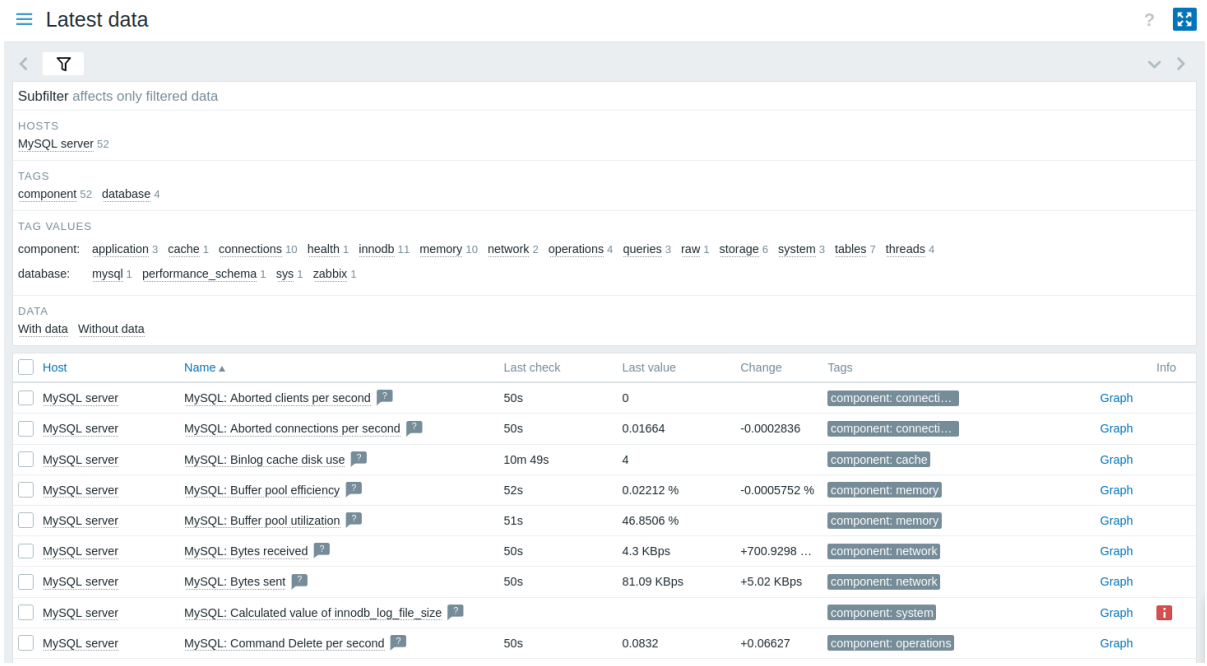
Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoring > Hosts* e clique em *Dashboards* ao lado do host.



Essa ação levará você ao dashboard do host (configurado no nível do template) com as métricas mais importantes coletadas do servidor MySQL.



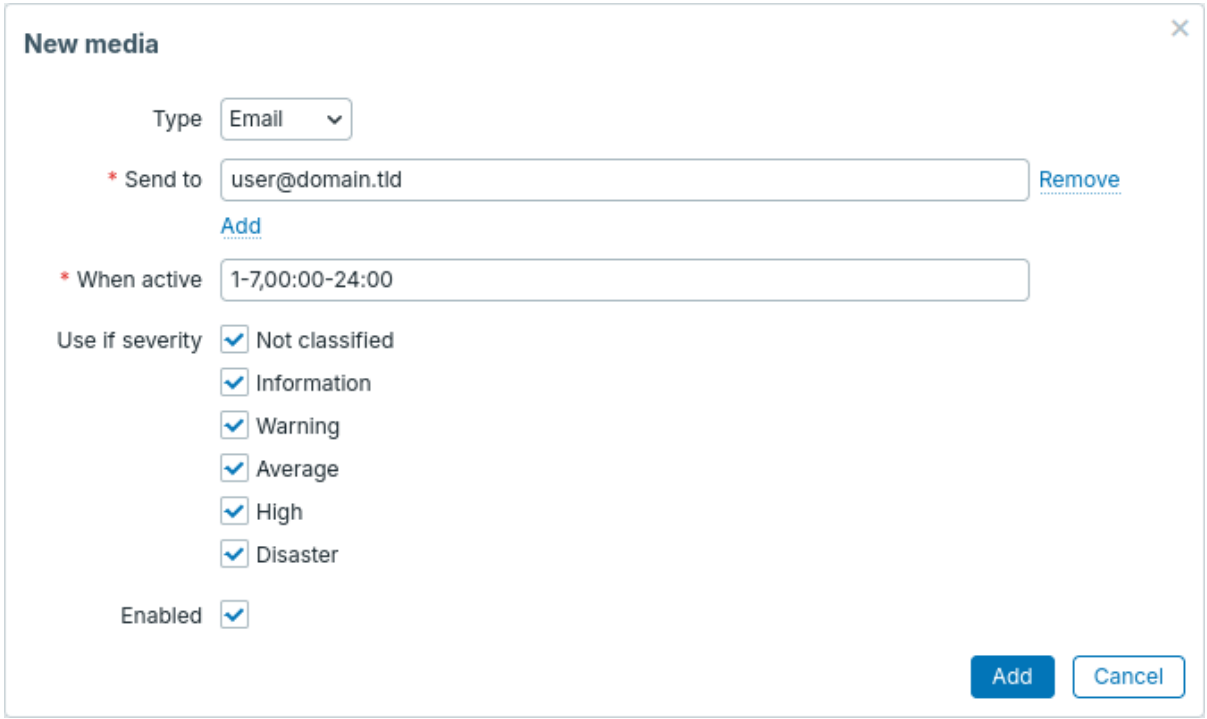
Como alternativa, na seção de menu *Monitoring > Hosts*, você pode clicar em *Latest data* para visualizar todas as métricas coletadas mais recentes em uma lista. Observe que o item *MySQL: Calculated value of innodb_log_file_size* deve ficar sem dados, pois o valor será calculado com base nos dados da última hora.



Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *User settings > Profile*, abra a aba *Media* e adicione seu e-mail.



2. Siga o guia para Receber uma notificação de problema.

Na próxima vez em 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.

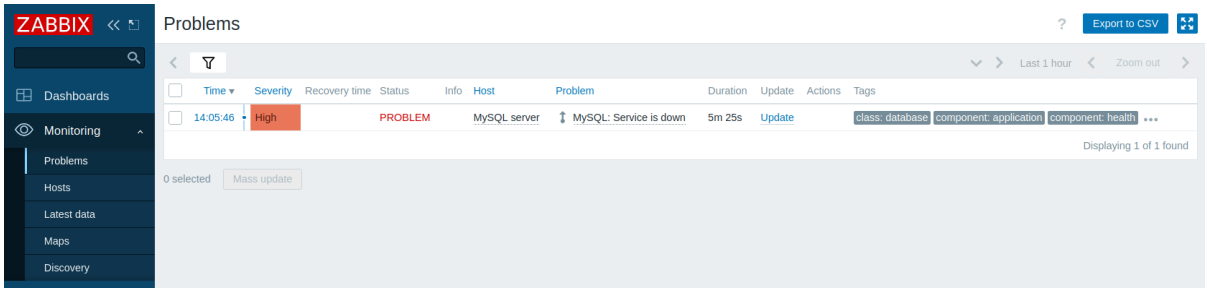
1. Abra a configuração do host do seu servidor MySQL no Zabbix.

2. Acesse a aba *Macros* e selecione *Inherited and host macros*.

3. Clique em *Change* ao lado, por exemplo, do valor da macro {$MYSQL.USER} configurado anteriormente e defina um nome de usuário MySQL diferente.

4. Clique em *Update* para atualizar a configuração do host.

5. Em alguns instantes, o Zabbix detectará o problema "MySQL: Service is down", porque não conseguirá se conectar ao servidor MySQL. O problema aparecerá em *Monitoring > Problems*.



Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta para o valor anterior para resolver o problema e continuar monitorando o servidor MySQL.

Outras abordagens para monitorar MySQL

Em vez de monitorar um server 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 oferecem suporte ao Zabbix agent 2 ou que exijam 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 server MySQL e coleta dados usando sua funcionalidade integrada, enquanto o ODBC depende de um driver ODBC para estabelecer uma conexão com o server MySQL e recuperar dados usando consultas SQL.

Embora muitas das etapas de configuração sejam semelhantes ao monitoramento de um server MySQL com o Zabbix agent 2, há algumas diferenças significativas - você precisa configurar o Zabbix agent ou o ODBC para poder monitorar um server MySQL. As instruções a seguir orientam você por essas **diferenças**.

Monitorar MySQL com Zabbix agent

Para monitorar um servidor MySQL com 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 se conectar ao servidor MySQL como o usuário "zbx_monitor" criado anteriormente.

Configurar o Zabbix agent

1. Navegue até o diretório de configurações adicionais do Zabbix agent.

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

> **Attention:**
> O diretório de configurações adicionais do Zabbix agent deve estar localizado no mesmo diretório que o arquivo de configuração do agent do Zabbix; o caminho padrão depende do sistema operacional e do método de instalação do Zabbix.

Em vez de definir todos os parâmetros de usuário necessários para monitorar o server MySQL no arquivo de configuração do Zabbix agent, 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 Zabbix agent.

```
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-o.

4. Reinicie o Zabbix agent para atualizar sua configuração.

```
systemctl restart zabbix-agent
```

Depois de configurar os parâmetros de usuário do Zabbix agent, você poderá prosseguir para configurar as credenciais que permitirão que o Zabbix agent acesse o server MySQL.

5. Navegue até o diretório inicial do Zabbix agent (se ele não existir no seu sistema, você precisará criá-lo; padrão: */var/lib/zabbix*).

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

6. Crie um arquivo *.my.cnf* no diretório inicial do Zabbix agent.

```
vi .my.cnf
```

7. Copie o conteúdo a seguir 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

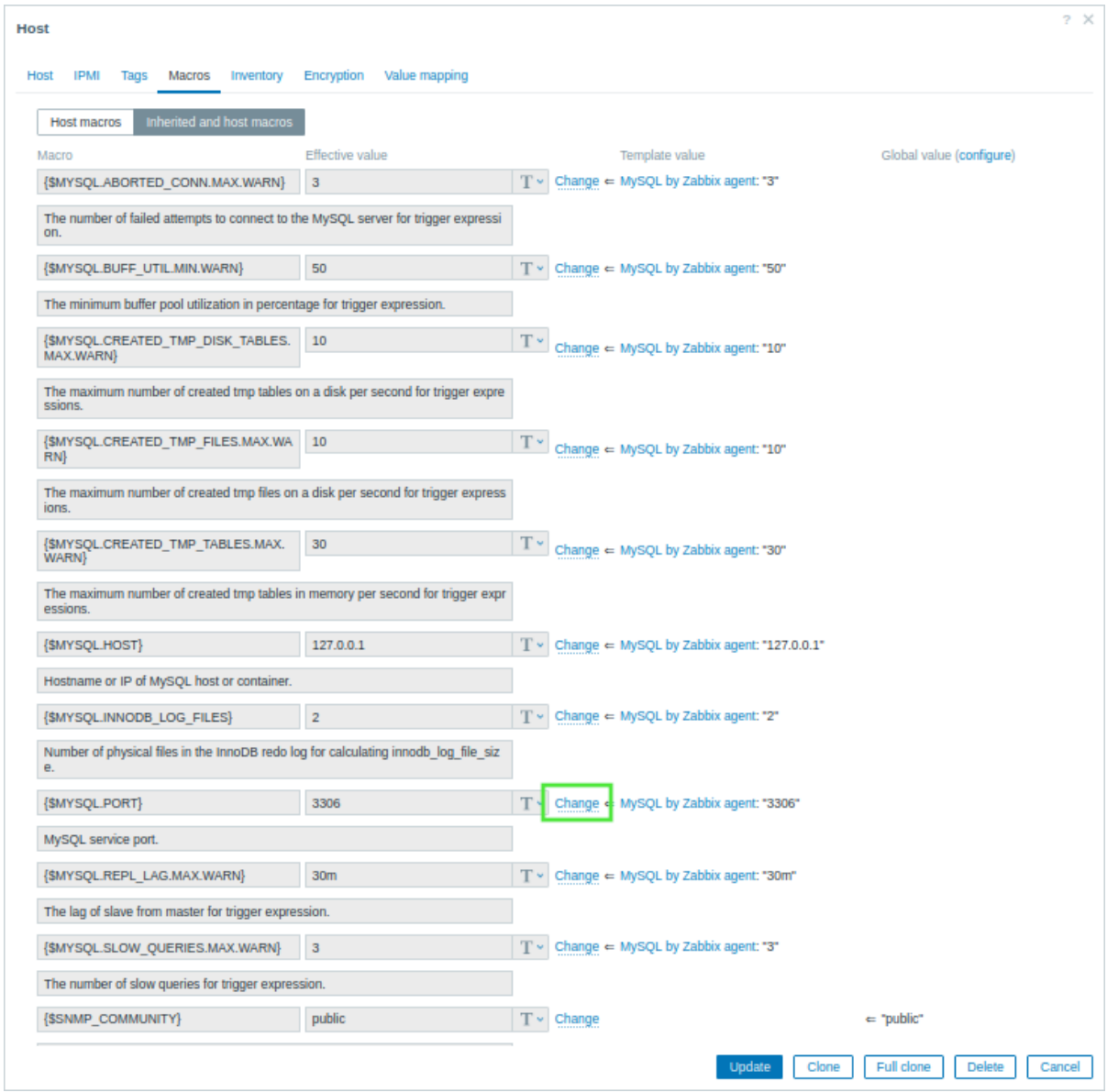
Para configurar o frontend do Zabbix, siga as instruções na seção *Configure Zabbix frontend* com os seguintes ajustes:

- 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 ver as métricas coletadas e configurar alertas de problema.

Para testar sua configuração, siga as instruções na seção *Test your configuration* com os seguintes ajustes:

- Na seção *Inherited and host macros* da configuração do host do servidor MySQL, clique em *Change* ao lado do valor da macro {$MYSQL.PORT} e defina uma porta diferente (por exemplo, "6033").



Monitorar MySQL com ODBC

Para monitorar um server 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*.

Depois de criar o usuário MySQL, você precisa configurar o ODBC. Isso inclui instalar uma das implementações de API ODBC de código aberto mais usadas - unixODBC - e um driver unixODBC, além de editar o arquivo de configuração do driver ODBC.

Configurar ODBC

1. Instale o unixODBC. A forma 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 o local 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 de arquivo *odbcinst.ini* a seguir, o nome do driver é "MariaDB Unicode".

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

5. Copie o conteúdo a seguir 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 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 *Inherited and host macros* da configuração do host do servidor MySQL deve ser definido como o nome do DSN do arquivo *odbc.ini*.

Depois de configurar o frontend do Zabbix, você pode visualizar as métricas coletadas, configurar alertas de problema e testar sua configuração.

Veja também

- Criando um item - como começar a monitorar métricas adicionais.
- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, em seguida, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).
- Monitoramento ODBC - como configurar o ODBC em outras distribuições Linux e como começar a monitorar métricas adicionais relacionadas a banco de dados com ODBC.
- Template *MySQL by Zabbix agent* - informações adicionais sobre o template *MySQL by Zabbix agent*.
- Template *MySQL by Zabbix agent 2* - informações adicionais sobre o template *MySQL by Zabbix agent 2*.
- Template *MySQL by ODBC* - informações adicionais sobre o template *MySQL by ODBC*.

Monitore VMware com Zabbix

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico do VMware.

Para quem este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico do VMware. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais sofisticada, consulte a seção Virtual machine monitoring ou a seção Configuration do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, algumas etapas deste guia podem diferir ligeiramente. Este guia é baseado em um ambiente executando Ubuntu.

Parte-se do pressuposto de que o VMware já está configurado. Este guia não aborda a configuração do VMware.

Configurar o Zabbix server

Para monitorar o VMware, os processos *vmware collector* do Zabbix precisam estar habilitados. Para obter mais informações sobre como o monitoramento do VMware é realizado, consulte Monitoramento de máquinas virtuais.

1. Abra o arquivo de configuração do Zabbix server.

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

2. Localize e defina o parâmetro `StartVMwareCollectors` no arquivo de configuração do Zabbix server como 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
```

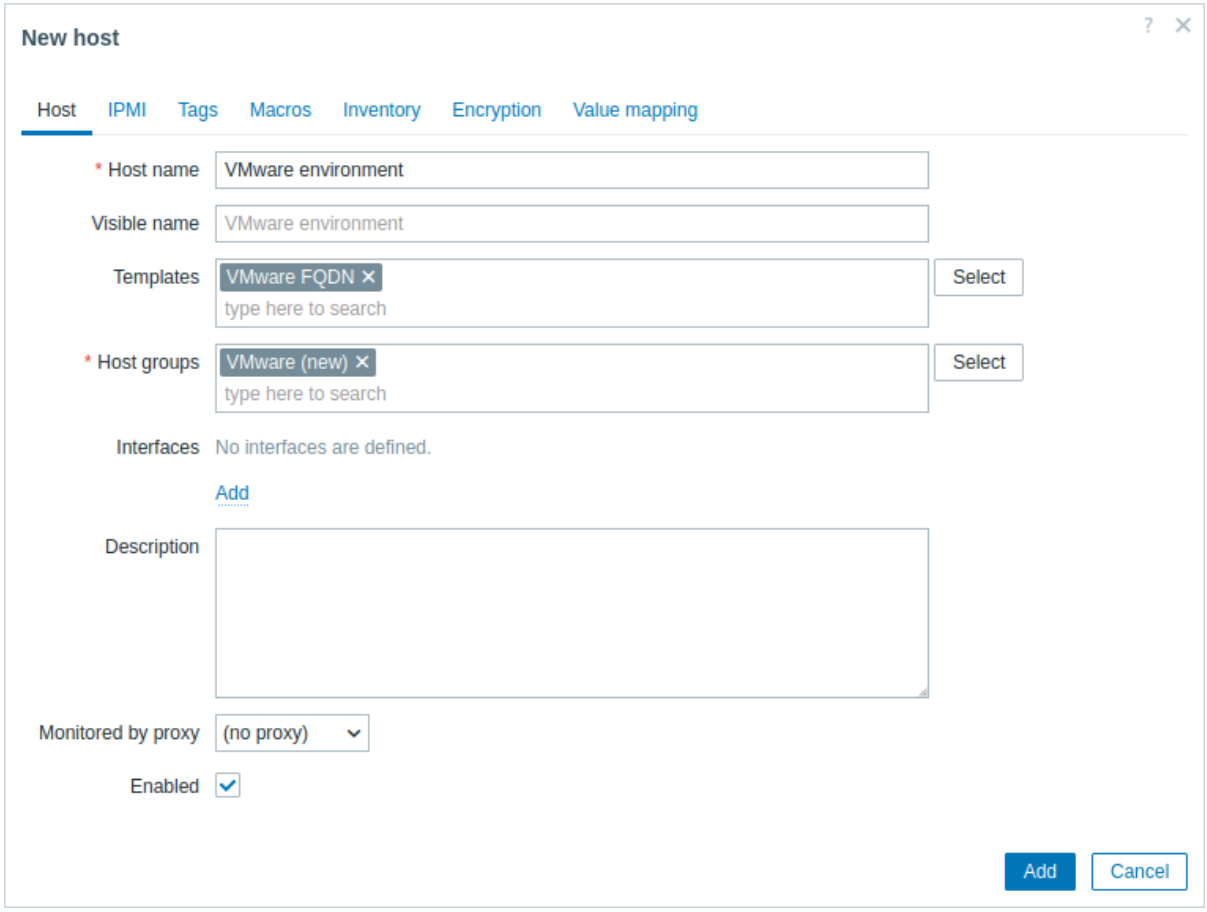
Depois que os processos *vmware collector* forem iniciados, prossiga para a próxima etapa.

Configurar o frontend do Zabbix

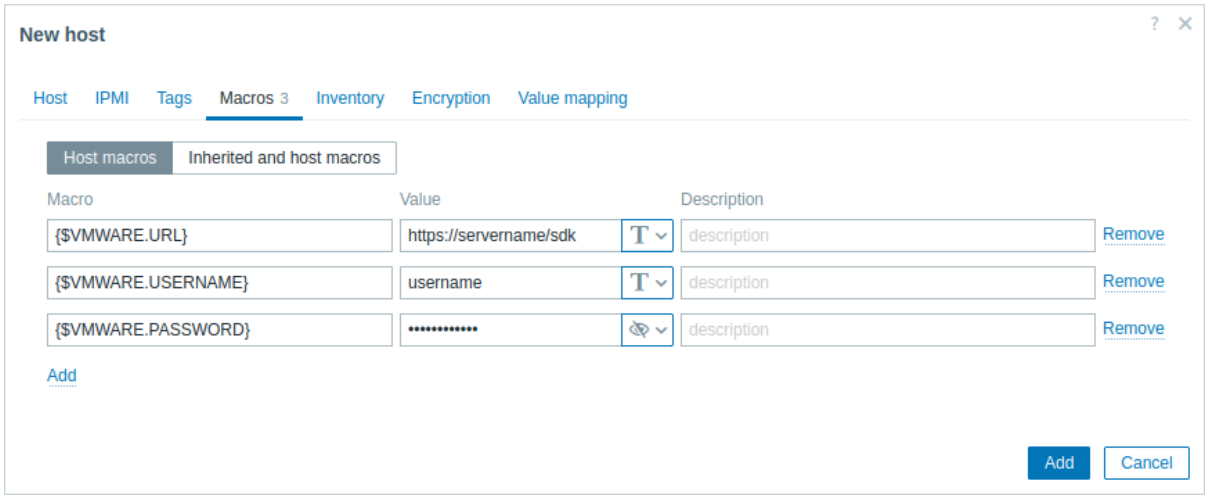
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Host name*, informe um nome de host (por exemplo, ”VMware environment”).
- No campo *Templates*, digite ou selecione o template ”VMware FQDN” (ou ”VMware”). Para mais informações sobre esses templates, consulte Monitoramento de máquina virtual.
- No campo *Host groups*, digite ou selecione um grupo de hosts (por exemplo, um novo grupo de hosts ”VMware”).



- Na aba *Macros*, defina as seguintes macros do 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



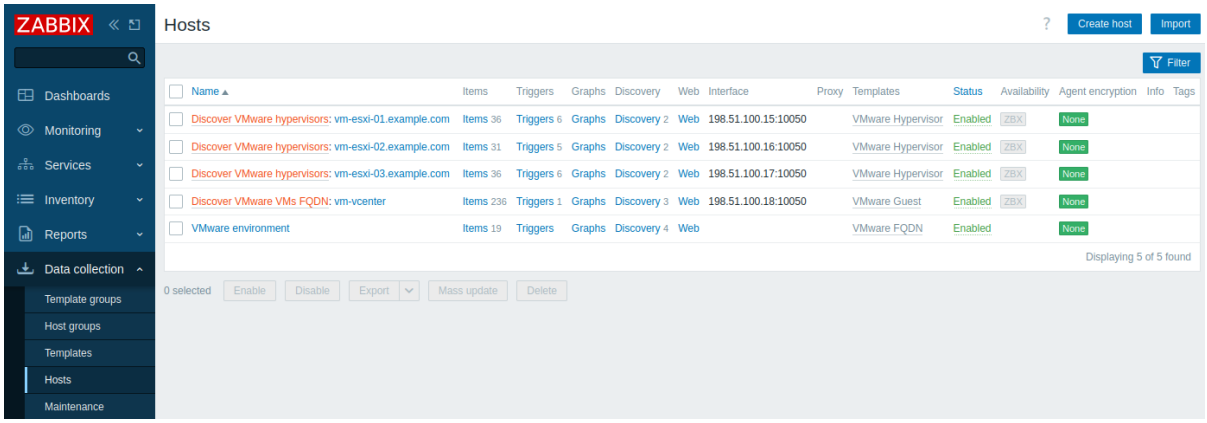
3. Clique no botão *Add* para criar o host. Este host representará seu ambiente VMware.

Ver métricas coletadas

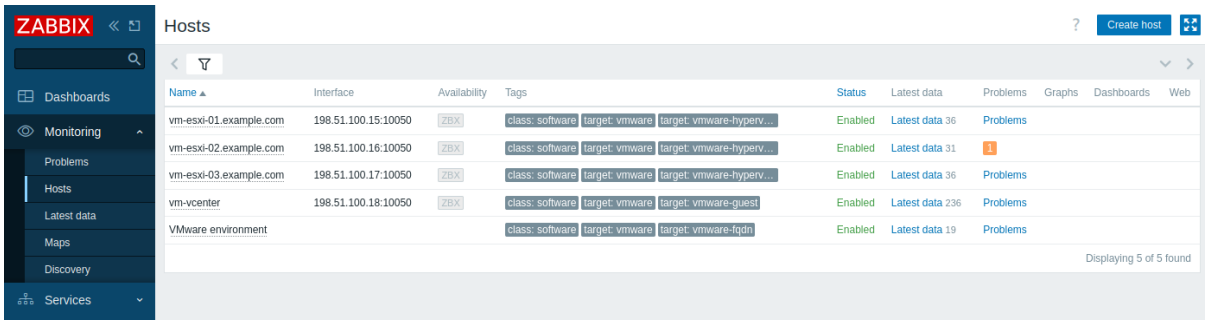
Parabéns! Neste ponto, o Zabbix já está monitorando seu ambiente VMware.

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.

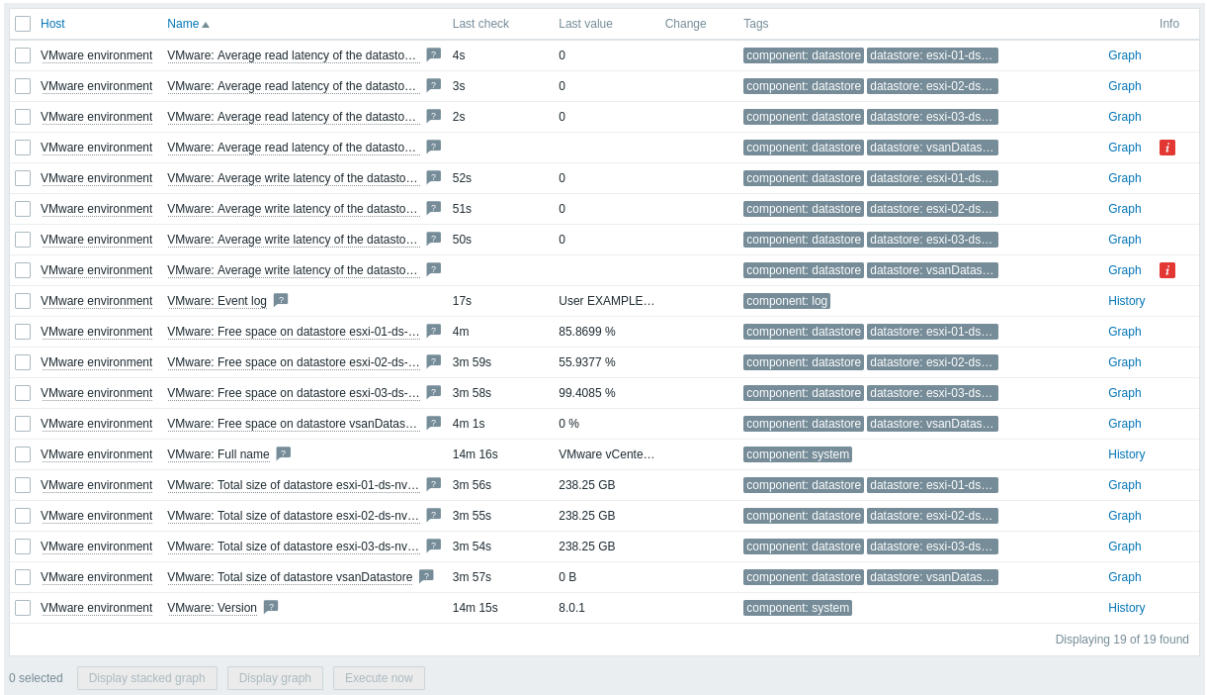
Para visualizar os hosts criados, navegue até a seção de menu *Coleta de dados > Hosts*.



Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento > Hosts* e clique em *Dados mais recentes* ao lado do host criado ”Ambiente VMware” ou de um dos hosts que foram criados para as entidades descobertas.



Essa ação abrirá uma lista de todas as métricas mais recentes coletadas do host selecionado.

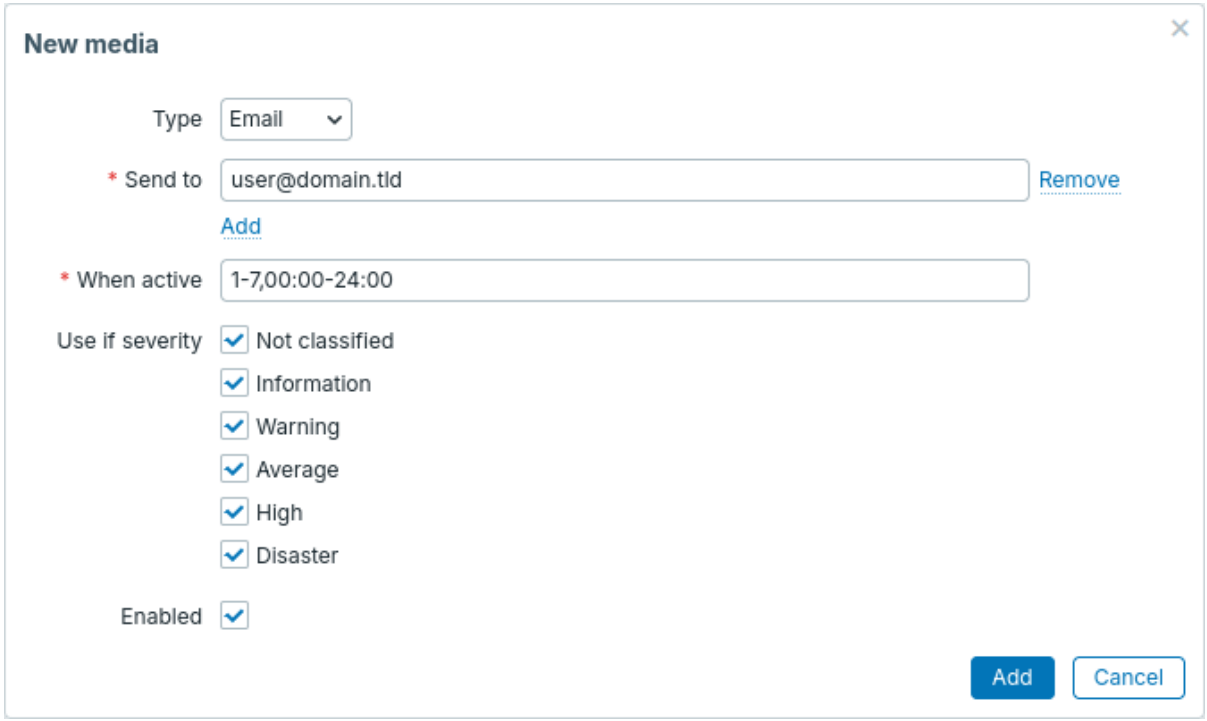


Observe que alguns items não têm dados e apresentam o estado *Não suportado*. Isso ocorre porque o Zabbix não consegue encontrar contadores de desempenho válidos no datastore específico, pois ele não está habilitado no ambiente VMware monitorado.

Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *User settings > Profile*, altere para a aba *Media* e adicione seu e-mail.



2. Siga o guia para Receiving a problem notification.

Na próxima vez em 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 problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, em seguida, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).
- Monitoramento de máquina virtual - informações adicionais sobre o monitoramento VMware (processo de coleta de dados, opções de configuração do server, orientações de 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*.

Monitore o tráfego de rede com o Zabbix

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico do tráfego da sua rede com o Zabbix.

Para quem este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico do tráfego da sua rede. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais avançada, consulte a seção Configuration 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 monitoramento de tráfego.
Essa pode ser a mesma host onde o Zabbix server está instalado ou uma host 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 em modo ativo ou passivo (simultaneamente). Consulte Passive and active agent checks para mais informações. Neste guia, será descrito o monitoramento por meio de verificações passivas.

Configurando o agent do Zabbix

1. Abra o arquivo de configuração do agent na máquina onde o agent está instalado:

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

2. Adicione o endereço IP ou o nome DNS do seu server Zabbix ao parâmetro *Server*. Por exemplo:

```
Server=192.0.2.22
```

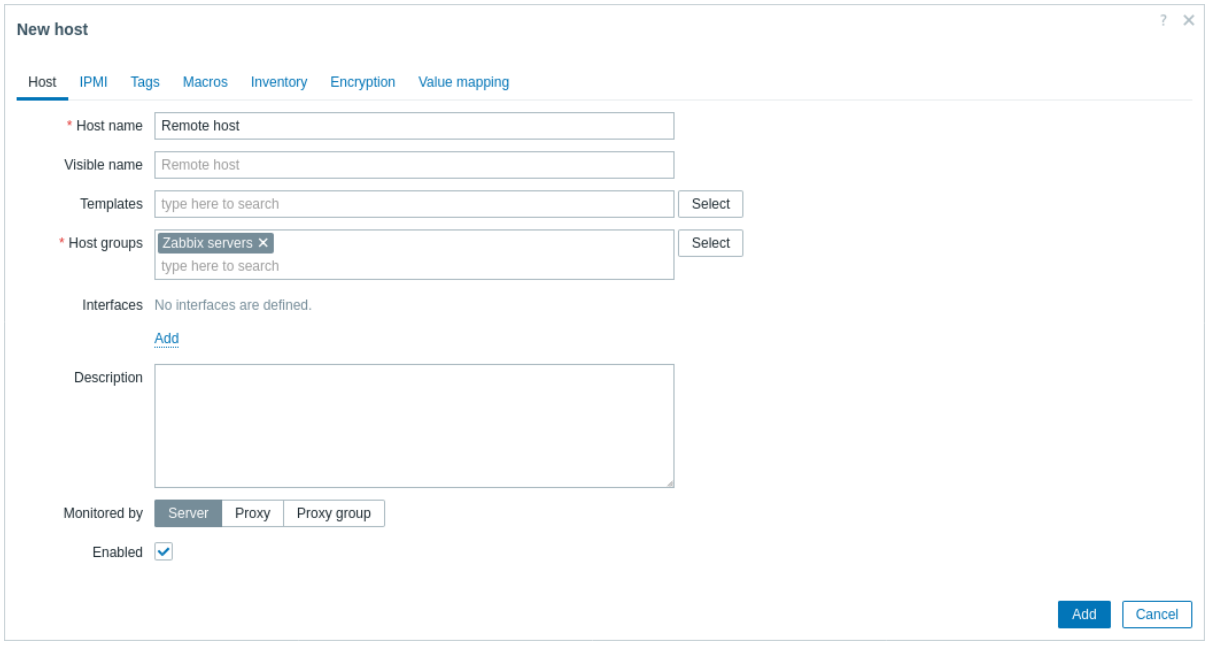
3. Reinicie o agent do Zabbix:

```
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.

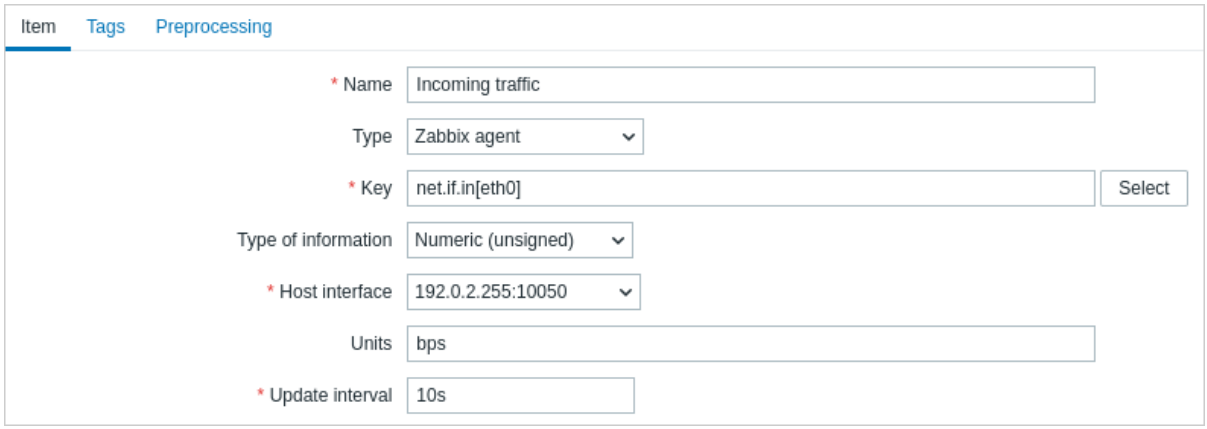


Criar itens

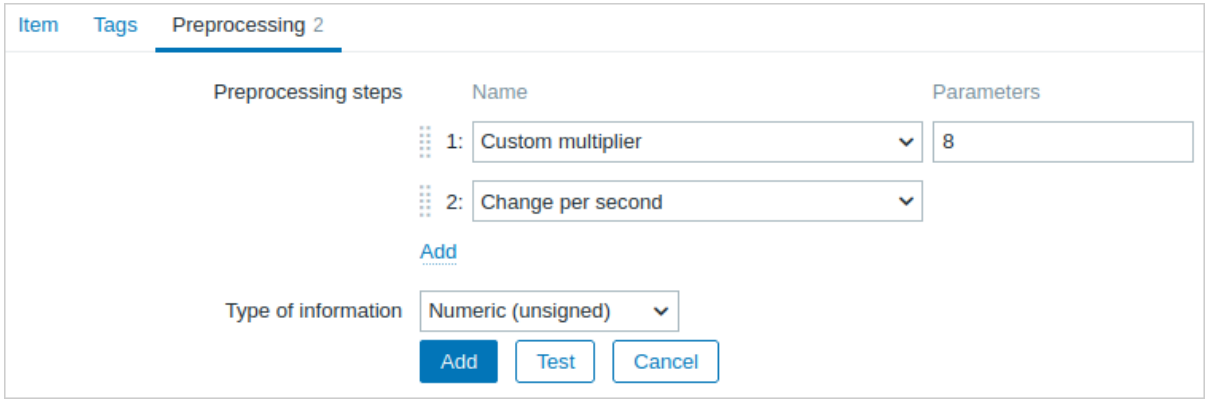
Siga as instruções em criar um item para adicionar os itens de monitoramento de tráfego, a saber:

- Tráfego de entrada
- Tráfego de saída
- Tráfego total

Uma configuração simples para o item de monitoramento de tráfego de entrada ficaria assim:



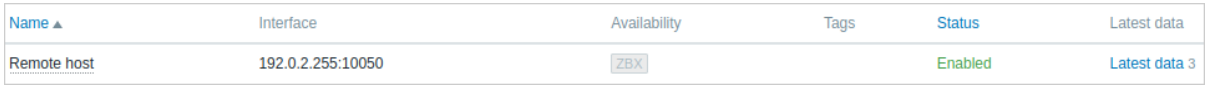
Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os itens. No caso apresentado, essas etapas podem ser a multiplicação por 8 (para converter bytes em bits) e a apresentação como variação por segundo.



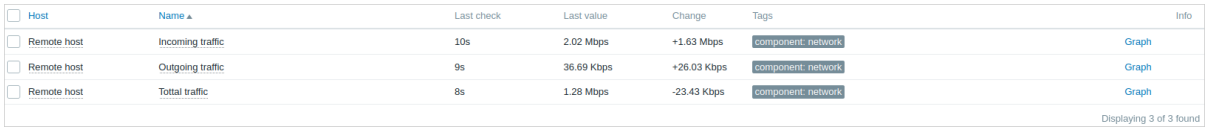
Visualizar dados coletados

Parabéns! Neste momento, o Zabbix já está monitorando o tráfego da sua rede.

Para visualizar as métricas coletadas, abra a seção do menu *Monitoring > Hosts* e clique em *Latest data* na linha do host.

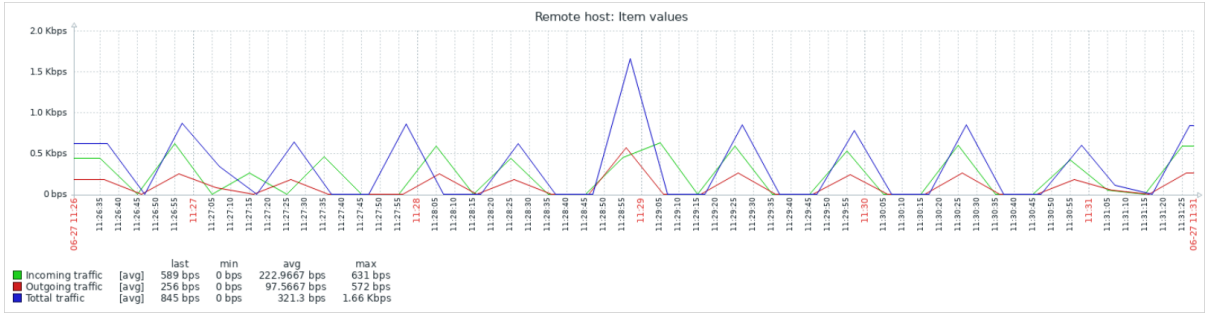


Você verá os dados de tráfego.



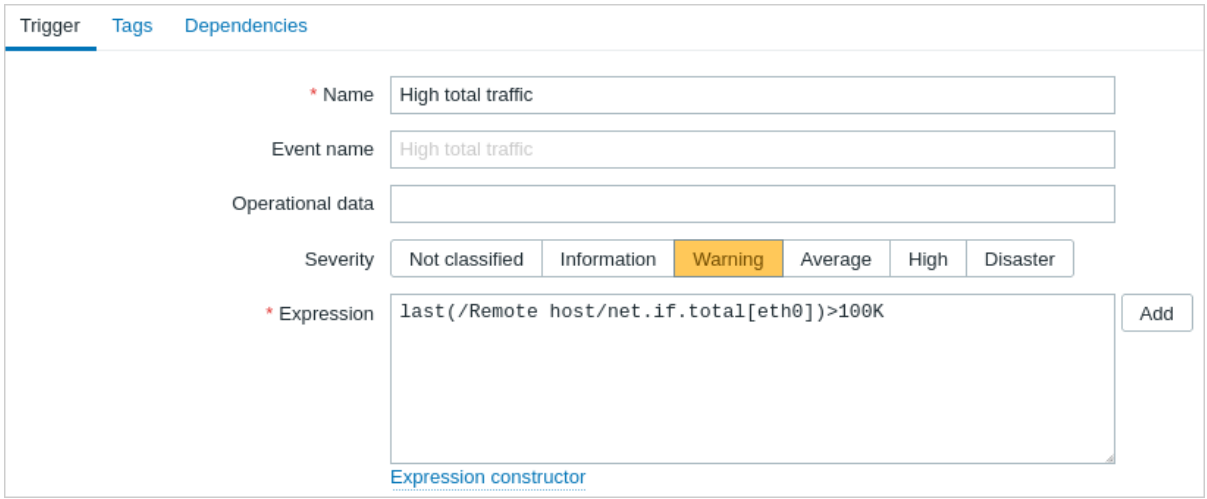
Ver gráficos

Os dados coletados podem ser exibidos como gráficos. Para visualizá-los, na seção *Latest data* clique em *Graph* na linha do item ou selecione os itens necessários e clique em *Display graph* abaixo.

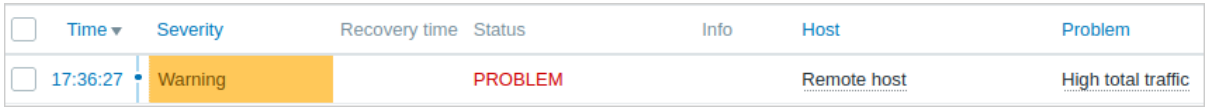


Configurar triggers

Você pode definir triggers para detectar tráfego de rede anormal. Consulte as instruções sobre como configurar um trigger e adicione o trigger que sinaliza que o tráfego total está muito alto, por exemplo:



Agora faça com que o tráfego ultrapasse o limite que você definiu na expressão do trigger e navegue até *Monitoring > Problems* para verificar se o problema está listado lá.



Configurar alertas de problema

Há várias maneiras de receber notificações sobre o problema. O email é a mais popular; siga as instruções para configurar uma notificação de problema por email. Você também pode escolher outros tipos de mídia para serem usados na entrega da notificação.

Veja também:

- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, em seguida, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).
- Reconhecimento de problema - como indicar que o problema é conhecido, fazer comentários sobre sua resolução, suprimir ou fechar o problema.
- Monitorar Linux com o agent do Zabbix - como iniciar o monitoramento básico dos itens mais importantes vinculando um template pré-configurado.

Monitorar o tráfego de rede usando verificações ativas

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico do tráfego da sua rede com o Zabbix usando verificações ativas.

Para quem este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico do seu tráfego de rede usando verificações ativas. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais avançada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent de acordo com as instruções para o seu sistema operacional. Observe que você deve instalar o Zabbix agent na máquina que requer monitoramento de tráfego. Essa pode ser a mesma host onde o Zabbix server está instalado ou uma host 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 em modo ativo ou passivo (simultaneamente). Consulte Passive and active agent checks para mais informações. Neste guia, será descrito o monitoramento por meio de **active checks**.

Configurando o Zabbix agent

1. Abra o arquivo de configuração do agent na máquina em que o agent está instalado.

Se você estiver usando o Zabbix agent:

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

Se você estiver usando o Zabbix agent 2:

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

2. Adicione o endereço IP ou o nome DNS (e a porta opcional) do seu Zabbix server ao parâmetro ServerActive. Por exemplo:

```
ServerActive=192.0.2.1:10051
```

O Zabbix agent usará esse endereço para se conectar à porta trapper do Zabbix server (padrão: 10051) e solicitar os dados de configuração das verificações ativas.

3. Defina o parâmetro Hostname, que deve corresponder ao nome do host que será definido no Zabbix frontend. Em nosso exemplo, ele é:

```
Hostname=Remote host
```

O valor de Hostname deve corresponder porque, para verificações ativas, o Zabbix agent o utiliza para recuperar a configuração correta do host no server. Especificamente, o agent inicia uma conexão com o server e se identifica usando o valor de Hostname. Em seguida, o server 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.

Se você estiver usando o Zabbix agent:

```
systemctl restart zabbix-agent
```

Se você estiver usando o Zabbix agent 2:

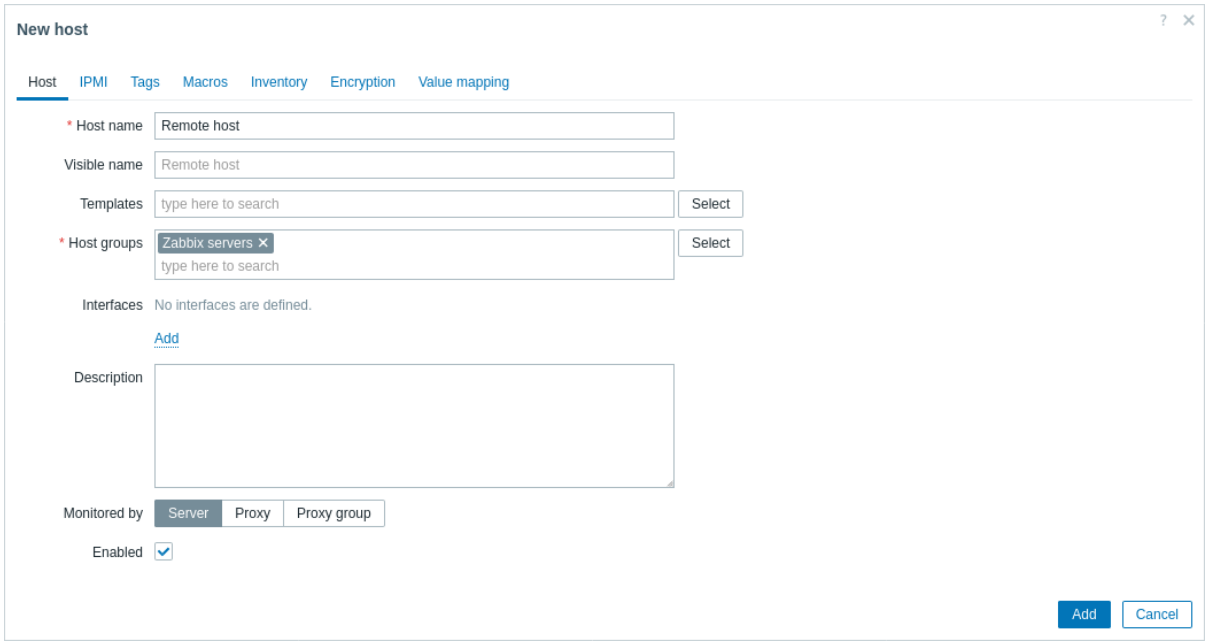
```
systemctl restart zabbix-agent2
```

frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

- No campo *Host name*, informe 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 *Host groups*, digite ou selecione um grupo de hosts (por exemplo, ”Zabbix servers”).
- Deixe *Interfaces* indefinido; uma interface não é necessária para verificações ativas porque o agent inicia a conexão com o server em vez de aguardar uma conexão do server.



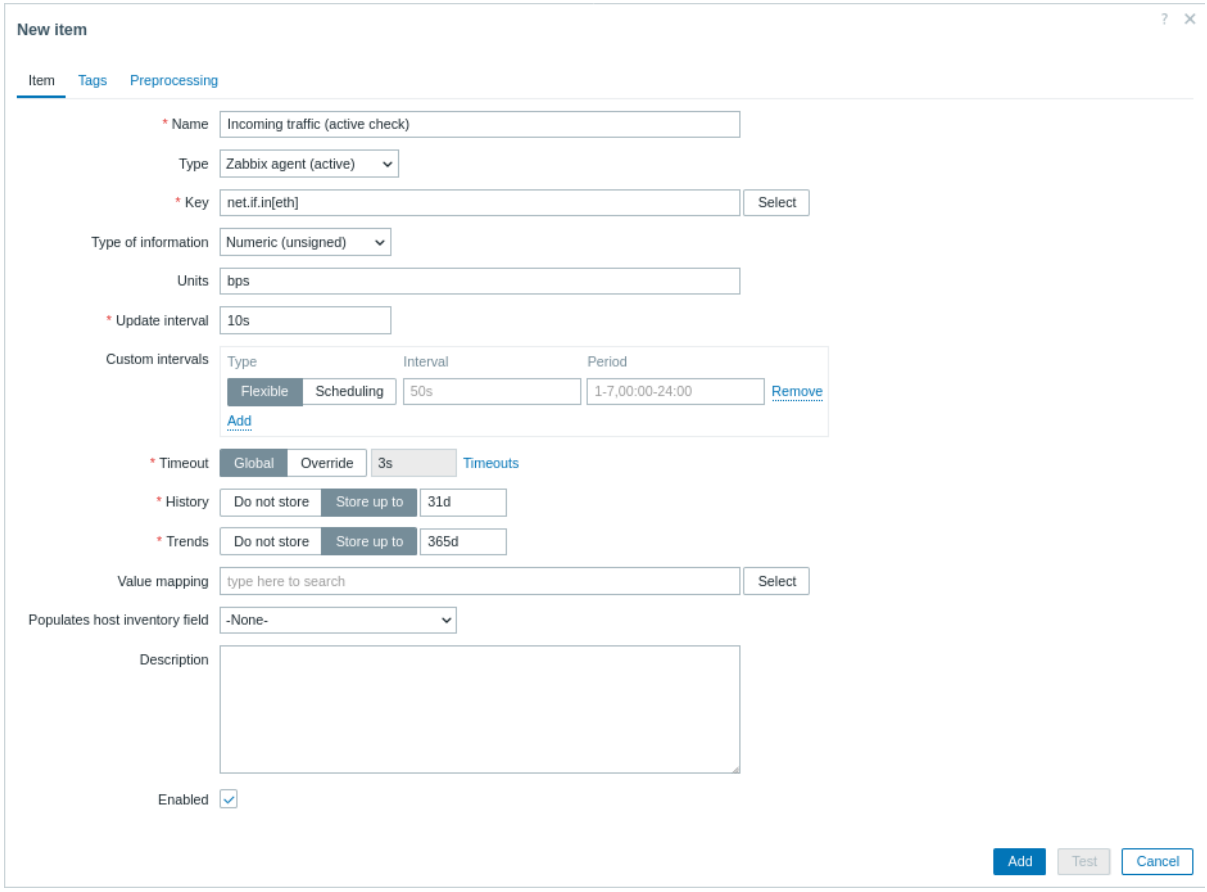
3. Clique em *Add* para adicionar o host. Este host representará a máquina Linux monitorada.

Criar items

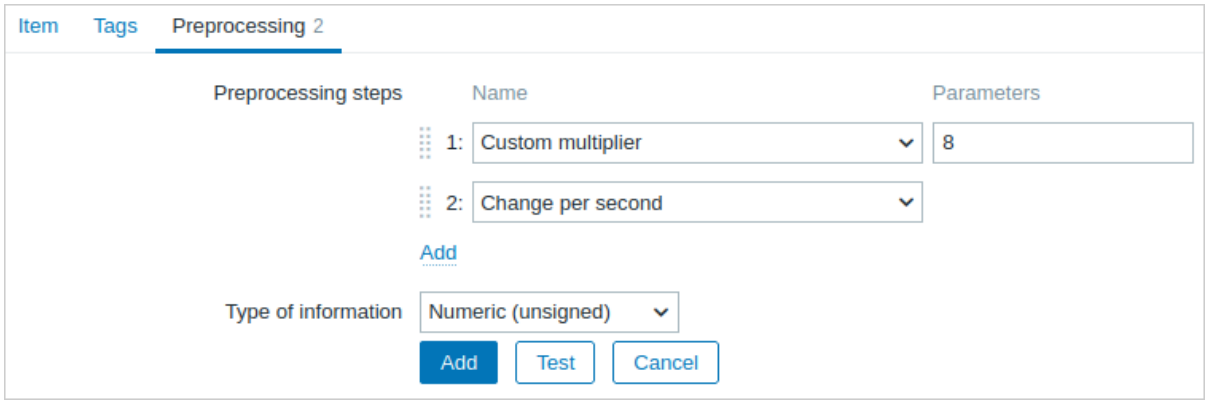
Siga as instruções em criar um item para adicionar os items para monitoramento de tráfego, a saber:

- Tráfego de entrada
- Tráfego de saída
- Tráfego total

Uma configuração simples para o item de monitoramento de tráfego de entrada com uma verificação ativa seria a seguinte:



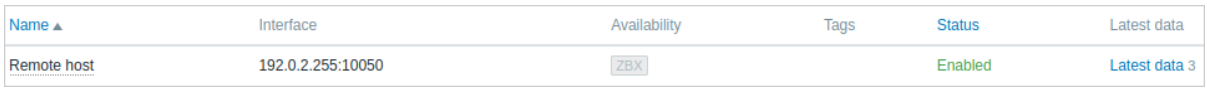
Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os items. No caso dado, essas etapas podem ser a multiplicação por 8 (para converter bytes em bits) e a apresentação como alteração por segundo.



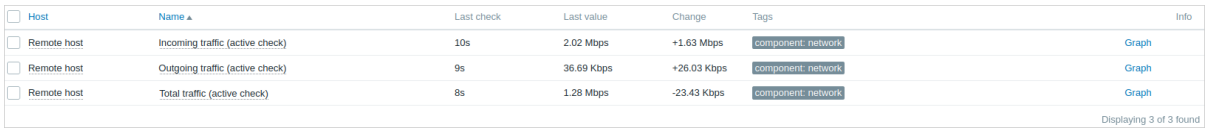
Ver dados coletados

Parabéns! Neste ponto, o Zabbix já está monitorando o tráfego da sua rede.

Para visualizar as métricas coletadas, abra a seção de menu *Monitoring > Hosts* e clique em *Latest data* na linha do host.



Você verá os dados de tráfego.



Veja também:

- Ver gráficos - como exibir os dados coletados como gráficos.
- Configurar triggers - como configurar triggers para detectar tráfego de rede anormal.
- Configurar alertas de problema - como configurar notificações sobre situações de problema.

- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, depois, se um problema não for resolvido em 45 minutos, enviar uma mensagem ao gerente do data center).
- Reconhecimento de problema - como indicar que o problema é conhecido, fazer comentários sobre sua resolução, suprimir ou fechar o problema.
- Monitorar Linux com Zabbix agent - como iniciar o monitoramento básico dos itens mais importantes vinculando um template pré-configurado.
- 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 às correções de bugs.

Monitore sites com itens de navegador

Introdução

Esta página orienta você pelas etapas necessárias para iniciar o monitoramento básico de sites com itens de navegador.

Para quem este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de sites com itens de Browser. Se você estiver procurando opções de personalização avançada ou precisar de uma configuração mais avançada, consulte a página Browser items ou a seção Configuration do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, algumas etapas deste guia podem diferir ligeiramente. Este guia é baseado em um ambiente executando Ubuntu.

Configurar o WebDriver

Itens de navegador exigem uma estrutura 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 contêiner Docker.

Pressupõe-se que o Docker já esteja configurado. Este guia não aborda a configuração do Docker. Para instruções de instalação, consulte Instalar o Docker Engine no Ubuntu.

1. Inicie o Selenium Server com Chrome em um contêiner Docker com as seguintes opções:

- **docker run --name browser** - executa um novo contêiner Docker chamado "browser";
- **-p 4444:4444** - mapeia a porta 4444 na sua máquina host para a porta 4444 no contêiner (esta é a porta usada pelo Selenium Server para aceitar comandos);
- **-p 7900:7900** - mapeia a porta 7900 na sua máquina host para a porta 7900 no contêiner (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 2 GB de memória compartilhada para o contêiner (isso é importante para o Chrome funcionar corretamente, pois ele pode exigir uma quantidade significativa de memória compartilhada para evitar falhas);
- **-d** - executa o contêiner em modo detached, ou seja, 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. Verifique se o contêiner Docker `browser` está em execução e acessível.

- Recupere o endereço IP do contêiner (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 contêiner usando 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 com 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 de 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 testes web configurado anteriormente.

Por padrão, o parâmetro StartBrowserPollers é definido como 1; portanto, você só precisa especificar o endpoint de testes web.

1. Abra o arquivo de configuração do server do Zabbix:

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

2. Localize e defina o parâmetro WebDriverURL no arquivo de configuração do server do Zabbix:

```
###### Option: WebDriverURL
#####WebDriver interface HTTP(S) URL. For example, http://localhost:4444 used with Selenium WebDriver star
#####
##### Mandatory: no
##### Default:
##### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Reinicie o Zabbix server.

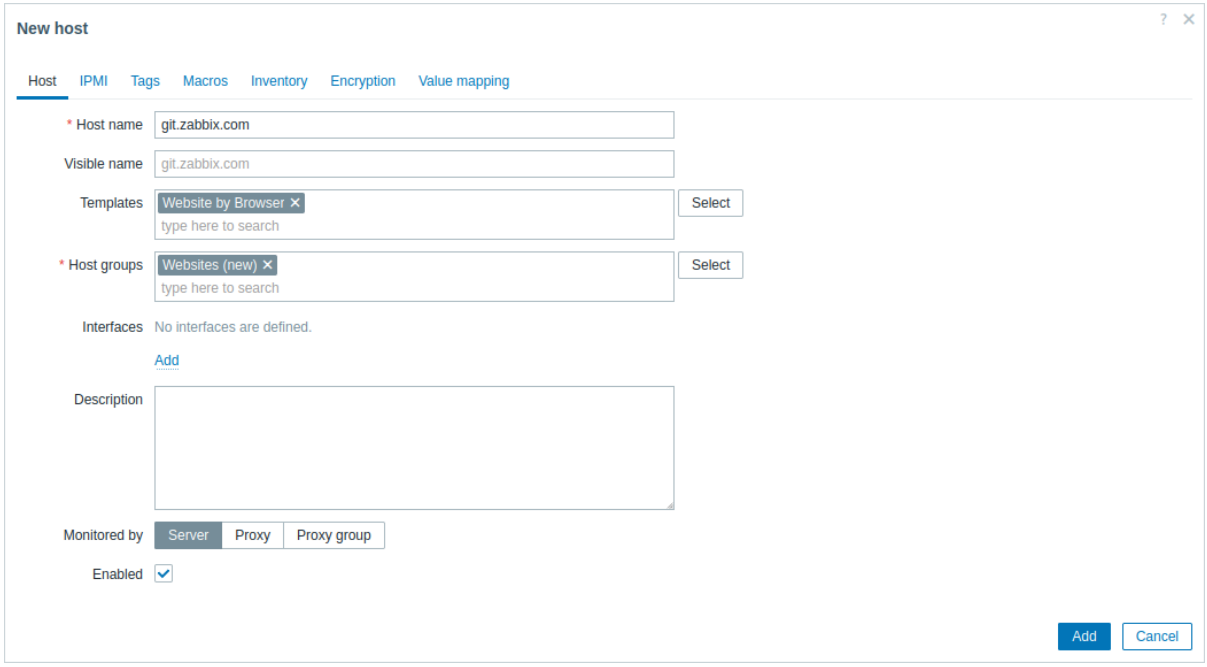
```
systemctl restart zabbix-server
```

Configurar o frontend do Zabbix

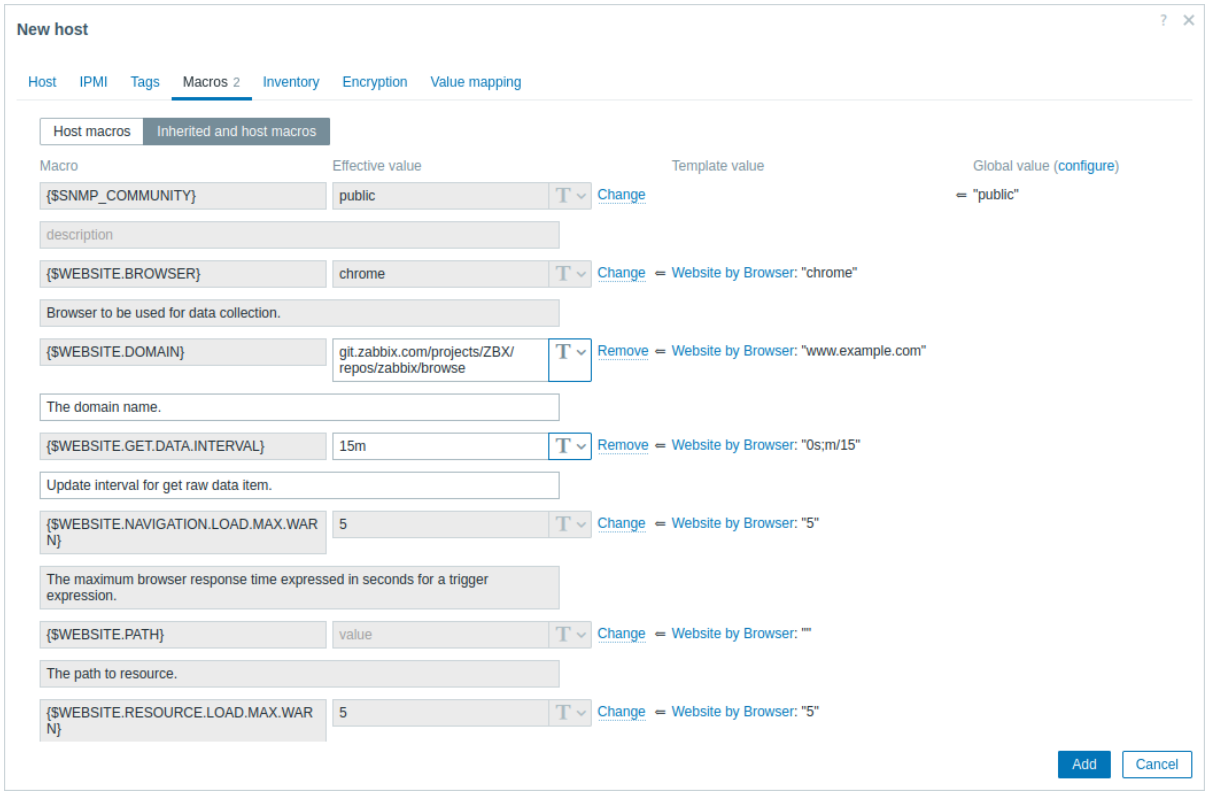
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Host name*, informe 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 *Host groups*, digite ou selecione um grupo de hosts (por exemplo, um novo grupo de hosts ”Websites”).



- Na aba *Macros*, mude para *Inherited and host macros*, procure as seguintes macros e clique em *Change* ao lado do valor da macro para atualizá-lo:
  - {$WEBSITE.DOMAIN} - Nome de domínio (por exemplo, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Intervalo de atualização dos dados do item (por exemplo, 15m)

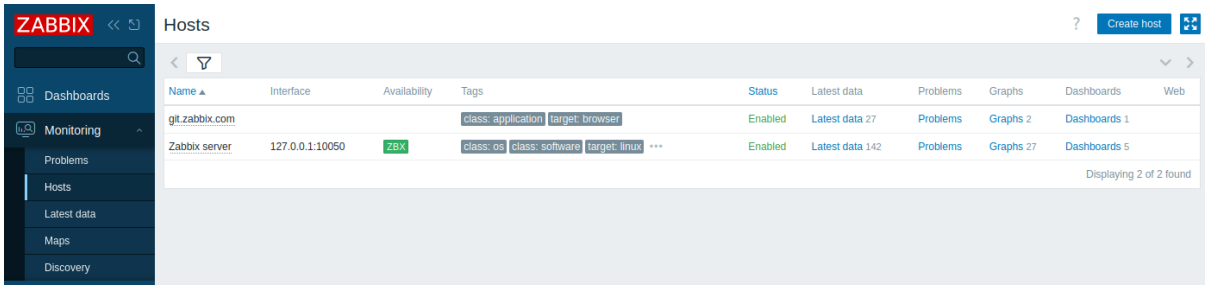


3. Clique no botão *Add* para criar o host. Este host representará o site que você deseja monitorar.

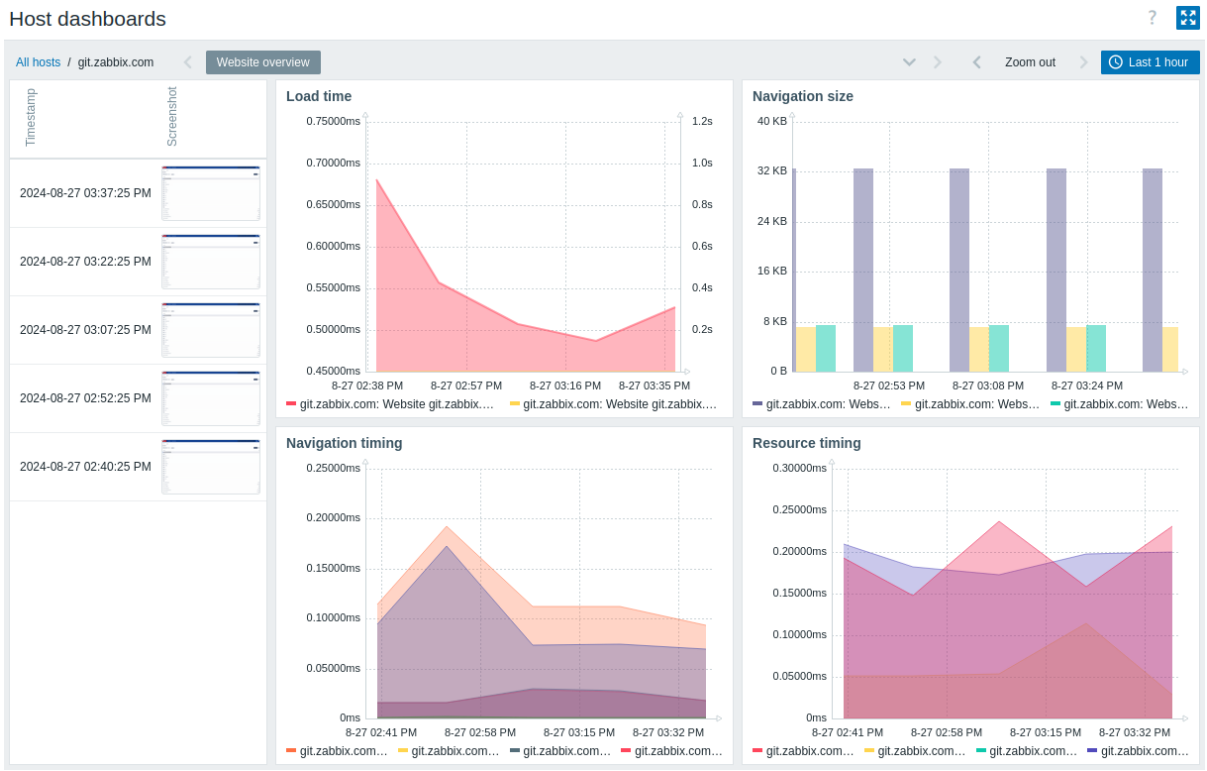
Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando o site que você especificou.

Para ver as métricas coletadas, navegue até a seção de menu *Monitoring > Hosts* e clique em *Dashboards* ao lado do host.



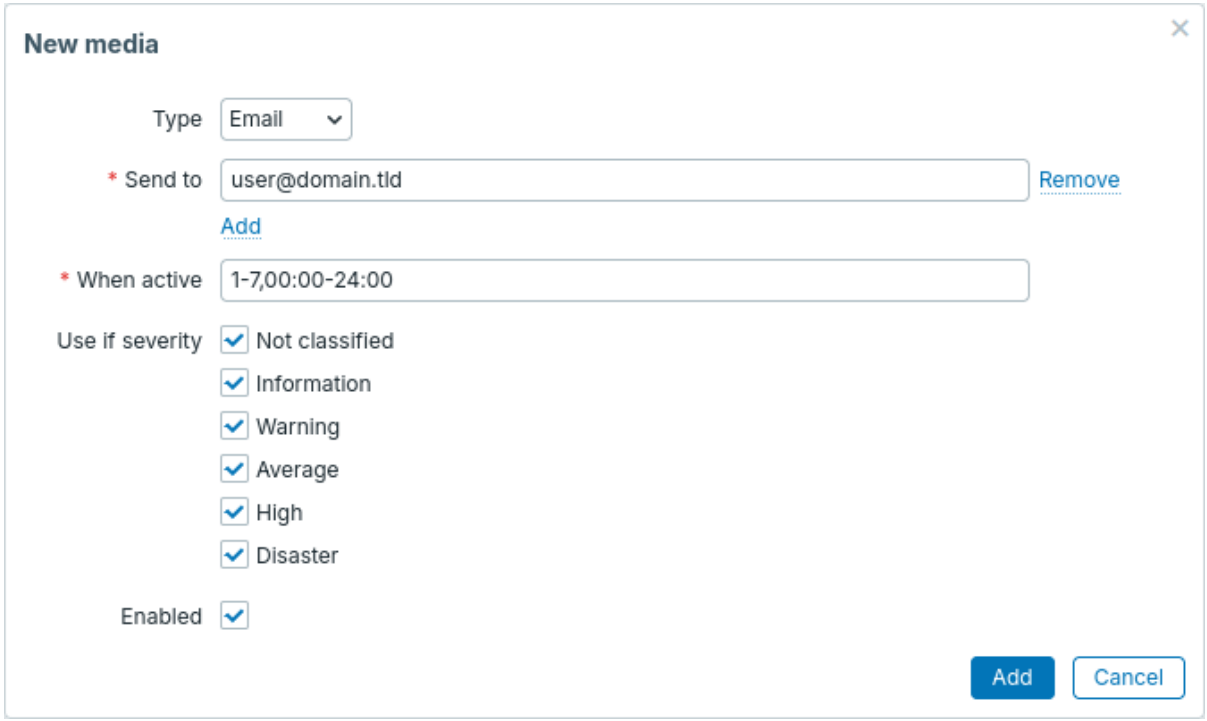
Essa ação levará você ao dashboard do host (configurado no nível do template) com as métricas mais importantes coletadas do site.



Configurar alertas de problema

O Zabbix pode notificá-lo sobre um problema na sua infraestrutura usando vários métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *User settings > Profile*, altere para a aba *Media* e adicione seu e-mail.



2. Siga o guia para Receiving a problem notification.

Na próxima vez em 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 Zabbix frontend.

1. Abra a configuração do host do seu website no Zabbix.

2. Acesse a aba *Macros* e selecione *Inherited and host macros*.

3. Clique em *Change* 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 *Update* para atualizar a configuração do host.

5. Em alguns instantes, o Zabbix detectará o problema "Failed to get JSON of the requested website", porque não conseguirá se conectar ao website especificado. O problema aparecerá em *Monitoring > Problems*.



Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta para o valor anterior para resolver o problema e continuar monitorando o website.

Veja também

- Criando um item - como começar a monitorar métricas adicionais.
- Escalonamentos de problema - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar uma mensagem ao administrador do sistema e, em seguida, se um problema não for resolvido em 45 minutos, enviar uma 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*.

Monitore 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 agent 2 do Zabbix. 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 este guia é destinado

Este guia foi elaborado para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de certificados de sites. Se você estiver procurando opções de personalização avançadas ou precisar de uma configuração mais sofisticada, consulte a seção Configuration 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 estejam 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 agent do Zabbix:

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

2. Defina o parâmetro Server como 127.0.0.1, pois o agent e o server estão sendo executados 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 nenhuma instalação ou configuração separada.

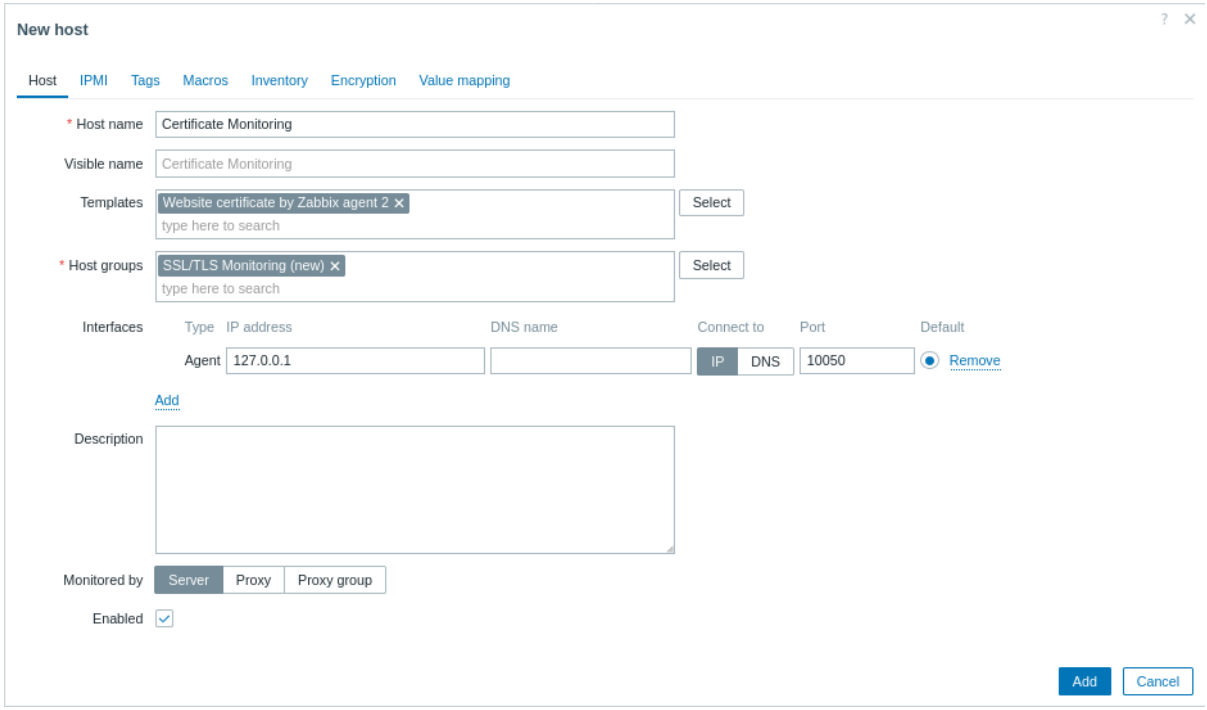
Configurar o frontend do Zabbix

1. Faça login no frontend do Zabbix.

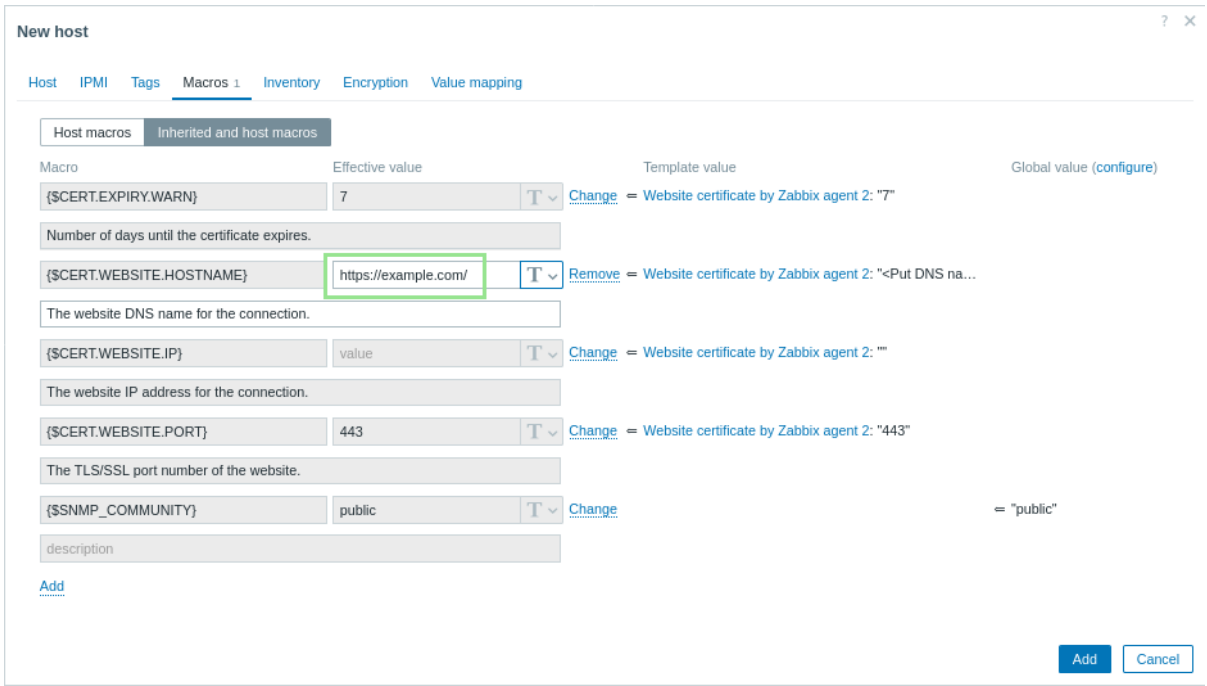
2. Navegue até *Monitoring* > *Hosts*.

3. Clique em um host existente no qual você deseja monitorar certificados de sites, ou crie um host se necessário:

- No campo *Host name*, informe um nome de host (por exemplo, ”Certificate Monitoring”).
- No campo *Templates*, digite ou selecione o template ”Website certificate by Zabbix agent 2” que será vinculado ao host.
- No campo *Host groups*, digite ou selecione um grupo de hosts (por exemplo, ”SSL/TLS Monitoring”).
- No campo *Interfaces*, adicione uma interface do tipo ”Agent” e especifique um endereço IP. Para este exemplo, é usado ”127.0.0.1”.



- Na aba *Macros*, mude para *Inherited and host macros*, procure as seguintes macros e clique em *Change* ao lado do valor da macro para atualizá-lo:
  - `{$CERT.WEBSITE.HOSTNAME}` - informe o nome DNS do site desejado como valor.



4. Clique em *Add* para adicionar o host.

Para monitorar vários sites, você pode especificar uma lista separada por vírgulas dos respectivos hostnames na macro `{$CERT.WEBSITE.HOSTNAME}`. Opcionalmente, você também pode fornecer uma lista separada por vírgulas de portas na macro `{$CERT.WEBSITE.PORT}`, em que cada porta corresponde ao respectivo hostname na mesma 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

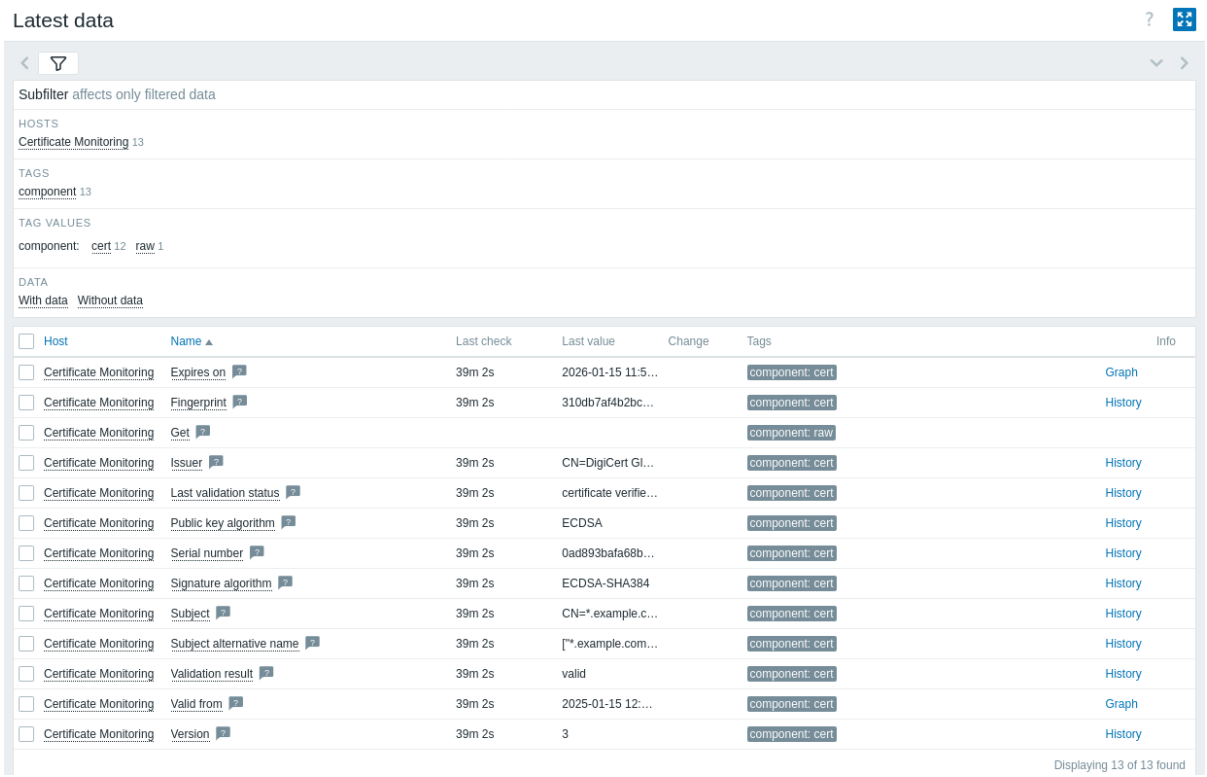
Para cada site especificado na macro `{$CERT.WEBSITE.HOSTNAME}`, o Zabbix criará um conjunto correspondente de items e triggers. Isso permite o monitoramento e o acionamento de alertas individualmente para o certificado SSL de cada site.

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:…

Ver métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando o certificado web desejado.

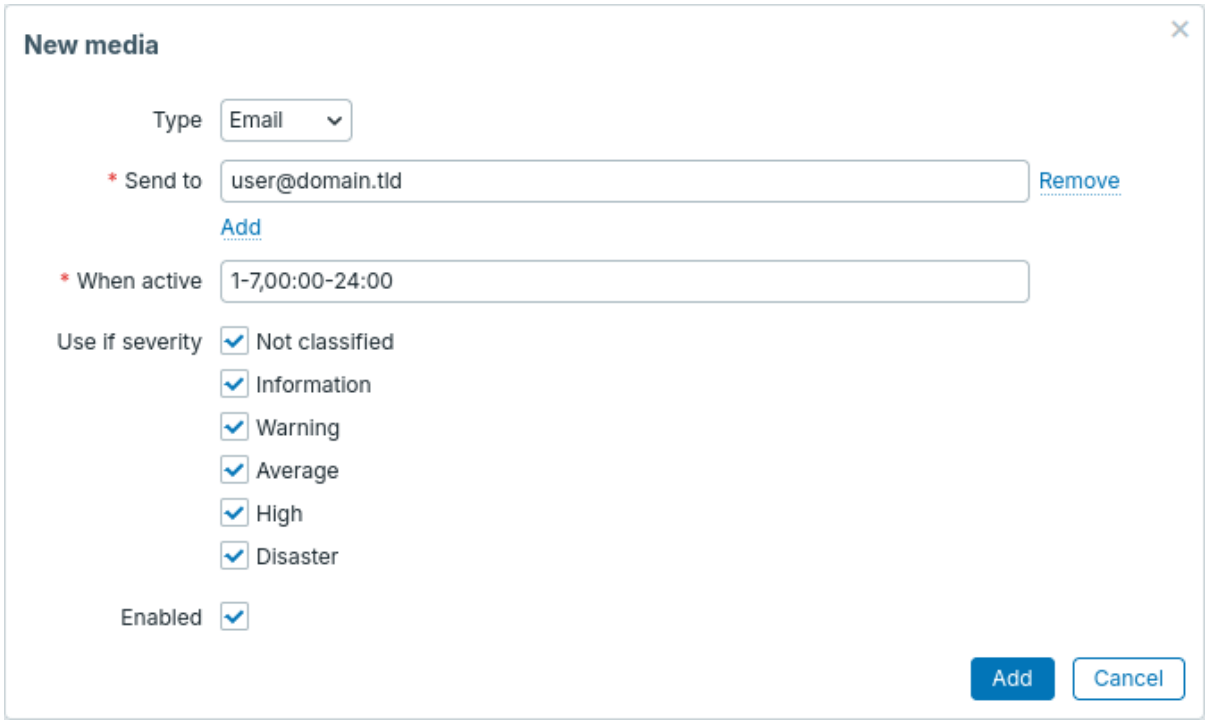
Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoring > Hosts* e clique em *Latest data* ao lado do host para ver todas as métricas coletadas mais recentes em uma lista, como data de expiração, emissor e assunto.



Configurar alertas de problema

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. Navegue até *User settings > Profile*, altere para a aba *Media* e adicione seu e-mail.



2. Siga o guia para Receiving a problem notification.

Na próxima vez em 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.

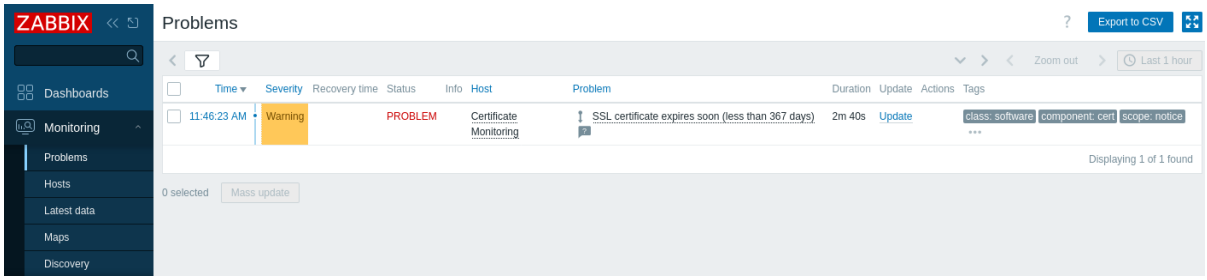
1. Abra a configuração do host ”Certificate Monitoring” no Zabbix.

2. Acesse a aba *Macros* e selecione *Inherited and host macros*.

3. Clique em *Change* ao lado do valor da macro `{$CERT.EXPIRY.WARN}` configurada anteriormente e defina um número de dias muito alto (um valor maior que 365 dias deve ser suficiente) para receber um aviso antes que o certificado expire.

4. Clique em *Update* 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 *Monitoring > Problems*.



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

- Zabbix agent 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*.
- Monitor websites with Browser items - como iniciar o monitoramento básico de sites com Browser items.

Monitore um switch ou roteador de rede com o Zabbix

Introdução

Este guia orienta você pelas etapas necessárias para iniciar o monitoramento básico do seu switch de rede ou roteador 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 este guia é destinado

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 mais detalhadas, consulte a página SNMP agent ou a seção Configuration do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, certifique-se de que você tem:

- Zabbix server e Zabbix frontend instalados: instale de acordo com as instruções para o seu sistema operacional (consulte Installation from packages e Web interface installation).
- Zabbix agent instalado, se estiver monitorando métricas locais da rede.
- 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 por humanos. Sem o suporte adequado a MIB, você poderá ver apenas valores numéricos, o que dificulta 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 `/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 <your_community_string> <device_IP>`) e confira se os OIDs são exibidos com nomes descritivos.

Dependendo do seu ambiente, algumas etapas deste guia podem diferir ligeiramente. Este guia é baseado em um ambiente executando Ubuntu e em um dispositivo de rede Cisco Catalyst 3750V2-24FS sendo monitorado.

Pressupõe-se que o seu dispositivo de rede já esteja fisicamente instalado e conectado.

Configure 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 conta configurações existentes. Cuidado: aplicar estes comandos pode substituir as configurações atuais de SNMP.

Para um roteador Cisco, a configuração normalmente envolve as etapas descritas abaixo.

Exemplo de SNMPv2

1. Ative o SNMP e defina a community string.

Faça login no console do seu roteador Cisco e entre no modo de configuração:

```
configure terminal
```

Em seguida, ative o SNMP especificando uma community string somente leitura. Por exemplo:

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

Substitua `<your_community_string>` pela sua community string 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 ao SNMP apenas aos dispositivos necessários, por motivos de segurança. Para 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 de SNMP permaneçam após uma reinicialização:

```
write memory
```

Exemplo de SNMPv3

O SNMPv3 oferece segurança aprimorada com autenticação e criptografia. Sua configuração é mais segura do que a do SNMPv2 e deve ser verificada na documentação específica do seu dispositivo.

1. Crie um grupo SNMP.

Configure um grupo SNMPv3 com privacidade (criptografia) habilitada:

```
configure terminal
snmp-server group <your_group> v3 priv
```

2. Crie um usuário SNMP.

Adicione um usuário SNMPv3 com autenticação e privacidade. Substitua os placeholders pelos valores desejados:

```
snmp-server user <your_user> <your_group> v3 auth md5 <auth_password> priv aes 128 <priv_password>
```

3. Salve a configuração:

```
write memory
```

Para mais detalhes ou instruções específicas do modelo, você pode consultar tutoriais externos de configuração SNMP da Cisco. Este guia, no entanto, fornece as etapas básicas para habilitar o monitoramento SNMP.

Configurar o frontend do Zabbix

Criar um host no frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Adicione um novo host.

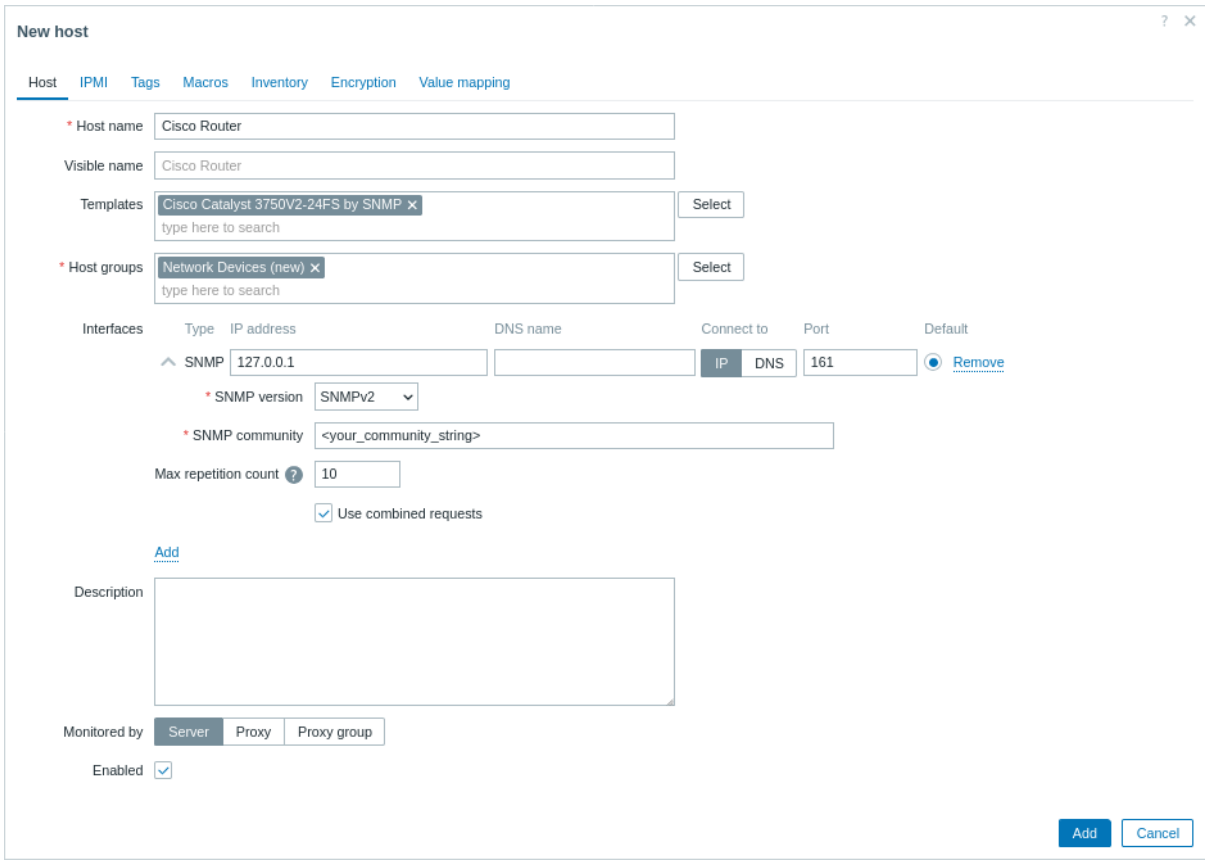
Navegue até *Data collection* > *Hosts* e clique em *Create host*.

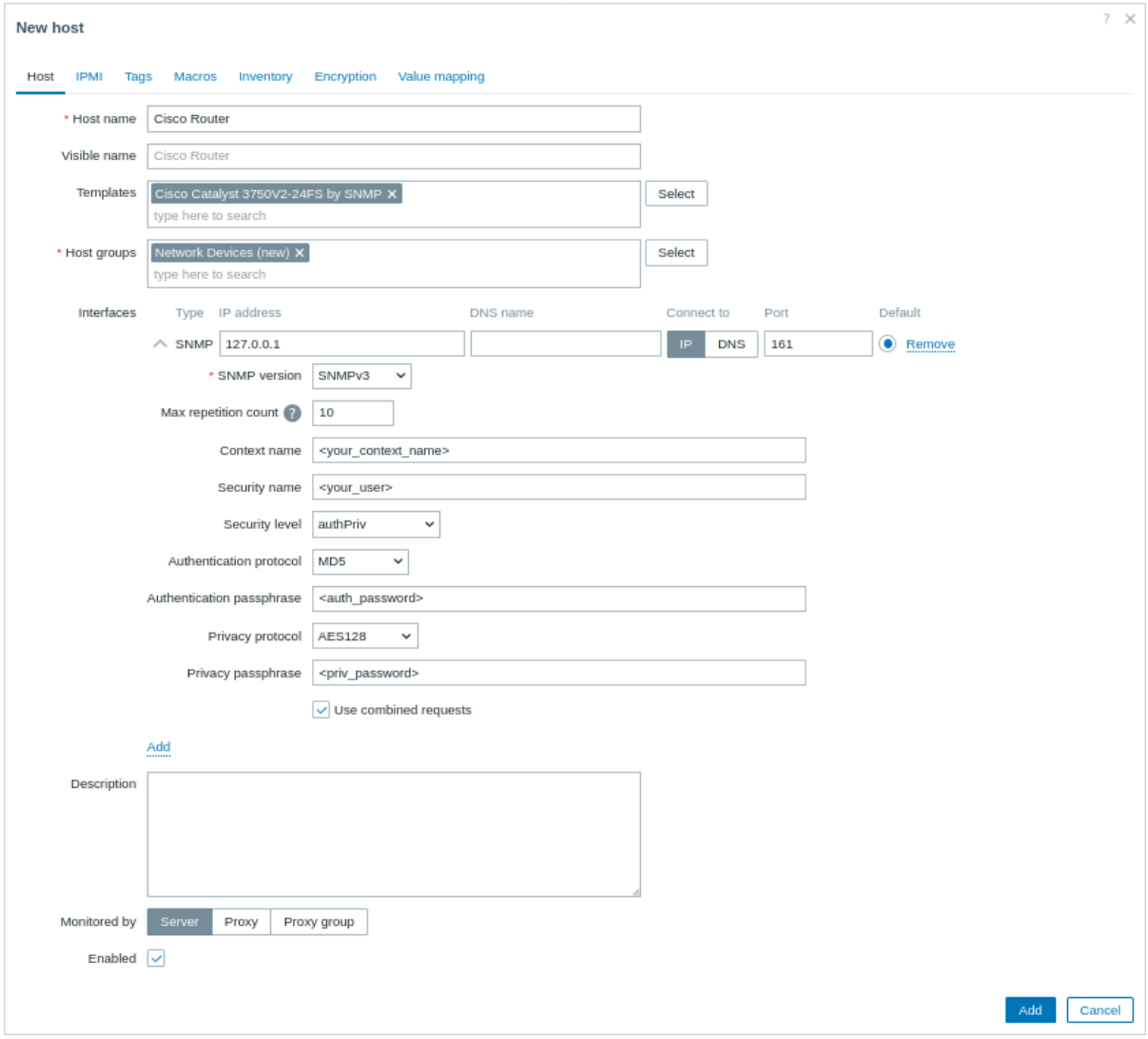
- *Host name*: informe um nome para o seu dispositivo (por exemplo, ”Cisco Router”).
- Host groups: selecione um grupo existente ou crie um novo grupo, como ”Network Devices”.
- Interfaces:
  - Clique em *Add* em Interfaces.
  - Escolha *SNMP* como tipo de interface.
  - Informe o endereço IP ou o nome DNS do seu roteador Cisco.
  - Defina a porta SNMP padrão (normalmente 161).
  - Use o menu suspenso para selecionar a versão SNMP apropriada (por exemplo, SNMPv2).
  - Para SNMPv1/v2, informe a community string no campo *SNMP community*. Para SNMPv3, credenciais adicionais (*Context name*, *Security name* e *Security level*, etc.) serão solicitadas.

3. Vincular templates

No campo *Templates*, selecione o template SNMP que melhor corresponda ao seu dispositivo. O Zabbix fornece uma variedade de templates SNMP pré-criados para muitas famílias de dispositivos. Por exemplo, se você estiver monitorando um dispositivo Cisco, escolha o template que corresponde ao sistema operacional ou ao modelo do seu dispositivo (como Cisco IOS SNMP ou Cisco Catalyst 3750<device model> SNMP).

4. Clique em *Add* para salvar o host.



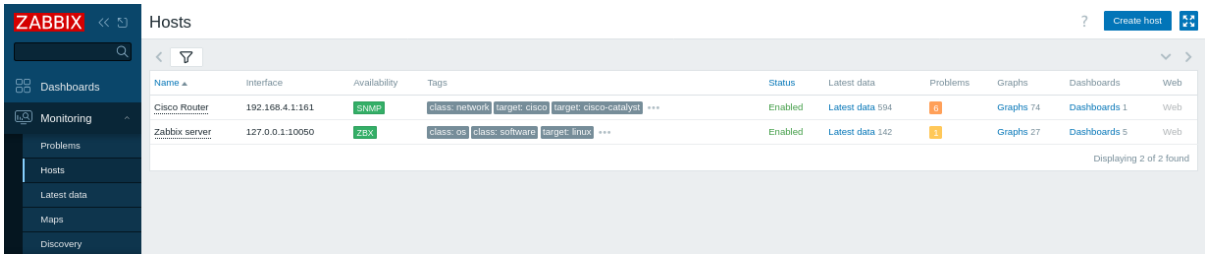


Ver métricas coletadas

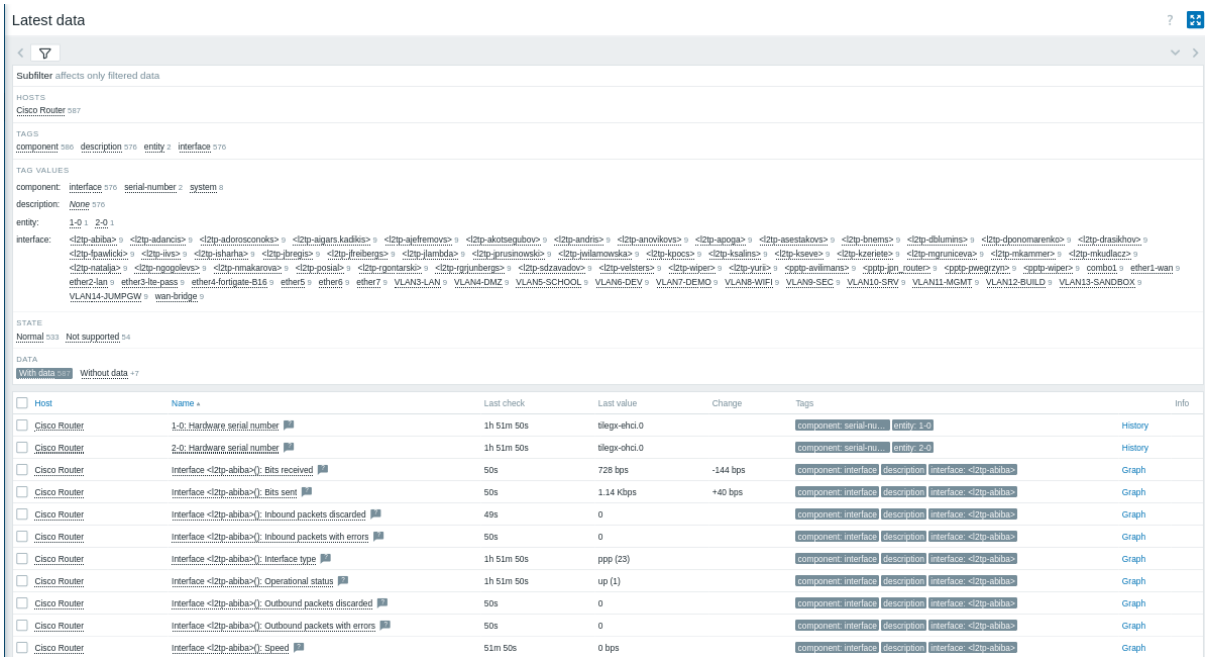
Parabéns! O Zabbix agora está configurado para monitorar seu dispositivo de rede.

Dados mais recentes:

- Navegue até Monitoring > Latest data no frontend do Zabbix.



- Selecione seu host ”Cisco Router” (ou hosts descobertos) para visualizar métricas como tempo de atividade de hardware e de rede, perda de ICMP, ping e tempo de resposta, etc.



- Gráficos e telas:

Para visualizar os dados de desempenho, clique em *Graphs* 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 problema 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 templates já contêm conjuntos padrão de items.

1. Identifique o OID SNMP:

Use o comando `snmpwalk` para listar os OIDs disponíveis no seu dispositivo. Por exemplo:

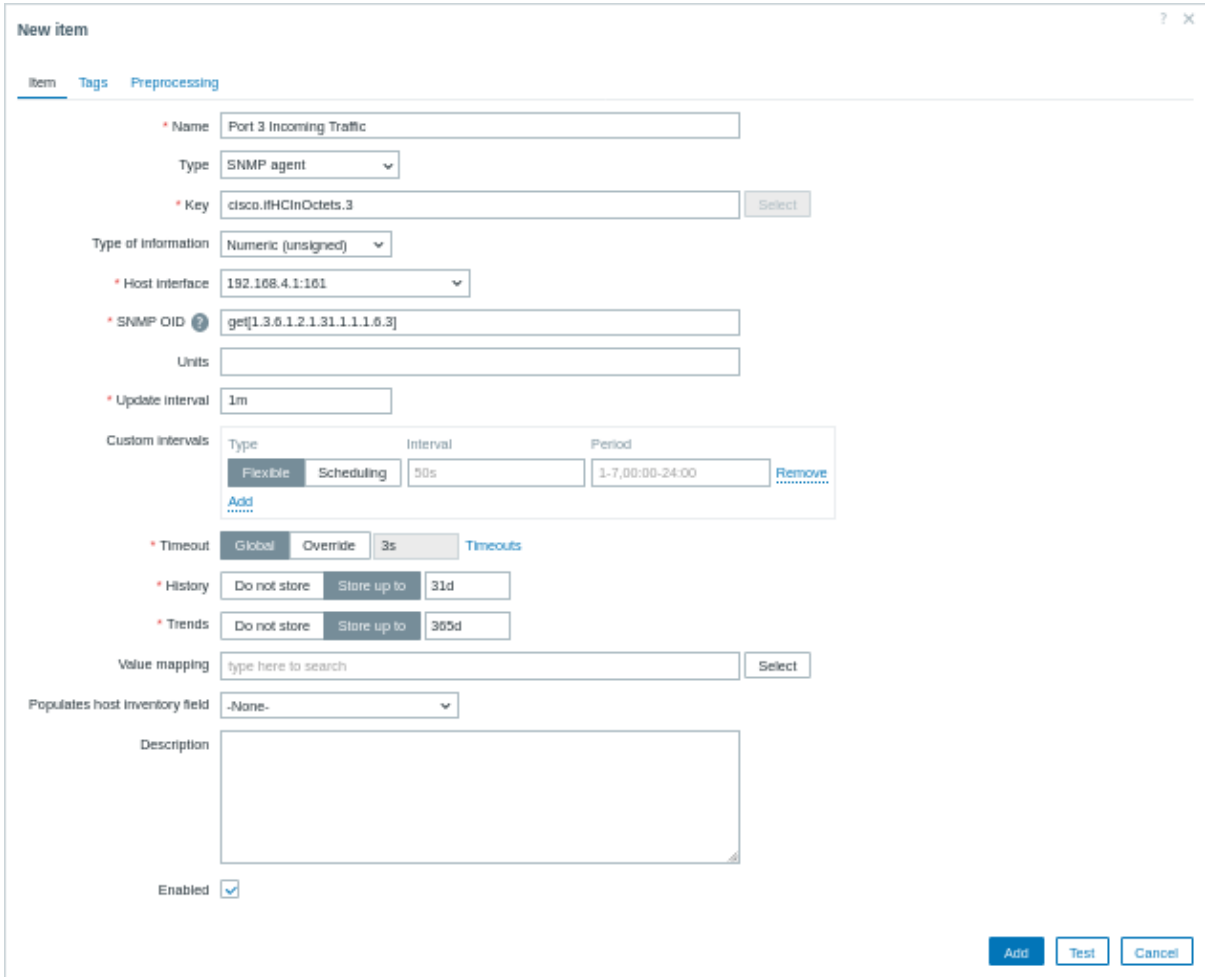
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Encontre o OID da métrica que você deseja monitorar (por exemplo, IF-MIB::ifHCInOctets.3 para tráfego de entrada na porta 3). Para obter o OID numérico, você pode usar:

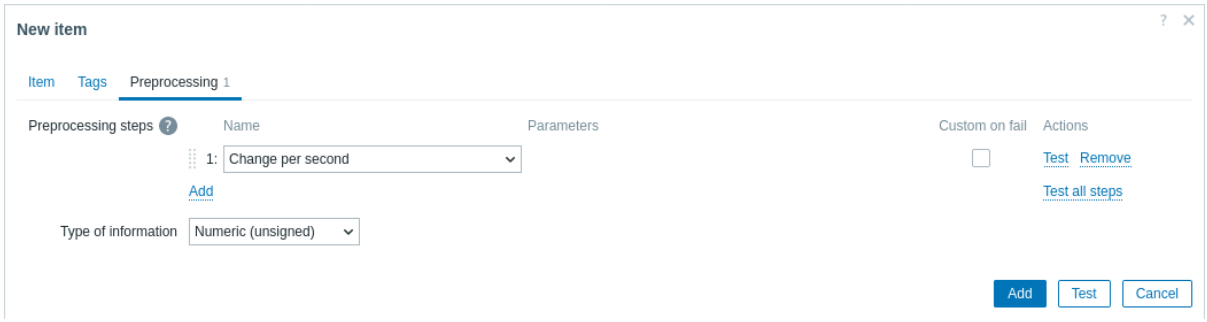
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Crie um item SNMP:

- Navegue até *Data collection* > *Hosts* e clique na aba *Items* do seu host SNMP e clique em *Create item*.
- *Name*: informe um nome descritivo (por exemplo, "Tráfego de entrada da porta 3").
- *Type*: selecione *SNMP agent*.
- *Key*: forneça uma key significativa (por exemplo, `cisco.ifHCInOctets.3`).
- *Host interface*: verifique se a interface SNMP está selecionada.
- *SNMP OID*: informe 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 de forma assíncrona uma subárvore de valores.



- *Preprocessing* (se necessário): se o item retornar um contador cumulativo (como tráfego de interface), navegue até a aba *Preprocessing* e adicione uma etapa de preprocessing como "Change per second" 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 nomes numéricos e nomes MIB

Ao trabalhar com SNMP, talvez seja necessário converter entre OIDs numéricos e seus respectivos nomes MIB. Essa tradução ajuda a identificar e solucionar problemas de métricas com mais facilidade.

- Para traduzir 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
```

Este comando pode retornar:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Para traduzir 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
```

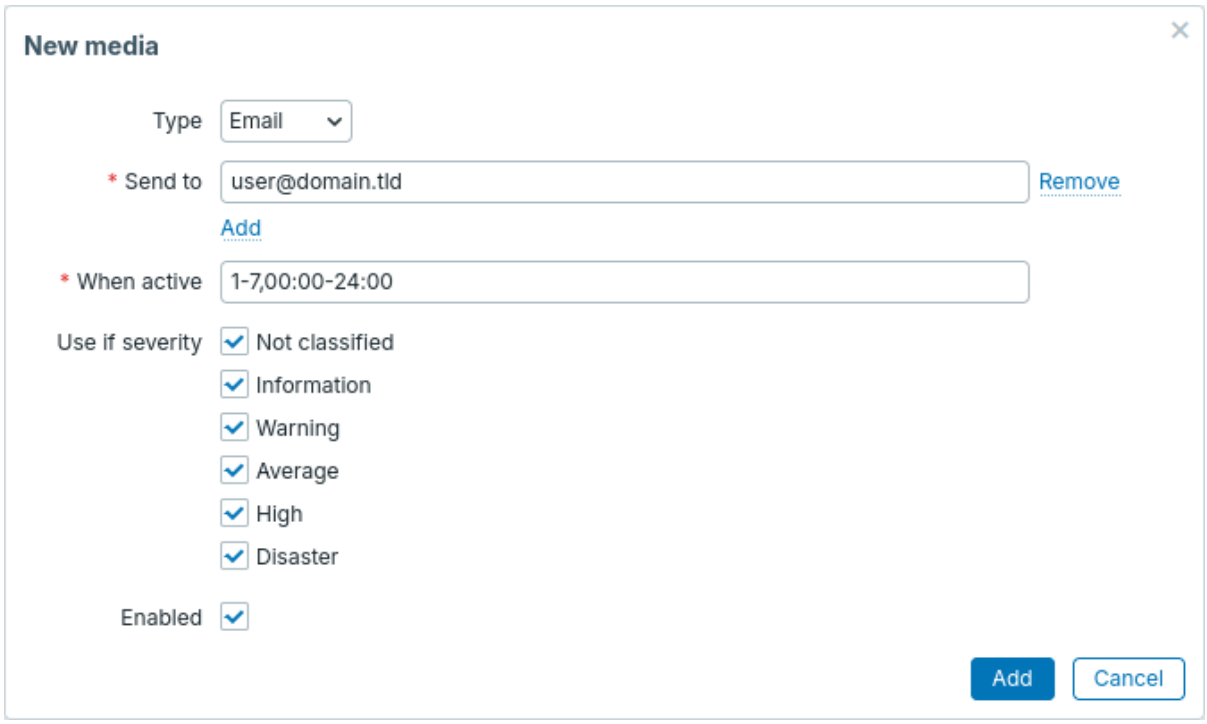
Este comando pode retornar:

```
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 *User settings* > *Profile*, altere para a aba *Media* e adicione seu e-mail.



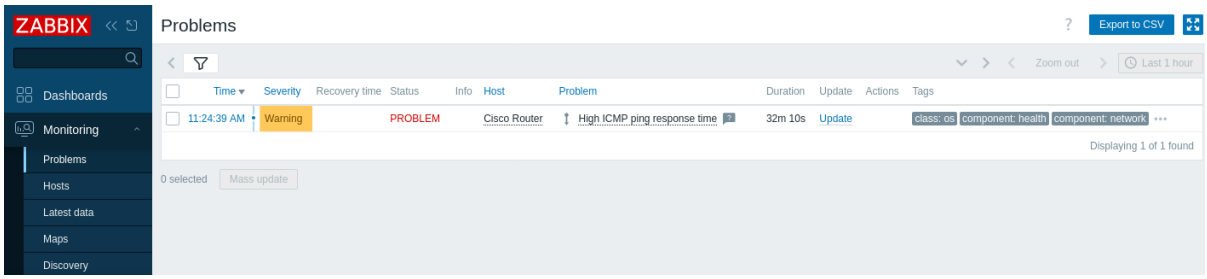
2. Siga o guia para Receiving a problem notification.

Na próxima vez em 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 host "Cisco Router" no Zabbix.
2. Navegue até a aba *Macros* e selecione *Inherited and host macros*.
3. Localize a macro `{$ICMP_RESPONSE_TIME_WARN}` (ou uma macro semelhante de limite de tempo de resposta).
4. Defina um valor muito baixo (por exemplo, 0.001) para disparar um alerta quando a resposta do ping exceder esse valor.
5. Clique em *Update* para aplicar as alterações.
6. Aguarde alguns instantes para que o Zabbix detecte o problema simulado.
7. Navegue até *Monitoring* > *Problems* para verificar se um alerta aparece (por exemplo, "High ICMP ping response time").



Se os alertas estiverem configurados, você também deverá receber um alerta de problema.

8. Reverta o valor da macro para a configuração original e clique em *Update* para salvar as alterações.
9. Confirme que o problema foi resolvido e desaparece da seção *Problems*.

Solução de problemas de monitoramento SNMP

Se você notar que o ícone de SNMP no frontend do Zabbix aparece em VERMELHO ou que nenhum dado está sendo coletado, tente as seguintes etapas:

1. Verifique a conectividade SNMP.

Para SNMPv2, execute o seguinte comando a partir do seu server Zabbix:

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

Este comando verifica se o dispositivo responde a consultas SNMP.

Para SNMPv3, inclua as credenciais apropriadas de SNMPv3:

```
snmpwalk -v3 -u <your_user> -l authPriv -a MD5 -A <auth_password> -x AES -X <priv_password> <device_IP> .
```

Isso verifica se as credenciais de 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 comando a seguir não deve retornar erro quando você 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 do SNMP e as credenciais configuradas no Zabbix correspondem às definidas no seu dispositivo. Por exemplo, revise as configurações de SNMP na configuração do host no Zabbix e compare-as com a configuração do seu dispositivo. Em um dispositivo Cisco, você pode verificar as configurações de SNMP executando:

```
show running-config | include snmp
```

Isso garante que a community string (para SNMPv2) ou os detalhes do usuário SNMPv3 estejam corretos.

4. Verifique se o SNMP está habilitado corretamente no 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 server 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 escutando 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 server Zabbix para identificar quaisquer erros relacionados a SNMP e ajudar a localizar 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.

Monitorar o log de eventos do Windows usando verificações ativas

Introdução

Este guia explica como monitorar logs de eventos do Windows com o Zabbix usando verificações ativas. Com as chaves de item específicas do Windows do Zabbix, você pode coletar e analisar eventos críticos (como tentativas de logon com falha, erros do sistema etc.) em tempo real.

Para quem este guia é destinado

Este guia foi desenvolvido para novos usuários do Zabbix e administradores de rede que desejam monitorar logs de eventos do Windows. Para opções avançadas de configuração, consulte a documentação de chaves de item específicas do Windows.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional. Você também precisa do Zabbix agent baixado e instalado na máquina Windows que deseja monitorar.

Configure o Zabbix agent para monitorar o log de eventos do Windows

1. Abra o arquivo de configuração do agent do Zabbix no seu host Windows e verifique se o parâmetro `ServerActive` está definido como o endereço IP do seu Zabbix server e se o parâmetro `Hostname` corresponde ao nome do host que será definido no Zabbix frontend. Isso permite que o agent solicite verificações ativas para seu host e do Zabbix server especificado. Por exemplo:

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

2. Reinicie o serviço do Zabbix agent 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 Zabbix agent esteja em execução no host Windows.
- Verifique se o host Windows consegue se conectar ao Zabbix server na porta 10051. Para testar a conectividade a partir do host Windows, abra o PowerShell e execute o seguinte comando:

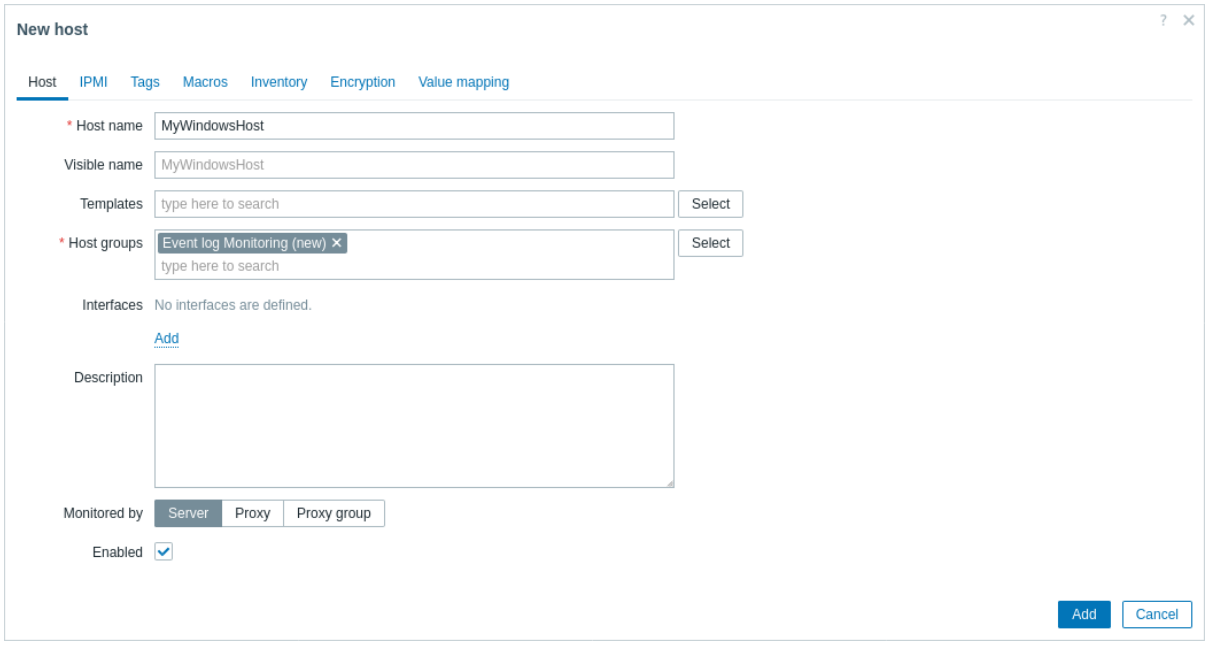
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configurar o frontend do Zabbix

1. Navegue até *Data collection* > *Hosts* e crie um host:

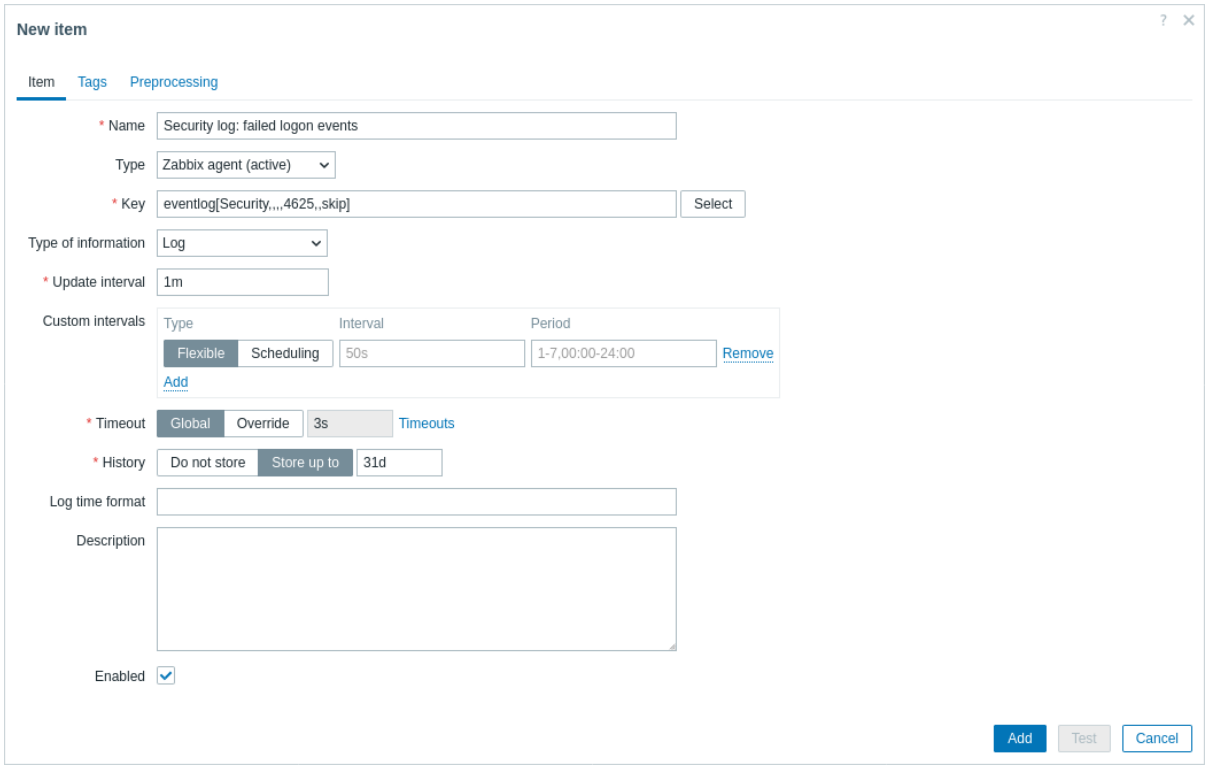
- No campo *Host name*, informe um nome de host (por exemplo, "MyWindowsHost").
- No campo *Host groups*, digite ou selecione um grupo de host (por exemplo, "Event log Monitoring").
- Clique em *Add* para salvar o host configurado.

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 *Name*, informe um nome descritivo para o item (por exemplo, ”Security log: failed logon events”).
- No menu suspenso *Type*, selecione ”Zabbix agent (active)” (obrigatório para o monitoramento de Event log).
- No campo *Key*, use a chave de item eventlog. Por exemplo, para monitorar tentativas de logon com falha (Event ID: 4625) no Security log e ignorar entradas mais antigas do que a última verificação do item (usando o parâmetro `skip`), informe a seguinte chave de item: `eventlog[Security,,,,4625,,skip]`
- No menu suspenso *Type of information*, selecione ”Log”.



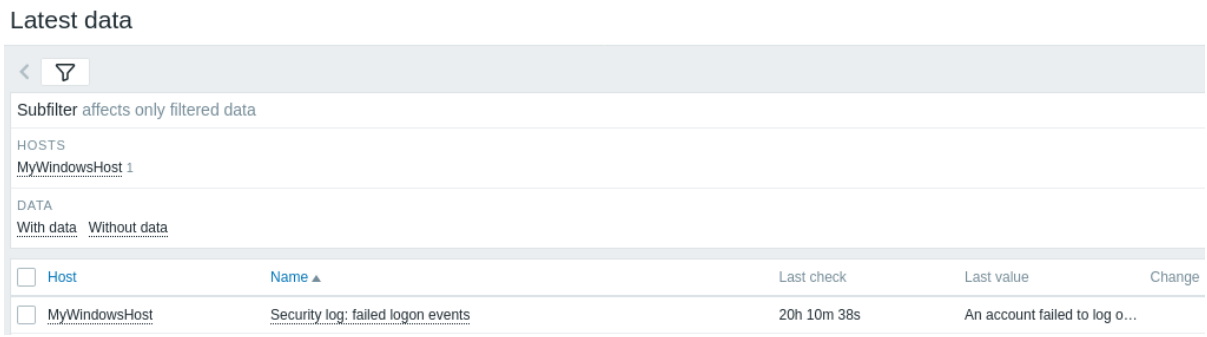
3. Clique em *Add* para salvar o item.

Testar e visualizar métricas coletadas

Parabéns! O Zabbix agora está configurado para coletar seus 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” fazendo logout 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 *Monitoring* > *Latest data* no frontend do Zabbix.



2. Filtre pelo host ”MyWindowsHost” no campo *Name*.

3. Clique em *History* 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 Troubleshooting do guia.

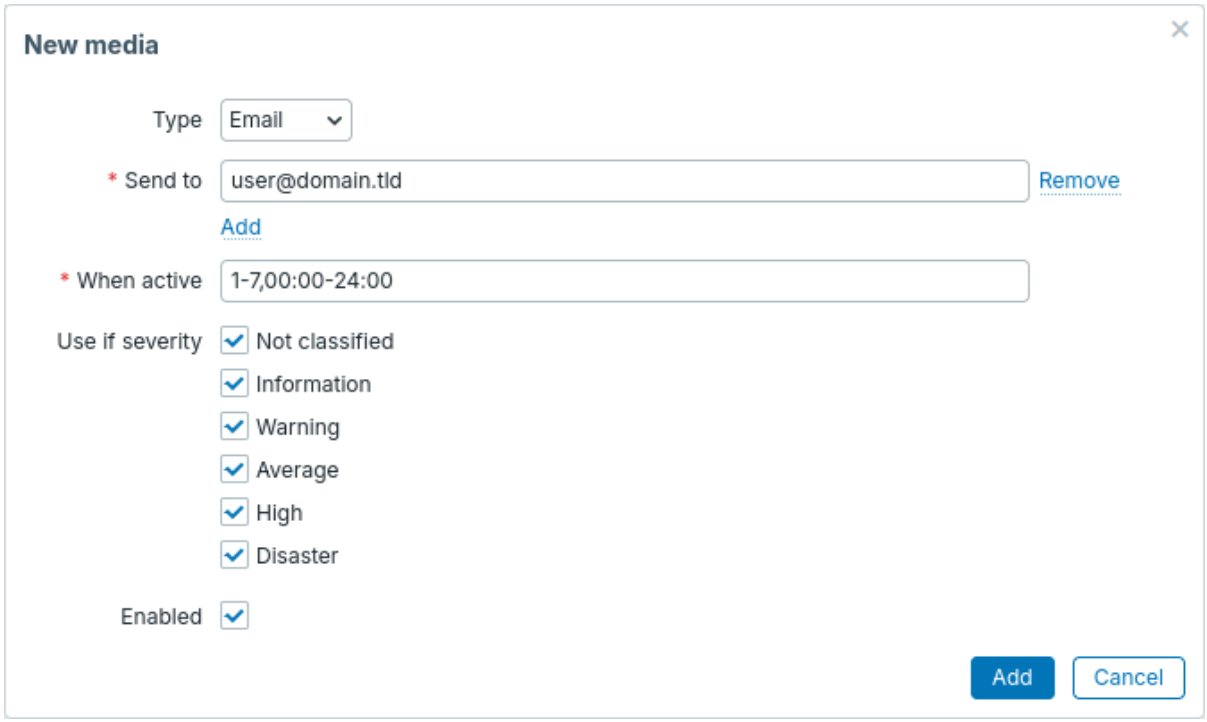
Configurar alertas de problema

Este guia fornece etapas básicas de configuração para enviar alertas por email.

1. Navegue até Coleta de dados > Hosts para definir um trigger que seja acionado quando o item de log de eventos registrar o padrão de seu interesse. Por exemplo, para detectar 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*, mude para a aba *Mídia* e adicione seu email.



3. Siga o guia para Receber uma notificação de problema.

Na próxima vez em que o Zabbix detectar um problema, você deverá receber um alerta por email.

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 de 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[...]` use exatamente o nome do log (diferenciando maiúsculas e minúsculas), o ID do evento, o modo (por exemplo, skip) e outros parâmetros exatamente como mostrado em Windows-specific item keys.

Veja também:

- Criando um item - saiba como adicionar métricas adicionais.
- Zabbix agent no Microsoft Windows - instruções detalhadas de instalação.
- Monitore Windows com Zabbix agent - um guia completo para configurar o monitoramento básico de máquinas Windows usando Zabbix agent.
- Chaves de item específicas do Windows - informações detalhadas sobre chaves de item específicas do Windows suportadas por Zabbix agents, incluindo as de monitoramento de log de eventos.
- Monitoramento de arquivos de log - instruções para configurar o Zabbix para monitoramento centralizado e análise de arquivos de log, aplicável a logs de eventos do Windows.

Instalação: guias adicionais

Se você está procurando a maneira mais fácil de instalar o Zabbix, consulte Instalação e primeiros passos.

Os guias abaixo abrangem maneiras **adicionais** ou específicas do sistema para instalar o Zabbix ou seus componentes:

- Instalação a partir de contêineres
- Instalação a partir do código-fonte
- Observações sobre a instalação a partir de pacotes
- Instalação do Zabbix agent a partir do Windows MSI
- Compilação do Zabbix agent 2 no Windows
- Compilação do Zabbix agent no Windows
- Instalação do agent no macOS a partir de PKG
- Zabbix agent no IBM AIX
- Compilação do Zabbix agent no macOS
- Zabbix appliance
- Guia de implantação do Zabbix server na AWS
- Guia de implantação do Zabbix proxy na AWS
- Guia de implantação do Zabbix server no Azure
- Guia de implantação do Zabbix proxy no Azure
- Guia de implantação no Google Cloud

Instalação a partir de containers

Visão geral

Esta página descreve como implantar o Zabbix usando Docker ou Docker Compose.

Docker Compose é a maneira mais rápida de implantar o Zabbix. Ele lê um arquivo de configuração e inicia automaticamente todos os containers que compõem uma instalação completa do Zabbix, na ordem correta.

Docker (implantação manual) alcança o mesmo resultado implantando cada componente passo a passo.

Pré-requisitos

Antes de começar, certifique-se de que o Docker (1.12.0 ou posterior) esteja instalado no seu sistema. Caso não esteja, siga o guia de instalação do Docker. Para a implantação com o Docker Compose, também é necessário o Docker Compose (2.24.0 ou posterior).

Alguns componentes do Zabbix exigem que portas específicas estejam abertas no host que executa o Docker (por exemplo, 10051/TCP para o Zabbix server e 162/UDP para traps SNMP). Consulte Requisitos para obter a lista completa de portas usadas

pelos componentes do Zabbix. Para o Zabbix server e o agent, a porta padrão pode ser alterada definindo a variável de ambiente `ZBX_LISTENPORT` no respectivo container.

Imagens Docker disponíveis

A Zabbix fornece uma imagem Docker para cada componente do Zabbix, todas publicadas no Docker Hub. Cada imagem é usada para criar um contêiner que executa esse componente.

Componente do Zabbix	Imagem Docker
Agent	zabbix/zabbix-agent
Agent 2	zabbix/zabbix-agent2
Server (MySQL)	zabbix/zabbix-server-mysql
Server (PostgreSQL)	zabbix/zabbix-server-pgsql
Interface web (Apache + MySQL)	zabbix/zabbix-web-apache-mysql
Interface web (Apache + PostgreSQL)	zabbix/zabbix-web-apache-pgsql
Interface web (Nginx + MySQL)	zabbix/zabbix-web-nginx-mysql
Interface web (Nginx + PostgreSQL)	zabbix/zabbix-web-nginx-pgsql
Proxy (SQLite3)	zabbix/zabbix-proxy-sqlite3
Proxy (MySQL)	zabbix/zabbix-proxy-mysql
Java gateway	zabbix/zabbix-java-gateway
Serviço web	zabbix/zabbix-web-service
Traps SNMP	zabbix/zabbix-snmptraps
Servidor MCP	zabbix/zabbix-mcp-server

Note:
Para usar traps SNMP, o contêiner de traps SNMP deve compartilhar um volume com o contêiner do server ou proxy do Zabbix (consulte como usar esta imagem no Docker Hub e o exemplo abaixo).

Todas as imagens dos componentes do Zabbix no Docker Hub são baseadas nas versões principais mais recentes dos sistemas operacionais compatíveis. Essas imagens são recriadas automaticamente quando as imagens do sistema operacional subjacente são atualizadas.

Tags de imagem

Cada imagem oferece suporte a tags para selecionar o sistema operacional base e a versão do Zabbix:

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

Valores `<os>` suportados:

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

Valores `<version>` suportados:

- `latest` - Versão estável mais recente do Zabbix no Alpine Linux
- `<os>-latest` - Versão estável mais recente do Zabbix no SO selecionado
- `<os>-trunk` - Build de desenvolvimento (nightly) mais recente no SO selecionado
- `<os>-X.X-latest` - Release secundária mais recente do Zabbix de uma versão principal específica do Zabbix no SO selecionado
- `<os>-X.X.*` - Release secundária específica do Zabbix no SO selecionado

Exemplos:

```
#### Zabbix proxy (SQLite3) estável mais recente no Alpine Linux:
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Zabbix proxy (SQLite3) estável mais recente no Ubuntu:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Build de desenvolvimento (nightly) mais recente do Zabbix server (MySQL) no Ubuntu:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk
```

```
#### Release secundária 8.0 mais recente do Zabbix server (MySQL) no Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

#### Versão 8.0.1 do Zabbix server (MySQL) no Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

O Docker Compose é a maneira mais rápida de implantar o Zabbix. Ele lê um arquivo de configuração (arquivo Compose) e gerencia toda a configuração — baixando imagens do Docker, criando uma rede interna entre os contêineres, configurando o armazenamento, inicializando o banco de dados e iniciando tudo na ordem correta.

O repositório de Dockerfiles oficiais do Zabbix fornece arquivos Docker Compose prontos para uso e um sistema de configuração baseado em `.env`, compatível com diferentes sistemas operacionais, backends de banco de dados e configurações dos componentes do Zabbix.

1. Clone o repositório, navegue até ele e altere para a versão 8.0:

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

2. (Opcional) Personalize a implantação com variáveis de ambiente. Esta etapa pode ser ignorada para uma configuração padrão.

3. Implante a configuração padrão — o server do Zabbix, a interface web (em execução no NGINX) e um backend de banco de dados (MySQL ou PostgreSQL), cada um em seu próprio contêiner no Alpine Linux.

```
#### Com MySQL como banco de dados:
docker compose -f ./compose.yaml up -d

#### Com PostgreSQL como banco de dados:
docker compose -f ./compose_pgsql.yaml up -d
```

Depois que os contêineres estiverem em execução (normalmente dentro de 1 a 3 minutos), o server do Zabbix será iniciado e a interface web ficará disponível para login.

> **Note:**
> Use `docker compose ps` para verificar o status dos contêineres. Todos os contêineres (exceto o contêiner `zabbix-docker-server-db-init-1`) devem apresentar o status `Up`. Se algum apresentar `Exited`, verifique os logs em busca de erros com `docker logs -f <container-name>`.

Variáveis de ambiente

O comportamento do Docker Compose e a configuração dos componentes do Zabbix podem ser personalizados usando variáveis de ambiente.

As **variáveis no nível do Compose** (definidas no arquivo `.env`) controlam quais imagens Docker, portas e intervalos de IP de rede são usados. Você pode usar essas variáveis na mesma linha, antes do comando `docker compose`, ou editar o arquivo `.env`.

Por exemplo, o comando a seguir implanta uma configuração completa do Zabbix com vários contêineres usando uma imagem baseada no Ubuntu (em vez do Alpine Linux, usado por padrão) e disponibilizando a interface web via NGINX em portas HTTP (8282) e HTTPS (8443) personalizadas:

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

As **variáveis no nível do componente** (definidas nos arquivos `env_vars/.env_<component>`) controlam a configuração dos componentes do Zabbix. Edite o arquivo `.env_<component>` correspondente antes de executar o comando `docker compose`.

Por exemplo, você pode ajustar o número de pollers passivos do Zabbix server e as configurações de cache, além de configurar o fuso horário da interface web, editando as variáveis a seguir:

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

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

> **Note:**
> As variáveis de ambiente no nível do componente correspondem aos parâmetros de configuração dos componentes do Zabbix, mas usam um estilo de nomenclatura diferente (por exemplo, `ZBX_STARTPOLLERS` corresponde a `StartPollers`). Algumas variáveis são específicas do Docker, e alguns parâmetros de configuração não podem ser alterados (por exemplo, `PIDFile` e `LogType`). Ao usar variáveis de ambiente, consulte a seção *Variáveis de ambiente* na descrição da imagem Docker de cada componente no Docker Hub.

Volumes

O Docker Compose armazena dados persistentes no diretório `zbx_env/`, criado ao lado do arquivo Compose. Esse diretório preserva os dados dos componentes entre reinicializações e atualizações dos contêineres.

O conteúdo de `zbx_env/` é predefinido para a imagem de cada componente. Por exemplo:

- `zbx_env/etc/` - usado pelo contêiner da interface web do Zabbix
- `zbx_env/usr/` e `zbx_env/var/` - usados pelo contêiner do server Zabbix

Para mais informações sobre cada volume, consulte a seção *Allowed volumes* na descrição da imagem Docker de cada componente no Docker Hub.

Makefile

O repositório de Dockerfiles do Zabbix também fornece um Makefile como atalho para tarefas comuns do Docker Compose. Em vez de usar o comando completo `docker compose`, você pode usar comandos `make` mais curtos (execute `make help` para ver as opções disponíveis):

```
#### Implantação padrão (Zabbix server, web frontend, MySQL, tudo no Alpine Linux)
make up

#### Implantação personalizada (Zabbix server, web frontend com portas NGINX personalizadas, PostgreSQL, t
make up \
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

> **Attention:**
> Ao parar ou remover containers, sempre especifique o mesmo tipo de banco de dados usado na implantação (por exemplo, `make down DB=pgsql`).

Por padrão, `make up` inicia apenas um conjunto mínimo de serviços (Zabbix server, web frontend e banco de dados). Isso mantém a configuração leve e evita iniciar componentes que talvez não sejam necessários. Para incluir componentes adicionais, você pode usar os seguintes perfis do Compose:

```
#### Implantação padrão + Zabbix agent, Java gateway, serviço web e traps SNMP:
make up COMPOSE_PROFILES=full

#### COMPOSE_PROFILES=full + Zabbix proxy (MySQL e SQLite3):
make up COMPOSE_PROFILES=all
```

Docker (implantação manual)

Use a implantação manual quando quiser implantar o Zabbix passo a passo, executar componentes individuais, integrá-los a um ambiente existente ou usar um runtime de contêiner alternativo, como o Podman.

Por exemplo, para implantar um proxy do Zabbix, execute o seguinte comando:

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

```
  -d \
  zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

Este comando:

- Faz o pull da imagem Docker `zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest`.
- Cria e inicia um contêiner `zabbix-proxy-sqlite3` com base na imagem Docker (com as flags `--init` e `-d`).
- Configura o parâmetro `Server` do proxy do Zabbix por meio da variável de ambiente `ZBX_SERVER_HOST`. Esse parâmetro especifica o endereço IP do server do Zabbix do qual o proxy recuperará os dados de configuração e para o qual enviará os dados coletados. Observe que outras variáveis necessárias para a operação de proxy ativo usam valores padrão e podem ser omitidas.

Note:
As variáveis de ambiente correspondem aos parâmetros de configuração dos componentes do Zabbix usando um estilo de nomenclatura diferente (por exemplo, `ZBX_SERVER_HOST` corresponde a `Server`). Algumas variáveis são específicas do Docker, e alguns parâmetros de configuração não podem ser alterados (por exemplo, `PIDFile` e `LogType`). Ao usar variáveis de ambiente, consulte a seção *Environment variables* na descrição da imagem Docker de cada componente no Docker Hub.

- Conecta um local de armazenamento gerenciado pelo Docker (por exemplo, `/var/lib/docker/volumes/zabbix-proxy-data/`) ao diretório `/var/lib/zabbix` do contêiner, para que os dados do proxy do Zabbix sejam armazenados permanentemente, mesmo que o contêiner seja removido.

Note:
Para mais informações sobre cada volume, consulte a seção *Allowed volumes* na descrição da imagem Docker de cada componente no Docker Hub.

Depois de implantar o contêiner do proxy do Zabbix, você pode prosseguir para adicionar o proxy na interface web do Zabbix e configurar hosts para serem monitorados por esse proxy.

Os exemplos abaixo abrangem três cenários adicionais de implantação:

- server do Zabbix (MySQL) com Java gateway
- server do Zabbix (PostgreSQL) com traps SNMP
- server do Zabbix (MySQL) com Java gateway no RHEL 8–10

Para mais exemplos, consulte a descrição da imagem Docker de cada componente no Docker Hub.

Implantar o server Zabbix (MySQL) com Java gateway

Este exemplo mostra como implantar um server Zabbix com backend MySQL, um Java gateway para monitoramento JMX e uma interface web baseada em NGINX.

1. Crie uma rede Docker dedicada para que todos os contêineres dos componentes do Zabbix possam alcançar uns aos outros pelo nome do contêiner:

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

2. Execute um contêiner vazio do server MySQL (substitua `zabbix_pwd` e `root_pwd` por senhas fortes e use esses valores nas etapas seguintes):

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

3. Acompanhe os logs do MySQL e aguarde até que o MySQL esteja pronto para conexões (use `Ctrl+C` para sair dos logs):

```
docker logs -f mysql-server

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

4. Execute um contêiner do Java gateway do Zabbix:

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

5. Habilite `log_bin_trust_function_creators` no contêiner do server MySQL. Isso relaxa temporariamente as restrições de segurança sobre funções armazenadas; essa configuração será desabilitada nas etapas posteriores. Isso permite que o usuário do banco de dados do Zabbix crie funções armazenadas sem o privilégio SUPER, exigido pelo MySQL quando o registro binário está habilitado (por padrão desde o MySQL 8.0). Execute o comando a seguir:

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

6. Execute um contêiner do server Zabbix e vincule-o aos contêineres do Java Gateway e do server 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 ZBX_JAVAGATEWAY="zabbix-java-gateway" \
  --network=zabbix-net \
  -p 10051:10051 \
  --restart unless-stopped \
  --init \
  -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

7. Acompanhe os logs do server Zabbix e aguarde até que o server Zabbix termine de inicializar o esquema do banco de dados (use Ctrl+C para sair dos logs):

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

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

8. Desabilite `log_bin_trust_function_creators` no contêiner do server MySQL. Isso restabelece a restrição de segurança que impede usuários que não têm SUPER de criar funções armazenadas. Execute o comando a seguir:

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

9. Execute um contêiner da interface web do Zabbix e vincule-o aos contêineres do server Zabbix e do server MySQL:

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

Depois que os contêineres estiverem em execução, o server Zabbix será iniciado e a interface web estará disponível para login.

> **Note:**
> Use `docker ps` para verificar o status dos contêineres. Todos os contêineres devem apresentar o status Up. Se algum apresentar Exited, verifique os logs em busca de erros com `docker logs -f <container-name>`.

Implantar o server Zabbix (PostgreSQL) com traps SNMP

Este exemplo mostra como implantar um server Zabbix com backend PostgreSQL, traps SNMP e uma interface web baseada em NGINX.

1. Crie uma rede Docker dedicada para que todos os contêineres dos componentes do Zabbix possam alcançar uns aos outros pelo nome do contêiner:

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

2. Execute um contêiner vazio do server PostgreSQL (substitua `zabbix_pwd` por uma senha forte e use esse valor nas etapas seguintes):

```
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. Execute um contêiner de traps SNMP do Zabbix:

```
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. Execute um contêiner do server Zabbix e vincule-o aos contêineres do server PostgreSQL e de traps SNMP:

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

5. Acompanhe os logs do server Zabbix e aguarde até que o server Zabbix termine de inicializar o esquema do banco de dados (use `Ctrl+C` para sair dos logs):

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

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

6. Execute um contêiner da interface web do Zabbix e vincule-o aos contêineres do server Zabbix e do server PostgreSQL:

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

Depois que os contêineres estiverem em execução, o server Zabbix será iniciado e a interface web estará disponível para login.

> **Note:**
> Use `docker ps` para verificar o status dos contêineres. Todos os contêineres devem apresentar o status Up. Se algum apresentar Exited, verifique os logs em busca de erros com `docker logs -f <container-name>`.

Implantar o server Zabbix (MySQL) com Java gateway no RHEL 8–10

Este exemplo mostra como implantar um server Zabbix com backend MySQL, um Java gateway para monitoramento JMX e uma interface web baseada em NGINX, todos executados no Red Hat Enterprise Linux 8, 9 ou 10.

No Red Hat Enterprise Linux, o runtime de contêineres recomendado é o Podman, em vez do Docker. O Podman funciona de maneira semelhante ao Docker, mas não exige um serviço em segundo plano executado como root, o que o torna mais adequado para ambientes Red Hat.

1. Crie um novo pod com o nome `zabbix` e exponha as portas da interface web do Zabbix e do trapper do server Zabbix:

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

2. (Opcional) Execute um contêiner do agent Zabbix no 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-80:latest
```

3. Crie um diretório `./mysql/` no host RHEL:

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

4. Execute um contêiner vazio do server MySQL (substitua `zabbix_pwd` e `root_pwd` por senhas fortes e use esses valores nas etapas seguintes):

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

5. Acompanhe os logs do MySQL e aguarde até que o MySQL esteja pronto para conexões (use `Ctrl+C` para sair dos logs):

```
podman logs -f mysql-server

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

6. Execute um contêiner do Java gateway do Zabbix:

```
podman run --name zabbix-java-gateway -t \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-80
```

7. Habilite `log_bin_trust_function_creators` no contêiner do server MySQL. Isso relaxa temporariamente as restrições de segurança sobre funções armazenadas; essa configuração será desabilitada nas etapas posteriores. Isso permite que o usuário do banco de dados do Zabbix crie funções armazenadas sem o privilégio `SUPER`, exigido pelo MySQL quando o registro binário está habilitado (por padrão desde o MySQL 8.0). Execute o comando a seguir:

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

8. Execute um contêiner do server Zabbix e vincule-o aos contêineres do Java Gateway e do server MySQL:

```
podman run --name zabbix-server-mysql -t \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
```

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

9. Acompanhe os logs do server Zabbix e aguarde até que o server Zabbix termine de inicializar o esquema do banco de dados (use Ctrl+C para sair dos logs):

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

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

10. Desabilite log_bin_trust_function_creators no contêiner do server MySQL. Isso restabelece a restrição de segurança que impede usuários que não têm SUPER de criar funções armazenadas. Execute o comando a seguir:

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

11. Execute um contêiner da interface web do Zabbix e vincule-o aos contêineres do server Zabbix e do server MySQL:

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

Note:
O pod zabbix expõe a porta 80/TCP (HTTP) para a máquina host a partir da porta 8080/TCP do contêiner zabbix-web-mysql.

Depois que os contêineres estiverem em execução, o server Zabbix será iniciado e a interface web estará disponível para login.

Note:
Use docker ps para verificar o status dos contêineres. Todos os contêineres devem apresentar o status Up. Se algum apresentar Exited, verifique os logs em busca de erros com docker logs -f <container-name>.

Instalação a partir do código-fonte

Você pode obter a versão mais recente do Zabbix compilando-a a partir do código-fonte. Veja também Obtendo o código-fonte do Zabbix.

Um tutorial passo a passo para instalar o Zabbix a partir do código-fonte é fornecido aqui.

Instalando os daemons do Zabbix

1 Baixe o arquivo de origem

Acesse a página de download do Zabbix e baixe o arquivo de origem. Após o download, extraia os arquivos de origem executando:

```
tar -zxvf zabbix-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 sem privilégios. Se um daemon do Zabbix for iniciado a partir de uma conta de usuário sem privilégios, ele continuará sendo executado como esse usuário.

Na configuração padrão, se um daemon for iniciado como `root`, ele mudará para a conta de usuário `zabbix`, que deve estar presente. Para criar um usuário e grupo de sistema `zabbix`, execute os comandos listados abaixo.

Sistema baseado em RedHat:

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

Sistema baseado em Debian:

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

Não é necessário criar uma conta de usuário separada para o frontend do Zabbix.

Recomendação de segurança

Se o server e o agent do Zabbix forem executados na mesma máquina, recomenda-se executá-los sob **contas de usuário separadas**. Executar ambos como o mesmo usuário permite que o agent acesse o arquivo de configuração do server, o que pode expor informações confidenciais—como a senha do banco de dados—para qualquer usuário com nível de Admin no Zabbix.

> **Attention:**
> Executar o Zabbix como `root`, `bin` ou qualquer outra conta com direitos especiais é um risco de segurança.

Diretório home (opcional)

Os processos do Zabbix não exigem um diretório home, portanto, geralmente não é recomendada a criação de um. No entanto, se você precisar de uma funcionalidade que exija um diretório home (por exemplo, armazenar credenciais do MySQL em $HOME/.my.cnf), você pode criá-lo usando os comandos listados abaixo.

Em sistemas baseados em RedHat, execute:

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

Em sistemas baseados em Debian, execute:

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

3 Criar o banco de dados do Zabbix

Para os processos do Zabbix server e proxy, bem como para o frontend do Zabbix, é necessário um banco de dados. Não é necessário para executar o Zabbix agent.

Scripts SQL são fornecidos para criar o esquema do banco de dados e inserir o conjunto de dados. O banco de dados do proxy Zabbix precisa apenas do esquema, enquanto o banco de dados do server Zabbix requer também o conjunto de dados além do esquema.

Tendo criado um banco de dados Zabbix, prossiga para as etapas seguintes de compilação do Zabbix.

4 Configure as fontes

C99 com extensões GNU é necessário para compilar o Zabbix server, o Zabbix proxy ou o Zabbix agent. Essa versão pode ser especificada explicitamente 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 as 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 compatíveis, dentro do diretório de fontes extraídas do Zabbix, execute:

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

Para configurar as 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 as fontes para um Zabbix server (com PostgreSQL etc.), você pode executar:

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

Para configurar as 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 as fontes para um Zabbix agent, você pode executar:

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

ou, para o Zabbix agent 2:

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

Note:
Uma versão compatível do Go deve estar instalada para compilar o Zabbix agent 2.

Observações sobre as opções de compilação:

- `--enable-agent` - compila o Zabbix agent, além dos utilitários de linha de comando Zabbix get e Zabbix sender.
- `--with-libcurl` - necessário para monitoramento de máquinas virtuais, autenticação SMTP e items do Zabbix agent `web.page.*`, além de items de consulta de telemetria. Consulte também: Requisitos (libcurl).
- `--with-libxml2` - necessário para monitoramento de máquinas virtuais.
- `--with-libpcre2[=DIR]` - o Zabbix sempre compila com a biblioteca PCRE2; essa opção permite apenas especificar um caminho de instalação personalizado do PCRE2.
- `--with-mysql=/path/to/mysql_config` - especifica o caminho para a configuração de uma biblioteca cliente MySQL específica. Útil quando várias versões do MySQL ou MariaDB estão instaladas.
- `--enable-static` - vincula bibliotecas estaticamente (não compatível com o Solaris). Use essa opção se 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=<value>` - define o tamanho da pilha por thread em kilobytes (por exemplo, `--with-stacksize=512`). Você pode aumentar esse valor se o Zabbix travar ou congelar devido a estouros de pilha (por exemplo, durante o pré-processamento em sistemas com limites padrão baixos para a pilha de threads).
- `--with-apm` (use somente com `--enable-proxy`) - necessário para o monitoramento de desempenho de aplicações (APM) no Zabbix proxy. Consulte também: Requisitos (g++, gRPC, protobuf, Abseil).

Attention:
Se `./configure` falhar devido à ausência de bibliotecas ou a outros problemas, verifique o arquivo `config.log` para obter informações detalhadas sobre o erro.
Por exemplo, se `libssl` estiver ausente, a mensagem de erro imediata poderá ser enganosa:

```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```

Nesse caso, `config.log` revela a causa real:

```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Consulte também:

- Compilando o Zabbix com suporte à criptografia
- Problemas de compilação conhecidos

5 Fazer e instalar 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 por um usuário com permissões suficientes (comumente 'root', ou usando sudo).

Ao executar `make install`, por padrão serão instalados os binários do daemon (zabbix_server, zabbix_agentd, zabbix_proxy) em /usr/local/sbin e os binários do cliente (zabbix_get, zabbix_sender) em /usr/local/bin.

Note:
Para especificar um local diferente de `/usr/local`, use uma chave `--prefix` na etapa anterior de configuração das fontes, por exemplo, `--prefix=/home/zabbix`. Nesse caso, os binários do daemon serão instalados em `<prefix>/sbin`, enquanto os utilitários em `<prefix>/bin`. As páginas de manual serão instaladas em `<prefix>/share`.

6 Revisar e editar arquivos de configuração

- edite o arquivo de configuração do agent do Zabbix (`zabbix_agentd.conf`).

Você precisa configurar esse arquivo para cada host com o `zabbix_agentd` instalado.

É necessário especificar o **endereço IP** do server do Zabbix no arquivo. As conexões de outros hosts serão negadas.

- edite o arquivo de configuração do server do Zabbix (`zabbix_server.conf`).

É necessário especificar o nome do banco de dados, o usuário e a senha (se estiverem sendo usados).

Os demais parâmetros atenderão às suas necessidades com os valores padrão se você tiver uma instalação pequena (até dez hosts monitorados). No entanto, altere os parâmetros padrão se quiser maximizar o desempenho do server (ou proxy) do Zabbix.

- se você tiver instalado um proxy do Zabbix, edite o arquivo de configuração do proxy (`zabbix_proxy.conf`).

É necessário especificar o endereço IP do server e o nome do host do proxy (que deve ser conhecido pelo server), além do nome do banco de dados, do usuário e da senha (se estiverem sendo usados).

Note:
Com o SQLite, o caminho completo para o arquivo do banco de dados deve ser especificado; o usuário e a 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 proxy Zabbix, execute o zabbix_proxy.

`zabbix_proxy`

Instalando a interface web do Zabbix

Copiando arquivos PHP

O frontend do Zabbix é escrito em PHP, portanto, para executá-lo, é necessário um servidor web com suporte a PHP. A instalação é feita simplesmente copiando os arquivos PHP do diretório ui para o diretório de documentos HTML do servidor web.

Os locais comuns dos diretórios de documentos HTML para servidores web Apache incluem:

- /usr/local/apache2/htdocs (diretório padrão ao instalar o Apache a partir do código-fonte)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

É recomendável 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 você planeja usar qualquer idioma diferente do inglês, consulte Instalação de idiomas adicionais do frontend para obter instruções.

Instalando o frontend

Consulte a página Instalação da interface web para obter informações sobre o assistente de instalação do frontend do Zabbix.

Instalando plugins carregáveis do Zabbix agent 2

A instalação de plugins carregáveis do Zabbix agent 2 só é necessária se você quiser monitorar alvos não cobertos pelos plugins integrados (por exemplo, servidores ou clusters MongoDB, PostgreSQL e seus forks etc.). Veja a lista completa de plugins carregáveis e plugins integrados.

Attention:
Antes de instalar um plugin, verifique o arquivo README dele. Ele pode conter requisitos específicos e instruções de instalação.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo compactado com o código-fonte do plugin carregável.

Para compilar o plugin, navegue até o diretório do plugin extraído e execute `make`:

```
make
```

Note:
Uma versão compatível do Go deve estar instalada para compilar plugins carregáveis do Zabbix agent 2.

O executável do plugin pode ser colocado em qualquer local, desde que possa ser carregado pelo Zabbix agent 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=/path/to/executable/zabbix-agent2-plugin-postgresql
```

O caminho para o arquivo de configuração do plugin deve ser especificado no parâmetro `Include` do arquivo de configuração do Zabbix agent 2:

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

Acesse configuração para mais detalhes sobre a configuração de plugins.

Os plugins carregáveis fornecidos pelo Zabbix usam makefiles simples com os seguintes alvos de compilação:

- `make` - compila o plugin
- `make clean` - exclui todos os arquivos criados pela compilação do plugin
- `make check` - executa autotestes (requer um alvo de monitoramento real, por exemplo, um banco de dados PostgreSQL)
- `make style` - verifica o estilo do código Go com `golangci-lint`
- `make format` - formata o código Go com `go fmt`
- `make dist` - cria um arquivo-fonte incluindo todas as dependências

Instalando o Java gateway

É necessário instalar o Java gateway somente se você quiser monitorar aplicações JMX. O Java gateway é leve e não requer um banco de dados.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo-fonte.

Para compilar o Java gateway, execute o script `./configure` com a opção `--enable-java`. É recomendável especificar a opção `--prefix` para definir um caminho de instalação diferente do padrão /usr/local, porque a instalação do Java gateway criará toda uma árvore de diretórios, e 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 path.

```
make
```

Agora você tem um arquivo zabbix-java-gateway-$VERSION.jar em `src/zabbix_java/bin`. Se você se sentir confortável em executar o Java gateway a partir de `src/zabbix_java` no diretório de distribuição, então pode prosseguir para as instruções de configuração e execução do Java gateway. Caso contrário, certifique-se de ter permissões suficientes e execute `make install`.

```
make install
```

Consulte configuração para mais detalhes sobre como configurar e executar o Java gateway.

Instalando o Zabbix web service

A instalação do Zabbix web service só é necessária se você quiser usar relatórios agendados.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo-fonte.

Para compilar o Zabbix web service, execute o script `./configure` com a opção `--enable-webservice`.

> **Note:**
> Uma versão compatível do Go deve estar instalada para compilar o Zabbix web service.

Execute zabbix_web_service na máquina onde o web service está instalado:

```
zabbix_web_service
```

Consulte configuração para mais detalhes sobre a configuração da geração de relatórios agendados.

Instalando o Zabbix MCP server

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo do código-fonte.

Para compilar o Zabbix MCP server, execute o seguinte comando no diretório do código-fonte do MCP server:

```
make build
```

> **Note:**
> Uma versão compatível do Go deve estar instalada para compilar o Zabbix MCP server.

Para informar uma versão específica na saída de `zabbix_mcp_server -V` e no `User-Agent` das solicitações à API do Zabbix, defina a variável `VERSION`, por exemplo, `make build VERSION=8.0.0`.

Para instalar o binário em `/usr/local/sbin` e o arquivo de configuração de exemplo em `/etc/zabbix`, execute:

```
sudo make install
```

Edite o arquivo de configuração `/etc/zabbix/zabbix_mcp_server.conf` e defina pelo menos os seguintes parâmetros:

- `FrontendURL` - a URL da API do Zabbix, terminando com `api_jsonrpc.php`;
- `AllowedIP` - os endereços dos clientes MCP que têm permissão para se conectar.

Para criptografar as conexões dos clientes MCP, defina `TLSAccept=cert` e especifique `TLSCertFile` e `TLSKeyFile`, ou execute o MCP server atrás de um proxy reverso confiável que gerencie o TLS.

Se o usuário `zabbix` não existir (por exemplo, se apenas o MCP server estiver instalado na máquina), crie-o conforme descrito em Criar conta de usuário.

Valide o arquivo de configuração e inicie o zabbix_mcp_server:

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

Consulte MCP server para obter mais detalhes sobre como configurar o server e conectar clientes MCP.

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 as versões suportadas 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
```

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 `/usr/lib/x86_64-linux-gnu/libcurl.so` mais antiga, mas está disponível na `/usr/local/lib/libcurl.so` mais recente.

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 de stack muito pequeno em alguns sistemas

Se o Zabbix travar ou congelar devido a estouros de stack, você pode aumentar o tamanho da stack por thread usando a opção `--with-stacksize` ao configurar as fontes. Esse problema pode ocorrer em sistemas com limites padrão baixos de stack de thread, especialmente durante o preprocessing, quando várias threads são criadas.

O exemplo a seguir define o tamanho da stack para 512 KB por thread:

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

Você pode verificar os limites de stack de thread do sistema em tempo de execução usando o comando `ulimit -s` em sistemas baseados em Linux.

Notas sobre a instalação a partir de pacotes

Visão geral

Os pacotes oficiais de **pré-lançamento** do Zabbix 8.0 estão disponíveis no site do Zabbix, onde você pode selecionar seu sistema operacional e o componente do Zabbix para gerar instruções e comandos de instalação.

Além das notas sobre a instalação de pacotes abaixo, consulte também a página Instalação e configuração.

Os pacotes estão disponíveis para as seguintes distribuições Linux:

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

Attention:
Algumas distribuições de SO, especialmente as baseadas em Debian, fornecem seus próprios pacotes do Zabbix. Esses pacotes **não são compatíveis** com o Zabbix e podem estar desatualizados ou não incluir os recursos e as correções de bugs mais recentes. Recomenda-se usar somente os pacotes oficiais do Repositório oficial do Zabbix. Se você instalou anteriormente o Zabbix a partir do repositório do seu sistema operacional, consulte as etapas para atualizar os pacotes do Zabbix a partir dos repositórios do SO.

Os pacotes são compatíveis com bancos de dados MySQL/PostgreSQL e servidores 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 estiverem instalados no mesmo host.

Se necessário, pacotes separados para o 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; consulte:

- Instalação do agent do Windows a partir de MSI
- Instalação do agent do macOS a partir de PKG
- Zabbix agent no IBM AIX
- Binários legados (para sistemas mais antigos ou menos comuns, como HP-UX, NetBSD, Tru64 e versões mais antigas do SLES)

Observações sobre a instalação de pacotes

As observações a seguir se aplicam a todos os sistemas:

- Se estiver usando PostgreSQL, `DBHost=localhost` (ou um endereço IP) na configuração do server/proxy do Zabbix faz com que o PostgreSQL use um socket de rede em vez de um socket UNIX local; consulte Configuração do SELinux para obter instruções relacionadas à configuração.
- Se estiver usando TimescaleDB, consulte a configuração adicional do TimescaleDB.
- Se estiver instalando o Java gateway do Zabbix (para monitorar aplicações JMX), consulte as instruções adicionais de configuração para sistemas baseados em RHEL e sistemas baseados em Debian.
- Se estiver instalando o MCP server do Zabbix, instale o pacote `zabbix-mcp-server` e atualize o arquivo de configuração (pelo menos `FrontendURL` e `AllowedIP`) antes de iniciar o serviço.
- Para executar o Zabbix agent como root, consulte Executando o agent como root.

As observações a seguir se aplicam ao RHEL e seus derivados:

- Se você tiver habilitado o repositório EPEL para EL9, que também fornece pacotes do Zabbix, ele deverá ser excluído da resolução de pacotes antes da instalação dos pacotes oficiais do Zabbix; consulte Instalação acidental de pacotes EPEL do Zabbix.
- Para instalar pacotes do Zabbix em ambientes Red Hat UBI, consulte Pacotes do Zabbix para RHEL em ambientes Red Hat UBI.
- Para usar itens de ping ICMP, os pacotes do `fping` também estão disponíveis no Repositório Oficial do Zabbix.

Configuração do SELinux

O Zabbix usa comunicação entre processos baseada em sockets. Em sistemas onde o Security-Enhanced Linux (SELinux) está habilitado, talvez seja necessário adicionar regras do SELinux para permitir que o Zabbix crie/use sockets de domínio UNIX no diretório SocketDir. Arquivos de socket são usados pelo server do Zabbix (alerter, preprocessing, IPMI) e pelo proxy do Zabbix (IPMI), e ficam presentes enquanto o processo está em execução.

Com o SELinux habilitado no modo enforcing, execute os seguintes comandos para permitir a comunicação entre o frontend do Zabbix e o server:

Para RHEL 7 (e posteriores), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (e posteriores):

```
setsebool -P httpd_can_connect_zabbix on
```

Se o banco de dados for acessado pela rede (incluindo `localhost` para PostgreSQL), permita também que o frontend do Zabbix se conecte ao banco de dados:

```
setsebool -P httpd_can_network_connect_db on
```

Para RHEL anterior à versão 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 `zabbix-selinux-policy` predefinido do Zabbix Official Repository. Esse pacote é fornecido para todas as versões de SO suportadas para simplificar a implantação do Zabbix e evitar que os usuários desativem o SELinux por causa da complexidade da configuração.
 Para a versão 10 do AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux, a política do SELinux é fornecida pelo pacote `zabbix-selinux-policy-main`; o pacote `zabbix-selinux-policy` tem uma dependência de `zabbix-selinux-policy-main` ou `selinux-policy-targeted-extra`. O pacote `zabbix-selinux-policy`

entra em conflito com `selinux-policy-targeted-extra`; portanto, ambos os pacotes não podem ser instalados ao mesmo tempo.

> **Attention:**
> Para máxima segurança, é recomendável definir configurações personalizadas do SELinux.

O pacote `zabbix-selinux-policy` (`zabbix-selinux-policy-main` para a versão 10 do AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux) contém uma política básica do SELinux que permite ao Zabbix criar e usar sockets e habilita a conexão do 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 avançada de problemas.

Observe que os pacotes `Debuginfo` estão disponíveis para as seguintes versões: Red Hat Enterprise Linux 7 e anteriores (incluindo derivados) e SUSE Linux Enterprise Server 15 e anteriores (incluindo derivados).

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

Depois de 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 por componente:

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

Habilitando repositórios de versões instáveis

As instruções abaixo são para habilitar repositórios de lançamento instáveis do Zabbix (desativados 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 no 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
```

Instalação do agent do Windows a partir do MSI

Visão geral

O agent do Zabbix pode ser instalado no Windows usando pacotes de instalação MSI de 32 ou 64 bits, disponíveis para download.

Os requisitos mínimos do sistema operacional para instalação via MSI são:

- **Para o agent do Zabbix:** Windows XP (64 bits) ou Windows Server 2003
- **Para o agent 2 do Zabbix:** Windows 10 (32 bits) ou Windows Server 2016

Pacotes de 32 bits não podem ser instalados em sistemas de 64 bits.

Os pacotes incluem:

- Suporte a TLS (a configuração de TLS é opcional)
- Utilitários Zabbix get e Zabbix sender (podem ser instalados junto com o agent/agent 2 do Zabbix ou separadamente)

> **Attention:**
> Os pacotes do agent 2 do Zabbix não incluem plugins carregáveis (MongoDB, PostgreSQL, MSSQL), que precisam ser baixados e instalados separadamente.

A instalação pode ser feita usando o Assistente de Instalação ou a linha de comando.

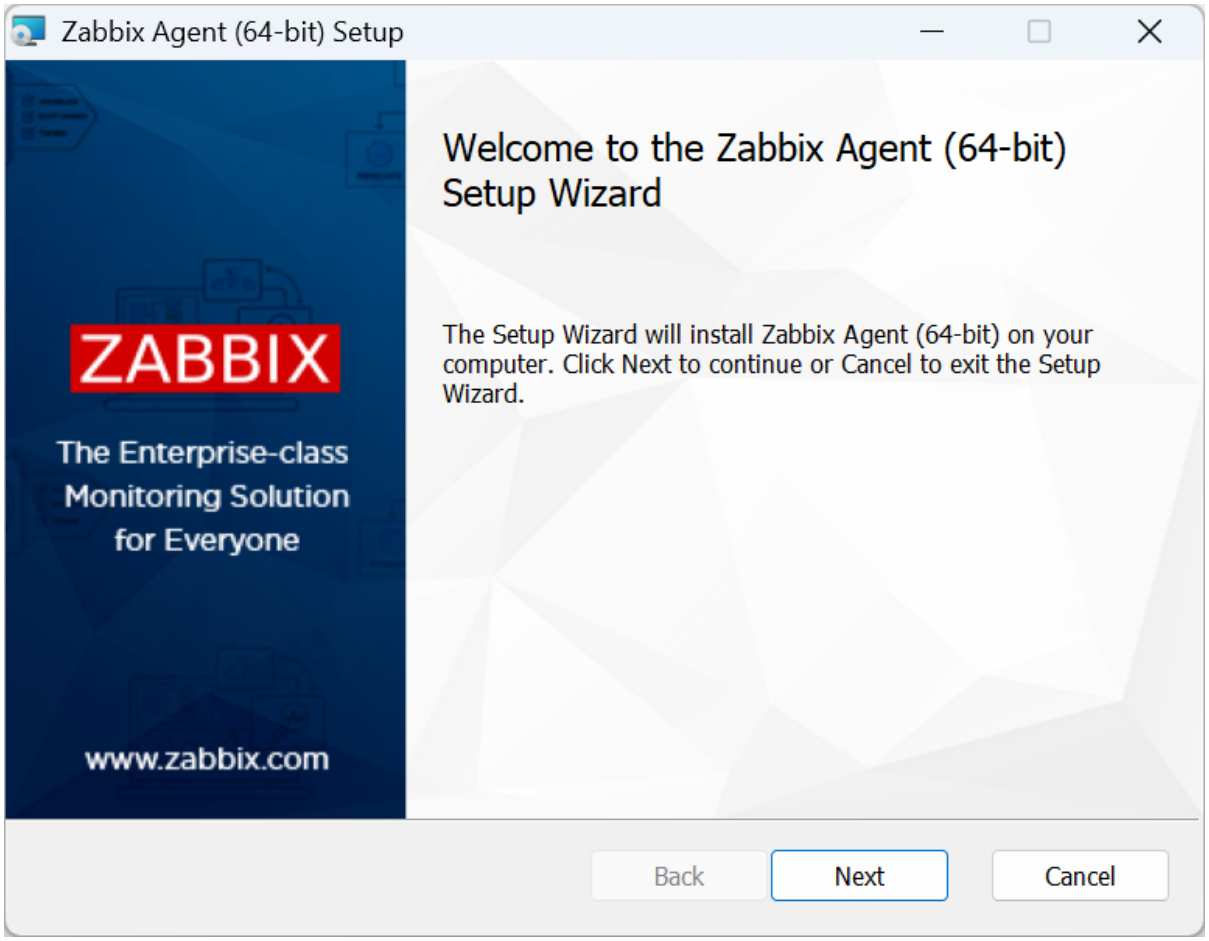
Embora a instalação usando pacotes MSI seja totalmente suportada, recomenda-se instalar pelo menos o Microsoft .NET Framework 2 para um tratamento adequado de erros.

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

Instalação pelo assistente de configuração

As etapas de instalação a seguir se aplicam tanto ao Zabbix agent quanto ao Zabbix agent 2.

1. Clique duas vezes no arquivo MSI baixado para iniciar a instalação:

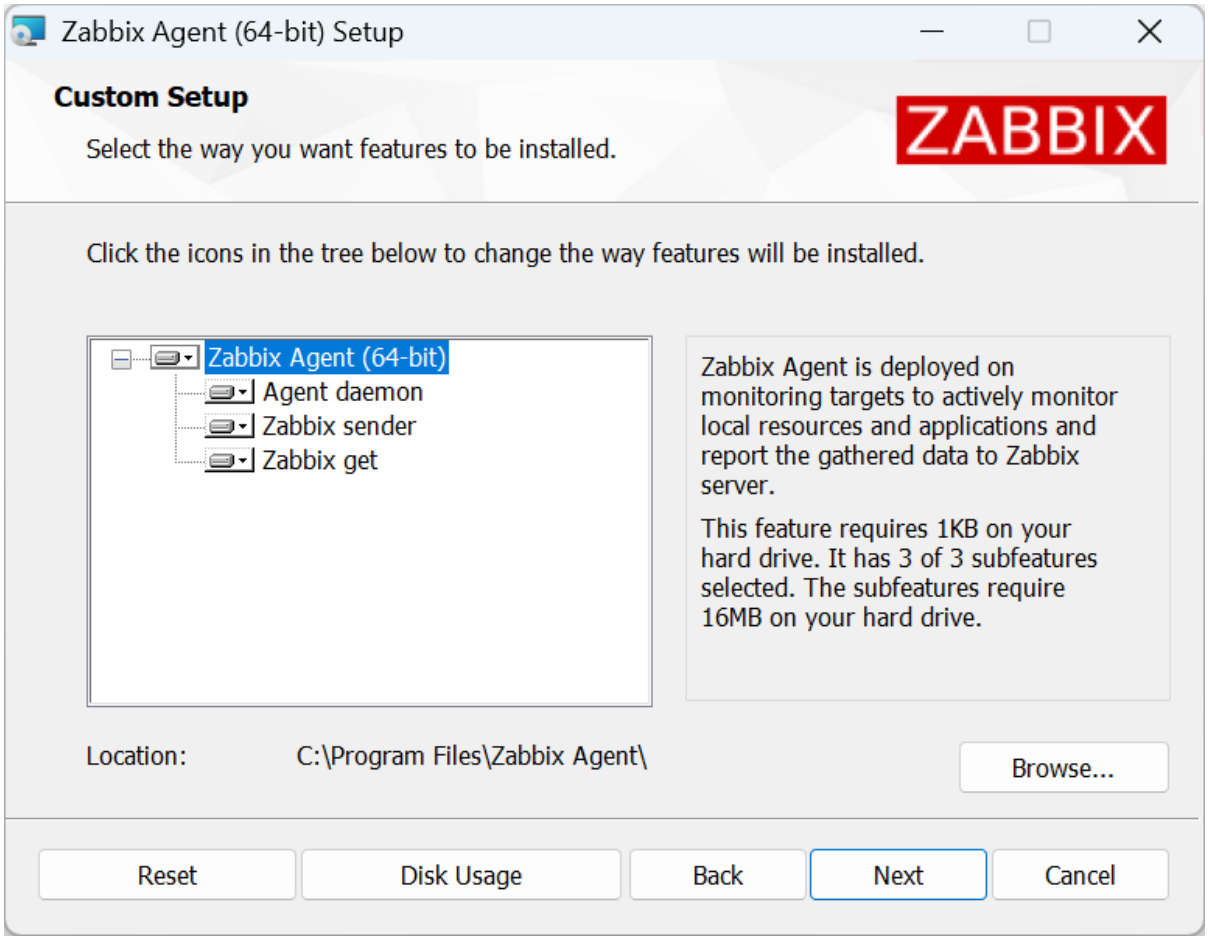


2. Aceite o Contrato de Licença do Usuário Final:



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

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

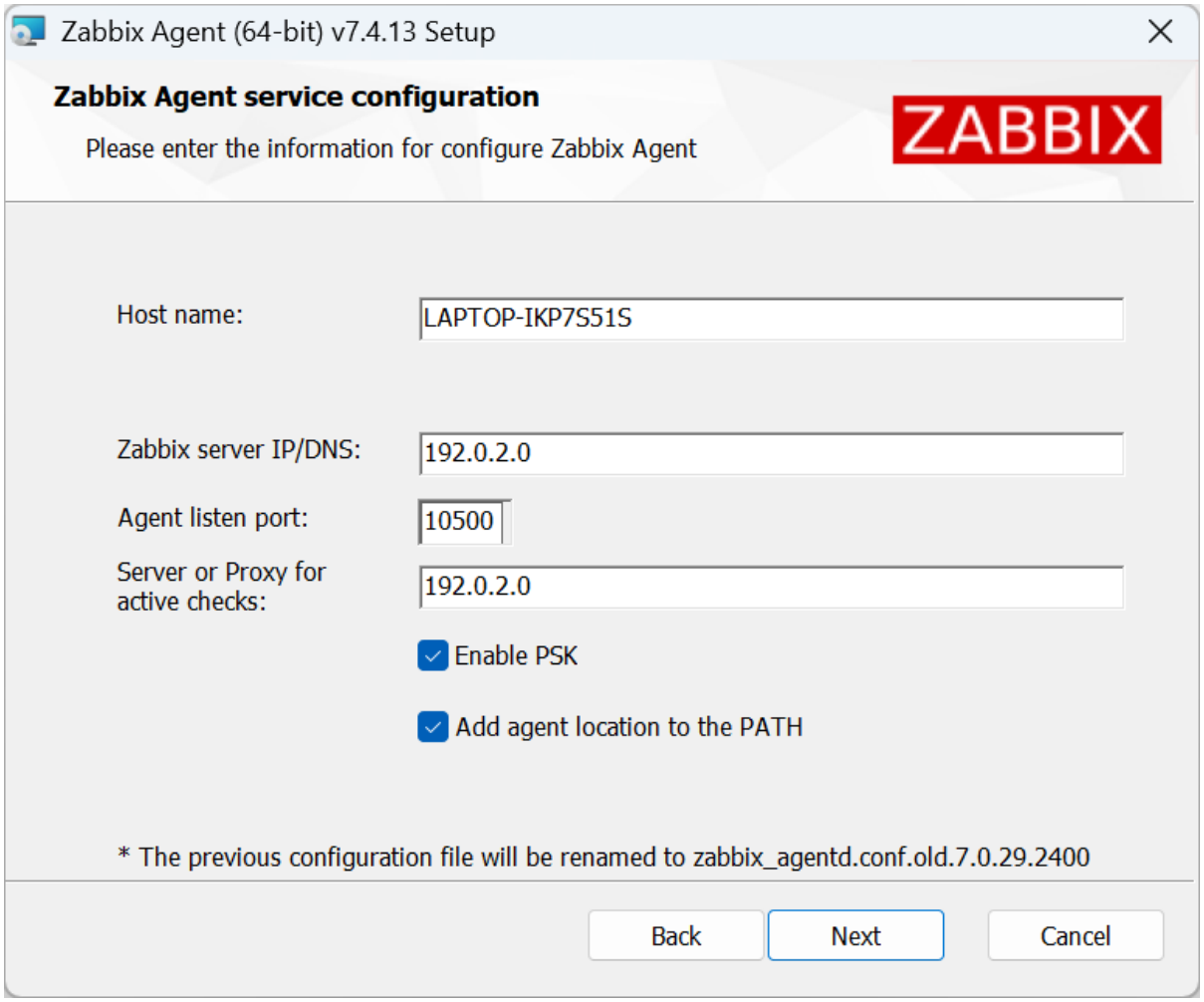


Note:
Se for usado um local personalizado fora do diretório `Program Files` e a pasta de destino já existir, o instalador não modificará suas permissões. Durante uma nova instalação, você será solicitado a definir manualmente as permissões apropriadas ou continuar sem modificá-las.

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

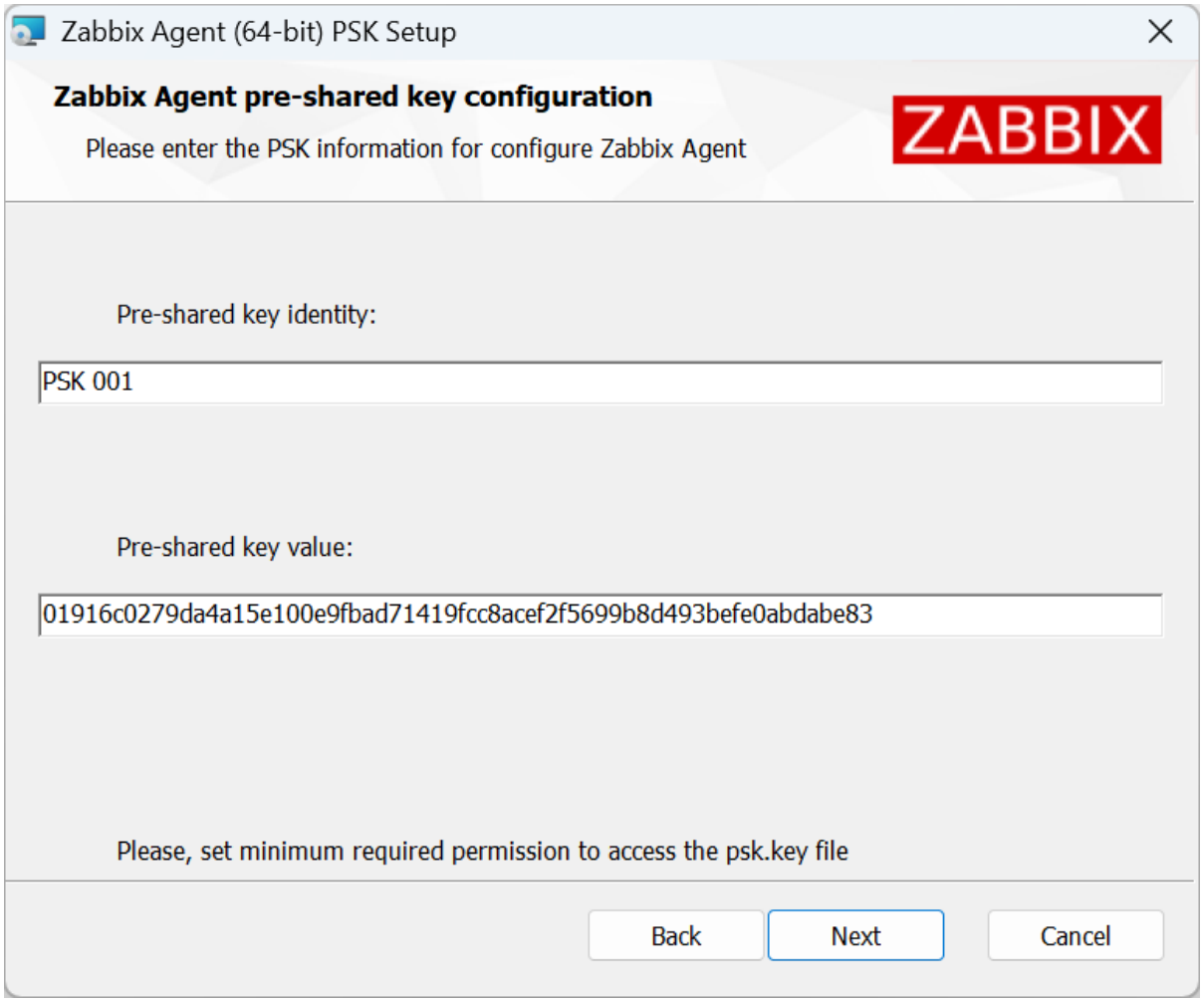
Parâmetro	Descrição
Nome do host	O nome do host da máquina na qual o Zabbix agent está sendo instalado. Define o parâmetro Hostname.
IP/DNS do server do Zabbix	Uma lista de endereços IP separados por vírgulas, opcionalmente em notação CIDR, ou nomes DNS de servers ou proxies do Zabbix. Este parâmetro é **obrigatório**. Define o parâmetro Server.
Porta de escuta do agent	O agent escutará nesta porta por conexões do server. Define o parâmetro ListenPort.
Server ou proxy para verificações ativas	O endereço do server/proxy do Zabbix ou a configuração de cluster da qual serão obtidas as verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e uma porta opcional separados por dois-pontos. Define o parâmetro ServerActive.
Ativar PSK	Marque a caixa de seleção para ativar o suporte a TLS usando chaves pré-compartilhadas. Define os parâmetros TLSConnect e TLSAccept como `psk`.
Adicionar o local do agent ao 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 do 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 de PSK se você marcou a caixa de seleção *Ativar PSK* na etapa anterior. Esses parâmetros também serão definidos no arquivo de configuração do Zabbix agent:

Parâmetro	Descrição
Identidade da chave pré-compartilhada	A string de identidade da chave pré-compartilhada. Define o parâmetro TLSPSKIdentity.
Valor da chave pré-compartilhada	O valor da string da chave pré-compartilhada. Cria o arquivo `psk.key` contendo a chave e define o parâmetro TLSPSKFile como o local da chave (padrão: `C:\Program Files\Zabbix Agent\psk.key`). É **recomendável** restringir o acesso ao arquivo da chave pré-compartilhada ajustando as configurações de segurança do arquivo para que somente o Zabbix agent (ou o usuário que executa o agent) possa lê-lo.

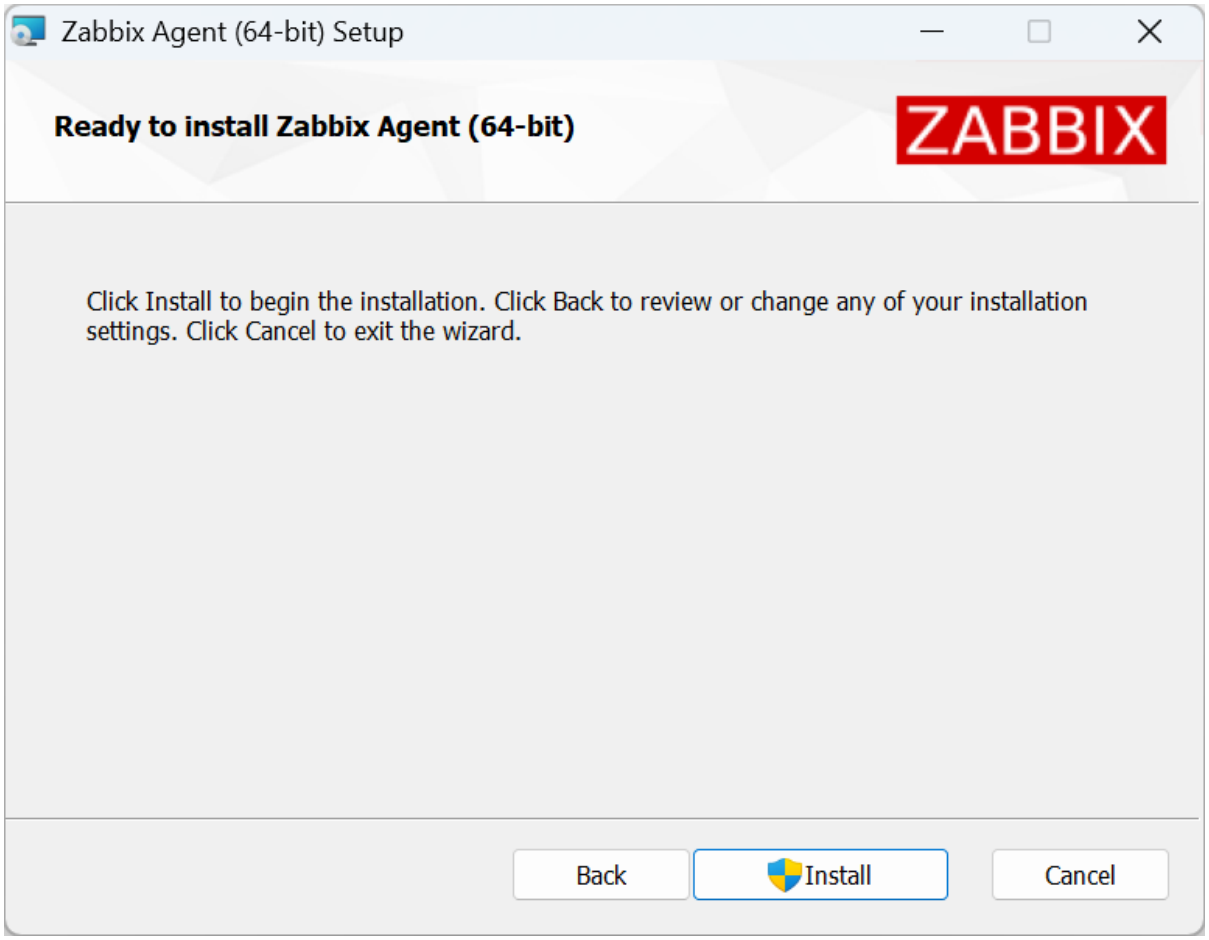


6. Clique em *Instalar* para iniciar a instalação.

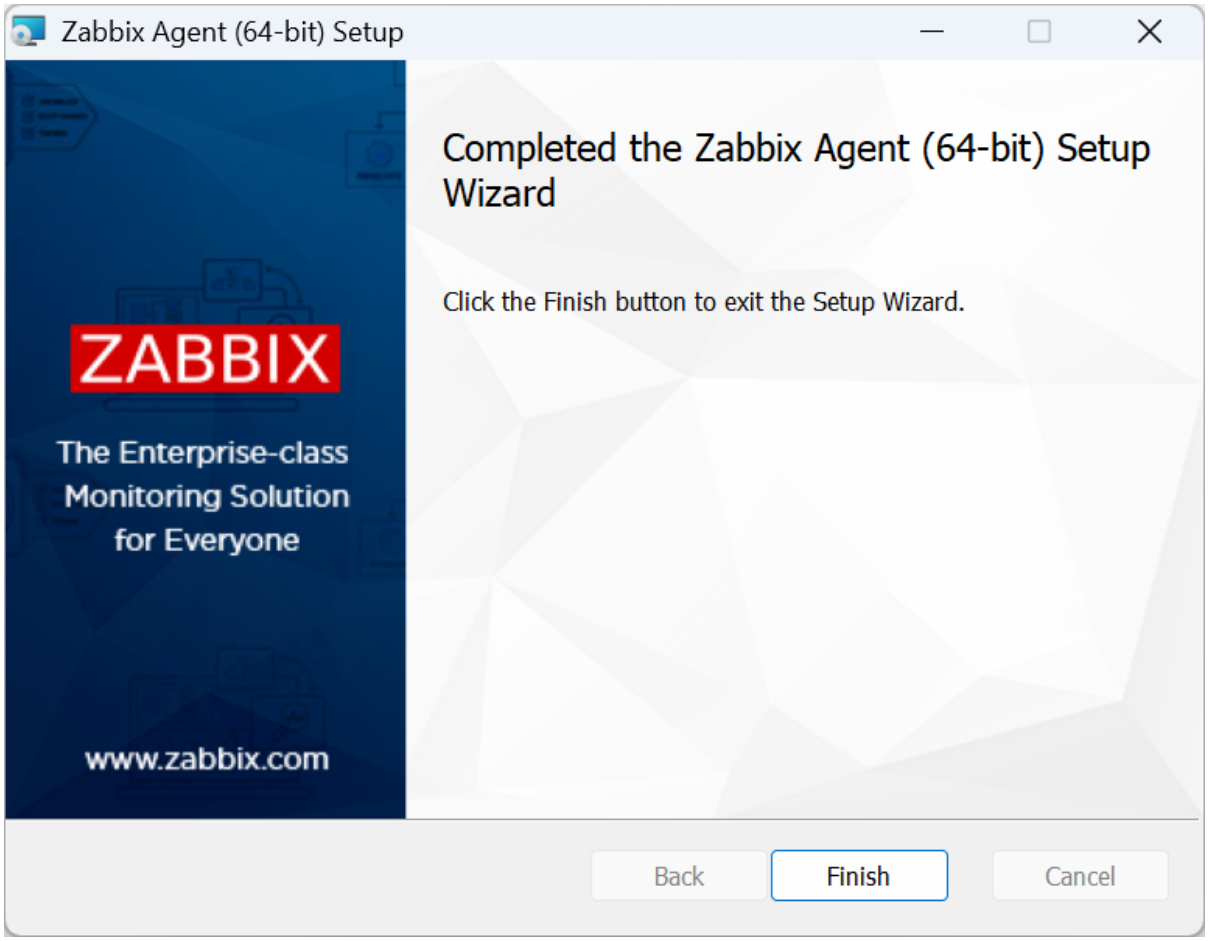
Todos os componentes do Zabbix selecionados e o arquivo de configuração do Zabbix agent serão instalados no local especificado (padrão: `C:\Program Files\Zabbix Agent`). O mesmo se aplica ao Zabbix agent 2, exceto que arquivos de configuração adicionais para seus plugins integrados serão instalados na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

Além disso, `zabbix_agentd.exe` (ou `zabbix_agent2.exe`) será configurado como um serviço do Windows com inicialização automática atrasada (ou inicialização automática nas versões do Windows anteriores ao Windows Vista/Server 2008).

Se uma versão diferente do Zabbix agent estiver em execução durante a instalação, você será solicitado a escolher entre fechar o aplicativo e tentar reiniciá-lo ou deixá-lo aberto; nesse caso, será necessário reiniciar o sistema.



7. Clique no botão *Concluir* para sair do assistente de configuração.



Instalação pela linha de comando

O agent do Zabbix pode ser instalado pela linha de comando executando o instalador MSI com `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
```

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

Parâmetros suportados

Os pacotes de instalação MSI do Zabbix agent suportam os seguintes parâmetros tanto para o Zabbix agent quanto para o Zabbix agent 2.

Note:
Os parâmetros do Zabbix agent/agent2 são definidos no arquivo de configuração durante a instalação. Clique no nome de um 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).

Parameter	Description
ADDDEFAULT	Uma lista de componentes separados por vírgula para instalar com sua configuração padrão. Para mais informações, consulte propriedade ADDDEFAULT. Valores possíveis: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemplo: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Uma lista de componentes separados por vírgula para instalar localmente. Para mais informações, consulte propriedade ADDLOCAL. Valores possíveis: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemplo: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Uma lista de parâmetros AllowKey ou DenyKey separados por ponto e vírgula para restringir verificações do Zabbix agent. Se necessário, use uma barra invertida para escapar do delimitador (`\;`). Define os parâmetros AllowKey e DenyKey no arquivo de configuração do agent. Exemplo: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	O caminho completo para um arquivo de configuração de template do Zabbix agent. Durante a instalação, esse arquivo se tornará o arquivo de configuração do agent. O arquivo deve conter, no mínimo, os parâmetros Server e LogFile. Exemplo: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Use `DONOTSTART=1` para impedir que o instalador MSI inicie o serviço do Zabbix agent.
ENABLEPATH	Use `ENABLEPATH=1` para adicionar o local do Zabbix agent à variável PATH do sistema.
ENABLEPERSISTENTBUFFER	Somente Zabbix agent 2. Habilita o uso de armazenamento persistente local para items ativos.
HOSTINTERFACE	Um parâmetro opcional que define a interface do host.
HOSTMETADATA	Um parâmetro opcional que define os metadados do host.
HOSTMETADATAITEM	Um parâmetro opcional que define um item usado para obter os metadados do host.
HOSTNAME	Um parâmetro opcional que define o nome do host.
HOSTNAMEITEM	Um parâmetro opcional que define um item usado para obter o nome do host.
INCLUDE	Uma lista de arquivos individuais ou de todos os arquivos em um diretório, separados por ponto e vírgula, para incluir no arquivo de configuração do Zabbix agent.
IGNOREALREADYEXISTSFOLDER	Para novas instalações, use `IGNOREALREADYEXISTSFOLDER=1` para permitir a instalação em uma pasta de destino existente fora do diretório `Program Files`. Se esse parâmetro não for especificado, a instalação falhará quando essa pasta de destino já existir.
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 integrados 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 escutará 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 do Zabbix agent. Durante a instalação, quaisquer parâmetros válidos de configuração do agent presentes nesse arquivo (limitados aos listados nesta tabela) serão gravados no novo arquivo de configuração do agent. O arquivo deve conter, no mínimo, o parâmetro Server. Exemplo: `NONMSICONFNAME="C:\full\path\to\example.conf"`

Parameter	Description
PERSISTENTBUFFERFILE	Somente Zabbix agent 2. O arquivo no qual o Zabbix agent 2 deve manter o banco de dados SQLite.
PERSISTENTBUFFERPERIOD	Somente Zabbix agent 2. O período de tempo durante o qual os dados devem ser armazenados quando não houver conexão com o server ou proxy.
SERVER	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS de Zabbix servers ou Zabbix proxies. Este parâmetro é **obrigatório**, exceto quando STARTAGENTS estiver definido como 0.
SERVERACTIVE	O endereço do Zabbix server/proxy ou a configuração de cluster de onde obter verificações ativas.
SKIP	Use `SKIP=fw` para impedir que o instalador MSI adicione uma regra de exceção ao Windows Firewall para o Zabbix agent.
SOURCEIP	O endereço IP de origem para conexões de saída para o Zabbix server ou Zabbix proxy, ou para realizar conexões durante a execução de alguns items (web.page.get, net.tcp.port etc.).
STARTAGENTS	O número de instâncias pré-forked de zabbix_agentd que processam verificações passivas. Se definido como 0, as verificações passivas são desabilitadas e o agent não escutará em nenhuma porta TCP.
STARTUPTYPE	Tipo de inicialização do serviço do Zabbix agent. Valores possíveis: **automatic** - inicia o serviço automaticamente na inicialização do Windows; **delayed** - *(padrão)* adia o início do serviço após a conclusão da inicialização dos serviços iniciados automaticamente (disponível no Windows Vista/Server 2008 e versões posteriores); **manual** - inicia o serviço manualmente (por um usuário ou aplicativo); **disabled** - desabilita o serviço para que ele não possa ser iniciado por um usuário ou aplicativo. Exemplo: `STARTUPTYPE=disabled`
STATUSPORT	Somente Zabbix agent 2. Se definido, o agent escutará nesta porta por solicitações HTTP de status (http://localhost:<port>/status).
TIMEOUT	Especifica quanto tempo aguardar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSACCEPT	As conexões de entrada a serem aceitas (usado para verificações passivas). Se definido como `psk`, então TLSCONNECT também será definido como `psk` (a menos que especificado de outra forma).
TLSCAFILE	O caminho completo de um arquivo contendo os certificados da CA de nível superior para verificação do certificado do par.
TLSCERTFILE	O caminho completo de um arquivo contendo o certificado do agent ou a 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 TLSPSKFILE e TLSPSKVALUE estiverem definidos, o valor de TLSPSKVALUE será gravado no arquivo especificado em TLSPSKFILE. É **recomendado** restringir o acesso ao arquivo da chave pré-compartilhada ajustando as configurações de segurança do arquivo para que somente 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 TLSPSKFILE e TLSPSKVALUE estiverem definidos, o valor de TLSPSKVALUE será gravado no arquivo especificado em TLSPSKFILE. Exemplo: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	O emissor do certificado do server (proxy) permitido.
TLSSERVERCERTSUBJECT	O subject do certificado do server (proxy) permitido.
UNSAFEUSERPARAMETERS	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário.

Exemplos

O exemplo a seguir instala o Zabbix agent com configuração personalizada. Ele também habilita 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=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
```

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

O próximo exemplo instala uma versão mais recente do Zabbix agent e usa um **arquivo de configuração de template** (`CONF="C:\agent-template.conf"`). Durante a instalação, esse arquivo se tornará o arquivo de configuração do agent. Para herdar parâmetros do arquivo de configuração antigo, use marcadores de posição de parâmetros (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]
```

Como alternativa, você pode usar um **arquivo de configuração personalizado** (`NONMSICONFNAME="C:\agent-custom.conf"`). Durante a instalação, quaisquer parâmetros válidos de configuração do agent presentes nesse 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 que devem ser 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 instalação MSI de 64 bits, disponíveis para download.

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

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

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

Instalação pelo Assistente de Configuração

1. Clique duas vezes no arquivo MSI baixado para iniciar a instalação.
2. Aceite o Contrato de Licença do Usuário Final.
3. Selecione os plugins carregáveis do Zabbix agent 2 a serem instalados.

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

4. Clique em *Instalar* para iniciar a instalação.

Todos os plugins carregáveis do Zabbix agent 2 selecionados serão instalados no local especificado (padrão: `C:\Program Files\Zabbix Agent 2`), com seus arquivos de configuração instalados na subpasta `zabbix_agent2.d`.

5. Clique no botão *Concluir* para sair do Assistente de Configuração.

Instalação pela linha de comando

Os plugins carregáveis do Zabbix agent 2 podem ser instalados pela linha de comando executando o instalador MSI com `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 oferecem suporte aos seguintes parâmetros.

Parameter	Description
ADDDEFAULT	Uma lista de componentes separados por vírgulas para instalar em sua configuração padrão. Para mais informações, consulte ADDDEFAULT property. Valores possíveis: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Exemplo: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Uma lista de componentes separados por vírgulas para instalar localmente. Para mais informações, consulte ADDLOCAL property. Valores possíveis: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Exemplo: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	O caminho completo para uma pasta onde os componentes do Zabbix serão instalados, com seus arquivos de configuração instalados na subpasta `zabbix_agent2.d`. Exemplo: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Compilando o Zabbix agent 2 no Windows

Visão geral

Esta página demonstra como compilar o Zabbix agent 2 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 somente a versão de 32 bits pode ser compilada em uma plataforma de 32 bits.

A compilação do Zabbix agent 2 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 Zabbix agent 2 usando um dos seguintes métodos:

- Usando o 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, 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 do MSYS2.
- É recomendável criar um diretório de trabalho em `C:\Zabbix` para todos os arquivos de código-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, selecione 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, especifique `C:\Zabbix\Go` como pasta de instalação.

3. Baixe a distribuição MinGW que usa a biblioteca de runtime Microsoft Visual C; por exemplo:

- Para compilações de 64 bits: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Para compilações de 32 bits: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
A versão do MinGW deve atender aos requisitos mínimos.

Em seguida, extraia-a para `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` para compilações de 32 bits).

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 de 64 bits:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### Para compilações de 32 bits:
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-o para `C:\Zabbix\zabbix-8.0.0`.

6. Navegue até o diretório de compilação do Zabbix (`C:\Zabbix\zabbix-8.0.0\build\mingw`) e crie o seguinte script `build.bat`:

- Para compilações de 64 bits:

```
:: Adicione o MinGW e o Go à variável de sistema `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 os sinalizadores 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 de 32 bits:

```
:: Adicione o MinGW e o Go à variável de sistema `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 os sinalizadores 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 de 64 bits) ou `C:\Zabbix\zabbix-8.0.0\bin\win32` (para compilações de 32 bits). 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 `zabbix_agent2.exe` e seus arquivos de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent2`) e execute o agent:

```
mkdir C:\Zabbix\agent2

#### Para compilações de 64 bits:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

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

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

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

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

Compilando plugins carregáveis do Zabbix agent 2

1. Baixe o código-fonte do plugin Zabbix que corresponda à sua versão do Zabbix agent 2 (por exemplo, `zabbix-agent2-plugin-postgres`) e extraia-o para `C:\Zabbix`.

> **Attention:**
> Antes de instalar um plugin, verifique o arquivo README dele. Ele pode conter requisitos específicos e instruções de instalação.

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

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

#### Para builds de 64 bits:
mingw32-make ARCH=AMD64

#### Para builds de 32 bits:
mingw32-make ARCH=x86
```

Após a compilação, o binário do plugin do Zabbix agent 2 e seu arquivo de configuração estarão localizados no mesmo diretório do plugin.

O executável do plugin pode ser colocado em qualquer local, desde que possa ser carregado pelo Zabbix agent 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=/path/to/executable/zabbix-agent2-plugin-postgresql
```

O caminho para o arquivo de configuração do plugin deve ser especificado no parâmetro Include do arquivo de configuração do Zabbix agent 2:

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

Acesse configuração para mais detalhes sobre a configuração de plugins.

Compilando o Zabbix agent 2 manualmente

> **Attention:**
> Este método de compilação do Zabbix agent 2 é adequado para usuários que precisam de controle total sobre o ambiente de compilação ou estão em um ambiente restrito onde usar o vcpkg não é possível.

Esta seção contém instruções para compilar o Zabbix agent 2 manualmente, o que inclui instalar as ferramentas de compilação e dependências necessárias e, em seguida, compilar o agent.

Configurando as ferramentas de build

1. Baixe e instale MSYS2 (disponível como instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\msys64` como a pasta de instalação.

2. Baixe e instale Go (disponível como instalador MSI; veja as versões do Go atualmente suportadas). Durante a instalação, certifique-se de especificar `C:\Zabbix\Go` como a pasta de instalação.

3. Baixe a distribuição MinGW que usa a biblioteca de runtime Microsoft Visual C; por exemplo:

- Para builds de 64 bits: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Para builds de 32 bits: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
A versão do MinGW deve atender aos requisitos mínimos.

Em seguida, extraia-o para `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` para builds de 32 bits).

Instalando o OpenSSL

Note:
Para compilar o Zabbix agent 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 código-fonte do OpenSSL e extraia-o em `C:\Zabbix\openssl-3.5.0`.

3. Navegue até o diretório do OpenSSL extraído e crie o seguinte script `build.sh`:

- Para compilações de 64 bits:

```
####!/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 compilações de 32 bits:

```
####!/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 que não sejam administradores ao diretório `C:\Zabbix\x64\OpenSSL-Win64-350-static`. Caso contrário, o agent carregará as configurações de SSL de um caminho que pode ser modificado por usuários sem privilégios, resultando em uma possível vulnerabilidade de segurança.

- A opção `no-shared` torna as bibliotecas estáticas do OpenSSL libcrypto.lib e libssl.lib autocontidas, 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 Zabbix agent 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 Zabbix agent; no entanto, ao copiá-lo para outras máquinas, as DLLs do OpenSSL também deverão ser copiadas.

Para obter mais informações sobre outras opções de configuração do OpenSSL, consulte a documentação do OpenSSL.

4. Configure e instale o OpenSSL executando o script (observe que isso pode levar algum tempo):

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

Instalando o PCRE2

1. Baixe o arquivo fonte do PCRE2 e extraia para `C:\Zabbix\pcre2-10.45`.

2. Abra o terminal MSYS2 MSYS com privilégios de administrador. Em seguida, crie um diretório `build` no diretório extraído do PCRE2 e navegue até ele:

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

3. Configure o PCRE2:

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

#### Para builds 32-bit:
export PATH="/c/Zabbix/mingw32/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE
```

> **Note:**
> Se ocorrerem erros, recomenda-se excluir o cache do CMake antes de tentar repetir o processo de build do CMake. O cache (`CMakeCachecache.txt`) pode ser localizado no diretório build do diretório extraído do PCRE2.

4. Instale o PCRE2:

```
make install
```

Compilando o Zabbix agent 2

1. Baixe o arquivo-fonte do Zabbix e extraia-o em `C:\Zabbix\zabbix-8.0.0`.

Se 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 comandos a seguir em uma máquina **Linux** com o Go instalado (necessário para configurar o Zabbix agent 2):

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

Isso criará um arquivo-fonte, que poderá ser copiado para uma máquina Windows.

2. Abra o Prompt de Comando com privilégios de administrador. Em seguida, navegue até o diretório de compilação do Zabbix e compile o Zabbix agent; certifique-se de especificar corretamente os diretórios nos quais o OpenSSL e o PCRE2 estão instalados:

- Para compilações de 64 bits:

```
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 compilações de 32 bits:

```
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-Win32-350-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 em `C:\Zabbix\zabbix-8.0.0\bin\win32` para compilações de 32 bits). 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 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.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

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

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).

Estas instruções se aplicam às versões do Windows compatíveis com o Visual Studio 2022.

A compilação do agent do Zabbix requer:

- Compilador C (incluído no Build Tools for 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 o 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 da compilação do agent.

Dependendo das suas necessidades de monitoramento, bibliotecas adicionais podem ser necessárias. Para obter mais informações, consulte Requisitos.

Attention:
Antes de iniciar o processo de compilação, tenha em mente:

- Para executar comandos, use o "x64 Native Tools Command Prompt" (incluído no Build Tools for Visual Studio 2022), iniciado por um usuário com permissões suficientes para gravar em pastas protegidas.
- É recomendável criar um diretório de trabalho em `C:\Zabbix` para todos os arquivos de código-fonte e pastas de compilação. No entanto, os componentes compilados devem ser instalados em `C:\Program Files\Zabbix\x64`.

Compilando o Zabbix agent 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, selecione a carga de trabalho *Desenvolvimento para desktop com C++*, que inclui as ferramentas necessárias para compilar o agent com o vcpkg:

- Compilador C (Microsoft Visual C++)
- Ferramenta de linha de comando NMake

- Gerenciador de pacotes vcpkg
- "x64 Native Tools Command Prompt"

2. Abra o `cmd.exe` com privilégios de administrador, inicialize o vcpkg e instale as dependências necessárias para compilar o Zabbix agent (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 compactado do código-fonte do Zabbix e extraia-o em `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 nos quais 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. Abra o "x64 Native Tools Command Prompt" e compile o Zabbix agent executando o script:

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

Se você abriu um prompt de comando diferente, inicialize primeiro o ambiente de compilação do Visual Studio:

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

Coloque entre aspas os caminhos que contêm espaços.

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 Zabbix agent está localizado em `C:\Zabbix\zabbix-8.0.0\conf`.

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

C:\Zabbix\agent\zabbix_agentd.exe -c C:\Zabbix\agent\zabbix_agentd.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 para desktop com C++*, que inclui as ferramentas necessárias para compilar o agent manualmente:

- Compilador C (Microsoft Visual C++)

- Ferramenta de linha de comando NMake
- "x64 Native Tools Command Prompt"

Instalando o OpenSSL

> **Note:**
> Para compilar o Zabbix agent sem suporte a TLS, prossiga para a seção Instalando o PCRE2.

1. Baixe e instale o Strawberry Perl (disponível como instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\Strawberry` como a pasta de instalação.

2. Instale o módulo Perl `Text::Template`:

```
cpanm Text::Template
```

3. Verifique se o Netwide Assembler (NASM; necessário para compilar o OpenSSL) foi compilado durante a instalação do Strawberry Perl:

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

Se o NASM não estiver compilado, instale-o manualmente. Para obter mais informações, consulte a documentação do NASM.

4. Baixe o arquivo-fonte do OpenSSL e extraia-o em `C:\Zabbix\openssl-3.5.0`.

5. Acesse 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 Zabbix agent no Windows (por exemplo, `C:\zabbix` ou `C:\openssl-64bit`), certifique-se de revogar o acesso de gravação dos usuários não administradores a esse diretório. Caso contrário, o agent carregará as configurações de 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 libcrypto.lib e libssl.lib sejam bibliotecas estáticas autocontidas do OpenSSL, 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, será necessário recompilar o Zabbix agent.
- 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 Zabbix agent; no entanto, ao copiá-lo para outras máquinas, as DLLs do OpenSSL também deverão ser copiadas.

Para obter 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 poderá causar resultados inesperados ou riscos de segurança. Se algum teste falhar, consulte a documentação do OpenSSL para obter informações sobre 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 somente os componentes de software (bibliotecas e arquivos de cabeçalho, mas não a 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 compactado do código-fonte do Zabbix e extraia-o em `C:\Zabbix\zabbix-8.0.0`.

Se precisar gerar um arquivo compactado do código-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 comandos a seguir em uma máquina **Linux**:

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

Isso criará um arquivo compactado do código-fonte, que poderá ser copiado para uma máquina Windows.

2. Navegue até o diretório de compilação do Zabbix, abra o "x64 Native Tools Command Prompt" e compile o agent do Zabbix (ou outros componentes); certifique-se de especificar corretamente os diretórios nos quais o OpenSSL e o PCRE2 estão instalados:

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

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
```

Coloque entre aspas os caminhos que contêm espaços.

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_agentd.exe` e seu arquivo de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent`) e 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.conf C:\Zabbix\agent\

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

Instalação do agent do macOS a partir do PKG

Visão geral

O Zabbix agent pode ser instalado no macOS usando pacotes de instalação PKG, disponíveis para download.

Os pacotes do Zabbix agent estão disponíveis com ou sem encryption.

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 do Zabbix.

Veja se o agent do Zabbix está em execução:

```
ps aux | grep zabbix_agentd
```

Veja se o agent do Zabbix foi instalado a partir de pacotes:

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

Veja os arquivos que foram instalados a partir do pacote de instalação (observe que o / inicial não é exibido nesta visualização):

```
pkgutil --only-files --files com.zabbix.pkg.ZabbixAgent
Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
usr/local/bin/zabbix_get
usr/local/bin/zabbix_sender
usr/local/etc/zabbix/zabbix_agentd/userparameter_examples.conf.NEW
usr/local/etc/zabbix/zabbix_agentd/userparameter_mysql.conf.NEW
usr/local/etc/zabbix/zabbix_agentd.conf.NEW
usr/local/sbin/zabbix_agentd
```

Pare o agent do Zabbix se ele foi iniciado com `launchctl`:

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

Remova os arquivos (incluindo configuração e logs) que foram instalados com o pacote de instalação:

```
sudo rm -f /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
sudo rm -f /usr/local/sbin/zabbix_agentd
sudo rm -f /usr/local/bin/zabbix_get
sudo rm -f /usr/local/bin/zabbix_sender
sudo rm -rf /usr/local/etc/zabbix
sudo rm -rf /var/log/zabbix
```

Esqueça que o agent do Zabbix foi instalado:

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

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 quiser compilar binários do agent com TLS, também precisará de OpenSSL ou GnuTLS.

Para instalar o Automake e o pkg-config, você precisará do gerenciador de pacotes Homebrew em https://brew.sh/. Para instalá-lo, abra o terminal e execute o seguinte comando:

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

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 tenha essas bibliotecas, pode usar bibliotecas pré-compiladas fornecidas pelo Homebrew. Normalmente, essas são máquinas macOS que usam o Homebrew para compilar binários do Zabbix agent ou para outros fins.

Se os binários do agent forem usados em máquinas macOS que não tenham a versão compartilhada das bibliotecas, você deverá compilar bibliotecas estáticas a partir do código-fonte e vincular o Zabbix agent 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
```

Guia de implantação do Zabbix server na AWS

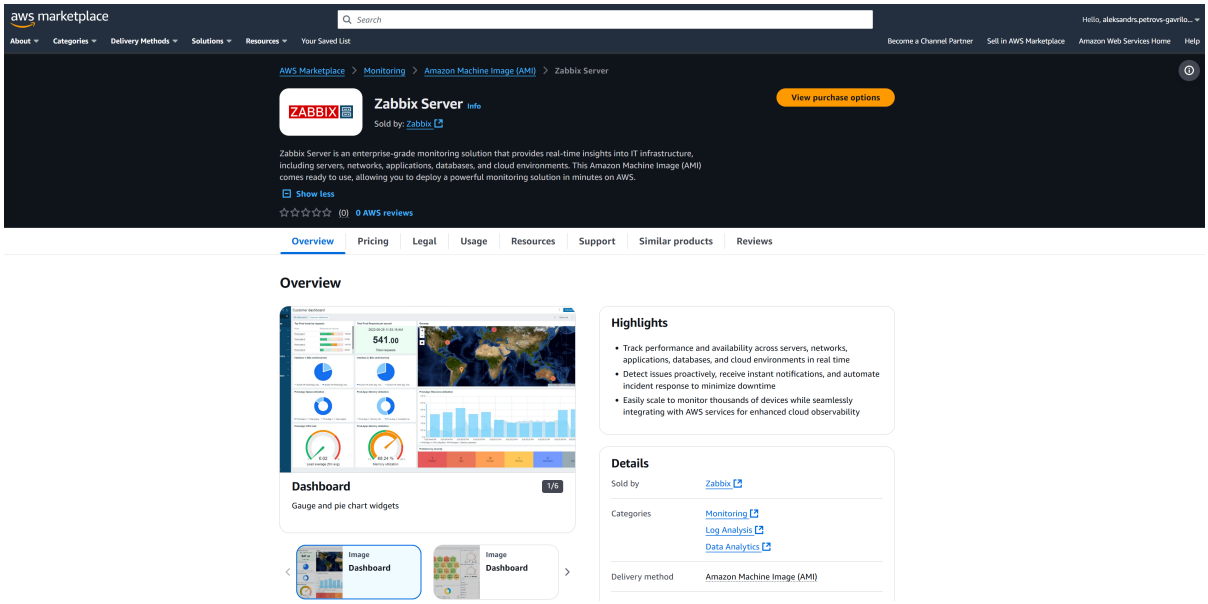
Visão geral

Esta página descreve como implantar o Zabbix server usando a AWS.

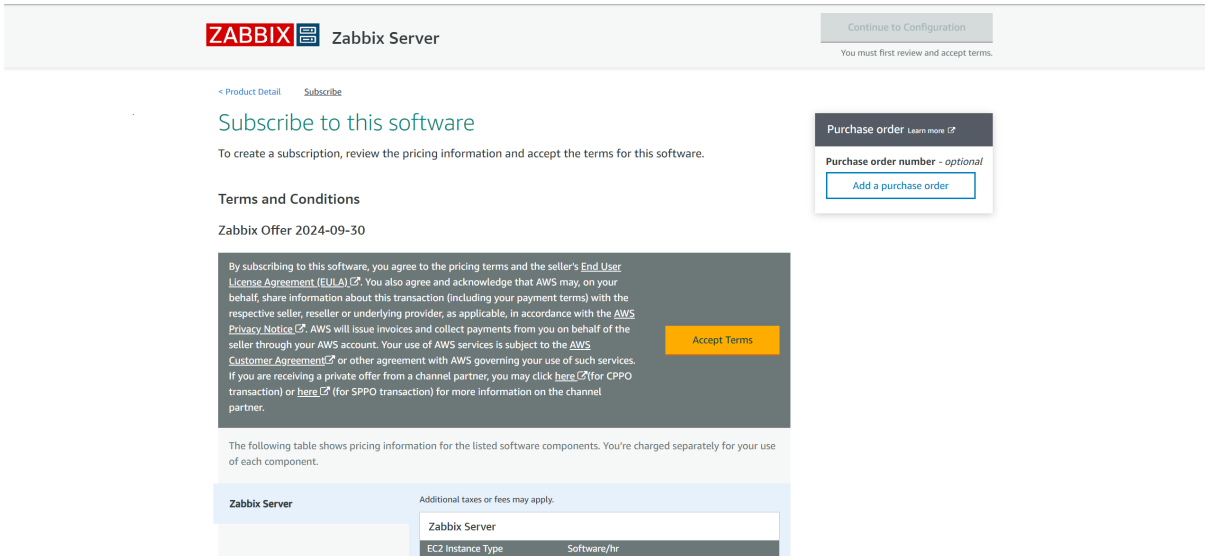
> **Attention:**
> Várias versões do Zabbix estão disponíveis para implantação na plataforma. Leia a documentação da mesma versão do Zabbix que você implantou. A documentação de diferentes versões do Zabbix pode ser selecionada no menu suspenso de versões.

Implantando o appliance

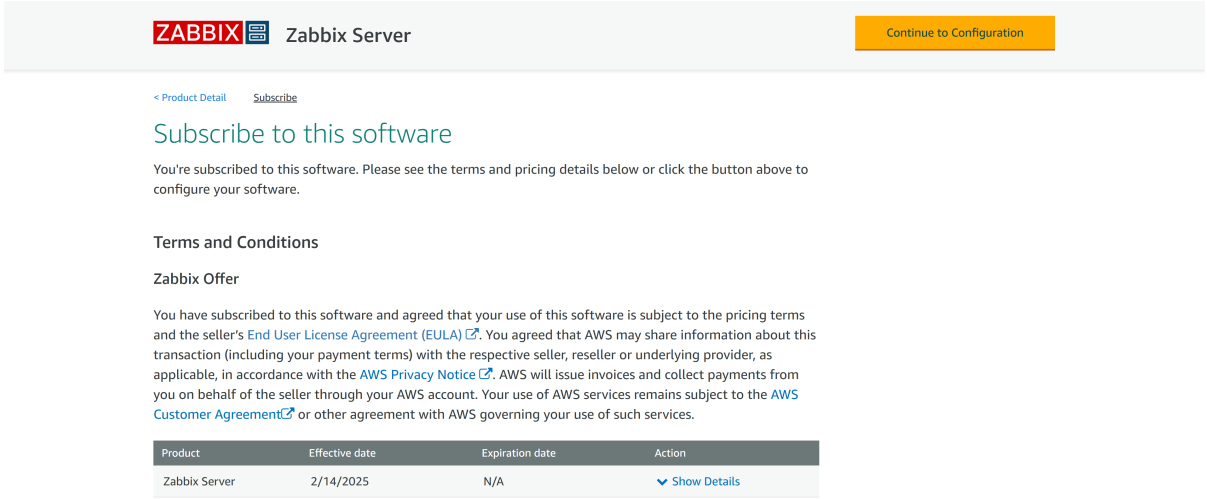
Para implantar o appliance do Zabbix, primeiro, acesse a página do Zabbix no AWS Marketplace. Em seguida, clique em *View purchase options* para implantar a imagem e prosseguir para a página de assinatura.



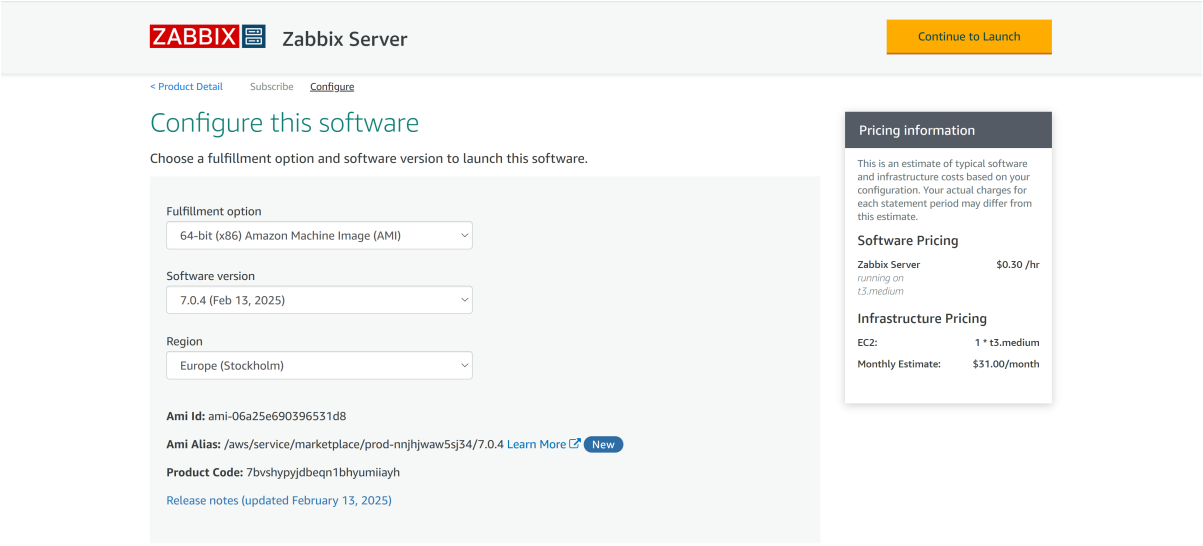
Clique no botão *Accept terms* para aceitar os *Terms and Conditions*.



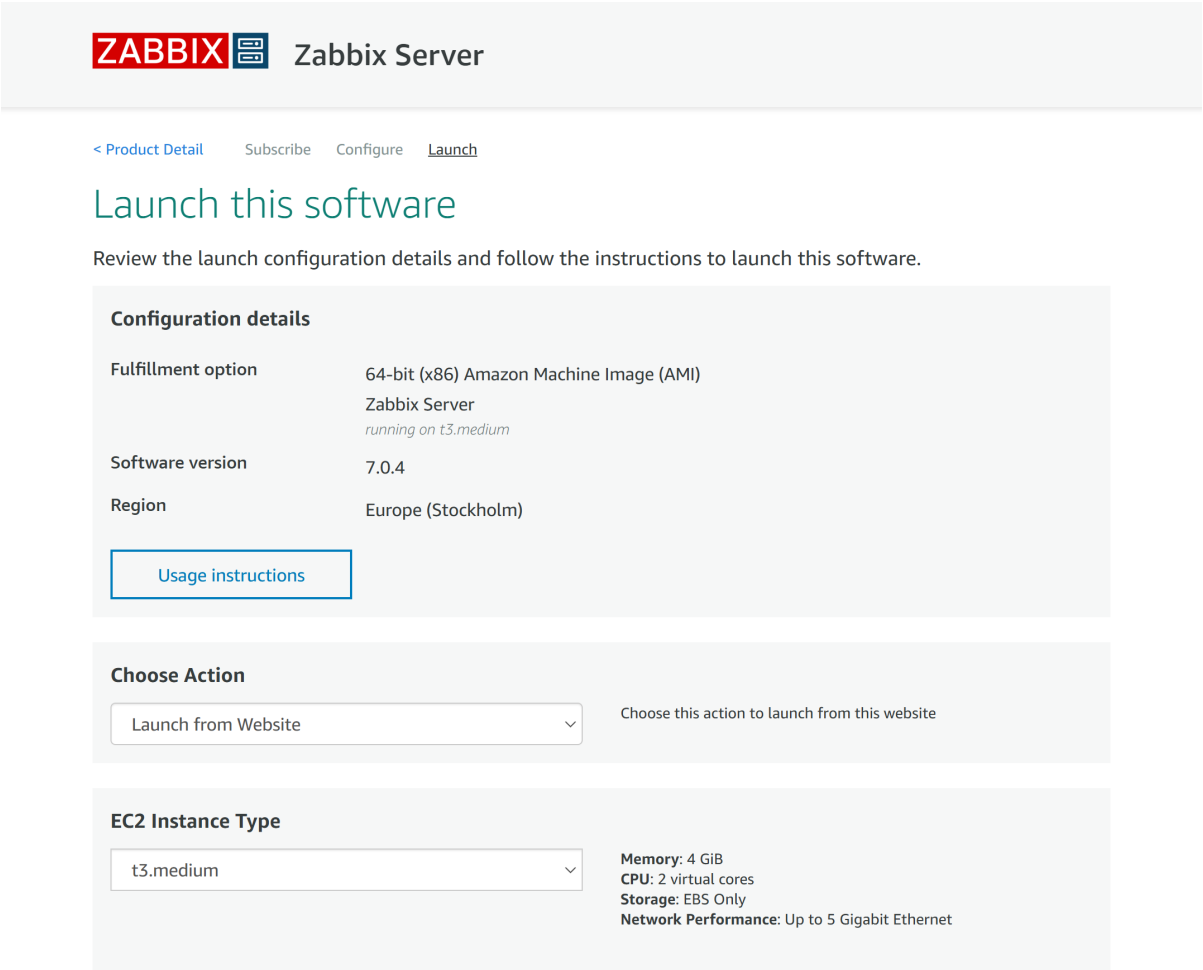
Clique no botão *Continue to Configuration* para acessar a seção de configuração do appliance do Zabbix.



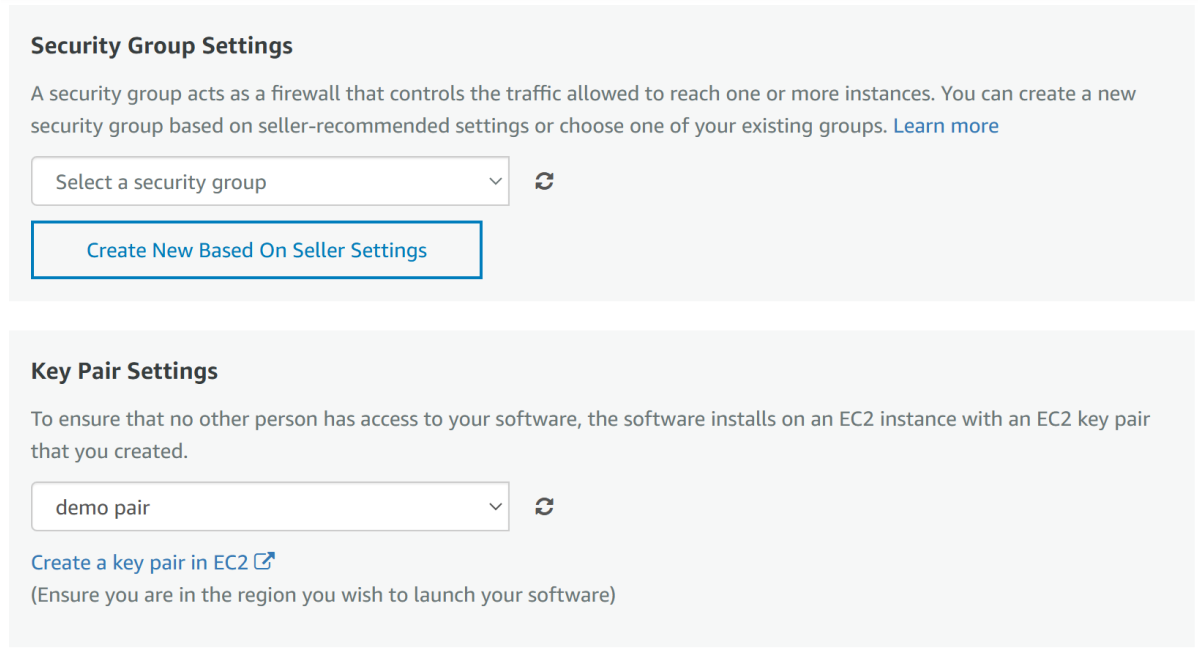
Selecione a opção Fulfilment, a versão do Zabbix e a Região de sua preferência.



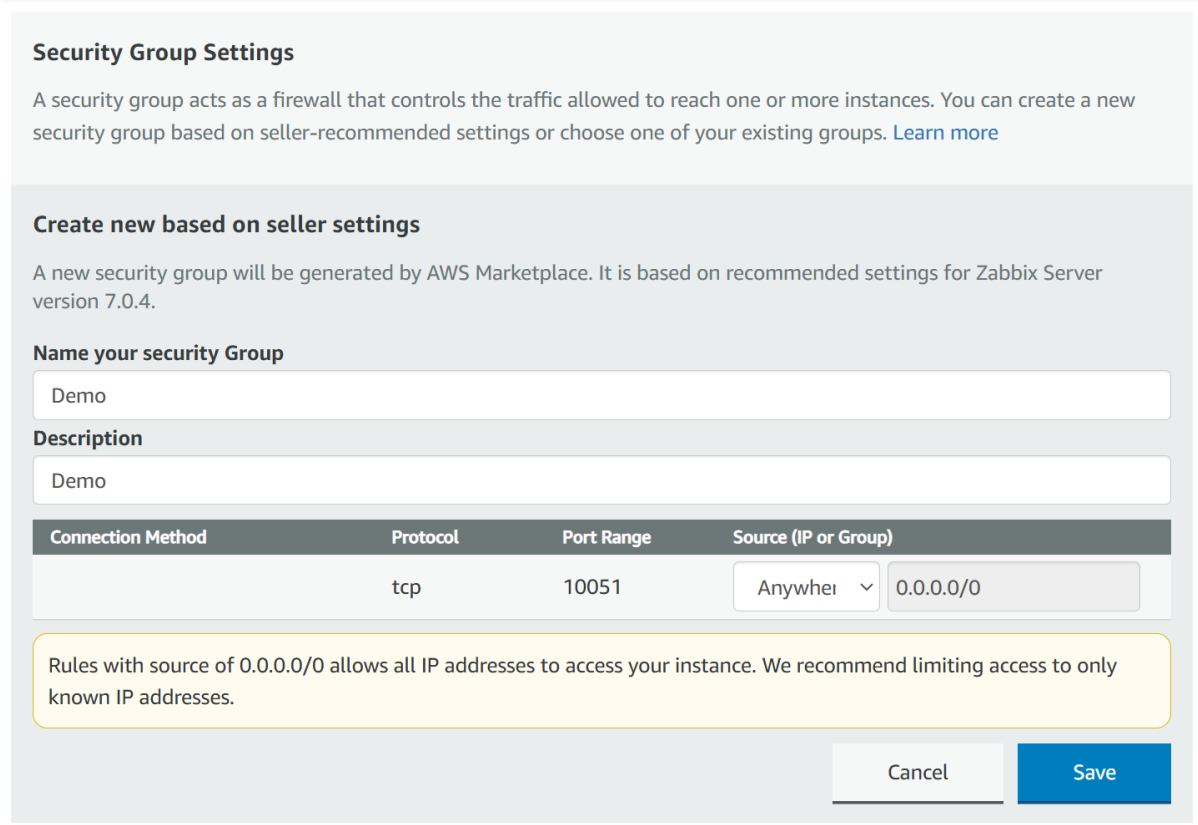
Clique no botão *Continue to Launch*. Na página Launch, selecione a opção Launch, o tipo de instância EC2, a VPC e as configurações de Subnet de sua preferência.



Na parte inferior da página Launch, especifique as configurações do Security Group e selecione ou crie um par de chaves do EC2. Certifique-se de salvar sua chave privada em um local seguro!



Se você selecionar *Create a new Security Group based on seller settings*, a porta TCP 10051 (Zabbix server) será permitida. Observação: se desejar acessar a instância via SSH e HTTP/HTTPS (web frontend), também será necessário permitir conexões SSH e HTTP/HTTPS nas configurações do Security Group.



Parabéns! Seu appliance do Zabbix foi implantado com sucesso.

ZABBIX Zabbix Server

Congratulations! An instance of this software is successfully deployed on EC2!

AMI ID: ami-06a25e690396531d8 (View Launch Configuration Details)
You can view this instance on EC2 Console. You can also view all instances on Your Software. Software and AWS hourly usage fees apply when the instance is running and will appear on your monthly bill.

You can launch this configuration again below or go to the configuration page to start a new one.

Configuration details

Fulfillment option	64-bit (x86) Amazon Machine Image (AMI) Zabbix Server *running on t3.medium*
Software version	7.0.4
Region	Europe (Stockholm)

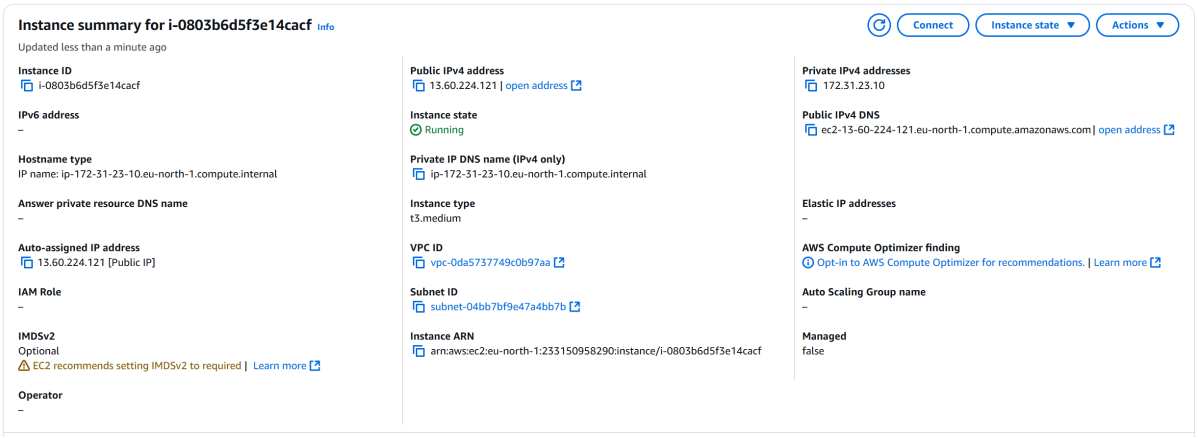
Usage instructions

Choose Action

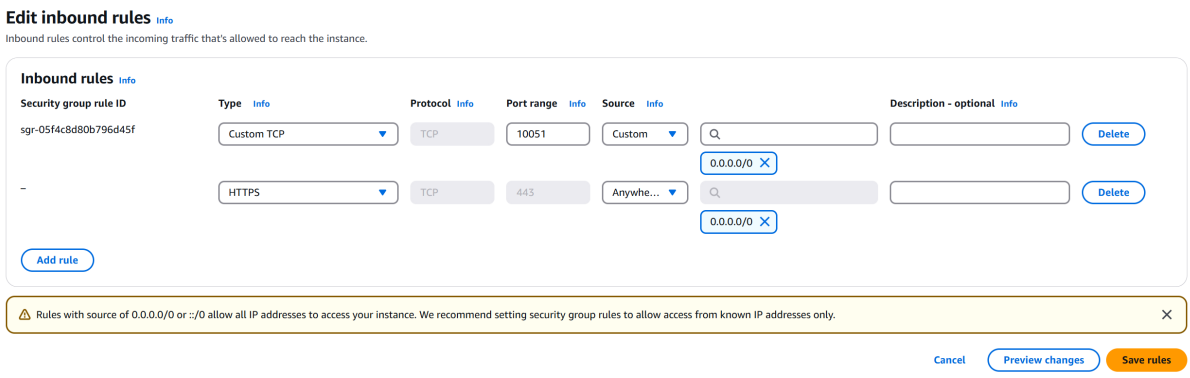
Select a launch action

Acessando o appliance

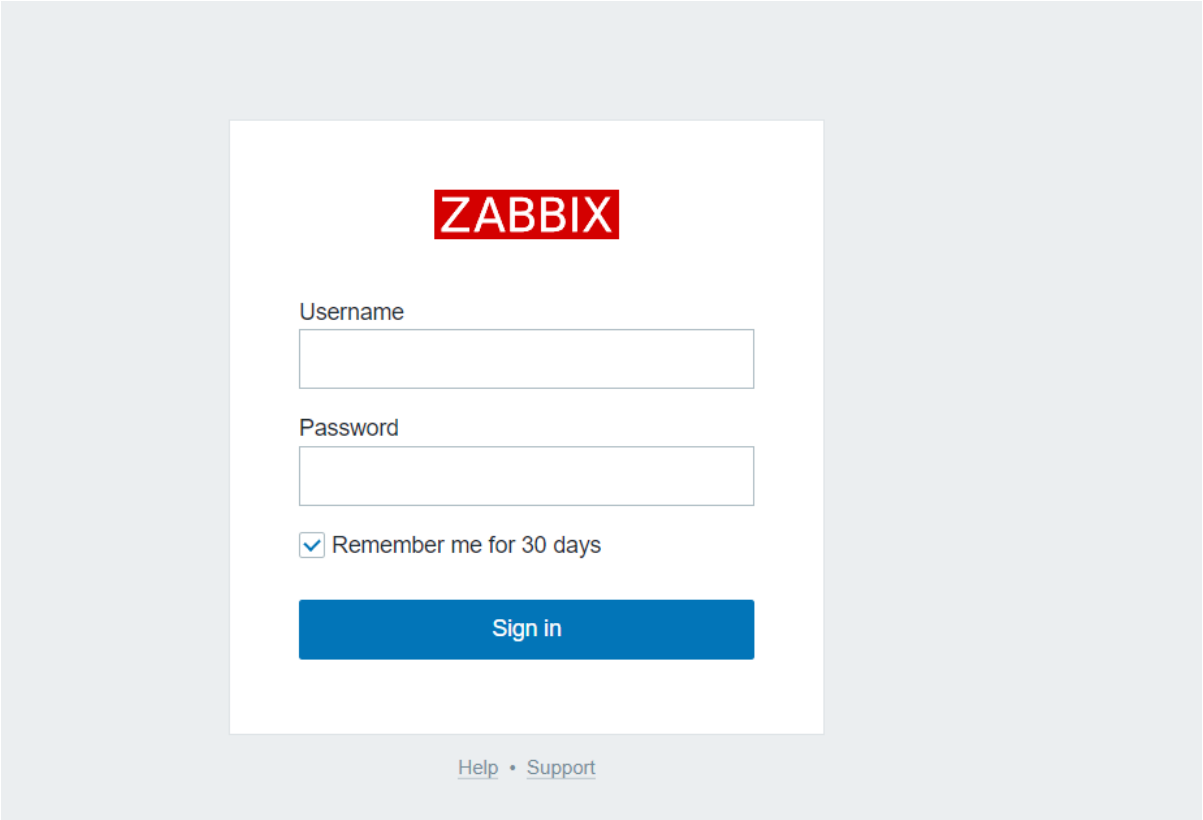
Abra a seção *Instances* e selecione a instância do appliance Zabbix recém-implantada.



Para acessar o frontend do Zabbix usando o endereço IPv4 público ou o DNS IPv4 público da instância, permita o tráfego HTTPS de entrada na porta 443 no Security Group.



Agora podemos acessar o frontend do Zabbix abrindo o endereço IPv4 público ou o DNS IPv4 público do appliance Zabbix.



Para obter a senha, precisamos nos conectar à máquina usando o console serial ou SSH. O nome de usuário SSH padrão é rocky. Não se esqueça de usar o par de chaves selecionado ou gerado anteriormente. Observe que a geração das credenciais pode levar algum tempo. A geração das credenciais pode levar alguns minutos. Aguarde alguns minutos antes de tentar se conectar.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

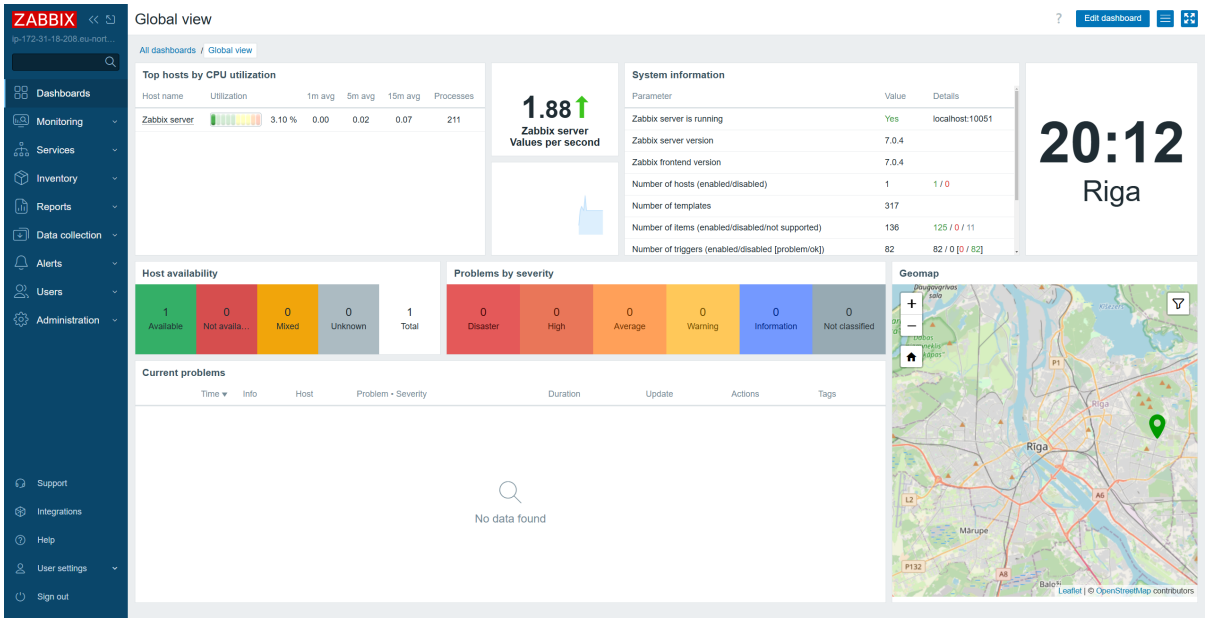
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

Agora você pode usar essas informações para fazer login no frontend do Zabbix.



Acessando o banco de dados

Você pode acessar o banco de dados do Zabbix usando o usuário `root` sem inserir uma senha.

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

As credenciais são armazenadas no arquivo de configuração `/root/.my.cnf`.

O fuso horário padrão do frontend do Zabbix pode ser alterado em *Administração* > *Geral* > *GUI*

Guia de implantação do AWS para o proxy do Zabbix

Visão geral

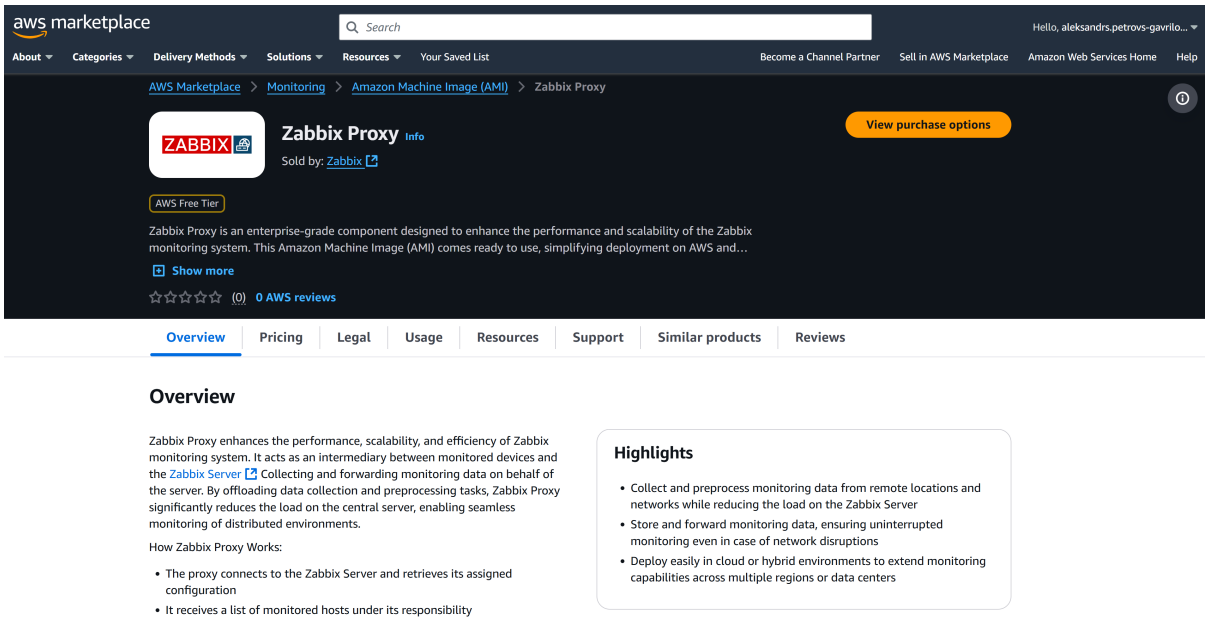
Esta página descreve como implantar o Zabbix proxy usando a AWS.

> **Attention:**
> Várias versões do Zabbix estão disponíveis para implantação na plataforma. Leia a documentação referente à mesma versão do Zabbix que você implantou. A documentação de diferentes versões do Zabbix pode ser selecionada no menu suspenso de versões.

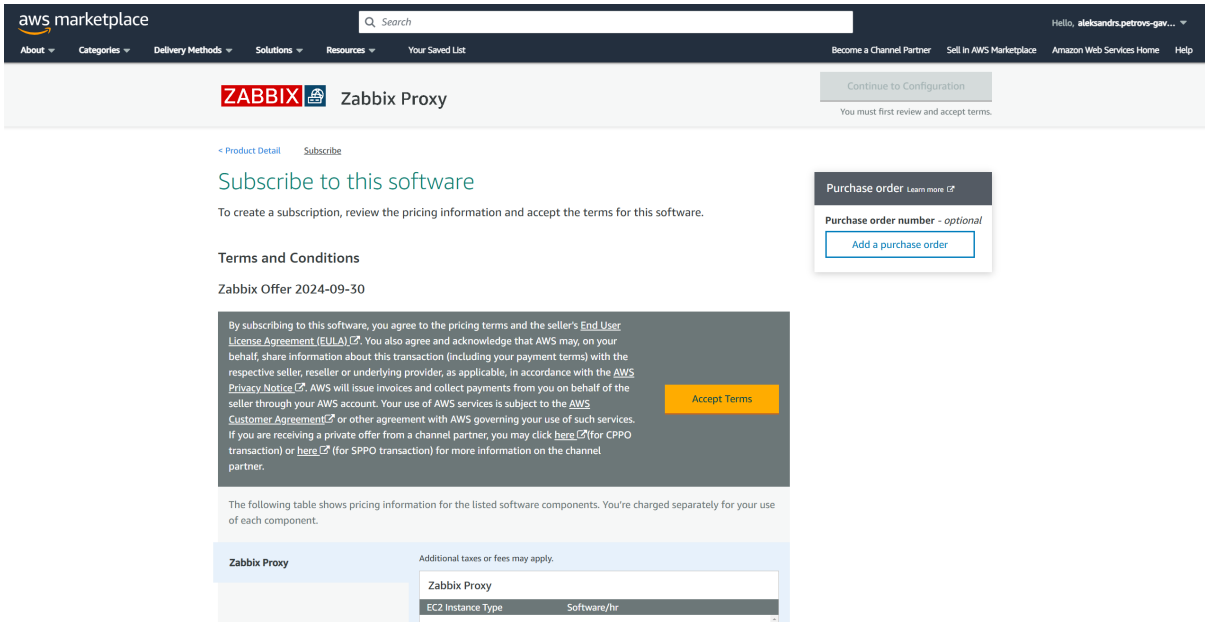
Implantando o Zabbix proxy

> **Attention:**
> É necessária uma instância funcional do Zabbix server para usar o componente Zabbix proxy.

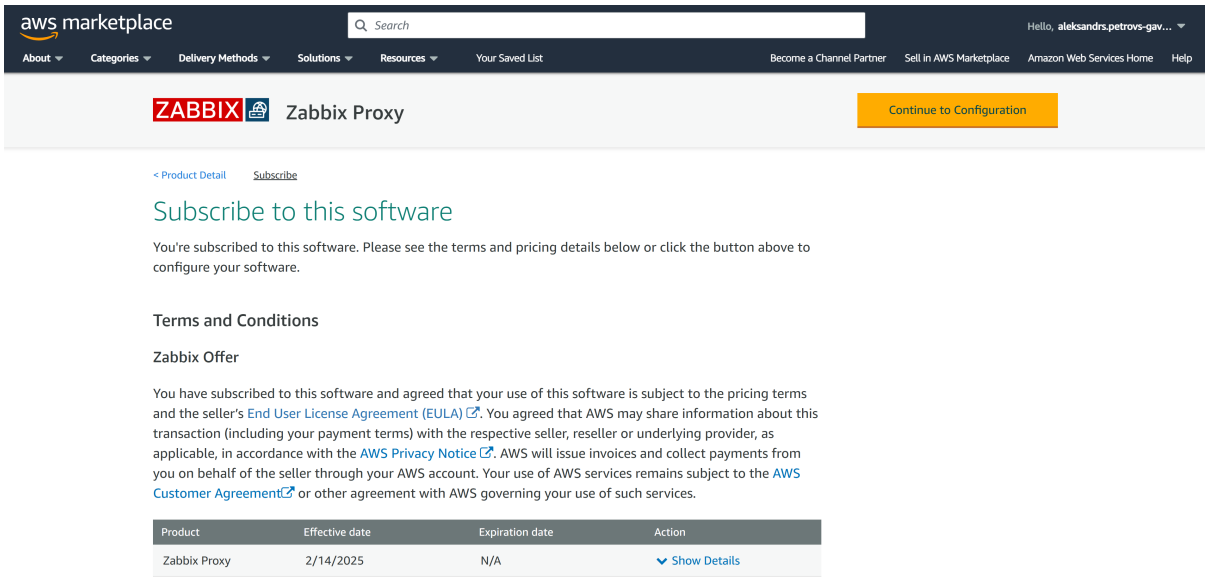
Para implantar o Zabbix proxy, primeiro acesse a página do Zabbix no AWS Marketplace. Em seguida, clique em *View purchase options* para implantar a imagem e prosseguir para a página de assinatura.



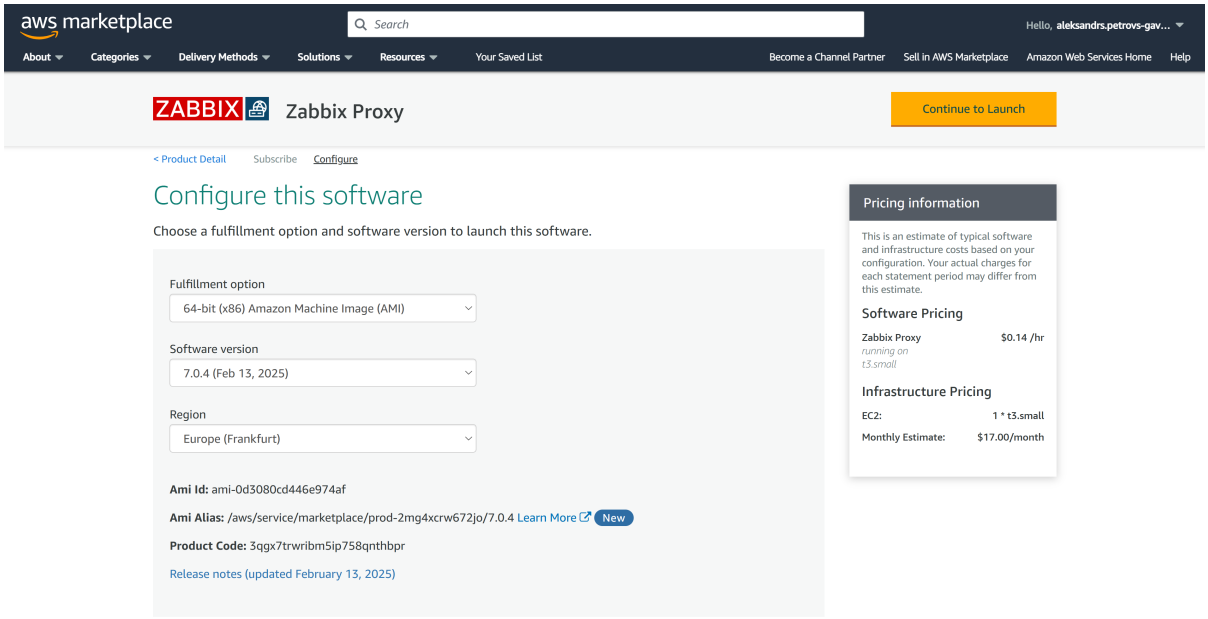
Clique no botão *Accept terms* para aceitar os *Terms and Conditions*.



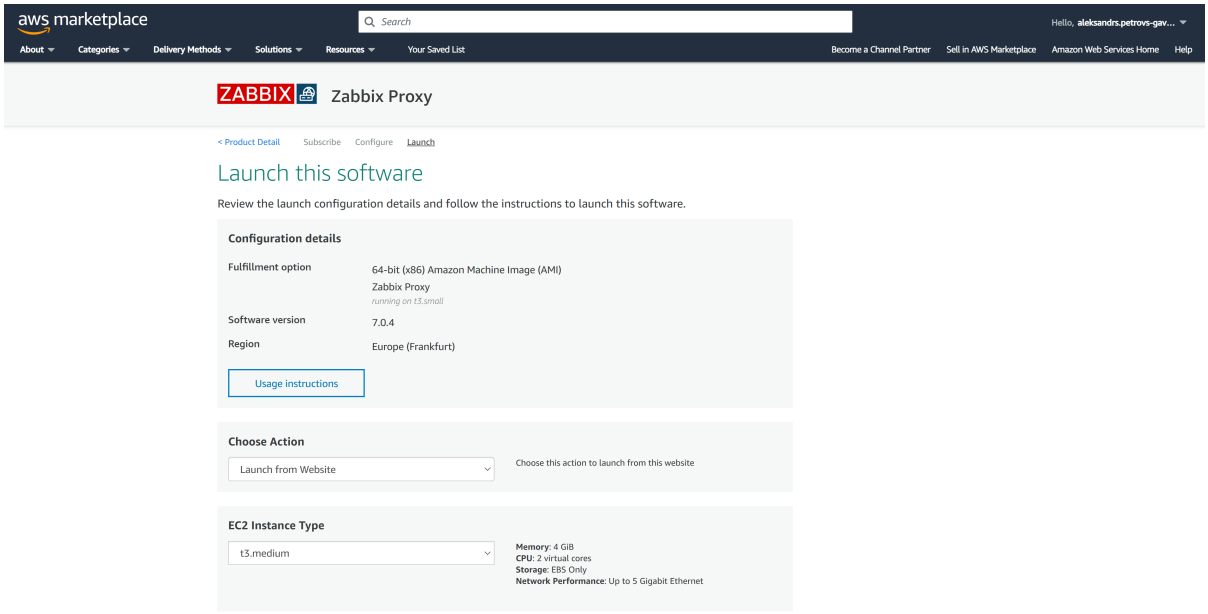
Clique no botão *Continue to Configuration* para acessar a seção Configuration do Zabbix proxy.



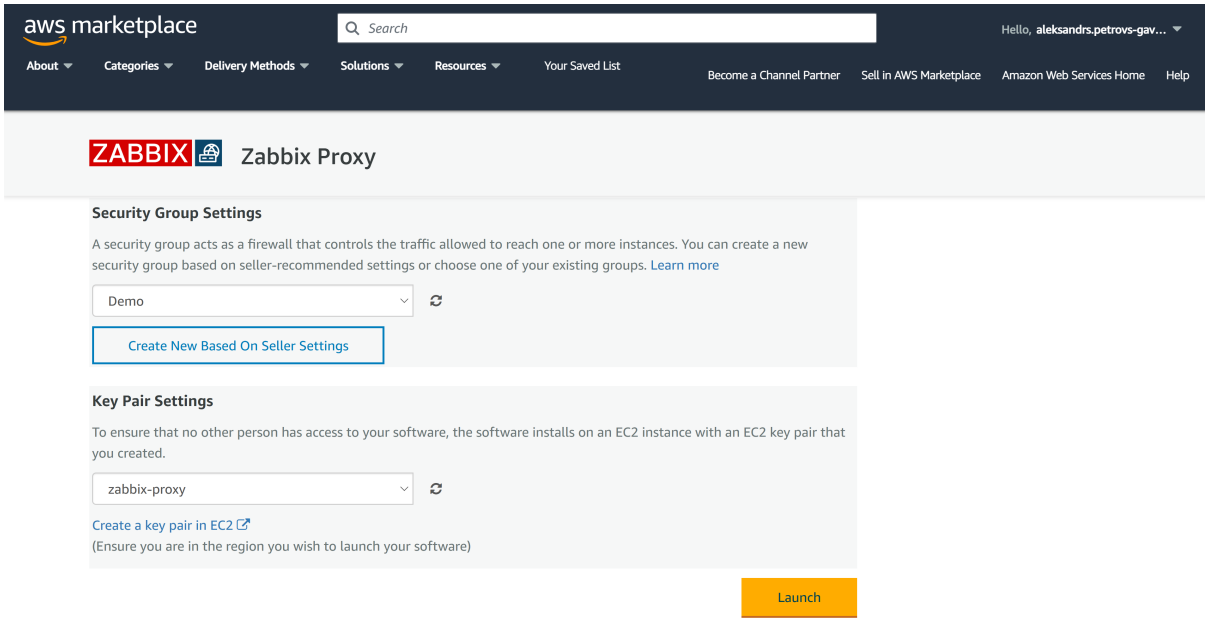
Selecione a opção Fulfilment, a versão do Zabbix e a Region de sua preferência.



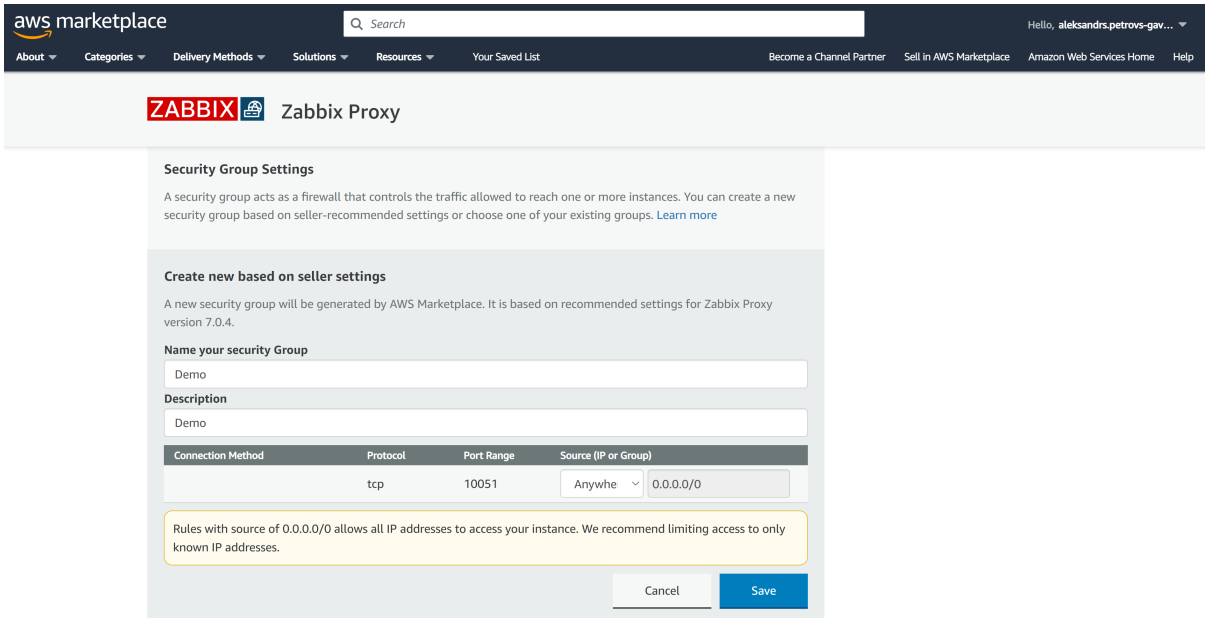
Clique no botão *Continue to Launch*. Na página Launch, selecione a opção Launch, o EC2 Instance Type, a VPC e as configurações de Subnet de sua preferência.



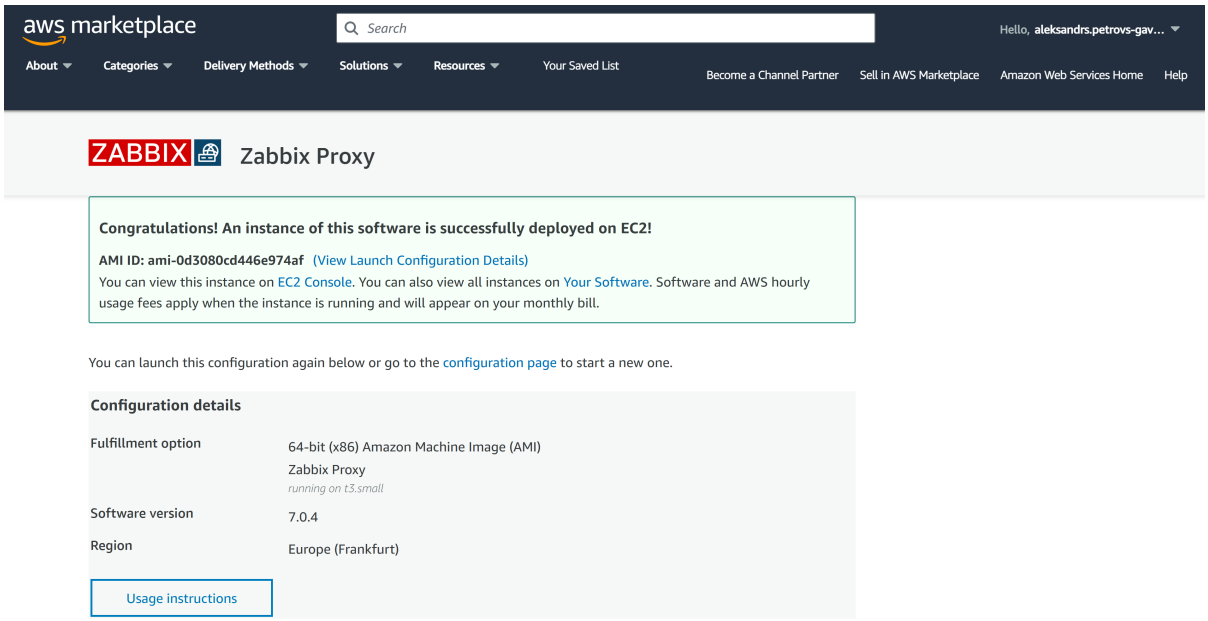
Na parte inferior da página Launch, especifique as configurações do Security Group e selecione ou crie um par de chaves do EC2. Certifique-se de salvar sua chave privada em um local seguro.



Se você selecionar *Create a new Security Group based on the Seller settings*, a porta TCP 10051 (Zabbix server) será permitida. Observação: se quiser acessar a instância via SSH, também será necessário permitir conexões SSH nas configurações do Security Group.

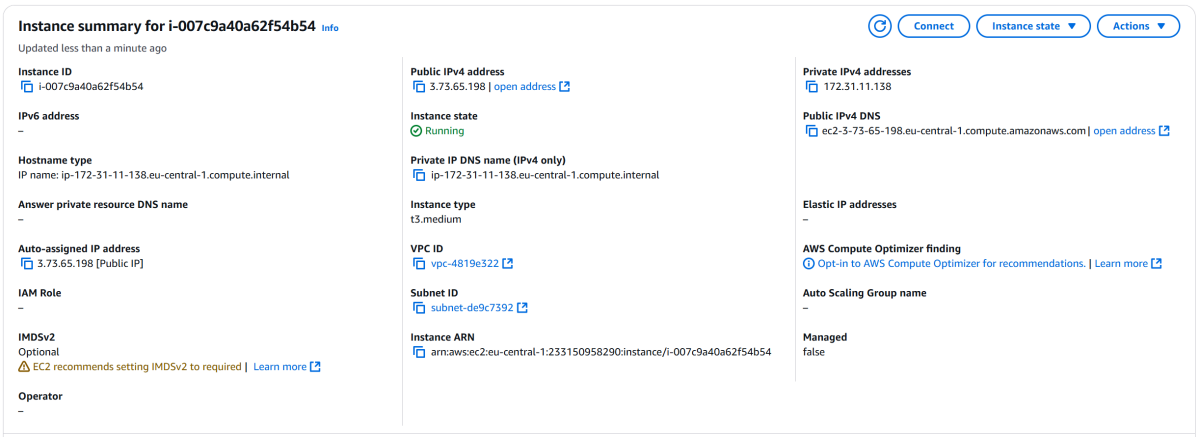


Por fim, clique no botão *Launch*. Parabéns, seu Zabbix proxy está sendo implantado.



Configurando e acessando o proxy do Zabbix

Abra a seção Instances no console do EC2 e selecione a instância do proxy do Zabbix recém-implantada.



Conecte-se ao endereço IPv4 público ou ao DNS IPv4 público do proxy do Zabbix usando SSH. O nome de usuário SSH padrão é `rocky`. Não se esqueça de usar o par de chaves gerado anteriormente.

A geração das credenciais pode levar alguns minutos. Aguarde alguns minutos antes de tentar se conectar.



O arquivo de configuração do proxy do Zabbix pode ser encontrado em `/etc/zabbix/zabbix_proxy.conf`.

O arquivo de configuração pode ser aberto com um editor de texto de sua preferência, por exemplo, vi:

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

A configuração inicial do proxy consiste em modificar os parâmetros `ProxyMode`, `Server` e `Hostname` no arquivo de configuração do proxy do Zabbix.

O modo do proxy pode ser alterado editando o parâmetro `ProxyMode`. Defina `ProxyMode=0` para o modo ativo ou `ProxyMode=1` para o modo passivo.

O parâmetro `Server` deve conter o endereço IP do server do Zabbix. Por exemplo:

```
Server=192.168.50.141
```

O parâmetro `Hostname` deve conter o nome do host do proxy do Zabbix. O nome do host definido aqui deve corresponder ao nome do proxy fornecido durante a configuração do proxy no frontend do Zabbix. Por exemplo:

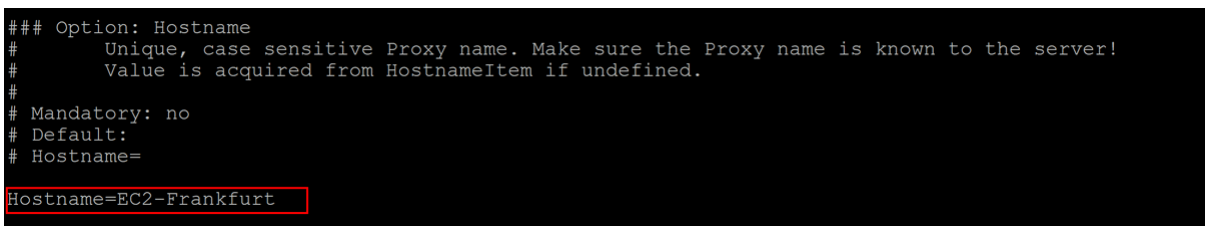
```
Hostname=EC2-Frankfurt
```

Quando a configuração estiver concluída, reinicie o proxy do Zabbix com o seguinte comando:

```
sudo systemctl restart zabbix-proxy
```

Observação: use sudo para editar arquivos de configuração e executar comandos na linha de comando.



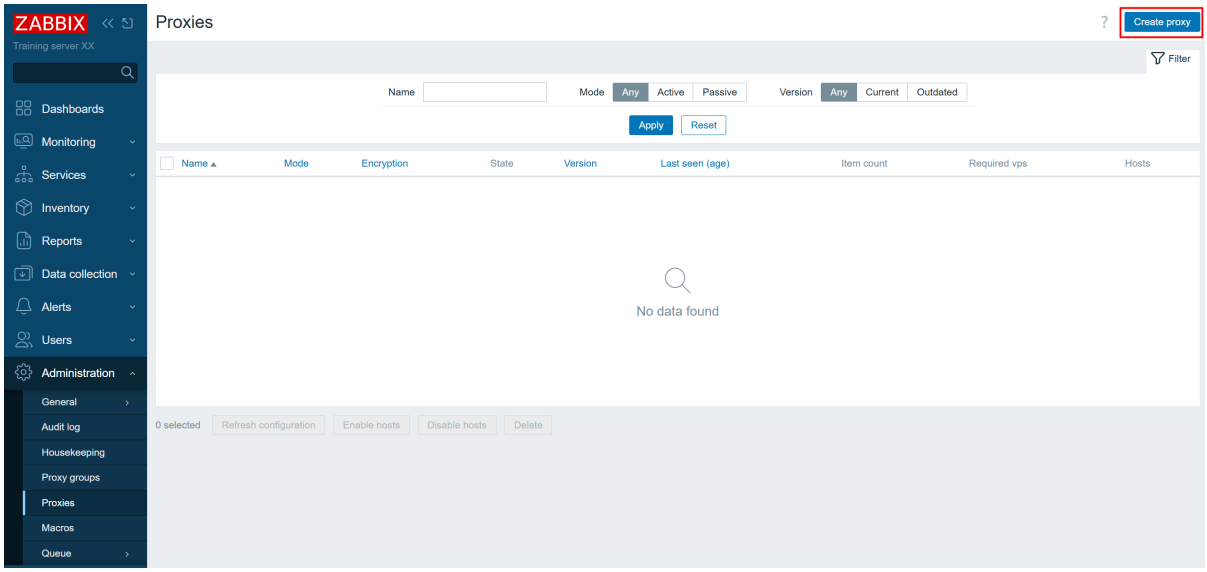


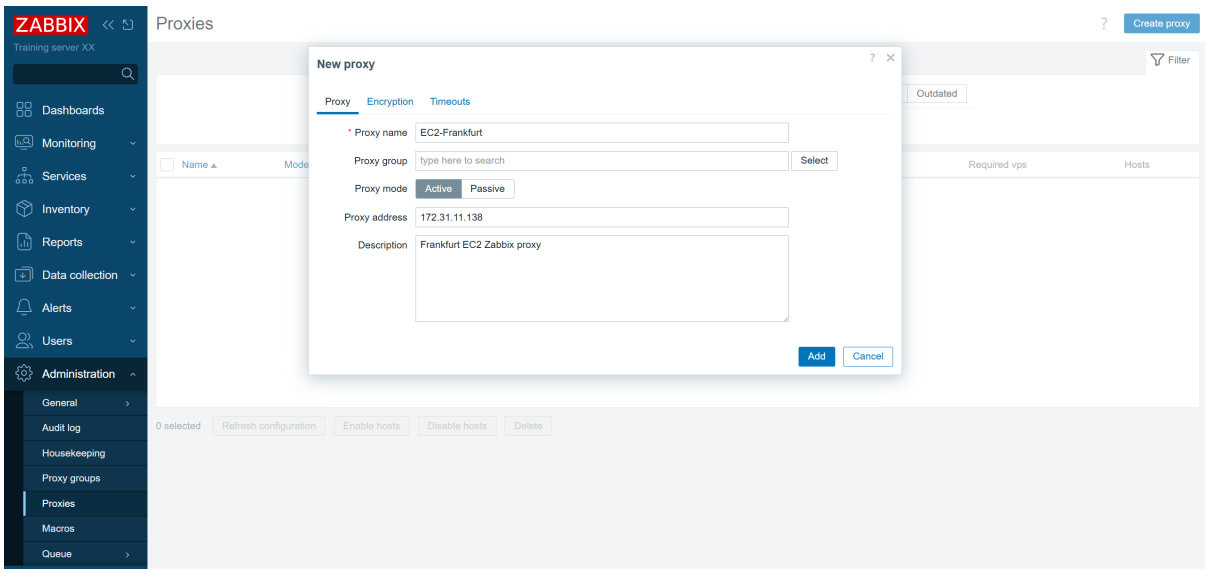
Abra o frontend do Zabbix e acesse *Administration* > *Proxies*. Clique no botão *Create proxy*.

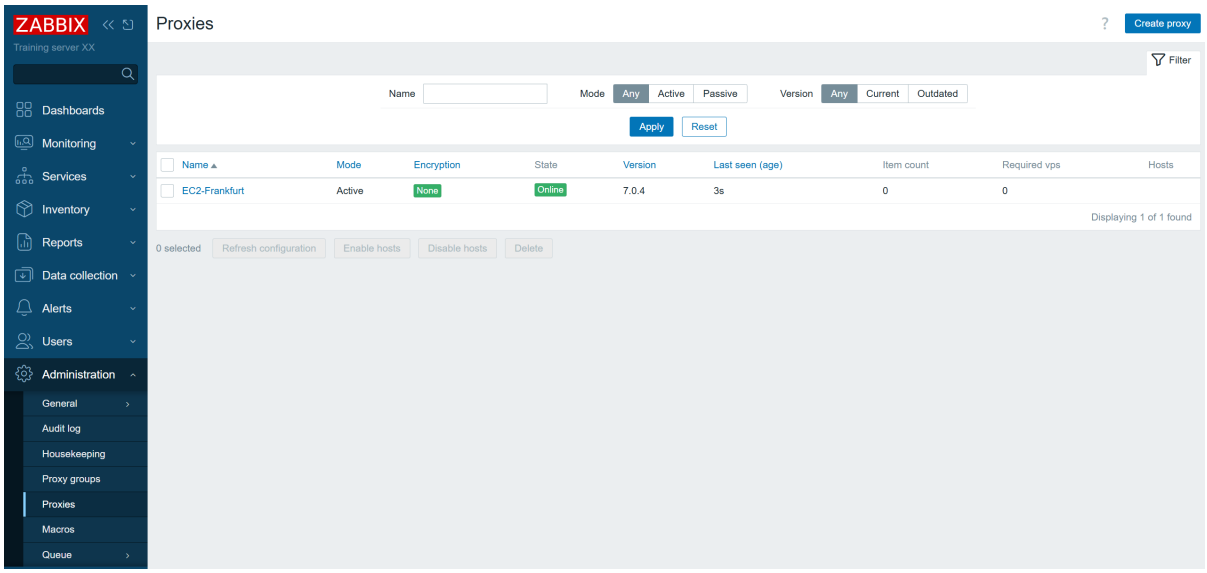
Insira o mesmo nome do proxy usado como valor do parâmetro `Hostname` no arquivo de configuração do proxy.

Selecione o modo do proxy e forneça o endereço IP do proxy (opcional para proxies ativos).

Clique em *Create proxy*.







Localização do banco de dados do proxy

O banco de dados SQLite do proxy é armazenado em `/tmp/zabbix_proxy.sqlite3` e especificado pelo parâmetro `DBName` no arquivo de configuração do proxy do Zabbix.

Guia de implantação do Zabbix server no Azure

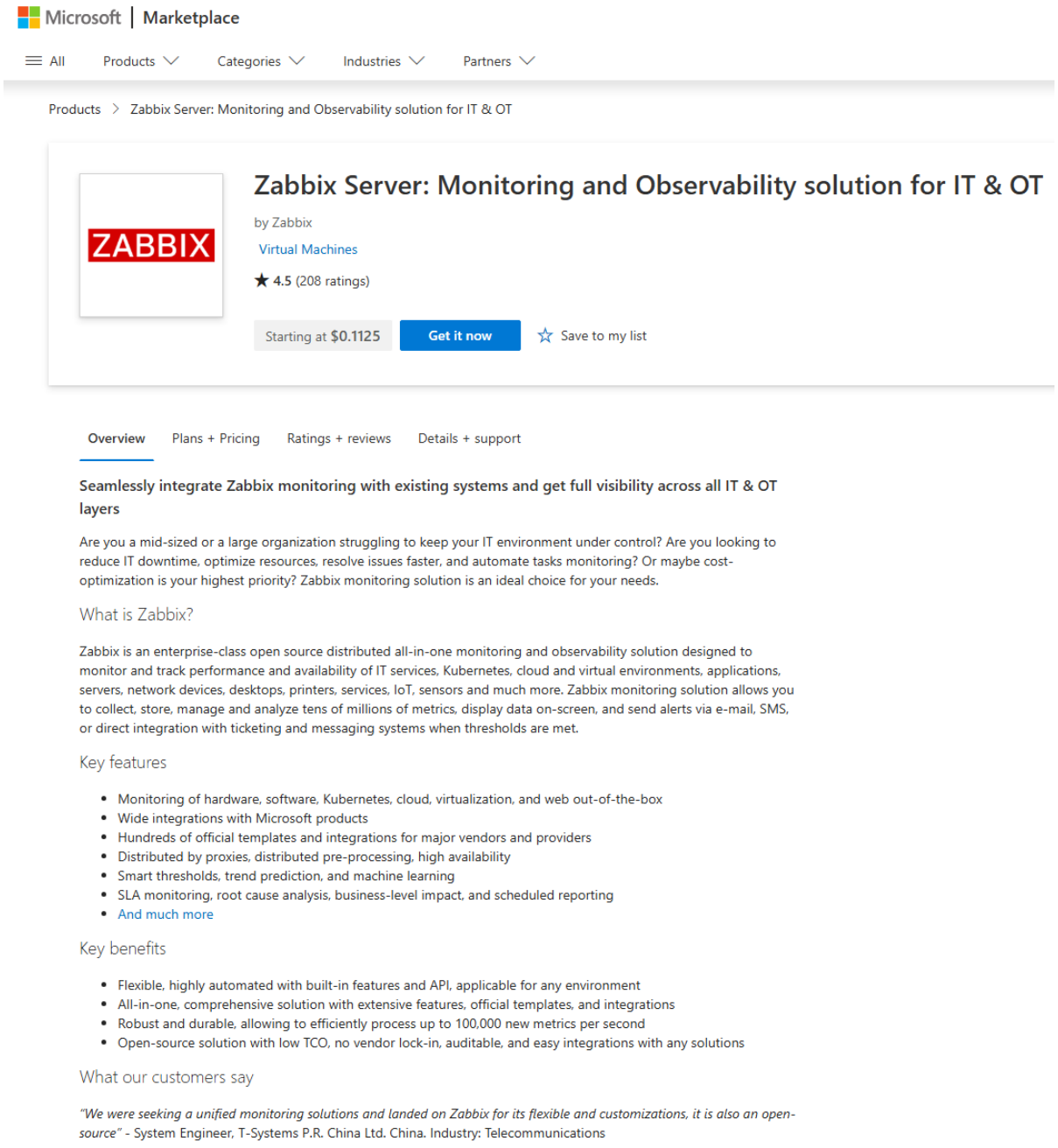
Visão geral

Esta página descreve como implantar o Zabbix server usando o Azure.

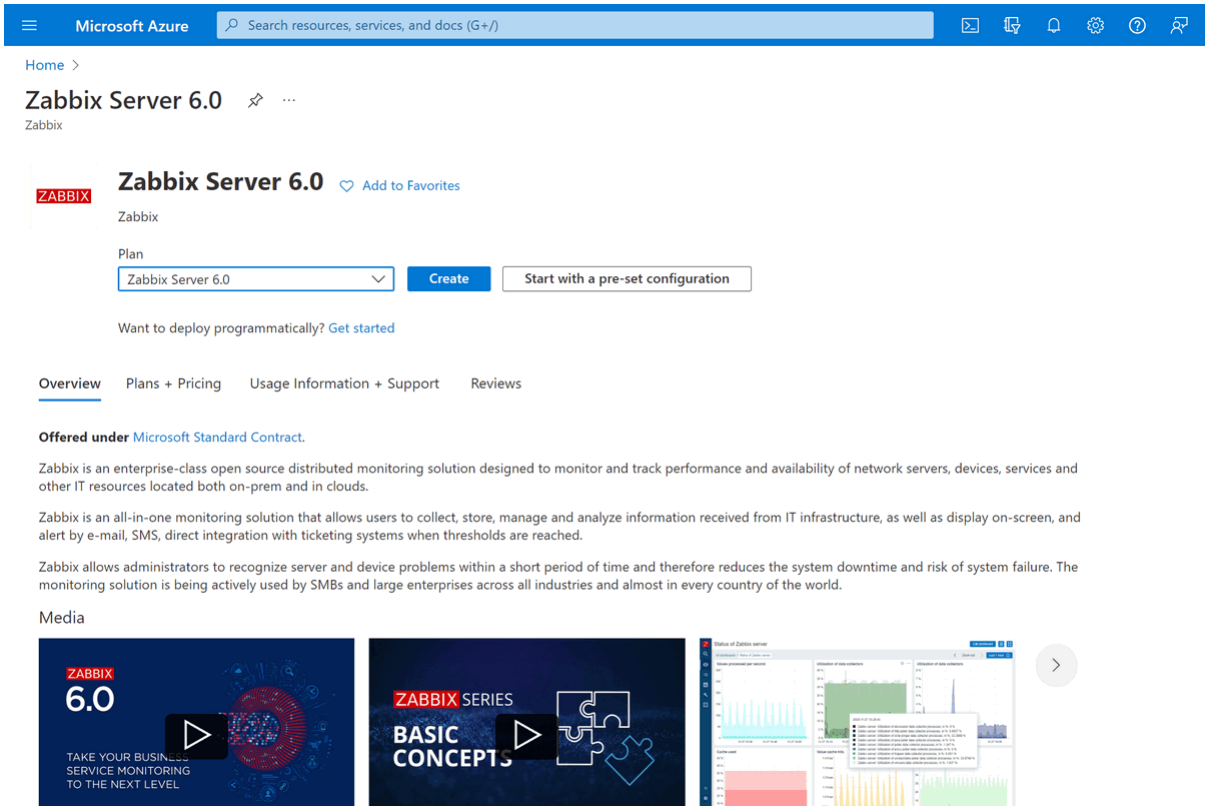
> **Attention:**
> Várias versões do Zabbix estão disponíveis para implantação na plataforma. Leia a documentação da mesma versão do Zabbix que você implantou. A documentação de diferentes versões do Zabbix pode ser selecionada no menu suspenso de versões.

Implantando o appliance

Para implantar o appliance do Zabbix, primeiro acesse a página do Zabbix no Azure Marketplace. Revise a visão geral e clique em *Get it now*.



Revise a visão geral da implantação, os planos de preços e outros detalhes. Em seguida, clique em *Create* ou em *Start with a pre-set configuration* para implantar seu appliance do Zabbix.

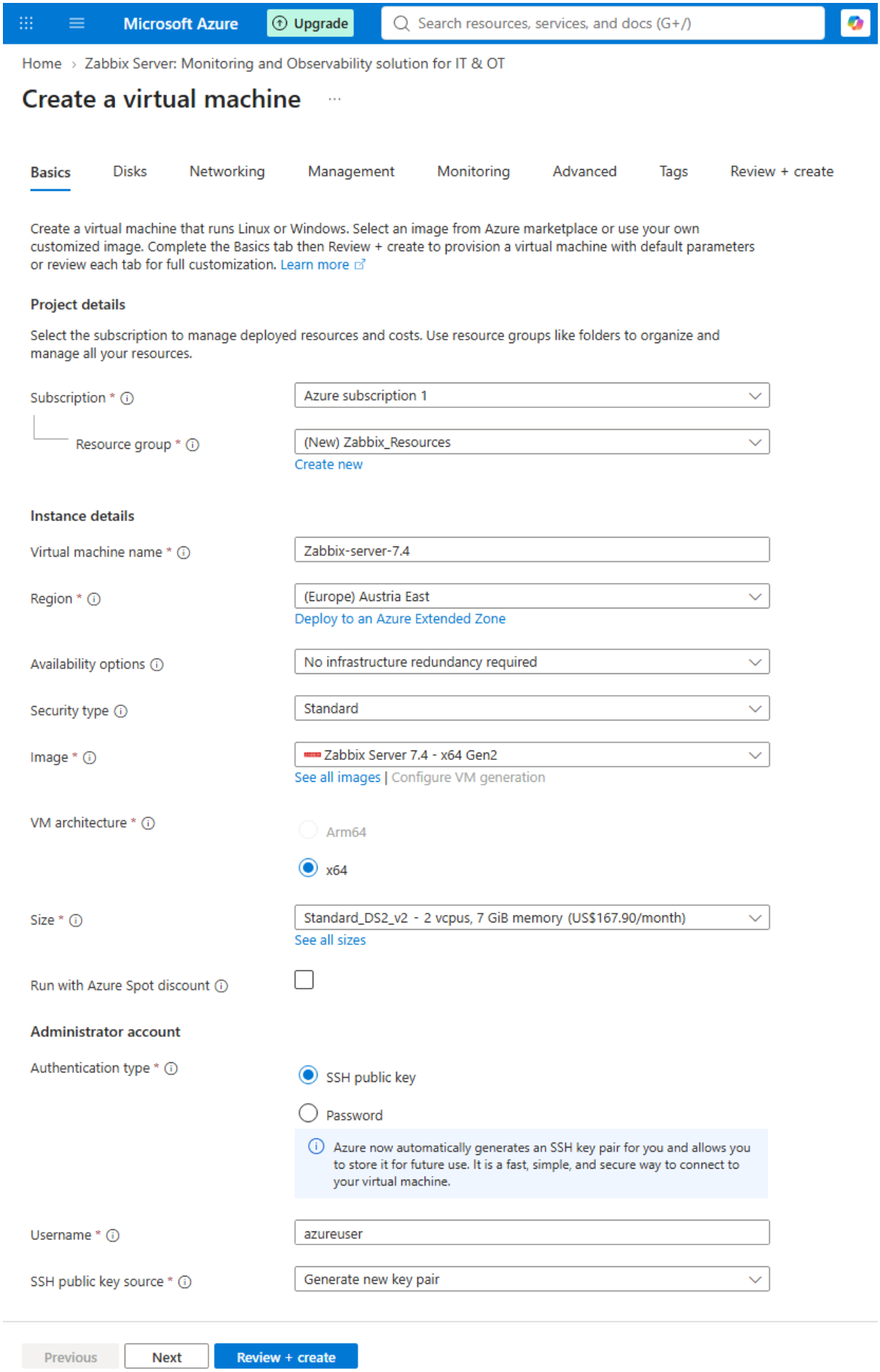


Configure a VM selecionando o tamanho da VM, criando uma conta de administrador e definindo as configurações de disco, rede, gerenciamento e outras. Essas configurações podem variar de acordo com os requisitos da sua infraestrutura.

Você pode usar a configuração a seguir como exemplo:

Aba *Basics*:

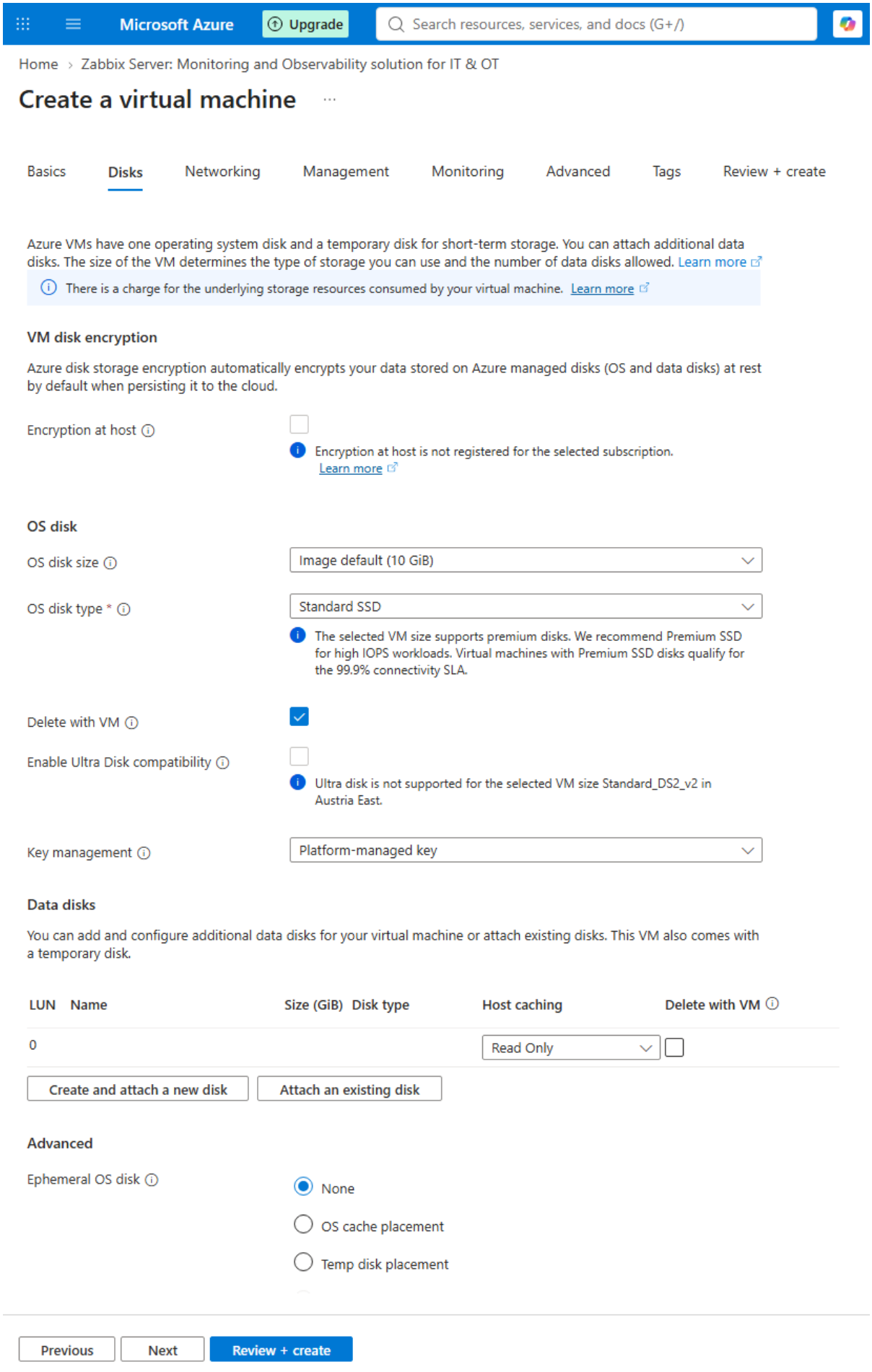
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Aba *Disks*:

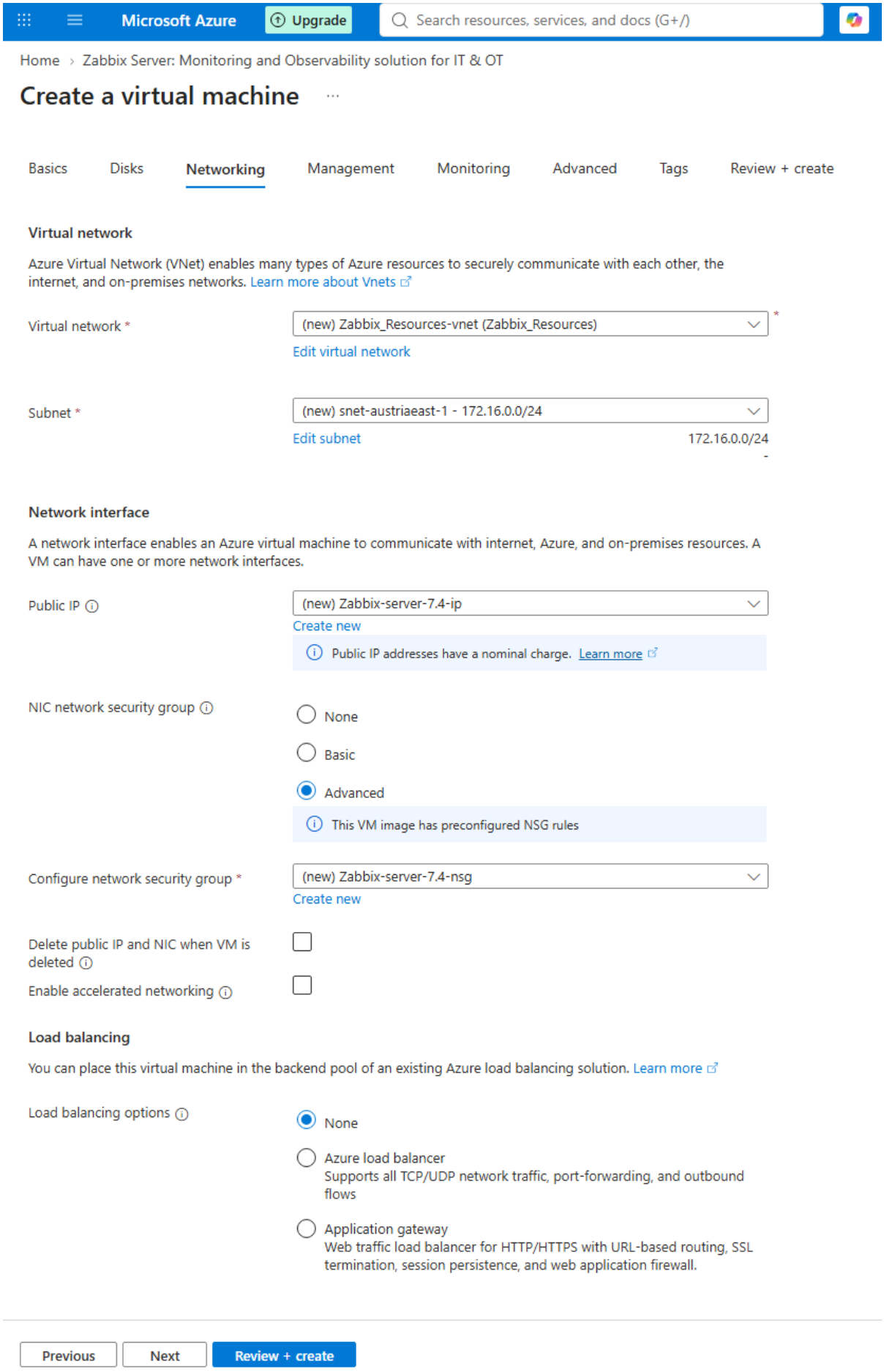
- *Encryption at host*: desmarcado
- *OS disk type*: Standard SSD

- *Delete with VM*: marcado
- *Enable Ultra Disk compatibility*: desmarcado

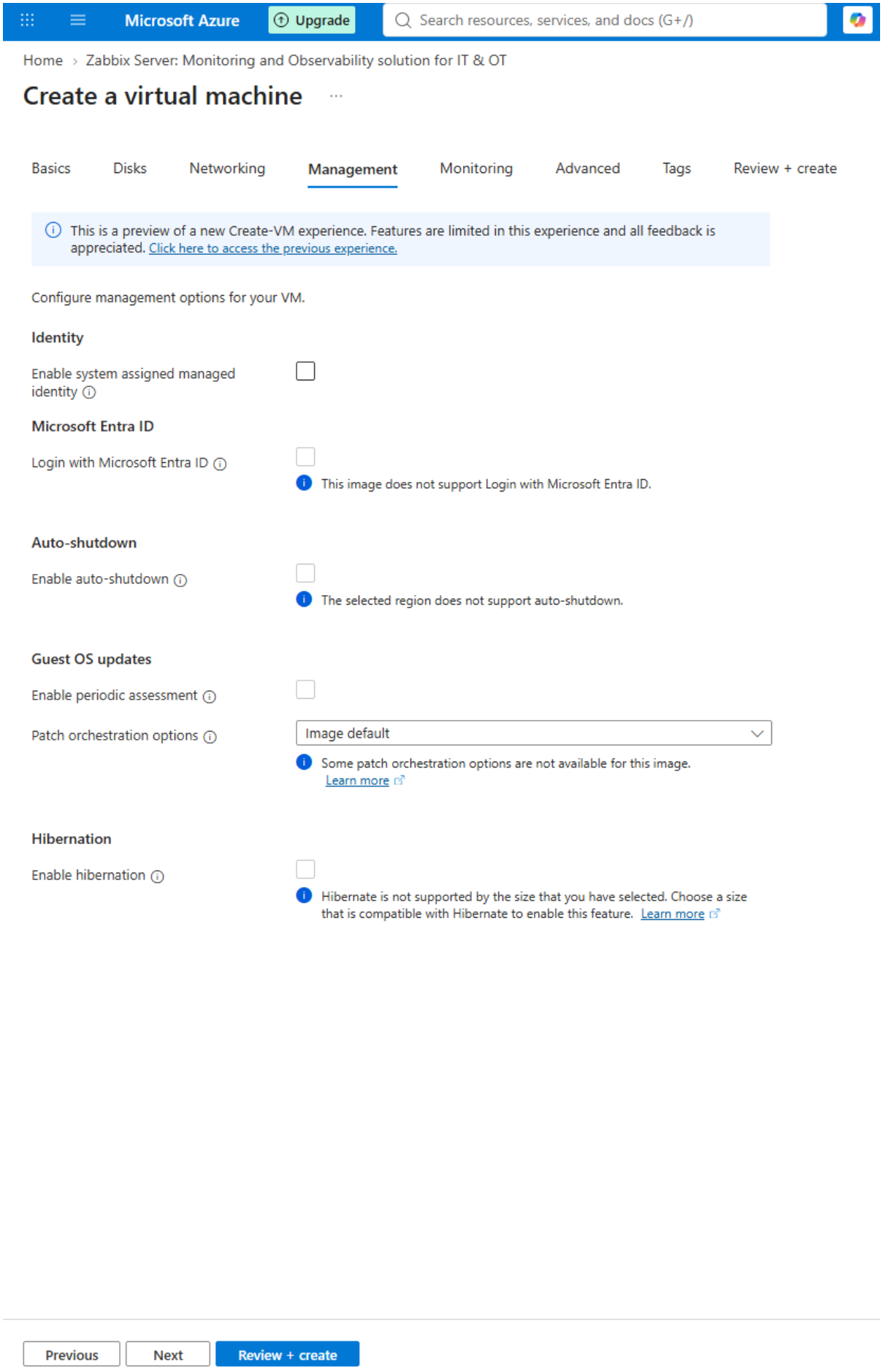


Aba *Networking*:

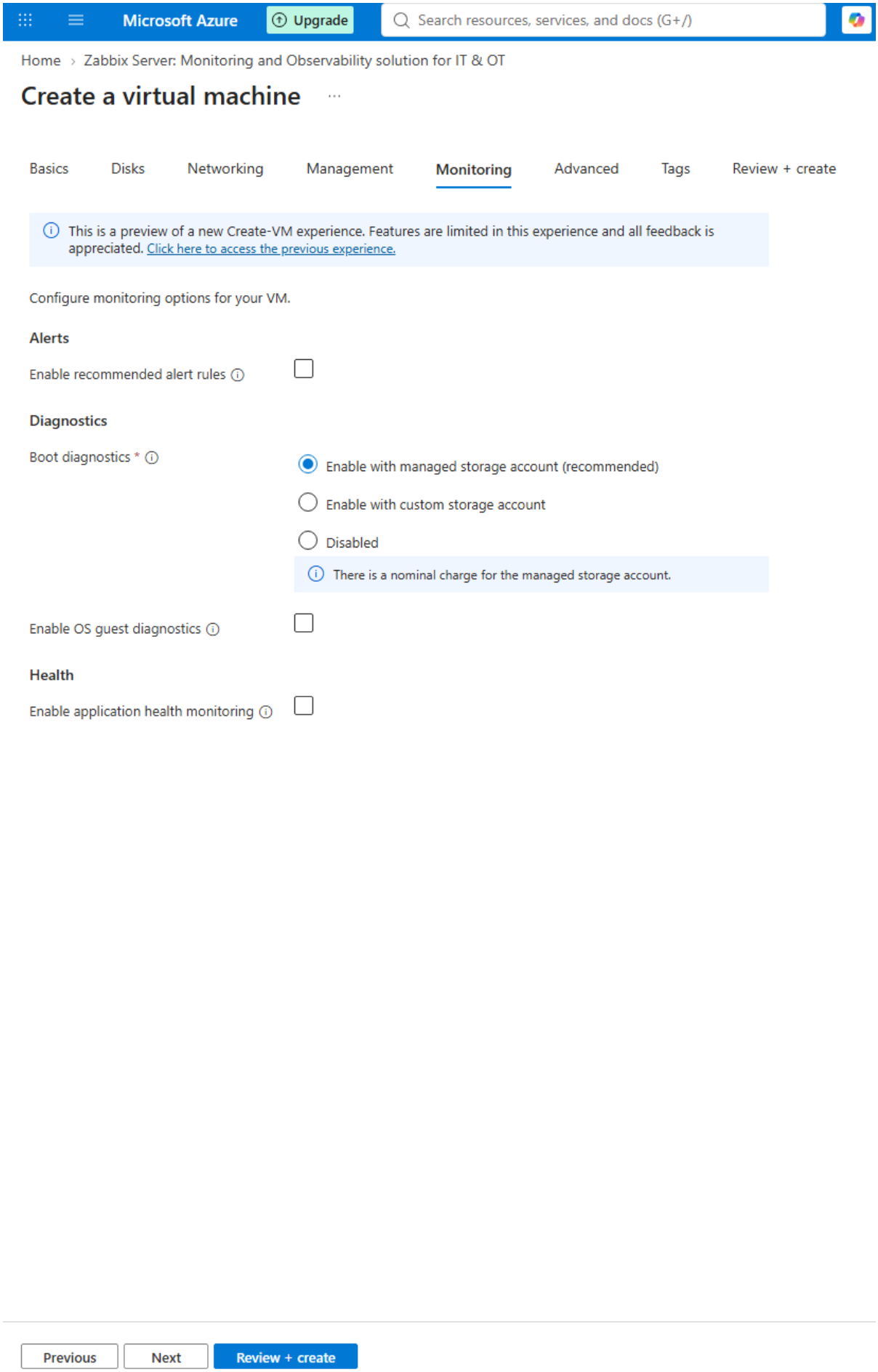
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



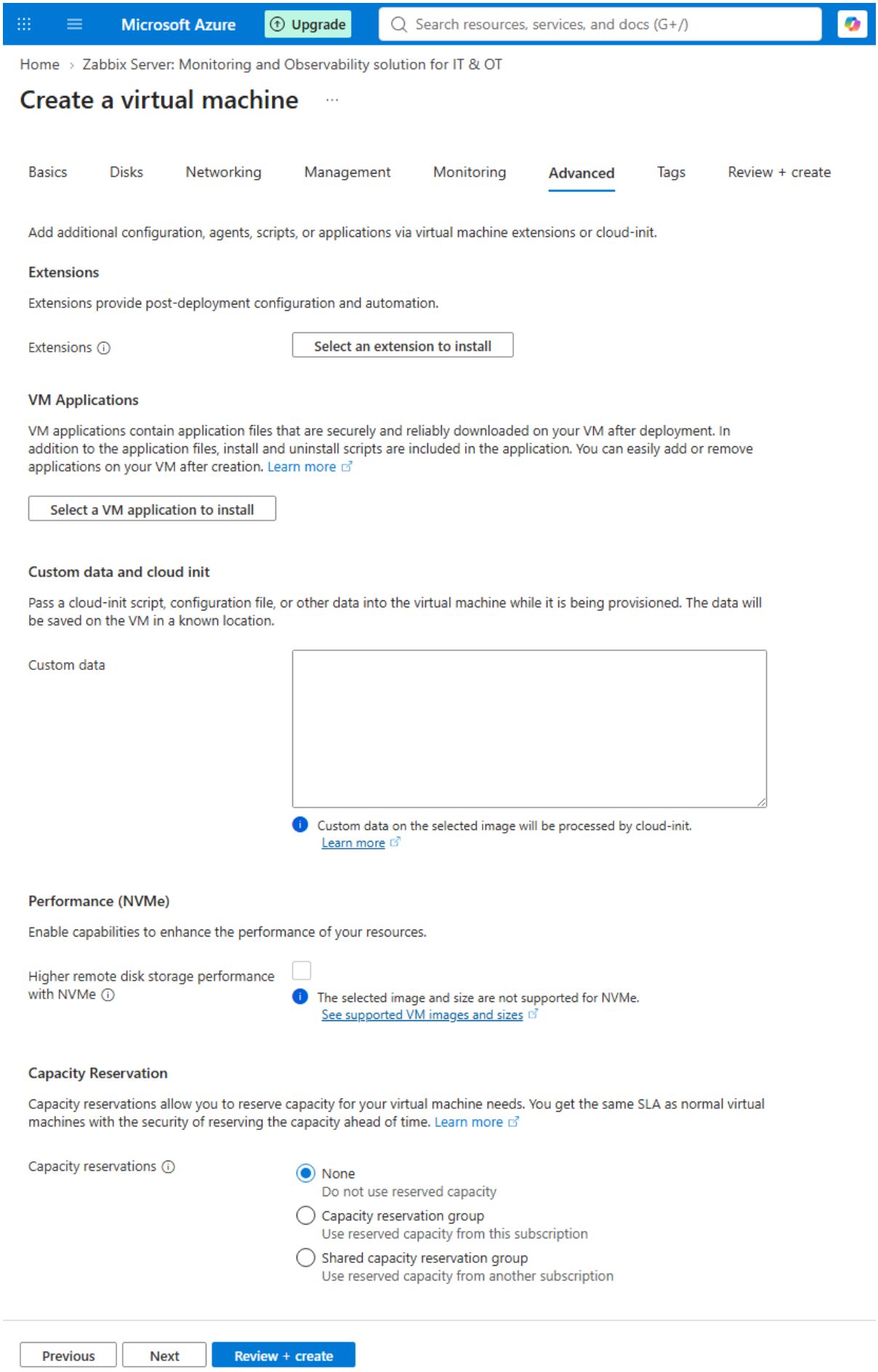
Aba *Management*: os parâmetros podem ser mantidos com os valores padrão.



Aba *Monitoring*: os parâmetros podem ser mantidos com os valores padrão.



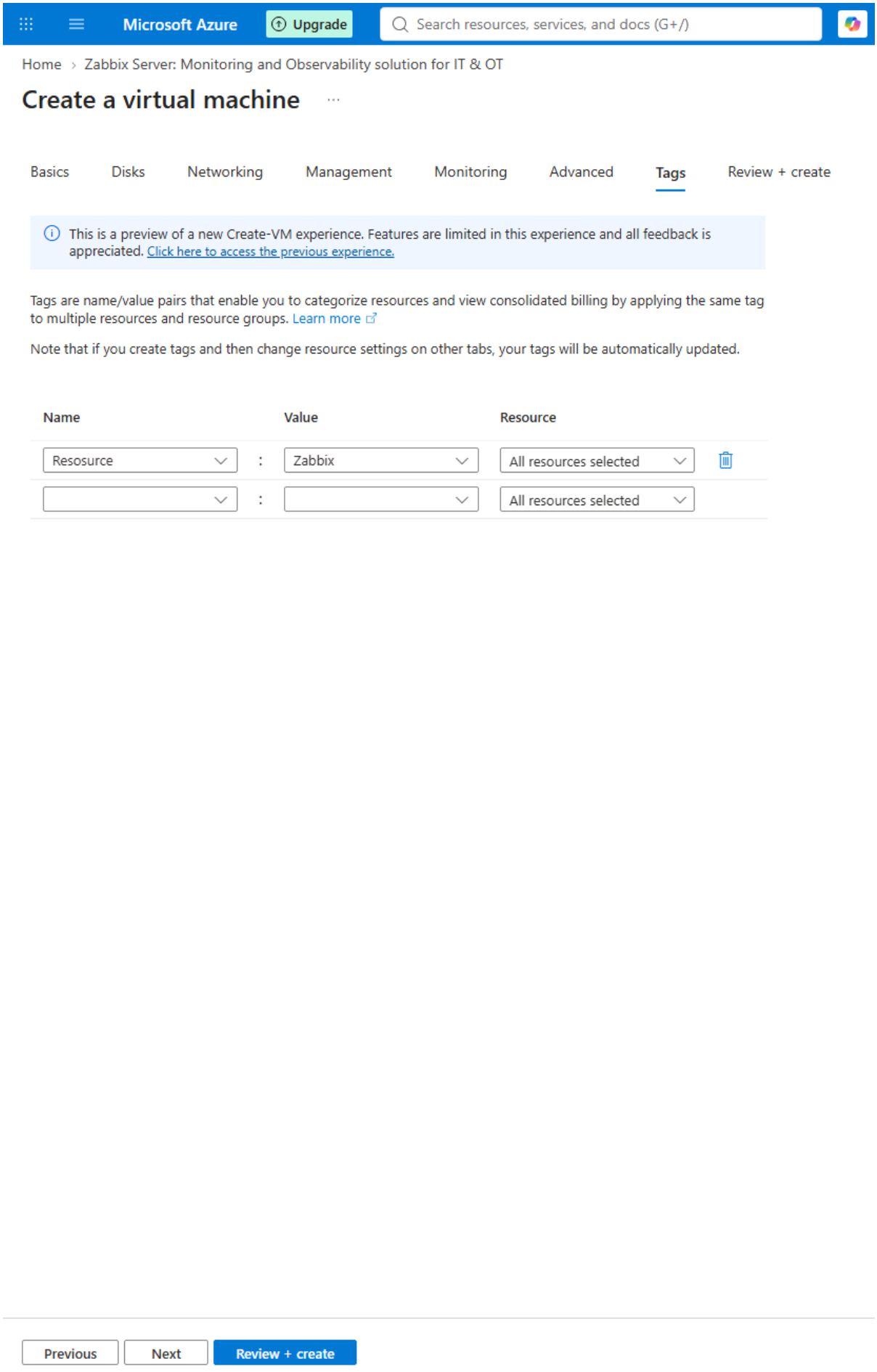
Aba *Advanced*: os parâmetros podem ser mantidos com os valores padrão.



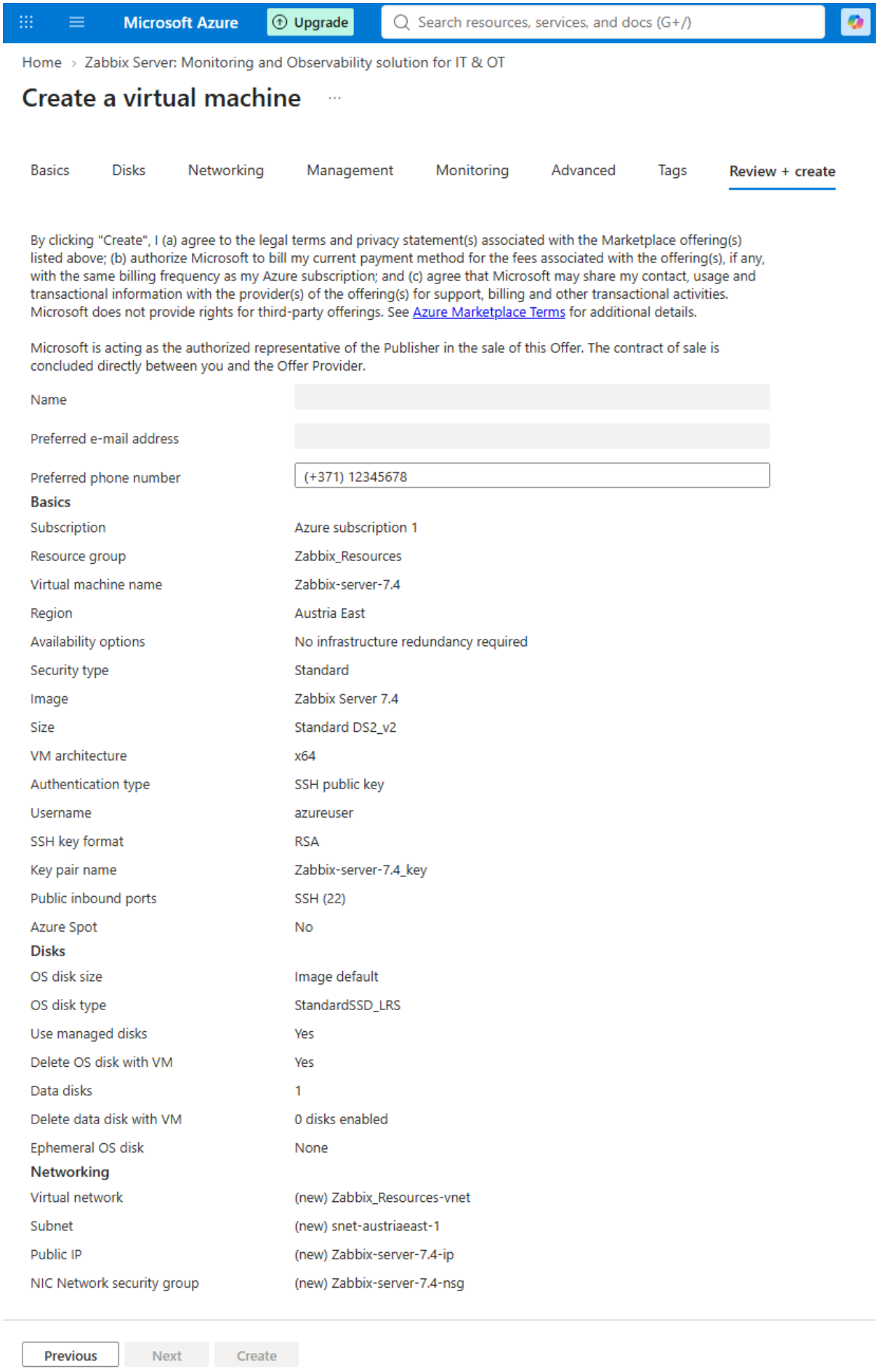
Aba *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Confirme os detalhes da VM e crie a VM.



Você será redirecionado para a página de status da implantação da VM. Aguarde até que a implantação seja concluída.

Quando a implantação for concluída, abra o recurso e anote seu endereço IP público. Agora, você deverá se conectar ao server do Zabbix usando SSH (ou outro método disponível) para obter as credenciais de login do frontend do Zabbix.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

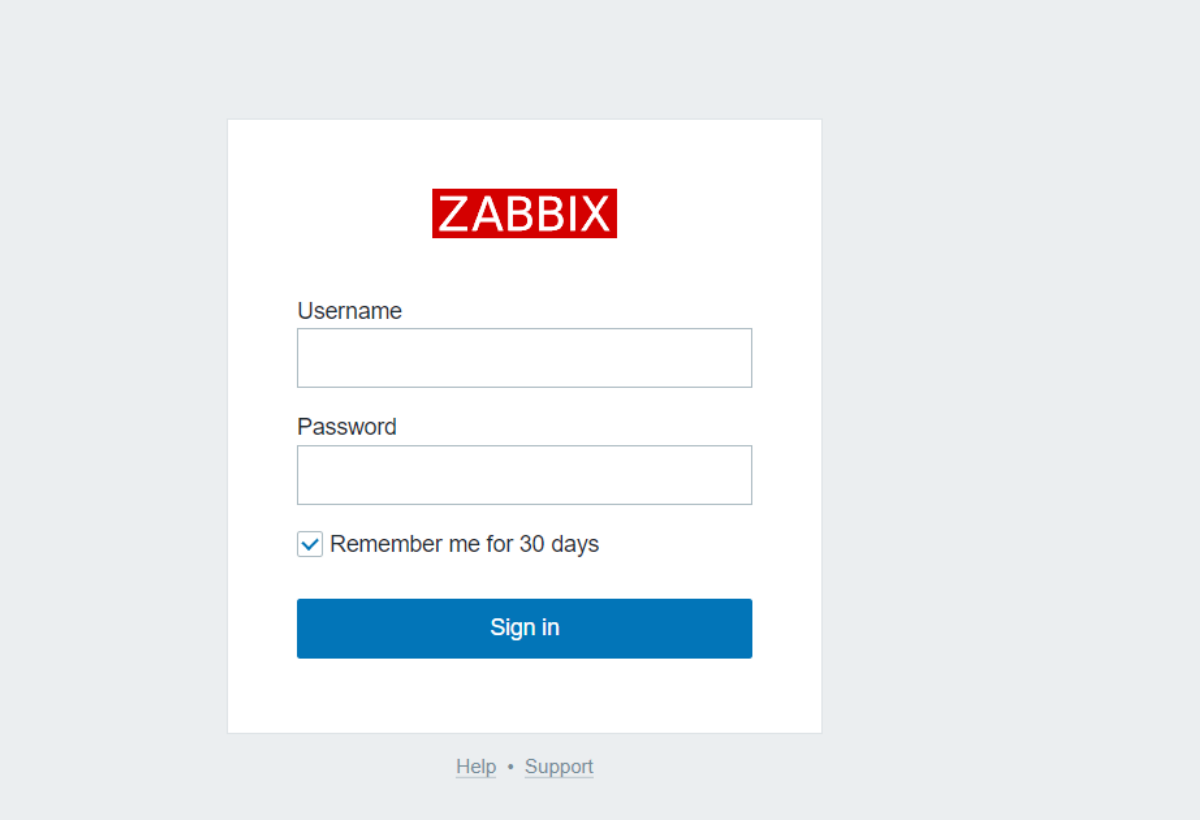

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Acesse o frontend do Zabbix abrindo o endereço IP público do appliance do Zabbix em um navegador da web. Para fazer login, use o nome de usuário e a senha obtidos na etapa anterior.



Acessando o banco de dados

Você pode acessar o banco de dados do Zabbix usando o usuário `root` sem inserir uma senha. As credenciais estão armazenadas no arquivo de configuração `/root/.my.cnf`.

Guia de implantação do Azure para o proxy do Zabbix

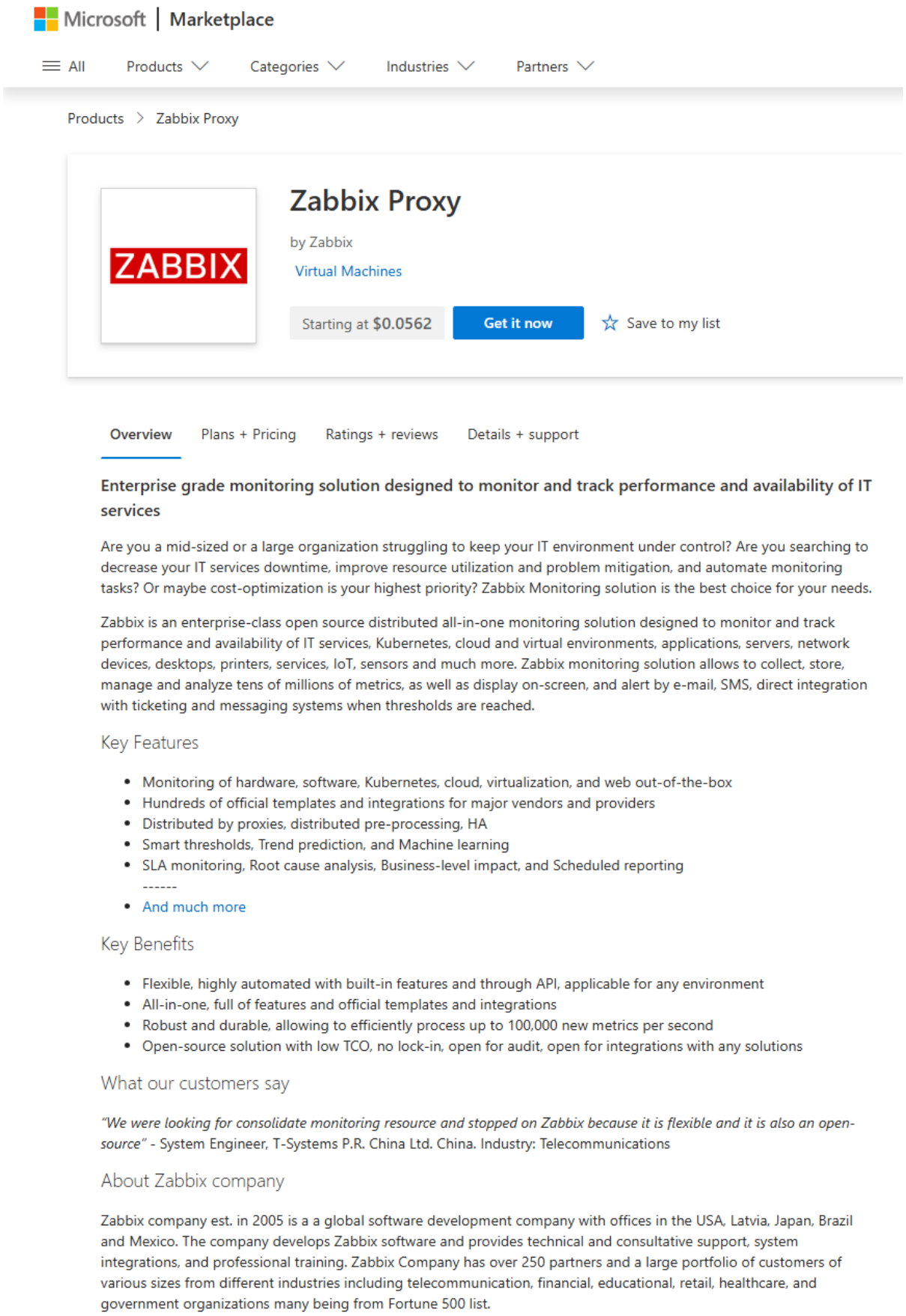
Visão geral

Esta página descreve como implantar o Zabbix proxy usando o Azure.

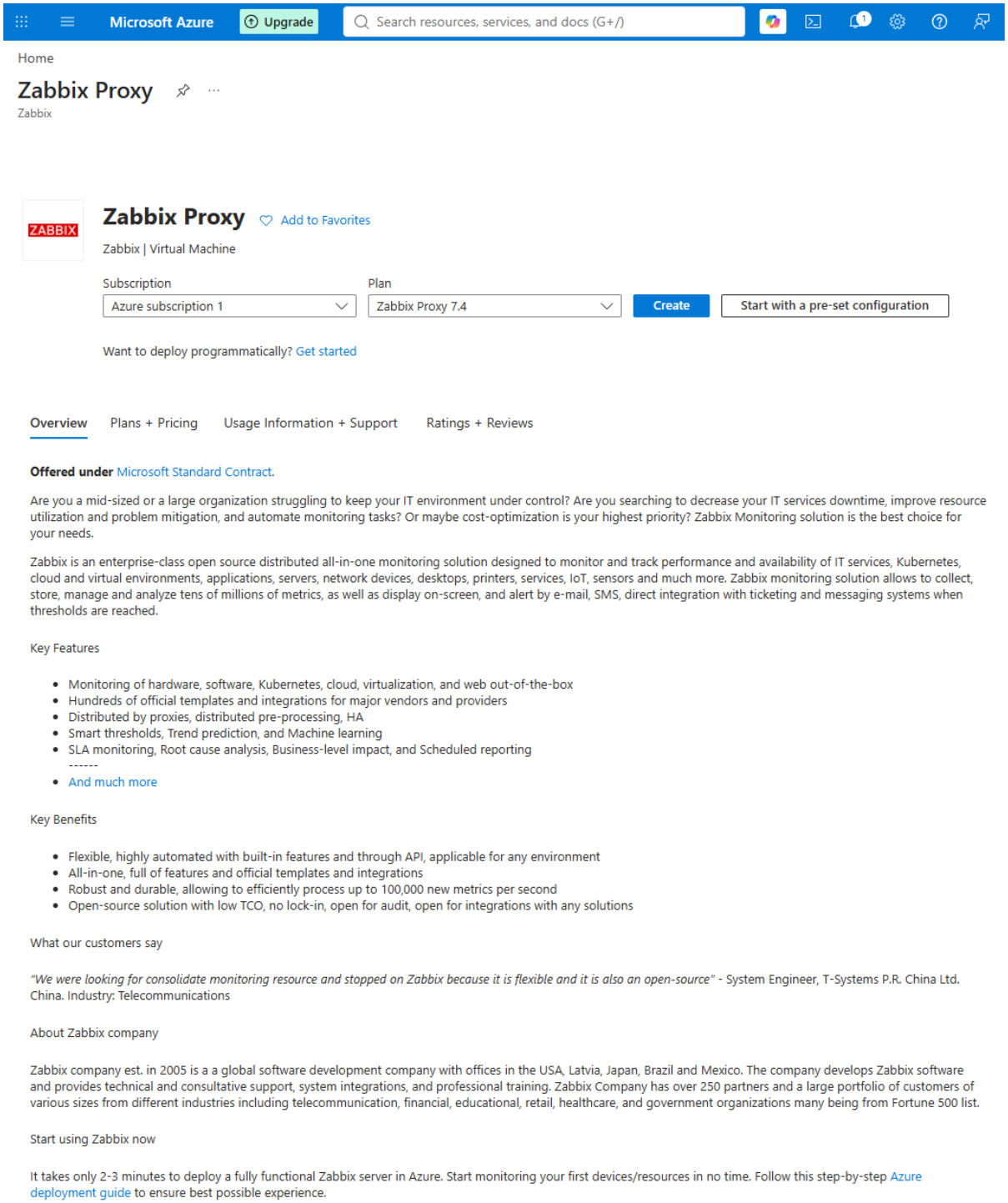
Attention:
Várias versões do Zabbix estão disponíveis para implantação na plataforma. Leia a documentação referente à mesma versão do Zabbix que você implantou. A documentação de diferentes versões do Zabbix pode ser selecionada no menu suspenso de versões.

Implantação do Zabbix proxy

Para implantar o Zabbix proxy, primeiro acesse a página do Zabbix no Azure Marketplace. Clique em *Get it now*.



Aqui, você pode consultar informações adicionais sobre a implantação atual, como a visão geral, os planos de preços e muito mais. Clique em *Create* ou em *Start with a pre-set configuration*.

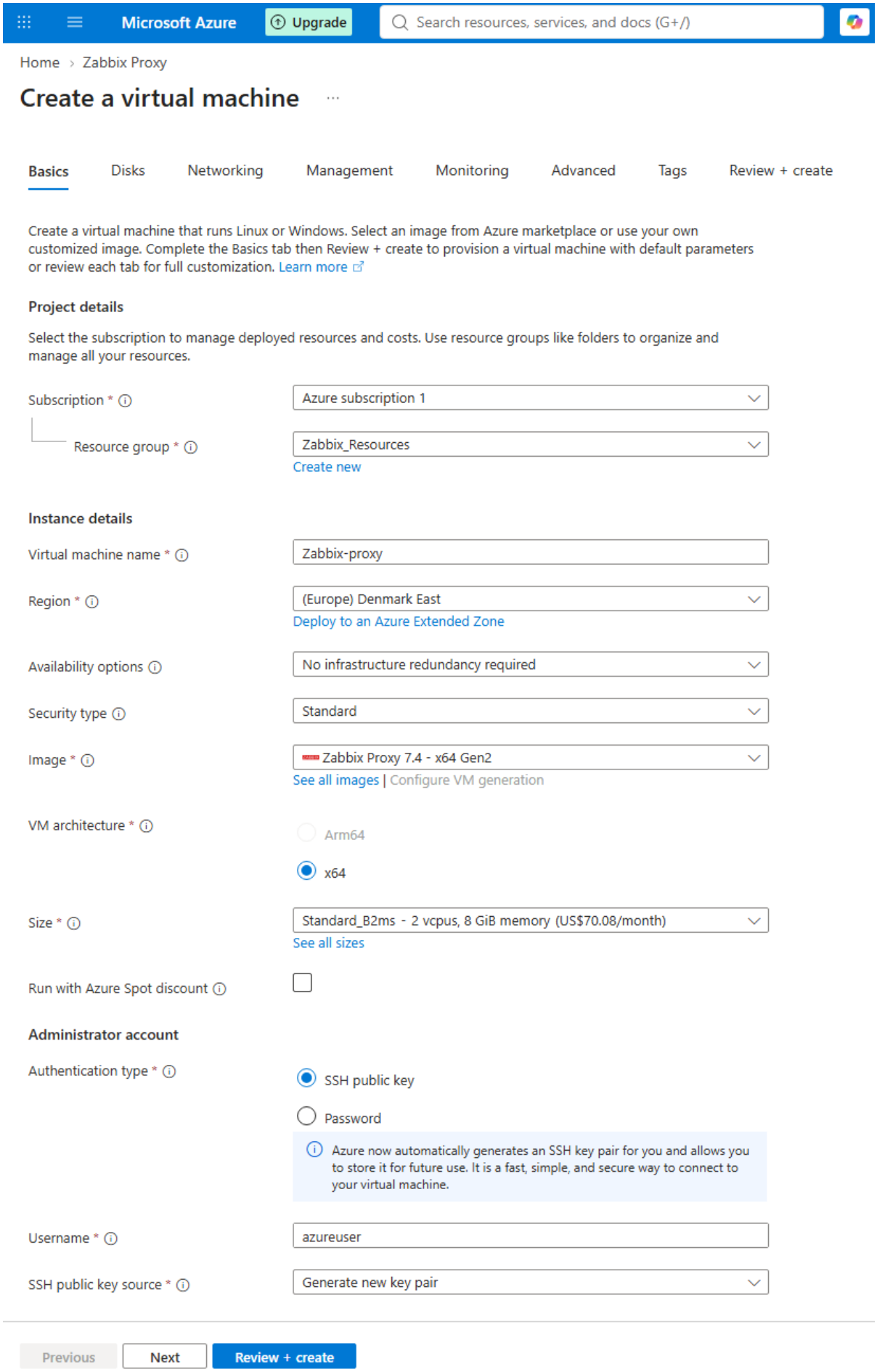


Prossiga com a configuração detalhada da VM: selecione o tamanho da VM, a conta de administrador, o disco, a rede, o gerenciamento e outras configurações. Essas configurações podem variar de acordo com os requisitos da sua infraestrutura.

Você pode usar a configuração a seguir como exemplo:

Aba *Basics*:

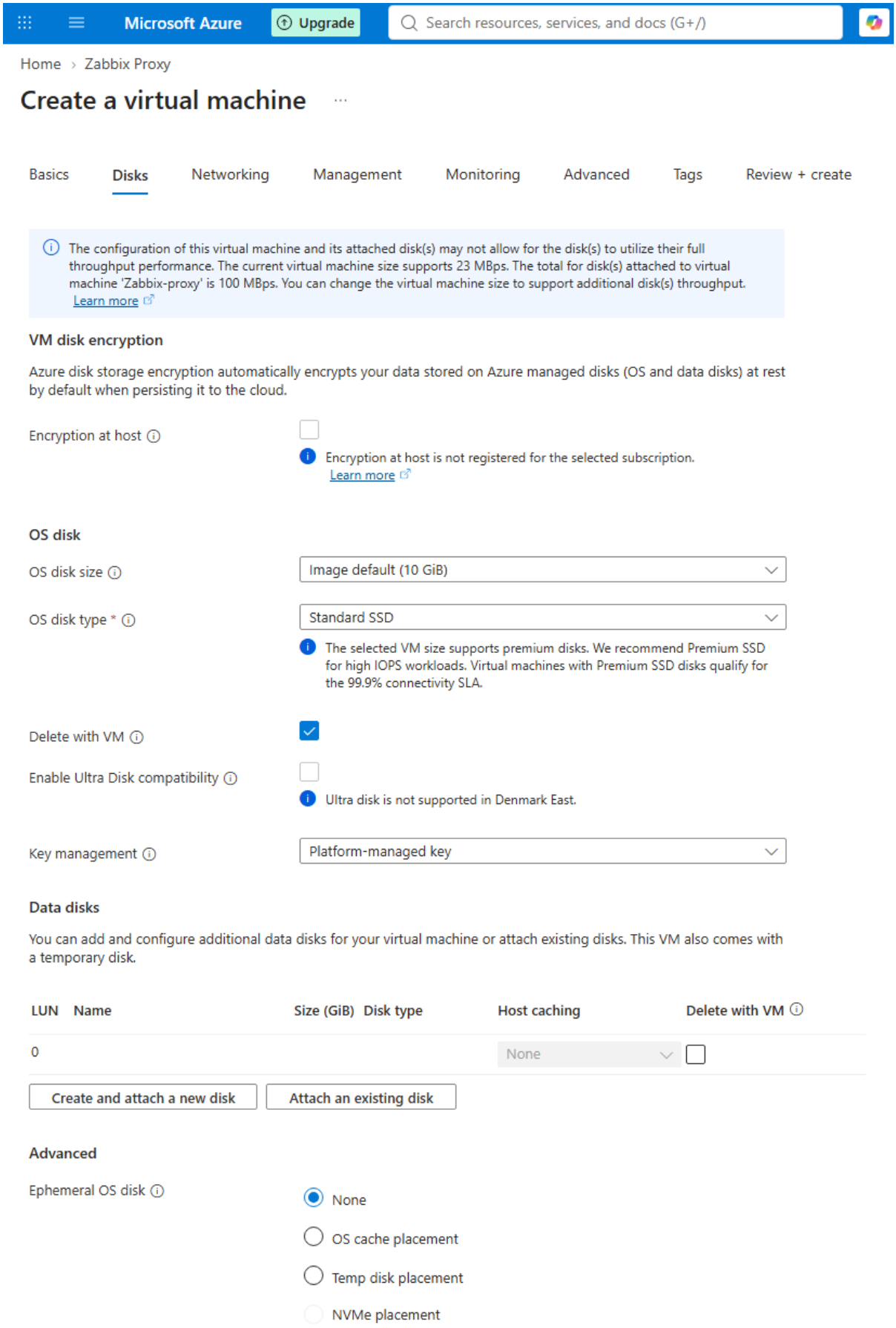
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Aba *Disks*:

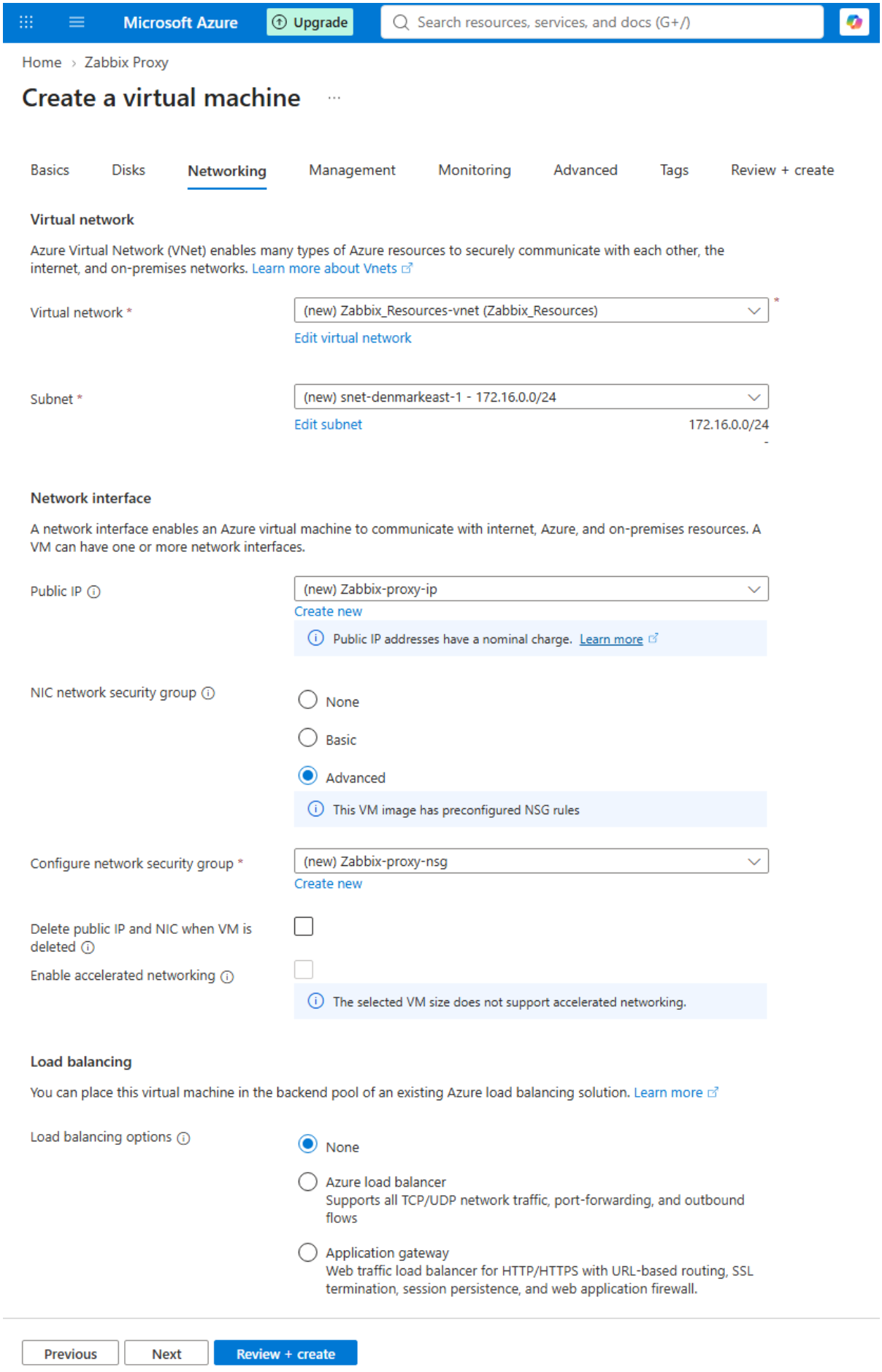
- *Encryption at host*: unchecked

- *OS disk type*: Standard SSD
- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

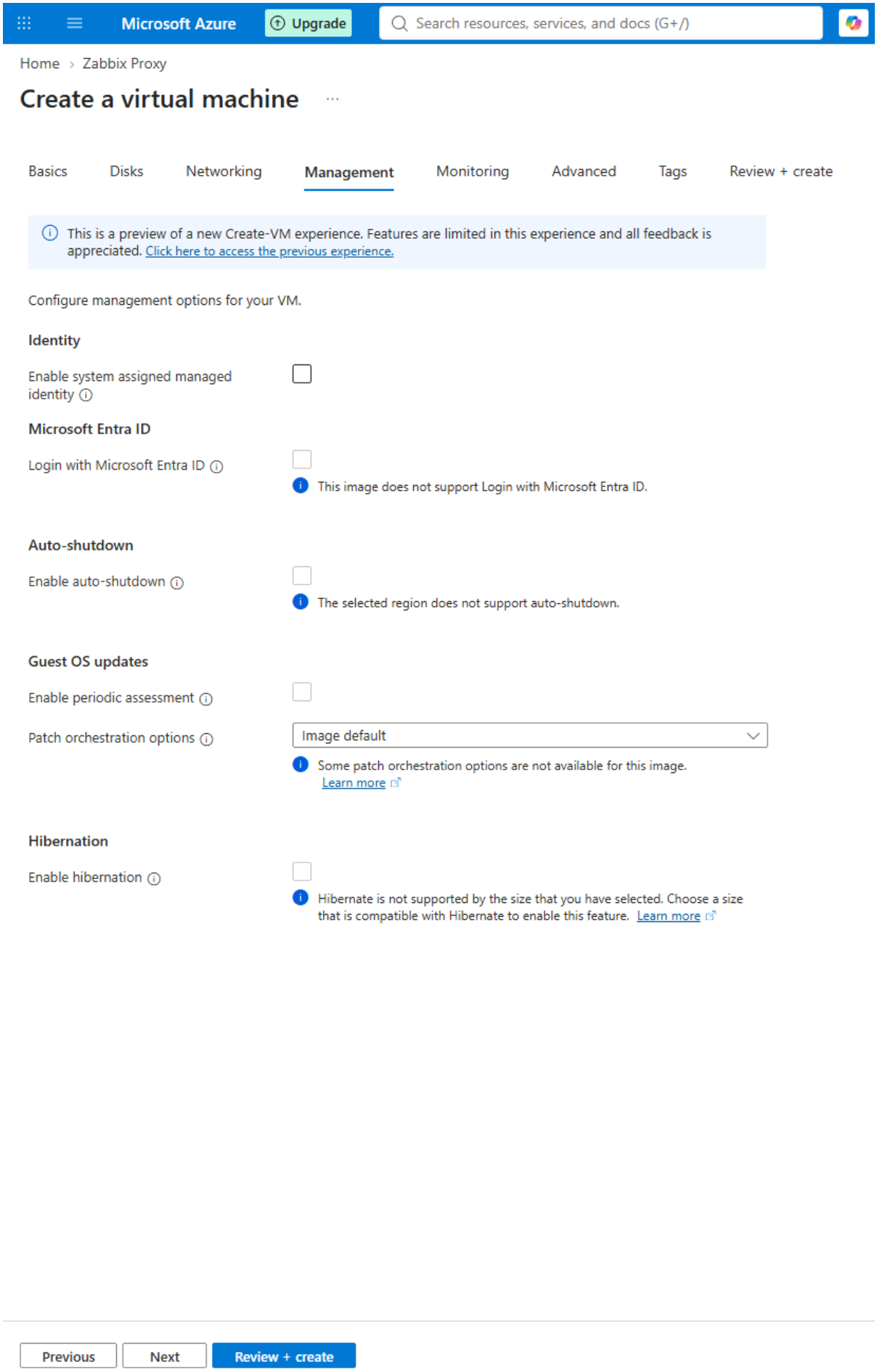


Aba *Networking*:

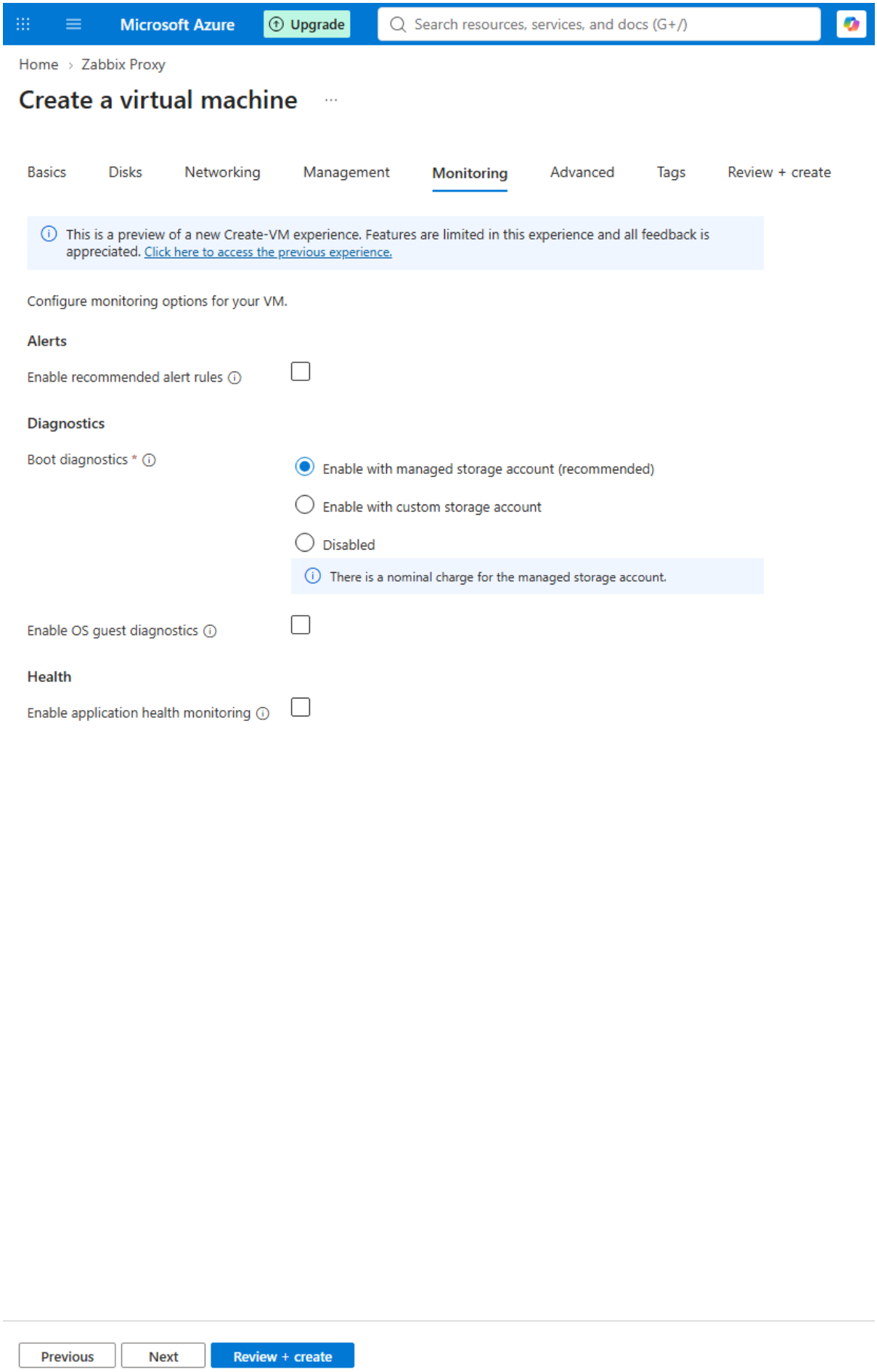
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



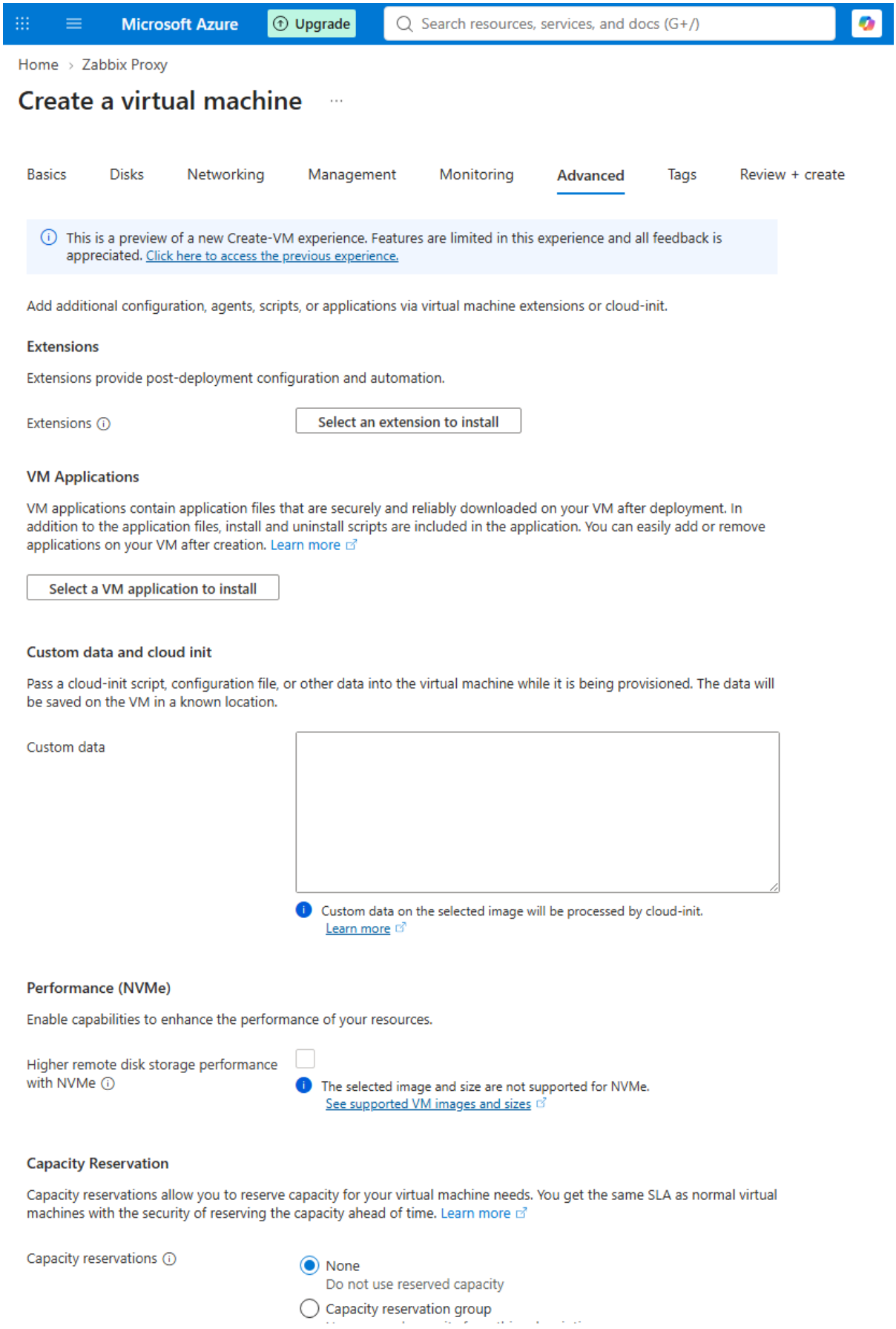
Aba *Management*: os parâmetros podem ser mantidos com os valores padrão.



Aba *Monitoring*: os parâmetros podem ser mantidos com os valores padrão.



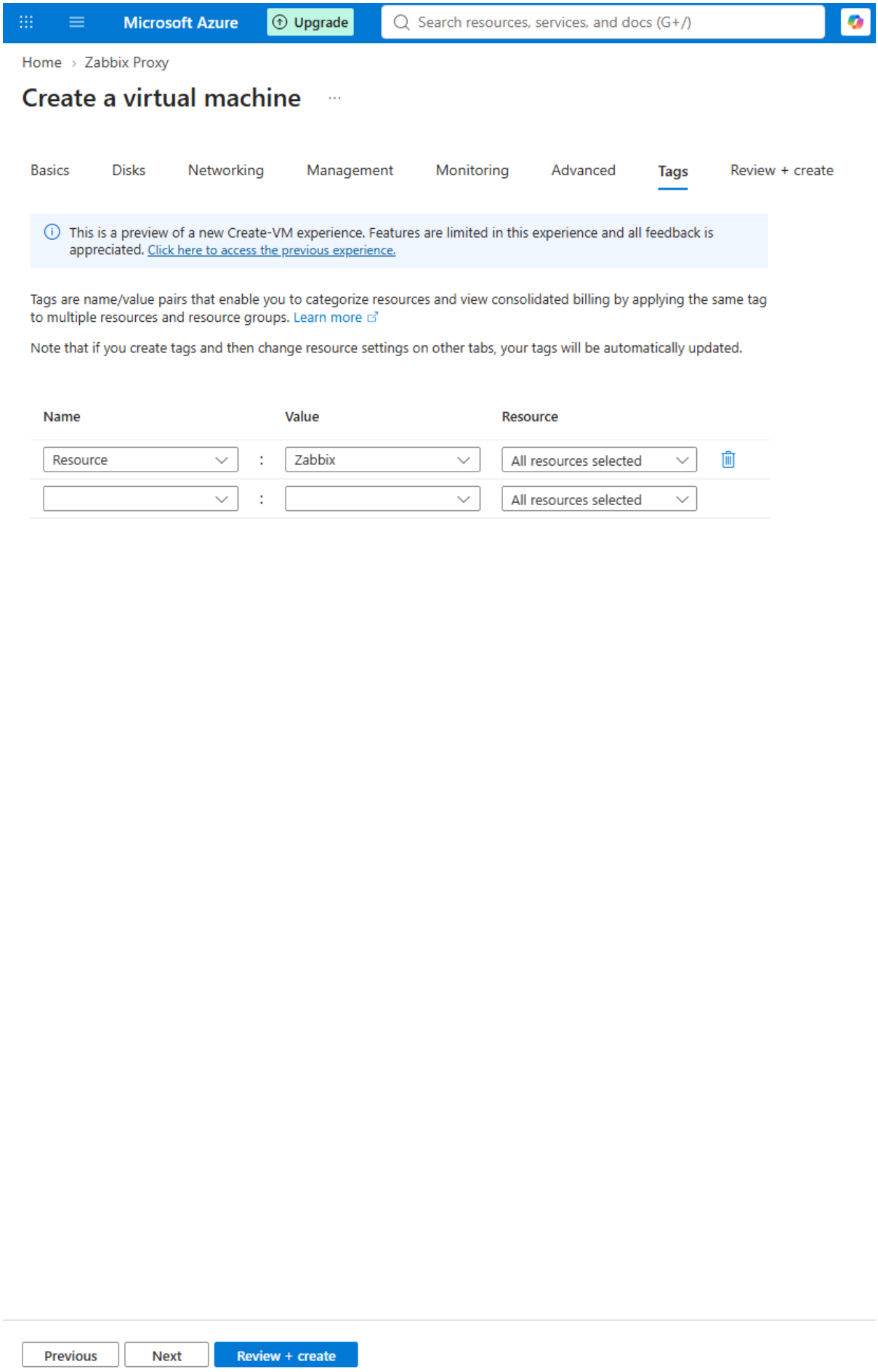
Aba *Advanced*: os parâmetros podem ser mantidos com os valores padrão.



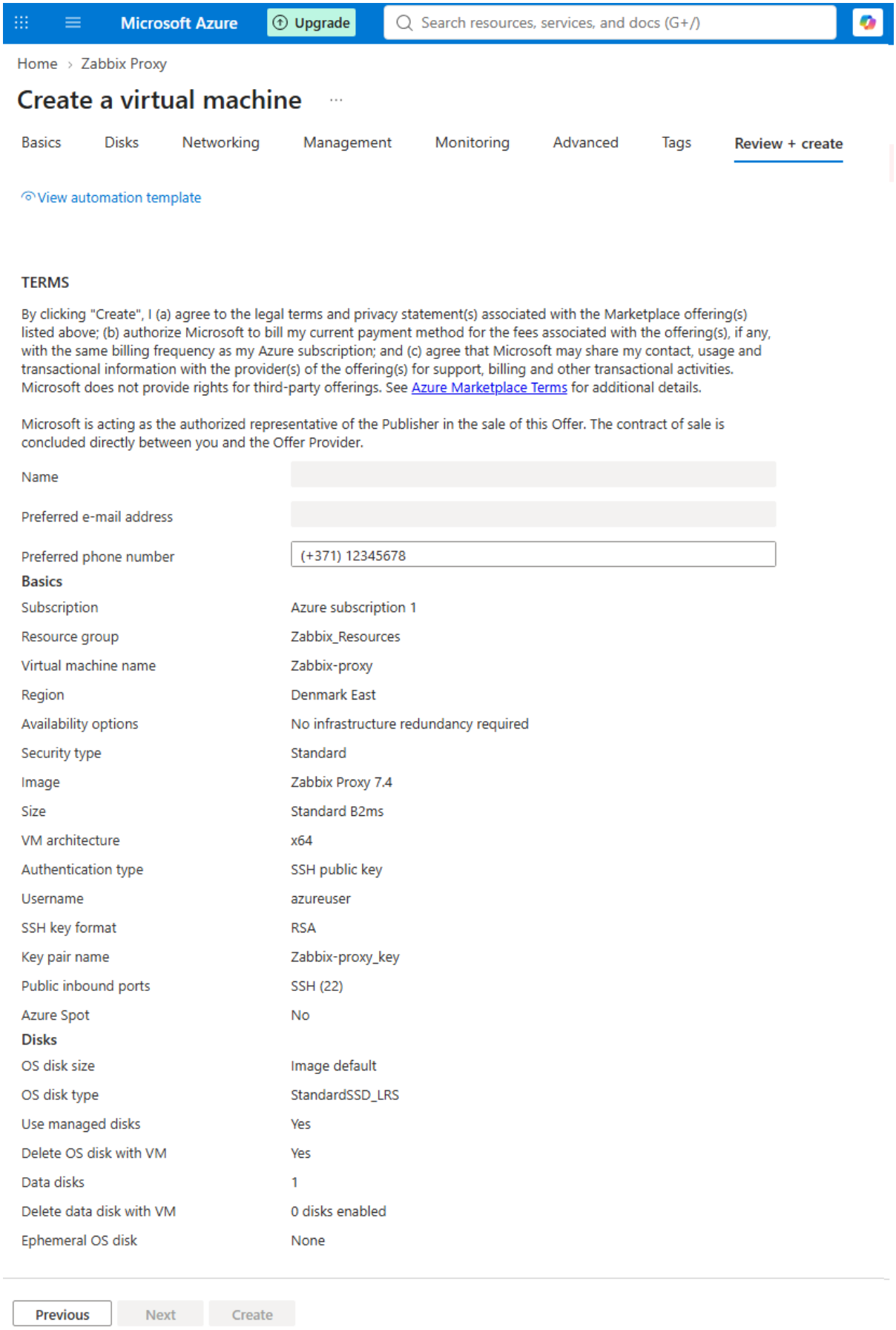
Aba *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Confirme os detalhes da VM e crie a VM.



Você será redirecionado para a página de status da implantação da VM. Aguarde até que a implantação seja concluída.

Quando a implantação for concluída, abra o recurso e anote seu endereço IP público. Conecte-se ao Zabbix proxy usando SSH e as credenciais de administrador criadas durante a implantação da VM para configurar o proxy.

```
********************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

********************************************************************************
[azureuser@Zabbix-proxy ~]$
```

O arquivo de configuração do Zabbix proxy pode ser encontrado em `/etc/zabbix/zabbix_proxy.conf`.

A configuração inicial do proxy consiste em modificar os parâmetros `ProxyMode`, `Server` e `Hostname` no arquivo de configuração do Zabbix proxy.

O modo do proxy pode ser alterado editando o parâmetro `ProxyMode`. Defina `ProxyMode=0` para o modo ativo ou `ProxyMode=1` para o modo passivo.

O parâmetro `Server` deve conter o endereço IP do Zabbix server.

O parâmetro `Hostname` deve conter o nome do host do Zabbix proxy. O nome do host definido aqui deve corresponder ao nome do proxy fornecido durante a configuração do proxy no Zabbix frontend.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Abra o Zabbix frontend e acesse *Administration* > *Proxies*. Clique no botão *Create proxy*.

Insira o mesmo nome do proxy usado como valor do parâmetro *Hostname* no arquivo de configuração do proxy.

Selecione o modo do proxy e forneça o endereço IP do proxy (opcional para proxies ativos).

Clique no botão *Create proxy* para criar um proxy.

Localização do banco de dados do proxy

O banco de dados SQLite do proxy pode ser encontrado em `/tmp/zabbix_proxy.sqlite3` e é especificado pelo parâmetro `DBName` no arquivo de configuração do proxy do Zabbix.

Guia de implantação no Google Cloud

Visão geral

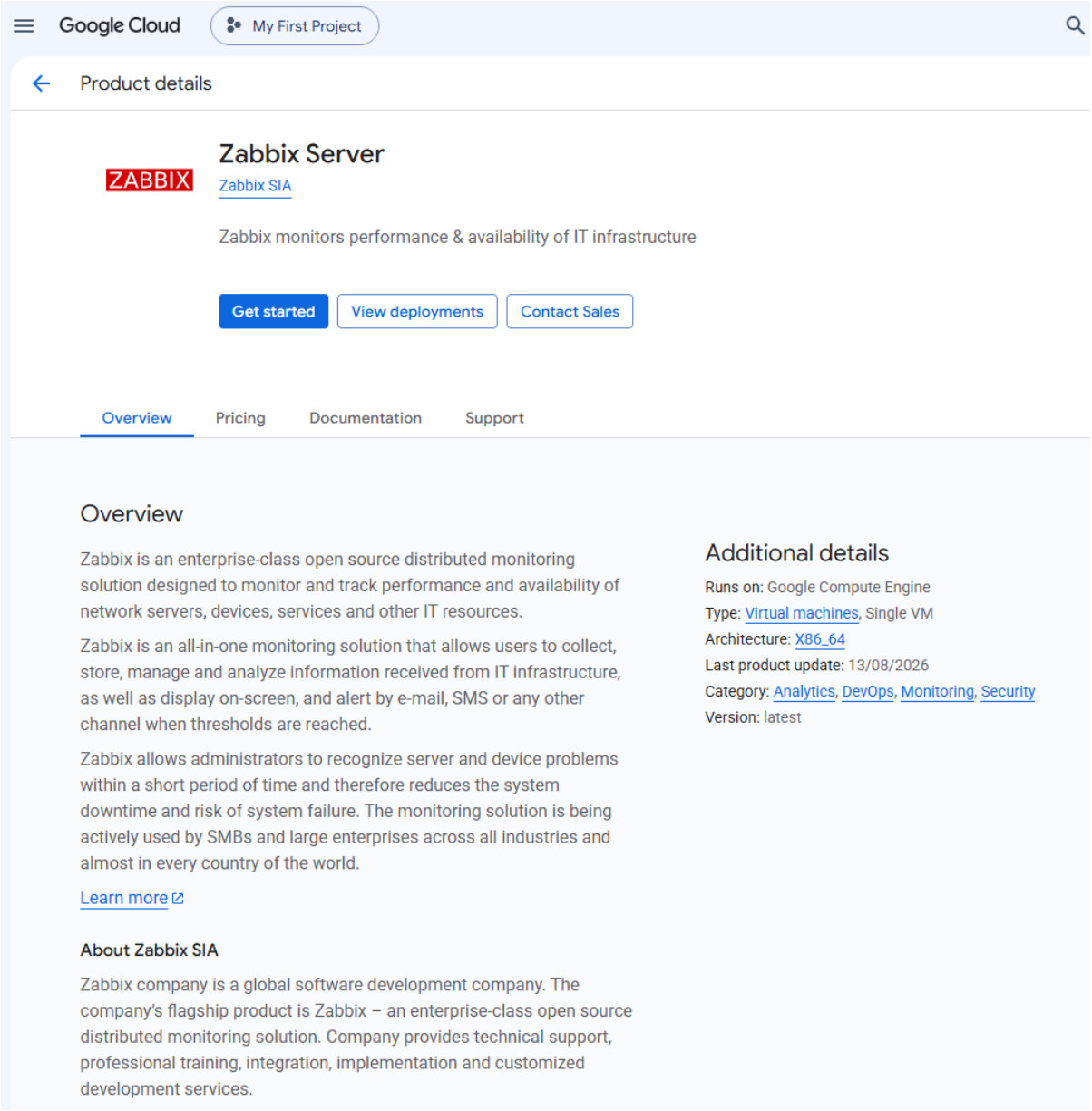
Esta página descreve como implantar o Zabbix usando o Google Cloud Marketplace.

Attention:
Várias versões do Zabbix estão disponíveis para implantação na plataforma. Leia a documentação referente à mesma versão do Zabbix que você implantou. A documentação de diferentes versões do Zabbix pode ser selecionada no menu suspenso de versões.

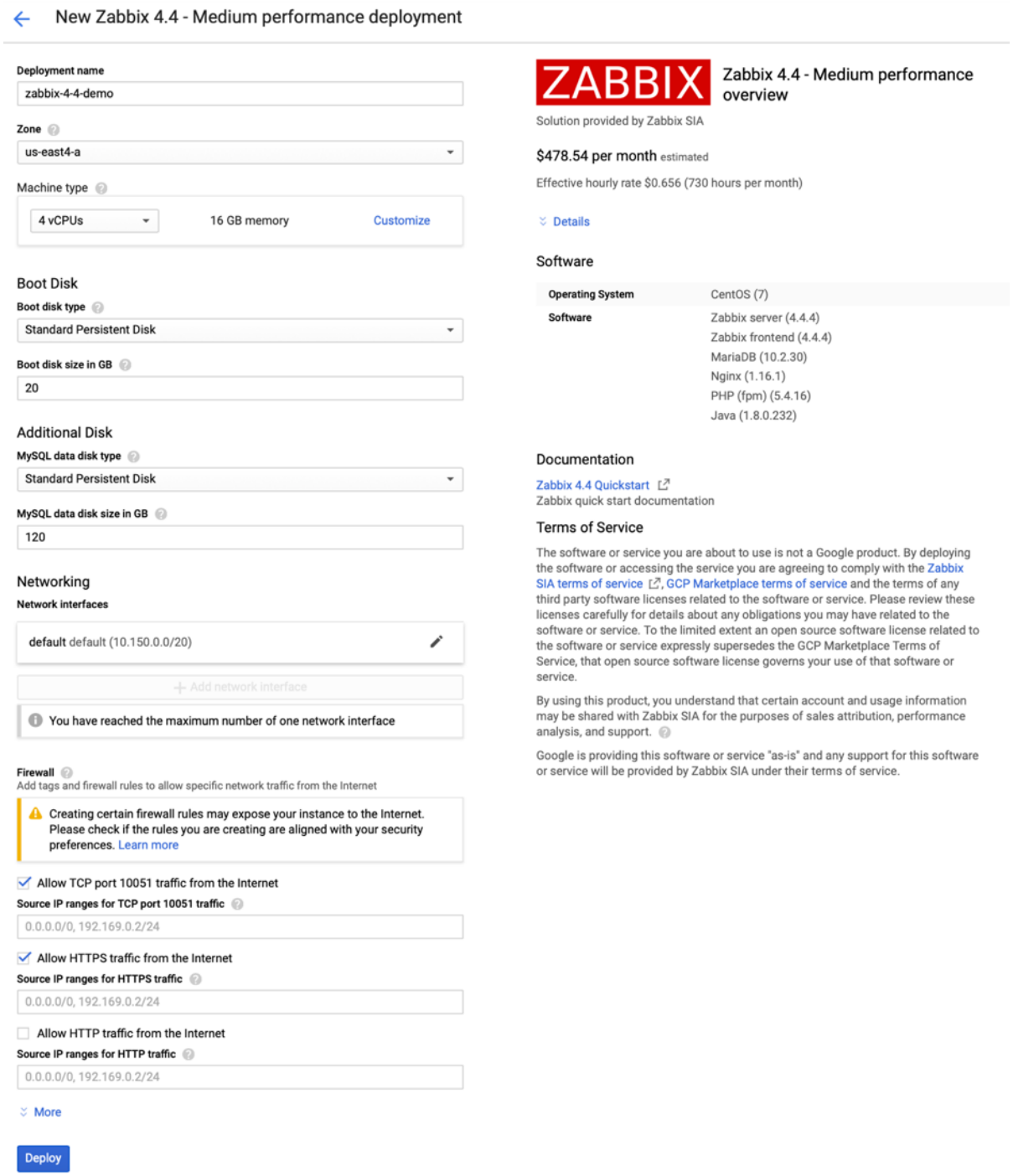
Implantando o appliance

Primeiro, implante o appliance do Zabbix. Isso é muito fácil com o Google Cloud Marketplace. Basta acessar o produto Zabbix Server no marketplace.

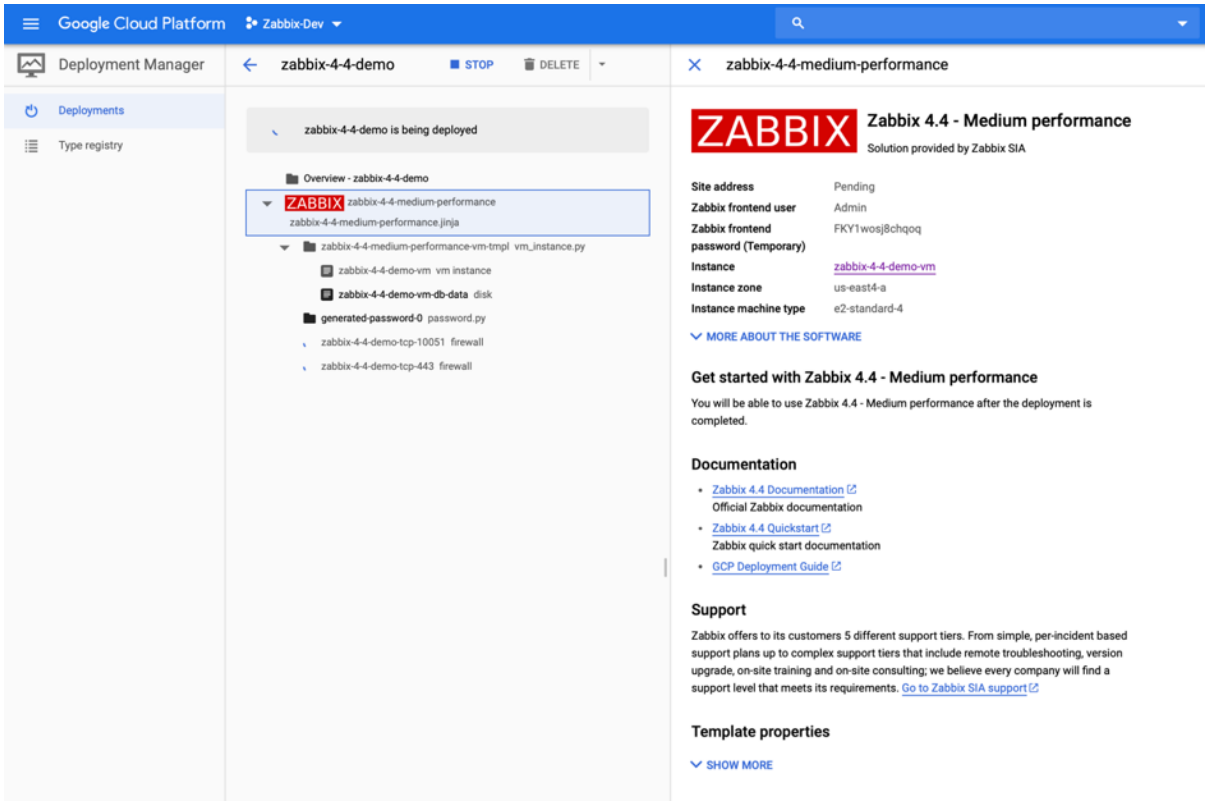
Clique em *Get started*.



Você pode manter as configurações padrão ou personalizá-las. Quando terminar, clique em *Deploy*.



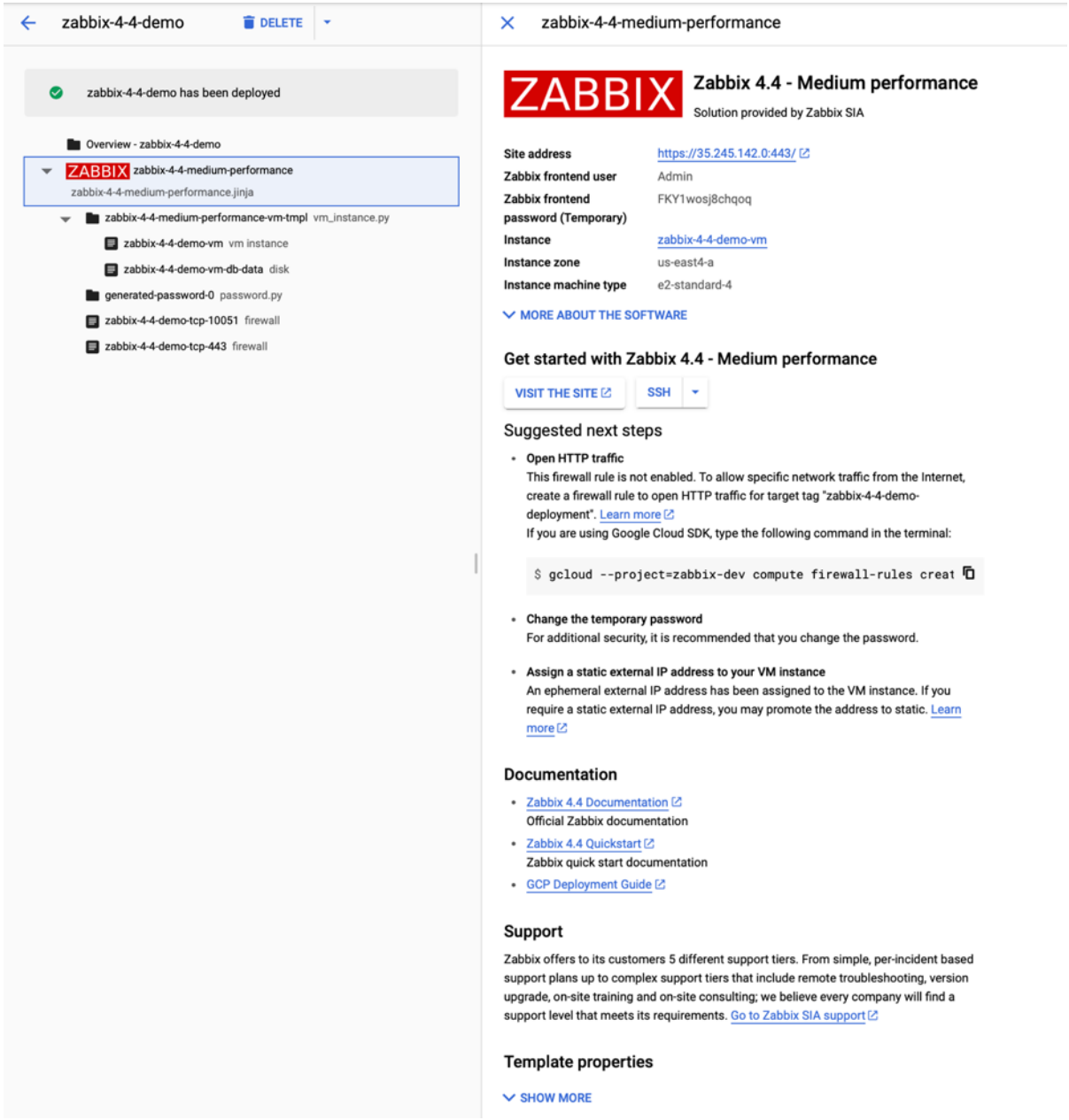
A implantação está em andamento. Aguarde até que ela seja concluída.



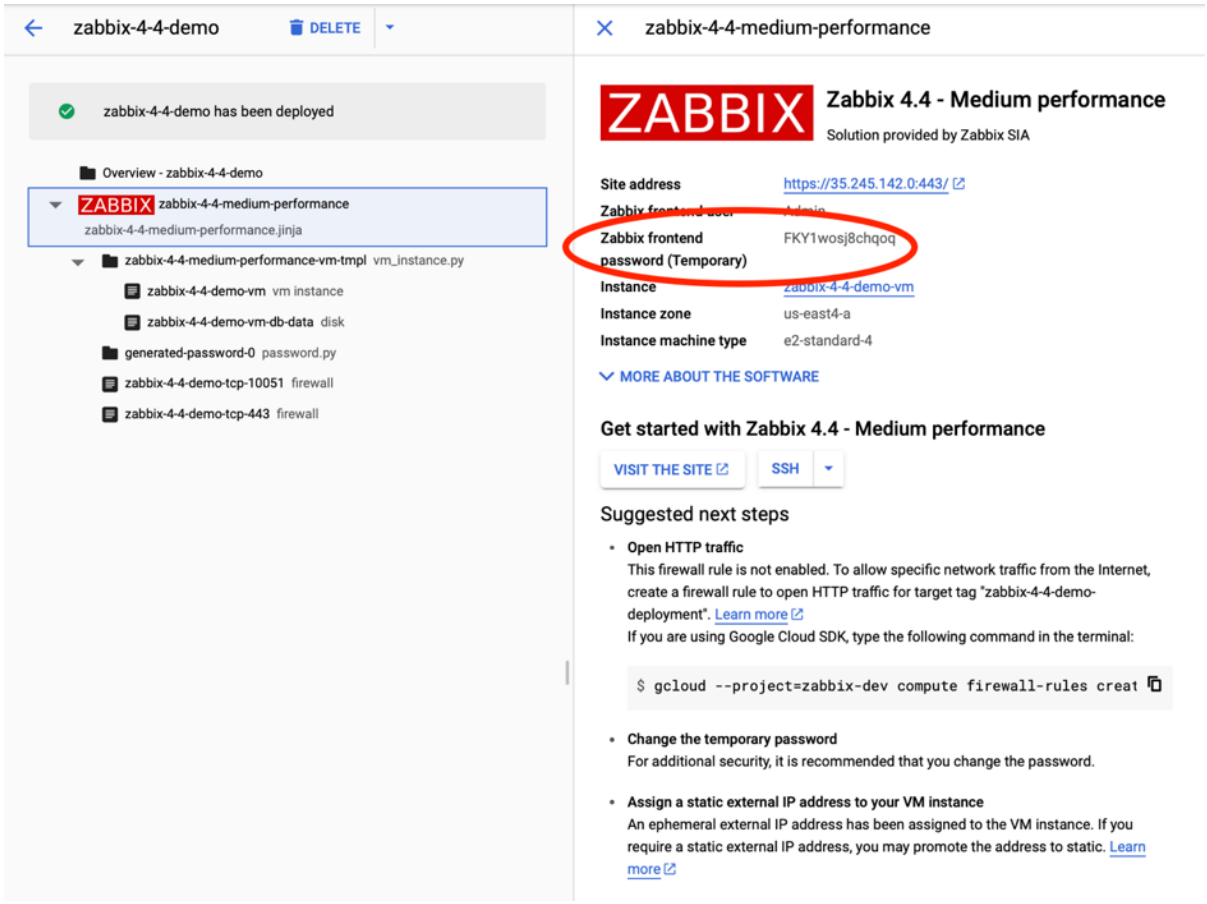
Pronto! A implantação foi iniciada.

Acessando o appliance

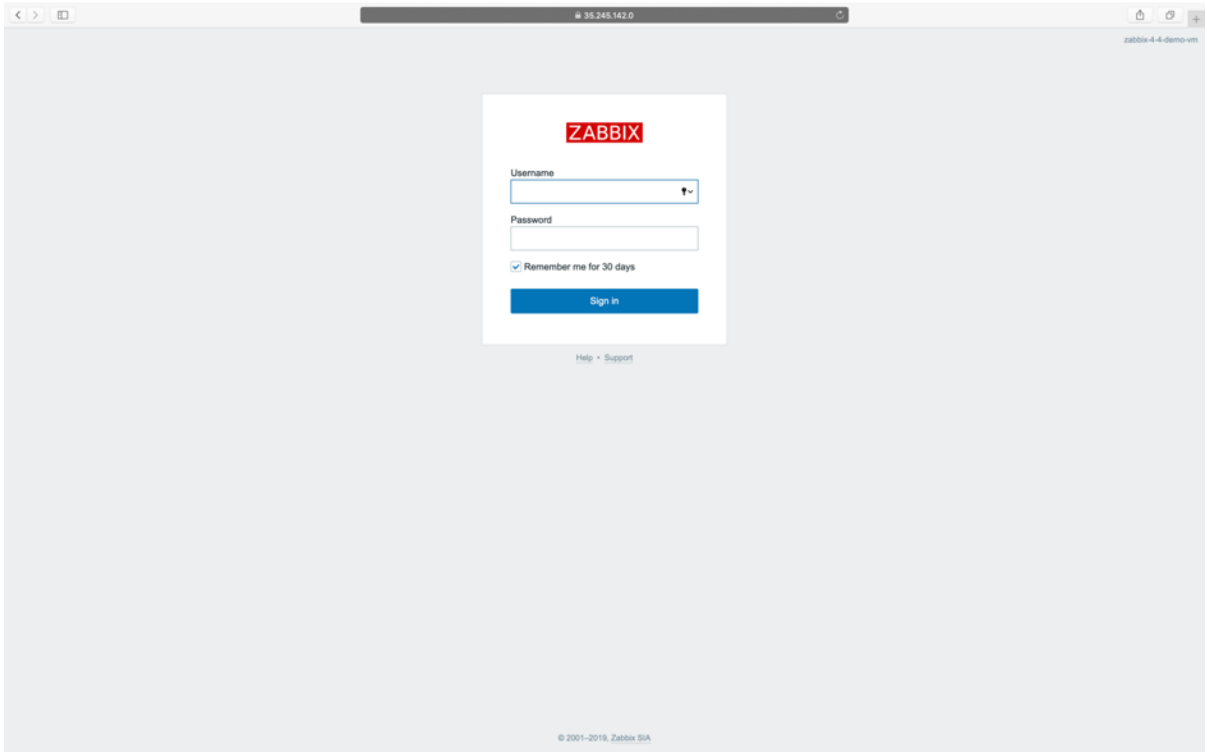
Quando a implantação for concluída, você deverá ver uma mensagem indicando que o appliance está sendo implantado.

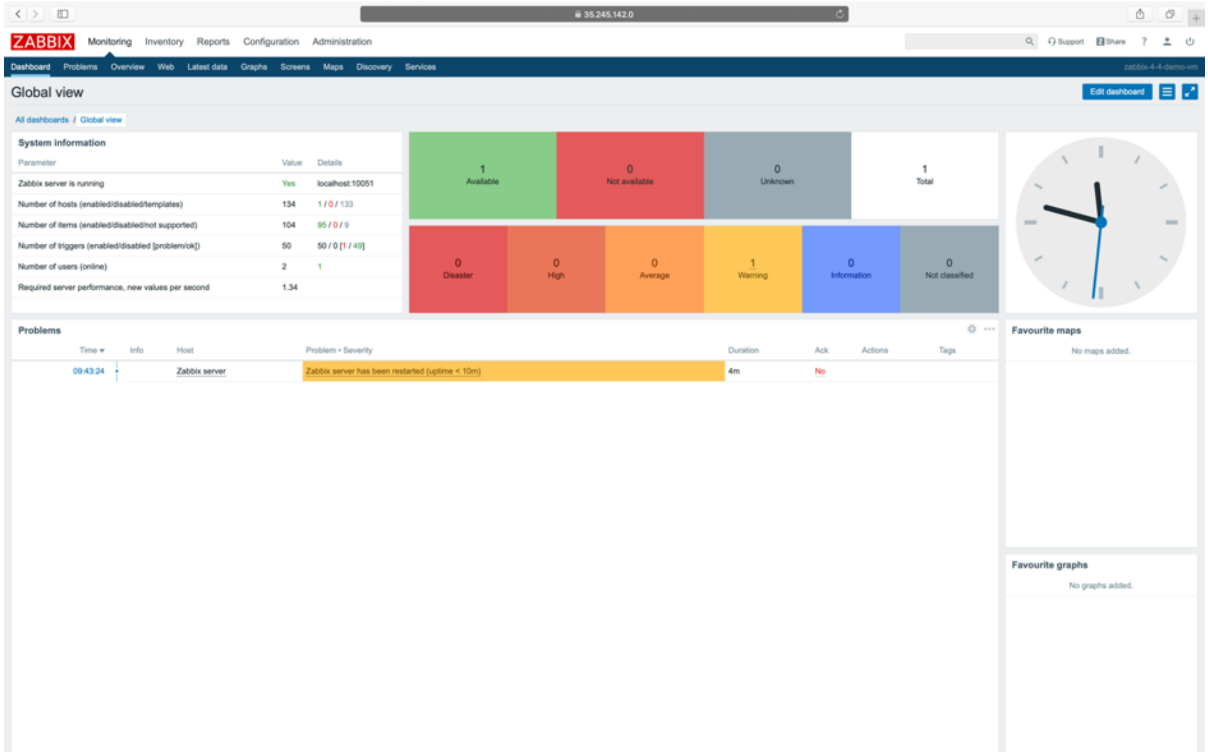


Abra um navegador da web em *Site Address* para visualizar o frontend do Zabbix. Antes disso, obtenha a senha do usuário *Admin* do Zabbix para fazer login no seu appliance do Zabbix.

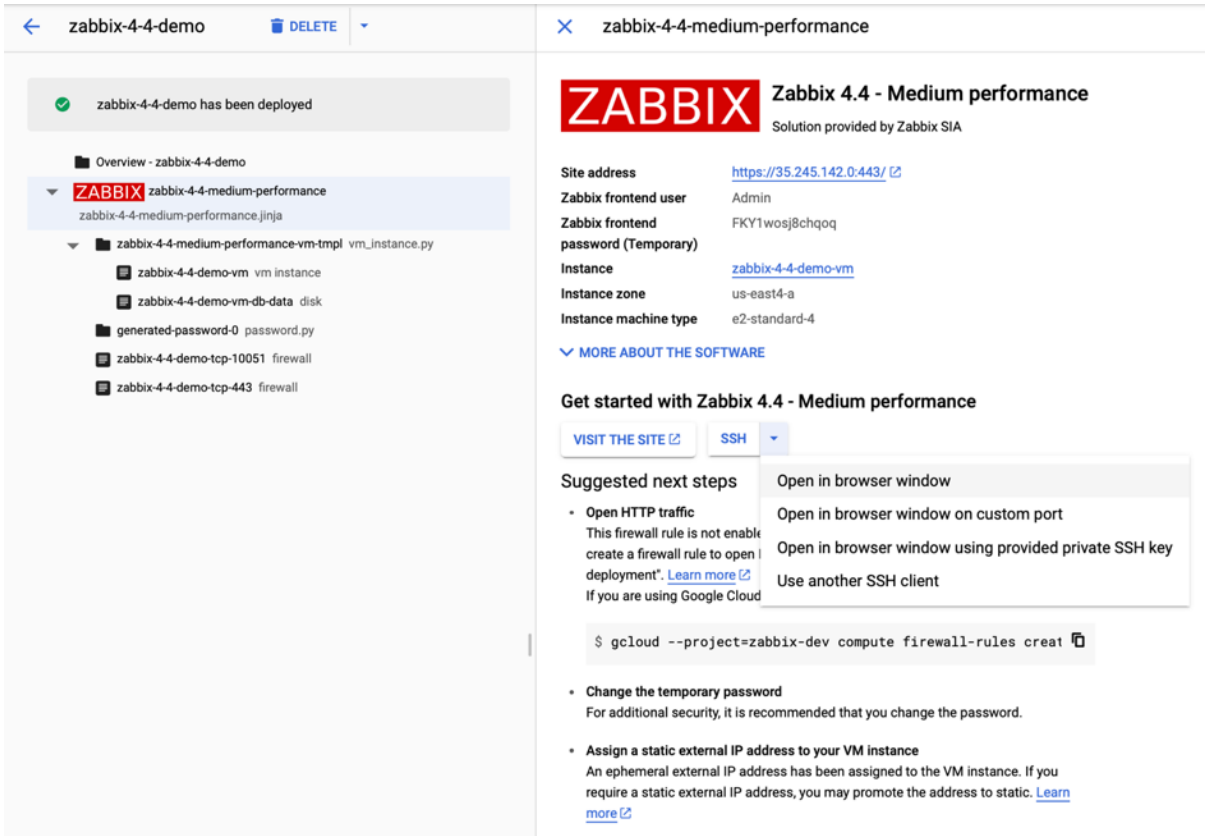


Pronto! Agora você tem um appliance do Zabbix.





Você também pode fazer login na instância para interagir com o banco de dados. Para isso, volte ao console do Google e siga a seta vermelha, conforme mostrado abaixo, para iniciar uma sessão SSH usando a opção *Open in browser window*.



O MySQL está disponível sem uma senha de `root`. A senha está armazenada no arquivo de configuração `/root/.my.cnf`. Por padrão, o appliance do Zabbix usa o fuso horário UTC. Se precisar alterá-lo, edite a diretiva de configuração PHP `date.timezone` em `/etc/php.d/99-zabbix.ini` e reinicie o processo php-fpm: `systemctl restart php-fpm`.

Atualização

Antes de atualizar, é altamente recomendável consultar todas as notas de atualização relevantes.

Também será útil saber quais componentes você deve atualizar.

Visão geral

Instruções de atualização passo a passo estão disponíveis para:

- Red Hat Enterprise Linux (usando pacotes)
- Debian/Ubuntu (usando pacotes)
- Atualização usando contêineres
- Atualização usando código-fonte

Instruções relacionadas:

- Para servers em um cluster de alta disponibilidade (HA), consulte Atualizando o cluster HA
- Para o banco de dados TimescaleDB, consulte Atualizando o esquema do TimescaleDB

Componentes a atualizar

A atualização dos Zabbix agents é recomendada, mas não obrigatória.

A atualização dos Zabbix proxies é altamente recomendada.
O Zabbix server oferece suporte completo a proxies que tenham a mesma versão principal do server.
O Zabbix server também oferece suporte a proxies que **não sejam mais antigos** que a versão LTS anterior do Zabbix server, mas com funcionalidade limitada (coleta de dados, execução de comandos remotos, verificações imediatas de valores de itens).
A atualização da configuração também é desabilitada, e os proxies desatualizados funcionarão somente com a configuração antiga.

> **Attention:**
> Proxies mais antigos que a versão LTS anterior do Zabbix server ou mais recentes que a versão principal do Zabbix server não são compatíveis. O Zabbix server ignorará os dados de proxies não compatíveis, e toda a comunicação com o Zabbix server falhará com um aviso. Para obter mais informações, consulte Compatibilidade de versões.

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 do Zabbix um após o outro. Durante o tempo de inatividade do server, os proxies em execução continuarão coletando dados. Quando o server estiver em funcionamento novamente, os proxies desatualizados enviarão os dados para o server mais recente (a configuração do proxy, no entanto, não será atualizada) e permanecerão parcialmente funcionais. Todas as notificações de problemas ocorridos durante o tempo de inatividade do Zabbix server serão geradas somente após a inicialização do server atualizado.

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

Observe que, se o proxy do Zabbix usar SQLite3 e detectar, durante a inicialização, que a versão do arquivo de banco de dados existente é mais antiga que a necessária, ele **excluirá o arquivo de banco de dados automaticamente** e criará um novo. Portanto, os dados históricos armazenados no arquivo do banco de dados SQLite serão perdidos. Se a versão do proxy do Zabbix for mais antiga que a versão do arquivo de banco de dados, o Zabbix registrará um erro e será encerrado.

Dependendo do tamanho do banco de dados, a atualização do banco de dados para a versão 8.0 poderá levar bastante tempo.

Notas de atualização

A atualização direta para o Zabbix 8.0.x é compatível a partir das versões **2.0**.x.
Para atualizar a partir de versões anteriores, consulte a documentação do Zabbix para a versão 2.0 e anteriores.

> **Note:**
> Esteja ciente de que, após a atualização, algumas integrações de software de terceiros no Zabbix podem ser afetadas caso o software externo não seja compatível com a versão atualizada do Zabbix.

As seguintes notas de atualização estão disponíveis:

Atualização de	Leia as notas completas de atualização	Alterações mais importantes entre as versões
7.4.x	Para: Zabbix 8.0	Os Zabbix agents anteriores à versão 1.6 não são mais compatíveis. As versões mínimas exigidas dos bancos de dados foram aumentadas. A versão mínima exigida do PHP foi aumentada de 8.0.0 para 8.2.0. O caractere % foi adicionado à lista `UnsafeUserParameters` do Zabbix agent e do Zabbix agent 2.

Atualização de	Leia as notas completas de atualização	Alterações mais importantes entre as versões
7.2.x	Para: Zabbix 7.4 Zabbix 8.0	O suporte à biblioteca PCRE foi removido.
7.0.x	Para: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	O suporte ao Oracle DB foi removido.
6.4.x	Para: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	A versão mínima exigida do PHP foi aumentada de 7.4.0 para 8.0.0. Pollers assíncronos para verificações de agent, HTTP agent, SNMP walk[OID] e SNMP get[OID]. Tabela de banco de dados separada para proxies. A localização padrão do arquivo de configuração do agent do Windows foi alterada. Oracle DB obsoleto. O tipo de valor numérico antigo (float) foi removido.
6.2.x	Para: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	A versão mínima exigida do MySQL foi 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 remove automaticamente a versão antiga do banco de dados (com todo o histórico) e cria uma nova.
6.0.x LTS	Para: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	A versão mínima exigida do PHP foi aumentada de 7.2.5 para 7.4.0. O monitoramento de serviços foi significativamente reformulado. Triggers determinísticos precisam ser criados durante a atualização. Se o registro binário estiver habilitado para MySQL/MariaDB, isso exigirá privilégios de superusuário ou a definição da variável/parâmetro de configuração *log_bin_trust_function_creators = 1*. Consulte Scripts de criação do banco de dados para obter instruções sobre 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	As versões mínimas exigidas dos bancos de dados foram aumentadas. O server/proxy não será iniciado se o banco de dados estiver desatualizado. Registros do log de auditoria foram perdidos devido à alteração na 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	As versões mínimas exigidas dos bancos de dados foram aumentadas. Os itens agregados foram 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	A versão mínima exigida do PHP foi aumentada de 7.2.0 para 7.2.5. O algoritmo de hash de senhas foi alterado de MD5 para bcrypt.

Atualização de	Leia as notas completas de atualização	Alterações mais importantes entre as versões
4.4.x	Para: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	O suporte ao IBM DB2 foi removido. A versão mínima exigida do PHP foi aumentada de 5.4.0 para 7.2.0. As versões mínimas exigidas dos bancos de dados foram aumentadas. O diretório dos arquivos PHP do Zabbix foi 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	Os tipos de mídia Jabber e Ez Texting foram removidos.
4.0.x LTS	Para: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Proxies mais antigos não poderão mais enviar 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	As bibliotecas ’libpthread’ e ’zlib’ agora são obrigatórias. O suporte ao protocolo de texto simples foi removido e o cabeçalho é obrigatório. Os Zabbix agents anteriores à versão 1.4 não são mais compatíveis. O parâmetro Server na configuração de proxy passivo agora é obrigatório.

Atualização de	Leia as notas completas de atualização	Alterações mais importantes entre as versões
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	O suporte ao SQLite como banco de dados de backend foi removido para o Zabbix server/frontend. Expressões regulares compatíveis com Perl (PCRE) são usadas em vez das expressões estendidas POSIX. As bibliotecas 'libpcre' e 'libevent' são obrigatórias para o Zabbix server. Foram adicionadas verificações do código de saída para parâmetros de usuário, comandos remotos e itens system.run[] sem a flag 'nowait', bem como para scripts executados pelo Zabbix server. O Zabbix Java gateway precisa ser atualizado para oferecer suporte à 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.
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	A versão mínima exigida do PHP foi aumentada de 5.3.0 para 5.4.0. O parâmetro LogFile do agent deve ser especificado.

Atualização de	Leia as notas completas de atualização	Alterações mais importantes entre as versões
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	O monitoramento distribuído baseado em nós foi 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	A versão mínima exigida do PHP foi aumentada de 5.1.6 para 5.3.0. É necessário um banco de dados MySQL que diferencie maiúsculas de minúsculas para o funcionamento correto do server; o conjunto de caracteres utf8 e a ordenação utf8_bin são necessários para que o Zabbix server funcione corretamente com o banco de dados MySQL. Consulte scripts de criação do banco de dados. A extensão PHP 'mysqli' é necessária em vez de 'mysql'.

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.

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, dos arquivos PHP e dos binários do Zabbix.

Para os arquivos de configuração, execute:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Para Apache:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Para Nginx:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

Para os arquivos PHP e os 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 atual do repositório Zabbix:

```
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 compatibilidade com os pacotes mais novos e incluir quaisquer patches de segurança ou correções de bugs recentes.

No **Debian 13**, 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+debian13_all.deb
```

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 no repositório Zabbix. Observe, porém, que os pacotes para essas versões podem não incluir todos os componentes do Zabbix e, para atualizar esses componentes por meio de pacotes, considere atualizar seu sistema operacional. Para ver a lista de componentes incluídos, consulte Pacotes Zabbix.

No **Ubuntu 26.04**, 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+ubuntu26.04_all.deb
```

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 no repositório Zabbix. Observe, porém, que os pacotes para essas versões podem não incluir todos os componentes do Zabbix e, para atualizar esses componentes por meio de pacotes, considere atualizar seu sistema operacional. Para ver a 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 os 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 resultar em um erro. Para obter mais informações, consulte *Problemas conhecidos*.

Você poderá ver uma solicitação 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] ?
```

Insira a opção que melhor se aplica à sua situação. Por exemplo, insira `D` para comparar a configuração atual com a nova e, em seguida, decida se deseja instalar a versão do mantenedor do pacote (`Y` ou `I`).

Em seguida, atualize o frontend do Zabbix e reinicie o web server.

Para o Apache, execute:

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

Para o Nginx, execute:

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Para o Nginx, você poderá ver uma solicitação sobre o arquivo de configuração `nginx.conf`. Insira a opção que melhor se aplica à sua situação:

```
Configuration file '/etc/zabbix/nginx.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** nginx.conf (Y/I/N/O/D/Z) [default=N] ?
```

6 Revisar os parâmetros de configuração do componente

Revise as notas de atualização relevantes para verificar se alguma alteração nos parâmetros de configuração é necessária.

Para novos parâmetros opcionais, consulte a página Novidades.

Se estiver usando Nginx como servidor web e o arquivo de configuração `/etc/zabbix/nginx.conf` foi reinstalado durante a atualização, revise o arquivo e atualize as diretivas `listen` e `server_name` para corresponder à configuração do seu frontend do Zabbix.

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.

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 atual do repositório para a versão mais recente para garantir compatibilidade com os pacotes mais novos e incluir correções de segurança ou de 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 do 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 dos componentes incluídos, consulte pacotes do 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` baixará metadados atualizados dos repositórios, já que os metadados antigos foram removidos.

Veja também: Problemas conhecidos para atualizar o pacote de configuração do repositório no RHEL.

5 Atualizar componentes do Zabbix

Para atualizar os componentes do Zabbix, execute:

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-selinux-policy 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.
- Se você tiver outros pacotes do Zabbix instalados, considere atualizá-los também, incluindo-os no comando.

Attention:
Atualizar o Zabbix agent 2 com o comando `dnf install zabbix-agent2` pode causar um erro. Para mais informações, consulte *Known issues*.

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 do componente

Revise as notas de atualização relevantes para verificar se alguma alteração nos parâmetros de configuração é necessária.

Para novos parâmetros opcionais, consulte a página Novidades.

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.

Atualização a partir de containers

Visão geral

Esta seção descreve as etapas necessárias para uma atualização bem-sucedida para os contêineres do Zabbix **8.0**.x.

Há conjuntos separados de instruções disponíveis para:

- Atualização da imagem dos componentes do Zabbix
- Atualização do arquivo 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 têm as permissões necessárias no banco de dados para fins de atualização.

 Para atualizações a partir do Zabbix 6.0 ou anterior, será necessário criar triggers determinísticos durante a atualização. Se o log binário estiver habilitado para MySQL/MariaDB, isso exigirá privilégios de superusuário ou a definição da variável/parâmetro de configuração `log_bin_trust_function_creators = 1`. Consulte Scripts de criação do banco de dados para obter instruções sobre como definir a variável.

 Observe que, se executada a partir de um console, a variável será definida apenas temporariamente e será removida quando um Docker for reiniciado. Nesse caso, mantenha seu serviço SQL em execução, interrompa 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 do banco de dados para a versão 8.0 pode levar muito tempo.

Atualização da imagem do componente Zabbix

Estas instruções atualizam suas imagens Docker usadas como base para os contêineres do seu componente Zabbix.

Os passos abaixo usam um proxy Zabbix como exemplo. Substitua `zabbix-proxy-sqlite3` pelo nome do contêiner do seu componente.

1. Verifique todos os contêineres para identificar seus nomes e as versões atuais das imagens:

```
docker ps -a

#### CONTAINER ID   IMAGE                                              (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest      (...) zabbix-proxy-sqlite3
```

2. Baixe a nova versão da imagem:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Pare o contêiner do componente Zabbix:

```
docker stop zabbix-proxy-sqlite3
```

4. Remova o contêiner do componente Zabbix:

Attention:
Certifique-se de ter anotado todas as flags de configuração e os pontos de montagem de volume usados no comando `docker run` original antes de prosseguir, pois eles serão necessários para recriar o contêiner.

```
docker rm zabbix-proxy-sqlite3
```

5. Inicie um novo contêiner usando a imagem atualizada:

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

Note:
Para mais informações sobre como executar contêineres Zabbix, consulte Docker (implantação manual).

6. Verifique se a atualização foi bem-sucedida conferindo os logs do contêiner:

```
docker logs -f zabbix-proxy-sqlite3
```

Atualização do arquivo Docker Compose

Estas instruções atualizam os arquivos Docker Compose usados para implantar os componentes do Zabbix. Siga estas instruções se você instalou o Zabbix usando Docker Compose.

1. Verifique todos os containers para identificar seus nomes e as versões atuais das imagens:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest    (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest    (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                               (...) zabbix-docker-mysql-server-1
```

2. Navegue até o repositório do Zabbix Docker que você clonou anteriormente, faça o pull das atualizações mais recentes e mude para a branch 8.0:

Attention:
Se você modificou `.env`, `compose.yaml` ou outros arquivos no seu repositório local, um `git pull` pode sobrescrevê-los. Faça backup de quaisquer alterações locais antes de executar esta etapa.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Implante os componentes do Zabbix usando o arquivo Docker Compose atualizado:

```
#### Com MySQL como banco de dados:
docker compose -f ./compose.yaml up -d

#### Com PostgreSQL como banco de dados:
docker compose -f ./compose_pgsql.yaml up -d
```

Note:
Depois que todos os containers forem recriados e iniciados, pode levar de 1 a 3 minutos até que a coleta de dados seja retomada. Para mais informações sobre a execução de containers do Zabbix, consulte Docker Compose.

4. Verifique se a atualização foi bem-sucedida conferindo os logs do container (substitua `zabbix-docker-zabbix-server-1` pelo nome do container do seu componente):

```
docker logs -f zabbix-docker-zabbix-server-1
```

Atualização a partir do código-fonte

Visão geral

Esta seção apresenta as etapas necessárias para realizar uma atualização bem-sucedida do Zabbix **7.4**.x para o Zabbix **8.0**.x usando fontes oficiais do Zabbix.

Warning:
Antes da atualização, leia as **notas de atualização** relevantes!

Você também pode consultar os requisitos da versão 8.0.

Note:
Pode ser útil executar duas sessões SSH em paralelo durante a atualização: execute as etapas de atualização em uma e monitore os logs do server/proxy na outra. Por exemplo, execute `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` na segunda sessão SSH para visualizar as entradas mais recentes do arquivo de log e possíveis erros em tempo real. Isso pode ser fundamental para instâncias em produção.

Processo de atualização do server

1 Parar o server

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

2 Faça backup do banco de dados existente do Zabbix

Esta é uma etapa muito importante. Certifique-se de ter um backup do seu banco de dados. Ele será útil se o procedimento de atualização falhar (falta de espaço em disco, desligamento, qualquer problema inesperado).

3 Fazer backup dos arquivos de configuração, arquivos PHP e binários do Zabbix

Faça uma cópia de backup dos binários do Zabbix, dos arquivos de configuração e do diretório de arquivos PHP.

4 Instalar novos binários do server

Use estas instruções para compilar o server do Zabbix a partir do código-fonte.

5 Revisar os parâmetros de configuração do server

Certifique-se de revisar as Notas de atualização para verificar se há necessidade de alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página Novidades.

6 Iniciar novos binários do Zabbix

Inicie os novos binários. Verifique os arquivos de log para confirmar se os binários foram iniciados com sucesso.

O Zabbix server atualizará o banco de dados automaticamente. Durante a inicialização, o Zabbix server informa as versões atuais (obrigatória e opcional) e as versões necessárias do banco de dados. Se a versão obrigatória atual for anterior à versão necessária, o Zabbix server executará automaticamente os patches de atualização necessários do banco de dados. O início e o nível de progresso (porcentagem) da atualização do banco de dados são gravados no arquivo de log do Zabbix server. Quando a atualização for concluída, a mensagem "database upgrade fully completed" será gravada 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 versão necessária. O Zabbix server será iniciado somente se a versão obrigatória atual do banco de dados corresponder à versão obrigatória necessária.

```
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 (criar tabela, excluir tabela, criar índice, excluir índice)
- Certifique-se de que haja espaço livre suficiente em disco.

7 Instalar a nova interface web do Zabbix

A versão mínima necessária do PHP é 8.2.0. Atualize, se necessário, e siga as instruções de instalação.

8 Limpar os cookies e o cache do navegador da web

Após a atualização, talvez seja necessário limpar os cookies e o cache do navegador da web 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 Instalar novos binários do proxy

Use estas instruções para compilar o proxy do Zabbix a partir do código-fonte.

4 Revisar os parâmetros de configuração do proxy

Certifique-se de revisar as Notas de atualização para verificar se há necessidade de alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página Novidades.

5 Iniciar o novo proxy do Zabbix

Inicie o novo proxy do Zabbix.
Verifique os arquivos de log para confirmar se o proxy foi iniciado com sucesso.

O proxy do Zabbix atualizará o banco de dados automaticamente.
A atualização do banco de dados ocorre de forma semelhante à inicialização do Zabbix server.

Processo de atualização do agent

> **Attention:**
> A atualização dos agents não é obrigatória. Você só precisa atualizar os agents se isso for necessário para acessar a nova funcionalidade.

O procedimento de atualização descrito nesta seção pode ser usado para atualizar tanto o Zabbix agent quanto o Zabbix agent 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 Instalar novos binários do agent

Use estas instruções para compilar o agent do Zabbix a partir do código-fonte.

Como alternativa, você pode baixar agents do Zabbix pré-compilados na página de download do Zabbix.

4 Revisar parâmetros de configuração do agent

Certifique-se de revisar as Notas de atualização para verificar se alguma alteração nos parâmetros de configuração é necessária.

Para novos parâmetros opcionais, consulte a página O que há de novo.

5 Iniciar o novo Zabbix agent

Inicie o novo Zabbix agent. Verifique os arquivos de log para confirmar se o agent foi iniciado com sucesso.

Atualização entre versões secundárias

Ao atualizar entre versões secundárias da série 8.0.x (por exemplo, da 8.0.1 para a 8.0.3), é necessário executar as mesmas ações para o server/proxy/agent que durante a atualização entre versões principais. A única diferença é que, ao atualizar entre versões secundárias, não são feitas alterações no banco de dados.

Problemas de atualização

Relacionado a escape Limite de caracteres excedido para parâmetros de função após a atualização

A escape adequada de barras invertidas foi adicionada nos parâmetros de string da função de histórico no Zabbix 7.0.

Como barras invertidas adicionais são adicionadas durante a atualização de versões do Zabbix anteriores à 7.0, isso leva a parâmetros mais longos, o que pode resultar em funções de trigger quebradas se o comprimento do parâmetro exceder o tamanho máximo de dados de 255 caracteres.

Para ilustrar quais problemas podem surgir durante a atualização do banco de dados, qual é o seu impacto, como reconhecê-los e corrigi-los, vamos tomar um exemplo de configuração com 3 items e 2 triggers.

Configuração

1. Versão 6.0 do Zabbix.
2. Arquivo `/tmp/ONE` que contém 114 barras invertidas com 'a' no final:

```
\\\
```

4. Os triggers foram atualizados da seguinte forma:

Trig_228:

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

Vazamento de memória e de descritores de arquivo com libcurl 8.21.0

Um problema conhecido no libcurl 8.21.0 pode causar vazamentos de memória e de descritores de arquivo em recursos do Zabbix que usam libcurl ao lidar com conexões protegidas por TLS.
 Uma solução alternativa para esse problema é fazer downgrade do libcurl para a versão 8.20.0 até que esteja disponível uma versão com a correção.

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 do Zabbix instalados a partir do EPEL:

```
dnf remove zabbix-server-mysql
```

2. Exclua os pacotes do Zabbix do EPEL adicionando a seguinte linha ao arquivo `/etc/yum.repos.d/epel.repo`:

```
[epel]
...
excludepkgs=zabbix*
```

3. Reinstale o pacote oficial do Zabbix server:

```
dnf install zabbix-server-mysql
```

Durante a instalação, os pacotes oficiais do Zabbix incluem a palavra `release` em sua string de versão (por exemplo, `7.0.0-release1.el8`), distinguindo-os dos pacotes do EPEL.

Pacotes Zabbix para RHEL em ambientes Red Hat UBI

Ao instalar o Zabbix a partir de pacotes do Red Hat Enterprise Linux em ambientes Red Hat Universal Base Image, 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.

Conexão TLS do banco de dados com MariaDB

A conexão TLS do banco de dados não é suportada com a opção `verify_ca` para o parâmetro DBTLSConnect se o MariaDB for usado.

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.

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.

IPMI — hosts não confiáveis podem causar a falha do OpenIPMI

Há um bug na biblioteca OpenIPMI usada pelo Zabbix para consultar dados de IPMI que pode ser acionado por respostas especialmente elaboradas de um dispositivo não confiável. Um dispositivo IPMI não confiável pode enviar dados elaborados que fazem a biblioteca OpenIPMI falhar, o que, por sua vez, pode fazer com que o processo do Zabbix server que realiza a consulta de IPMI seja encerrado.

Verificações SSH

- Algumas distribuições Linux como Debian, Ubuntu não suportam chaves privadas criptografadas (com senha) se a biblioteca libssh2 estiver instalada a partir de pacotes. Consulte ZBX-4850 para mais detalhes.

- Ao usar libssh 0.9.x em algumas distribuições Linux com OpenSSH 8, as verificações SSH podem ocasionalmente relatar "Cannot read data from SSH server". Isso é causado por um problema na libssh (relatório mais detalhado). Espera-se que o erro tenha sido corrigido na versão estável libssh 0.9.5. Veja também ZBX-17756 para detalhes.
- Usar o pipe "|" no script SSH pode levar ao erro "Cannot read data from SSH server". Neste caso, recomenda-se atualizar a versão da biblioteca libssh. Veja também ZBX-21337 para detalhes.

Verificações ODBC

- O driver MySQL unixODBC não deve ser usado com o Zabbix server ou Zabbix proxy compilado com a biblioteca conectora 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

  Consulte ZBX-7665 para mais informações e soluções alternativas disponíveis.
- Dados XML consultados do Microsoft SQL Server podem ser truncados de várias formas 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 falhe.
 Veja também: ZBX-18402, ZBX-20803.
- Se estiver usando o driver FreeTDS UnixODBC, você precisa prefixar uma instrução 'SET NOCOUNT ON' a uma consulta SQL (por exemplo, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Caso contrário, o item monitor de banco de dados no Zabbix não conseguirá recuperar as informações e apresentará o erro "A consulta SQL retornou resultado vazio".
 Consulte 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 localidades do frontend

Foi observado que as localidades do frontend podem alternar sem uma lógica aparente, ou seja, algumas páginas (ou partes das páginas) são exibidas em um idioma, enquanto outras páginas (ou partes das páginas) são exibidas em um idioma diferente. Normalmente, o problema pode ocorrer quando há vários usuários, alguns dos quais usam uma localidade, enquanto outros usam outra.

Uma solução alternativa conhecida para esse problema é desabilitar o multithreading no PHP e no Apache.

O problema está relacionado à forma como a configuração da localidade funciona no PHP: as informações de localidade são mantidas por processo, não por thread.
Assim, em um ambiente multithread, quando há vários projetos executados pelo mesmo processo do Apache, é possível que a localidade seja alterada em outra thread, modificando a forma como os dados podem ser processados na thread do Zabbix.

Para obter mais informações, consulte os relatórios de problemas relacionados:

- ZBX-10911 (Problema com a alternância das localidades do frontend)
- ZBX-16297 (Problema com o processamento de números em gráficos que usam a função `bcdiv` das funções BC Math)

Gráficos

Problemas com gráficos (clássicos)

Se você encontrar problemas com gráficos clássicos, recomenda-se atualizar a biblioteca GD (libgd) para a versão 2.3.3-13 ou superior, e o PHP para a versão 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 ou superior.

Horário de Verão

Mudanças no Horário de Verão (DST) resultam em irregularidades ao exibir os rótulos do eixo X (duplicação de datas, datas ausentes, etc).

Agregação por soma

Ao usar agregação por soma em um gráfico para um período inferior a uma hora, os gráficos exibem valores incorretos (multiplicados) quando os dados vêm de tendências.

Sobreposição de texto

Para alguns idiomas do frontend (por exemplo, japonês), fontes locais podem causar sobreposição de texto na legenda do gráfico. Para evitar isso, use a versão 2.3.0 (ou posterior) da extensão PHP GD.

Monitoramento de arquivos de log

Os items `log[]` e `logrt[]` relêem repetidamente o arquivo de log desde o início se o sistema de arquivos estiver 100% cheio e o arquivo de log estiver sendo anexado (veja ZBX-10884 para mais informações).

Consultas lentas do MySQL

O Zabbix server gera consultas `SELECT` lentas no caso de valores inexistentes para 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 Zabbix agent no Windows

Entradas DNS inexistentes em um parâmetro `Server` do arquivo de configuração do Zabbix agent (`zabbix_agentd.conf`) podem aumentar o tempo de resposta do Zabbix agent no Windows. Isso acontece porque o daemon de cache DNS do Windows não armazena em cache respostas negativas para endereços IPv4. No entanto, para endereços IPv6, as respostas negativas são armazenadas em cache, então uma possível solução alternativa para isso é desabilitar IPv4 no host.

Exportação/importação de YAML

Há alguns problemas conhecidos com exportação/importação de YAML:

- As mensagens de erro não são traduzíveis;
- JSON válido com extensão de arquivo .yaml às vezes não pode ser importado;
- Datas legíveis por humanos sem aspas são convertidas automaticamente em timestamps Unix.

Assistente de configuração no SUSE com NGINX e php-fpm

O assistente de configuração do frontend não consegue 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).

Há duas opções de contorno 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 migração 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 versões mais recentes, SLES (todas as versões), Debian 9 e versões mais recentes, Ubuntu 16.04 e versões mais recentes foram atualizados para usar PCRE2. Apesar de trazer muitos benefícios, a migração para PCRE2 pode fazer com que determinados padrões de regexp PCRE existentes se tornem inválidos ou se comportem de forma 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 hífen como um delimitador, criando um intervalo dentro de uma classe de caracteres.

Erro do widget Geomap

Os mapas no widget Geomap podem não carregar corretamente se você tiver atualizado de uma versão mais antiga do Zabbix com NGINX e não tiver mudado 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 PHP for Zabbix frontend*.

Como alternativa, você pode editar manualmente um arquivo de configuração do NGINX existente (normalmente, `/etc/zabbix/nginx.conf`). Para fazer isso, abra o arquivo e localize o seguinte bloco:

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

Em seguida, substitua esse 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, $\varnothing$, $\notin$). 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.

Perfilando uso excessivo de memória com tcmalloc

Se você suspeita que sua instalação do Zabbix está usando memória demais, você pode usar o recurso de perfilamento de memória do tcmalloc para investigar o consumo de memória do server/proxy do Zabbix.

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, o Zabbix proxy ou o Zabbix agent. A flag `-g` adiciona informações extras de depuração, enquanto `-O0` desativa otimizações, que podem interferir no perfilamento 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. Gere um dump do perfil enviando o sinal 5 para o processo de destino. Substitua 1234 pelo ID real do processo (PID):

```
kill -5 1234
```

4. Exiba 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 os relatórios de problema relacionados.
- GCC Option Summary sobre opções de compilação (`-std=gnu99`, `-g`, `-O0`, etc.).
- Gperftools Heap Profiler documentação sobre variáveis de ambiente para perfilamento do tcmalloc.

Replicação de Grupo do MySQL 8.0 no modo multi-primary

Ao usar a Replicação de Grupo do MySQL 8.0 no modo multi-primary, você pode encontrar um erro durante os commits 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 do problema relacionado.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" para o problema upstream.

Login e systemd

Recomendamos criar um usuário *zabbix* como usuário do sistema, ou seja, sem capacidade de fazer 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 fazer o daemon do Zabbix falhar ao sair da sessão. Nesse caso, você verá algo semelhante ao seguinte no log do Zabbix server:

```
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 Zabbix agent:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Isso acontece devido à configuração padrão do systemd `RemoveIPC=yes`, definida em `/etc/systemd/logind.conf`. Quando você sai do sistema, os semáforos criados anteriormente pelo Zabbix são removidos, o que causa a falha.

Um trecho da documentação do systemd:

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

Há 2 soluções para esse problema:

1. (recomendado) Pare de usar a conta *zabbix* para qualquer finalidade que não seja executar processos do Zabbix e crie uma conta dedicada para outras atividades.
2. (não recomendado) Defina `RemoveIPC=no` em `/etc/systemd/logind.conf` e reinicie o sistema. Observe que `RemoveIPC` é um parâmetro de todo o sistema; alterá-lo afetará o sistema inteiro.

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, considerando 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 compreender completamente seus requisitos antes de implantar a instalação do Zabbix no ambiente de produção.

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

[1] Para fins de dimensionamento, 1 métrica monitorada é representada aproximadamente pela fórmula 1 métrica = 1 item + 1 trigger + 1 gráfico. Esta é apenas uma regra de estimativa, não uma fórmula para calcular métricas monitoradas.
 [2] Exemplo com instâncias EC2 de uso geral da Amazon, usando a arquitetura ARM64 ou x86_64; um tipo de instância adequado, como otimizado para computação/memória/armazenamento, deve ser selecionado durante a avaliação e os testes da instalação do Zabbix, antes da instalação no ambiente de produção.

> **Note:**
> A configuração real depende muito do número de itens ativos e das taxas de atualização (consulte a seção tamanho do banco de dados desta página para obter detalhes). É altamente recomendável executar o banco de dados em um server 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

Quando indicado como obrigatório, o software/biblioteca necessário é estritamente indispensável.
Os componentes opcionais são necessários para oferecer suporte a alguma função específica.

Software	Status obrigatório	Versões com-patíveis	Comentários
MySQL/Percona	One of	8.4.0-9.7.X	Necessário se o MySQL (ou Percona) for usado como banco de dados de backend do Zabbix. O mecanismo InnoDB é obrigatório. Recomendamos usar a biblioteca C API (libmysqlclient) para compilar o server/proxy. Recomendamos enfaticamente usar apenas versões LTS em ambientes de produção. Versões identificadas como ”rolling release”, ”early access” ou semelhantes não são recomendadas.

Software	Status obrigatório	Versões com-patíveis	Comentários
MariaDB		10.11.00-12.3.X	O mecanismo InnoDB é obrigatório. Recomendamos usar a biblioteca MariaDB Connector/C para compilar o server/proxy. Consulte também: Possíveis deadlocks com MariaDB e Acesso a elementos da UI com MariaDB 10.5.1-10.5.9. Recomendamos enfaticamente usar apenas versões LTS em ambientes de produção. Versões identificadas como ”rolling release”, ”early access” ou semelhantes não são recomendadas.
PostgreSQL		15.0-18.X	Necessário se o PostgreSQL for usado como banco de dados de backend do Zabbix. Dependendo do tamanho da instalação, pode ser necessário aumentar a propriedade de configuração *work_mem* do PostgreSQL (o valor padrão é 4 MB), para que a quantidade de memória usada pelo banco de dados para uma operação específica seja suficiente e a execução da consulta não demore muito.
TimescaleDB para PostgreSQL		2.20.X-2.29.X	Necessário se o TimescaleDB for usado como extensão de banco de dados do PostgreSQL. Certifique-se de instalar a TimescaleDB Community Edition, que oferece suporte à compactação. Consulte a documentação do TimescaleDB para obter detalhes sobre a compatibilidade entre as versões do PostgreSQL e do TimescaleDB.
Elasticsearch		7.X	Necessário se o Elasticsearch for usado como banco de dados de backend do Zabbix. O Elasticsearch pode ser usado somente para armazenar dados de histórico e é compatível apenas com Zabbix servers. O suporte ao Elasticsearch é atualmente experimental. Consulte também o software necessário para server/proxy.
ClickHouse		26.3.0.0-26.9.X.X	Necessário se o ClickHouse for usado como banco de dados de backend do Zabbix para armazenar dados de histórico (compatível apenas com Zabbix servers) ou como uma fonte de dados de telemetria. Para usar o ClickHouse como fonte de dados de telemetria, é necessário o ClickHouse 26.8.17.4 ou posterior. O horário do sistema no Zabbix server e no server do ClickHouse também deve estar sincronizado; caso contrário, as consultas de telemetria podem não encontrar dados ou retornar os mesmos dados duas vezes.
SQLite	Opcional	3.3.5-3.53.X	O SQLite é compatível somente com Zabbix proxies. Necessário se o SQLite for usado como banco de dados do Zabbix proxy.
smartmontools		7.1 ou posterior	Necessário para o Zabbix agent 2.
who			Necessário para o plugin de contagem de usuários.
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 dos 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 obter a melhor experiência, recomendamos usar bancos de dados instalados a partir dos repositórios oficiais dos desenvolvedores do banco de dados.

frontend

A largura mínima de tela compatível com o frontend do Zabbix é de 1200px.

Quando indicado como obrigatório, o software/biblioteca necessário é estritamente indispensável.
Os itens opcionais são necessários para oferecer suporte a alguma função específica.

Software	Status obrigatório	Versões com-patíveis	Comentários
PHP	Sim	8.2.0 - 8.5.X	
Apache	Um dos	2.4 ou posterior	
Nginx		1.20 ou posterior	
MySQL	Um dos	Consulte Software externo de terceiros e de suporte	
PostgreSQL			
Extensões do PHP			
mysqli	Sim		Obrigatório se o MySQL for usado como banco de dados de backend do Zabbix.
pgsql			Obrigatório se o PostgreSQL for usado como banco de dados de backend do Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); obrigatório para oferecer suporte a scripts de usuário.
gd		2.0.28 ou posterior	php-gd (se fornecido como um pacote separado pelo distribuidor); a extensão GD do PHP deve oferecer suporte a 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 possível sobreposição de texto nos gráficos em alguns idiomas do frontend.
libxml		2.6.15 ou posterior	php-xml (se fornecido como um pacote separado pelo distribuidor)
xmlwriter			php-xmlwriter (se fornecido como um pacote separado pelo distribuidor)
xmlreader			php-xmlreader (se fornecido como um pacote separado pelo distribuidor)
ctype			php-ctype (*--enable-ctype*)
session			php-session (se fornecido como um pacote separado pelo distribuidor)
ldap	Não		php-ldap; obrigatório para autenticação LDAP.
openssl			php-openssl; obrigatório para autenticação SAML.
gettext			php-gettext (*--with-gettext*); obrigatório para traduções.
cURL		7.19.4 ou posterior	php-curl; obrigatório para o Duo Universal Prompt MFA, autenticação SMTP e para testar a fonte de dados de telemetria.

Bibliotecas de frontend de terceiros fornecidas com o Zabbix:

Biblioteca	Status obrigatório	Versão fornecida	Comentários
jQuery JavaScript Library	Sim	3.6.0	Biblioteca JavaScript que simplifica o processo de desenvolvimento entre navegadores.
jQuery UI		1.12.1	Um conjunto de interações de interface do usuário, efeitos, widgets e temas desenvolvido sobre o jQuery.
SAML PHP Toolkit		4.3.1	Um toolkit PHP que adiciona suporte à autenticação SAML 2.0 para permitir o login no Zabbix.

Biblioteca	Status obrigatório	Versão fornecida	Comentários
Duo Universal PHP library		1.1.2	Uma biblioteca PHP que adiciona suporte à autenticação multifator Duo Universal Prompt ao Zabbix.
Symfony Yaml Component		5.1.0	Adiciona suporte à exportação e importação de elementos de configuração do Zabbix no formato YAML.

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

Attention:
Para fontes diferentes da DejaVu padrão, a função PHP imagerotate pode ser necessária. Se estiver ausente, essas fontes poderão ser renderizadas incorretamente quando um gráfico for exibido. Essa função só está disponível quando o PHP é compilado com o GD integrado, o que não ocorre 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 testes unitários PHP para testar o frontend do Zabbix.
SASS		3.4.22	Uma linguagem de script de pré-processamento 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 indicado como obrigatório, o software/biblioteca necessário é estritamente indispensável.
Os componentes opcionais são necessários para oferecer suporte a alguma função específica.

Requisito	Status obrigatório	Descrição
libpcre2	Sim	A biblioteca PCRE2 é necessária para oferecer suporte a expressões regulares compatíveis com Perl (PCRE). Há suporte à versão 10.x do PCRE2.
libevent		Necessário para comunicação entre processos. Versão 2.0.10 ou superior.
libevent-pthreads		Necessário para comunicação entre processos.
libpthread		Necessário para oferecer suporte a mutex e bloqueio de leitura e gravação (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 oferecer suporte à compactação.
libm		Biblioteca matemática. Necessária apenas pelo Zabbix server.
libmysqlclient	Um dos	Necessário se o MySQL for usado.

Requisito	Status obrigatório	Descrição
libmariadb		Necessário se o MariaDB for usado.
libpq5		Necessário se o PostgreSQL for usado; a versão do *libpq5* deve ser igual ou 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 oferecer suporte a IPMI. Necessário apenas pelo Zabbix server.
libssh2 ou *libssh*		Necessário para verificações SSH. Versão 1.8.0 ou superior (libssh2); 0.9.0 ou superior (libssh).
libcurl		Necessário para os seguintes recursos: - itens de monitoramento web, monitoramento VMware e HTTP agent (para todos: versão 7.19.1 ou superior); - itens web.page.* do Zabbix agent (versão 7.19.1 ou superior; consulte também os requisitos do agent); - autenticação SMTP (Básica: versão 7.20.0 ou superior; OAuth: versão 7.33 ou superior; consulte também os requisitos do frontend); - Elasticsearch (versão 7.28.0 ou superior); - itens de consulta de telemetria (versão 7.28.0 ou superior). É recomendável usar a versão 7.28.0 ou superior para todos os recursos. Para usar os recursos atualizados do cURL nos 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 oferecer 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 versão 5.8 da biblioteca net-snmp; em sistemas baseados no RHEL 8 ou superior, recomenda-se usar o 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 oferecer suporte a LDAP.
fping		Necessário para itens de ping ICMP.
c-ares		Necessário para resolução assíncrona de DNS se o Zabbix estiver configurado com a opção `--with-ares`. Caso contrário, o *libevent* será usado. Versão mínima: 1.16.0. O recurso de armazenamento em cache de consultas DNS e failover (o cache de consultas é habilitado por padrão) requer o c-ares 1.26.0 ou posterior. Em sistemas nos quais o c-ares é fornecido pelo pacote do sistema operacional, certifique-se de que a biblioteca de runtime instalada exponha os recursos do resolvedor necessários para o armazenamento em cache e o failover de DNS.
g++		Compilador C++. Necessário para compilar o Zabbix proxy com a opção `--with-apm`, para monitoramento de desempenho de aplicações (APM).
gRPC		Necessário para APM no Zabbix proxy, se compilado com a opção `--with-apm`. Bibliotecas necessárias: *grpc++*, *grpc*, *gpr*. O utilitário `grpc_cpp_plugin` também é necessário para compilar o Zabbix proxy. A versão necessária está disponível nos repositórios de pacotes padrão destes sistemas operacionais: - Ubuntu 23.04 ou posterior; - Debian 12 (Bookworm) ou posterior; - Alpine Linux 3.15 ou posterior; - RHEL, Rocky Linux, CentOS Stream 9 ou posterior (somente no repositório EPEL); - Fedora X ou posterior. Em versões mais antigas dos sistemas operacionais, o *gRPC* deve ser compilado a partir do código-fonte.
protobuf		Necessário para APM no Zabbix proxy, se compilado com a opção `--with-apm`. O compilador `protoc` também é necessário para compilar o Zabbix proxy.
Abseil		Necessário para APM no Zabbix proxy, se compilado com a opção `--with-apm`. Bibliotecas necessárias: *absl_synchronization*, *absl_base*. Geralmente instalado junto com o *gRPC*.

Agent

Requirement	Mandatory status	Description
libpcre2	Yes	A biblioteca PCRE2 é necessária para suporte a Expressões Regulares Compatíveis com Perl (PCRE). A versão 10.x do PCRE2 é suportada. Necessária para monitoramento de logs. Também é necessária no Windows.
libpthread		Necessária para suporte a mutex e bloqueio de leitura e gravação (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 or *libopenssl*	No	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 posterior.
libldap		Necessária se LDAP for usado. Não 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 oferece suporte a recursos adicionais, como URLs HTTP com credenciais (por exemplo, http://user:password@example.com) e URLs HTTPS. É necessária a versão 7.19.1 ou superior (recomenda-se 7.28.0 ou superior). Para usar recursos atualizados do cURL, reinicie o Zabbix agent.
libmodbus		Necessária apenas se o monitoramento Modbus for usado. Versão 3.0 ou superior.

Agent 2

Requirement	Mandatory status	Description
Go	Yes	Necessário para compilar o Zabbix agent 2 e seus plugins a partir do código-fonte. Go 1.25 ou posterior é suportado. Consulte go.dev para instruções de instalação. As bibliotecas Go usadas pelo Zabbix agent 2 e seus plugins estão listadas no repositório Git do Zabbix (as bibliotecas marcadas como `indirect` no repositório são dependências de outras bibliotecas necessárias): - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	A biblioteca PCRE2 é necessária para suporte a Expressões Regulares Compatíveis com Perl (PCRE). PCRE2 v10.x é suportado. Necessária para monitoramento de logs. Também é necessária no Windows.
libopenssl	No	Necessária ao usar criptografia. OpenSSL 1.0.1 ou posterior é necessário em plataformas UNIX. A biblioteca OpenSSL deve ter suporte a PSK habilitado. LibreSSL não é suportado. Em sistemas Microsoft Windows, OpenSSL 1.1.1 ou posterior é necessário.

Serviço web

A versão estável mais recente do Google Chrome é suportada para gerar relatórios agendados usando o serviço web do Zabbix.

A versão do Go necessária para compilar o serviço web corresponde à utilizada para o Zabbix agent 2.

Java gateway

Se você obteve o Zabbix do repositório de código-fonte ou de um arquivo compactado, as dependências necessárias já estão incluídas na árvore de código-fonte.

Se você obteve o Zabbix do pacote da sua distribuição, as dependências necessárias já são fornecidas pelo sistema de empacotamento.

Em ambos os casos acima, o software está pronto para ser usado e não são necessários downloads adicionais.

No entanto, se você quiser fornecer suas próprias versões dessas dependências (por exemplo, se estiver preparando um pacote para alguma distribuição Linux), abaixo está a lista de versões das bibliotecas com as quais o Java gateway é conhecido por funcionar. O Zabbix também pode funcionar com outras versões dessas bibliotecas.

A tabela a seguir lista os arquivos JAR atualmente incluídos no Java gateway no código original:

Biblioteca	Status obrigatório	Versão incluída	Comentários
android-json	Sim	4.3r1	JSON (JavaScript Object Notation) é um formato leve de intercâmbio de dados. Esta é a implementação compatível com `org.json` do Android, extraída do Android SDK.
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

O Java gateway pode ser compilado usando o Oracle Java ou o OpenJDK de código aberto (versão 8 ou posterior). 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 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)

Distribuição	Pacote OpenJDK
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Adaptador de bridge

O adaptador de bridge do Zabbix não tem requisitos especiais.

Portas padrão

A lista a seguir contém as portas que os componentes do Zabbix podem usar em uma configuração padrão:

Componente do Zabbix	Protocolo e porta	Conexão iniciada por
Zabbix agent	TCP 10050	Zabbix server ou Zabbix proxy (somente verificações passivas)
Zabbix agent 2	TCP 10050	Zabbix server ou Zabbix proxy (somente verificações passivas)
Zabbix server	TCP 10051	Zabbix agents ativos, Zabbix proxy ativo, Zabbix sender, Zabbix frontend
Zabbix proxy	TCP 10051	Zabbix agents ativos, Zabbix sender, Zabbix server (somente proxy passivo)
Zabbix Java gateway	TCP 10052	Zabbix server ou proxy
Zabbix web service	TCP 10053	Zabbix server
Zabbix MCP server	TCP 8443	Clientes MCP (agents de IA)
Zabbix frontend	TCP 80	Navegadores da web, Zabbix web service (relatórios agendados), clientes da API
	TCP 443	
Traps SNMP (snmptrapd)	UDP 162	Dispositivos SNMP
Adaptador de ponte do Zabbix	TCP 10054	Zabbix server

Warning:
Abra somente as portas necessárias para os componentes que você realmente executa e restrinja cada regra de firewall aos hosts específicos que precisam se conectar (por exemplo, permita a porta 10050 somente a partir do Zabbix server ou proxy, e não de qualquer endereço). Não exponha as portas do Zabbix diretamente à internet.
 Antes de abrir as portas, proteja a instalação: habilite a criptografia entre os componentes, restrinja os hosts permitidos nos arquivos de configuração dos componentes e revise as práticas recomendadas de segurança.

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 server do 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 history

O Zabbix mantém 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 history e recebermos 50 valores por segundo, o número total de valores será em torno de (**30***24*3600)* **50** = 129.600.000, ou cerca de 130M de valores.

Dependendo do mecanismo de banco de dados usado, do tipo de valores recebidos (floats, integers, 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 com exatidão, mas você pode esperar cerca de 500 bytes por valor.

- Configuração do housekeeper para trends

O Zabbix mantém um conjunto de valores de max/min/avg/count 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 queiramos manter dados de trends por 5 anos. Os valores para 3000 items exigirão 3000*24*365* **90** = **2.2GB** por ano, ou **11GB** por 5 anos.

- Configurações do housekeeper para events

Cada event do Zabbix requer aproximadamente 250 bytes de espaço em disco[1]. É difícil estimar o número de events gerados pelo Zabbix diariamente. No pior cenário, podemos assumir que o Zabbix gera um event por segundo.

Para cada event recuperado, um registro event_recovery é criado. Normalmente, a maioria dos events será recuperada, então podemos assumir um registro event_recovery por event. Isso significa 80 bytes adicionais por event.

Opcionalmente, events podem ter tags, e cada registro de tag requer aproximadamente 100 bytes de espaço em disco[1]. O número de tags por event (#tags) depende da configuração. Portanto, cada um precisará de #tags * 100 bytes adicionais de espaço em disco.

Isso significa que, se quisermos manter 3 anos de events, isso exigirá 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B de espaço em disco[2].

Note:
[1] Mais quando houver nomes, tags e valores de events não ASCII.
 [2] As estimativas de tamanho são baseadas no 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:

Parameter	Formula for required disk space (in bytes)
Zabbix configuration	Fixed size. Normally 10MB or less.
History	days*(items/refresh rate)*24*3600*bytes items : number of items days : number of days to keep history refresh rate : average refresh rate of items bytes : number of bytes required to keep single value, depends on database engine, normally ~90 bytes.
Trends	days*(items/3600)*24*3600*bytes items : number of items days : number of days to keep history bytes : number of bytes required to keep single trend, depends on the database engine, normally ~90 bytes.
Events	days*events*24*3600*bytes events : number of event per second. One (1) event per second in worst-case scenario. days : number of days to keep history bytes : number of bytes required to keep single event, depends on the database engine, normally ~330 + average number of tags per event * 100 bytes.

Portanto, o espaço total em disco necessário pode ser calculado como:

Configuration + History + Trends + Events

O espaço em disco NÃO será usado imediatamente após a instalação do Zabbix. O tamanho do banco de dados crescerá e então deixará de crescer em algum momento, o que depende das configurações do housekeeper.

Sincronização de tempo

É muito importante ter o tempo do sistema preciso no servidor com o Zabbix em execução. ntpd é o daemon mais popular que sincroniza o tempo do host com o tempo de outras máquinas. É altamente recomendável manter o tempo do sistema sincronizado em todos os sistemas em que os componentes do Zabbix estão em execução.

Novidades e notas de atualização: 8.0

O Zabbix 8.0.0 dá continuidade ao Zabbix 7.4.0, adicionando novos recursos e aprimoramentos.

Também nesta página:

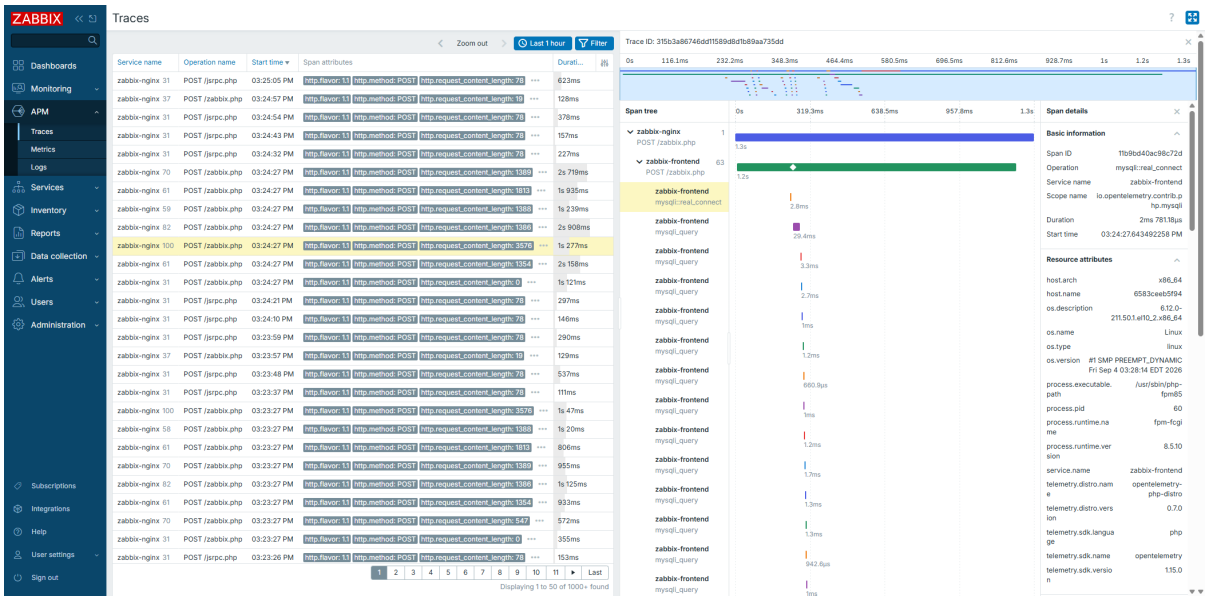
- Novidades nos templates, incluindo novos templates e alterações nos templates existentes.
- Notas de atualização, incluindo alterações incompatíveis.

Consulte também: Alterações na API.

Monitoramento de desempenho de aplicações O Zabbix agora oferece suporte ao monitoramento de desempenho de aplicações (APM) com base no OpenTelemetry, um padrão aberto para coletar dados de telemetria de aplicações e serviços. Esses dados ajudam você a visualizar como as solicitações fluem pelas suas aplicações e serviços, medir o desempenho deles e encontrar a causa de operações lentas ou com falha.

Com o APM, você pode:

- Coletar dados de telemetria com o proxy do Zabbix, que agora pode receber dados de aplicações ou coletores do OpenTelemetry por meio do protocolo OpenTelemetry (OTLP/gRPC) e gravá-los na fonte de dados; consulte a nova aba *APM* em configuração do proxy.
- Especificar uma fonte de dados de telemetria (ClickHouse) na nova seção *Administração* > *Fonte de dados* e visualizar os dados no novo menu *APM*.
- Usar dados de telemetria em triggers e gráficos com o novo tipo de item consulta de telemetria.



Os seguintes parâmetros e itens também foram adicionados:

- Parâmetros do server do Zabbix: `TelemetryProvider` e `StartTelemetryQueryPollers`.
- Parâmetros do proxy do Zabbix: `TelemetryProvider`, `StartTelemetryQueryPollers`, `StartAPMCollectors`, `APMListenIP`, `APMListenPort`, `APMTLSCAFile`, `APMTLSCertFile` e `APMTLSKeyFile`.
- Parâmetro do frontend do Zabbix: `$TELEMETRY_PROVIDERS[]`.
- Opção de compilação do proxy do Zabbix: `--with-apm`; os pacotes oficiais do proxy do Zabbix são compilados com essa opção.
- Item interno para monitorar a coleta de dados de telemetria: `zabbix[apm,<type>]`.

Suporte ao ClickHouse Agora você pode usar o ClickHouse como banco de dados backend do Zabbix para armazenar o histórico dos valores dos item e como fonte de dados de telemetria para monitoramento do desempenho de aplicações.

Para obter detalhes e requisitos da configuração do armazenamento de histórico, consulte Configuração do ClickHouse. Se quiser migrar seus dados de histórico de um banco de dados Zabbix existente (MySQL ou PostgreSQL) para o ClickHouse, consulte os scripts de migração do esquema e do histórico do ClickHouse.

Esse recurso também altera a forma como você define os provedores de armazenamento de histórico nos arquivos de configuração do Zabbix. Essa alteração afeta as configurações do ClickHouse e do Elasticsearch. Para obter detalhes, consulte as Notas de atualização.

Processamento de eventos complexos O processamento de eventos complexos (CEP) realiza o processamento baseado em regras para eventos do Zabbix.

Por exemplo, você pode querer:

- agrupar eventos relacionados e marcar o primeiro evento de uma série como a causa (e os seguintes como sintomas)
- especificar um período de retenção para os eventos
- especificar um número máximo para determinados eventos
- receber notificações sobre eventos semelhantes por localização
- receber uma notificação se determinado evento ocorrer com muita frequência (ou não ocorrer) dentro de um período

O processamento de eventos complexos introduz **janelas de tempo** deslizantes para a análise de eventos.
Por exemplo, em uma janela de 10 minutos, todos os eventos que ocorrerem dentro da janela formarão um único contexto.
À medida que a janela avança, os eventos mais antigos deixam a janela (são ”expulsos”), enquanto novos eventos podem entrar.

O conceito de janela de tempo apresenta pontos lógicos para a execução de operações.
Por exemplo, um evento pode ser transformado ou ”enriquecido” quando entra na janela: a severidade pode ser aumentada ou reduzida, tags podem ser adicionadas ou ajustadas, a supressão pode ser ativada etc.

Operações semelhantes podem ser executadas quando o evento deixa a janela ou quando um padrão é encontrado.

A análise do fluxo de eventos é definida por regras de CEP.
Uma regra de CEP normalmente consiste em uma janela de tempo, condições e operações.
As operações estão vinculadas a um ponto de execução.

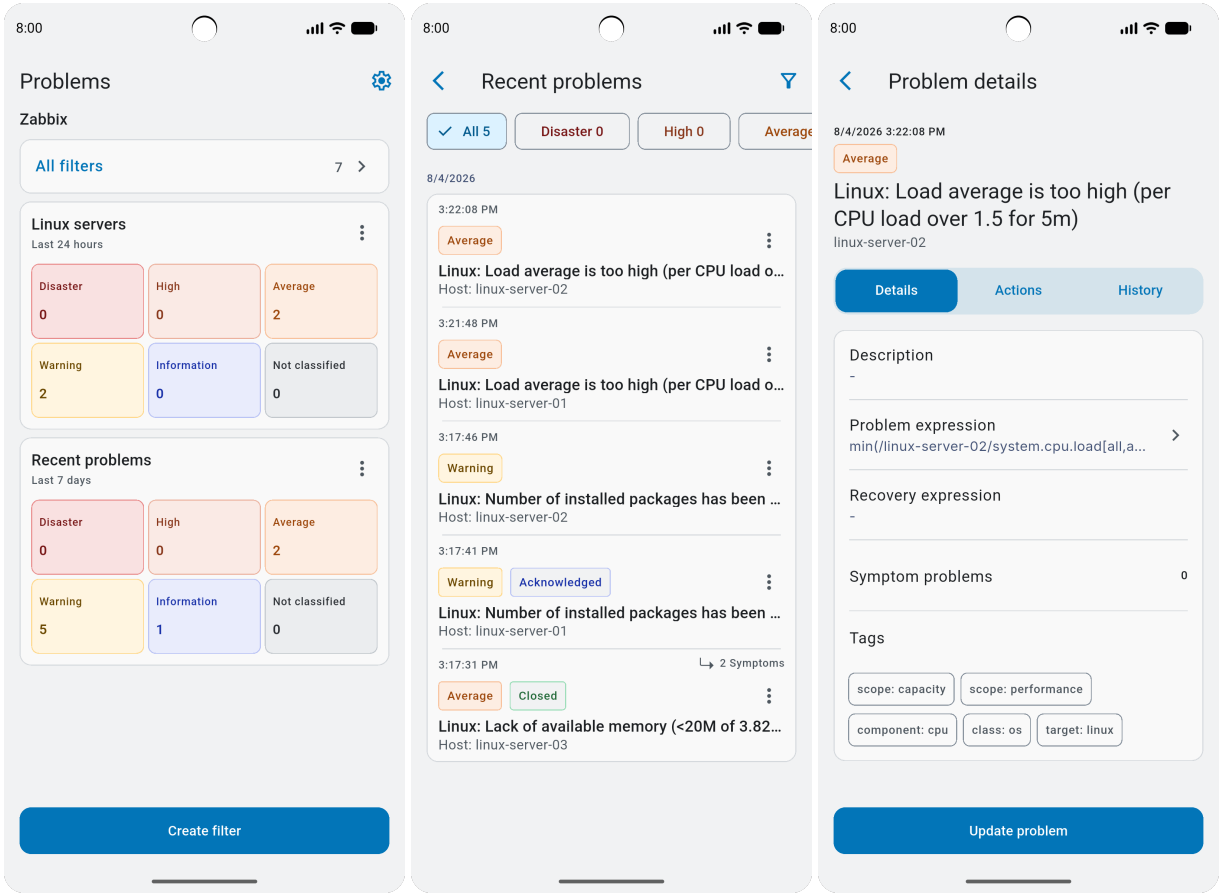
Uma documentação mais detalhada estará disponível quando a versão 8.0 estável for lançada.

Aplicativo móvel O aplicativo móvel do Zabbix (em breve) permite monitorar seu ambiente Zabbix pelo dispositivo móvel. Visualize problemas, verifique seus detalhes, atualize problemas e receba notificações push quando algo precisar de atenção.

Com o aplicativo, os seguintes recursos são novos no Zabbix 8.0:

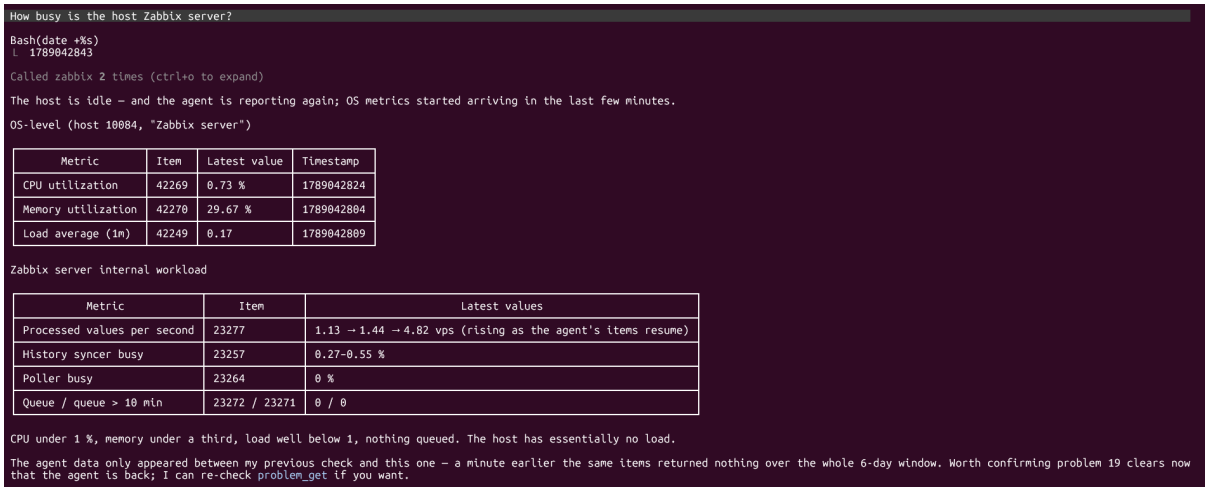
- Seções da interface web para gerenciar dispositivos móveis: *Usuários* > *Dispositivos* e *Configurações do usuário* > *Dispositivos*.
- Tipo de mídia para enviar notificações de problemas do Zabbix server para o aplicativo móvel: Notificação push.
- Componente que permite que seu Zabbix server se comunique com o aplicativo móvel: Adaptador de bridge.
- Parâmetros de configuração do Zabbix server: BridgeAdapterURL, EnableMobileDevices.

Para obter detalhes sobre como configurar o Zabbix para uso com o aplicativo móvel, consulte a seção Aplicativo móvel.



MCP server O novo Zabbix MCP server conecta agentes de IA ao Zabbix por meio do Model Context Protocol (MCP). Agora você pode perguntar a um assistente de IA sobre seu monitoramento em linguagem natural, por exemplo:

- *Quais problemas nos servidores de banco de dados ainda não foram reconhecidos?*
- *Por que o web-01 está relatando problemas desde esta manhã?*
- *Coloque todos os hosts do grupo de servidores Linux em manutenção hoje à noite, das 22:00 às 02:00.*



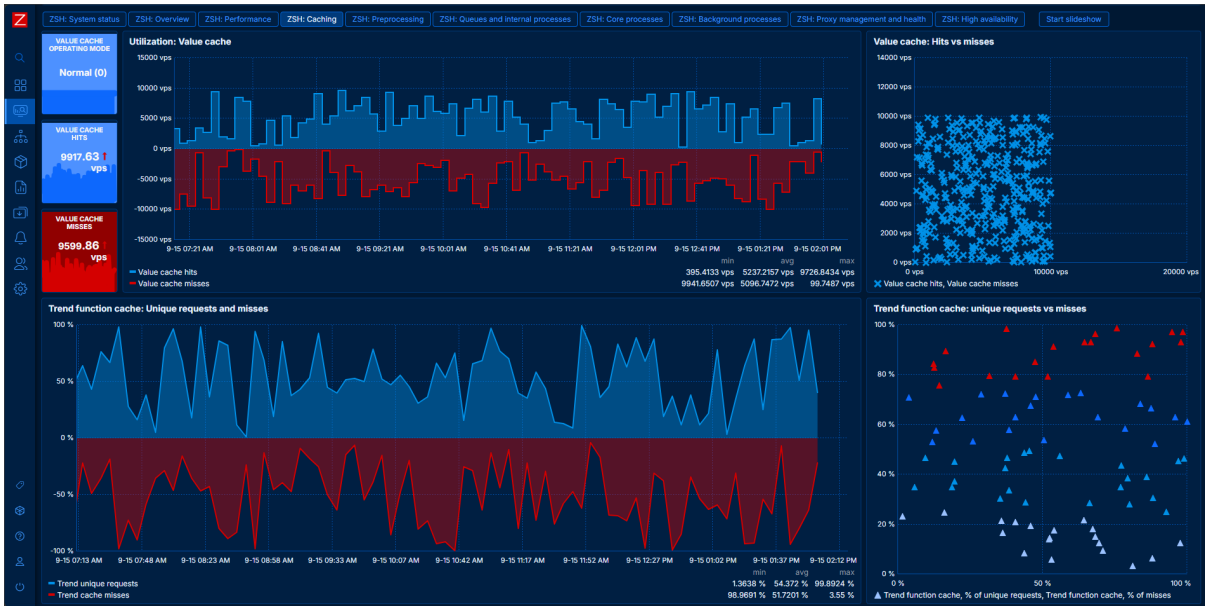
O agent se conecta com um token da API do Zabbix, portanto, pode acessar apenas o que o proprietário do token tem permissão para acessar, e todas as alterações são registradas no log de auditoria desse usuário. Para ver a lista de operações disponíveis, consulte Ferramentas e recursos.

Manutenção sob demanda para triggers Os períodos de manutenção agora podem ser definidos para triggers ou nomes de eventos.

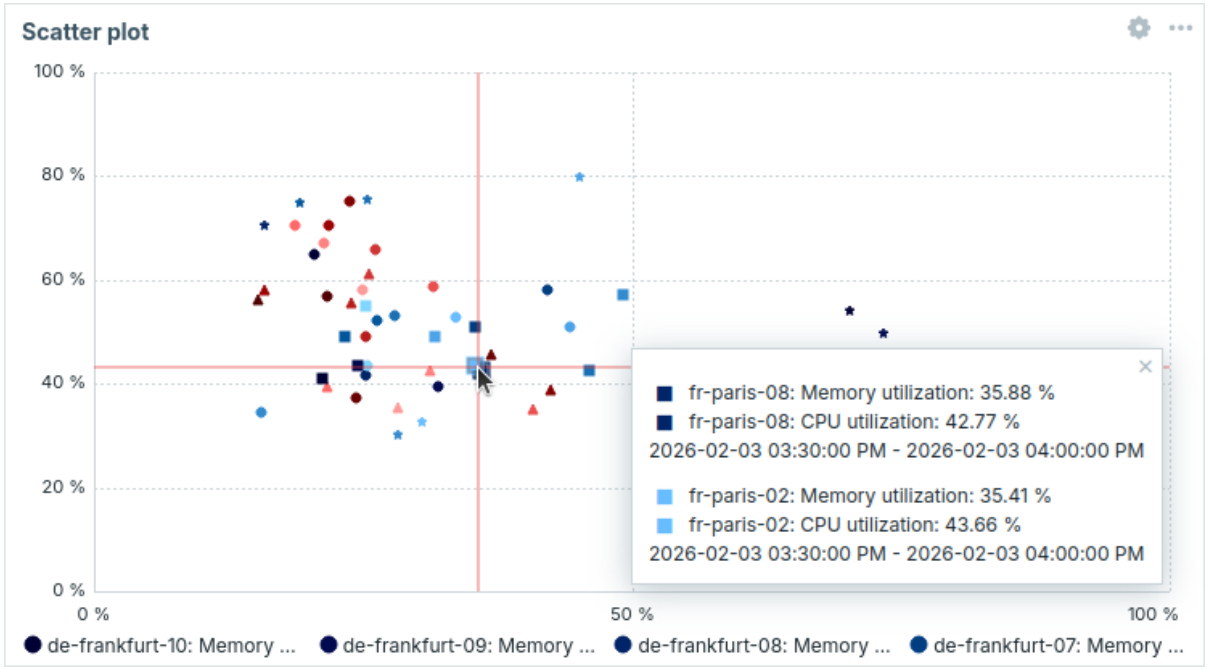
Um período de manutenção também pode ser criado sob demanda no menu de eventos. A manutenção sob demanda pode ser iniciada para um host, trigger, nome de evento ou tags.

Outra forma de criar uma manutenção para um trigger é por meio do formulário de atualização do problema.

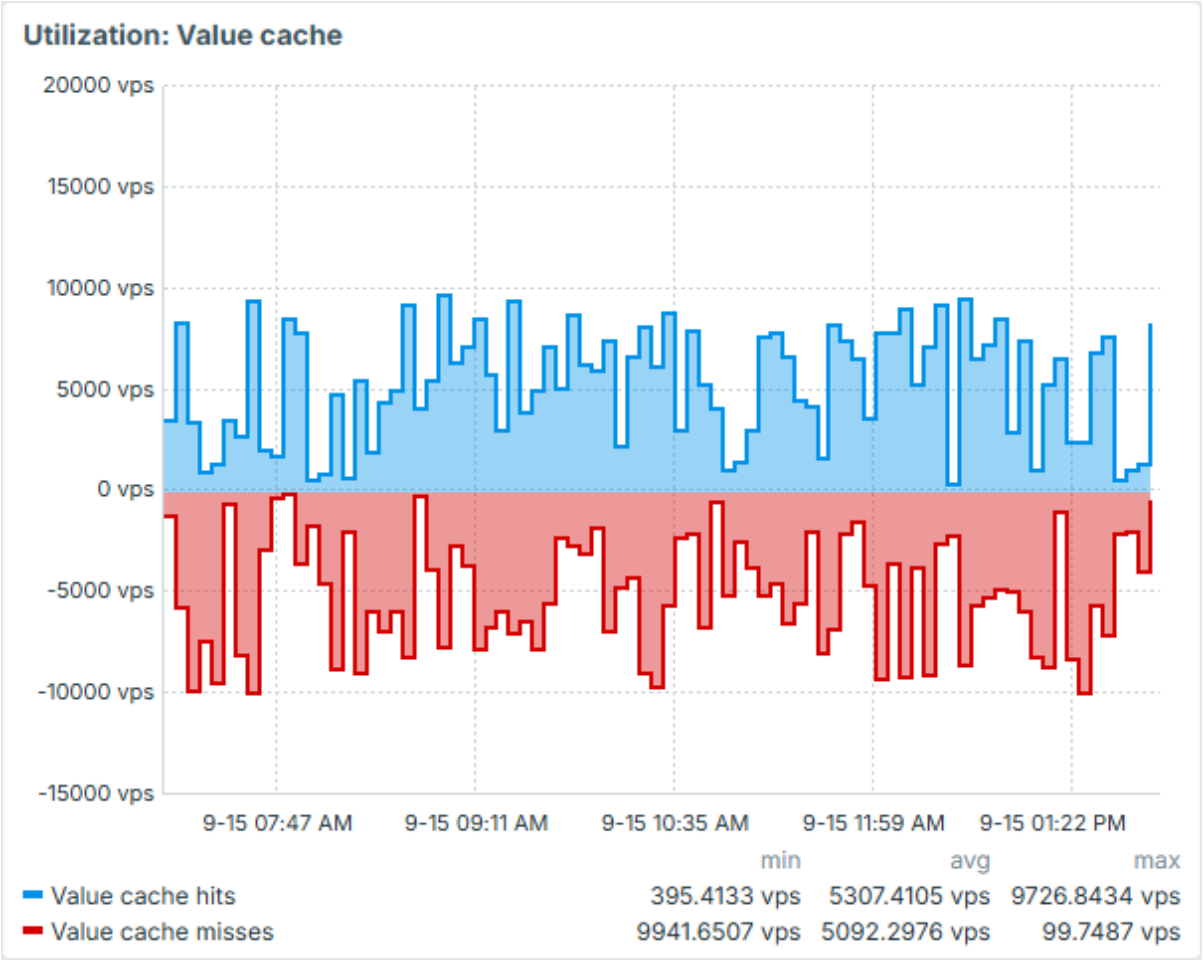
Tema azul-escuro A interface web do Zabbix agora oferece um novo tema azul-escuro. Ele está disponível no perfil do usuário e como o tema padrão:



Widget de gráfico de dispersão O widget Gráfico de dispersão foi adicionado aos widgets do dashboard. Ele exibe a relação entre duas métricas, plotando pontos de dados individuais ao longo dos eixos X e Y. Isso ajuda a revelar padrões, agrupamentos, correlações e valores discrepantes no conjunto de dados.



Inversão do eixo Y para o widget de gráfico Gráficos no widget de gráfico agora podem ser exibidos com valores invertidos no eixo Y. Uma nova configuração *Invert values* permite multiplicar os valores do eixo Y por −1 sem alterar os dados originais.



Veja também: Melhorias no widget de gráfico.

Tipo de dado JSON O Zabbix agora oferece suporte a JSON como um tipo de dado para valores de item.

Anteriormente, valores JSON eram coletados por itens de texto e armazenados como strings com um limite de 64 KB. Agora, o Zabbix pode armazenar valores JSON nativamente com um limite de 128 MiB e também rejeitar valores JSON inválidos (por exemplo, contendo chaves sem aspas, vírgulas finais ou colchetes incompatíveis).

O tipo de dado JSON é suportado por todos os tipos de item e protótipo de item, exceto *Calculated*, e está disponível na exportação de dados em tempo real e em connectors. Valores JSON podem ser armazenados em todos os bancos de dados suportados e no Elasticsearch. Se você estiver usando TimescaleDB, consulte as notas de atualização.

Observe que itens com o tipo de dado JSON não podem ser usados em triggers; no entanto, você pode extrair campos JSON com itens dependentes que tenham um tipo de dado não JSON e usá-los em triggers.

Itens que retornam uma string JSON (`net.if.discovery`, `vfs.file.get`, etc.) ainda são itens de texto; no entanto, você pode alterá-los para JSON, se necessário.

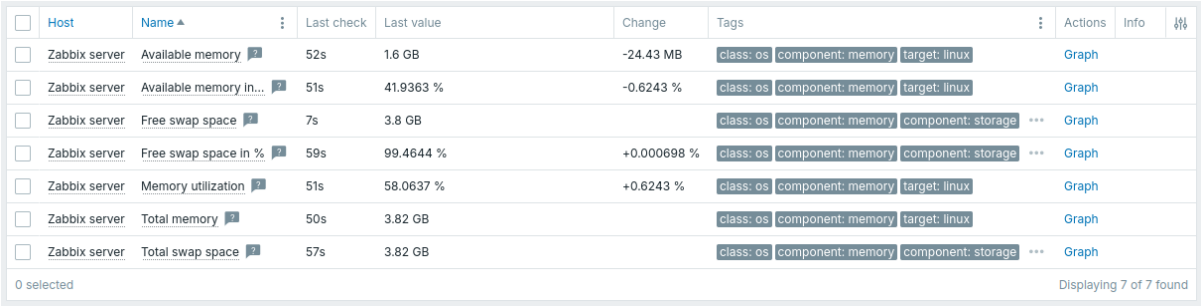
Para mais detalhes e limites de dados JSON, consulte a configuração do item.

Layout personalizado para tabelas Em várias tabelas do frontend, agora você pode ocultar/exibir, reordenar e redimensionar colunas. Também é possível adicionar uma coluna *Texto personalizado* com texto personalizado, valores de macro ou ambos.

Layouts personalizados estão disponíveis nestes locais:

- *Monitoring* > *Hosts*, *Latest data* e *Problems*
- *Data collection* > *Templates* e *Hosts*

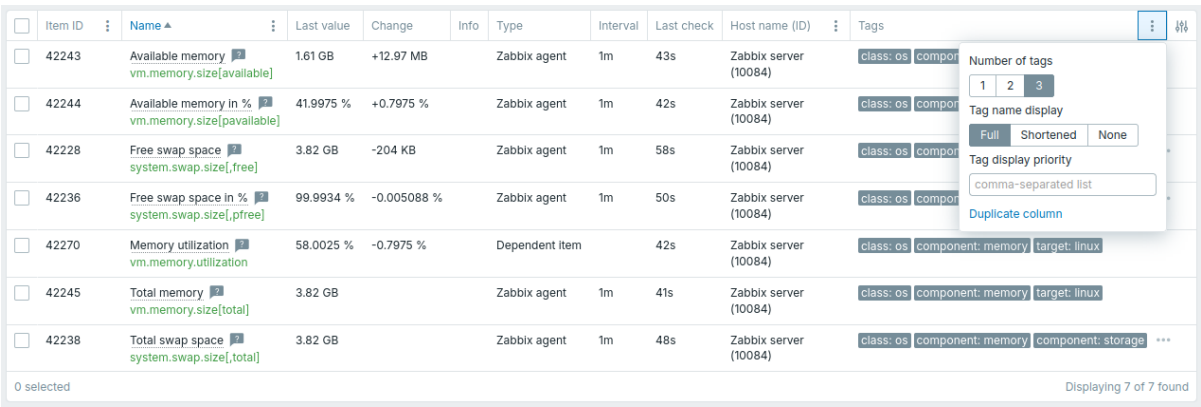
Por exemplo, o layout padrão da tabela em *Monitoring* > *Latest data*:



Host	Name	Last check	Last value	Change	Tags	Actions	Info
Zabbix server	Available memory	52s	1.6 GB	-24.43 MB	class: os component: memory target: linux	Graph	
Zabbix server	Available memory in...	51s	41.9363 %	-0.6243 %	class: os component: memory target: linux	Graph	
Zabbix server	Free swap space	7s	3.8 GB		class: os component: memory component: storage	Graph	
Zabbix server	Free swap space in %	59s	99.4644 %	+0.000698 %	class: os component: memory component: storage	Graph	
Zabbix server	Memory utilization	51s	58.0637 %	+0.6243 %	class: os component: memory target: linux	Graph	
Zabbix server	Total memory	50s	3.82 GB		class: os component: memory target: linux	Graph	
Zabbix server	Total swap space	57s	3.82 GB		class: os component: memory component: storage	Graph	

0 selected — Displaying 7 of 7 found

Exemplo de um layout personalizado:



Item ID	Name	Last value	Change	Info	Type	Interval	Last check	Host name (ID)	Tags
42243	Available memory vm.memory.size[available]	1.61 GB	+12.97 MB		Zabbix agent	1m	43s	Zabbix server (10084)	class: os compon
42244	Available memory in % vm.memory.size[pavailable]	41.9975 %	+0.7975 %		Zabbix agent	1m	42s	Zabbix server (10084)	class: os compon
42228	Free swap space system.swap.size[,free]	3.82 GB	-204 KB		Zabbix agent	1m	58s	Zabbix server (10084)	class: os compon
42236	Free swap space in % system.swap.size[,pfree]	99.9934 %	-0.005088 %		Zabbix agent	1m	50s	Zabbix server (10084)	class: os compon
42270	Memory utilization vm.memory.utilization	58.0025 %	-0.7975 %		Dependent item		42s	Zabbix server (10084)	class: os component: memory target: linux
42245	Total memory vm.memory.size[total]	3.82 GB			Zabbix agent	1m	41s	Zabbix server (10084)	class: os component: memory target: linux
42238	Total swap space system.swap.size[,total]	3.82 GB			Zabbix agent	1m	48s	Zabbix server (10084)	class: os component: memory component: storage

Number of tags: 1 2 3
Tag name display: Full Shortened None
Tag display priority: comma-separated list
Duplicate column

0 selected — Displaying 7 of 7 found

Algumas colunas também têm o botão ⋮ *Options*, que fornece controles adicionais sobre como essa coluna exibe os dados. Por isso, alguns parâmetros de filtro foram movidos da aba de filtro para as opções da coluna. Se você estiver usando abas para filtros favoritos, consulte Notas de atualização.

Expressões regulares globais Foram adicionadas várias melhorias de usabilidade às expressões regulares globais:

- O formulário de configuração de expressões regulares globais agora é aberto em uma janela modal:
- Os tipos de expressão foram renomeados.
- O formulário agora inclui um campo *Descrição*.
- O teste não está mais em uma aba separada.
- A lista de expressões regulares globais agora pode ser filtrada por nome e descrição.
- As expressões regulares globais agora podem ser exportadas e importadas.

Templates

Novos templates foram adicionados, e muitos templates foram atualizados.

Os templates que se conectam a endpoints HTTPS agora verificam o certificado TLS do endpoint por padrão. Isso afeta todos os templates de monitoramento de nuvem e API, como *Claude API by HTTP*, *Cloudflare by HTTP*, *Kubernetes Cluster by HTTP* etc. Consulte as notas de atualização antes de atualizar qualquer template baseado em HTTP.

Ao atualizar o Zabbix, os templates existentes não são atualizados automaticamente para evitar a substituição de modificações personalizadas. Para atualizar templates ou adicionar novos, consulte Atualização de templates.

> **Note:**
> Informamos que, desde o Zabbix 6.0, todos os templates seguem um formato atualizado, o que pode afetar a importação de templates anteriores à versão 6.0. Para obter mais informações, consulte Alterações nos templates na versão 6.0.

Novos templates **Monitoramento do uso de IA**

- Claude API by HTTP, um template para monitorar o uso e os custos da API Claude.
- OpenAI Platform by HTTP, um template para monitorar a plataforma de desenvolvimento da OpenAI.

Nuvem

- O conjunto de templates AWS by HTTP foi complementado com o template *AWS Backup Vault by HTTP*.
- O conjunto de templates Azure by HTTP foi complementado com os templates *Azure Sentinel by HTTP* e *Azure Container Apps by HTTP*.
- O conjunto de templates GCP by HTTP foi complementado com os templates *GCP Cloud Run Service by HTTP*, *GCP Cloud Storage Bucket by HTTP*, *GCP Application Load Balancer by HTTP* e *GCP Cost Monitoring by HTTP*.

- O conjunto de templates Oracle Cloud by HTTP foi complementado com os templates *Oracle Cloud Load Balancer by HTTP* e *Oracle Cloud Costs by HTTP*.

Virtualização e contêineres

- Kubernetes Cluster by HTTP, um template para monitorar clusters Kubernetes, com nós descobertos automaticamente e monitorados por meio de hosts dedicados; o template substitui os templates Kubernetes anteriores.
- Microsoft Hyper-V Failover Cluster by SSH e Microsoft Hyper-V Standalone by SSH, templates para monitorar clusters e hosts independentes do Microsoft Hyper-V via SSH.
- Podman by Zabbix agent, by Zabbix agent active e by HTTP: templates para monitorar contêineres Podman e a integridade dos serviços por meio da API do Podman. Observe que a API do Podman não oferece suporte a métodos de autorização.

Bancos de dados

- MariaDB by ODBC, um template para monitorar bancos de dados MariaDB via ODBC.
- MariaDB by Zabbix agent e MariaDB by Zabbix agent active, templates para monitorar bancos de dados MariaDB via Zabbix agent e Zabbix agent active.
- Percona by ODBC, um template para monitorar bancos de dados Percona via ODBC.
- Percona by Zabbix agent e Percona by Zabbix agent active, templates para monitorar bancos de dados Percona via Zabbix agent e Zabbix agent active.

Rede, segurança e telecomunicações

- Aruba CX 8300s by SNMP, um template que fornece monitoramento baseado em SNMP para a série de switches Aruba CX 8300.
- Ciena 3906 by SNMP, um template para monitorar dispositivos Ciena 3906, incluindo sistema de arquivos e carga da CPU.
- Cisco Secure Firewall Threat Defense by HTTP, um template que fornece recursos de monitoramento para dispositivos Cisco Secure Firewall Threat Defense usando a REST API.
- Cradlepoint NCM v2 by HTTP, um conjunto de templates para monitorar o Cradlepoint NCM v2 e seus dispositivos via HTTP.
- Huawei AR600 by SNMP, um template para monitorar roteadores Huawei AR600, incluindo métricas de CoS e NQA.
- Ribbon SBC Edge by HTTP, um template para monitorar dispositivos Ribbon SBC Edge (anteriormente SWe Lite) via HTTP.
- Ribbon SBC SWe Core by HTTP, um template para monitorar dispositivos SBC SWe Core via HTTP, e Ribbon SBC SWe CE by HTTP, um template para monitorar instâncias do Ribbon SBC SWe Call Engine (CE) via HTTP.
- Stormshield SNS by SNMP, um template para monitorar dispositivos Stormshield Network Security (SNS) via SNMP.
- VeloCloud SD-WAN Edge by HTTP, um template para monitorar dispositivos VeloCloud SD-WAN Edge via HTTP.
- Vyatta Virtual Router by SNMP, um template para monitorar o roteador virtual Vyatta 1908e.

Aplicações e serviços

- Domain RDAP by HTTP, um template para monitorar dados de registro de domínios por meio do protocolo RDAP.
- Envoy Proxy by OTLP, um template para coletar traces e logs do OpenTelemetry do Envoy Proxy por meio de itens *Telemetry query*.
- Generic OpenTelemetry by OTLP, um template para coletar traces, métricas e logs do OpenTelemetry por meio de itens *Telemetry query*.
- GLPI by HTTP, um template para monitorar o GLPI via HTTP usando a REST API v2 com autenticação OAuth2.
- OpenRouter traces by OTLP, um template para coletar dados de trace do OpenRouter.
- Proxmox VE by OTLP, um template para monitorar um nó do Proxmox VE usando seu fluxo de métricas do OpenTelemetry por meio de itens *Telemetry query*.

Templates atualizados

- O trigger de tempo de atividade do dispositivo nos templates SNMP foi aprimorado para lidar com redefinições do contador de tempo de atividade do SNMP e evitar alertas falsos de reinicialização.

Sistemas operacionais

- Nos templates AIX, FreeBSD, HP-UX, Linux, macOS, OpenBSD, Solaris e Windows, o monitoramento de sistemas de arquivos agora usa os novos parâmetros `vfs.fs.get` (consulte Descoberta leve de sistemas de arquivos). Para obter detalhes, consulte Alterações nos templates.
- Linux by Zabbix agent e Linux by Zabbix agent active foram atualizados com dashboards aprimorados e monitoramento avançado de interfaces de rede e dispositivos de bloco.
- Linux by Prom e Linux by SNMP foram atualizados com dashboards aprimorados.
- Os templates Windows by Zabbix agent e Windows by Zabbix agent active foram aprimorados com novas métricas, regras de descoberta, protótipos, macros de usuário e widgets de dashboard, incluindo suporte a tarefas agendadas do Windows, perfis do Firewall do Windows e volumes criptografáveis pelo BitLocker.

Monitoramento do uso de IA

- GitHub organization by HTTP foi atualizado para incluir itens de monitoramento do Microsoft Copilot.

- Microsoft 365 reports by HTTP foi atualizado para incluir itens de monitoramento do Microsoft Copilot.

Nuvem

- *AWS Cost Explorer by HTTP*, do conjunto de templates AWS by HTTP, e *Azure Cost Management by HTTP*, do conjunto de templates Azure by HTTP, foram atualizados com dashboards aprimorados.

Virtualização

- Proxmox VE by HTTP foi atualizado com a funcionalidade LLD aninhada. Além disso, o formato da unidade dos itens que exibem porcentagens foi ajustado para maior clareza.
- Nos conjuntos de templates VMware e VMware FQDN, os nomes dos protótipos de item e trigger para a descoberta de alarmes do VMware foram atualizados para incluir o nome da entidade do alarme.

Bancos de dados

- MySQL by Zabbix agent, MySQL by Zabbix agent 2, MySQL by Zabbix agent active, MySQL by Zabbix agent 2 active e MySQL by ODBC foram atualizados para oferecer suporte a `SHOW SLAVE STATUS` (sintaxe antiga) e `SHOW REPLICA STATUS` (sintaxe nova). Além disso, MySQL by Zabbix agent, MySQL by Zabbix agent active e MySQL by ODBC foram atualizados com novas métricas, um dashboard aprimorado e regras de descoberta expandidas para tabelas (bem como réplicas no caso de *MySQL by ODBC*).
- PostgreSQL by ODBC foi atualizado com novas métricas, regras de descoberta e dashboards, além de melhorias gerais de desempenho e observabilidade.
- PostgreSQL by Zabbix agent e PostgreSQL by Zabbix agent active foram atualizados com métricas adicionais, regras de descoberta, protótipos de item e trigger e recursos aprimorados de monitoramento.
- Redis by Zabbix agent 2 foi atualizado com autenticação por meio de macros de usuário e senha, além de descrições de item aprimoradas.

Rede, segurança e telecomunicações

- VeloCloud SD-WAN by HTTP, anteriormente denominado *VMWare SD-WAN VeloCloud by HTTP*, foi renomeado e atualizado para se alinhar à versão mais recente da plataforma VeloCloud SD-WAN.

Armazenamento e backup

- Ceph by Zabbix agent 2 agora oferece suporte ao modo de conexão nativa. Consulte Plugin do Ceph para obter detalhes.
- Veeam Backup and Replication by HTTP e Veeam Backup Enterprise Manager by HTTP foram atualizados com novos recursos de monitoramento e suporte ao Veeam Backup & Replication v13.1.1.18, incluindo a revisão 1.3-rev2 da REST API e o monitoramento atualizado da expiração da licença.

Monitoramento de sistemas de arquivos nos templates de SO Nos templates de AIX, FreeBSD, HP-UX, Linux, macOS, OpenBSD, Solaris e Windows (incluindo as variantes de verificação ativa dos templates de Linux e Windows), o monitoramento de sistemas de arquivos agora usa os novos parâmetros de `vfs.fs.get`:

- o item *Get filesystems* (`vfs.fs.get`) foi removido;
- a regra *Mounted filesystem discovery* agora é um item do Zabbix agent com a chave `vfs.fs.get[short]` (intervalo de atualização: 1h), em vez de uma regra de descoberta dependente;
- o protótipo de item *FS [{#FSNAME}]: Get data* (*FS [{#FSLABEL}({#FSNAME})]: Get data* nos templates de Windows) agora é um item do Zabbix agent com a chave `vfs.fs.get[full,{#FSNAME}]`, que coleta estatísticas de apenas um sistema de arquivos. Os outros protótipos de item de sistemas de arquivos usam esse item como item mestre.

Como resultado, as estatísticas são coletadas por uma verificação separada do agent para cada sistema de arquivos, e os sistemas de arquivos recém-montados são descobertos dentro do intervalo de atualização da descoberta (1h por padrão).

O monitoramento de sistemas de arquivos nesses templates requer o Zabbix agent 8.0 ou posterior.

Tipos de mídia

Este tipo de mídia foi adicionado:

- O tipo de mídia webhook IBM Maximo Service Request permite vincular o Zabbix ao IBM Maximo.

Estes tipos de mídia foram atualizados:

- O tipo de mídia webhook GLPI foi atualizado para oferecer suporte à API RESTful v2 com autorização OAuth2 (habilitada por padrão), suporte opcional à API REST legada v1 por meio de uma flag de configuração, criação de tickets para eventos de problema e mapeamento de urgência configurável pelo usuário.
- O tipo de mídia webhook MS Teams Workflow agora é a integração recomendada para o Microsoft Teams. O tipo de mídia webhook MS Teams, baseado no recurso Office 365 Connector (Incoming Webhook), que a Microsoft descontinuou em 22 de maio de 2026, não funciona mais. As configurações existentes do webhook MS Teams precisam ser migradas para o tipo de mídia webhook MS Teams Workflow.

Além disso, os tipos de mídia que se conectam a endpoints HTTPS agora verificam o certificado TLS do endpoint por padrão. Isso afeta todos os tipos de mídia webhook, como *Slack*, *Jira*, *Telegram* etc. Consulte as notas de atualização antes de atualizar qualquer tipo de mídia webhook.

Items

Valor padrão para hosts permitidos de item trapper O campo *Hosts permitidos* para itens Zabbix trapper e para itens HTTP agent com *Enable trapping* habilitado agora usa, por padrão, a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}`. Se esse valor estiver vazio, conexões de entrada serão negadas de todos os hosts. Para permitir o envio de dados, especifique um ou mais endereços IP ou nomes DNS.

Monitoramento da utilização do cache da função de tendência O parâmetro `paccessed` foi adicionado ao item interno `zabbix[tcache,cache,<parameter>]`. Ele retorna a porcentagem de entradas do cache da função de tendência acessadas nas últimas 24 horas.

Redirecionamentos em itens web.page.* Os itens web.page.get, web.page.perf e web.page.regexp do Zabbix agent e do Zabbix agent 2 agora podem seguir redirecionamentos.
O número máximo de redirecionamentos é definido no novo parâmetro opcional `redirect_limit`; por padrão, os redirecionamentos não são seguidos.

Plugins

Plugin Ceph O Zabbix agent 2 plugin Ceph agora opera em dois modos:

- **native** - 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 é suportado apenas **no Linux** e a partir do Ceph 16.
- **restful** (obsoleto) - Este modo usa a API RESTful do Ceph para comunicação. É o modo padrão por compatibilidade com versões anteriores, mas não funcionará com a versão 20 do Ceph (Tentacle) ou mais recente devido à remoção do módulo `mgr/restful`.

Importante: Ao monitorar o Ceph 20 (Tentacle) e versões mais recentes com o plugin Ceph para o Zabbix agent 2, use o modo native, pois o modo restful não é mais suportado.

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

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

Observe que o conjunto de credenciais do usuário é diferente para cada modo e eles não são compatíveis entre si. O parâmetro `Plugins.Ceph.InsecureSkipVerify` é ignorado no modo native 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 à exigência do pacote librados (para o modo native). Consulte o readme do plugin Ceph para obter detalhes.

Plugin MongoDB O Zabbix agent 2 MongoDB plugin oferece várias melhorias: - Suporte completo à análise convencional de URI do MongoDB, com suporte aos esquemas `mongodb://` e `mongodb+srv://`. - Descoberta automática dos nós do conjunto de réplicas do MongoDB, aprimorando a capacidade de monitorar clusters MongoDB. - Suporte à autenticação x509, permitindo conexões seguras ao MongoDB usando certificados de cliente.

Plugin Oracle O Zabbix agent 2 Oracle plugin agora oferece suporte a conexões criptografadas com bancos de dados Oracle usando o protocolo TCPS (TLS). Isso adiciona a capacidade de monitorar instâncias Oracle por meio de sockets protegidos por TLS, melhorando a segurança do monitoramento remoto.

Plugin Redis — suporte a TLS e validação no momento da inicialização O suporte a TLS foi adicionado ao plugin Redis do Zabbix agent 2.

Foi implementada a validação da configuração de TLS do plugin no momento da inicialização, e as mensagens de validação/erro foram aprimoradas. Uma lógica de configuração inválida (por exemplo: usar o tipo de conexão `verify_full` sem especificar `TLSCAFile`) pode impedir que o Zabbix agent 2 seja iniciado.

Modo de execução de teste para plugins carregáveis Plugins carregáveis agora podem ser iniciados no modo de teste usando o parâmetro `-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.

Descoberta e registro automático

Condição de grupo de proxy Agora, uma condição de grupo de proxy é compatível com ações de descoberta e de registro automático.

Descoberta de baixo nível

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 de baixo nível e ao formulário de protótipo de descoberta, exibida quando "HTTP agent" é selecionado na lista suspensa *Tipo*. Essa opção permite marcar os dados recuperados para conversão automática em JSON antes do processamento adicional.

Suporte a macros para descoberta de baixo nível aninhada Macros de descoberta de baixo nível agora são suportadas em regras de descoberta de baixo nível aninhadas, em:

- parâmetros de pré-processamento JSONPath parâmetros
- campo JSONPath para macros LLD personalizadas

Tags editáveis para triggers criados por descoberta de baixo nível Triggers criados a partir de protótipos de triggers agora podem ter tags adicionadas manualmente. As tags herdadas dos protótipos de triggers ainda são aplicadas automaticamente. As tags adicionadas manualmente podem ser modificadas em triggers descobertos e serão incluídas em arrays de tags de eventos e estarão disponíveis para recursos que reconhecem tags, como filtros, widgets do dashboard e macros de notificação.

Descoberta de dispositivos de bloco no Linux e AIX A nova chave de item `vfs.dev.get` foi adicionada para a descoberta de dispositivos de bloco no Linux e AIX. Ela pode ser usada em descoberta de baixo nível para dispositivos de bloco. A chave de item `vfs.dev.discovery` continua disponível por compatibilidade.

Descoberta de interfaces de rede no Linux A nova chave de item `net.if.get` foi adicionada ao Zabbix agent e ao Zabbix agent 2 no Linux. Ela pode ser usada na descoberta em baixo nível de interfaces de rede para retornar uma lista de interfaces de rede com informações detalhadas. A chave de item `net.if.discovery` continua disponível por compatibilidade.

Descoberta leve de sistemas de arquivos O item `vfs.fs.get` do Zabbix agent e do Zabbix agent 2 agora é compatível com dois parâmetros opcionais que tornam a descoberta de sistemas de arquivos mais leve e confiável:

- **mode** - no modo `short`, o item retorna apenas a lista de sistemas de arquivos montados com suas propriedades básicas, sem estatísticas de espaço em disco e de inodes. Como os próprios sistemas de arquivos não são consultados, a descoberta não aciona a montagem automática e não é afetada por sistemas de arquivos de rede que não respondem. O modo `full` (padrão) retorna os mesmos dados de antes.
- **mountpoint** - retorna dados somente para o ponto de montagem especificado, permitindo coletar estatísticas separadamente para cada sistema de arquivos descoberto (por exemplo, com `vfs.fs.get[full,{#FSNAME}]`).

Os templates de SO agora usam essa abordagem; consulte Alterações nos templates.

Para obter mais informações, consulte vfs.fs.get e Descoberta de sistemas de arquivos montados.

Bancos de dados

Suporte para múltiplos hosts de banco de dados PostgreSQL O Zabbix agora oferece suporte à especificação de múltiplos hosts de banco de dados PostgreSQL como uma lista separada por vírgulas de valores `host:port`, aproveitando o suporte de failover de conexão do PostgreSQL. Os hosts são testados em ordem até que uma conexão de leitura e gravação seja estabelecida. Esse comportamento está disponível durante a instalação do frontend no campo *Database host* e em `DBHost` nos arquivos `zabbix_server.conf` e `zabbix_proxy.conf`.

Inicialização otimizada do server Zabbix O server Zabbix agora inicia significativamente mais rápido. Isso ocorre ao carregar os dados de histórico do banco de dados para vários itens de uma vez, em vez de um por vez. Essa melhoria se aplica a todos os tipos de banco de dados suportados pelo server Zabbix.

Desempenho otimizado do Elasticsearch O server do Zabbix agora funciona melhor com Elasticsearch. O server recupera dados de forma mais eficiente sem usar o recurso *scroll* do Elasticsearch e reutiliza conexões existentes com o Elasticsearch em vez de abrir uma nova para cada solicitação. O server também não fica mais travado ao enviar várias solicitações ao Elasticsearch ao mesmo tempo (usando o *multi-handle* do cURL).

Processos

Cache e reutilização do SNMPv3 EngineID 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 sondas e melhorando o desempenho do poller. Se um EngineID reutilizado não responder, o poller retorna para uma sonda EngineID e pode remover entradas obsoletas após alterações de interface ou falhas persistentes.

Limitação refinada de proxy durante a recuperação do cache de histórico A lógica de limitação do proxy foi refinada para melhorar a estabilidade do server durante a recuperação do cache de histórico. Quando o uso do cache de histórico atinge o limite de limitação, o server continua a deixar de aceitar dados do proxy, como antes. Quando o uso do cache cai para 60%, o server começa a processar a lista de limitação, mas ainda pode rejeitar envios do proxy que contenham lotes muito grandes (aproximadamente mais de 10 mil registros) até que a pressão sobre o 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 zabbix_get e zabbix_js O valor máximo do parâmetro `timeout` dos utilitários de linha de comando zabbix_get e zabbix_js foi aumentado para 600 segundos.

Otimização do housekeeping O housekeeping foi otimizado movendo a criação de tarefas do housekeeper para triggers do banco de dados. Quando items (incluindo regras de descoberta de baixo nível), triggers, serviços ou regras de descoberta de rede são excluídos, triggers do banco de dados (em vez de chamadas explícitas em nível de aplicação) agora preenchem a tabela `housekeeper` com tarefas de limpeza. Para mais detalhes, consulte o procedimento de housekeeping.

O housekeeper agora também remove eventos de descoberta de rede gerados a partir de hosts ou serviços descobertos que foram excluídos, bem como todos os eventos associados a problemas de triggers que foram excluídos (anteriormente, apenas os próprios problemas eram removidos; os eventos associados só eram removidos após a expiração do *Período de armazenamento de dados de trigger* do housekeeper).

Fechamento manual — tags de trigger herdadas por eventos de recuperação Eventos de recuperação criados após um fechamento manual herdam tags de trigger além das tags de item e host. Essas tags estão presentes no array de tags do evento e estão disponíveis para macros de notificação, como `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

Cache de consultas DNS para os daemons do Zabbix O Zabbix server, o Zabbix proxy e o Zabbix agent receberam suporte para usar o resolvedor c-ares para todas as solicitações DNS, fornecendo cache de consultas DNS e failover aprimorado do resolvedor quando o Zabbix é compilado com `--with-ares`. O cache de consultas DNS requer c-ares 1.26.0 ou posterior.

Suporte à compilação para c-ares no Windows O agent do Zabbix agora pode ser compilado com o resolvedor c-ares no Microsoft Windows. O c-ares pode ser instalado via vcpkg, e a compilação do agent suporta tanto `ARES=<prefixo vcpkg>` quanto caminhos separados `ARESINCDIR`/`ARESLIBDIR` para diretórios de include e biblioteca.

Funções de trigger baseadas em contagem aprimoradas A operação das funções baseadas em contagem (count(), item_count(), changecount(), etc.) em expressões de trigger foi aprimorada, permitindo que mais valores sejam processados mais rapidamente.

Atualizações dos alvos de controle em tempo de execução A lista de alvos disponíveis de controle em tempo de execução nas mensagens de ajuda do Zabbix server e proxy e nas manpages foi atualizada para corresponder aos tipos de processo suportados.

O alvo `supervisor` foi adicionado, enquanto os alvos não suportados `discovery worker` e `preprocessing worker` foram removidos.

Segurança

Controle de acesso ao proxy e ao server A nova aba *Lista de acesso ao proxy* na configuração do grupo de usuários permite conceder acesso de grupos de usuários a proxies, individualmente ou por meio de grupos de proxies.

A nova opção *Selecionar "Server" para monitoramento e descoberta* na configuração da função de usuário permite impedir que os usuários criem hosts monitorados pelo server e por regras de descoberta de rede executadas pelo server, bem como que testem itens com o server.

Novos parâmetros `AllowKeyRegexp` e `DenyKeyRegexp` na configuração do Zabbix agent e do Zabbix agent 2 Os novos parâmetros `AllowKeyRegexp` e `DenyKeyRegexp` foram adicionados à configuração do Zabbix agent e do Zabbix agent 2. Esses parâmetros permitem definir padrões baseados em expressões regulares para uma correspondência mais precisa e flexível de chaves e seus parâmetros.

Autenticação

Certificados SAML importáveis para login único Usuários superadministradores agora podem importar certificados e chaves privadas diretamente no frontend para a configuração do SAML. Três novos campos foram adicionados em *Usuários* > *Autenticação* > *Configurações 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.

Post-login URL per user role It is now possible to enforce a specific URL after login based on the user role. This is handy for enforcing a specific page after login for provisioned users.

For more details, see User profile settings in user role configuration.

Autenticação do HashiCorp vault com o método AppRole A autenticação do HashiCorp vault agora oferece suporte ao método AppRole:

- Na configuração do server/proxy, usando os novos parâmetros `VaultAppRoleID` e `VaultAppSecretID` (consulte o exemplo).
- Na configuração do frontend, fornecendo os valores de ID da função de autenticação e ID secreto.

Widgets

Alterações na opção Fuso horário do widget de relógio A opção *Fuso horário* no formulário de configuração do widget de relógio foi movida da seção *Configuração avançada* para a seção principal de configuração do widget e renomeada para *Substituir fuso horário*.

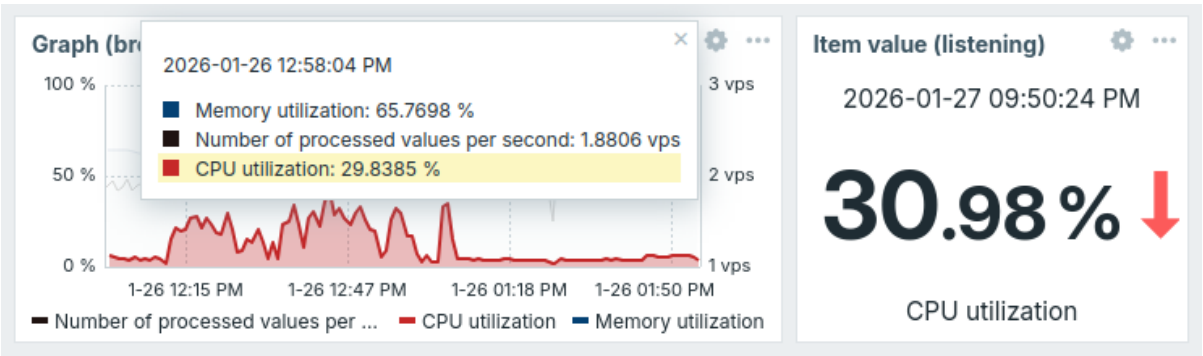
Essa opção está disponível para os tipos de horário *Hora local* e *Hora do server*.

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.

Agregar colunas em Top items O widget Top items agora pode agrupar padrões de item em uma única coluna ou linha agregada, facilitando a apresentação de métricas combinadas (por exemplo: o tráfego total de entrada em todas as interfaces de rede de um host). Três novas opções foram adicionadas ao configurar colunas de item: *Agregar colunas*, *Função de agregação de colunas* e *Nome da coluna combinada*.

Aprimoramentos no widget Graph O widget Graph oferece vários aprimoramentos de usabilidade:

- Os itens na dica de dados do gráfico, exibida ao passar o cursor sobre o gráfico, são classificados por valor em ordem decrescente.
- Ao passar o cursor sobre um item na dica, o respectivo gráfico é destacado, enquanto os demais ficam esmaecidos.
- A seleção de um item na dica transmite seus dados para outros widgets que oferecem suporte à escuta desses dados.
- O limite do número de itens exibidos na dica foi removido.
- Uma nova configuração *Host names in labels* permite escolher se os nomes dos hosts serão exibidos na dica e na legenda do gráfico.
- Uma nova configuração *Invert values* permite inverter um gráfico.



Aumento do limite de caracteres para os campos de descrição dos widgets O limite de caracteres do campo *Descrição* nos widgets Gauge e Item value foi aumentado de 2048 para 65535 caracteres.

Frontend

Dashboards importáveis Novos dashboards importáveis estão disponíveis no repositório de Dashboards:

- Dashboard de custos multinuvem - um dashboard para monitorar custos de nuvem na AWS, Azure, Google Cloud e Oracle Cloud.
- Dashboard de máquinas virtuais multinuvem - um dashboard para monitorar máquinas virtuais em execução na AWS, Google Cloud, Oracle Cloud e Azure.

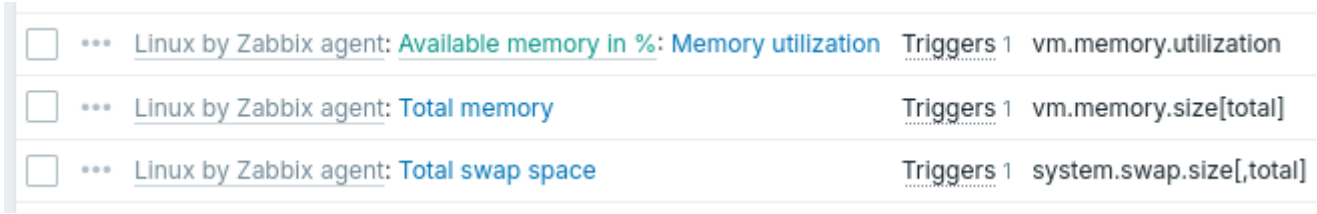
Esses dashboards foram projetados para funcionar com integrações existentes do Zabbix, combinando dados de vários hosts e alvos de monitoramento em um dashboard global unificado.

Exportação e importação de dashboard Os dashboards agora podem ser exportados e importados, facilitando a migração de dashboards entre instâncias do Zabbix ou seu reaproveitamento em diferentes ambientes.

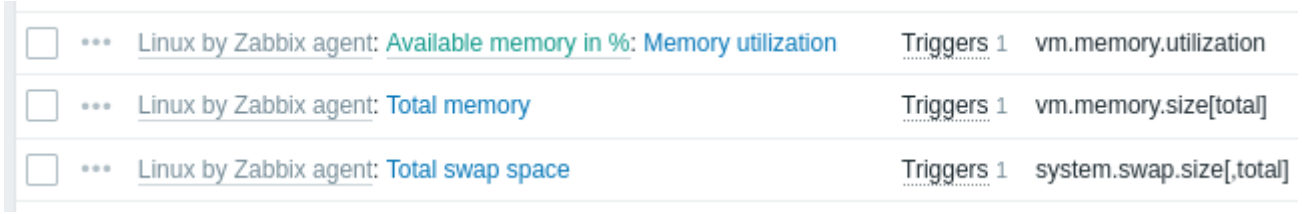
Uma nova opção *Export* permite salvar dashboards selecionados em arquivos YAML, JSON ou XML. Esses arquivos podem depois ser importados por meio da funcionalidade *Import* para recriar os dashboards com suas páginas, widgets e configuração.

Novas fontes e fontes incorporadas O Zabbix agora inclui novas fontes incorporadas que melhoram a legibilidade, carregam mais rápido e exibem idiomas adicionais no frontend com impacto mínimo no layout. Por serem incorporadas, essas fontes também garantem uma aparência consistente entre os sistemas.

Novas fontes:



Fontes legadas:

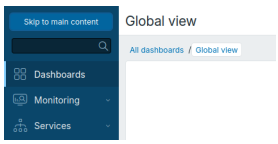


As novas fontes são usadas em quase todos os temas. Se necessário, os temas recém-adicionados *Azul (clássico)* e *Escuro (clássico)* estão disponíveis com as fontes legadas.

A família de fontes para conteúdo monoespaçado e gráficos permanece inalterada.

Melhorias na interface do usuário do Zabbix A interface do usuário do Zabbix foi modernizada para melhorar a acessibilidade e a usabilidade.

Aprimoramento de acessibilidade O link *Ir para o conteúdo principal* foi adicionado para melhorar a navegação por teclado e a acessibilidade.



O link permite que usuários de teclado e de leitores de tela pulem a navegação repetitiva da barra lateral e vão diretamente para o conteúdo principal. Ele permanece oculto durante o uso normal e aparece apenas quando recebe foco, como quando a tecla Tab é pressionada após o carregamento da página ou quando é acessado por meio de um leitor de tela.

Validação inline Os seguintes formulários no frontend foram adicionados ao conjunto de formulários que suportam validação inline:

- API token
- Audit log
- Authentication
- Autoregistration
- Connector
- Discovery prototype
- Low-level discovery rule
- Network discovery rule
- Event correlation
- Geographical maps
- Graph prototypes
- Custom graphs
- GUI
- Host group
- Host prototypes
- Housekeeping
- Icon mapping
- Images
- Maintenance
- Media type
- Notifications
- Other configuration parameters
- Profile
- Proxy
- Proxy group
- Regular expressions
- Scheduled reports
- Script
- Service
- SLA
- Template group
- Test item
- Test media type
- Timeouts
- Trigger displaying options
- Trigger expression constructor
- Update problem
- User groups
- Global user macros
- User roles
- Users

Os erros de entrada são exibidos imediatamente após o preenchimento dos campos, melhorando a usabilidade e reduzindo erros de configuração.

Formulários modais Os seguintes formulários agora são abertos em uma janela modal:

- Formulário de configuração do Discovery prototype
- Formulário de configuração da discovery rule de baixo nível
- Formulário de configuração do Host prototype
- Formulário de configuração global de expressão regular

Tags herdadas visíveis em hosts, templates, itens e triggers As tags herdadas agora são exibidas e retornadas de forma consistente em templates, hosts, itens, cenários web e triggers.
As tags herdadas de cadeias de template/host ficam visíveis em *Monitoring > Latest data*.
A filtragem por tags herdadas está disponível sempre que a filtragem por tags é compatível — incluindo as seções *Monitoring > Latest data* e *Data collection*, além de todos os widgets de dashboard que permitem filtrar por tags de host, item, trigger ou cenário web — para que a seleção e a subfiltragem baseadas em tags funcionem da mesma forma, independentemente de onde uma tag foi definida.

A aba *Tags* nos formulários de configuração de template, host e protótipo de host agora apresenta um elemento de opção para escolher como as tags serão exibidas: templates mostram *Template tags* / *Inherited and template tags*, enquanto hosts e protótipos

de host mostram *Host tags* / *Inherited and host tags*.
As tags herdadas são diferenciadas visualmente por um novo ícone de documento contornado ao lado do rótulo da tag.

Os widgets Graph e Pie chart incluem uma nova configuração *Item tags*.

Posicionamento de dicas de ferramenta As dicas de ferramenta agora podem ser reposicionadas arrastando-as para um novo local. Isso se aplica, por exemplo, à dica de ferramenta do Gráfico, bem como às dicas de ferramenta com descrições nas seções Últimos dados ou Problemas.

Nova variável $ZBX_FEATURE_FLAGS na configuração do frontend A nova variável `$ZBX_FEATURE_FLAGS[]` foi implementada na configuração do frontend para controlar o acesso à autenticação HTTP de usuários e às configurações de módulos e tipos de mídia. As seguintes flags de funcionalidade foram adicionadas ao arquivo de configuração `zabbix.conf.php`:

- `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` desabilita o acesso à configuração da autenticação HTTP de usuários (substitui a variável `$ALLOW_HTTP_AUTH`).
- `$ZBX_FEATURE_FLAGS['modules_config_enabled'] = false` desabilita o acesso à configuração de módulos.
- `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` restringe a edição de tipos de mídia que usam os métodos de entrega especificados (`'email'`, `'script'`, `'sms'` ou `'webhook'`; um ou mais valores podem ser especificados).

O acesso a `setup.php` é restrito se o arquivo de configuração `zabbix.conf.php` existir.

Detalhes do evento mostram o motivo da supressão A página de detalhes do evento, aberta em *Monitoring* > *Problems*, agora mostra se um evento foi suprimido devido à manutenção do host ou suprimido manualmente por um usuário.

Os ícones laranja  e  indicam supressão/remoção de supressão relacionada à manutenção, enquanto os ícones cinza  e  indicam supressão/remoção de supressão manual.

Nos históricos de problem e event, as ações executadas por usuários excluídos agora são exibidas como *Inaccessible user*.

Aprimoramentos no relatório de informações do sistema O relatório Informações do sistema agora exibe o ID do server (UUIDv7).
 O relatório de informações do sistema agora também está disponível para usuários *Admin* do Zabbix.

Alterações no pacote de política SELinux do Zabbix Para a versão 10 do AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux, a política SELinux é fornecida pelo novo pacote `zabbix-selinux-policy-main`, que contém a política SELinux do repositório upstream do SELinux.

O pacote `zabbix-selinux-policy` existente agora tem uma dependência de `zabbix-selinux-policy-main` ou `selinux-policy-targeted-extra`. O pacote `zabbix-selinux-policy-main` entra em conflito com `selinux-policy-targeted-ex` portanto, ambos os pacotes não podem ser instalados ao mesmo tempo.

Se você já tiver uma instalação existente de `zabbix-selinux-policy`, nenhuma alteração manual será necessária. Durante a atualização do pacote, `zabbix-selinux-policy-main` será instalado automaticamente. Se `selinux-policy-targeted-extra` já estiver instalado, nenhum pacote adicional será instalado.

Configuração da apresentação de slides do dashboard Ao criar um dashboard, a caixa de seleção *Iniciar apresentação de slides* fica desabilitada por padrão e deve ser habilitada explicitamente. Um parâmetro de URL `slideshow` aceita os valores `on` ou `off`, permitindo substituir rapidamente a configuração da apresentação de slides do dashboard. O parâmetro `slideshow=1` está obsoleto e não terá mais suporte.

Link para proxies do Zabbix Cloud A lista de proxies em *Administration* > *Proxies* agora inclui um botão *Create proxy in Cloud*. Ao clicar nele, o Zabbix Cloud é aberto em uma nova aba do navegador, onde você pode implantar um proxy sem instalá-lo na sua própria infraestrutura. Para obter mais informações, consulte Configurar um proxy do Zabbix Cloud.

Documentação

Páginas de documentação consolidadas para versões menores A documentação de lançamento para versões menores de uma versão principal do Zabbix agora será reunida em uma única página de documentação, tanto para novos recursos quanto para notas de atualização.

Notas de atualização

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
- **Outros** - todas as demais informações que descrevem as alterações na funcionalidade do Zabbix

Veja também:

- Procedimento de atualização para todas as informações relevantes sobre a atualização a partir de versões anteriores ao Zabbix 8.0.0;
- Atualizando o cluster HA para instruções sobre a atualização de servers em um cluster de **alta disponibilidade** (HA).

Alterações incompatíveis

Métodos de API de interface de host removidos das funções de usuário Após atualizar o Zabbix, os métodos de API `hostinterface.create`, `hostinterface.update`, `hostinterface.delete`, `hostinterface.massadd` e `hostinterface.massremove` serão removidos das listas *Acesso à API* das funções de usuário existentes.

Temas personalizados e tipo de mídia O Zabbix adiciona três novos temas predefinidos do frontend: *Azul (clássico)*, *Escuro (clássico)* e *Azul escuro*, além de um novo tipo de mídia Notificação push.

Se você tiver um tema personalizado com um desses nomes (`blue-classic-theme`, `dark-classic-theme`, `dark-blue-theme`) ou um tipo de mídia chamado `Notificação push`, renomeie o tema ou o tipo de mídia antes de atualizar.

Caso contrário, após a atualização:

- O tema predefinido substituirá a folha de estilos do seu tema personalizado, enquanto as cores personalizadas dos seus gráficos continuarão em uso, resultando em uma combinação dos dois temas.
- Seu tipo de mídia existente permanecerá inalterado, mas o novo tipo de mídia não será criado.

Temas e tipos de mídia personalizados com outros nomes não são afetados pela atualização.

Itens internos de proxy restritos a hosts monitorados pelo server Os itens internos `zabbix[proxy,*]` e `zabbix[proxy group,*]` não são mais compatíveis com hosts monitorados por um proxy ou grupo de proxies.

Suporte descontinuado para agents Zabbix anteriores à versão 1.6 Devido à remoção do suporte aos protocolos legados de texto simples e XML do trapper do Zabbix server/proxy, os agents Zabbix anteriores à versão 1.6 não são mais compatíveis.

Versão mínima da biblioteca Logback A versão mínima da biblioteca Logback foi elevada de 1.5.16 para 1.5.25.

Validação inline Os seguintes formulários no frontend foram adicionados ao conjunto de formulários que suportam validação inline:

- API token
- Audit log
- Authentication
- Autoregistration
- Connector
- Event correlation
- Geographical maps
- Graph prototypes
- Custom graphs
- GUI
- Host group
- Host prototypes
- Housekeeping
- Icon mapping
- Images
- Maintenance
- Media type
- Notifications
- Other configuration parameters
- Profile
- Proxy

- Proxy group
- Regular expressions
- Scheduled reports
- Script
- Service
- SLA
- Template group
- Test item
- Test media type
- Timeouts
- Trigger displaying options
- Trigger expression constructor
- Update problem
- User groups
- Global user macros
- User roles
- Users

Os erros de entrada são exibidos imediatamente após o preenchimento dos campos, melhorando a usabilidade e reduzindo erros de configuração.

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.

O esquema `tcp://` está obsoleto no plugin MongoDB, mas é mantido por compatibilidade com configurações existentes.

O parâmetro `Plugins.Oracle.CallTimeout` no arquivo de configuração do plugin Oracle não tem mais efeito. O tempo máximo de espera agora é determinado pelo item timeout.

Suporte a versões do MongoDB pelo plugin Esta versão do plugin não oferece mais suporte ao MongoDB 2.6–4.0. Ela usa o MongoDB Go Driver v2, que requer o MongoDB 4.2 ou posterior.

Para monitorar o MongoDB 2.6–4.0, use uma versão anterior do plugin. As versões anteriores oferecem suporte ao MongoDB 2.6–5.3 e podem ser usadas com o Zabbix agent 2 versão 6.0.0 ou posterior.

Versão mínima necessária do PHP A versão mínima necessária do PHP foi elevada de 8.0.0 para 8.2.0.

Lista estendida de UnsafeUserParameters O caractere % foi adicionado à lista `UnsafeUserParameters` para Zabbix agent e Zabbix agent 2.

Parâmetros de filtro movidos Como parte do recurso de tabelas personalizáveis, alguns parâmetros de filtro foram movidos da aba de filtros para as opções de coluna. Se você tiver salvo abas para filtros favoritos que utilizem algum dos seguintes parâmetros de filtro, elas serão redefinidas e deverão ser reconfiguradas manualmente após a atualização:

- *Monitoramento* > *Hosts*: *Mostrar problemas suprimidos*
- *Monitoramento* > *Dados mais recentes*: *Mostrar tags*, *Nome da tag*, *Prioridade de exibição da tag*, *Mostrar detalhes*
- *Monitoramento* > *Problemas*: *Mostrar tags*, *Nome da tag*, *Prioridade de exibição da tag*, *Mostrar dados operacionais*, *Visualização compacta*, *Mostrar detalhes*, *Mostrar linha do tempo*, *Destacar linha inteira*

Pré-processamento de JavaScript: métodos de objeto integrados agora são somente leitura Os métodos de objeto JavaScript integrados em scripts de preprocessing agora são somente leitura. Se você já modificou métodos integrados em protótipos JavaScript em scripts de pré-processamento de item, poderá encontrar problemas, pois não é mais permitido modificar métodos integrados de protótipo.

Conjunto de caracteres e collation do banco de dados MySQL Se o banco de dados MySQL do Zabbix foi criado com o conjunto de caracteres utf8mb3 e a collation utf8mb3_bin, é altamente recomendável convertê-lo para o conjunto de caracteres utf8mb4 e a collation utf8mb4_bin antes de atualizar o Zabbix, conforme descrito em Reparando o conjunto de caracteres e a collation do banco de dados Zabbix.

Outros

Assinatura SAML habilitada por padrão Para aumentar a segurança, todas as opções de assinatura SAML (*Mensagens*, *Asserções*, *Solicitações AuthN*, *Solicitações de logout* e *Respostas de logout*) agora são habilitadas por padrão ao configurar a autenticação SAML em *Usuários* > *Autenticação*. Como resultado, a chave privada do SP e o certificado do SP agora são obrigatórios para novas configurações SAML, a menos que as opções de assinatura sejam desmarcadas.

As configurações SAML existentes não são afetadas pela atualização.

Templates do Kubernetes obsoletos Os templates do Kubernetes estão obsoletos e foram substituídos pelo template *Kubernetes Cluster by HTTP*. Os templates antigos são mantidos para garantir a compatibilidade retroativa com as configurações existentes.

Macros obsoletas removidas O suporte para as seguintes macros internas foi removido:

Removida	Use em vez disso
{ACK.DATE}	{EVENT.UPDATE.DATE}
{ACK.MESSAGE}	{EVENT.UPDATE.MESSAGE}
{ACK.TIME}	{EVENT.UPDATE.TIME}
{EVENT.ACK.HISTORY}	{EVENT.UPDATE.HISTORY}
{HOSTNAME<1-9>}	{HOST.HOST}
{IPADDRESS<1-9>}	{HOST.IP}
{PROFILE.*}	{INVENTORY.*}
{TRIGGER.COMMENT}	{TRIGGER.DESCRIPTION}
{TRIGGER.KEY}	{ITEM.KEY}
{STATUS}	{TRIGGER.STATUS}
{USER.ALIAS}	{USER.USERNAME}

Parâmetros obsoletos para provedores de armazenamento de histórico Os seguintes parâmetros foram descontinuados:

- Arquivo de configuração do Zabbix server: `HistoryStorageDateIndex`, `HistoryStorageURL` e `HistoryStorageTypes`. Use o novo parâmetro `HistoryProvider`.
- Arquivo de configuração do Zabbix frontend: `$HISTORY`. Use o novo parâmetro `$HISTORY_PROVIDERS`.

Ambos os novos parâmetros permitem escolher o provedor de armazenamento de histórico: Elasticsearch, ClickHouse ou SQL (padrão). Observe que não é possível usar esses parâmetros novos e obsoletos em conjunto.

Removida a validação de JSON e XML de itens de HTTP agent A validação de JSON e XML foi removida do campo *Request body* em itens de HTTP agent. O campo não verifica mais se o valor é XML ou JSON válido, e selecionar *XML data* como o *Request body type* não exige mais a biblioteca libxml2.

Links desativados ocultos em hosts monitorados Os links desativados de *Graphs*, *Dashboards* e *Web* não são mais exibidos na lista de hosts monitorados.

Tipo de dados JSON O Zabbix agora oferece suporte a JSON como um tipo de dados para valores de item. Se você estiver usando o TimescaleDB, a nova hypertable `history_json` (usada para armazenar valores JSON) deverá ser configurada manualmente. Se você estiver usando o Elasticsearch, o valor padrão do parâmetro de configuração do Zabbix server `HistoryStorageTypes`, que foi descontinuado, agora inclui `json`.

Botão de opção removido no formulário Novo filtro de tags O botão de opção que exigia a escolha entre *Todas as tags* e *Lista de tags* ao criar um novo filtro de tags foi removido.

Valores de item exibidos como HTML usando iframes Os dados de item no widget Item history, quando exibidos como texto formatado em HTML, agora são isolados em iframes.

Verificação de TLS para templates baseados em HTTP e tipos de mídia Os templates e tipos de mídia que se conectam a endpoints HTTPS agora verificam o certificado TLS do endpoint por padrão.

Essa alteração afeta todos os templates de monitoramento de nuvem e API (por exemplo, *Claude API by HTTP*, *Cloudflare by HTTP*, *Kubernetes Cluster by HTTP*) e todos os tipos de mídia webhook (por exemplo, *Slack*, *Jira*, *Telegram*). Você pode alterar esse comportamento usando a nova macro de usuário `{$HTTP.TLS.VERIFY}`, adicionada a esses templates e tipos de mídia. Defina o valor da macro como `none` para desabilitar a verificação, `peer` para verificar apenas a cadeia de certificados e a validade, ou `full` para realizar a verificação completa. Para obter detalhes, consulte o arquivo README de cada template HTTP ou webhook.

Ao atualizar o Zabbix, os templates e tipos de mídia não são atualizados automaticamente. Isso evita substituir quaisquer alterações personalizadas que você tenha feito. Para atualizar ou adicionar templates e tipos de mídia manualmente, consulte as instruções de atualização de templates.

Compatibilidade de template com o Host Wizard Todos os templates padrão agora são compatíveis com o Host Wizard. Para atualizá-los, consulte Template upgrade.

Redimensionamento automático para campos de configuração A maioria dos campos de configuração na interface do usuário agora é redimensionada automaticamente para se ajustar ao seu conteúdo.

Variável de configuração do frontend substituída A variável `$ALLOW_HTTP_AUTH` foi substituída pela flag de recurso `$ZBX_FEATURE_FLAGS['http_auth_enabled']` no arquivo de configuração do frontend (`zabbix.conf.php`).

Processamento concorrente de verificações SNMPv1 e SNMPv2 As verificações assíncronas de serviço SNMPv1 e SNMPv2 agora são processadas simultaneamente pelos workers de descoberta disponíveis.

Alterações no pacote de política SELinux do Zabbix Para a versão 10 do AlmaLinux, CentOS Stream, RHEL, Rocky Linux e Oracle Linux, o pacote `zabbix-selinux-policy` existente agora tem uma dependência de `zabbix-selinux-policy-main` ou `selinux-policy-targeted-extra`.
 Se você já tiver uma instalação existente de `zabbix-selinux-policy`, nenhuma alteração manual será necessária. Durante a atualização do pacote, `zabbix-selinux-policy-main` será instalado automaticamente. Se `selinux-policy-targeted-extra` já estiver instalado, nenhum pacote adicional será instalado.

APM support in Zabbix proxy packages To use application performance monitoring after upgrading Zabbix proxy, choose the option for your operating system:

- RHEL 10 and its derivatives - install the EPEL repository, enable the CRB repository, and install the Zabbix proxy APM package (see details).
- Debian 12 or later, Raspberry Pi OS (Raspbian) 12 or later, Ubuntu 24.04 or later - upgrade Zabbix proxy packages as usual; APM is included.
- Other systems - compile Zabbix proxy from sources with the `--with-apm` option.

Estado do período de manutenção renomeado O estado do período de maintenance foi renomeado de *Approaching* para *Upcoming*.

Parâmetro de apresentação de slides do dashboard obsoleto O suporte ao parâmetro `slideshow=1` será removido. Novos valores para o parâmetro `slideshow` foram introduzidos para iniciar e parar a apresentação de slides:

- `slideshow=on` - iniciar a apresentação de slides
- `slideshow=off` - parar a apresentação de slides

Alterações nas expressões regulares globais

- O campo *Description* foi adicionado ao formulário de configuração de expressões regulares.
- As opções de *Expression type* foram renomeadas:

Before	Now
Character string included	*Contains string*
Any character string included	*Contains any substring from list*
Character string not included	*Does not contain string*
Result is TRUE	*Matches regular expression*
Result is FALSE	*Does not match regular expression*

Acesso padrão à API Para novas instalações, o acesso à API é desabilitado por padrão para todas as funções, incluindo funções recém-criadas e funções de usuário padrão.

Para instalações existentes, o acesso à API é desabilitado por padrão somente para funções recém-criadas.

Empty API allow list In user roles, an *Allow list* with no API methods specified now denies access to all API methods. Previously, the empty list was ignored, and all API methods were allowed.

To preserve existing access, the upgrade adds the * wildcard (any method) to all user roles that have *Allow list* selected but no methods specified. After the upgrade, review these roles in *Users* > *User roles*: replace * with the methods the role actually needs, or clear the *Enabled* checkbox in the *Access to API* section if the role does not need API access.

Novidades e notas de atualização: 8.0.x

Esta página apresenta todos os novos recursos incluídos nas versões secundárias da versão principal do Zabbix. As alterações nos templates e as notas de atualização também estão incluídas.

Consulte também Novidades da versão principal.

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

Veja as notas de atualização desta versão.

Notas de atualização

Appliance do Zabbix

Visão geral O appliance do Zabbix oferece 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.

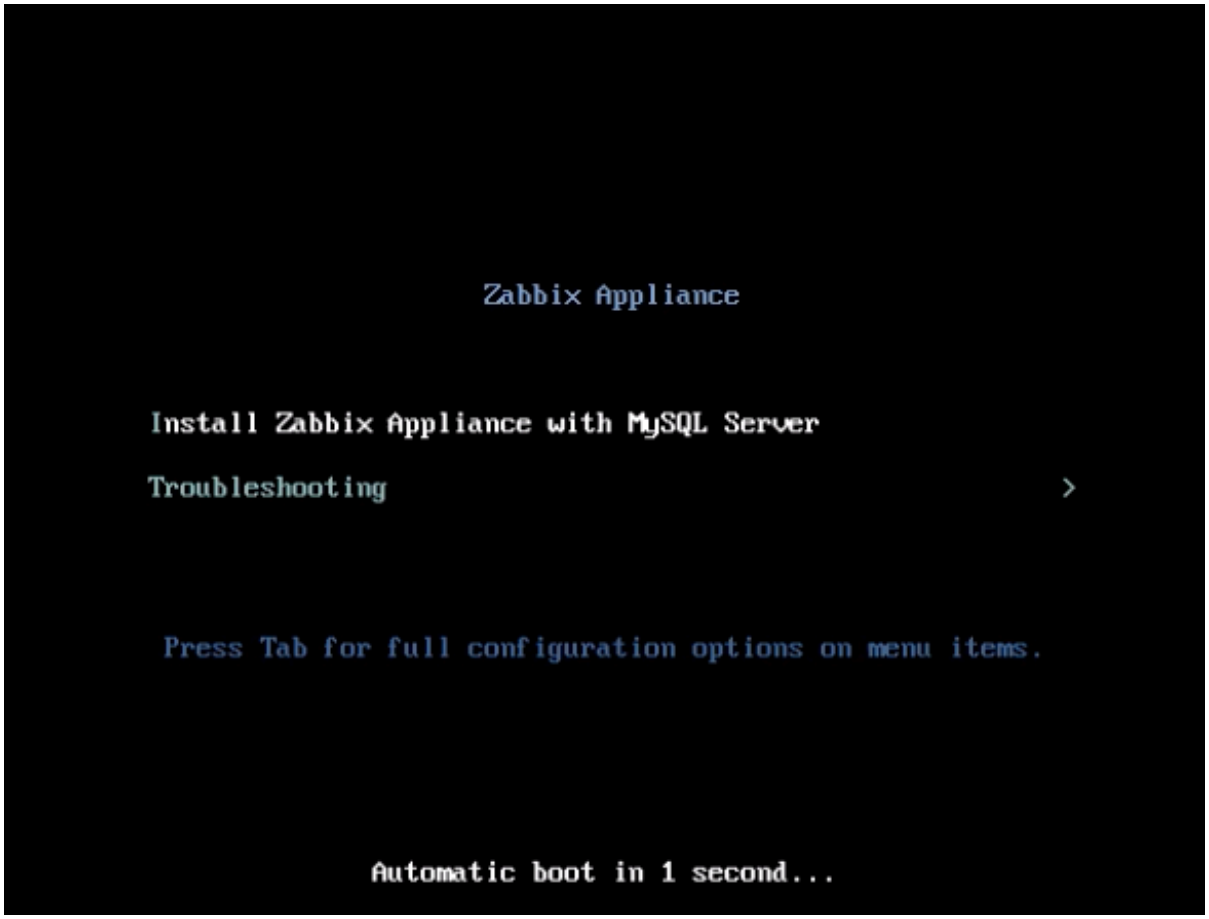
O appliance do Zabbix 8.0 é baseado em derivados do Red Hat Enterprise Linux 10, como AlmaLinux ou Rocky Linux, e contém um Zabbix server pré-configurado executado no MySQL e o frontend executado em um web server NGINX.

> **Attention:**
> Este appliance foi projetado para avaliar o Zabbix. Não é recomendável utilizá-lo em ambientes de produção críticos.

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

- CD/DVD de instalação (.iso)
- VMware (.vmx) - consulte as observações
- Formato de virtualização aberta (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - consulte as observações
- KVM, Parallels, QEMU, pendrive USB, VirtualBox, Xen (.raw) - consulte as observações
- 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. Para instruções, consulte a documentação do seu software de virtualização, por exemplo, documentação do VirtualBox.
2. Configure as definições de rede da máquina virtual para permitir acesso a partir de um navegador na máquina host. Isso pode ser feito habilitando 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-o 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`

Zabbix frontend

- Nome de usuário: `Admin`
- Senha: `zabbix`

Após fazer 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 de 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 está armazenada no arquivo `/root/.my.cnf` Não é necessário inserir uma senha na conta root.

Zabbix server:

- Nome de usuário: `zabbix_srv`
- Senha: a senha está armazenada no arquivo de configuração do server

Zabbix frontend:

- Nome de usuário: `zabbix_web`
- Senha: a senha está armazenada no arquivo de configuração do frontend

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`. Depois de salvar o arquivo editado, reinicie o NGINX fazendo login 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:

1. Faça login como usuário root.
2. Execute os seguintes comandos, substituindo os valores pelo seu endereço IP personalizado:

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # Endereço IP/CIDR do appliance
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # Endereço IP do gateway
nmcli con mod eth0 ipv4.dns 8.8.8.8    # Endereço IP do servidor DNS
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

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
dns=8.8.8.8;
gateway=192.168.1.1
method=manual
```

Depois de salvar o arquivo editado, execute os comandos:

```
nmcli con reload
nmcli con up eth0
```

Para verificar se as alterações na configuração de rede foram aplicadas, execute o seguinte comando:

```
nmcli dev show eth0
```

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.

Localizações dos arquivos

- Os arquivos de configuração do Zabbix server e do agent estão localizados em `/etc/zabbix`.
- Os arquivos de log do Zabbix server e do agent estão localizados em `/var/log/zabbix`.
- O arquivo de configuração do Zabbix frontend está localizado em `/etc/zabbix/web`.

- O Zabbix frontend está localizado em `/usr/share/zabbix`.
- O diretório inicial 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 (é necessária autenticação para o download). Se você usar o VMware vCenter Converter, poderá 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:

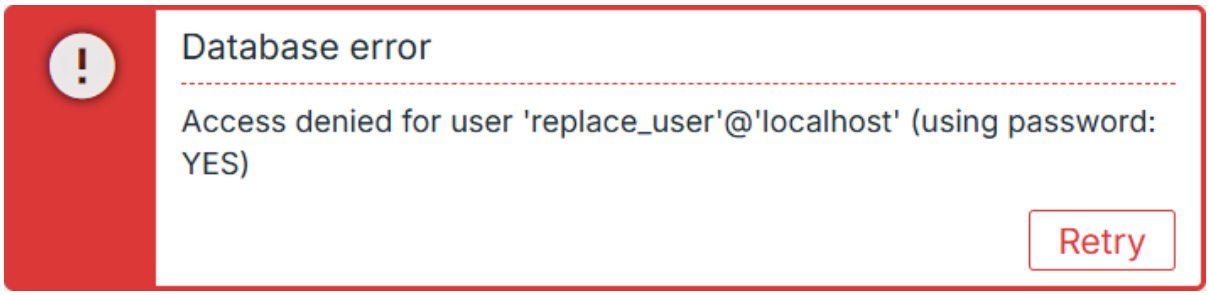
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

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 aguardar alguns minutos, ou se você observar qualquer outro comportamento inesperado, isso 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 normalmente resolve o problema.

Observe que tentar corrigir manualmente uma instalação com falha não é recomendado, pois isso pode levar a complicações adicionais.

Processos

Visão geral O Zabbix consiste em vários processos que trabalham em conjunto para monitorar sistemas e enviar notificações. Cada processo tem uma função específica na infraestrutura de monitoramento.

Processos principais

- Server - o processo central que realiza a coleta e o recebimento de dados, calcula os triggers e envia notificações aos usuários.
- Agent - coleta dados dos hosts monitorados.
- Agent 2 - um agent de nova geração com arquitetura de plugins e recursos aprimorados.
- Proxy - coleta dados de monitoramento dos dispositivos monitorados em nome do Zabbix server e envia as informações para um Zabbix server centralizado em uma configuração distribuída.
- Java gateway - permite o monitoramento de aplicações Java.
- MCP server - permite que agentes de IA trabalhem com o Zabbix usando o Model Context Protocol.

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

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

Cabeçalho 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 comprimida ou não, e se é um pacote grande ou não.

O protocolo de comunicação do Zabbix tem um limite de tamanho de pacote de 1 GB por conexão. O limite de 1 GB é aplicado tanto ao comprimento dos dados do pacote recebido quanto ao comprimento dos dados descompactados.

Ao enviar a configuração para o Zabbix proxy, o limite de tamanho do pacote é aumentado para 4 GB para permitir a sincronização de configurações grandes. Quando o comprimento dos dados antes da compressão excede 4 GB, o Zabbix server automaticamente
passa a usar o formato de pacote grande (flag 0x04), o que aumenta o limite de tamanho do pacote para 16 GB.

Observe que, embora o formato de pacote grande possa ser usado para enviar qualquer dado, atualmente apenas o sincronizador de configuração do Zabbix proxy
pode processar pacotes maiores que 1 GB.

Estrutura do cabeçalho O cabeçalho consiste em quatro campos. Todos os números no cabeçalho são formatados em little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" ou 5A 42 58 44
<FLAGS>	1	1	Flags do protocolo: 0x01 - protocolo de comunicação do 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 de cabeçalho Aqui estão alguns trechos de código que mostram como adicionar o cabeçalho do protocolo Zabbix aos dados que você deseja enviar, a fim de obter o pacote que você deve enviar ao Zabbix para que ele seja interpretado corretamente. Esses trechos de código assumem que os dados não são maiores que 1 GB; 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"
```

server

Visão geral O Zabbix server é o processo central do software Zabbix.

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

O server é o repositório central no qual todos os dados de configuração, 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 Zabbix server básico é dividido em três componentes distintos; eles são: Zabbix server, web frontend e armazenamento de banco de dados.

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

Veja também:

- Configuração de alta disponibilidade para Zabbix servers
- Parâmetros de configuração: Zabbix server

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, talvez seja necessário executar:

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

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

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

Iniciar manualmente

Se o procedimento acima não funcionar, você precisará 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 server Zabbix:

Parameter	Description
`-c --config <config-file>`	Caminho para o arquivo de configuração.
`-f --foreground`	Executa o server Zabbix em primeiro plano.
`-R --runtime-control <option>`	Executa funções administrativas.
`-T --test-config`	Valida o arquivo de configuração e encerra.
`-h --help`	Exibe informações de ajuda e encerra.
`-V --version`	Exibe o número da versão.

Exemplos de execução do server Zabbix 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.	

Opção	Descrição	Destino
`history_cache_clear=<target>`	Limpa o cache de histórico do item especificado pelo ID. Afeta todos os valores do item, exceto o primeiro e o último valor.	**target** - ID do item
`diaginfo[=<section>]`	Coleta informações de diagnóstico no arquivo de log do server.	`historycache` - estatísticas do cache de histórico; `valuecache` - estatísticas do cache de valores; `preprocessing` - estatísticas do gerenciador de pré-processamento; `alerting` - estatísticas do gerenciador de alertas; `lld` - estatísticas do gerenciador de LLD; `locks` - lista de mutexes (vazia em sistemas *BSD*); `connector` - estatísticas dos connectors com a maior fila.
`ha_status`	Registra o status do cluster de alta disponibilidade (HA).	
`ha_remove_node=<target>`	Remove o nó de alta disponibilidade (HA) especificado pelo nome ou ID. Observe que nós ativos/em espera não podem ser removidos.	**target** - nome ou ID do nó (pode ser obtido executando `ha_status`).
`ha_set_failover_delay=<delay>`	Define o atraso de failover da alta disponibilidade (HA). Sufixos de tempo são compatíveis, por exemplo, 10s, 1m.	
`proxy_config_cache_reload[=<target>]`	Recarrega o cache de configuração do proxy.	**target** - lista de nomes de proxy delimitada por vírgulas. Se nenhum destino for especificado, recarrega a configuração de todos os proxies.
`secrets_reload`	Recarrega os 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 (tempo do mecanismo, inicializações do mecanismo, ID do mecanismo e credenciais) para todos os hosts. Use para forçar uma limpeza global do cache ao solucionar problemas de SNMP.	
`housekeeper_execute`	Inicia o procedimento de housekeeping. Ignorado se o procedimento de housekeeping estiver em andamento.	
`trigger_housekeeper_execute`	Inicia o procedimento de housekeeping de trigger. Ignorado se o procedimento de housekeeping de trigger estiver em andamento.	
`log_level_increase[=<target>]`	Aumenta o nível de log; afeta todos os processos se o destino não for especificado. Não compatível com sistemas *BSD*.	**tipo de processo** - todos os processos do tipo especificado (por exemplo, `poller`). Consulte todos os tipos de processo e threads do server. **tipo de processo,N** - tipo e número do processo (por exemplo, `poller,3`). **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'.
`log_level_decrease[=<target>]`	Diminui o nível de log; afeta todos os processos se o destino não for especificado. Não compatível com sistemas *BSD*.	

Opção	Descrição	Destino
`prof_enable[=<target>]`	Ativa a criação de perfil. Afeta todos os processos se o destino não for especificado. A criação de perfil ativada fornece detalhes de todos os rwlocks/mutexes por nome de função.	**tipo de processo** - todos os processos do tipo especificado (por exemplo, `history syncer`) Tipos de processo compatíveis como destinos de criação de perfil: `alerter`, `alert manager`, `availability manager`, `configuration syncer`, `discovery manager`, `escalator`, `history poller`, `history syncer`, `housekeeper`, `http poller`, `icmp pinger`, `ipmi manager`, `ipmi poller`, `java poller`, `lld manager`, `lld worker`, `odbc poller`, `poller`, `preprocessing manager`, `proxy poller`, `self-monitoring`, `service manager`, `snmp trapper`, `task manager`, `timer`, `trapper`, `unreachable poller`, `vmware collector`. **tipo de processo,N** - tipo e número do processo (por exemplo, `history syncer,1`). **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'. **scope** - `rwlock`, `mutex`, `processing` podem ser usados com o tipo e o número do processo (por exemplo, `history syncer,1,processing`) ou com todos os processos do tipo (por exemplo, `history syncer,rwlock`).
`prof_disable[=<target>]`	Desativa a criação de perfil. 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 processo compatíveis como destinos de criação de perfil: consulte `prof_enable`. **tipo de processo,N** - tipo e número do processo (por exemplo, `history syncer,1`). **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'.

Quando um tipo de processo é especificado como alvo para controle em tempo de execução, o comando se aplica apenas aos processos em execução desse tipo. Se nenhum processo do tipo especificado estiver em execução no momento, o comando retorna um erro:

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

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 de 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 do Zabbix, o Zabbix recarregará automaticamente, na maioria dos casos, as novas credenciais SNMPv3 dessa interface; use `-R snmp_cache_reload` somente se a coleta ainda falhar após a alteração das credenciais (por exemplo, devido a inconsistências de engineBoots/engineID ou dispositivos não compatíveis com RFC), ou quando for necessário 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
```

Usuário do processo

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

Se você tentar executá-lo como 'root', ele mudará para um usuário 'zabbix' codificado, que deve estar presente no 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 server do Zabbix 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 pelo mesmo usuário, o agent poderá acessar o arquivo de configuração do server e qualquer usuário com nível Admin no Zabbix poderá obter facilmente, por exemplo, a senha do banco de dados.

Arquivo de configuração

O caminho padrão para o arquivo de configuração do server depende do sistema operacional e do método de instalação do Zabbix:

- Para instalação a partir do código-fonte: `/usr/local/etc/zabbix_server.conf` ou conforme definido pelas variáveis de compilação `--sysconfdir` ou `--prefix`.
- Para instalação a partir de pacotes e appliance: `/etc/zabbix/zabbix_server.conf`.

Consulte 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 e threads do server

- `agent poller` - processo assíncrono de polling para verificações passivas com uma thread de trabalho;
- `alert manager` - gerenciador da fila de alertas;
- `alert syncer` - processo que grava alertas no banco de dados;
- `alerter` - processo para envio de notificações;
- `availability manager` - processo para atualização da disponibilidade dos hosts;
- `browser poller` - poller para verificações de itens do 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 nos nomes dos itens;
- `connector manager` - processo gerenciador de conectores;
- `connector worker` - processo para tratar solicitações do connector manager;
- `discovery manager` - processo gerenciador da descoberta de dispositivos;
- `discovery worker` - processo para tratar tarefas de descoberta do discovery manager;
- `escalator` - processo para escalonamento de ações;
- `ha manager` - processo para gerenciar a alta disponibilidade;
- `history poller` - processo para tratar verificações calculadas que exigem uma conexão com o banco de dados;
- `history syncer` - processo que grava o histórico no banco de dados;
- `housekeeper` - processo para remover dados desatualizados (histórico e tendências de itens, sessões de usuários, eventos etc.), bem como dados deixados por objetos excluídos;
- `http agent poller` - processo assíncrono de polling 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 do 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 do balanceamento de carga e da alta disponibilidade de proxies;
- `report manager`- gerenciador de tarefas agendadas de geração de relatórios;
- `report writer` - processo para gerar relatórios agendados;
- `self-monitoring` - processo para coletar estatísticas internas do server;
- `service manager` - processo para gerenciar serviços recebendo informações sobre problemas, tags de problemas e recuperação de problemas do history syncer, do task manager e do alert manager;
- `snmp poller` - processo assíncrono de polling para verificações SNMP com uma thread de trabalho (somente itens `walk[OID]` e `get[OID]`);
- `snmp trapper` - trapper para traps SNMP;
- `supervisor` - coordena a inicialização e o encerramento dos processos e threads do server e monitora sua prontidão;
- `task manager` - processo para execução remota de tarefas solicitadas por outros componentes (por exemplo, fechar problema, reconhecer problema, verificar o valor do item agora e funcionalidade de comando remoto);
- `telemetry query poller` - poller para verificações de itens de consulta de telemetria;
- `timer` - temporizador para processar períodos de manutenção;
- `trapper` - trapper para verificações ativas, traps e comunicação com o proxy;
- `trigger housekeeper` - processo para remover problemas e eventos gerados por triggers que foram excluídos posteriormente;
- `unreachable poller` - poller para dispositivos inacessíveis;
- `vmware collector` - coletor de dados do VMware responsável por coletar dados dos serviços VMware.

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

O arquivo de log do server é criado com permissões de leitura e gravação somente para o proprietário do arquivo. Além disso, o arquivo pode ser lido pelo grupo do proprietário. Todas as outras permissões são negadas.

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

Estatísticas de transações 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 history;
- B - triggers processados devido a temporizadores.

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

- Tempo gasto gravando os valores dos items no banco de dados;
- Tempo gasto atualizando os dados do item (estado, erros, inventário do host, etc.);
- Tempo gasto enviando trends para o banco de dados;
- Tempo gasto calculando triggers;
- Tempo gasto processando eventos e ações.

Procedimento de housekeeping

O processo `housekeeper` remove periodicamente dados desatualizados (histórico e tendências de item, sessões de usuário, eventos etc.), bem como dados deixados por objetos excluídos. Ele é executado em ciclos, com a frequência e o limite de exclusão por ciclo determinados por `HousekeepingFrequency` e `MaxHousekeeperDelete`. Quaisquer dados não removidos em um ciclo são transferidos para o próximo. O housekeeping automático pode ser habilitado e configurado em *Administration* > *Housekeeping*.

Quando o processo `housekeeper` remove dados deixados por objetos excluídos, ele depende de tarefas da tabela `housekeeper`, que é preenchida sempre que um objeto é excluído. Por exemplo, quando você exclui um host, o Zabbix exclui seus items, mas não o histórico, as tendências ou os problemas. Em vez disso, triggers do banco de dados preenchem a tabela `housekeeper` com tarefas compostas pelos seguintes campos:

- `housekeeperid` - ID da tarefa
- `object` - tipo de objeto (0 - item; 1 - trigger; 2 - serviço; 3 - host descoberto; 4 - serviço descoberto)
- `objectid` - ID do objeto (ajuda o housekeeper a localizar dados relacionados ao objeto)

Por exemplo, excluir um host com dois items e uma trigger preenche a tabela `housekeeper` da seguinte forma:

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

As triggers do banco de dados preenchem a tabela `housekeeper` com tarefas, sem verificar dados relacionados ao objeto; essa verificação é feita pelo processo `housekeeper` quando ele remove os dados:

- Para items excluídos (incluindo regras de LLD) - remove seus dados de todas as tabelas de histórico e tendências (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Também remove seus eventos internos da tabela `problem`.
- Para triggers excluídas - remove seus eventos das tabelas relacionadas a eventos (`problem`, `event_symptom`, `events`; o próprio banco de dados também remove eventos da tabela `event_recovery`). Também notifica o processo `service manager` sobre os eventos removidos.

Note:
Um processo separado `trigger housekeeper` trata de uma tarefa mais restrita — remover problems e eventos que não têm uma trigger de origem conhecida. Sua frequência de execução é controlada por `ProblemHousekeepingFrequency`. Até que o procedimento de housekeeping de triggers seja iniciado, problems criados por triggers que já tenham sido excluídas ainda podem gerar problems de serviço e atribuí-los a serviços. Se sua configuração envolve muitas regras de cálculo de status de serviço com base em triggers descobertas/não descobertas com frequência, considere aumentar a frequência do procedimento de housekeeping ajustando o parâmetro de configuração do server `ProblemHousekeepingFrequency`.

- Para serviços excluídos - remove seus problems da tabela `problem`; seus eventos da tabela `events` são removidos somente após o período de armazenamento de dados configurado.
- Para hosts/serviços descobertos excluídos (network discovery) - remove seus eventos da tabela `events`.

O processo `housekeeper` remove apenas os eventos que não estão associados a problems. Por exemplo, um evento desatualizado de problem/recuperação não será removido se estiver associado a um problem aberto. Quando o `housekeeper` remove entidades desatualizadas, ele primeiro remove os problems e depois os eventos.

Note:
Tabelas que usam o modo `partition` (tabelas particionadas do TimescaleDB) são ignoradas; somente tabelas que usam o modo `regular` são processadas.

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 de forma consistente o desempenho, a tolerância a falhas e a resiliência necessários. O Zabbix opera nas versões líderes de mercado.

O Zabbix server é testado nas seguintes plataformas:

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

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

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

Configuração de alta disponibilidade para servers Zabbix

Visão geral

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

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

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



A migração para a HA do Zabbix não é definitiva. Você pode voltar à operação independente a qualquer momento.

Veja também: Detalhes de implementação

Ativando alta disponibilidade

Iniciando o Zabbix server como nó do cluster

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

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

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

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

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

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

```
zabbix_server -R ha_status
```

Esse comando de runtime registrará o status atual do cluster de HA no log do Zabbix server (e na saída padrão):

```
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 `address:port` do Zabbix server não esteja definido no arquivo de configuração do frontend do Zabbix (`zabbix.conf.php`).

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

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

Configuração do proxy

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

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

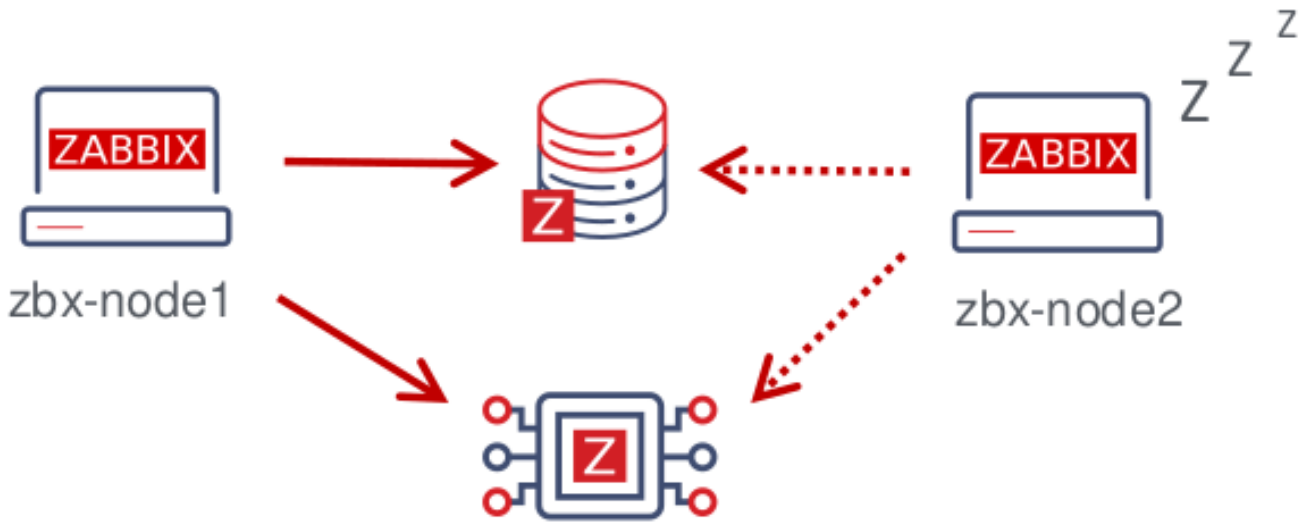
```
Server=zabbix-node-01,zabbix-node-02
```

Para um proxy ativo, os nomes dos nós devem ser listados no parâmetro Server 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 do 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 nó em standby

O Zabbix fará failover para outro nó automaticamente se o nó ativo parar. É necessário haver pelo menos um nó com status standby para que o failover aconteça.

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

- Se o nó ativo for desligado e conseguir informar seu status como "stopped", outro nó assumirá em até **5 segundos**.
- Se o nó ativo for desligado/tornar-se indisponível sem conseguir atualizar seu status, os nós em standby aguardarão o **failover delay** + 5 segundos para assumir.

O failover delay é configurável, com 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 de HA

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

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

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

Observe que os nós ativos/em espera não podem ser removidos.

- `ha_set_failover_delay=delay` - define o atraso de failover de HA (entre 10 segundos e 15 minutos; sufixos de tempo são compatíveis (por exemplo, `30s`, `1m`)).

O status do nó pode ser monitorado:

- Em *Relatórios* > *Informações do sistema*.
- No widget do painel *Informações do sistema*.
- Usando a opção de controle em tempo de execução `ha_status` do server (consulte acima).

O item interno `zabbix[cluster,discovery,nodes]` pode ser usado para a 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:

1. Faça cópias de backup dos arquivos de configuração.
2. Pare os nós em standby.
3. Remova o parâmetro HANodeName do server primário ativo.
4. Reinicie o server primário (ele será iniciado no modo autônomo).

Atualizando o cluster HA

Para realizar uma atualização de versão principal dos nós HA:

1. Pare todos os nós.
2. Crie um backup completo do banco de dados.
3. Se o banco de dados usar replicação, certifique-se de que todos os nós estejam sincronizados e sem problemas. Não faça a atualização se a replicação estiver quebrada.
4. Selecione um único nó que realizará a atualização do banco de dados, altere sua configuração para o modo standalone comentando `HANodeName` e faça a atualização dele.
5. Certifique-se de que a atualização do banco de dados foi concluída totalmente (*System information* deve exibir que o Zabbix server está em execução).
6. Reinicie o nó no modo HA.
7. Atualize e inicie o restante dos nós (não é necessário alterá-los para o modo standalone, pois o banco de dados já estará atualizado neste ponto).

Em uma atualização de versão menor, é suficiente atualizar o primeiro nó, certificar-se de que ele foi atualizado e está em execução, e então iniciar a atualização no próximo nó.

Detalhes de 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 tem requisitos específicos para os bancos de dados reconhecidos pelo Zabbix. Os usuários podem 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 execução) por vez. Um nó em standby executa apenas um processo - o gerenciador de HA. Um nó em standby não faz coleta de dados, processamento ou outras atividades regulares do server; ele não escuta em portas; ele tem o mínimo de conexões com o banco de dados.

Tanto os nós ativos quanto os em standby atualizam seu último horário de acesso a cada 5 segundos. Cada nó em standby monitora o último horário de acesso do nó ativo. Se o último horário de acesso do nó ativo ultrapassar os segundos de ”failover delay”, o nó em standby se torna o nó ativo e atribui o status `unavailable` ao nó que estava ativo anteriormente.

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 interromper todo o processamento e mudar para o modo standby. O nó ativo também monitora o status dos nós em standby - se o último horário de acesso de um nó em standby ultrapassar os segundos de ”failover delay”, o nó em standby recebe o status `unavailable`.

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

Parâmetros de configuração: Zabbix server

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix server.

Especifique os valores dos parâmetros no arquivo de configuração do server do Zabbix (`zabbix_server.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver todos os detalhes de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
AlertScriptsPath	Especifique o local dos scripts de alerta.
AllowRoot	Permita que o server seja executado como `root`.
AllowSoftwareUpdateCheck	Permita o recebimento de informações sobre atualizações de software de `zabbix.com`.
AllowUnsupportedDBVersions	Permita que o server funcione com versões de banco de dados não compatíveis.
BridgeAdapterURL	Defina a URL HTTP(S) do adaptador de bridge do Zabbix.
CacheSize	Defina o tamanho do cache de configuração (em bytes).
CacheUpdateFrequency	Defina a frequência de atualização do cache de configuração (em segundos).
DBHost	Especifique o host do banco de dados (ou o diretório do socket).
DBName	Especifique o nome do banco de dados. **Obrigatório**.
DBPassword	Especifique a senha do banco de dados.
DBPort	Especifique a porta do host do banco de dados.
DBSchema	Especifique o nome do esquema do banco de dados. Usado para PostgreSQL.
DBSocket	Especifique o caminho para o arquivo de socket do MySQL.
DBUser	Especifique o usuário do banco de dados.
DBTLSConnect	Especifique um valor para exigir uma conexão TLS com o banco de dados.
DBTLSCAFile	Especifique o nome de caminho completo do arquivo que contém os certificados de CA raiz para verificação do certificado do banco de dados.
DBTLSCertFile	Especifique o nome de caminho completo do arquivo que contém o certificado do Zabbix server para autenticação no banco de dados.
DBTLSKeyFile	Especifique o nome de caminho completo do arquivo que contém a chave privada para autenticação no banco de dados.
DBTLSCipher	Especifique a lista de cifras de criptografia permitidas pelo Zabbix server para protocolos TLS até a versão TLS 1.2. Compatível somente com MySQL.

Parâmetro	Descrição
DBTLSCipher13	Especifique a lista de conjuntos de cifras de criptografia permitidos pelo Zabbix server para o protocolo TLS 1.3. Compatível somente com MySQL a partir da versão 8.0.16.
DebugLevel	Defina o nível de depuração.
EnableGlobalScripts	Habilite scripts globais no Zabbix server.
EnableMobileDevices	Habilite o suporte a dispositivos móveis no Zabbix server.
ExportDir	Especifique um diretório para habilitar a exportação em tempo real de eventos, histórico e tendências.
ExportFileSize	Defina o tamanho máximo do arquivo de exportação (em bytes).
ExportType	Especifique os tipos de entidades para exportação em tempo real (eventos, histórico, tendências).
ExternalScripts	Especifique o local dos scripts externos.
Fping6Location	Especifique o local do fping6.
FpingLocation	Especifique o local do fping.
FrontendAllowedIP	Especifique os endereços IP ou intervalos CIDR autorizados a se conectar a partir do frontend.
HANodeName	Especifique o nome do nó em um cluster de alta disponibilidade.
HistoryCacheSize	Defina o tamanho do cache de histórico.
HistoryIndexCacheSize	Defina o tamanho do cache de índices de histórico.
HistoryProvider	Defina o provedor de armazenamento de histórico (Elasticsearch, ClickHouse ou SQL).
HistoryStorageDateIndex	Este parâmetro está **obsoleto** (use HistoryProvider). Habilite o armazenamento de valores de histórico em índices diferentes com base na data.
HistoryStorageURL	Este parâmetro está **obsoleto** (use HistoryProvider). Especifique a URL do armazenamento de histórico.
HistoryStorageTypes	Este parâmetro está **obsoleto** (use HistoryProvider). Especifique os tipos de valores a serem enviados ao armazenamento de histórico.
HousekeepingFrequency	Defina a frequência do procedimento de housekeeping (em horas).
Include	Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
JavaGateway	Especifique o endereço IP (ou nome do host) do Java gateway do Zabbix.
JavaGatewayPort	Especifique a porta do Java gateway do Zabbix.
ListenBacklog	Defina o número máximo de conexões pendentes na fila TCP.
ListenIP	Especifique o endereço IP de escuta do trapper.
ListenPort	Especifique a porta de escuta do trapper.
LoadModule	Especifique o módulo a ser carregado na inicialização do server.
LoadModulePath	Especifique o caminho completo para o local dos módulos do server.
LogFile	Especifique o arquivo de log do server.
LogFileSize	Defina o tamanho máximo do arquivo de log.
LogSlowQueries	Defina a duração máxima que uma consulta ao banco de dados pode levar antes de ser registrada (em milissegundos).
LogType	Especifique o tipo de saída do log.
MaxConcurrentChecksPerPoller	Defina o número máximo de verificações assíncronas que podem ser executadas simultaneamente por cada poller de HTTP agent, poller de agent ou poller SNMP.
MaxHousekeeperDelete	Defina o número máximo de linhas a serem excluídas por tarefa em um ciclo de housekeeping.
NodeAddress	Especifique o IP ou nome do host, com porta opcional, para substituir a forma como o frontend se conecta ao server.
PidFile	Especifique o arquivo PID.
ProblemHousekeepingFrequency	Defina a frequência de exclusão de problemas relacionados a triggers excluídos.
ProxyConfigFrequency	Defina a frequência com que o server envia dados de configuração ao Zabbix proxy.
ProxyDataFrequency	Defina a frequência com que o server solicita dados de histórico do Zabbix proxy.
ServiceManagerSyncFrequency	Defina a frequência de sincronização da configuração de um gerenciador de serviços.
SMSDevices	Especifique os arquivos de modem autorizados a usar o Zabbix server.
SNMPTrapperFile	Especifique o arquivo temporário usado para passar dados do daemon de traps SNMP para o server.
SocketDir	Especifique o diretório para armazenar sockets IPC usados pelos serviços internos do Zabbix.
SourceIP	Especifique o endereço IP de origem.
SSHKeyLocation	Especifique o local das chaves públicas e privadas para verificações e ações SSH.
SSLCertLocation	Especifique o local dos arquivos de certificado do cliente SSL para autenticação do cliente.
SSLKeyLocation	Especifique o local dos arquivos de chave privada SSL para autenticação do cliente.
SSLCALocation	Especifique o local dos arquivos da autoridade certificadora (CA) para verificação do certificado do server SSL.
StartAgentPollers	Defina o número de instâncias pré-fork de pollers assíncronos de Zabbix agent.
StartAlerters	Defina o número de instâncias pré-fork de alerters.
StartBrowserPollers	Defina o número de instâncias pré-fork de pollers de itens de navegador.

Parâmetro	Descrição
StartConnectors	Defina o número de instâncias pré-fork de workers de conectores.
StartDBSyncers	Defina o número de instâncias pré-fork de sincronizadores de histórico.
StartDiscoverers	Defina o número de instâncias pré-fork de workers de descoberta.
StartEscalators	Defina o número de instâncias pré-fork de escalators.
StartHistoryPollers	Defina o número de instâncias pré-fork de pollers de histórico.
StartHTTPAgentPollers	Defina o número de instâncias pré-fork de pollers assíncronos de HTTP agent.
StartHTTPPollers	Defina o número de instâncias pré-fork de pollers HTTP.
StartIPMIPollers	Defina o número de instâncias pré-fork de pollers IPMI.
StartJavaPollers	Defina o número de instâncias pré-fork de pollers Java.
StartLLDProcessors	Defina o número de instâncias pré-fork de workers de descoberta de baixo nível (LLD).
StartODBCPollers	Defina o número de instâncias pré-fork de pollers ODBC.
StartPingers	Defina o número de instâncias pré-fork de pingers ICMP.
StartPollersUnreachable	Defina o número de instâncias pré-fork de pollers para hosts inacessíveis (incluindo IPMI e Java).
StartPollers	Defina o número de instâncias pré-fork de pollers.
StartPreprocessors	Defina o número de instâncias pré-iniciadas de workers de pré-processamento.
StartProxyPollers	Defina o número de instâncias pré-fork de pollers para proxies passivos.
StartReportWriters	Defina o número de instâncias pré-fork de geradores de relatórios.
StartSNMPPollers	Defina o número de instâncias pré-fork de pollers SNMP assíncronos.
StartSNMPTrapper	Defina como ”1” para iniciar um processo de trapper SNMP.
StartTelemetryQueryPollers	Defina o número de instâncias pré-fork de pollers assíncronos de consultas de telemetria.
StartTimers	Defina o número de instâncias pré-fork de timers.
StartTrappers	Defina o número de instâncias pré-fork de trappers.
StartVMwareCollectors	Defina o número de instâncias pré-fork de coletores VMware.
StatsAllowedIP	Especifique os endereços IP ou nomes DNS de instâncias externas do Zabbix autorizadas a solicitar estatísticas.
TelemetryProvider	Defina a fonte de dados de telemetria.
Timeout	Especifique quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o Zabbix proxy, agent, serviço web e para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`).
TLSCAFile	Especifique o nome de caminho completo do arquivo que contém os certificados de CA raiz para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especifique o nome de caminho completo do arquivo que contém o certificado ou a cadeia de certificados do server, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	Especifique a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherAll13	Especifique a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherCert	Especifique a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherCert13	Especifique a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherPSK	Especifique a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSCipherPSK13	Especifique a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSCRLFile	Especifique o nome de caminho completo do arquivo que contém certificados revogados. Este parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.
TLSFrontendAccept	Especifique o nível de criptografia para conexões recebidas do frontend.
TLSFrontendCertIssuer	Especifique o emissor de certificado permitido para o frontend.
TLSFrontendCertSubject	Especifique o assunto de certificado permitido para o frontend.
TLSKeyFile	Especifique o nome de caminho completo do arquivo que contém a chave privada do server, usada nas comunicações criptografadas entre componentes do Zabbix.
TLSListen	Exija conexões somente TLS no socket do trapper.
TmpDir	Especifique o diretório temporário.
TrapperTimeout	Especifique o tempo limite do trapper, em segundos, para: - recuperação de dados históricos do Zabbix proxy; - envio de dados de configuração ao Zabbix proxy; - execução de scripts globais ou execução de comandos remotos no Zabbix server.

Parâmetro	Descrição
TrendCacheSize	Defina o tamanho do cache de tendências.
TrendFunctionCacheSize	Defina o tamanho do cache de funções de tendências.
UnavailableDelay	Defina a frequência de verificação da disponibilidade do host durante o período de indisponibilidade.
UnreachableDelay	Defina a frequência de verificação da disponibilidade do host durante o período de inacessibilidade.
UnreachablePeriod	Defina após quantos segundos de inacessibilidade o host deve ser considerado indisponível.
User	Remova os privilégios para um usuário específico e existente no sistema.
ValueCacheSize	Defina o tamanho do cache de valores de histórico.
Vault	Especifique o provedor do vault.
VaultAppRoleID	Especifique o ID do AppRole do vault HashiCorp.
VaultAppSecretID	Especifique o ID secreto do AppRole do vault HashiCorp.
VaultDBPath	Especifique o local para recuperar credenciais do banco de dados por chaves.
VaultPrefix	Especifique um prefixo personalizado para o caminho ou a consulta do vault.
VaultTLSCertFile	Especifique o nome do arquivo de certificado SSL usado para autenticação do cliente.
VaultTLSKeyFile	Especifique o nome do arquivo de chave privada SSL usado para autenticação do cliente.
VaultToken	Especifique o token de autenticação do vault HashiCorp.
VaultURL	Especifique a URL do server do vault.
VMwareCacheSize	Defina o tamanho da memória compartilhada para armazenar dados do VMware.
VMwareFrequency	Defina a frequência de coleta de dados de um único serviço VMware, em segundos.
VMwarePerfFrequency	Defina a frequência de recuperação das estatísticas dos contadores de desempenho de um único serviço VMware, em segundos.
VMwareTimeout	Defina o número máximo de segundos que um `vmware collector` aguardará por uma resposta do serviço VMware.
WebDriverURL	Especifique a URL da interface do WebDriver.
WebServiceURL	Especifique a URL do serviço web no formato `http[s]://host:port/report`.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores dos arquivos de configuração fornecidos.
- Os valores aceitam variáveis de ambiente.
- O Zabbix aceita arquivos de configuração somente na codificação UTF-8, sem BOM.
- Comentários iniciados com # são aceitos somente no início da linha.

Warning:
Um número excessivo de processos pode esgotar o limite de descritores de arquivo por processo do gerenciador de pré-processamento.

Esses processos são a coleta de dados realizada por pollers, pollers de hosts inacessíveis, pollers ODBC, pollers HTTP, pollers Java, pingers, trappers e proxypollers, juntamente com o gerenciador IPMI, o trapper SNMP, os workers de pré-processamento e os workers de descoberta. O esgotamento do limite de descritores de arquivo fará com que o Zabbix server **pare**, normalmente pouco depois da inicialização, mas às vezes somente após um período mais longo.

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, verifique e ajuste os limites do sistema para garantir que o limite de descritores de arquivo esteja definido com um valor suficientemente alto.

Detalhes do parâmetro

AlertScriptsPath

Especifique o local do alert script. Depende da variável de instalação em tempo de compilação `datadir`.

Padrão: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Permite que o server seja executado como `root`. Se estiver desativado e o server for iniciado por `root`, o server tentará alternar para o usuário `zabbix`. Não tem efeito se for iniciado por um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

AllowSoftwareUpdateCheck

Permite que a interface do Zabbix receba informações sobre atualizações de software de `zabbix.com`.

Padrão: 1
 Valores: 0 - não permitir; 1 - permitir

AllowUnsupportedDBVersions

Permite que o server funcione com versões de banco de dados não suportadas.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

BridgeAdapterURL

Defina a URL HTTP(S) do adaptador de bridge do Zabbix.

Exemplo:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Define 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

Defina a frequência de atualização do cache de configuração (em segundos). Veja também as opções de runtime control.

Padrão: 10
 Intervalo: 1-3600

DBHost

Especifique o host do banco de dados (ou o diretório do socket).

Para MySQL:

- `localhost` ou uma string vazia usa o socket UNIX de domínio padrão.

Para PostgreSQL:

- `localhost` é resolvido via DNS (normalmente para 127.0.0.1).
- Uma string vazia usa o socket UNIX de domínio padrão.
- Um caminho (por exemplo, `/var/run/pgbouncer`) resulta no uso do socket UNIX nesse caminho.
- Uma lista separada por vírgulas pode conter vários valores `host:port`. Cada host é tentado em ordem até que uma conexão de leitura e gravação seja estabelecida.

Exemplo: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]`

Padrão: `localhost`

DBName

Especifique o nome do banco de dados.

Obrigatório: **Sim**

DBPassword

Especifique a senha do banco de dados. Comente esta linha se nenhuma senha for usada.

DBPort

Especifique a porta do host do banco de dados do Zabbix (veja `DBHost`).
 `DBSocket` e `DBPort` são mutuamente exclusivos na configuração do server. Especifique apenas um, ou deixe ambos indefinidos.

Padrão para MySQL: 3306

Padrão para PostgreSQL: 5432

Intervalo: 1024-65535

DBSchema

Especifique o nome do esquema do banco de dados. Usado para PostgreSQL.

DBSocket

Especifique o caminho para o arquivo de socket do MySQL.
 `DBSocket` e `DBPort` são mutuamente exclusivos na configuração do server. Especifique apenas um, ou deixe ambos indefinidos.

DBUser

Especifique o usuário do banco de dados.

DBTLSConnect

Especifique um valor para impor 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 nenhum valor e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da CA de nível superior para a verificação do certificado do banco de dados.

Obrigatório: não (sim, se DBTLSConnect estiver definido como `verify_ca` ou `verify_full`)

DBTLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do server Zabbix para autenticação no banco de dados.

DBTLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada para autenticação no banco de dados.

DBTLSCipher

Especifique a lista de cifras de criptografia que o Zabbix server permite para protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.

DBTLSCipher13

Especifique a lista de suites 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

Defina o nível de depuração: - *0* - informações básicas sobre a inicialização 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).
 Veja também as opções de controle em tempo de execução.

Padrão: 3
 Intervalo: 0-5

EnableGlobalScripts

Habilita scripts globais no Zabbix server.
 Observação: a execução de scripts globais está habilitada por padrão. No entanto, para novas instalações a partir do Zabbix 7.0, EnableGlobalScripts é definido explicitamente como 0 (desabilitado).

Padrão: 1
 Valores: 0 - desabilitar; 1 - habilitar

EnableMobileDevices

Habilita o suporte a dispositivos móveis no Zabbix server.

Quando habilitado, o Zabbix server processará solicitações para adicionar ou remover dispositivos móveis e enviará notificações push para esses dispositivos.

Padrão: 1
 Valores: 0 - desabilitar; 1 - habilitar

ExportDir

Especifique um diretório para habilitar a exportação em tempo real de eventos, histórico e tendências.

ExportFileSize

Define o tamanho máximo do arquivo de exportação (em bytes). Usado para rotação se `ExportDir` estiver definido.

Padrão: 1G
 Intervalo: 1M-1G

ExportType

Especifique os tipos de entidade para exportação em tempo real (events, history, trends). Uma lista separada por vírgulas. Todos os tipos são exportados por padrão. Válido somente se `ExportDir` estiver definido.
 Observe que, se `ExportType` for especificado, mas `ExportDir` não for, isso será um erro de configuração e o server não iniciará.

Exemplo para exportação de history e trends:

```
ExportType=history,trends
```

Exemplo para exportação somente de events:

```
ExportType=events
```

ExternalScripts

Especifique o local do script externo. Depende da variável de instalação em tempo de compilação `datadir`.

Padrão: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Especifique o local do fping6. Certifique-se de que o binário fping6 tenha propriedade de root e o sinalizador SUID definido. Deixe em branco (`Fping6Location=`) se o utilitário fping for capaz de processar endereços IPv6.

Padrão: `/usr/sbin/fping6`

FpingLocation

Especifique o local do fping. Certifique-se de que o binário do fping tenha propriedade de root e o sinalizador SUID definido.

Padrão: `/usr/sbin/fping`

FrontendAllowedIP

Especifique endereços IP ou intervalos CIDR que têm permissão para se conectar a partir do frontend. Uma lista separada por vírgulas. 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

Especifique o nome do nó em um cluster de alta disponibilidade. Quando estiver vazio, o server estará operando em modo autônomo e um nó com nome vazio será criado.

HistoryCacheSize

Defina o tamanho do cache de histórico (em bytes). O tamanho da memória compartilhada para armazenar dados de histórico.

Padrão: `16M<br>` Intervalo: 128K-16G

HistoryIndexCacheSize

Defina o tamanho do cache de índice de histórico (em bytes). O tamanho da memória compartilhada para indexar os dados de histórico armazenados no cache de histórico. O tamanho do cache de índice precisa de aproximadamente 100 bytes para armazenar em cache um item.

Padrão: `4M<br>` Intervalo: 128K-16G

HistoryProvider

Defina o provedor de armazenamento de histórico usando o formato `<provider>[;<options>]`, onde:

- `<provider>` - um dos provedores de armazenamento de histórico suportados: `sql` (padrão), `elasticsearch` ou `clickhouse`.
- `<options>` - pares `key=value` separados por vírgula. Os valores podem estar entre aspas.

Opções para todos os provedores de armazenamento de histórico:

- `value_types` - define os tipos de valor a serem armazenados (`uint`, `dbl`, `str`, `log`, `text`, `json`). Cada um pode ser usado uma vez e apenas para um único provedor.
- `precache` - define 1 (padrão) para pré-carregar em massa os dados do provedor de armazenamento de histórico para o cache de valores. Isso pode melhorar significativamente o aquecimento do cache de valores após a reinicialização.

Opções para Elasticsearch e ClickHouse:

- `url` - define a URL HTTP ou HTTPS usada para conectar ao provedor de armazenamento de histórico.
- `log_slow_queries` - define a duração máxima (em milissegundos) que uma consulta ao provedor de armazenamento de histórico pode levar antes de ser registrada no log. Esta opção passa a ser habilitada a partir de `DebugLevel=3`.
- `source_ip` - define o endereço IP de origem usado para conexões de saída com o provedor de armazenamento de histórico.

Opções para Elasticsearch:

- `date_index` - define 1 para habilitar o armazenamento de valores de histórico em índices diferentes com base na data.

Opções para ClickHouse:

- `db` - define o nome do banco de dados.
- `username` - define o nome de usuário usado para fazer login no banco de dados.
- `password` - define a senha usada para fazer login no banco de dados.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - defina essas opções para configurar uma conexão segura (TLS/SSL) com o banco de dados.

Exemplo para ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Exemplo para Elasticsearch:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

Exemplo para ClickHouse e Elasticsearch:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Este parâmetro está obsoleto (use HistoryProvider em vez disso).
 Ative o armazenamento de valores de histórico em índices diferentes com base na data. Este parâmetro é usado na configuração do Elasticsearch.

Padrão: 0
 Valores: 0 - desativado; 1 - ativado

HistoryStorageURL

Este parâmetro está obsoleto (use HistoryProvider em vez disso).
 Especifique a URL de armazenamento do histórico. Este parâmetro é usado para a configuração do Elasticsearch.

HistoryStorageTypes

Este parâmetro está obsoleto (use HistoryProvider em vez disso).
 Especifique os tipos de valor a serem enviados para o armazenamento de histórico. Uma lista delimitada por vírgulas. Este parâmetro é usado na configuração do Elasticsearch.

Padrão: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Define a frequência do 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 history e trend são reduzidos significativamente), em um único ciclo de housekeeping são excluídas, para cada item, no máximo 4 vezes `HousekeepingFrequency` horas de informações desatualizadas. Assim, se `HousekeepingFrequency` for 1, serão excluídas por ciclo no máximo 4 horas de informações desatualizadas (a partir da entrada mais antiga).
Nota: Para reduzir a carga na inicialização do server, o housekeeping é adiado por 30 minutos após a inicialização do server. Assim, se `HousekeepingFrequency` for 1, o primeiro procedimento de housekeeping após a inicialização do server será executado após 30 minutos e, depois disso, será repetido com um atraso de uma hora.
É possível desativar o housekeeping automático definindo `HousekeepingFrequency` como 0. Nesse 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 menor que 4 horas e não maior que 4 dias.
Consulte também as opções e detalhes de runtime control e o procedimento de housekeeping.

Padrão: 1
 Intervalo: 0-24

Include

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você altere isso no momento da compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_server.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê, portanto considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ter permissão de leitura.
- Cada arquivo deve ter sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivo:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir apenas os arquivos relevantes de um diretório, use o curinga (`*`) para correspondência de padrão.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Especifique o endereço IP ou o nome do host do Zabbix Java gateway. Necessário apenas se os Java pollers estiverem iniciados.

JavaGatewayPort

Especifique a porta do Zabbix Java gateway.

Padrão: 10052
 Intervalo: 1024-32767

ListenBacklog

Define o número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado também depende do sistema; valores muito altos podem ser truncados silenciosamente para o ”máximo especificado pela implementação”.

Padrão: SOMAXCONN
 Intervalo: 0 - INT_MAX

ListenIP

Especifique o endereço IP de escuta do trapper. Uma lista delimitada por vírgulas.
 O trapper escutará em todas as interfaces de rede se este parâmetro não for especificado.

Padrão: 0.0.0.0

ListenPort

Especifique a porta de escuta do trapper.

Padrão: 10051
 Intervalo: 1024-32767

LoadModule

Especifique 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 precedente for absoluto (começar com /), LoadModulePath será ignorado.
 Formatos:

- LoadModule=<module.so>
- LoadModule=<path/module.so>
- LoadModule=</abs_path/module.so>

Vários parâmetros LoadModule podem ser especificados.

LoadModulePath

Especifique o caminho completo para o local dos módulos do server. O padrão depende das opções de compilação.

LogFile

Especifique o arquivo de log do server.

Obrigatório: Sim, se LogType estiver definido como file; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa 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 reiniciado.

Padrão: 1
 Intervalo: 0-1024
 Obrigatório: Sim, se LogType estiver definido como file; caso contrário, não

LogSlowQueries

Define a duração máxima que uma consulta ao banco de dados pode levar antes de ser registrada no log (em milissegundos).
0 - não registrar consultas lentas.
Esta opção passa a ser habilitada a partir de DebugLevel=3.

Padrão: 0
 Intervalo: 0-3600000

LogType

Especifique 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

Define o número máximo de verificações assíncronas que podem ser executadas ao mesmo tempo por cada HTTP agent poller, agent poller ou SNMP poller. Veja StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Padrão: 1000
 Intervalo: 1-1000

MaxHousekeeperDelete

Define o número máximo de linhas a serem excluídas por tarefa em um ciclo de housekeeping nas tabelas `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` e `problem`.
As tarefas de housekeeping existem na tabela `housekeeper` no formato: `[housekeeperid]`, `[object]`, `[objectid]`.
 Este parâmetro se aplica apenas à exclusão de dados deixados para trás por items já excluídos.
 Se definido como 0, então nenhum limite é usado.
 Nesse caso, você deve saber o que está fazendo, para não sobrecarregar o banco de dados.
 Quando muitos items são excluídos, isso aumenta a carga no banco de dados, porque o housekeeper precisará remover todos os dados de histórico que esses items tinham.
 Por exemplo, se precisarmos remover apenas 1 item prototype do template, mas esse template estiver vinculado a 50 hosts e, para cada host, o prototype for expandido para 100 items reais, será necessário remover 5000 items no total (1*50*100).
 Se 500 estiver definido para `MaxHousekeeperDelete` (`MaxHousekeeperDelete=500`), o processo de housekeeper terá que remover até 2500000 valores (5000*500) dos items excluídos das tabelas history e trends em um ciclo.

Veja também detalhes sobre o procedimento de housekeeping.

Padrão: 5000
 Intervalo: 0-1000000

NodeAddress

Especifique o IP ou hostname com porta opcional para substituir a forma como o frontend se conecta ao server.
Formato: `<address>[:<port>]`

Se IP ou hostname não estiver definido, será usado o valor de ListenIP. Se `ListenIP` não estiver definido, será usado o valor `localhost`.
Se a porta não estiver definida, será usado o valor de `ListenPort`. Se `ListenPort` não estiver definido, será usado o valor 10051.

Esta opção pode ser substituída pelo endereço especificado na configuração do frontend.

Veja também: parâmetro HANodeName; Habilitando alta disponibilidade.

Padrão: `localhost:10051`

PidFile

Especifique o arquivo PID.

Padrão: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Define a frequência de exclusão de problems para triggers excluídos (em segundos).
 Veja também detalhes sobre o procedimento de housekeeping.

Padrão: 60
 Intervalo: 1-3600

ProxyConfigFrequency

Define a frequência de envio dos dados de configuração para o Zabbix proxy pelo server. Usado apenas para proxies no modo passivo.

Padrão: 10
 Intervalo: 1-604800

ProxyDataFrequency

Define a frequência de solicitação de dados de histórico do Zabbix proxy pelo server. Usado apenas para proxies no modo passivo.

Padrão: 1
 Intervalo: 1-3600

ServiceManagerSyncFrequency

Define a frequência de sincronização da configuração de um service manager (em segundos).

Padrão: 60
 Intervalo: 1-3600

SMSDevices

Especifique os arquivos de modem que têm permissão para uso pelo Zabbix server. Uma lista delimitada por vírgulas.
 O envio de SMS não é possível se este parâmetro não estiver definido.

Exemplo:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Especifique o arquivo temporário usado para passar dados do daemon de traps SNMP para o server.
Deve ser o mesmo definido em `zabbix_trap_receiver.pl` ou no arquivo de configuração do SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

Especifique o diretório para armazenar os sockets IPC usados pelos serviços internos do Zabbix.

Padrão: `/tmp`

SourceIP

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix proxy e o Zabbix agent
- conexões sem agent (VMware, SSH, JMX, SNMP, Telnet e verificações simples)
- conexões do HTTP agent
- solicitações HTTP JavaScript do item script
- solicitações HTTP JavaScript de pré-processamento
- envio de e-mails de notificação (conexões com o servidor SMTP)
- notificações de webhook (conexões HTTP JavaScript)
- conexões com o Vault

SSHKeyLocation

Especifique o local das chaves pública e privada para verificações e ações SSH.

SSLCertLocation

Especifique o local dos arquivos de certificado do cliente SSL para autenticação do cliente.
 Este parâmetro é usado no monitoramento web, na comunicação com o Vault, para o provedor de armazenamento de histórico ClickHouse e para a fonte de dados de telemetria ClickHouse.

SSLKeyLocation

Especifique o local dos arquivos de chave privada SSL para autenticação do cliente.
 Este parâmetro é usado no monitoramento web, na comunicação com o Vault, para o provedor de armazenamento de histórico ClickHouse e para a fonte de dados de telemetria ClickHouse.

SSLCALocation

Especifique o local dos arquivos da autoridade de certificação (CA) para a verificação de certificados do servidor SSL. Se não for definido, será usado o diretório de todo o sistema.
 Observe que o valor deste parâmetro será definido como a opção `CURLOPT_CAPATH` da libcurl. Para versões da libcurl anteriores à 7.42.0, isso só terá efeito se a libcurl tiver sido compilada para usar OpenSSL. Para obter mais informações, consulte a página da web do cURL.
 Este parâmetro é usado no monitoramento web, na autenticação SMTP, em itens do HTTP agent, na comunicação com o Vault, para o provedor de armazenamento de histórico ClickHouse e para a fonte de dados de telemetria ClickHouse.

StartAgentPollers

Define o número de instâncias pré-forked do Zabbix agent pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartAlerters

Define o número de instâncias pré-forked de alerters.

Padrão: 3
 Intervalo: 1-100

StartBrowserPollers

Define o número de instâncias pré-forked de pollers de item de navegador.

Padrão: 1
 Intervalo: 0-1000

StartConnectors

Define o número de instâncias pré-forkadas de connector workers. O processo gerenciador de connector é iniciado automaticamente quando um connector worker é iniciado.

Padrão: 0
 Intervalo: 0-1000

StartDBSyncers

Define o número de instâncias pré-forked de history syncers.
Nota: Tenha cuidado ao alterar esse valor; aumentá-lo pode causar mais prejuízo do que benefício. Em geral, o valor padrão deve ser suficiente para lidar com até 4000 NVPS.

Padrão: 4
 Intervalo: 1-100

StartDiscoverers

Define o número de instâncias pré-forkadas de discovery workers[1].

Padrão: 5
 Intervalo: 0-1000

StartEscalators

Define o número de instâncias pré-forked de escalators.

Padrão: 1
 Intervalo: 1-100

StartHistoryPollers

Define o número de instâncias pré-forkadas de history pollers.
Necessário apenas para verificações calculadas.

Padrão: 5
 Intervalo: 0-1000

StartHTTPAgentPollers

Define o número de instâncias pré-forked de HTTP agent pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartHTTPPollers

Define o número de instâncias pré-forkadas de HTTP pollers[1].

Padrão: 1
 Intervalo: 0-1000

StartIPMIPollers

Define o número de instâncias pré-forked de IPMI pollers.

Padrão: 0
 Intervalo: 0-1000

StartJavaPollers

Define o número de instâncias pré-forked de Java pollers[1].

Padrão: 0
 Intervalo: 0-1000

StartLLDProcessors

Define o número de instâncias pré-forkadas de workers de descoberta de baixo nível (LLD)[1].
 O processo gerenciador de LLD é iniciado automaticamente quando um worker de LLD é iniciado.

Padrão: 2
 Intervalo: 1-100

StartODBCPollers

Define o número de instâncias pré-forked de ODBC pollers[1].

Padrão: 1
 Intervalo: 0-1000

StartPingers

Define o número de instâncias pré-forked de ICMP pingers[1].

Padrão: 1
 Intervalo: 0-1000

StartPollersUnreachable

Define o número de instâncias pré-forked de pollers para hosts inacessíveis (incluindo IPMI e Java)[1].
 Pelo menos um poller para hosts inacessíveis deve estar em execução se pollers regulares, IPMI ou Java forem iniciados.

Padrão: 1
 Intervalo: 0-1000

StartPollers

Define o número de instâncias pré-forkadas de pollers[1].

Padrão: 5
 Intervalo: 0-1000

StartPreprocessors

Defina o número de instâncias pré-iniciadas de threads de worker[1] de pré-processamento para não ser menor que a quantidade 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

Define o número de instâncias pré-forked de pollers for passive proxies[1].

Padrão: 1
 Intervalo: 0-250

StartReportWriters

Define o número de instâncias pré-forked de report writers.
Se definido como 0, a geração agendada de relatórios é desativada.
O processo do gerenciador de relatórios é iniciado automaticamente quando um report writer é iniciado.

Padrão: 0
 Intervalo: 0-100

StartSNMPPollers

Define o número de instâncias pré-forked de SNMP pollers. Veja `MaxConcurrentChecksPerPoller`.

Padrão: 1
 Intervalo: 0-1000

StartSNMPTrapper

Defina como 1 para iniciar um processo de SNMP trapper.

Padrão: 0
 Intervalo: 0-1

StartTelemetryQueryPollers

Define o número de instâncias pré-criadas de pollers de consultas assíncronas de Telemetry. Consulte MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartTimers

Define o número de instâncias pré-forked de timers.
O processo de timers trata dos períodos de manutenção.

Padrão: 1
 Intervalo: 1-1000

StartTrappers

Define o número de instâncias pré-forked de trappers[1].
 Trappers aceitam conexões de entrada do Zabbix sender, agents ativos e proxies ativos.

Padrão: 5
 Intervalo: 0-1000

StartVMwareCollectors

Define o número de instâncias pré-forked de VMware collector.

Padrão: 0
 Intervalo: 0-250

StatsAllowedIP

Especifique endereços IP (opcionalmente em notação CIDR) ou nomes DNS de instâncias externas do Zabbix que têm permissão para solicitar estatísticas. Uma lista separada por vírgulas.
 Se este parâmetro não estiver 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
```

TelemetryProvider

Defina a fonte de dados de telemetria usando o formato `<provider>[;<options>]`, em que:

- `<provider>` - uma das fontes de dados de telemetria compatíveis: `clickhouse`
- `<options>` - pares `key=value` delimitados por vírgulas. Os valores podem estar entre aspas.

Opções para o ClickHouse:

- `url` - define a URL HTTP ou HTTPS usada para conectar-se à fonte de dados de telemetria.
- `db` - define o nome do banco de dados.
- `username` - define o nome de usuário usado para fazer login no banco de dados.
- `password` - define a senha usada para fazer login no banco de dados.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - defina essas opções para configurar uma conexão segura (TLS/SSL) com o banco de dados.
- `source_ip` - define o endereço IP de origem usado para conexões de saída com a fonte de dados de telemetria.
- `vault_path` - define o caminho para um segredo do Vault que contém os valores de `username` e `password`. Não use junto com as opções `username` e `password`.

Exemplo para o ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Exemplo para o ClickHouse com o Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Especifique por quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o Zabbix proxy, agent, web service e verificações SNMP legadas (número OID único ou string).

Este parâmetro define a duração de várias operações de comunicação:

- execução de comando remoto no Zabbix agent
- execução de comando SSH/Telnet
- solicitações ao Zabbix web service
- tempo limite de comunicação para a solicitação de teste de mediatype e o método `history.push`
- reagendamento de items quando uma interface IPMI se torna indisponível
- envio de resposta ao Zabbix proxy ao falhar a troca de dados devido a problemas de permissões ou criptografia
- prazo para sockets assíncronos de IPC e opções de controle em tempo de execução
- conexões JMX
- obtenção de estatísticas de um Zabbix proxy ou server remoto
- envio de respostas ao frontend do Zabbix
- solicitações DNS do poller assíncrono
- resposta para heartbeat de check ativo
- recuperação de dados (valores) do Zabbix agent a partir de agents ativos
- recuperação de dados do Zabbix sender
- resposta quando o trapper falha ao processar uma solicitação
- envio da lista de checks ativos ao Zabbix agent

Este timeout **não** será usado para os checks que tenham configurações de timeout flexível definidas no frontend (no nível global, de proxy ou por item). Por exemplo, items SNMP `walk[OID]` e `get[OID]` usam o timeout configurado no frontend; as verificações SNMP legadas ainda usam o valor de timeout do server.

Padrão: 3
 Intervalo: 1-30

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do server ou a cadeia de certificados, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCipherAll

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitua os critérios padrão de seleção da 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

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.

TLSCipherPSK

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSFrontendAccept

Especifique o nível de criptografia para conexões de entrada do frontend.

Vários valores podem ser especificados, separados por vírgula:

- unencrypted - aceita conexões sem criptografia.
- cert - aceita conexões protegidas com TLS e um certificado.

Padrão: `unencrypted`

TLSFrontendCertIssuer

Especifique o emissor do certificado permitido para o frontend.

TLSFrontendCertSubject

Especifique o subject do certificado permitido do frontend.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do server, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSListen

Exigir conexões somente TLS no socket do trapper.

Valores suportados:

- required - aceitar apenas conexões TLS

TmpDir

Especifique o diretório temporário.

Padrão: `/tmp`

TrapperTimeout

Especifique o tempo limite do trapper 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

Padrão: 300
 Intervalo: 1-300

TrendCacheSize

Defina o tamanho do cache de tendências, em bytes.
O tamanho da memória compartilhada para armazenar tendências.

Padrão: 4M
 Intervalo: 128K-16G

TrendFunctionCacheSize

Define o tamanho do cache de trend-function, em bytes.
O tamanho da memória compartilhada para armazenar em cache os dados calculados da trend-function.

Padrão: 4M
 Intervalo: 128K-2G

UnavailableDelay

Define a frequência de verificação da disponibilidade do host durante o período de indisponibilidade, em segundos.

Padrão: 60
 Intervalo: 1-3600

UnreachableDelay

Define a frequência de verificação da disponibilidade do host durante o período de inacessibilidade em segundos.

Padrão: 15
 Intervalo: 1-3600

UnreachablePeriod

Define após quantos segundos de inacessibilidade tratar o host como indisponível.

Padrão: 45
 Intervalo: 1-3600

Usuário

Reduz os privilégios para um usuário específico e existente no sistema.
Tem efeito apenas se executado como 'root' e AllowRoot estiver desabilitado.

Padrão: `zabbix`

ValueCacheSize

Defina 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 desabilita 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

Especifique 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`

VaultAppRoleID

Especifique o AppRole ID do HashiCorp vault.
 Se definido, habilita a autenticação AppRole. Não pode ser usado com VaultToken.
 Suportado apenas para HashiCorp vault.

VaultAppSecretID

Especifique o secret ID do AppRole do HashiCorp vault.
 Suportado apenas para HashiCorp vault.

VaultDBPath

Especifique o local para recuperar as credenciais do banco de dados por meio de chaves. Pode ser um caminho ou uma consulta do Vault, dependendo do Vault.

As chaves usadas para HashiCorp são `password` e `username`.

Exemplo de caminho com `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Exemplo de caminho sem `VaultPrefix`:

```
secret/zabbix/database
```

As chaves usadas para CyberArk são `Content` e `UserName`.

Exemplo:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Esta opção só pode ser usada se `DBUser` e `DBPassword` não estiverem especificados.

VaultPrefix

Especifique um prefixo personalizado para o caminho ou consulta do vault, dependendo do Vault. Os padrões mais adequados serão usados se não forem especificados.
 Observe que `data` é automaticamente acrescentado 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

Especifique 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 em branco.
 O diretório que contém este arquivo é especificado pelo parâmetro de configuração SSLCertLocation.
 Esta opção pode ser omitida, mas é recomendada para o vault CyberArkCCP.

VaultTLSKeyFile

Especifique o nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM1.
O diretório que contém esse arquivo é especificado pelo parâmetro de configuração SSLKeyLocation.
Esta opção pode ser omitida, mas é recomendada para o vault CyberArkCCP.

VaultToken

Especifique o token de autenticação do HashiCorp Vault. O token de autenticação do HashiCorp Vault deve ter sido gerado exclusivamente para o server Zabbix, com permissão somente leitura para os caminhos especificados em Vault macros 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` forem definidos ao mesmo tempo.
 Não pode ser usado com VaultAppRoleID.
 Suportado apenas para HashiCorp vault.

VaultURL

Especifique a URL do server do vault. O diretório de certificados CA do sistema será usado se SSLCALocation não for especificado.

Padrão: `https://127.0.0.1:8200`

VMwareCacheSize

Define 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 (veja Internal checks).
Observe que a memória compartilhada não é alocada se não houver instâncias de `vmware collector` configuradas para iniciar.

Padrão: 8M
 Intervalo: 256K-2G

VMwareFrequency

Defina a frequência de coleta de dados de um único serviço VMware (em segundos). Essa frequência deve ser definida como o menor intervalo de atualização de qualquer item de monitoramento VMware.

Padrão: 60
 Intervalo: 10-86400

VMwarePerfFrequency

Defina a frequência de recuperação das estatísticas dos contadores de desempenho de um único serviço VMware (em segundos).
Essa frequência deve ser definida como o menor intervalo de atualização de qualquer item de monitoramento VMware que use contadores de desempenho VMware.

Padrão: 60
 Intervalo: 10-86400

VMwareTimeout

Define o número máximo de segundos que um `vmware collector` aguardará uma resposta do serviço VMware (vCenter ou hipervisor ESX).

Padrão: 10
 Intervalo: 1-300

WebDriverURL

Especifique a URL da interface do WebDriver.

Exemplo (usado com o servidor standalone do Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Especifique a URL do serviço web no formato `http[s]://host:port/report`.

Exemplo:

```
WebServiceURL=http://localhost:10053/report
```

O esquema `http://` pode ser omitido apenas para conexões sem TLS (HTTP); com TLS configurado, `https://` deve ser usado.

Notas de rodapé

[1] Observe que muitos processos de coleta de dados (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers), juntamente com o IPMI manager, SNMP trapper, preprocessing workers e discovery workers, podem esgotar o limite de descritores de arquivo por processo para o preprocessing manager.

Warning:
Esgotar o limite de descritores de arquivo fará com que o Zabbix server pare, normalmente logo após a inicialização, mas às vezes levando mais tempo. 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 arquivo esteja definido em um valor suficientemente alto, verificando e ajustando os limites do sistema.

proxy

Visão geral O proxy do Zabbix é um processo que pode coletar dados de monitoramento de um ou mais dispositivos monitorados e enviar essas informações ao server do Zabbix, atuando 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.

A implantação de um proxy é opcional, mas pode ser muito útil para distribuir a carga de um único server do Zabbix. Se apenas os proxies coletarem dados, o processamento no server exigirá menos CPU e operações de E/S em disco.

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

O proxy do Zabbix requer um banco de dados separado.

Attention:
Observe que os bancos de dados compatíveis com o proxy do Zabbix são SQLite, MySQL e PostgreSQL.

Veja também:

- Monitoramento distribuído usando proxies
- Balanceamento de carga e alta disponibilidade do proxy
- Sincronização da configuração de monitoramento
- Protocolo de troca de dados entre server e proxy
- Parâmetros de configuração: proxy do Zabbix
- Configurar um proxy do Zabbix Cloud

Executando 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 procedimento acima não funcionar, você precisará iniciá-lo manualmente. Encontre o caminho para o binário `zabbix_proxy` e execute:

```
zabbix_proxy
```

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

Parameter	Description
`-c --config <config-file>`	Caminho para o arquivo de configuração.
`-f --foreground`	Executar o Zabbix proxy em primeiro plano.
`-R --runtime-control <option>`	Executar funções administrativas.
`-T --test-config`	Validar o arquivo de configuração e sair.
`-h --help`	Exibir informações de ajuda e sair.
`-V --version`	Exibir 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:

Option	Description	Target
`config_cache_reload`	Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O proxy Zabbix ativo se conectará ao server Zabbix e solicitará os dados de configuração. O proxy Zabbix passivo solicitará os dados de configuração ao server Zabbix na próxima vez em que o server se conectar ao proxy.	
`history_cache_clear=target`	Limpa o cache de histórico do item especificado pelo seu ID. Afeta todos os valores do item, exceto o primeiro e o último valor.	**target** - ID do item.

Option	Description	Target
`diaginfo[=<section>]`	Coleta informações de diagnóstico no arquivo de log do proxy.	`historycache` - estatísticas do cache de histórico; `preprocessing` - estatísticas do gerenciador de pré-processamento; `locks` - lista de mutexes (está vazia em sistemas *BSD*).
`snmp_cache_reload`	Recarrega o cache SNMP - limpa as propriedades do engine SNMP (engine time, engine boots, engine id, credentials) 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 no momento.	
`log_level_increase[=<target>]`	Aumenta o nível de log, afeta todos os processos se o target não for especificado. Não suportado em sistemas *BSD*.	**process type** - todos os processos do tipo especificado (por exemplo, `poller`). Veja todos os tipos de processo do proxy. **process type,N** - tipo de processo e número (por exemplo, `poller,3`) **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o target como 'process type,N'.
`log_level_decrease[=<target>]`	Diminui o nível de log, afeta todos os processos se o target não for especificado. Não suportado em sistemas *BSD*.	
`prof_enable[=<target>]`	Ativa o profiling. Afeta todos os processos se o target não for especificado. O profiling ativado fornece detalhes de todos os rwlocks/mutexes por nome de função.	**process type** - todos os processos do tipo especificado (por exemplo, `history syncer`). Veja todos os tipos de processo do proxy. **process type,N** - tipo de processo e número (por exemplo, `history syncer,1`). **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o target como 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` podem ser usados com o tipo de processo e número (por exemplo, `history syncer,1,processing`) ou com todos os processos do tipo (por exemplo, `history syncer,rwlock`).
`prof_disable[=<target>]`	Desativa o profiling. Afeta todos os processos se o target não for especificado.	**process type** - todos os processos do tipo especificado (por exemplo, `history syncer`). Veja todos os tipos de processo do proxy. **process type,N** - tipo de processo e número (por exemplo, `history syncer,1`). **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o target como 'process type,N'.

Quando um tipo de processo é especificado como target para o controle em tempo de execução, o comando se aplica apenas aos processos em execução desse tipo. Se nenhum processo do tipo especificado estiver em execução no momento, o comando retornará um erro:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

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 recarregará automaticamente as novas credenciais SNMPv3 para essa interface na maioria dos casos; use `-R snmp_cache_reload` somente se a coleta ainda falhar após a alteração das credenciais (por exemplo, devido a inconsistências de engineBoots/engineID ou dispositivos não compatíveis com RFC), ou quando for necessário 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_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"
```

Processar usuário

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

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

Arquivo de configuração

O caminho padrão para o arquivo de configuração depende do sistema operacional e do método de instalação do Zabbix:

- Para instalação a partir do código-fonte: `/usr/local/etc/zabbix_proxy.conf`, ou conforme definido pelas variáveis de tempo de compilação `--sysconfdir` ou `--prefix`.
- Para instalação a partir de pacotes: `/etc/zabbix/zabbix_proxy.conf`.

Consulte as opções do arquivo de configuração para obter detalhes sobre como configurar `zabbix_proxy`.

Tipos de processos e threads do proxy

- `agent poller` - processo assíncrono de polling para verificações passivas com uma thread de trabalho;
- `apm manager` - processo gerenciador para monitoramento do desempenho de aplicações (APM);
- `apm worker` - processo para lidar com tarefas de monitoramento do desempenho de aplicações (APM) provenientes do `apm manager`;
- `availability manager` - processo para atualizações de disponibilidade do host;
- `browser poller` - poller para verificações de item do navegador;
- `configuration syncer` - processo para gerenciar o cache na 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 provenientes do discovery manager;
- `history syncer` - gravador do banco de dados de histórico;
- `housekeeper` - processo para remover o histórico desatualizado dos itens;
- `http agent poller` - processo assíncrono de polling 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 pollers IPMI;
- `ipmi poller` - poller para verificações IPMI;
- `java poller` - poller para verificações Java;
- `odbc poller` - poller para verificações ODBC;
- `poller` - poller normal para verificações passivas;
- `preprocessing manager` - gerenciador de tarefas de pré-processamento com threads de trabalho de pré-processamento;
- `preprocessing worker` - thread para pré-processamento de dados;
- `self-monitoring` - processo para coletar estatísticas internas do server;
- `snmp poller` - processo assíncrono de polling para verificações SNMP com uma thread de trabalho (somente itens `walk[OID]` e `get[OID]`);
- `snmp trapper` - trapper para traps SNMP;
- `supervisor` - coordena a inicialização e o encerramento dos processos e threads do server e monitora sua prontidão;
- `task manager` - processo para execução remota de tarefas solicitadas por outros componentes (por exemplo, fechar problema, reconhecer problema, verificar o valor do item agora e funcionalidade de comando remoto);
- `telemetry query poller` - poller para verificações de itens de consulta de telemetria;
- `trapper` - trapper para verificações ativas, traps e comunicação com o proxy;
- `unreachable poller` - poller para dispositivos inacessíveis;
- `vmware collector` - coletor de dados VMware responsável pela coleta de dados dos serviços VMware.

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

O arquivo de log do proxy é criado com permissões de leitura e gravação somente para o proprietário do arquivo. Além disso, o arquivo pode ser lido pelo grupo do proprietário. Todas as outras permissões são negadas.

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

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

Procedimento de housekeeping

O Zabbix proxy possui o processo `housekeeper` que remove o histórico e as tendências desatualizados dos item. Ele é executado em ciclos, com a frequência determinada por `HousekeepingFrequency`, e o limite de exclusão por ciclo determinado por `ProxyLocalBuffer` e `ProxyOfflineBuffer`. Ao contrário do procedimento de housekeeping do Zabbix server, o processo `housekeeper` do proxy não usa a tabela `housekeeper` — ele exclui todos os dados desatualizados uma vez por ciclo de housekeeping.

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

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

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

Para obter um desempenho otimizado, é recomendável configurar o uso do buffer de memória no proxy. Isso pode ser feito modificando o valor de `ProxyBufferMode` de `disk` (padrão codificado 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, descarregando os dados não enviados no banco de dados se o proxy for interrompido, se o buffer estiver cheio ou se os dados estiverem muito antigos. Quando todos os valores tiverem sido descarregados no banco de dados, o proxy voltará a usar o buffer de memória.

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

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

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

Observe que, com uma configuração conflitante, o proxy exibirá um erro e não será iniciado, 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 Unix-like modernos tem uma localidade UTF-8 como padrão; no entanto, há alguns sistemas em que isso pode precisar ser definido especificamente.

Cálculo de filas durante a manutenção

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

Usando o frontend do Zabbix atrás de um proxy Se o frontend do Zabbix for executado atrás de um proxy server, o caminho do cookie no arquivo de configuração do proxy precisará ser reescrito para corresponder ao caminho submetido ao proxy reverso. Veja os exemplos abaixo. Se o caminho do cookie não for reescrito, os usuários poderão 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 o Apache

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

Monitoramento distribuído usando proxies

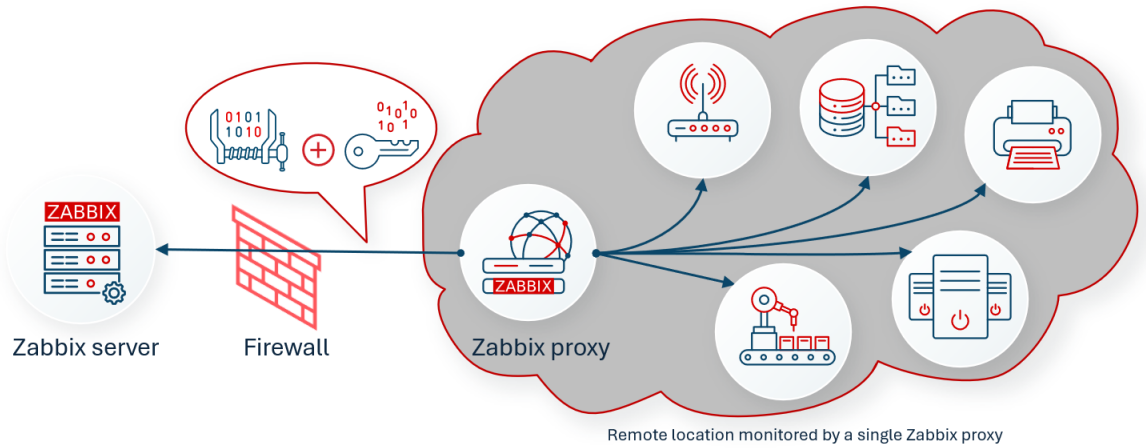
Visão geral

Os proxies do Zabbix podem ser usados para criar uma configuração de monitoramento distribuído eficaz e confiável.

Os proxies são implantados localmente para coletar dados de desempenho e disponibilidade. Em seguida, os dados são enviados a um server centralizado.

Um proxy do Zabbix pode ser usado para:

- Monitorar locais remotos
- Monitorar locais com comunicações não confiáveis
- Reduzir a carga do server do Zabbix ao monitorar milhares de dispositivos
- Coletar dados de telemetria para monitoramento do desempenho de aplicações (APM)



O proxy requer apenas uma conexão TCP com o server do Zabbix. Dessa forma, é mais fácil contornar um firewall, pois é necessário configurar apenas uma regra de firewall.

Attention:
O proxy do Zabbix deve usar um banco de dados separado. Apontá-lo para o banco de dados do server do Zabbix interromperá a configuração.

Todos os dados coletados pelo proxy são armazenados localmente antes de serem transmitidos ao server. Dessa forma, nenhum dado é perdido devido a problemas temporários de comunicação com o server. Os parâmetros `ProxyLocalBuffer` e `ProxyOfflineBuffer` no arquivo de configuração do proxy controlam por quanto tempo os dados são mantidos localmente.

Attention:
Pode acontecer de um proxy, que recebe as alterações de configuração mais recentes diretamente do banco de dados do server do Zabbix, ter uma configuração mais atualizada do que o server do Zabbix, cuja configuração pode não ser atualizada com a mesma rapidez devido ao valor de CacheUpdateFrequency. Como resultado, o proxy pode começar a coletar dados e enviá-los ao server do Zabbix, que ignora esses dados.

O proxy do Zabbix é um coletor de dados. Ele não calcula triggers, processa eventos nem envia alertas. Para obter uma visão geral das funcionalidades do proxy, consulte a tabela a seguir:

Function		Supported by proxy
Items		
	Zabbix agent checks	**Yes**
	Zabbix agent checks (active)	**Yes** [1]
	Simple checks	**Yes**
	Trapper items	**Yes**
	SNMP checks	**Yes**
	SNMP traps	**Yes**
	IPMI checks	**Yes**
	JMX checks	**Yes**
	Log file monitoring	**Yes**
	Internal checks	**Yes**
	SSH checks	**Yes**
	Telnet checks	**Yes**
	External checks	**Yes**
	Dependent items	**Yes**
	Script items	**Yes**
	Browser items	**Yes**
	Telemetry query items	**Yes**
Built-in web monitoring		**Yes**
Item value preprocessing		**Yes**
Network discovery		**Yes**
Active agent autoregistration		**Yes**
Low-level discovery		**Yes** [2]
Remote commands		**Yes**
Telemetry data collection (APM)		**Yes** [3]

Function	Supported by proxy
Calculating triggers	*No*
Processing events	*No*
Event correlation	*No*
Sending alerts	*No*

Note:
[1] Para garantir que um agent solicite ao proxy (e não ao server) as verificações ativas, o proxy deve estar listado no parâmetro **ServerActive** no arquivo de configuração do agent.
 [2] Para LLD, o proxy do Zabbix apenas coleta e pré-processa os dados e, em seguida, envia-os ao server do Zabbix para processamento adicional.
 [3] Os dados de telemetria são coletados somente se o APM estiver habilitado no proxy (consulte a aba *APM* em configuração do proxy); o server do Zabbix não pode receber dados de telemetria.

Recursos do proxy

Ao decidir usar ou não um proxy, várias considerações devem ser levadas em conta.

	Proxy
Leve	**Sim**
GUI	Não
Funciona de forma independente	**Sim**
Fácil manutenção	**Sim**
Criação automática de DB	**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 DB funciona apenas com SQLite. Outros bancos de dados suportados exigem configuração manual.

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

Proteção contra sobrecarga

Se o Zabbix server ficou indisponível por algum tempo e os proxies coletaram muitos dados, quando o server iniciar ele pode ficar sobrecarregado (o uso do history cache permanece em 95-100% por algum tempo). Essa sobrecarga pode resultar em perda de desempenho, fazendo com que as verificações sejam processadas mais lentamente do que deveriam. A proteção contra esse cenário foi implementada para evitar problemas que surgem devido à sobrecarga do history cache.

Quando o history cache do Zabbix server está cheio, o acesso de gravação ao history cache é limitado, interrompendo os processos de coleta de dados do server. O caso mais comum de sobrecarga do history cache ocorre após uma indisponibilidade do server, quando os proxies estão enviando os dados coletados. Para evitar isso, foi adicionada a limitação de proxy (atualmente ela não pode ser desativada).

Quando o uso do history cache atinge 80%, o Zabbix server entra no modo de limitação. No modo de limitação, o server aceita dados de proxy somente quando o uso do history cache está abaixo de 60%, alternando entre os proxies aceitos. Assim que o uso do history cache cai abaixo de 20%, o server volta 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 history cache exceder 60%. Essa decisão é aplicada no instante em que o server avalia um envio de proxy e, portanto, nem sempre pode ser refletida imediatamente nos gráficos de uso do history cache (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, ou caia para 20%, ou a lista de limitação fique vazia. No primeiro caso, o server deixará de aceitar dados de 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 history write cache	Modo do Zabbix server	Ação do Zabbix server
Atinge 80%	Aguarda	Para de aceitar dados de 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 envios de proxy muito grandes (mais de 10 mil registros) ao decidir se aceita os dados; continua aceitando outros dados de proxy.
Cai para 20%	Normal	Descarta a lista de limitação e começa a aceitar dados de proxy normalmente.

Você pode usar o item interno zabbix[wcache,history,pused] para correlacionar esse comportamento do Zabbix server com uma métrica.

Configuração

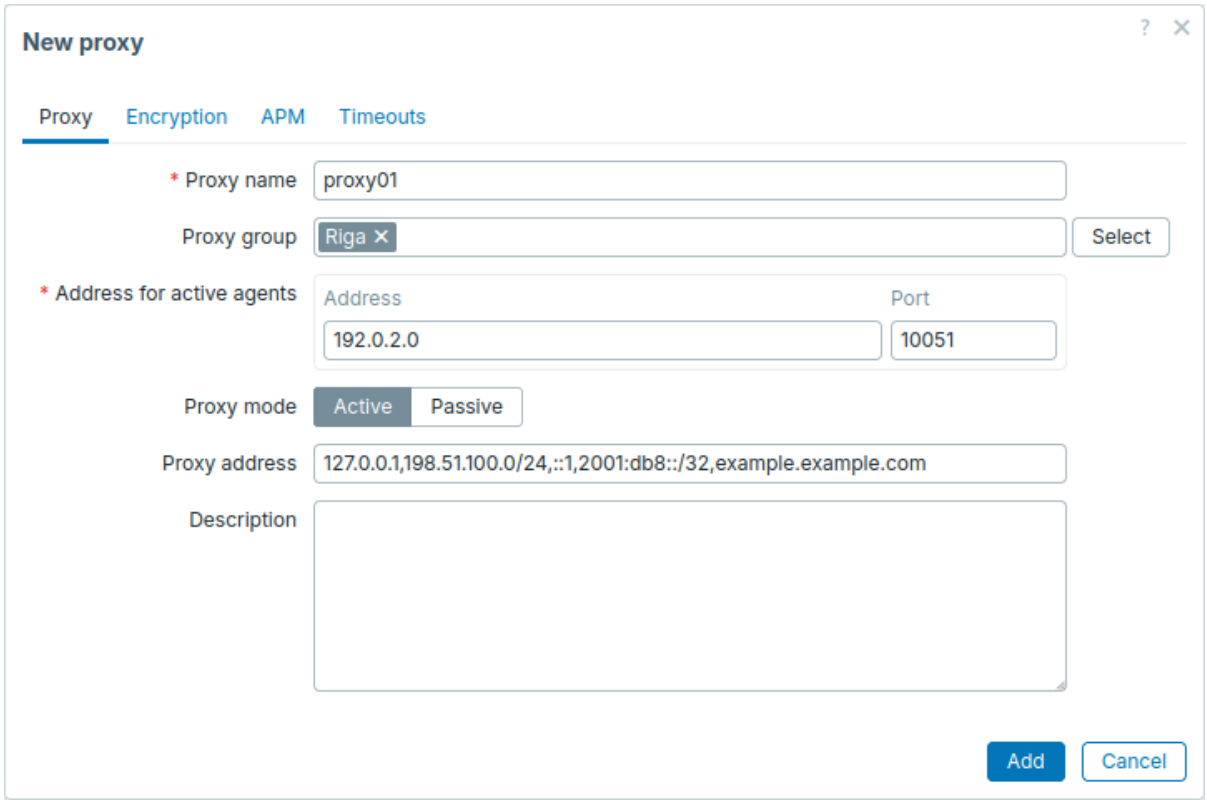
Depois de instalar e configurar um proxy, é hora de configurá-lo no frontend do Zabbix.

Adicionando proxies

Note:
Em vez de instalar um proxy em sua própria infraestrutura, você pode implantar um no Zabbix Cloud. Para isso, clique em *Create proxy in Cloud* em *Administration* > *Proxies*; o Zabbix Cloud será aberto em uma nova aba do navegador. Depois de criar um nó de proxy lá, configure o proxy no frontend do Zabbix conforme descrito abaixo. Para obter mais informações, consulte Configurar um proxy do Zabbix Cloud.

Para configurar um proxy no frontend do Zabbix:

1. Acesse *Administration* > *Proxies*.
2. Clique em *Create proxy*.
3. Insira os parâmetros do proxy no formulário.



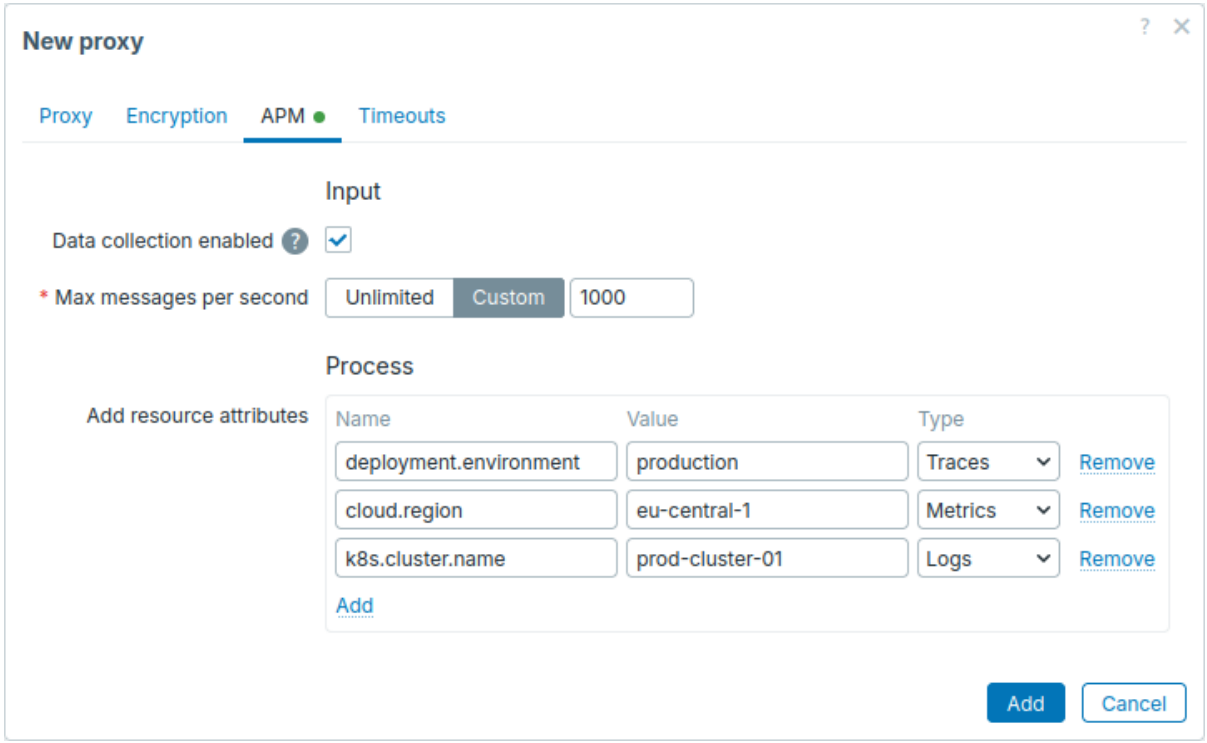
Parameter	Description
Proxy name	Insira o nome do proxy. Ele deve ser igual ao nome definido no parâmetro *Hostname* no arquivo de configuração do proxy.
Proxy group	Selecione um grupo de proxy para balanceamento de carga/alta disponibilidade do proxy.

Parameter	Description
Address for active agents	Insira o endereço ao qual os agents ativos monitorados ou os senders devem se conectar. Compatível **somente** com agents do Zabbix 7.0 ou posterior. Esse endereço é usado para conectar-se a proxies ativos e passivos. Este campo só estará disponível se um grupo de proxy for selecionado no campo *Proxy group*.
Address	Endereço IP/nome DNS ao qual se conectar.
Port	Número da porta TCP (10051 por padrão) à qual se conectar. Macros de usuário são compatíveis.
Proxy mode	Selecione o modo do proxy. **Active** - o proxy se conectará ao Zabbix server e solicitará dados de configuração **Passive** - o Zabbix server se conectará ao proxy *Observe* que, sem comunicações criptografadas (sensíveis), os dados de configuração 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 se a autenticação não ocorrer ou se os endereços do proxy não forem limitados no campo *Proxy address*.
Proxy address	Se especificado, as solicitações de proxy ativo serão aceitas somente desta lista de endereços IP separados por vírgulas, opcionalmente em notação CIDR, ou de nomes DNS do proxy ativo do Zabbix. Este campo só estará disponível se um proxy ativo for selecionado no campo *Proxy mode*. Macros não são compatíveis.
Interface	Insira os detalhes da interface de um proxy passivo. Este campo só estará disponível se um proxy passivo for selecionado no campo *Proxy mode*.
Address	Endereço IP/nome DNS do proxy passivo.
Port	Número da porta TCP do proxy passivo (10051 por padrão). Macros de usuário são compatíveis.
Description	Insira a descrição do proxy.

A aba **Encryption** permite exigir conexões criptografadas com o proxy.

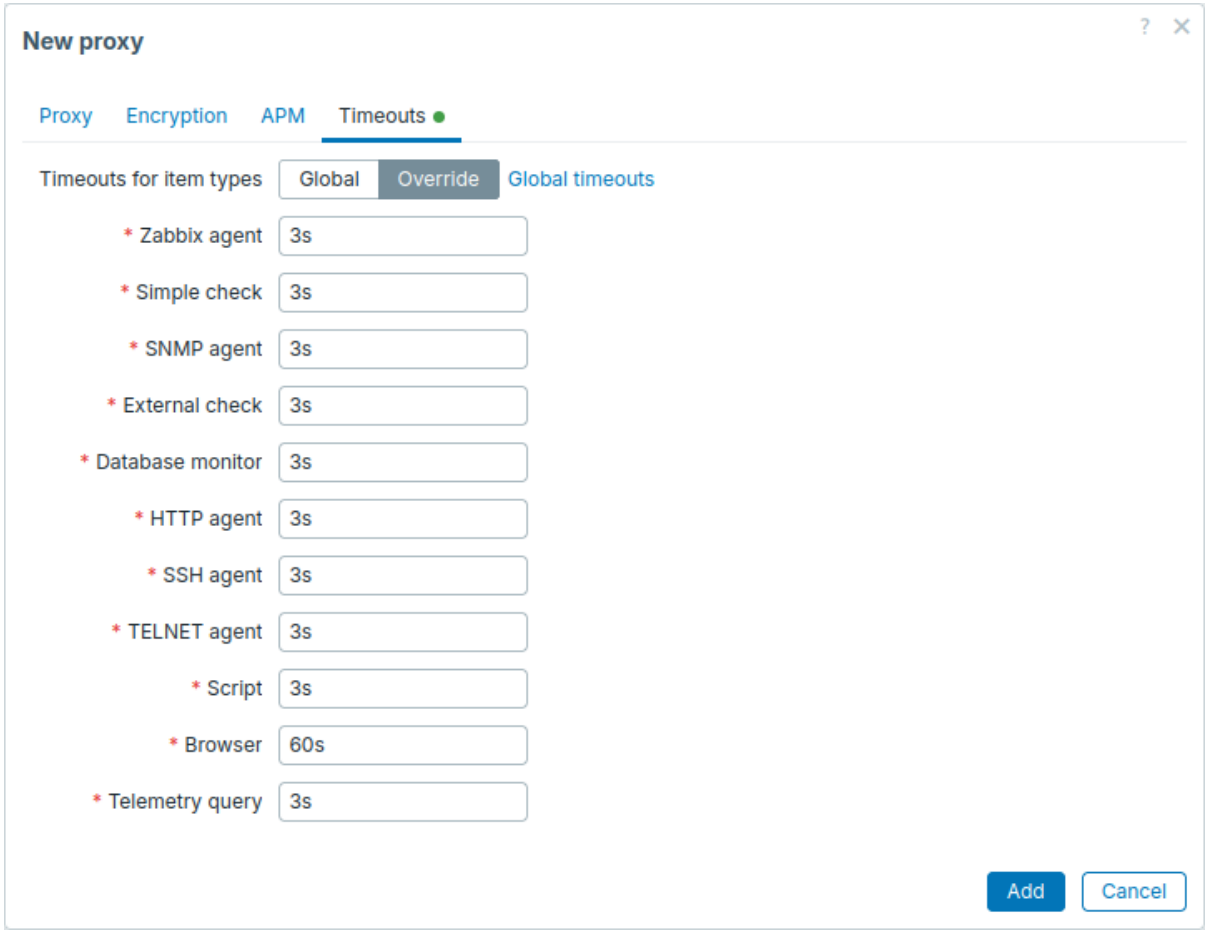
Parameter	Description
Connections to proxy	Como o server se conecta ao proxy passivo: sem criptografia (padrão), usando PSK (chave pré-compartilhada) ou certificado.
Connections from proxy	Selecione quais tipos de conexão são permitidos a partir do proxy ativo. Vários tipos de conexão podem ser selecionados ao mesmo tempo (útil para testes e para alternar para outro tipo de conexão). O padrão é ”No encryption”.
Issuer	Emissor permitido do certificado. Primeiro, o certificado é validado com a CA (autoridade certificadora). Se for válido e assinado pela CA, o campo *Issuer* poderá ser usado para restringir ainda mais as CAs permitidas. Este campo é opcional e destina-se a instalações do Zabbix que usam certificados de várias CAs.
Subject	Subject permitido do certificado. Primeiro, o certificado é validado com a CA. Se for válido e assinado pela CA, o campo *Subject* poderá ser usado para permitir apenas um valor da string *Subject*. Se este campo estiver vazio, qualquer certificado válido assinado pela CA configurada será aceito.
PSK identity	String de identidade da chave pré-compartilhada. Não inclua informações confidenciais na identidade PSK, pois ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK deve ser usada.
PSK	Chave pré-compartilhada (string hexadecimal). 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 **APM** permite configurar o monitoramento de desempenho de aplicações (APM) no proxy. O proxy pode coletar dados de telemetria (traces, métricas e logs) enviados por aplicações ou coletores OpenTelemetry por meio do protocolo OpenTelemetry (OTLP/gRPC) e gravar os dados na fonte de dados de telemetria.



Parameter	Description
Data collection enabled	Marque a caixa de seleção para habilitar a coleta de dados de telemetria no proxy. Antes de habilitar a coleta de dados de telemetria, defina o parâmetro `StartAPMCollectors` como 1 no arquivo de configuração do proxy.
Max messages per second	Defina o número máximo de mensagens por segundo que o proxy aceita: **Unlimited** - aceita todas as mensagens recebidas. **Custom** - aceita até o número especificado de mensagens por segundo. Uma mensagem é uma solicitação OTLP enviada por uma aplicação, um coletor OpenTelemetry ou outra fonte de telemetria. Seu tamanho depende do remetente; por exemplo, um coletor OpenTelemetry pode agrupar milhares de spans em uma mensagem, enquanto uma aplicação simples pode enviar mensagens menores (por exemplo, uma por trace). Se o limite for excedido, o proxy responderá com o código de status gRPC `UNAVAILABLE` (erro temporário), portanto o remetente poderá tentar enviar a mensagem novamente mais tarde, dependendo de sua configuração.
Add resource attributes	Especifique os atributos de recurso a serem adicionados aos dados de telemetria recebidos do tipo selecionado (por exemplo, para marcar todos os dados recebidos por este proxy com seu ambiente ou região). Você poderá usar esses atributos para filtrar dados de telemetria, por exemplo, nas condições de item de consulta de telemetria.
Name	Insira o nome do atributo (por exemplo, `deployment.environment`, `cloud.region`). Deve ser exclusivo para o *Type* selecionado.
Value	Insira o valor do atributo (por exemplo, `production`, `eu-central-1`). Pode ficar vazio.
Type	Selecione o tipo de dados recebidos ao qual o atributo se aplica: *Traces*, *Metrics* ou *Logs*.

A aba **Timeouts** permite substituir os tempos limite globais para tipos de item compatíveis.



Parameter	Description
Timeouts for item types	Defina o tempo limite do item (com base no tipo): **Global** - usa o tempo limite global (exibido no campo *Timeout* esmaecido de cada tipo de item); **Override** - define um tempo limite personalizado (no campo *Timeout* de cada tipo de item). Intervalo permitido: 1 - 600s (padrão: herdado dos tempos limite globais). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são compatíveis. Clicar no link *Global timeouts* permite configurar os tempos limite globais. Observe que o link *Global timeouts* fica visível somente para usuários do tipo *Super admin* com permissões para a seção do frontend *Administration* > *General*. Observe que, embora os tempos limite no nível do proxy substituam os globais, eles serão substituídos pelos tempos limite de itens individuais, se configurados.

Note:

Se a versão principal do proxy não corresponder à versão principal do server, o ícone será exibido ao lado de *Timeouts for item types*, com a mensagem ao passar o mouse: "Timeouts disabled because the proxy and server versions do not match". Nesses casos, o proxy usará o parâmetro `Timeout` do arquivo de configuração do proxy.

O formulário de edição de um proxy existente contém os seguintes botões adicionais:

- *Refresh configuration* - atualiza a configuração do proxy;
- *Clone* - cria um novo proxy com base nas propriedades do proxy existente;
- *Delete* - exclui o proxy.

Configuração do host

Você pode especificar que um host individual deve ser monitorado por um proxy ou grupo de proxy no formulário de configuração do host, usando o campo *Monitored by*.



A atualização em massa de host é outra forma de especificar que hosts devem ser monitorados por um proxy ou grupo de proxy.

Balanceamento de carga e alta disponibilidade do proxy

Visão geral

Os proxies do Zabbix podem ser organizados em grupos de proxy para permitir balanceamento de carga e alta disponibilidade dos proxies.

O balanceamento de carga e a alta disponibilidade dos proxies consistem na redistribuição automática de hosts entre proxies dentro de um grupo de proxy:

- Se um proxy ficar offline, seus hosts serão movidos para outros proxies, mantendo assim a alta disponibilidade do proxy.
- Se um proxy tiver um número de hosts muito maior/menor do que os 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 server do Zabbix. Se estiver usando Zabbix agent (passivo), a versão do proxy deve corresponder à versão do agent. Agents ativos exigem apenas Zabbix 7.0 ou posterior.
- O grupo de proxy tem um estado online.
- Os hosts estão configurados para serem monitorados por um grupo de proxy, em vez de proxies individuais.

Note:
A integridade do grupo de proxy pode ser monitorada com verificações internas por qualquer host atribuído a um grupo de proxy. 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 a alta disponibilidade de proxy são gerenciados pelo Zabbix server por meio do proxy group manager, que monitora continuamente o estado de todos os proxies em cada grupo de proxy e a distribuição de seus hosts.

A alta disponibilidade de proxy dentro de um grupo é garantida por failover de proxy: quando um proxy fica offline, seus hosts são imediatamente redistribuídos para outros proxies. O balanceamento de carga de proxy também ocorre, pois os hosts são reatribuídos aos proxies com o menor número de hosts atribuídos.

Além disso, o balanceamento de carga de proxy é acionado quando a contagem de hosts do proxy difere da média do grupo em pelo menos 10 hosts e por um fator de 2 (excesso de hosts ou déficit de hosts). Se o desequilíbrio persistir após um período de tolerância (10 x failover delay), o grupo de proxy é colocado na fila para redistribuição de hosts.

O proxy group manager redistribui os hosts usando a seguinte lógica:

1. Calcular o número médio de hosts por proxy.
2. Para proxies com excesso de hosts — mover os hosts excedentes para o pool de proxies não atribuídos.
3. Para proxies com déficit de hosts — calcular quantos hosts são necessários para atingir o equilíbrio.
4. Remover o número necessário de hosts dos proxies com mais hosts.
5. Mover os hosts não atribuídos para os proxies com menos hosts.

Exemplos de redistribuição de hosts:

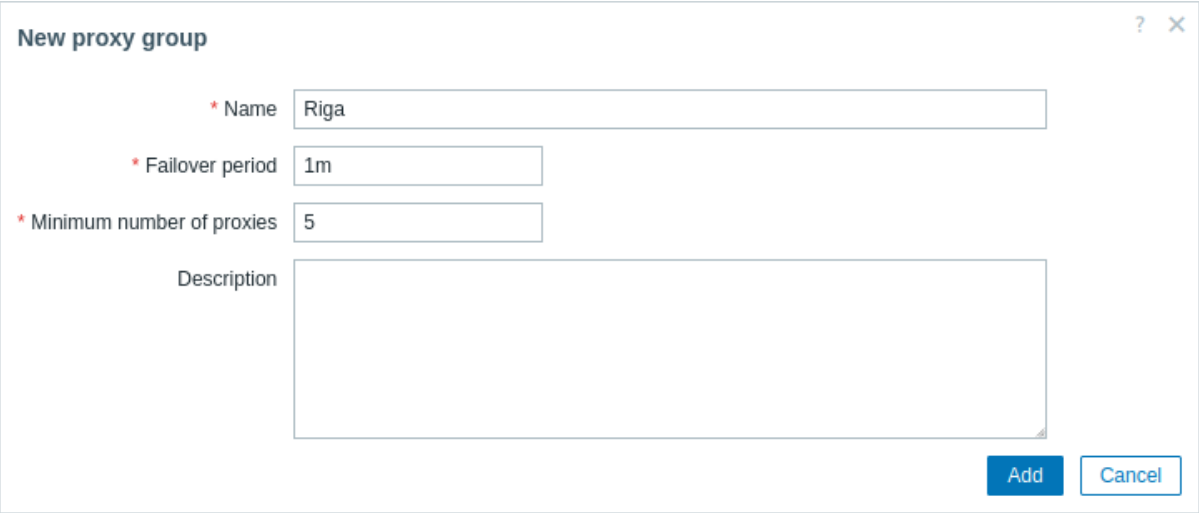
Hosts on proxy	Group average	Host reassignment
100	50	Yes
60	50	No
40	50	No
25	50	Yes
15	5	Yes
10	5	No

Attention:
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 proxy

Para configurar um grupo de proxy no frontend do Zabbix:

1. Acesse *Administração* > *Grupos de proxy*.
2. Clique em *Criar grupo de proxy*.
3. Insira os parâmetros do grupo de proxy no formulário.



Parâmetro	Descrição
Nome	Nome do grupo de proxy.
Período de failover	Período, em segundos, durante o qual um proxy no grupo de proxy deve se comunicar com o Zabbix server para ser considerado online (padrão: 1m; intervalo: 10s-15m). Se o proxy não se comunicar durante esse período, seu estado 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 proxy 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 com 10 proxies, definir o mínimo como 10 fará com que o grupo fique offline se qualquer proxy falhar. Observe que os proxies online em um grupo offline continuam funcionando normalmente, mas o balanceamento de carga/alta disponibilidade não ocorrerá.
Descrição	Descrição do grupo de proxy.
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 de proxy

Para usar o balanceamento de carga de proxy, você precisa configurar um grupo de proxy no Zabbix frontend (veja acima) e garantir que os hosts sejam monitorados por um grupo de proxy, e não por proxies individuais (você pode usar a atualização em massa de host para mover hosts de proxies para o grupo de proxy).

Se estiver usando o Zabbix agent, configure-o também da seguinte forma:

- Para verificações passivas, liste todos os proxies do grupo de proxy no parâmetro Server.
- Para verificações ativas, é recomendável listar todos os proxies do grupo de proxy ou o Zabbix server no parâmetro Server-Active. Observe que apenas o Zabbix agent 7.0 (ou posterior) funcionará com grupos de proxy no modo ativo.

Se o parâmetro `ServerActive` incluir apenas um proxy de um grupo de proxy (ou o Zabbix server), o agent ainda conseguirá se conectar ao proxy correto. Quando o serviço do agent iniciar e se conectar 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 verificações ativas serão redirecionadas para o proxy online correto para o host, com base na atribuição atual de proxy-host dentro do grupo de proxy.

Warning:
Ter apenas um único proxy especificado no parâmetro `ServerActive` do Zabbix agent 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 proxy. 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 Zabbix agent também deve conseguir se conectar a todos os proxies do grupo de proxy através do firewall. Caso contrário, as verificações ativas podem travar ou falhar durante o redirecionamento ou o failover. Por exemplo:

- Durante as verificações ativas, um proxy pode redirecionar o agent para outro proxy. Se esse proxy estiver bloqueado por um firewall, a comunicação ficará travada aguardando uma resposta.
- Em configurações estáveis de alta disponibilidade sem rebalanceamento recente, os agents talvez nunca entrem em contato com proxies de backup. Se as regras de firewall tiverem sido alteradas e não tiverem sido testadas, o failover poderá falhar.

Testando o balanceamento de carga do proxy

Para testar o balanceamento de carga do proxy:

1. Configure um grupo de proxy.
2. Certifique-se de que o grupo de proxy tenha um estado online.
3. Certifique-se de que os hosts sejam monitorados por um grupo de proxy, e não por proxies individuais (você pode usar a atualização em massa de hosts para mover hosts de proxies para o grupo de proxy).
4. Aguarde alguns segundos pela atualização da configuração e pela distribuição dos hosts entre os proxies no grupo de proxy. Observe a mudança atualizando a lista de hosts em *Administration* > *Proxies*.

Notas importantes

- SNMP traps não são suportados por proxies em um grupo de proxy.
- Verificações que dependem de configuração externa (por exemplo, scripts para external checks ou configuração ODBC para database checks) devem ter a mesma configuração em todos os proxies em um grupo de proxy.
- Database checks exigem permissões estendidas no objeto do banco de dados/server.
- Hosts VMware monitorados por um grupo de proxy 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 autoregistro de um proxy em um grupo de proxy são definidos para serem monitorados por esse grupo de proxy. No entanto, hosts criados com base em dados de descoberta de rede de um proxy em um grupo de proxy são definidos para serem monitorados por esse proxy.

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, somente as entidades modificadas são atualizadas (portanto, se nenhuma entidade tiver sido 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 as entidades com revisões maiores que a revisão da configuração do proxy são incluídas nos dados de configuração enviados ao proxy.

As entidades consideradas em uma sincronização de configuração são as seguintes:

Entidade	Detalhes
dados TLS de registro automático	Todos os dados TLS de registro automático.
expressões	Todas as expressões (expressões regulares, tabelas de expressões).
configuração global	Configuração global definida na tabela 'config'

Entidade	Detalhes
host	Todas as propriedades, interfaces, inventário, itens, pré-processamento de itens, parâmetros de itens e cenários web de um host.
macros de host	Todas as macros definidas em um host e todos os IDs de templates vinculados a ele.
regra de descoberta do proxy	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 elemento do host tiver sido alterado.

O comando `-R config_cache_reload` no proxy também iniciará uma atualização incremental.

Observe que uma sincronização completa da configuração ocorrerá quando um proxy for iniciado/reiniciado, durante um 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 durante o recebimento dos 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 será encerrado.

Cálculo de filas durante a manutenção

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

Protocolo de troca de dados entre server e proxy

Server - a troca de dados com o proxy é baseada no formato JSON.

As mensagens de solicitação e resposta devem começar com header and data length.

proxy passivo

Solicitação de configuração

O server enviará primeiro uma solicitação `proxy config` vazia. Essa solicitação é enviada a cada `ProxyConfigFrequency` segundos (parâmetro de configuração do server).

O proxy responde com a versão atual do proxy, o token da sessão e a revisão da configuração. O server responde com os dados de configuração que precisam ser atualizados.

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	Versão do proxy (<major>.<minor>.<build>).
session	*string*	Token da sessão de configuração do proxy.
config_revision	*number*	Revisão da configuração do proxy.

name	value type	description
server→proxy:		
full_sync	*number*	1 - se os dados completos da configuração forem enviados; ausente caso contrário (opcional).
data	*array*	Objeto com os dados das tabelas. Ausente se a configuração não tiver sido 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 sobre macros secretas, ausentes se não houver alterações nas macros do vault (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*	Informações sobre o 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", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
```

```
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
                [44162, 0, 0]
            ]
        },
        "item_preproc": {
            "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
            "data": []
        },
        "item_parameter": {
            "fields": ["item_parameterid", "itemid", "name", "value"],
            "data": []
        },
        "globalmacro": {
            "fields": ["globalmacroid", "macro", "value", "type"],
            "data": [
                [2, "{$SNMP_COMMUNITY}", "public", 0]
            ]
        },
        "hosts_templates": {
            "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
            "data": []
        },
        "hostmacro": {
            "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
            "data": [
                [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
            ]
        },
        "drules": {
            "fields": ["druleid", "name", "iprange", "delay"],
            "data": [
                [2, "Rede local", "127.0.0.1", "10s"]
            ]
        },
        "dchecks": {
            "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
            "data": [
                [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
            ]
        },
        "regexps": {
            "fields": ["regexpid", "name"],
            "data": [
```

```
            [1, "Sistemas de arquivos para descoberta"],
            [2, "Interfaces de rede para descoberta"],
            [3, "Dispositivos de armazenamento para descoberta via SNMP"],
            [4, "Nomes de serviços do Windows para descoberta"],
            [5, "Estados de inicialização de serviços do Windows para descoberta"]
        ]
    },
    "expressions": {
        "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
        "data": [
            [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
            [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
            [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
            [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
            [7, 2, "^Software Loopback Interface", 4, ",", 1],
            [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
            [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
            [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
            [11, 2, "^[Ss]ystem$", 4, ",", 1],
            [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
        ]
    },
    "settings": {
        "fields": ["name", "type", "value_str", "value_int"],
        "data": [
            ["autoreg_tls_accept", 2, "", 1],
            ["hk_history_global", 2, "", 0],
            ["snmptrap_logging", 2, "", 1],
            ["proxy_secrets_provider", 2, "", 0],
            ["hk_history", 1, "31d", 0],
            ["timeout_db_monitor", 1, "3s", 0],
            ["timeout_external_check", 1, "3s", 0],
            ["timeout_http_agent", 1, "3s", 0],
            ["timeout_simple_check", 1, "3s", 0],
            ["timeout_snmp_agent", 1, "3s", 0],
            ["timeout_ssh_agent", 1, "3s", 0],
            ["timeout_telnet_agent", 1, "3s", 0],
            ["timeout_zabbix_agent", 1, "3s", 0],
            ["timeout_browser", 1, "30s", 0]
        ]
    },
    "httptest": {
        "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
        "data": []
    },
    "httptestitem": {
        "fields": ["httptestitemid", "httptestid", "itemid", "type"],
        "data": []
    },
    "httptest_field": {
        "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
        "data": []
    },
    "httpstep": {
        "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
        "data": []
    },
    "httpstepitem": {
        "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
        "data": []
    },
    "httpstep_field": {
```

```
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```

proxy→server:

```
{
  "response": "success",
  "version": "8.0.0"
}
```

Solicitação de dados

A solicitação `proxy data` é usada para obter disponibilidade da interface do host, dados históricos, de descoberta e de autoregistro do proxy. Essa solicitação é enviada a cada `ProxyDataFrequency` (parâmetro de configuração do server) segundos.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Token da sessão de dados.
interface availability	*array*	*(optional)* Array de objetos 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*	*(optional)* 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 (nanosegundos).
value	*string*	*(optional)* Valor do item.
id	*number*	Identificador do valor (contador crescente, exclusivo dentro de uma sessão de dados).
timestamp	*number*	*(optional)* Timestamp de itens do tipo log.
source	*string*	*(optional)* Valor de origem do item eventlog.
severity	*number*	*(optional)* Valor de severidade do item eventlog.
eventid	*number*	*(optional)* Valor de eventid do item eventlog.
state	*string*	*(optional)* Estado do item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Último tamanho de log dos itens do tipo log.
mtime	*number*	*(optional)* Hora de modificação dos itens do tipo log.

name	value type	description
discovery data	*array*	*(optional)* 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 null para dados da regra de descoberta.
type	*number*	Tipo de verificação de descoberta: **-1** dados da regra de descoberta **0**, *SVC_SSH* - verificação de serviço SSH **1**, *SVC_LDAP* - verificação de serviço LDAP **2**, *SVC_SMTP* - verificação de serviço SMTP **3**, *SVC_FTP* - verificação de serviço FTP **4**, *SVC_HTTP* - verificação de serviço HTTP **5**, *SVC_POP* - verificação de serviço POP **6**, *SVC_NNTP* - verificação de serviço NNTP **7**, *SVC_IMAP* - verificação de serviço IMAP **8**, *SVC_TCP* - verificação de disponibilidade de porta TCP **9**, *SVC_AGENT* - agent do Zabbix **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **14**, *SVC_HTTPS* - verificação de serviço HTTPS **15**, *SVC_TELNET* - verificação de disponibilidade de Telnet
ip	*string*	Endereço IP do host.
dns	*string*	Nome DNS do host.
port	*number*	*(optional)* Número da porta do serviço.
key_	*string*	*(optional)* Chave do item para verificação de descoberta do tipo **9** *SVC_AGENT*
value	*string*	*(optional)* Valor recebido do serviço, pode estar vazio para a maioria dos serviços.
status	*number*	*(optional)* Status do serviço: **0**, *DOBJECT_STATUS_UP* - serviço UP **1**, *DOBJECT_STATUS_DOWN* - serviço DOWN
auto registration	*array*	*(optional)* Array de objetos de dados de autoregistro.
clock	*number*	Timestamp dos dados de autoregistro.
host	*string*	Nome do host.
ip	*string*	*(optional)* Endereço IP do host.
dns	*string*	*(optional)* Nome DNS resolvido a partir do endereço IP.
port	*string*	*(optional)* Porta do host.
host_metadata	*string*	*(optional)* Metadados do host enviados pelo agent (com base no parâmetro de configuração do agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(optional)* Array de tarefas.
type	*number*	Tipo de tarefa: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - resultado de comando remoto
status	*number*	Status de execução do comando remoto: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - comando remoto concluído com sucesso **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - falha no comando remoto
error	*string*	*(optional)* Mensagem de erro.
parent_taskid	*number*	ID da tarefa pai.
more	*number*	*(optional)* 1 - há mais dados históricos a serem enviados.

name	value type	description
clock	*number*	*(optional)* Timestamp da transferência de dados (segundos).
ns	*number*	*(optional)* Timestamp da transferência de dados (nanosegundos).
version	*string*	Versão do proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informação de sucesso da solicitação ('success' ou 'failed').
tasks	*array*	*(optional)* Array de tarefas.
type	*number*	Tipo de tarefa: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Hora de criação da tarefa.
ttl	*number*	Tempo em segundos após o qual a tarefa expira.
commandtype	*number*	Tipo de comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usar script personalizado **1**, *ZBX_SCRIPT_TYPE_IPMI* - usar IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usar SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usar Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usar script global (atualmente funcionalmente equivalente a script personalizado)
command	*string*	Comando remoto a ser executado.
execute_on	*number*	Destino de execução para scripts personalizados: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - executar script no agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - executar script no server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - executar script no proxy
port	*number*	*(optional)* Porta para comandos Telnet e SSH.
authtype	*number*	*(optional)* Tipo de autenticação para comandos SSH.
username	*string*	*(optional)* Nome de usuário para comandos Telnet e SSH.
password	*string*	*(optional)* Senha para comandos Telnet e SSH.
publickey	*string*	*(optional)* Chave pública para comandos SSH.
privatekey	*string*	*(optional)* 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
      }
  ]
}
```

Solicitação de tasks

A troca de `proxy tasks` trata da entrega e do reconhecimento de tasks de comandos remotos entre o server e um proxy passivo. Quando o server consulta o proxy por resultados de tasks, ele envia uma solicitação vazia de `proxy tasks` no intervalo definido por `ProxyDataFrequency`. O proxy responde com sua versão atual e quaisquer resultados de tasks pendentes (incluindo timestamps). Em seguida, o server confirma o recebimento retornando uma `response` e pode incluir novas tasks para o proxy executar.

name	value type	description
server→proxy: **request** proxy→server:	*string*	'proxy tasks'

name	value type	description
version	*string*	Versão do proxy (<major>.<minor>.<build>).
clock	*number*	*(optional)* Timestamp da transferência de dados (segundos).
ns	*number*	*(optional)* Timestamp da transferência de dados (nanosegundos).
server→proxy:		
response	*string*	Informações de sucesso da solicitação (’success’ ou ’failed’).

Exemplo:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

Proxy ativo

Solicitação de configuração

A solicitação `proxy config` é enviada pelo proxy ativo para obter dados de configuração do proxy. Essa solicitação é enviada a cada `ProxyConfigFrequency` (parâmetro de configuração do proxy) segundos.

name	value type	description
proxy→server:		
request	*string*	’proxy config’
host	*string *	Nome do proxy.
version	*string*	Versão do proxy (<major>.<minor>.<build>).
session	*string*	Token da sessão de configuração do proxy.
config_revision	*number*	Revisão da configuração do proxy.
hostmap_revision	*number*	Revisão atual do mapeamento de host para proxy.
server→proxy:		
full_sync	*number*	1 - se os dados completos de configuração forem enviados; caso contrário, ausente (opcional).
data	*array*	Objeto com dados das tabelas. Ausente se a configuração não tiver sido 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 sobre macros secretas, ausentes se não houver alterações nas macros do vault (opcional).
proxy_group	*string*	Nome do grupo de proxies ao qual o proxy pertence.
config_revision	*number*	Revisão do cache de configuração - enviada com os dados de configuração (opcional).
del_hostids	*array*	Array de hostids removidos (opcional).

name	value type	description
-	*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, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
```

```
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
```

```
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Solicitação de dados

A solicitação `proxy data` é enviada pelo proxy para fornecer dados de disponibilidade de interface do host, histórico, descoberta e autoregistro. Essa solicitação é enviada a cada `DataSenderFrequency` (parâmetro de configuração do proxy) segundos. Observe que o proxy ativo ainda consultará o server do Zabbix a cada segundo para tarefas de comando remoto (com uma solicitação `proxy data` vazia).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Nome do proxy.

name	value type	description
session	*string*	Token da sessão de dados.
interface availability	*array*	*(optional)* Array de objetos 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*	*(optional)* 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 (nanosegundos).
value	*string*	*(optional)* Valor do item.
id	*number*	Identificador do valor (contador crescente, único dentro de uma sessão de dados).
timestamp	*number*	*(optional)* Timestamp de itens do tipo log.
source	*string*	*(optional)* Valor de origem do item eventlog.
severity	*number*	*(optional)* Valor de severidade do item eventlog.
eventid	*number*	*(optional)* Valor de eventid do item eventlog.
state	*string*	*(optional)* Estado do item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Último tamanho de log dos itens do tipo log.
mtime	*number*	*(optional)* Hora de modificação dos itens do tipo log.
discovery data	*array*	*(optional)* 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 null para dados da regra de descoberta.
type	*number*	Tipo de verificação de descoberta: **-1** dados da regra de descoberta **0**, *SVC_SSH* - verificação de serviço SSH **1**, *SVC_LDAP* - verificação de serviço LDAP **2**, *SVC_SMTP* - verificação de serviço SMTP **3**, *SVC_FTP* - verificação de serviço FTP **4**, *SVC_HTTP* - verificação de serviço HTTP **5**, *SVC_POP* - verificação de serviço POP **6**, *SVC_NNTP* - verificação de serviço NNTP **7**, *SVC_IMAP* - verificação de serviço IMAP **8**, *SVC_TCP* - verificação de disponibilidade de porta TCP **9**, *SVC_AGENT* - agent do Zabbix **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **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*	*(optional)* Número da porta do serviço.
key_	*string*	*(optional)* Chave do item para verificação de descoberta do tipo **9** *SVC_AGENT*
value	*string*	*(optional)* Valor recebido do serviço, pode estar vazio para a maioria dos serviços.

name	value type	description
status	*number*	*(optional)* Status do serviço: **0**, *DOBJECT_STATUS_UP* - serviço UP **1**, *DOBJECT_STATUS_DOWN* - serviço DOWN
autoregistration	*array*	*(optional)* Array de objetos de dados de autoregistro.
clock	*number*	Timestamp dos dados de autoregistro.
host	*string*	Nome do host.
ip	*string*	*(optional)* Endereço IP do host.
dns	*string*	*(optional)* Nome DNS resolvido a partir do endereço IP.
port	*string*	*(optional)* Porta do host.
host_metadata	*string*	*(optional)* Metadados do host enviados pelo agent (com base no parâmetro de configuração do agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(optional)* Array de tarefas.
type	*number*	Tipo de tarefa: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - resultado de comando remoto
status	*number*	Status de 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*	*(optional)* Mensagem de erro.
parent_taskid	*number*	ID da tarefa pai.
more	*number*	*(optional)* 1 - há mais dados de histórico para enviar
clock	*number*	*(optional)* Timestamp da transferência de dados (segundos).
ns	*number*	*(optional)* Timestamp da transferência de dados (nanosegundos).
version	*string*	Versão do proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informação de sucesso da solicitação (’success’ ou ’failed’).
upload	*string*	Controle de envio para dados históricos (history, autoregistration, disponibilidade do host, descoberta de rede). Valores possíveis: **enabled** - operação normal **disabled** - o server não está aceitando dados (possivelmente devido ao cache interno acima do limite)
tasks	*array*	*(optional)* Array de tarefas.
type	*number*	Tipo de tarefa: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Hora de criação da tarefa.
ttl	*number*	Tempo em segundos após o qual a tarefa expira.
commandtype	*number*	Tipo de comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usar script personalizado **1**, *ZBX_SCRIPT_TYPE_IPMI* - usar IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usar SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usar Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usar script global (atualmente funcionalmente equivalente a script personalizado)
command	*string*	Comando remoto a ser executado.
execute_on	*number*	Destino de execução para scripts personalizados: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - executar script no agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - executar script no server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - executar script no proxy
port	*number*	*(optional)* Porta para comandos Telnet e SSH.
authtype	*number*	*(optional)* Tipo de autenticação para comandos SSH.
username	*string*	*(optional)* Nome de usuário para comandos Telnet e SSH.
password	*string*	*(optional)* Senha para comandos Telnet e SSH.

name	value type	description
publickey	*string*	*(optional)* Chave pública para comandos SSH.
privatekey	*string*	*(optional)* 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
}]
}
```


Parâmetros de configuração: Zabbix proxy

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix proxy.

Especifique os valores dos parâmetros no arquivo de configuração do proxy do Zabbix (`zabbix_proxy.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver todos os detalhes de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
AllowRoot	Permitir que o proxy seja executado como `root`.
AllowUnsupportedDBVersions	Permitir que o proxy funcione com versões de banco de dados não compatíveis.
APMListenIP	Especificar o endereço IP no qual o coletor de APM deve escutar dados de telemetria recebidos.
APMListenPort	Especificar a porta na qual o coletor de APM deve escutar dados de telemetria recebidos.
APMTLSCAFile	Especificar o caminho completo do arquivo que contém os certificados de CA de nível superior para verificar os certificados de aplicações ou coletores OpenTelemetry que enviam dados ao coletor de APM.
APMTLSCertFile	Especificar o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do coletor de APM.
APMTLSKeyFile	Especificar o caminho completo do arquivo que contém a chave privada do coletor de APM.
CacheSize	Definir o tamanho do cache de configuração.
ConfigFrequency	Este parâmetro está **obsoleto** (use ProxyConfigFrequency). Definir a frequência de recuperação dos dados de configuração do Zabbix server.
DataSenderFrequency	Definir a frequência de envio dos dados coletados ao Zabbix server.

Parâmetro	Descrição
DBHost	Especificar o host do banco de dados (ou o diretório do socket).
DBName	Especificar o nome do banco de dados ou o caminho para o arquivo de banco de dados do SQLite3.
DBPassword	Especificar a senha do banco de dados.
DBPort	Especificar a porta do host do banco de dados do Zabbix.
DBSchema	Especificar o nome do esquema do banco de dados. Usado para PostgreSQL.
DBSocket	Especificar o caminho para o arquivo de socket do MySQL.
DBUser	Especificar o usuário do banco de dados.
DBTLSConnect	Especificar um valor para exigir uma conexão TLS com o banco de dados.
DBTLSCAFile	Especificar o caminho completo do arquivo que contém os certificados de CA de nível superior para a verificação do certificado do banco de dados.
DBTLSCertFile	Especificar o caminho completo do arquivo que contém o certificado do Zabbix proxy para autenticação no banco de dados.
DBTLSKeyFile	Especificar o caminho completo do arquivo que contém a chave privada para autenticação no banco de dados.
DBTLSCipher	Especificar a lista de cifras de criptografia permitidas pelo Zabbix proxy para protocolos TLS até TLS v1.2. Compatível somente com MySQL.
DBTLSCipher13	Especificar a lista de conjuntos de cifras de criptografia permitidos pelo Zabbix proxy para o protocolo TLS v1.3. Compatível somente com MySQL, a partir da versão 8.0.16.
DebugLevel	Definir o nível de depuração.
EnableRemoteCommands	Permitir comandos remotos do Zabbix server.
ExternalScripts	Especificar o local dos scripts externos.
Fping6Location	Especificar o local do fping6.
FpingLocation	Especificar o local do fping.
HistoryCacheSize	Definir o tamanho do cache de histórico.
HistoryIndexCacheSize	Definir o tamanho do cache de índices de histórico.
Hostname	Especificar o nome do proxy.
HostnameItem	Especificar o item usado para definir Hostname (se não estiver definido).
HousekeepingFrequency	Definir a frequência do procedimento de housekeeping (em horas).
Include	Especificar um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
JavaGateway	Especificar o endereço IP (ou nome do host) do Zabbix Java gateway.
JavaGatewayPort	Especificar a porta do Zabbix Java gateway.
ListenBacklog	Definir o número máximo de conexões pendentes na fila TCP.
ListenIP	Especificar o endereço IP de escuta do trapper.
ListenPort	Especificar a porta de escuta do trapper.
LoadModule	Especificar o módulo a ser carregado na inicialização do proxy.
LoadModulePath	Especificar o caminho completo para o local dos módulos do proxy.
LogFile	Especificar o arquivo de log do proxy.
LogFileSize	Definir o tamanho máximo do arquivo de log.
LogRemoteCommands	Ativar o registro de comandos shell executados como avisos.
LogSlowQueries	Definir a duração máxima que uma consulta ao banco de dados pode levar antes de ser registrada (em milissegundos).
LogType	Especificar o tipo de saída do log.
MaxConcurrentChecksPerPoller	Definir o número máximo de verificações assíncronas que podem ser executadas simultaneamente por cada poller de HTTP agent, poller de agent ou poller SNMP.
PidFile	Especificar o arquivo PID.
ProxyBufferMode	Especificar o mecanismo de armazenamento dos dados de histórico, descoberta e registro automático (disco/memória/híbrido).
ProxyConfigFrequency	Definir a frequência de recuperação dos dados de configuração do Zabbix server, em segundos.
ProxyLocalBuffer	Definir por quanto tempo os dados serão mantidos localmente (em horas), mesmo que já tenham sido sincronizados com o server.
ProxyMemoryBufferAge	Definir a idade máxima dos dados no buffer de memória do proxy, em segundos.
ProxyMemoryBufferSize	Definir o tamanho do cache de memória compartilhada para os dados coletados de histórico, descoberta e registro automático.
ProxyMode	Definir o modo de operação do proxy (ativo/passivo).
ProxyOfflineBuffer	Definir por quanto tempo os dados serão mantidos (em horas) caso não haja conectividade com o Zabbix server.
Server	Se ProxyMode estiver definido como `active`: especificar o endereço IP ou o nome DNS do Zabbix server (address:port) ou do cluster (address:port;address2:port) do qual obter dados de configuração e para o qual enviar dados. Se ProxyMode estiver definido como `passive`: especificar os endereços IP ou nomes DNS do Zabbix server.

Parâmetro	Descrição
SNMPTrapperFile	Especificar o arquivo temporário usado para passar dados do daemon de traps SNMP para o proxy.
SocketDir	Especificar o diretório para armazenar sockets IPC usados pelos serviços internos do Zabbix.
SourceIP	Especificar o endereço IP de origem.
SSHKeyLocation	Especificar o local das chaves públicas e privadas para verificações e ações SSH.
SSLCertLocation	Especificar o local dos arquivos de certificado do cliente SSL para autenticação do cliente.
SSLKeyLocation	Especificar o local dos arquivos de chave privada SSL para autenticação do cliente.
SSLCALocation	Especificar o local dos arquivos da autoridade certificadora (CA) para verificação do certificado do server SSL.
StartAgentPollers	Definir o número de instâncias pré-fork de pollers assíncronos de Zabbix agent.
StartAPMCollectors	Definir como 1 para iniciar o coletor de APM.
StartBrowserPollers	Definir o número de instâncias pré-fork de pollers de itens de navegador.
StartDBSyncers	Definir o número de instâncias pré-fork de sincronizadores de histórico.
StartDiscoverers	Definir o número de instâncias pré-fork de workers de descoberta.
StartHTTPAgentPollers	Definir o número de instâncias pré-fork de pollers assíncronos de HTTP agent.
StartHTTPPollers	Definir o número de instâncias pré-fork de pollers HTTP.
StartIPMIPollers	Definir o número de instâncias pré-fork de pollers IPMI.
StartJavaPollers	Definir o número de instâncias pré-fork de pollers Java.
StartODBCPollers	Definir o número de instâncias pré-fork de pollers ODBC.
StartPingers	Definir o número de instâncias pré-fork de pingers ICMP.
StartPollersUnreachable	Definir o número de instâncias pré-fork de pollers para hosts inacessíveis (incluindo IPMI e Java).
StartPollers	Definir o número de instâncias pré-fork de pollers.
StartPreprocessors	Definir o número de instâncias pré-iniciadas de workers de pré-processamento.
StartSNMPPollers	Definir o número de instâncias pré-fork de pollers SNMP assíncronos.
StartSNMPTrapper	Definir como "1" para iniciar um processo de trapper SNMP.
StartTelemetryQueryPollers	Definir o número de instâncias pré-fork de pollers assíncronos de consultas de telemetria.
StartTrappers	Definir o número de instâncias pré-fork de trappers.
StartVMwareCollectors	Definir o número de instâncias pré-fork de coletores VMware.
StatsAllowedIP	Especificar os endereços IP ou nomes DNS de instâncias externas do Zabbix autorizadas a solicitar estatísticas.
TelemetryProvider	Definir a fonte de dados de telemetria.
Timeout	Especificar por quanto tempo aguardar (em segundos) o estabelecimento da conexão e a troca de dados com o Zabbix proxy, agent, serviço web e para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`).
TLSAccept	Especificar o nível de criptografia exigido para conexões recebidas do Zabbix server.
TLSCAFile	Especificar o caminho completo do arquivo que contém os certificados de CA de nível superior para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especificar o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do server, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	Especificar a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherAll13	Especificar a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherCert	Especificar a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherCert13	Especificar a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherPSK	Especificar a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSCipherPSK13	Especificar a string de cifras para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substituir os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSConnect	Especificar o nível de criptografia exigido para conexões com o Zabbix server.
TLSCRLFile	Especificar o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	Especificar o caminho completo do arquivo que contém a chave privada do proxy, usada nas comunicações criptografadas entre componentes do Zabbix.
TLSListen	Exigir conexões exclusivamente TLS no socket do trapper.
TLSPSKFile	Especificar o caminho completo do arquivo que contém a chave pré-compartilhada do proxy, usada nas comunicações criptografadas com o Zabbix server.

Parâmetro	Descrição
TLSPSKIdentity	Especificar a string de identidade da chave pré-compartilhada, usada nas comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	Especificar o emissor permitido do certificado do server.
TLSServerCertSubject	Especificar o assunto permitido do certificado do server.
TmpDir	Especificar o diretório temporário.
TrapperTimeout	Especificar o tempo limite, em segundos, para: - recuperação dos dados de configuração do Zabbix server; - execução de scripts globais ou execução de comandos remotos.
UnavailableDelay	Definir a frequência de verificação da disponibilidade do host durante o período de indisponibilidade.
UnreachableDelay	Definir a frequência de verificação da disponibilidade do host durante o período de inacessibilidade.
UnreachablePeriod	Definir após quantos segundos de inacessibilidade o host deve ser considerado indisponível.
User	Remover privilégios e usar um usuário específico existente no sistema.
Vault	Especificar o provedor do vault.
VaultAppRoleID	Especificar o ID do AppRole do vault HashiCorp.
VaultAppSecretID	Especificar o ID secreto do AppRole do vault HashiCorp.
VaultDBPath	Especificar o local para recuperar credenciais do banco de dados por chaves.
VaultPrefix	Especificar um prefixo personalizado para o caminho ou a consulta do vault.
VaultTLSCertFile	Especificar o nome do arquivo de certificado SSL usado para autenticação do cliente.
VaultTLSKeyFile	Especificar o nome do arquivo de chave privada SSL usado para autenticação do cliente.
VaultToken	Especificar o token de autenticação do vault HashiCorp.
VaultURL	Especificar a URL do server do vault.
VMwareCacheSize	Definir o tamanho da memória compartilhada para armazenar dados do VMware.
VMwareFrequency	Definir a frequência de coleta de dados de um único serviço VMware, em segundos.
VMwarePerfFrequency	Definir a frequência de recuperação das estatísticas dos contadores de desempenho de um único serviço VMware, em segundos.
VMwareTimeout	Definir o número máximo de segundos que um `vmware collector` aguardará por uma resposta do serviço VMware.
WebDriverURL	Especificar a URL da interface do WebDriver.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores dos arquivos de configuração fornecidos.
- Os valores são compatíveis com variáveis de ambiente.
- O Zabbix aceita arquivos de configuração somente na codificação UTF-8, sem BOM.
- Comentários iniciados com # são compatíveis somente no início da linha.

Detalhes do parâmetro

AllowRoot

Permite que o proxy seja executado como `root`. Se estiver desativado e o proxy for iniciado por `root`, o proxy tentará alternar para o usuário `zabbix`. Não tem efeito se for iniciado por um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

AllowUnsupportedDBVersions

Permite que o proxy funcione com versões de banco de dados não suportadas.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

APMListenIP

Especifique o endereço IP no qual o coletor APM deve escutar os dados de telemetria recebidos.
 Se não for especificado, o coletor APM escutará em todas as interfaces de rede.

Padrão: `0.0.0.0`

APMListenPort

Especifique a porta na qual o coletor APM deve escutar os dados de telemetria recebidos.
 A porta padrão é a porta padrão do OpenTelemetry (OTLP/gRPC), usada por padrão pelos SDKs e coletores do OpenTelemetry.

Padrão: 4317
 Intervalo: 1024-32767

APMTLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da CA de nível superior para verificar os certificados de aplicações ou coletores OpenTelemetry que enviam dados para o coletor APM.
 Se este parâmetro estiver definido, o coletor APM aceitará apenas conexões criptografadas; caso contrário, aceitará apenas conexões não criptografadas.

APMTLSCertFile

Especifique o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do coletor APM.
 Este parâmetro deve ser definido juntamente com `APMTLSKeyFile` e somente se `APMTLSCAFile` estiver definido.

APMTLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do coletor APM.
 Este parâmetro deve ser definido junto com `APMTLSCertFile` e somente se `APMTLSCAFile` estiver definido.

CacheSize

Defina 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).
Define a frequência de obtenção dos dados de configuração do Zabbix server (em segundos).
Parâmetro do proxy ativo. Ignorado para proxies passivos (consulte o parâmetro `ProxyMode`).

Padrão: 3600
 Intervalo: 1-604800

DataSenderFrequency

Define a frequência de envio dos dados coletados do proxy para o server do Zabbix. Observe que um proxy ativo ainda consultará o server do Zabbix a cada segundo para tarefas de comando remoto.
Parâmetro do proxy ativo. Ignorado para proxies passivos (consulte o parâmetro `ProxyMode`).

Padrão: 1
 Intervalo: 1-3600

DBHost

Especifique o host do banco de dados (ou o diretório do socket).

Para MySQL:

- `localhost` ou uma string vazia usa o socket UNIX de domínio padrão.

Para PostgreSQL:

- `localhost` é resolvido via DNS (normalmente para `127.0.0.1`).
- Uma string vazia usa o socket UNIX de domínio padrão.
- Um caminho (por exemplo, `/var/run/pgbouncer`) resulta no uso do socket UNIX nesse caminho.
- Uma lista separada por vírgulas pode conter vários valores `host:port`. Cada host é tentado em ordem até que uma conexão de leitura e gravação seja estabelecida. Exemplo: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,`

Padrão: `localhost`

DBName

Especifique o nome do banco de dados ou o caminho para o arquivo do banco de dados do SQLite3 (a arquitetura multiprocesso do Zabbix não permite o uso de um banco de dados na 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 server do Zabbix está usando.

Obrigatório: Sim

DBPassword

Especifique a senha do banco de dados. Comente esta linha se nenhuma senha for usada. Ignorado para SQLite.

DBPort

Especifique a porta do host do banco de dados do Zabbix (veja `DBHost`). Ignorado para SQLite.
 DBSocket e DBPort são mutuamente exclusivos na configuração do proxy. Especifique apenas um, ou deixe ambos indefinidos.

Padrão para MySQL: 3306

Padrão para PostgreSQL: 5432

Intervalo: 1024-65535

DBSchema

Especifique o nome do esquema do banco de dados. Usado para PostgreSQL.

DBSocket

Especifique o caminho para o arquivo de socket do MySQL.
DBSocket e DBPort são mutuamente exclusivos na configuração do proxy. Especifique apenas um, ou deixe ambos indefinidos.

DBUser

Especifique o usuário do banco de dados. Ignorado para SQLite.

DBTLSConnect

Especifique um valor para impor 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
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 é definido nenhum valor e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da CA de nível superior para a verificação de certificados do banco de dados.

Obrigatório: não (sim, se DBTLSConnect estiver definido como *verify_ca* ou *verify_full*)

DBTLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do proxy Zabbix para autenticação no banco de dados.

DBTLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada para autenticação no banco de dados.

DBTLSCipher

Especifique a lista de cifras de criptografia que o Zabbix proxy permite para os protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.

DBTLSCipher13

Especifique a lista de suites 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

Defina 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 (produz muitas informações); - *5* - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

EnableRemoteCommands

Permite comandos remotos do Zabbix server.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

ExternalScripts

Especifique o local do script externo (depende da variável de instalação em tempo de compilação `datadir`).

Padrão: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Especifique o local do fping6. Certifique-se de que o binário fping6 tenha propriedade de root e o sinalizador SUID definido. Deixe em branco (`Fping6Location=`) se o utilitário `fping` for capaz de processar endereços IPv6.

Padrão: `/usr/sbin/fping6`

FpingLocation

Especifique o local do fping. Certifique-se de que o binário do fping tenha propriedade de root e o sinalizador SUID definido.

Padrão: `/usr/sbin/fping`

HistoryCacheSize

Defina o tamanho do cache de histórico, em bytes. O tamanho da memória compartilhada para armazenar dados de histórico.

Padrão: 16M
 Intervalo: 128K-16G

HistoryIndexCacheSize

Defina o tamanho do cache de índice do histórico, em bytes. O tamanho da memória compartilhada para indexar os dados de histórico armazenados no cache de histórico. O tamanho do cache de índice precisa de aproximadamente 100 bytes para armazenar em cache um item.

Padrão: 4M
 Intervalo: 128K-16G

Nome do host

Especifique um nome de proxy exclusivo, com diferenciação entre 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

Especifique um item para definir `Hostname` (se não estiver definido).
 Este item será executado no proxy da mesma forma que em um agent.
 Ignorado se `Hostname` estiver definido.
Não oferece suporte a parâmetros de usuário, contadores de desempenho ou aliases; oferece suporte a `system.run[]`.

Padrão: `system.hostname`

HousekeepingFrequency

Defina a frequência do procedimento de housekeeping (em horas). Housekeeping é a remoção de informações desatualizadas do banco de dados.
Nota: Para reduzir a carga na inicialização do proxy, o housekeeping é adiado por 30 minutos após a inicialização do proxy. Assim, se `HousekeepingFrequency` for 1, o primeiro procedimento de housekeeping após a inicialização do proxy será executado após 30 minutos e, depois disso, será repetido a cada hora.
É possível desativar o housekeeping automático definindo HousekeepingFrequency como 0. Nesse caso, o procedimento de housekeeping só pode ser iniciado pela opção de controle em tempo de execução `housekeeper_execute`.
Consulte também as opções e detalhes de runtime control e do housekeeping procedure.

Padrão: 1
 Intervalo: 0-24

Include

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você o altere durante o tempo de compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_proxy.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê, portanto considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivo:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento a cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir apenas os arquivos relevantes de um diretório, use o curinga (*) para correspondência de padrão.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Especifique o endereço IP ou o nome do host do Zabbix Java gateway. Necessário apenas se os Java pollers estiverem iniciados.

JavaGatewayPort

Especifique a porta do Zabbix Java gateway.

Padrão: 10052
 Intervalo: 1024-32767

ListenBacklog

Define o número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado também depende do sistema; valores muito altos podem ser truncados silenciosamente para o 'maximum specified by the implementation'.

Padrão: `SOMAXCONN`
 Intervalo: 0 - INT_MAX

ListenIP

Especifique o endereço IP de escuta do trapper. Uma lista delimitada por vírgulas.
O trapper escutará em todas as interfaces de rede se este parâmetro não for especificado.

Padrão: `0.0.0.0`

ListenPort

Especifique a porta de escuta do trapper.

Padrão: 10051
 Intervalo: 1024-32767

LoadModule

Especifique 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 precedente for absoluto (começar com /), `LoadModulePath` será ignorado.
 Formatos: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Vários parâmetros `LoadModule` podem ser especificados.

LoadModulePath

O caminho completo para o local dos módulos do proxy. O padrão depende das opções de compilação.

LogFile

Especifique o arquivo de log do proxy.

Obrigatório: Sim, se `LogType` estiver definido como `file`; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa 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

Ative o registro dos comandos de shell executados como avisos.

Padrão: 0
 Valores: 0 - desativado, 1 - ativado

LogSlowQueries

Define a duração máxima que uma consulta ao banco de dados pode levar antes de ser registrada no log (em milissegundos).
0 - não registrar consultas lentas.
Esta opção passa a ficar habilitada a partir de `DebugLevel=3`.

Padrão: 0
 Intervalo: 0-3600000

LogType

Especifique 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

Define o número máximo de verificações assíncronas que podem ser executadas de uma vez por cada HTTP agent poller, agent poller ou SNMP poller. Veja `StartHTTPAgentPollers`, `StartAgentPollers` e `StartSNMPPollers`.

Padrão: 1000
 Intervalo: 1-1000

PidFile

Especifique o arquivo PID.

Padrão: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Especifique o mecanismo de armazenamento para os dados de history, discovery e autoregistration: - *disk* - os dados são armazenados no banco de dados e enviados a partir do banco de dados; - *memory* - os dados são armazenados na memória e enviados a partir da memória. Se o buffer ficar sem memória, os dados antigos serão descartados. Ao desligar, o buffer é descartado. - *hybrid* - o buffer do proxy normalmente funciona como no modo de memória até ficar sem memória ou o registro mais antigo exceder a idade configurada. Se isso acontecer, o buffer é descarregado para o banco de dados e passa a funcionar como no modo de disco até que todos os dados tenham sido enviados e então volta a funcionar com a memória. Ao desligar, o buffer de memória é descarregado para o banco de dados.

Veja também: Proxy memory buffer.

Padrão: `disk`
 Valores: disk; memory; hybrid

ProxyConfigFrequency

Defina a frequência de obtenção dos dados de configuração do Zabbix server em segundos.
Parâmetro do proxy ativo. Ignorado para proxies passivos (consulte o parâmetro `ProxyMode`).

Padrão: 10
 Intervalo: 1-604800

ProxyLocalBuffer

Defina a duração de retenção dos dados localmente (em horas), mesmo que os dados já tenham sido sincronizados com o server..
Este parâmetro pode ser usado se os dados locais forem utilizados por aplicativos de terceiros.

Padrão: 0
 Intervalo: 0-720

ProxyMemoryBufferAge

Define a idade máxima dos dados no buffer de memória do proxy, em segundos.
 Quando habilitado (diferente de zero) e os registros no buffer de memória do proxy estiverem mais antigos, ele força o buffer do proxy a mudar para o modo de banco de dados até que todos os registros sejam enviados para o server. Este parâmetro deve ser menor ou igual ao parâmetro `ProxyOfflineBuffer`.

Padrão: 0
 Intervalo: 0;600-864000

ProxyMemoryBufferSize

Define o tamanho do cache de memória compartilhada para os dados coletados de history, discovery e auto registration (em bytes). Se habilitado (diferente de zero), o proxy manterá os dados de history, discovery e autoregistration na memória, a menos que o cache esteja cheio ou que os registros armazenados sejam mais antigos do que o `ProxyMemoryBufferAge` definido. Este parâmetro não pode ser usado em conjunto com o parâmetro `ProxyLocalBuffer`.

Padrão: 0
 Intervalo: 0;128K-2G

ProxyMode

Define o modo de operação do proxy.
0 - proxy no modo ativo
1 - proxy no modo passivo
Nota que os 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 é realizada.

Padrão: 0
 Intervalo: 0-1

ProxyOfflineBuffer

Defina a duração de retenção dos dados (em 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 `active`: especifique o endereço IP ou o nome DNS do Zabbix server (address:port) ou cluster (address:port;address2:port) de onde obter os dados de configuração e para onde enviar os 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 `passive`: especifique endereços IP, opcionalmente em notação CIDR, ou nomes DNS do Zabbix server. Uma lista separada por vírgulas.
 Conexões de entrada serão aceitas somente 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` serão 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

Especifique o arquivo temporário usado para passar dados do daemon de traps SNMP para o proxy.
Deve ser o mesmo definido em `zabbix_trap_receiver.pl` ou no arquivo de configuração do SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

Especifique o diretório para armazenar os sockets IPC usados pelos serviços internos do Zabbix.

Padrão: `/tmp`

SourceIP

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix server
- conexões sem agent (VMware, SSH, JMX, SNMP, Telnet e verificações simples)
- conexões do HTTP agent
- solicitações HTTP JavaScript do item script
- solicitações HTTP JavaScript de pré-processamento
- conexões com o Vault

SSHKeyLocation

Especifique o local das chaves pública e privada para verificações e ações SSH.

SSLCertLocation

Especifique o local dos arquivos de certificado SSL do cliente para autenticação do cliente.
Este parâmetro é usado no monitoramento web, na comunicação com o Vault e para a fonte de dados de telemetria ClickHouse.

SSLKeyLocation

Especifique o local dos arquivos de chave privada SSL para autenticação do cliente.
Este parâmetro é usado no monitoramento web, na comunicação com o Vault e para a fonte de dados de telemetria ClickHouse.

SSLCALocation

Especifique o local dos arquivos da autoridade certificadora (CA) para a verificação do certificado do server SSL.
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 terá efeito se o libcurl tiver sido compilado para usar OpenSSL. Para obter mais informações, consulte a página web do cURL.
Este parâmetro é usado no monitoramento web, na autenticação SMTP, em itens do HTTP agent, na comunicação com o Vault e para a fonte de dados de telemetria ClickHouse.

StartAgentPollers

Define o número de instâncias pré-forkadas de `pollers` do Zabbix agent. Veja `MaxConcurrentChecksPerPoller`.

Padrão: 1
 Intervalo: 0-1000

StartAPMCollectors

Defina como 1 para iniciar uma instância pré-fork de um coletor APM (os processos `apm manager` e `apm worker`).

Padrão: 0
 Intervalo: 0-1

StartBrowserPollers

Define o número de instâncias pré-forked de browser item `pollers`.

Padrão: 1
 Intervalo: 0-1000

StartDBSyncers

Define o número de instâncias pré-forked de `history syncers`.
Nota: Tenha cuidado ao alterar este valor; aumentá-lo pode causar mais prejuízo do que benefício.

Padrão: 4
 Intervalo: 1-100

StartDiscoverers

Define o número de instâncias pré-forked de discovery workers.

Padrão: 5
 Intervalo: 0-1000

StartHTTPAgentPollers

Define o número de instâncias pré-forked de HTTP agent pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartHTTPPollers

Define o número de instâncias pré-forked de HTTP pollers.

Padrão: 1
 Intervalo: 0-1000

StartIPMIPollers

Define o número de instâncias pré-forked de IPMI pollers.

Padrão: 0
 Intervalo: 0-1000

StartJavaPollers

Define o número de instâncias pré-forked de Java pollers.

Padrão: 0
 Intervalo: 0-1000

StartODBCPollers

Define o número de instâncias pré-forked de ODBC pollers.

Padrão: 1
 Intervalo: 0-1000

StartPingers

Define o número de instâncias pré-forked de ICMP pingers.

Padrão: 1
 Intervalo: 0-1000

StartPollersUnreachable

Define o número de instâncias pré-forked de pollers para hosts inacessíveis (incluindo IPMI e Java). Pelo menos um poller para hosts inacessíveis deve estar em execução se pollers regulares, IPMI ou Java forem iniciados.

Padrão: 1
 Intervalo: 0-1000

StartPollers

Define o número de instâncias pré-fork de pollers.

Padrão: 5
 Intervalo: 0-1000

StartPreprocessors

Defina o número de instâncias pré-iniciadas de threads de worker de pré-processamento para não ser menor que a quantidade de núcleos de CPU disponíveis. Mais workers devem ser definidos se o pré-processamento não estiver limitado pela CPU e tiver muitas requisições de rede.

Padrão: 16
 Intervalo: 1-1000

StartSNMPPollers

Define o número de instâncias pré-forked de SNMP pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartTelemetryQueryPollers

Defina o número de instâncias pré-criadas de pollers de consultas assíncronas de Telemetry. Consulte MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartSNMPTrapper

Defina como "1" para iniciar um processo SNMP trapper.

Padrão: 0
 Intervalo: 0-1

StartTrappers

Define o número de instâncias pré-forkadas de trappers.
Trappers aceitam conexões de entrada do Zabbix sender e de agents ativos.

Padrão: 5
 Intervalo: 0-1000

StartVMwareCollectors

Define o número de instâncias pré-forked de vmware collector.

Padrão: 0
 Intervalo: 0-250

StatsAllowedIP

Especifique endereços IP (opcionalmente em notação CIDR) ou nomes DNS de instâncias externas do Zabbix que têm permissão para solicitar estatísticas. Uma lista separada por vírgulas.
 Se este parâmetro não estiver 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
```

TelemetryProvider

Defina a fonte de dados de telemetria usando o formato `<provider>[;<options>]`, em que:

- `<provider>` - uma das fontes de dados de telemetria compatíveis: `clickhouse`
- `<options>` - pares `key=value` separados por vírgulas. Os valores podem estar entre aspas.

Opções para o ClickHouse:

- `url` - define a URL HTTP ou HTTPS usada para conectar-se à fonte de dados de telemetria.
- `db` - define o nome do banco de dados.
- `username` - define o nome de usuário usado para fazer login no banco de dados.
- `password` - define a senha usada para fazer login no banco de dados.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - defina essas opções para configurar uma conexão segura (TLS/SSL) com o banco de dados.
- `source_ip` - define o endereço IP de origem usado para conexões de saída com a fonte de dados de telemetria.
- `vault_path` - define o caminho para um segredo do Vault que contém os valores de `username` e `password`. Não use junto com as opções `username` e `password`.

Exemplo para o ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Exemplo para o ClickHouse com o Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Especifique quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o server Zabbix, agent, web service e verificações SNMP legadas (número OID único ou string).

Este parâmetro define a duração de várias operações de comunicação:

- Execução de comando remoto no agent Zabbix
- Execução de comando SSH / Telnet
- Reagendamento de items quando a interface IPMI se torna indisponível
- Envio de resposta ao server Zabbix ao falhar a troca de dados devido a problemas de permissões ou criptografia
- Prazo para sockets assíncronos de IPC e opções de controle em tempo de execução
- Solicitações DNS do poller assíncrono
- Resposta para heartbeat de verificação ativa
- Recuperação de dados (valores) do agent Zabbix a partir de agents ativos
- Recuperação de dados do Zabbix sender
- Envio da lista de verificações ativas para o agent Zabbix

Este timeout **não** será usado para as verificações que tenham configurações de timeout flexível definidas no frontend (em nível global, de proxy ou por item). Por exemplo, os items SNMP `walk[OID]` e `get[OID]` usam o timeout configurado no frontend; as verificações SNMP legadas ainda usam o valor de timeout do server.

Padrão: 3
 Intervalo: 1-30

TLSAccept

Especifique o nível de criptografia exigido para conexões de entrada do Zabbix server. Usado para um proxy passivo, ignorado em um proxy ativo. 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 para proxy passivo, se os parâmetros de certificado TLS ou PSK estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do proxy ou a cadeia de certificados, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCipherAll

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitua os critérios padrão de seleção da ciphersuite para criptografia baseada em certificado e em 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

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.

TLSCipherPSK

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Especifique o nível de criptografia exigido para conexões do proxy com o server do Zabbix. Usado para um proxy ativo, ignorado em um 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

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do proxy, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSListen

Exigir conexões somente TLS no socket do trapper.

Valores suportados:

- required - aceitar apenas conexões TLS

TLSPSKFile

Especifique o caminho completo do arquivo que contém a chave pré-compartilhada do proxy, usada para comunicações criptografadas com o Zabbix server.

TLSPSKIdentity

Especifique a string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.

TLSServerCertIssuer

Especifique o emissor do certificado do server permitido.

TLSServerCertSubject

Especifique o subject do certificado do server permitido.

TmpDir

Especifique o diretório temporário.

Padrão: `/tmp`

TrapperTimeout

Especifique o tempo limite em segundos para:
 - recuperação de dados de configuração do server Zabbix;
 - execução de script global ou execução de comando remoto.

Padrão: 300
 Intervalo: 1-300

UnavailableDelay

Define a frequência de verificação da disponibilidade do host durante o período de indisponibilidade, em segundos.

Padrão: 60
 Intervalo: 1-3600

UnreachableDelay

Define a frequência de verificação da disponibilidade do host durante o período de inacessibilidade em segundos.

Padrão: 15
 Intervalo: 1-3600

UnreachablePeriod

Define após quantos segundos de inacessibilidade tratar o host como indisponível.

Padrão: 45
 Intervalo: 1-3600

Usuário

Reduz os privilégios para um usuário específico e existente no sistema.
Tem efeito apenas se for executado como `root` e `AllowRoot` estiver desabilitado.

Padrão: `zabbix`

Vault

Especifique 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`

VaultAppRoleID

Especifique o AppRole ID do HashiCorp vault.
 Se definido, habilita a autenticação AppRole. Não pode ser usado com VaultToken.
 Suportado apenas para HashiCorp vault.

VaultAppSecretID

Especifique o secret ID do AppRole do HashiCorp vault.
 Suportado apenas para HashiCorp vault.

VaultDBPath

Especifique o local para recuperar credenciais do banco de dados por meio de chaves. Trata-se de um caminho ou consulta do Vault, dependendo do Vault.

As chaves usadas pelo 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 pelo 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

Especifique um prefixo personalizado para o caminho ou consulta do vault, dependendo do Vault. Os padrões mais adequados serão usados se não forem especificados.
Observe que 'data' é automaticamente acrescentado após o ponto de montagem para HashiCorp se `VaultPrefix` não for especificado.
Observe que 'data' é automaticamente acrescentado 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

Especifique 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 em branco.
O diretório que contém este arquivo é especificado pelo parâmetro de configuração SSLCertLocation.
Esta opção pode ser omitida, mas é recomendada para o vault CyberArkCCP.

VaultTLSKeyFile

Especifique o nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM1.
O diretório que contém esse arquivo é especificado pelo parâmetro de configuração SSLKeyLocation.
Esta opção pode ser omitida, mas é recomendada para o vault CyberArkCCP.

VaultToken

Especifique o token de autenticação do HashiCorp Vault que deve ter sido gerado exclusivamente para o Zabbix proxy com 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.
 Não pode ser usado com VaultAppRoleID.
 Suportado apenas para o HashiCorp Vault.

VaultURL

Especifique a URL do server do vault. O diretório de certificados CA do sistema será usado se SSLCALocation não for especificado.

Padrão: `https://127.0.0.1:8200`

VMwareCacheSize

Define 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

Defina a frequência de coleta de dados de um único serviço VMware (em segundos).
Essa frequência deve ser definida como o menor intervalo de atualização de qualquer item de monitoramento VMware.

Padrão: 60
 Intervalo: 10-86400

VMwarePerfFrequency

Defina a frequência de recuperação das estatísticas dos contadores de desempenho de um único serviço VMware (em segundos).
Essa frequência deve ser definida como o menor intervalo de atualização de qualquer item de monitoramento VMware que use contadores de desempenho VMware.

Padrão: 60
 Intervalo: 10-86400

VMwareTimeout

Define o número máximo de segundos que um `vmware collector` aguardará por uma resposta do serviço VMware (vCenter ou hipervisor ESX).

Padrão: 10
 Intervalo: 1-300

WebDriverURL

Especifique a URL da interface do WebDriver.

Exemplo (usado com o servidor standalone do Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

agent

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

O agent coleta informações operacionais localmente e envia os dados ao server do Zabbix para processamento adicional. Em caso de falhas (como um disco rígido ficando cheio ou um processo de serviço travando), 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.

É possível executar várias instâncias do agent em um host.

Veja também:

- Protocolos de verificação passiva e ativa
- Executando o agent como root
- Parâmetros de configuração: agent do Zabbix
- Parâmetros de configuração: agent do Zabbix (Windows)
- Agent do Zabbix no Microsoft Windows
- Agent do Zabbix no IBM AIX

Verificações passivas e ativas

Os agents do Zabbix podem executar verificações passivas e ativas:

- Verificações passivas - o agent do Zabbix responde a uma solicitação do Zabbix server (ou proxy). Por exemplo, o server solicita dados (por exemplo, uso de CPU), e o agent retorna o resultado.

- Verificações ativas - o agent do Zabbix coleta e envia dados sem aguardar uma solicitação do Zabbix server (ou proxy). Primeiro, ele recupera uma lista de itens de monitoramento do server (uso de CPU, memória disponível etc.), depois coleta os dados necessários e envia periodicamente novos valores de volta para ele.

O tipo de verificação do agent é configurado selecionando o respectivo item type de monitoramento. O agent do Zabbix processa itens do tipo *Zabbix agent* ou *Zabbix agent (active)*.

Plataformas suportadas Para as plataformas suportadas, consulte a página Requisitos.

Agent em sistemas Unix-like O Zabbix agent em sistemas Unix-like é executado no host monitorado.

Instalação

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

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

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

Note:
Binários pré-compilados do Zabbix agent estão disponíveis para download para macOS, IBM AIX, FreeBSD, OpenBSD e Solaris. Binários legados, compatíveis com a versão atual do Zabbix server/proxy, estão disponíveis para NetBSD e HP-UX. Para instalar e executar o agent no IBM AIX, consulte Zabbix agent no IBM AIX.

Se instalado como pacote

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

```
systemctl start zabbix-agent
```

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

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

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

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

Se instalado a partir de um arquivo binário

Após extrair o arquivo, localize o binário `zabbix_agentd` e o arquivo de configuração `zabbix_agentd.conf`. A localização deles dentro do arquivo depende da plataforma (por exemplo, `sbin/` e `conf/`, ou `usr/sbin/` e `etc/zabbix/`). Execute o binário diretamente do diretório extraído e especifique o arquivo de configuração com a opção `-c`. Por exemplo:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Em sistemas semelhantes ao UNIX, o Zabbix agent foi projetado para ser executado como um usuário não root. Se iniciado como root, ele muda para o usuário `zabbix` (ou para o usuário especificado pelo parâmetro [User](<PARAMS>#user)), a menos que [AllowRoot=1](<PARAMS>#allowroot) esteja definido no arquivo de configuração. Esse usuário deve existir no sistema; caso contrário, o agent não será iniciado.

No IBM AIX, o agent também pode ser executado com o System Resource Controller (SRC) e iniciado durante a inicialização do sistema; consulte Zabbix agent no IBM AIX.

Iniciar manualmente

Você pode iniciar o Zabbix agent executando diretamente o binário `zabbix_agentd`, por exemplo:

```
zabbix_agentd
```

Isso funciona quando o diretório que contém `zabbix_agentd` está incluído em `PATH`.

Agent em sistemas Windows O agent do Zabbix no Windows é executado como um serviço do Windows.

Instalação

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

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

Consulte a página Zabbix agent no Microsoft Windows para obter detalhes adicionais sobre a instalação do Zabbix agent (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:

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 diferente do arquivo padrão.
`-f --foreground`	Executa o Zabbix agent em primeiro plano (padrão: `true`).
`-p --print`	Exibe os items conhecidos e encerra. Observe que, para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se ele não estiver no local padrão).
`-t --test <item key>`	Testa o item especificado e encerra. Observe que, para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se ele não estiver no local padrão).
`-T --test-config`	Valida o arquivo de configuração e encerra.
`-h --help`	Exibe informações de ajuda.
`-V --version`	Exibe o número da versão.
Somente agent UNIX	
`-R --runtime-control <option>`	Executa funções administrativas. Consulte controle em tempo de execução.
Somente agent Windows	
`-m --multiple-agents`	Usa várias 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 do Zabbix Windows agent. Valores permitidos: `automatic` - *(padrão)* inicia o serviço automaticamente na inicialização do Windows; `delayed` - atrasa a inicialização do serviço até que os serviços iniciados automaticamente tenham concluído a inicialização (disponível no Windows Server 2008/Vista e versões posteriores); `manual` - inicia o serviço manualmente (por um usuário ou aplicativo); `disabled` - desabilita o serviço, impedindo que ele seja 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 Windows agent como serviço.
`-d --uninstall`	Desinstala o serviço do Zabbix Windows agent.
`-s --start`	Inicia o serviço do Zabbix Windows agent.
`-x --stop`	Interrompe o serviço do Zabbix Windows agent.

Exemplos específicos de uso dos parâmetros de linha de comando:

- Exibir todos os items integrados do agent com seus valores.
- Testar um parâmetro de usuário com a chave `mysql.ping` definida no arquivo de configuração especificado.
- Instalar um serviço do Zabbix Agent para Windows usando o caminho padrão para o arquivo de configuração.
- Instalar um serviço do Zabbix Agent [Hostname] para Windows usando o arquivo de configuração localizado na mesma pasta que o executável do agent e tornar o nome do serviço exclusivo, estendendo-o com o valor de `Hostname` do arquivo de configuração.
- Modificar o tipo de inicialização de um serviço do Zabbix Agent para Windows instalado usando o arquivo de configuração 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[=<target>]`	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`) Consulte todos os tipos de processo do agent. **tipo de processo,N** - tipo e número do processo (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[=<target>]`	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:

- Aumentar o nível de log de todos os processos
- Aumentar o nível de log do terceiro processo `listener`
- Aumentar o nível de log do processo com PID 1234
- Diminuir o nível de log de todos os processos de verificação ativa

```
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 compatível com OpenBSD, NetBSD e Windows.

Tipos de processo do agent Os tipos de processo do agent são os seguintes:

- `active checks` - processo para executar 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 processo.

O arquivo de log do agent é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo pode ser lido pelo grupo do proprietário. Todas as outras permissões são negadas.

Executar como usuário O Zabbix agent no UNIX foi projetado para ser executado como um usuário não root. Ele será executado com o usuário não root que o iniciou. 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', definido no código, que deve estar presente no seu sistema. Você só poderá executar o agent como 'root' se modificar o parâmetro `AllowRoot` no arquivo de configuração do agent adequadamente.

Arquivo de configuração O caminho padrão para o arquivo de configuração depende do sistema operacional e do método de instalação do Zabbix:

- No UNIX, o padrão é:

    - Para instalação a partir do código-fonte: `/usr/local/etc/zabbix_agentd.conf`, ou conforme definido pelas variáveis de compilação `--sysconfdir` ou `--prefix`.
    - Para instalação a partir de pacotes e do appliance: `/etc/zabbix/zabbix_agentd.conf`.
- No Windows, o padrão é `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Se uma única instância do agent estiver em execução em um host, ela poderá usar o arquivo de configuração padrão ou um arquivo de configuração especificado na linha de comando. No caso de várias instâncias, cada instância do agent deverá ter seu próprio arquivo de configuração (uma das instâncias poderá usar o arquivo de configuração padrão).

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

Localidade Observe que o agent requer uma localidade UTF-8 para que alguns itens textuais do agent possam retornar o conteúdo esperado. A maioria dos sistemas Unix-like modernos tem uma localidade UTF-8 como padrão; no entanto, há alguns sistemas em que isso pode precisar ser definido especificamente.

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

Protocolos de verificação passiva e ativa

Esta seção fornece detalhes do protocolo sobre verificações passivas e ativas realizadas por Zabbix agent e Zabbix agent 2.

O Zabbix usa um protocolo de comunicação baseado em JSON para se comunicar com os agents.

Os protocolos do Zabbix agent e do Zabbix agent 2 foram unificados desde o Zabbix 7.0. A diferença entre as solicitações/respostas do Zabbix agent e do Zabbix agent 2 é expressa pelo valor da tag ”variant”.

Verificações passivas

Uma verificação passiva é uma solicitação simples de dados. O server ou proxy do Zabbix solicita alguns dados (por exemplo, carga de CPU) e o agent do Zabbix envia de volta o resultado para o server.

As verificações passivas 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.

O poller do agent 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 verificações assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller).

O número de pollers assíncronos do agent é definido pelo parâmetro StartAgentPollers.

Solicitação do server

Para a definição do cabeçalho e do comprimento dos dados, consulte os detalhes do protocolo.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	`"passive checks"`
data		*array of object*	yes	Item de verificação passiva.
	key	*string*	yes	Chave do item com macros expandidas.
	timeout	*number*	yes	Tempo limite de comunicação.

Resposta do agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field	Type	Mandatory	Value
version	*string*	yes	O número da versão do agent.
variant	*number*	yes	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
data	*array of object*	yes	Contém o resultado da verificação.
value	*string*	no	O valor do item se a verificação foi bem-sucedida.
error	*string*	no	A mensagem de erro se a verificação 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' no nosso 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":"Unsupport item key."}]}**
4. O server processa os dados, altera o estado do item para não suportado com a mensagem de erro especificada
5. A conexão TCP é fechada

Failover para o protocolo antigo

Para garantir que o Zabbix server ou proxy possa funcionar com agents de versões anteriores à 7.2, que usam o protocolo em texto simples, foi implementado um failover para o protocolo antigo.

As verificações passivas são realizadas usando o protocolo JSON (7.0 e posteriores) após a reinicialização ou quando a configuração da interface é alterada.
Se nenhuma resposta JSON válida for recebida (o agent enviou "ZBX_NOTSUPPORTED"), o Zabbix armazenará em cache a interface como protocolo antigo e **tentará 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.

Verificações ativas

As verificações ativas exigem um processamento mais complexo. Primeiro, o agent deve obter do server/proxy uma lista de itens e/ou comandos remotos para processamento independente.

Os servers/proxies dos quais as verificações ativas são obtidas estão listados no parâmetro `ServerActive` do arquivo de configuração do agent. A frequência de solicitação dessas verificações é definida pelo parâmetro `RefreshActiveChecks` no mesmo arquivo de configuração. No entanto, se a atualização das verificações ativas falhar, ela será repetida após 60 segundos, conforme definido no código.

Note:
Desde o Zabbix 6.4, o agent (no modo ativo) não recebe mais do server/proxy uma cópia completa da configuração a cada dois minutos (padrão). Em vez disso, para reduzir 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 **somente** se o agent ainda não a tiver recebido, ou se algo tiver mudado na configuração do host, nas macros globais ou nas expressões regulares globais.

O agent então envia periodicamente os novos valores para o server(s). Se o agent recebeu algum remote commands para executar, o resultado da execução também será enviado. Observe que a execução de remote command 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 Active checks, porque nesse caso você não precisaria modificar o firewall para permitir conexões de entrada iniciais.

Obtendo a lista de items

Solicitação do agent

A solicitação de active checks é usada para obter os active checks a serem processados pelo agent.
Essa solicitação é enviada pelo agent na inicialização e, depois, em intervalos de 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"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Host name.
host_metadata	*string*	no	O parâmetro de configuração HostMetadata ou o valor da métrica HostMetadataItem.
interface	*string*	no	O parâmetro de configuração HostInterface ou o valor da métrica HostInterfaceItem.
ip	*string*	no	O primeiro IP do parâmetro de configuração ListenIP, se definido.
port	*number*	no	O valor do parâmetro de configuração ListenPort, se definido e se não for a porta padrão de escuta do agent.
version	*string*	yes	O número da versão do agent.
variant	*number*	yes	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Identificador de configuração para sincronização incremental de configuração.
session	*string*	no	Identificador de sessão para sincronização incremental de configuração.

Resposta do server

A resposta de active checks é enviada pelo server de volta ao agent após o processamento da solicitação de active checks.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
```

```
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Informações de erro em caso de falha.
data	*array of objects*	no	Items de active check. Omitido se a configuração do host não tiver sido alterada.
key	*string*	no	Chave do item com macros expandidas.
itemid	*number*	no	Identificador do item.
delay	*string*	no	Intervalo de atualização do item. Intervalos flexíveis/de agendamento são suportados por Zabbix agent e Zabbix agent 2 desde o Zabbix 7.0.
lastlogsize	*number*	no	lastlogsize do item.
mtime	*number*	no	mtime do item.
timeout	*string*	no	Tempo limite do item.
refresh_unsupported	*number*	no	Intervalo de atualização de item não suportado.
regexp	*array of objects*	no	Expressões regulares globais.
name	*string*	no	Nome da expressão regular global.
expression	*string*	no	Expressão regular global.
expression_type	*number*	no	Tipo da expressão regular global.
exp_delimiter	*string*	no	Delimitador da expressão regular global.
case_sensitive	*number*	no	Configuração de sensibilidade a maiúsculas e minúsculas da expressão regular global.
commands	*array of objects*	no	Comandos remotos a serem executados. Incluídos se a execução de comando remoto tiver sido acionada por uma operação de ação ou pela execução manual de um 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 active checks.
command	*string*	no	Comando remoto.
id	*number*	no	Identificador do comando remoto.
wait	*number*	no	Modo de execução do comando remoto (”0” (nowait) para comandos de operações de ação; ”1” (wait) para comandos da execução manual de script).
timeout	*number*	no	Tempo limite de execução do comando remoto na configuração de server/proxy.
config_revision	*number*	no	Identificador de configuração para sincronização incremental de configuração. Omitido se a configuração do host não tiver sido alterada. Incrementado se a configuração do host for alterada.

O server deve responder com success.

Por exemplo:

1. O agent abre uma conexão TCP
2. O agent solicita a lista de checks
3. O server responde com uma lista de items e comandos remotos a serem executados
4. O agent interpreta a resposta
5. A conexão TCP é encerrada
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 um active check. Isso é possível porque qualquer pessoa pode se passar por um agent ativo e solicitar dados de configuração de item; a autenticação não ocorre, a menos que você use opções de encryption.

Envio de dados coletados

Agent envia

A solicitação de dados do agent contém os valores de item coletados e os valores de comandos remotos executados (se houver).

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	Valores de item.
id	*number*	yes	O identificador do valor (contador incremental usado para verificar valores duplicados em caso de problemas de rede).
itemid	*number*	yes	O identificador do item.
value	*string*	no	O valor do item.
lastlogsize	*number*	no	O lastlogsize do item.
mtime	*number*	no	O mtime do item.
state	*number*	no	O estado do item.
source	*string*	no	A origem do event log do valor.
eventid	*number*	no	O eventid do event log do valor.
severity	*number*	no	A severidade do event log do valor.
timestamp	*number*	no	O timestamp do event log do valor.
clock	*number*	yes	O timestamp do valor (segundos desde a Epoch).
ns	*number*	yes	Os nanossegundos do timestamp do valor.
commands	*array of objects*	no	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 para verificações ativas.

Field		Type	Mandatory	Value
	id	*number*	no	Identificador do comando remoto.
	value	*string*	no	Resultado da execução do comando remoto, se a execução tiver sido bem-sucedida.
	error	*string*	no	Mensagem de erro da execução do comando remoto, se a execução tiver falhado.
session		*string*	yes	Identificador único da sessão gerado sempre que o agent é iniciado (com 32 dígitos hexadecimais).
host		*string*	yes	Nome do host.
version		*string*	yes	O número da versão do agent.
variant		*number*	yes	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 simples crescente, exclusivo dentro de uma sessão de dados (identificada pelo token da sessão). Esse ID é usado para descartar valores duplicados que possam 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"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	Resultados do processamento dos itens.

Attention:
Se o envio de alguns valores falhar no server (por exemplo, porque o host ou o item foi desativado 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 encerrada

Attention:
A mensagem de erro será truncada para 2048 símbolos no lado do server.

Mensagem de heartbeat

O agent envia

A mensagem de heartbeat é enviada por um agent ativo ao Zabbix server/proxy a cada `HeartbeatFrequency` segundos (configurado no arquivo de configuração do Zabbix agent ou do Zabbix agent 2).

Ela é usada para monitorar a disponibilidade das verificações 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.

Campo	Tipo	Obrigatório	Valor
heartbeat_frequency	*number*	sim	A frequência de heartbeat do agent (parâmetro de configuração HeartbeatFrequency).
version	*string*	sim	O número da versão do agent.
variant	*number*	sim	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Resposta de redirecionamento

Quando um host é reatribuído, o server pode instruir o agent a redirecionar seu heartbeat (e as verificações ativas subsequentes) para outra instância de proxy ou server.

```
{
  "response": "failed",
  "redirect": {
    "revision": 2,
    "address": "192.0.2.1:10055"
  }
}
```

Campo		Tipo	Obrigatório	Valor
response		*string*	sim	`success` | `failed`
redirect		*object*	sim	Instruções de redirecionamento.
	revision	*number*	sim	Identificador da revisão da configuração.
	address	*string*	sim	Endereço do server/proxy de destino.

Protocolo XML antigo

Note:
O Zabbix aceitará até 16 MB de dados XML codificados em Base64, mas um único valor decodificado não deve ter mais de 64 KB; caso contrário, ele será truncado para 64 KB durante a decodificação.

Executando o agent como root

Desde o Zabbix **5.0.0**, o arquivo de serviço systemd do Zabbix agent nos pacotes oficiais inclui explicitamente as diretivas `User` e `Group`. Ambas estão definidas como `zabbix`.

Não é mais possível configurar qual usuário o Zabbix agent usa por meio do arquivo `zabbix_agentd.conf`, porque o agent ignorará essa configuração e será executado como o usuário especificado no arquivo de serviço systemd. Para executar o Zabbix agent como root, você precisa fazer as modificações descritas abaixo.

Zabbix agent

Para substituir o usuário e o grupo padrão do Zabbix agent, execute:

```
systemctl edit zabbix-agent
```

Em seguida, adicione o seguinte conteúdo:

```
[Service]
User=root
Group=root
```

Recarregue os daemons e reinicie o serviço zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Para o **Zabbix agent**, isso reativa a funcionalidade de configurar o usuário no arquivo de configuração do agent. Agora, você precisa definir os parâmetros de configuração `User=root` e `AllowRoot=1` no arquivo de configuração.

Zabbix agent 2

Para substituir o usuário e o grupo padrão do Zabbix agent 2, execute:

```
systemctl edit zabbix-agent2
```

Em seguida, adicione o seguinte conteúdo:

```
[Service]
User=root
Group=root
```

Recarregue os daemons e reinicie o serviço zabbix-agent2:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Para **Zabbix agent2**, isso determina completamente o usuário com o qual ele é executado. Nenhuma modificação adicional é necessária.

Parâmetros de configuração: Zabbix agent

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix agent.

Especifique os valores dos parâmetros no arquivo de configuração do agent do Zabbix (`zabbix_agentd.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver todos os detalhes de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
AllowKeyRegexp	Permite a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
AllowRoot	Permite que o agent seja executado como `root`.
BufferSend	Define a frequência de envio de valores do buffer para o Zabbix server/proxy.
BufferSize	Define o número máximo de valores no buffer de memória.
DebugLevel	Define o nível de depuração.
DenyKey	Impede a execução das chaves de item que correspondem a um padrão.
DenyKeyRegexp	Impede a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
EnableRemoteCommands	Permite comandos remotos do Zabbix server. Este parâmetro está obsoleto; use `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp` em seu lugar.
HeartbeatFrequency	Define a frequência das mensagens de heartbeat em segundos.
HostInterface	Especifica a interface do host.
HostInterfaceItem	Especifica um item para obter a interface do host.
HostMetadata	Especifica os metadados do host.
HostMetadataItem	Especifica um item do Zabbix agent para obter os metadados do host.
Hostname	Especifica o nome do host.
HostnameItem	Especifica um item do Zabbix agent para obter o nome do host.
Include	Especifica um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
ListenBacklog	Define o número máximo de conexões pendentes na fila TCP.
ListenIP	Especifica o endereço IP de escuta do agent para conexões do server/proxy.
ListenPort	Especifica a porta de escuta do agent.
LoadModule	Especifica o módulo a ser carregado na inicialização do agent.
LoadModulePath	Especifica o caminho completo para o local dos módulos do agent.
LogFile	Especifica o arquivo de log do agent.
LogFileSize	Define o tamanho máximo do arquivo de log.
LogRemoteCommands	Ativa o registro de comandos shell executados como avisos.
LogType	Especifica o tipo de saída do log.
MaxLinesPerSecond	Define o número máximo de novas linhas por segundo a serem enviadas ao Zabbix server/proxy durante o processamento de verificações ativas `log` ou `logrt`.
PidFile	Especifica o arquivo PID.
RefreshActiveChecks	Especifica a frequência de atualização da lista de verificações ativas.
Server	Especifica os endereços IP ou nomes DNS dos Zabbix servers/proxies.
ServerActive	Especifica o endereço do Zabbix server/proxy ou a configuração do cluster do qual obter as verificações ativas.

Parâmetro	Descrição
SourceIP	Especifica o endereço IP de origem.
StartAgents	Define o número de instâncias pré-fork de `zabbix_agentd` para processar verificações passivas.
Timeout	Especifica por quanto tempo aguardar (em segundos) o estabelecimento da conexão e a troca de dados com o Zabbix proxy ou server.
TLSAccept	Especifica o nível de criptografia exigido para conexões de entrada.
TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados de CA de nível superior para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do agent, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	Especifica a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherAll13	Especifica a string de cifras do OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.
TLSCipherCert	Especifica a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherCert13	Especifica a string de cifras do OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.
TLSCipherPSK	Especifica a string de prioridade do GnuTLS ou a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSCipherPSK13	Especifica a string de cifras do OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSConnect	Especifica o nível de criptografia exigido para conexões com o Zabbix server ou proxy.
TLSCRLFile	Especifica o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent, usada nas comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	Especifica o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSPSKIdentity	Especifica a string de identidade da chave pré-compartilhada, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSServerCertIssuer	Especifica o emissor permitido do certificado do server (proxy).
TLSServerCertSubject	Especifica o assunto permitido do certificado do server (proxy).
UnsafeUserParameters	Permite que todos os caracteres sejam passados nos argumentos para parâmetros definidos pelo usuário.
User	Reduz os privilégios para um usuário específico e existente no sistema.
UserParameter	Especifica um parâmetro definido pelo usuário para monitoramento.
UserParameterDir	Especifica o caminho de pesquisa padrão para comandos UserParameter.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores dos arquivos de configuração fornecidos.
- Os valores aceitam variáveis de ambiente.
- O Zabbix aceita arquivos de configuração somente na codificação UTF-8, sem BOM.
- Comentários iniciados com # são aceitos somente no início da linha.

Detalhes do parâmetro

Alias

Defina um alias para uma chave de item. Ele pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros `Alias` podem estar presentes. Não são permitidos vários parâmetros com a mesma chave `Alias`.
 Chaves `Alias` diferentes podem fazer referência à mesma chave de item.
 Os 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 os 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 que processam 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 daqueles item keys que correspondem a um padrão. O padrão da key é uma expressão curinga que suporta o caractere * para corresponder a qualquer número de quaisquer caracteres.
 Várias regras de correspondência de key podem ser definidas em combinação com `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Veja também: Restricting agent checks.

AllowKeyRegexp

Permite a execução das chaves de item que correspondem a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `DenyKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra de correspondência terá prioridade. Veja também: Restringindo verificações do agent.

Exemplo:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Permite que o agent seja executado como `root`. Se estiver desativado e o agent for iniciado por `root`, o agent tentará alternar para o usuário `zabbix`. Não tem efeito se for iniciado por um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

BufferSend

Define a frequência de envio dos valores do buffer para o Zabbix server/proxy (em segundos).

Padrão: 5
 Intervalo: 1-3600

BufferSize

Define o número máximo de valores no buffer de memória.
 O agent enviará todos os dados coletados para o server ou proxy do Zabbix se o buffer estiver cheio.

Padrão: 100
 Intervalo: 2-65535

DebugLevel

Defina o nível de depuração: - *0* - informações básicas sobre a inicialização e a finalização 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

Impede a execução dos item keys que correspondem a um padrão. O padrão de chave é uma expressão curinga que aceita o caractere * para corresponder a qualquer quantidade de quaisquer caracteres.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restricting agent checks.

Padrão: `DenyKey=system.run[*]`

DenyKeyRegexp

Impede a execução dos item keys que correspondam a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá

prioridade.
 Todos os items `system.run[]` são negados por padrão (`DenyKey=system.run[*]`), a menos que outra regra `system.run[*]` seja especificada.

Consulte também: Restringindo verificações do agent.

EnableRemoteCommands

Permite comandos remotos do Zabbix server.
 Este parâmetro está **obsoleto**; use `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` ou `DenyKeyRegexp=^system\.run\[.*\]$` em seu lugar.
 É um alias interno para os parâmetros `AllowKey`, `AllowKeyRegexp`, `DenyKey` e `DenyKeyRegexp`, dependendo do valor: - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.run\[.*\]$`)

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

HeartbeatFrequency

Define a frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de active checks.
0 - mensagens de heartbeat desativadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Especifique a interface do host (endereço IP ou nome DNS) usada durante a autoregistration do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente um endereço IP ou DNS. Para mais detalhes, consulte Using DNS as default interface.

Se não for definido, o valor será obtido de `HostInterfaceItem`.

O agent emitirá um erro e não iniciará se o valor exceder o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Especifique um item para determinar a interface do host (endereço IP ou nome DNS) durante o registro automático do host.
Esse valor será usado somente se `HostInterface` não estiver definido.
Para obter mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de registro automático, 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 compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Especifique os metadados para identificar ou distinguir o host durante o registro automático do host (agent ativo).
`HostMetadata` permite distinguir os hosts além do nome do host.

Se não for definido, o valor será obtido de `HostMetadataItem`.

O agent emitirá um erro e não será iniciado se o valor especificado exceder o limite de 2034 bytes ou não for uma string UTF-8.
Quando um parâmetro espera um endereço IP ou nome DNS, os valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Especifique um item do Zabbix agent para obter metadados do host. Esta opção só é usada quando `HostMetadata` não está definido.

O valor de `HostMetadataItem` é recuperado a cada tentativa de autoregistro e usado somente no processo de autoregistro do host (agent ativo). `HostMetadataItem` permite distinguir entre hosts além do nome do host.

Parâmetros de usuário e aliases são compatíveis. O item `system.run[]` é compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

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 sejam UTF-8 válido, mas não sejam IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Nome do host

Especifique nomes de host exclusivos e sensíveis a maiúsculas e minúsculas. Uma lista delimitada por vírgulas.
 Obrigatório para verificações ativas e deve corresponder aos nomes de host conforme configurados no server.
 O valor é obtido de `HostnameItem` se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para a linha inteira.

Padrão: Definido por `HostnameItem`

HostnameItem

Especifique um item do Zabbix agent para obter o nome do host.
 Esta opção só é usada quando Hostname não está definido.
 Parâmetros de usuário ou aliases não são compatíveis, mas o item `system.run[]` é compatível independentemente dos valores de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Padrão: `system.hostname`

Include

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você o altere durante o tempo de compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_agentd.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê, portanto considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivo:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois você não pode prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir apenas arquivos relevantes de um diretório, use o curinga (*) para correspondência de padrões.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Define o número máximo de conexões pendentes na fila TCP.
 O valor padrão é uma constante codificada, que depende do sistema.
 O valor máximo suportado também depende do sistema; valores muito altos podem ser truncados silenciosamente para o 'implementation-specified maximum'.

Padrão: `SOMAXCONN`
 Intervalo: 0 - INT_MAX

ListenIP

Especifique o endereço IP de escuta do agent para conexões vindas do server/proxy. Uma lista delimitada por vírgulas.

Padrão: `0.0.0.0`

ListenPort

Especifique a porta de escuta do agent para conexões provenientes do server/proxy.

Padrão: 10050
 Intervalo: 1024-32767

LoadModule

Especifique 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 (começa com /), então `LoadModulePath` será ignorado.
 Formatos: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

É permitido incluir vários parâmetros `LoadModule`.

LoadModulePath

Especifique o caminho completo para o local dos módulos do agent.
 O padrão depende das opções de compilação.

LogFile

Especifique o arquivo de log do agent.

Obrigatório: Sim, se LogType estiver definido como `file`; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa 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 reiniciado.

Padrão: 1
 Intervalo: 0-1024

LogRemoteCommands

Ative o registro dos comandos de shell executados como avisos.
 Os comandos serão registrados somente se forem 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 - desativado, 1 - ativado

LogType

Especifique 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

Defina o número máximo de novas linhas por segundo para envio ao Zabbix server/proxy ao processar verificações ativas `log` ou `logrt`.
 O valor fornecido será substituído pelo parâmetro `maxlines`, informado na chave do item `log` ou `logrt`.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em `MaxLinesPerSecond` para localizar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

PidFile

Especifique o arquivo PID.

Padrão: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Especifique a frequência de atualização da lista de active checks, em segundos.
 Observe que, após falhar ao atualizar os active checks, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Especifique os endereços IP, opcionalmente em notação CIDR, ou os nomes DNS dos servers/proxies do Zabbix. Uma lista separada por vírgulas.
 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. Observe que "endereços IPv6 compatíveis com IPv4" (`prefixo 0000::/96`) são suportados, mas estão obsoletos conforme 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 definido explicitamente como 0

ServerActive

Especifique o endereço do server/proxy do Zabbix ou a configuração de cluster para obter verificações ativas.
 O endereço do server/proxy é um endereço IP ou nome DNS, com uma porta opcional separada por dois-pontos.
A configuração de cluster é um ou mais endereços de membros do grupo de server ou proxy separados por ponto e vírgula. Vários servers/clusters do Zabbix e proxies do Zabbix podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um proxy do Zabbix para cada server/cluster do Zabbix. Se um proxy do Zabbix for especificado, então o server/cluster do Zabbix para esse proxy não deve ser especificado.
Vários endereços delimitados por vírgula podem

ser fornecidos para usar vários servers independentes do Zabbix 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 desse 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 proxy do Zabbix:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de proxy do Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para vários 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

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy;
- estabelecer conexões ao executar alguns items (`web.page.get`, `net.tcp.port`, etc.).

StartAgents

Define o número de instâncias pré-forked de `zabbix_agentd` para processar verificações passivas.
 Se definido como 0, as verificações passivas serão desativadas e o agent não escutará em nenhuma porta TCP.

Padrão: 10
 Intervalo: 0-100

Timeout

Especifique por quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o proxy ou server do Zabbix.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do server do Zabbix;
- enviar solicitações ao server do Zabbix, incluindo solicitações de configuração de item e dados de item em active checks;
- recuperar dados de log por meio de logfile;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- ser usado por módulos do agent do Zabbix;
- servir como fallback em cenários em que um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações do agent que têm configurações de timeout ajustáveis no frontend (em nível global, de proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

Especifique o nível de criptografia exigido para conexões de entrada.
 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e em 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

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

TLSCipherPSK

Especifique a string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Especifique o nível de criptografia exigido para conexões do agent com o 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

Especifique o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada para comunicações criptografadas com o server ou proxy do Zabbix.

TLSPSKIdentity

Especifique a string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server ou proxy.

TLSServerCertIssuer

Especifique o emissor do certificado do server (proxy) permitido.

TLSServerCertSubject

Especifique o subject do certificado do server (proxy) permitido.

UnsafeUserParameters

Permite que todos os caracteres sejam passados nos 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.
Tem efeito apenas se for executado como `root` e `AllowRoot` estiver desabilitado.

Padrão: `zabbix`

UserParameter

Especifique um parâmetro definido pelo usuário para monitorar. Pode haver vários parâmetros definidos pelo usuário.
Formato: `UserParameter=<key>,<shell command>`
Observe que o comando shell não deve retornar uma string vazia nem apenas EOL. Os comandos shell podem ter caminhos relativos, se o parâmetro `UserParameterDir` estiver especificado.

Exemplo:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Especifique o caminho de pesquisa padrão para comandos `UserParameter`. Se usado, o agent alterará seu diretório de trabalho para o diretório especificado aqui antes de executar um comando. Assim, os comandos `UserParameter` podem usar 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 Zabbix agent para verificações ativas e passivas a partir da versão 2.0.0

Parâmetros de configuração: Zabbix agent (Windows)

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix agent no Windows.

Especifique os valores dos parâmetros no arquivo de configuração do agent do Zabbix (`zabbix_agentd.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver todos os detalhes de um parâmetro, clique no nome dele).
- Detalhes dos parâmetros.

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.

Parâmetro	Descrição
AllowKeyRegexp	Permite a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
BufferSend	Define a frequência de envio de valores do buffer para o Zabbix server/proxy.
BufferSize	Define o número máximo de valores no buffer de memória.
DebugLevel	Define o nível de depuração.
DenyKey	Impede a execução das chaves de item que correspondem a um padrão.
DenyKeyRegexp	Impede a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
EnableRemoteCommands	Permite comandos remotos do Zabbix server. Este parâmetro está obsoleto; use `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp` em seu lugar.
HeartbeatFrequency	Define a frequência das mensagens de heartbeat em segundos.
HostInterface	Especifica a interface do host.
HostInterfaceItem	Especifica um item para obter a interface do host.
HostMetadata	Especifica os metadados do host.
HostMetadataItem	Especifica um item do Zabbix agent para obter os metadados do host.
Hostname	Especifica o nome do host.
HostnameItem	Especifica um item do Zabbix agent para obter o nome do host.
Include	Especifica um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
ListenBacklog	Define o número máximo de conexões pendentes na fila TCP.
ListenIP	Especifica o endereço IP no qual o agent escuta conexões do server/proxy.
ListenPort	Especifica a porta na qual o agent escuta.
LogFile	Especifica o arquivo de log do agent.
LogFileSize	Define o tamanho máximo do arquivo de log.
LogRemoteCommands	Habilita o registro de comandos shell executados como avisos.
LogType	Especifica o tipo de saída do log.
MaxLinesPerSecond	Define o número máximo de novas linhas por segundo a serem enviadas ao Zabbix server/proxy durante o processamento de verificações ativas `log`, `logrt` ou `eventlog`.
PerfCounter	Define um novo parâmetro <parameter_name>, que corresponde ao valor médio do contador de desempenho do sistema <perf_counter_path> durante o período especificado <period> (em segundos).
PerfCounterEn	Define um novo parâmetro <parameter_name>, que corresponde ao valor médio do contador de desempenho do sistema <perf_counter_path> durante o período especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos dos contadores de desempenho devem estar em inglês.
RefreshActiveChecks	Especifica a frequência de atualização da lista de verificações ativas.
Server	Especifica os endereços IP ou nomes DNS dos Zabbix servers/proxies.
ServerActive	Especifica o endereço do Zabbix server/proxy ou a configuração de cluster de onde obter as verificações ativas.
SourceIP	Especifica o endereço IP de origem.
StartAgents	Define o número de instâncias pré-fork de `zabbix_agentd` para processar verificações passivas.
Timeout	Especifica por quanto tempo aguardar (em segundos) o estabelecimento da conexão e a troca de dados com o Zabbix server ou proxy.
TLSAccept	Especifica o nível de criptografia exigido para conexões de entrada.
TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da CA de nível superior para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do agent, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSConnect	Especifica o nível de criptografia exigido para conexões com o Zabbix server ou proxy.
TLSCRLFile	Especifica o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent, usada nas comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	Especifica o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSPSKIdentity	Especifica a string de identidade da chave pré-compartilhada, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSServerCertIssuer	Especifica o emissor permitido do certificado do server (proxy).
TLSServerCertSubject	Especifica o assunto permitido do certificado do server (proxy).
UnsafeUserParameters	Permite que todos os caracteres sejam passados nos argumentos dos parâmetros definidos pelo usuário.

Parâmetro	Descrição
UserParameter	Especifica um parâmetro definido pelo usuário para monitoramento.
UserParameterDir	Especifica o caminho de pesquisa padrão para comandos UserParameter.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores dos arquivos de configuração fornecidos.
- Os valores aceitam variáveis de ambiente.
- O Zabbix aceita arquivos de configuração somente na codificação UTF-8, sem BOM.
- Comentários iniciados com # são aceitos somente no início da linha.

Detalhes do parâmetro

Alias

Defina um alias para uma chave de item. Ele pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros `Alias` podem estar presentes. Não são permitidos vários parâmetros com a mesma chave `Alias`.
 Chaves `Alias` diferentes podem fazer referência à mesma chave de item.
 Os 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 low-level discovery que processam 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 daqueles item keys que correspondem a um padrão. O padrão da key é uma expressão curinga que suporta o caractere * para corresponder a qualquer número de quaisquer caracteres.
 Várias regras de correspondência de key podem ser definidas em combinação com `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Veja também: Restricting agent checks.

AllowKeyRegexp

Permite a execução dos item keys que correspondem a um padrão. O padrão da key é uma expressão regular.
 Várias regras de correspondência de keys podem ser definidas em conjunto com `DenyKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restringindo verificações do agent.

Exemplo:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Defina a frequência de envio dos valores do buffer para o Zabbix server/proxy (em segundos).

Padrão: 5
 Intervalo: 1-3600

BufferSize

Define o número máximo de valores no buffer de memória.
 O agent enviará todos os dados coletados para o server ou proxy do Zabbix se o buffer estiver cheio.

Padrão: 100
 Intervalo: 2-65535

DebugLevel

Defina o nível de depuração: - *0* - informações básicas sobre a inicialização e a finalização 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

Impede a execução dos item keys que correspondem a um padrão. O padrão da chave é uma expressão curinga que aceita o caractere * para corresponder a qualquer quantidade de quaisquer caracteres.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um a um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restringindo verificações do agent.

Padrão: `DenyKey=system.run[*]`

DenyKeyRegexp

Impede a execução dos item keys que correspondam a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade.
 Todos os itens `system.run[]` são negados por padrão (`DenyKey=system.run[*]`), a menos que outra regra `system.run[*]` seja especificada.

Consulte também: Restringindo verificações do agent.

EnableRemoteCommands

Permitir comandos remotos do Zabbix server.
 Este parâmetro está **obsoleto**; use `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` ou `DenyKeyRegexp=^system\.run\[.*\]$` em seu lugar.
 É um alias interno para os parâmetros `AllowKey`, `AllowKeyRegexp`, `DenyKey` e `DenyKeyRegexp`, dependendo do valor: - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.run\[.*\]$`)

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

HeartbeatFrequency

Define a frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de active checks.
0 - mensagens de heartbeat desativadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Especifique a interface do host (endereço IP ou nome DNS) usada durante a autoregistration do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente um endereço IP ou DNS. Para mais detalhes, consulte Using DNS as default interface.

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

Especifique um item para determinar a interface do host (endereço IP ou nome DNS) durante o registro automático do host. Esse valor será usado somente se `HostInterface` não estiver definido. Para obter mais detalhes, consulte Usando o DNS como interface padrão.

Durante uma solicitação de registro automático, 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 compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Especifique os metadados usados para identificar ou distinguir o host durante o registro automático do host (agent ativo). `HostMetadata` permite distinguir os hosts além do nome do host.

Se não for definido, o valor será obtido de `HostMetadataItem`.

O agent emitirá um erro e não será iniciado se o valor especificado exceder o limite de 2034 bytes ou não for uma string UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, os valores que forem UTF-8 válidos, mas não forem IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Não há suporte para metadados com várias linhas — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Especifique um item do Zabbix agent para obter metadados do host. Esta opção é usada somente quando `HostMetadata` não está definido.

O valor de `HostMetadataItem` é recuperado a cada tentativa de autoregistro e usado somente no processo de autoregistro do host (agent ativo). `HostMetadataItem` permite distinguir os hosts além do nome do host.

Parâmetros de usuário, contadores de desempenho e aliases são compatíveis. O item `system.run[]` é compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

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 sejam UTF-8 válidos, mas não sejam IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Nome do host

Especifique nomes de host exclusivos e sensíveis a maiúsculas e minúsculas. Uma lista delimitada por vírgulas.
 Obrigatório para verificações ativas e deve corresponder aos nomes de host conforme configurados no server. O valor é obtido de `HostnameItem` se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para a linha inteira.

Padrão: definido por HostnameItem

HostnameItem

Especifique um item do Zabbix agent 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 compatíveis, mas o item `system.run[]` é compatível independentemente dos valores de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.
Consulte também uma descrição mais detalhada.

Padrão: `system.hostname`

Incluir

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração (localizado em `C:\Program Files\Zabbix Agent` por padrão, se o Zabbix agent tiver sido instalado usando os pacotes do instalador MSI do Windows; localizado na pasta especificada durante a instalação, se o Zabbix agent tiver sido instalado como um arquivo zip).

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_agentd.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê. Portanto, considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter a sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivos:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir somente os arquivos relevantes de um diretório, use o caractere curinga (*) para correspondência de padrões.

Exemplo:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Define o número máximo de conexões pendentes na fila TCP.
 O valor padrão é uma constante codificada, que depende do sistema.
 O valor máximo suportado também depende do sistema; valores muito altos podem ser truncados silenciosamente para o ”máximo especificado pela implementação”.

Padrão: SOMAXCONN
 Intervalo: 0 - INT_MAX

ListenIP

Especifique o endereço IP de escuta do agent para conexões provenientes do server/proxy. Uma lista delimitada por vírgulas.

Padrão: 0.0.0.0

ListenPort

Especifique a porta de escuta do agent para conexões provenientes do server/proxy.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

Especifique o arquivo de log do agent.

Padrão: c:\zabbix_agentd.log
 Obrigatório: Sim, se LogType estiver definido como file; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa 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 reiniciado.

Padrão: 1
 Intervalo: 0-1024

LogRemoteCommands

Ative o registro dos comandos de shell executados como avisos.
 Os comandos serão registrados somente se forem executados remotamente.
 Entradas de log não serão criadas se system.run[] for iniciado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desativado, 1 - ativado

LogType

Especifique 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

Define o número máximo de novas linhas por segundo para envio ao server/proxy do Zabbix ao processar verificações ativas log, logrt ou eventlog.
 O valor fornecido será substituído pelo parâmetro maxlines, informado na chave do item log, logrt ou eventlog.
Nota: o Zabbix processará 10 vezes mais novas linhas do que o definido em MaxLinesPerSecond para localizar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

PerfCounter

Defina um novo parâmetro <parameter_name> que seja o valor médio do 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 quiser receber o número médio de interrupções do processador por segundo no último minuto, você 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 do contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos do perfcounter devem estar em inglês. Suportado apenas no **Windows Server 2008/Vista** e versões posteriores.
Sintaxe: <parameter_name>,”<perf_counter_path>”,<period>

Por exemplo, se você quiser receber o número médio de interrupções do processador por segundo no último minuto, você pode definir um novo parâmetro "interrupts" da seguinte forma:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Observe as aspas duplas ao redor do caminho do contador de desempenho. O nome do parâmetro (interrupts) deve ser usado como a chave do item ao criar um item. As amostras para calcular o valor médio serão coletadas a cada segundo.
Você pode encontrar a lista de strings em inglês visualizando a seguinte chave do registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Especifique a frequência de atualização da lista de verificações ativas, 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

Especifique os endereços IP, opcionalmente em notação CIDR, ou os nomes DNS dos servers/proxies do Zabbix. Uma lista separada por vírgulas.
 As conexões de entrada serão aceitas somente 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" (`0000::/96 prefix`) são suportados, mas estão obsoletos conforme 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 definido explicitamente como 0

ServerActive

Especifique o endereço do server/proxy do Zabbix ou a configuração de cluster para obter checks ativos.
 O endereço do server/proxy é um endereço IP ou nome DNS, com porta opcional separada por dois-pontos.
A configuração de cluster é um ou mais endereços de membros de grupo de server ou proxy separados por ponto e vírgula. Vários servers/clusters do Zabbix e proxies do Zabbix podem ser especificados, separados por vírgula. A menos que estejam sendo usados grupos de proxy, não deve ser especificado mais de um proxy do Zabbix para cada server/cluster do Zabbix. Se um proxy do Zabbix for especificado, então o server/cluster do Zabbix para esse proxy não deve ser especificado.
Vários endereços delimitados por vírgula podem ser fornecidos para usar vários servers independentes do Zabbix em paralelo. Espaços são permitidos.
Se a porta não for especificada, será usada a porta padrão.
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, os checks ativos serão desativados.

Exemplo para proxy do Zabbix:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de proxy do Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para vários 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: O número de servers ativos listados em `ServerActive` mais o número de instâncias pré-fork para checks passivos especificadas em `StartAgents` deve ser menor que 64.

SourceIP

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy;
- estabelecer conexões ao executar alguns items (`web.page.get`, `net.tcp.port`, etc.).

StartAgents

Define o número de instâncias pré-forked de `zabbix_agentd` para processar verificações passivas.
 Se definido como 0, as verificações passivas são desativadas e o agent não escutará em nenhuma porta TCP.

Padrão: 10
 Intervalo: 0-100 (*)

Timeout

Especifique por quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o Zabbix proxy ou server.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do Zabbix server;
- enviar solicitações ao Zabbix server, incluindo solicitações de configuração de item e dados de item em active checks;
- recuperar dados de log por meio do monitoramento de logfile ou do Windows event log;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- usado por módulos do Zabbix agent;
- servir como fallback em cenários em que um server ou proxy mais antigo que a versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações de agent que têm configurações de timeout ajustáveis no frontend (em nível global, de proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

Especifique o nível de criptografia exigido para conexões de entrada.
 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSConnect

Especifique o nível de criptografia exigido para conexões do agent com o 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

Especifique o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada para comunicações criptografadas com o server ou proxy do Zabbix.

TLSPSKIdentity

Especifique a string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server ou proxy.

TLSServerCertIssuer

Especifique o emissor do certificado do server (proxy) permitido.

TLSServerCertSubject

Especifique o subject do certificado permitido do server (proxy).

UnsafeUserParameters

Permite que todos os caracteres sejam passados nos 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

Especifique um parâmetro definido pelo usuário para monitorar. Pode haver vários parâmetros definidos pelo usuário.
Formato: `UserParameter=<key>,<shell command>`
Observe que o comando de shell não deve retornar uma string vazia nem apenas EOL. Os comandos de shell podem ter caminhos relativos, se o parâmetro `UserParameterDir` estiver especificado.

Exemplo:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Especifique o caminho de pesquisa padrão para comandos `UserParameter`. Se usado, o agent alterará seu diretório de trabalho para o diretório especificado aqui antes de executar um comando. Assim, os comandos `UserParameter` podem usar um prefixo relativo `./` em vez de um caminho completo. Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=C:\opt\myscripts
```

Zabbix agent no Microsoft Windows

Configurando o agent

Ambas as gerações de agents do 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 Zabbix agent ou várias instâncias do agent em um host Microsoft Windows. Uma única instância pode usar uma das seguintes opções:

- 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 várias 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 de código-fonte do Zabbix como:

- `conf/zabbix_agentd.conf` para o Zabbix agent;
- `conf/zabbix_agent2.conf` para o Zabbix agent2.

Se quiser instalar o Zabbix agent/agent 2 para Windows como um serviço a partir de um arquivo sem especificar explicitamente o arquivo de configuração, antes de instalar o agent:

- `conf/zabbix_agentd.conf` deve ser copiado manualmente para o diretório onde `zabbix_agentd.exe` será instalado;
- `conf/zabbix_agent2.conf` e o diretório `conf/zabbix_agent2.d` devem ser copiados manualmente para o diretório onde `zabbix_agent2.exe` será instalado.

Consulte as opções do arquivo de configuração do agent do Zabbix ou do arquivo de configuração do agent 2 do Zabbix para obter detalhes sobre a configuração do agent do Zabbix para Windows.

Parâmetro Hostname

Para realizar active checks em um host, o Zabbix agent precisa ter o hostname definido. Além disso, o valor do hostname definido no lado do agent deve corresponder exatamente ao *Host name* 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 serão usados se nenhum desses parâmetros for especificado.

O valor padrão do parâmetro `HostnameItem` é o valor retornado pela chave de agent `system.hostname`. No Windows, ela 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 será retornado.

O valor padrão de `Hostname` é o valor retornado pelo parâmetro `HostnameItem`. Assim, na prática, se ambos os parâmetros não forem especificados, o hostname real será o nome NetBIOS do host; o Zabbix agent usará o nome do host NetBIOS para obter a lista de active checks do Zabbix server e enviar os resultados para ele.

A chave `system.hostname` aceita dois parâmetros opcionais - *type* e *transform*.

Type determina o tipo de nome que o item deve retornar:

- *netbios* (padrão) - retorna o nome do host NetBIOS, limitado a 15 caracteres e somente em LETRAS MAIÚSCULAS;
- *host* - diferencia maiúsculas de minúsculas e retorna o nome completo e real do host Windows (sem um domínio);
- *shorthost* - retorna a parte do hostname anterior ao 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 a capitalização original;
- *lower* - converte o texto para letras minúsculas.

Assim, para simplificar a configuração do arquivo `zabbix_agentd.conf` e torná-la uniforme, diferentes abordagens podem ser usadas:

1. Deixe os parâmetros `Hostname` ou `HostnameItem` indefinidos, e o Zabbix agent usará o nome do host NetBIOS como hostname.
2. Deixe o parâmetro `Hostname` indefinido e defina `HostnameItem` da seguinte forma:
    - `HostnameItem=system.hostname[host]` - para que o Zabbix agent use o nome completo e real do host Windows (diferencia maiúsculas de minúsculas) como hostname.
    - `HostnameItem=system.hostname[shorthost,lower]` - para que o Zabbix agent use somente a parte do hostname anterior ao primeiro ponto, convertida para letras minúsculas.
    - `HostnameItem=system.hostname[fqdn]` - para que o Zabbix agent use o Fully Qualified Domain Name como hostname.

O nome do host também é usado como parte do nome do serviço do Windows, que é usado para instalar, iniciar, parar e desinstalar o serviço do Windows. Por exemplo, se o arquivo de configuração do Zabbix agent especificar `Hostname=Windows_db_server`, 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 Zabbix agent, cada instância deve usar um nome de host diferente.

Instalando o agent como serviço do Windows

Antes de instalar o agent, copie `conf/zabbix_agentd.conf` manualmente para o diretório onde `zabbix_agentd.exe` será instalado.

Para instalar uma única instância do Zabbix agent com o arquivo de configuração padrão:

```
zabbix_agentd.exe --install
```

Attention:
Em um sistema de 64 bits, é necessária uma versão de 64 bits do Zabbix agent para que todas as verificações relacionadas à execução de processos de 64 bits funcionem corretamente.

Se você quiser usar um arquivo de configuração diferente do padrão, deve usar o seguinte comando para a instalação do serviço:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

Um caminho completo para o arquivo de configuração deve ser especificado.

Várias instâncias do Zabbix agent podem ser instaladas como serviços desta forma:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

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 fazê-lo pela linha de comando.

Para iniciar uma única instância do Zabbix agent com o arquivo de configuração padrão:

```
zabbix_agentd.exe --start
```

Para iniciar uma única instância do Zabbix agent com outro arquivo de configuração:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Para iniciar uma das várias instâncias do Zabbix agent:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --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 Zabbix agent iniciada com o arquivo de configuração padrão:

```
zabbix_agentd.exe --stop
```

Para parar uma única instância do Zabbix agent iniciada com outro arquivo de configuração:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Para parar uma das várias instâncias do Zabbix agent:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Desinstalando o serviço do Windows do agent

Para desinstalar uma única instância do Zabbix agent usando o arquivo de configuração padrão:

```
zabbix_agentd.exe --uninstall
```

Para desinstalar uma única instância do Zabbix agent usando um arquivo de configuração não padrão:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Para desinstalar várias instâncias do Zabbix agent dos serviços do Windows:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

Limitações

O agent do Zabbix para Windows não oferece suporte a configurações não padrão do Windows em que as CPUs são distribuídas de forma não uniforme entre 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.

Zabbix agent no IBM AIX

Visão geral

Os binários pré-compilados do Zabbix agent para IBM AIX estão disponíveis para download como um arquivo `.tar.gz`. Esta página descreve como instalar o agent a partir do arquivo, executá-lo manualmente ou com o System Resource Controller (SRC), iniciá-lo na inicialização do sistema e removê-lo do SRC.

Instalando o agent

O comando `tar` do AIX não oferece suporte à opção `-z`; portanto, descompacte o arquivo com `gunzip`. Use a opção `-o` do `tar` para que os arquivos extraídos pertençam ao usuário que os extrai, e não aos IDs numéricos de usuário e grupo armazenados no arquivo. Por exemplo, para colocar o agent em `/opt/zabbix` (execute como root; `/opt/zabbix` ainda não deve existir):

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

No arquivo do AIX, o binário do agent está localizado em `usr/sbin/` e o arquivo de configuração em `etc/zabbix/`. Antes de iniciar o agent, ajuste `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` conforme necessário (pelo menos os parâmetros `Server`, `ServerActive` e `Hostname`).

Se iniciado como root, o agent muda para o usuário `zabbix` (ou para o usuário especificado pelo parâmetro `User`), a menos que `AllowRoot=1` esteja definido no arquivo de configuração. Esse usuário deve existir no sistema; caso contrário, o agent não será iniciado. Se o usuário `zabbix` ainda não existir, crie-o junto com um grupo de mesmo nome:

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Os atributos `login=false` e `rlogin=false` impedem que a conta seja usada para login; o agent ainda pode ser executado como esse usuário.

Iniciando o agent manualmente

O agent pode ser iniciado manualmente:

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Executando o agent com SRC

Registrando o agent

O agent também pode ser registrado no System Resource Controller (SRC). Quando iniciado pelo SRC, o agent deve ser executado no modo foreground (a opção `-f`). Caso contrário, o SRC perderá o controle do processo do agent e não poderá interrompê-lo. Por exemplo (execute como root):

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

Neste exemplo:

- `-s` define o nome do subsistema usado com os comandos do SRC;
- `-p` define o caminho completo para o binário do agent;
- `-a` define os argumentos de linha de comando do agent como uma única string entre aspas; o SRC divide essa string em argumentos separados ao iniciar o agent (somente uma opção `-a` é aceita);
- `-u` define o usuário com o qual o agent é executado; o nome do usuário é convertido em um ID de usuário numérico quando o subsistema é criado, portanto, o usuário já deve existir;
- `-G` adiciona o agent ao grupo de subsistemas `tcpip`, para que os comandos do grupo também se apliquem a ele (por exemplo, `stopsrc -g tcpip` também interrompe o agent);
- `-S -n 15 -f 9` faz com que o SRC controle o agent com sinais: `stopsrc` envia `SIGTERM` (15) para uma interrupção normal e `SIGKILL` (9) para uma interrupção forçada (`stopsrc -f`);
- `-R` faz com que o SRC reinicie o agent se ele for encerrado de forma anormal (por exemplo, se um de seus processos falhar); o agent não é reiniciado após `stopsrc`.

Como o ID de usuário é armazenado na definição do subsistema, atualize a definição se o usuário `zabbix` for recriado com um ID de usuário diferente:

```
chssys -s zabbix_agentd -u zabbix
```

Iniciando e parando o agent

O agent pode ser iniciado, consultado e parado com `startsrc`, `lssrc` e `stopsrc`:

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Evite a parada forçada (`stopsrc -f`). Ela encerra apenas o processo principal do agent; os outros processos do agent continuam em execução e mantêm a porta de escuta aberta. Portanto, o agent não poderá ser iniciado novamente até que esses processos sejam encerrados manualmente (por exemplo, com `kill -9`).

Iniciando o agent na inicialização do sistema

Registrar o agent no SRC não faz com que ele seja iniciado na inicialização do sistema. Para iniciar o agent automaticamente durante a inicialização, adicione uma entrada ao `/etc/inittab`:

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Removendo o agent do SRC

Para remover o agent da inicialização do sistema e do SRC:

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Agent 2

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

- Reduzir o número de conexões TCP;
- Fornecer concorrência de verificações aprimorada;
- Ser facilmente extensível com plugins que fornecem verificações simples com código mínimo e oferecem suporte a verificações complexas, compostas por scripts de longa execução e coleta de dados independente com relatórios periódicos;
- Funcionar como substituto do Zabbix agent, com suporte a todos os recursos anteriores.

Todas as métricas no Zabbix agent 2 são coletadas por plugins.

Esses recursos são específicos dos plugins:

- Intervalos agendados e flexíveis para verificações passivas e ativas
- Gerenciamento da fila de tarefas com relação ao agendamento e à concorrência de tarefas
- Timeouts no nível do plugin
- Verificação de compatibilidade do Zabbix agent 2 e seus plugins na inicialização

Existem dois tipos de plugins:

- Plugins integrados
- Plugins carregáveis

Você também pode criar seus próprios plugins.

Plugins criados sob medida não precisam ser incorporados ao agent 2. Você pode adicioná-los como plugins carregáveis, simplificando assim a criação de plugins para novas métricas.

Plugins carregáveis podem ser adicionados em tempo de execução, sem recompilar o Zabbix agent 2.

Veja também:

- Comparação entre agent e agent 2
- Parâmetros de configuração: Zabbix agent 2
- Parâmetros de configuração: Zabbix agent 2 (Windows)
- Parâmetros de configuração: Plugins
- Protocolo de plugin do Zabbix agent 2

Verificações passivas e ativas

O Zabbix agent 2 é compatível com verificações passivas e ativas, assim como o Zabbix agent. Além disso, as verificações ativas do Zabbix agent 2 são compatíveis com intervalos flexíveis/de agendamento e concorrência de verificações em um único server ativo.

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

Concorrência de verificações

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 no próprio código (1000 sendo o padrão) que pode ser reduzida usando a configuração `Plugins.<PluginName>.System.Capacity=N` no parâmetro de configuração `Plugins`.

Plataformas suportadas Para as plataformas suportadas, consulte a página Requisitos.

Agent 2 em sistemas Unix-like O Zabbix agent 2 em sistemas Unix-like é executado no host sendo monitorado.

Instalação

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

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

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

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

Instalação

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

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

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

Consulte a página Zabbix agent no Microsoft Windows para detalhes adicionais sobre a instalação do Zabbix agent 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 diferente do arquivo padrão.
`-f --foreground`	Executar o Zabbix agent em primeiro plano (padrão: true).
`-p --print`	Imprimir os itens conhecidos e sair. Observe que, para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se ele não estiver no local padrão).
`-t --test <item key>`	Testar o item especificado e sair. Observe que, para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se ele não estiver no local padrão).

Parâmetro	Descrição
`-T --test-config`	Validar o arquivo de configuração e sair.
`-h --help`	Imprimir as informações de ajuda e sair.
`-v --verbose`	Imprimir informações de depuração. Use esta opção com as opções `-p` e `-t`.
`-V --version`	Imprimir a versão do agent e as informações de licença.
`-R --runtime-control <option>`	Executar funções administrativas. Consulte controle em tempo de execução.
Somente agent Windows	
`-m --multiple-agents`	Usar várias instâncias do agent (com as opções `-i`, `-d`, `-s`, `-x`). Para distinguir os nomes de serviço das instâncias, cada nome de serviço incluirá o valor Hostname do arquivo de configuração especificado.
`-S --startup-type <value>`	Definir o tipo de inicialização do serviço do Zabbix Windows agent. Valores permitidos: `automatic` - *(padrão)* iniciar o serviço automaticamente na inicialização do Windows; `delayed` - atrasar a inicialização do serviço até que os serviços iniciados automaticamente tenham concluído a inicialização; `manual` - iniciar o serviço manualmente (por um usuário ou aplicativo); `disabled` - desabilitar o serviço, impedindo que ele seja 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`	Instalar o Zabbix Windows agent como serviço.
`-d --uninstall`	Desinstalar o serviço do Zabbix Windows agent.
`-s --start`	Iniciar o serviço do Zabbix Windows agent.
`-x --stop`	Parar o serviço do Zabbix Windows agent.

Exemplos específicos de uso dos parâmetros de linha de comando:

- Imprimir todos os itens integrados do agent com seus valores.
- Testar um parâmetro de usuário com a chave `mysql.ping` definida no arquivo de configuração especificado.
- Instalar um serviço do Zabbix Agent para Windows usando o caminho padrão para o arquivo de configuração.
- Modificar o tipo de inicialização de um serviço do Zabbix Agent para Windows instalado usando o arquivo de configuração localizado na mesma pasta que o executável do agent.

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

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.
`periodic_prof_enable`	Ativa a criação periódica de perfis. Todas as métricas coletam dados no momento e os gravam, exceto os dados da CPU. Os dados periódicos da CPU são coletados ao longo do tempo em um arquivo temporário oculto. Quando o intervalo termina, o arquivo é publicado como um perfil da CPU, e o número configurado de arquivos concluídos é mantido. O arquivo temporário não é contabilizado em `ProfilerMaxFilesPerProfile`.
`periodic_prof_disable`	Desativa a criação periódica de perfis.
`periodic_prof_execute`	Grava os perfis após uma amostragem de 5 segundos. Quando a criação periódica de perfis está ativada, este comando encerra o intervalo atual da CPU, grava um perfil separado da CPU de 5 segundos e inicia um novo arquivo periódico da CPU. Os perfis solicitados explicitamente são alternados de acordo com `ProfilerMaxFilesPerProfile` quando o comando é concluído. Enquanto este comando coleta perfis, todos os comandos de tempo de execução do profiler são rejeitados com "profiler is busy".
`periodic_prof_set_interval`	Definir o intervalo de criação de perfis.

Opção	Descrição
`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:

- Aumentar o nível de log do agent 2.
- Imprimir as opções de controle em tempo de execução.

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

Arquivo de configuração O caminho padrão para o arquivo de configuração depende do sistema operacional e do método de instalação do Zabbix:

- No UNIX, o padrão é:
  - Para instalação a partir do código-fonte: `/usr/local/etc/zabbix_agent2.conf`, ou conforme definido pelas variáveis de compilação `--sysconfdir` ou `--prefix`.
  - Para instalação a partir de pacotes: `/etc/zabbix/zabbix_agent2.conf`.
- No Windows, o padrão é `C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf`.

Se uma única instância do agent estiver em execução em um host, ela poderá usar o arquivo de configuração padrão ou um arquivo de configuração especificado na linha de comando. No caso de várias instâncias, cada instância do agent deverá ter seu próprio arquivo de configuração (uma das instâncias poderá usar o arquivo de configuração padrão).

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

Parâmetros		Descrição
Parâmetros de configuração específicos do Zabbix agent 2		
	`ControlSocket`	O caminho do socket de controle em tempo de execução. O Zabbix agent 2 usa um socket de controle para comandos em tempo de execução.
	`EnablePersistentBuffer`, `PersistentBufferFile`, `PersistentBufferPeriod`	Esses parâmetros são usados para configurar o armazenamento persistente no Zabbix agent 2 para itens ativos.
	`ForceActiveChecksOnStart`	Determina se o agent deve executar verificações ativas imediatamente após a reinicialização ou distribuí-las uniformemente ao longo do tempo.
	`Plugins`	Os plugins podem ter seus próprios parâmetros, no formato `Plugins.<Plugin name>.<Parameter>=<value>`. Um parâmetro comum de plugin é `System.Capacity`, que define o limite de verificações que podem ser executadas simultaneamente.
	`StatusPort`	A porta na qual o Zabbix agent 2 ficará escutando para solicitações de status HTTP e para a exibição de uma lista de plugins configurados e de alguns parâmetros internos.

Parâmetros		Descrição
Parâmetros de configuração específicos do Zabbix agent não compatíveis com o Zabbix agent 2		
	`AllowRoot`, `User`	Não são compatíveis com o Zabbix agent 2 porque ele não oferece suporte à execução como daemon.
	`EnableRemoteCommands`	Obsoleto; os parâmetros `AllowKey`, `AllowKeyRegexp`, `DenyKey` e `DenyKeyRegexp` são usados em seu lugar em ambos os agents.
	`ListenBacklog`	Usado no Zabbix agent para definir o número máximo de conexões pendentes na fila TCP. Não é compatível com o Zabbix agent 2.
	`LoadModule`, `LoadModulePath`	Módulos carregáveis não são compatíveis com o Zabbix agent 2.
	`MaxLinesPerSecond`, `LogRemoteCommands`	No Zabbix agent 2, `Plugins.Log.MaxLinesPerSecond` e `Plugins.SystemRun.LogRemoteCommands` são usados em seu lugar.
	`StartAgents`	Esse parâmetro é usado no Zabbix agent para aumentar a simultaneidade de verificações passivas ou desativá-las. No Zabbix agent 2, a simultaneidade é configurada no nível do plugin e pode ser limitada por uma configuração de capacidade. As verificações passivas são desativadas no Zabbix agent 2 se o parâmetro `Server` não for especificado.

Para obter mais detalhes, consulte as opções do arquivo de configuração do Zabbix agent 2.

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

Nesse caso, o Zabbix fornece mutexes para bloqueio no OpenSSL. Se uma operação de bloqueio ou desbloqueio de mutex falhar, uma mensagem de erro será exibida no fluxo de erro padrão (STDERR) e o Agent 2 encerrará com o código de retorno 2 ou 3, respectivamente.

Comparação entre Agent e agent 2

Esta seção descreve as diferenças entre o Zabbix agent e o Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Linguagem de programação	C	Go com algumas partes em C
Daemonização	sim	apenas pelo systemd (sim no Windows)
Extensões suportadas	Módulos carregáveis personalizados em C.	plugins personalizados em Go.
Requisitos		

Parameter	Zabbix agent	Zabbix agent 2
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 o XP	Linux, Windows: todas as versões desktop e server, nas quais uma versão compatível do Go possa ser instalada.
Bibliotecas criptográficas 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 (algumas limitações se aplicam; veja a página Encryption 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 separado de verificação ativa para cada registro de server/proxy.	Processo único com threads criadas automaticamente. O número máximo de threads é determinado pela variável de ambiente `GOMAXPROCS`.
Métricas	**UNIX**: veja uma lista de items suportados. **Windows**: veja uma lista de items adicionais específicos do Windows.	**UNIX**: Todas as métricas suportadas pelo Zabbix agent. Além disso, o agent 2 fornece uma 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 Zabbix agent e também verificações `net.tcp.service*` de HTTPS, LDAP. Além disso, o agent 2 fornece uma solução de monitoramento nativa do Zabbix para: PostgreSQL, Redis.
Concorrência	As verificações ativas para um único server são executadas sequencialmente.	Verificações de diferentes plugins ou várias verificações dentro de um único plugin podem ser executadas simultaneamente.
Traps de terceiros	não	sim
Recursos adicionais		
Armazenamento persistente	não	sim
Arquivos persistentes para métricas `log*[]`	sim (apenas no Unix)	não
Envio 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 fica cheio; portanto, o parâmetro `BufferSize` deve ser pelo menos `MaxLinesPerSecond` x 2.
Altera o usuário em tempo de execução	sim (apenas sistemas do tipo Unix)	não (controlado pelo systemd)
Suites de cifras configuráveis pelo usuário	sim	não

Veja também:

- *Descrição dos processos do Zabbix*: Zabbix agent, Zabbix agent 2
- *Parâmetros de configuração*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Plugins integrados

Esses plugins são incorporados ao Zabbix agent 2 e estão disponíveis imediatamente, sem necessidade de configuração adicional. 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 têm os mesmos parâmetros que as chaves do Zabbix agent.
CPU	Monitoramento da CPU do sistema (número de CPUs/núcleos de CPU, CPUs descobertas, percentual de utilização).	system.cpu.discovery, system.cpu.num, system.cpu.util	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
Docker	Monitoramento de contêineres 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 têm os mesmos parâmetros que as chaves do Zabbix agent.
Kernel	Monitoramento do kernel.	kernel.maxfiles, kernel.maxproc	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
Log	Monitoramento de arquivos de log.	log, log.count, logrt, logrt.count	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent. Veja também: Parâmetros de configuração do plugin (Unix/Windows)
Memcached	Monitoramento do server Memcached.	memcached.ping, memcached.stats	
Modbus	Lê dados Modbus.	modbus.get	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
MQTT	Recebe valores publicados de tópicos MQTT.	mqtt.get	Para configurar uma conexão criptografada com o broker MQTT, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros da chave do item.
MySQL	Monitoramento do MySQL e de seus forks.	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	Para configurar uma conexão criptografada com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros da chave do item.
NetIf	Monitoramento de interfaces de rede.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
Oracle	Monitoramento do Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Instale o Oracle Instant Client antes de usar o plugin.
Proc	Percentual de utilização de CPU do processo.	proc.cpu.util	A chave suportada tem os mesmos parâmetros que a chave do Zabbix agent.
Redis	Monitoramento do server Redis.	redis.config, redis.info, redis.ping, redis.slowlog.count	Para configurar uma conexão criptografada com o Redis, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Os parâmetros TLS não podem ser passados como parâmetros da chave do item. Observe que uma configuração TLS incorreta ou inválida pode impedir que o Zabbix agent 2 seja iniciado; portanto, verifique os arquivos de certificado, permissões e caminhos antes de habilitar o TLS.

Smart	Monitoramento S.M.A.R.T.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	A versão mínima exigida do smartctl é 7.1. São necessários privilégios de sudo/root para o smartctl para o usuário que executa o Zabbix agent 2. O plugin usa 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 somente em Linux/Windows, tanto como verificação passiva quanto ativa.
			Veja também:
			Parâmetros de configuração

SW	Listagem de pacotes instalados.	system.sw.packages, system.sw.packages.get	As chaves suportadas têm os mesmos parâmetros que a chave do Zabbix agent.
Swap	Tamanho do espaço de swap em bytes/percentual.	system.swap.size	A chave suportada tem os mesmos parâmetros que a chave do Zabbix agent.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
SystemRun	Executa o comando especificado.	system.run	A chave suportada tem os mesmos parâmetros que a 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 da disponibilidade de conexão TCP.	net.tcp.port	A chave suportada tem os mesmos parâmetros que a chave do Zabbix agent.
UDP	Monitoramento da disponibilidade e do desempenho dos serviços UDP.	net.udp.service, net.udp.service.perf	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
Uname	Recuperação de informações sobre o sistema.	system.hostname, system.sw.arch, system.uname	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
Uptime	Coleta de métricas de tempo de atividade do sistema.	system.uptime	A chave suportada tem os mesmos parâmetros que a chave do Zabbix agent.
VFSDev	Coleta de métricas de VFS.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	As chaves suportadas têm os mesmos parâmetros que as 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 têm os mesmos parâmetros que as chaves do Zabbix agent.
ZabbixAsync	Coleta assíncrona de métricas.	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.
ZabbixStats	Métricas internas do server/proxy Zabbix ou número de items atrasados em uma fila.	zabbix.stats	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
ZabbixSync	Coleta síncrona de métricas.	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	As chaves suportadas têm os mesmos parâmetros que as chaves do Zabbix agent.

Plugins carregáveis

Os plugins carregáveis não estão disponíveis por padrão no Zabbix agent 2 e precisam ser instalados separadamente:

- No Linux, você pode usar pacotes (por exemplo, instalar o Ember+ no Ubuntu com `apt install zabbix-agent2-plugin-ember-p`) ou compilar plugins
- No Windows, você pode instalar a partir do MSI ou compilar plugins

Attention:
Antes de instalar um plugin, verifique o arquivo README correspondente. Ele pode conter requisitos específicos e instruções de instalação.

Clique no nome do plugin para acessar o repositório do plugin, que contém o arquivo README 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	Consulte também Parâmetros de configuração do plugin Ceph.
Ember+	Monitoramento do Ember+.	ember.get	Consulte 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.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	Para configurar conexões criptografadas com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como parâmetros de uma sessão nomeada. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Consulte 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 uma conexão criptografada com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como parâmetros de uma sessão nomeada ou padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Consulte 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	Consulte também Parâmetros de configuração do plugin NVIDIA GPU.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
PostgreSQL	Monitoramento do PostgreSQL e de 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 uma sessão nomeada ou padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Consulte também Parâmetros de configuração do plugin PostgreSQL.

Note:
Plugins carregáveis, quando iniciados com:
 - `-V --version` - exibe a versão do plugin e as informações de licença;
 - `-h --help` - exibe as informações de ajuda.
 - `-t, --test <item key>` - inicia o plugin para testes (a configuração do plugin é ignorada).

Parâmetros de configuração: Zabbix agent 2

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix agent 2.

O Zabbix agent 2 é uma nova geração do Zabbix agent e pode ser usado no lugar do Zabbix agent.

Especifique os valores dos parâmetros no arquivo de configuração do agent 2 do Zabbix (`zabbix_agent2.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
AllowKeyRegexp	Permite a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
BufferSend	Define a frequência de envio de valores do buffer para o Zabbix server/proxy.
BufferSize	Define o número máximo de valores no buffer de memória.
ControlSocket	Especifica o socket de controle para enviar comandos em tempo de execução com a opção '-R'.
DebugLevel	Define o nível de depuração.
DenyKey	Impede a execução das chaves de item que correspondem a um padrão.
DenyKeyRegexp	Impede a execução das chaves de item que correspondem a um padrão baseado em expressão regular.

Parâmetro	Descrição
EnablePersistentBuffer	Habilita o uso de armazenamento persistente local para itens ativos.
EnableProfiler	Habilita a criação periódica de perfil do Go na inicialização do agent.
ForceActiveChecksOnStart	Executa verificações ativas imediatamente após a reinicialização para a primeira configuração recebida.
HeartbeatFrequency	Define a frequência das mensagens de heartbeat em segundos.
HostInterface	Especifica a interface do host.
HostInterfaceItem	Especifica um item para obter a interface do host.
HostMetadata	Especifica os metadados do host.
HostMetadataItem	Especifica um item do Zabbix agent para obter os metadados do host.
Hostname	Especifica o nome do host.
HostnameItem	Especifica um item do Zabbix agent para obter o nome do host.
Include	Especifica um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
ListenIP	Especifica o endereço IP de escuta do agent para conexões do server/proxy.
ListenPort	Especifica a porta de escuta do agent.
LogFile	Especifica o arquivo de log do agent.
LogFileSize	Define o tamanho máximo do arquivo de log.
LogType	Especifica o tipo de saída do log.
PersistentBufferFile	Especifica o arquivo usado para manter o banco de dados SQLite.
PersistentBufferPeriod	Define o período de armazenamento de dados quando não há conexão com o Zabbix server ou proxy.
PidFile	Especifica o arquivo PID.
Plugins.<PluginName>.System.Capacity	Define o limite de verificações por plugin que podem ser executadas simultaneamente.
Plugins.Log.MaxLinesPerSecond	Define o número máximo de novas linhas por segundo a serem enviadas ao Zabbix server/proxy ao processar verificações ativas `log` ou `logrt`.
Plugins.SystemRun.LogRemoteCommands	Habilita o registro dos comandos shell executados como avisos.
PluginSocket	Especifica o caminho para o socket UNIX usado na comunicação com plugins carregáveis.
PluginTimeout	Define o tempo limite para conexões com plugins carregáveis.
ProfilerDir	O diretório onde os dados de criação de perfil serão gravados.
ProfilerMaxFilesPerProfile	O número máximo de arquivos a serem mantidos para cada tipo de perfil.
ProfilerInterval	O intervalo de gravação dos dados de criação de perfil.
RefreshActiveChecks	Especifica a frequência de atualização da lista de verificações ativas.
Server	Especifica os endereços IP ou nomes DNS dos Zabbix servers/proxies.
ServerActive	Especifica o endereço do Zabbix server/proxy ou a configuração do cluster de onde obter as verificações ativas.
SourceIP	Especifica o endereço IP de origem.
StatusPort	Especifica a porta para escutar solicitações de status HTTP (`http://localhost:<port>/status`).
Timeout	Especifica por quanto tempo aguardar (em segundos) o estabelecimento da conexão e a troca de dados com o Zabbix server ou proxy.
TLSAccept	Especifica o nível de criptografia exigido para conexões de entrada.
TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados de CA de nível superior para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do agent, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	Especifica a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados e PSK.
TLSCipherAll13	Especifica a string de cifras do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou posterior. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados e PSK.
TLSCipherCert	Especifica a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados.
TLSCipherCert13	Especifica a string de cifras do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou posterior. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados.
TLSCipherPSK	Especifica a string de cifras do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSCipherPSK13	Especifica a string de cifras do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou posterior. Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSConnect	Especifica o nível de criptografia exigido para conexões com o Zabbix server ou proxy.
TLSCRLFile	Especifica o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent, usada nas comunicações criptografadas entre componentes do Zabbix.

Parâmetro	Descrição
TLSPSKFile	Especifica o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSPSKIdentity	Especifica a string de identidade da chave pré-compartilhada, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSServerCertIssuer	Especifica o emissor permitido do certificado do server (proxy).
TLSServerCertSubject	Especifica o assunto permitido do certificado do server (proxy).
UnsafeUserParameters	Permite que todos os caracteres sejam passados nos argumentos dos parâmetros definidos pelo usuário.
UserParameter	Especifica um parâmetro definido pelo usuário para monitoramento.
UserParameterDir	Especifica o caminho de pesquisa padrão para comandos UserParameter.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores dos arquivos de configuração fornecidos.
- Os valores aceitam variáveis de ambiente.
- O Zabbix aceita arquivos de configuração somente na codificação UTF-8 sem BOM.
- Comentários iniciados com # são aceitos somente no início da linha.

Detalhes do parâmetro

Alias

Defina um alias para uma chave de item. Ele pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros `Alias` podem estar presentes. Não são permitidos vários parâmetros com a mesma chave `Alias`.
 Chaves `Alias` diferentes podem fazer referência à mesma chave de item.
 Os aliases podem ser usados em `HostMetadataItem`, mas não no parâmetro `HostnameItem`.

Exemplo 1: Obtendo 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 obter 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 os 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 low-level discovery que processam 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 daqueles item keys que correspondem a um padrão. O padrão da key é uma expressão curinga que suporta o caractere * para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de key podem ser definidas em combinação com `DenyKey`. Várias regras de correspondência de key podem ser definidas em combinação com `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Veja também: Restricting agent checks.

AllowKeyRegexp

Permite a execução dos item keys que correspondem a um padrão. O padrão da key é uma expressão regular.
 Várias regras de correspondência de keys podem ser definidas em conjunto com `DenyKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restringindo verificações do agent.

Exemplo:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Defina a frequência de envio dos valores do buffer para o Zabbix server/proxy (em segundos).
 Observe que, se o buffer estiver cheio, os dados serão enviados mais cedo.

Padrão: 5
 Intervalo: 1-3600

BufferSize

Define o número máximo de valores no buffer de memória.
 O agent enviará todos os dados coletados para o server ou proxy do Zabbix se o buffer estiver cheio. Este parâmetro deve ser usado somente se o buffer persistente estiver desativado (`EnablePersistentBuffer=0`).

Padrão: 1000
 Intervalo: 2-65535

ControlSocket

Especifique o socket de controle para enviar comandos em tempo de execução com a opção `-R`.

Padrão: `/tmp/agent.sock`

DebugLevel

Defina o nível de depuração: - *0* - informações básicas sobre a inicialização e a finalização 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

Impede a execução dos item keys que correspondem a um padrão.
 O padrão da chave é uma expressão curinga que aceita o caractere `*` para corresponder a qualquer quantidade de caracteres.
Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKey`. Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restringindo verificações do agent.

Padrão: `DenyKey=system.run[*]`

DenyKeyRegexp

Impede a execução dos item keys que correspondem a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade.
 Todos os itens `system.run[]` são negados por padrão (`DenyKey=system.run[*]`), a menos que outra regra `system.run[*]` seja especificada.

Consulte também: Restringindo verificações do agent.

Exemplo:

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

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

EnableProfiler

Habilita a criação periódica de perfis do Go na inicialização do agent. A criação de perfis também pode ser habilitada ou desabilitada em tempo de execução usando a opção `-R`. Enquanto `periodic_prof_execute` estiver coletando perfis, todos os comandos de tempo de execução do profiler serão rejeitados por estarem ocupados.

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 - desativado, 1 - ativado

HeartbeatFrequency

Define a frequência das mensagens de heartbeat em segundos.
 Usado para monitorar a disponibilidade de active checks.
0 - mensagens de heartbeat desativadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Especifique a interface do host (endereço IP ou nome DNS) usada durante a autoregistration do host.
 Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente um endereço IP ou DNS.
 Para mais detalhes, consulte Using DNS as default interface.

Se não for definido, o valor será obtido de `HostInterfaceItem`.

O agent emitirá um erro e não será iniciado se o valor ultrapassar o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Especifique um item para determinar a interface do host (endereço IP ou nome DNS) durante o registro automático do host.
 Esse valor será usado somente se `HostInterface` não estiver definido.
 Para obter mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de registro automático, 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 compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Especifique os metadados usados para identificar ou distinguir o host durante o registro automático do host.
 `HostMetadata` permite distinguir os hosts além do nome do host.

Se não for definido, o valor será obtido de `HostMetadataItem`.

O agent emitirá um erro e não será iniciado se o valor especificado exceder o limite de 2034 bytes ou não for uma string UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, os valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Não há suporte para metadados com várias linhas — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Especifique um item para obter metadados do host.
 Esta opção só é usada quando `HostMetadata` não está definido.

O valor de `HostMetadataItem` é recuperado a cada tentativa de autoregistro e usado somente no processo de autoregistro do host. `HostMetadataItem` permite distinguir hosts além do nome do host.

Parâmetros de usuário e aliases são compatíveis. O item `system.run[]` é compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

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 sejam UTF-8 válidos, mas não sejam IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Nome do host

Especifique nomes de host exclusivos e sensíveis a maiúsculas e minúsculas. Uma lista delimitada por vírgulas.
 Obrigatório para verificações ativas e deve corresponder aos nomes de host conforme configurados no server.
 O valor é obtido de `HostnameItem` se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para a linha inteira.

Padrão: Definido por HostnameItem

HostnameItem

Especifique um item para obter o nome do host.
 Esta opção só é usada quando `Hostname` não está definido.
 Parâmetros de usuário ou aliases não são compatíveis, mas o item `system.run[]` é compatível independentemente dos valores de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Padrão: `system.hostname`

Include

Especifique um diretório ou arquivos individuais para serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você altere isso durante o tempo de compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_agent2.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê, portanto considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivo:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir apenas os arquivos relevantes de um diretório, use o curinga (`*`) para correspondência de padrão.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Especifique o endereço IP de escuta do agent para conexões do server/proxy. Uma lista delimitada por vírgulas.
 O primeiro endereço IP é enviado ao Zabbix server, se estiver se conectando a ele, para recuperar a lista de verificações ativas.

Padrão: `0.0.0.0`

ListenPort

Especifique a porta de escuta do agent para conexões vindas do server/proxy.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

Especifique o arquivo de log do agent.

Padrão: `/tmp/zabbix_agent2.log`
 Obrigatório: Sim, se `LogType` estiver definido como `file`; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogType

Especifique 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

Especifique o arquivo para manter o banco de dados SQLite. Deve ser um nome de arquivo completo.
 Este parâmetro é usado somente se o buffer persistente estiver habilitado (`EnablePersistentBuffer=1`).

PersistentBufferPeriod

Defina o período de tempo para armazenar dados quando não houver conexão com o Zabbix server ou proxy. Dados mais antigos serão perdidos.
 Os dados de log serão preservados. Este parâmetro é usado somente se o buffer persistente estiver habilitado (`EnablePersistentBuffer=1`).

Padrão: 1h
 Intervalo: 1m-365d

PidFile

Especifique o arquivo PID.

Padrão: /tmp/zabbix_agent2.pid

Plugins.<PluginName>.System.Capacity

Define o limite de verificações por plugin <PluginName> que podem ser executadas ao mesmo tempo.

Padrão: 1000 Intervalo: 1-1000

Plugins.Log.MaxLinesPerSecond

Define o número máximo de novas linhas por segundo para envio ao server/proxy do Zabbix ao processar verificações ativas log ou logrt.
 O valor fornecido será substituído pelo parâmetro maxlines, informado na chave do item log e logrt.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em MaxLinesPerSecond para localizar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Ative o registro dos comandos de shell executados como avisos.
 Os comandos serão registrados somente se forem executados remotamente.
 Entradas de log não serão criadas se system.run[] for executado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desativado, 1 - ativado

PluginSocket

Especifique o caminho para o socket UNIX para comunicações de plugins carregáveis.

Padrão: /tmp/agent.plugin.sock

PluginTimeout

Especifique o tempo limite para conexões com plugins carregáveis, em segundos.

Padrão: Timeout
 Intervalo: 1-30

ProfilerDir

O diretório onde os dados de criação de perfil serão gravados.

Padrão: /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

O número máximo de arquivos a manter para cada tipo de perfil. O limite é aplicado aos arquivos concluídos após cada despejo de perfil. O perfil de CPU ativo é coletado em um arquivo temporário oculto e não conta para o limite. Os arquivos criados por periodic_prof_execute também estão sujeitos ao limite.

Padrão: 10
 Intervalo: 1-100

ProfilerInterval

O intervalo de gravação dos dados de criação de perfil, em segundos.

Padrão: 3600
 Intervalo: 1-86400

RefreshActiveChecks

Especifique a frequência de atualização da lista de active checks, em segundos.
 Observe que, após falhar ao atualizar os active checks, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Especifique os endereços IP, opcionalmente em notação CIDR, ou os nomes DNS dos servers/proxies do Zabbix. Uma lista separada por vírgulas.
 As conexões de entrada serão aceitas somente 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, desativa as verificações passivas e o agent não escutará em nenhuma porta TCP.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Especifique o endereço do server/proxy do Zabbix ou a configuração de cluster para obter checks ativos.
 O endereço do server/proxy é um endereço IP ou nome DNS, com porta opcional separada por dois-pontos.
A configuração de cluster é um ou mais endereços de membros de grupo de server ou proxy separados por ponto e vírgula. Vários servers/clusters do Zabbix e proxies do Zabbix podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um proxy do Zabbix para cada server/cluster do Zabbix. Se um proxy do Zabbix for especificado, então o server/cluster do Zabbix para esse proxy não deve ser especificado.
Vários endereços delimitados por vírgula podem ser fornecidos para usar vários servers independentes do Zabbix 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, os checks ativos serão desativados.

Exemplo para proxy do Zabbix:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de proxy do Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para vários 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

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy.
- realizar conexões ao executar alguns items (`web.page.get`, `net.tcp.port`, etc.).

StatusPort

Especifique a porta para ouvir solicitações de status HTTP (`http://localhost:<port>/status`).

Intervalo: 1024-32767

Tempo limite

Especifique por quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o proxy ou server do Zabbix.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do server do Zabbix;
- enviar solicitações ao server do Zabbix, incluindo solicitações de configuração de item e dados de item em active checks;
- recuperar dados de log por meio de logfile;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- atuar como fallback em cenários em que um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações do agent que têm configurações de timeout ajustáveis no frontend (em nível global, de proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

Especifique o nível de criptografia exigido para conexões de entrada.
 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll

Especifique a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e em PSK.

Exemplo:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Especifique a string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e em PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Especifique a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras 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

Especifique a string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.

Observe que este parâmetro não pode ser usado junto com `TLSAccept=cert,psk`; para conexão por certificado (`TLSConnect=cert`), use `TLSCipherAll13` em vez disso.

TLSCipherPSK

Especifique a string de cifra do OpenSSL (TLS 1.2). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Especifique a string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Especifique o nível de criptografia exigido para conexões do agent com o 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

Especifique o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada para comunicações criptografadas com o server ou proxy do Zabbix.

TLSPSKIdentity

Especifique a string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server ou proxy.

TLSServerCertIssuer

Especifique o emissor do certificado do server (proxy) permitido.

TLSServerCertSubject

Especifique o subject do certificado do server (proxy) permitido.

UnsafeUserParameters

Permite que todos os caracteres sejam passados nos 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

Especifique um parâmetro definido pelo usuário para monitorar. Pode haver vários parâmetros definidos pelo usuário.
Formato: `UserParameter=<key>,<shell command>`
Observe que o comando shell não deve retornar uma string vazia nem apenas EOL. Os comandos shell podem ter caminhos relativos, se o parâmetro `UserParameterDir` estiver especificado.

Exemplo:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Especifique o caminho de pesquisa padrão para comandos `UserParameter`. Se usado, o agent alterará seu diretório de trabalho para o diretório especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem usar um prefixo relativo `./` em vez de um caminho completo.
Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=/opt/myscripts
```

Parâmetros de configuração: Zabbix agent 2 (Windows)

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do Zabbix agent 2 no Windows.

Especifique os valores dos parâmetros no arquivo de configuração do agent 2 do Zabbix (`zabbix_agent2.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
AllowKeyRegexp	Permite a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
BufferSend	Define a frequência de envio de valores do buffer para o Zabbix server.
BufferSize	Define o número máximo de valores no buffer de memória.
ControlSocket	Especifica o socket de controle, usado para enviar comandos em tempo de execução com a opção `-R`.
DebugLevel	Define o nível de depuração.
DenyKey	Impede a execução das chaves de item que correspondem a um padrão.

Parâmetro	Descrição
DenyKeyRegexp	Impede a execução das chaves de item que correspondem a um padrão baseado em expressão regular.
EnablePersistentBuffer	Ativa o uso de armazenamento persistente local para itens ativos.
EnableProfiler	Ativa a criação periódica de perfis do Go na inicialização do agent.
ForceActiveChecksOnStart	Executa verificações ativas imediatamente após a reinicialização para a primeira configuração recebida.
HeartbeatFrequency	Define a frequência das mensagens de heartbeat em segundos.
HostInterface	Especifica a interface do host.
HostInterfaceItem	Especifica um item para obter a interface do host.
HostMetadata	Especifica os metadados do host.
HostMetadataItem	Especifica um item do Zabbix agent usado para obter os metadados do host.
Hostname	Especifica o nome do host.
HostnameItem	Especifica um item do Zabbix agent para obter o nome do host.
Include	Especifica um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
ListenIP	Especifica o endereço IP no qual o agent aguarda conexões do server/proxy.
ListenPort	Especifica a porta na qual o agent aguarda conexões.
LogFile	Especifica o arquivo de log do agent.
LogFileSize	Define o tamanho máximo do arquivo de log.
LogType	Especifica o tipo de saída do log.
PerfCounter	Especifica um novo parâmetro <parameter_name>, que corresponde ao valor médio do contador de desempenho do sistema <perf_counter_path> durante o período especificado <period> (em segundos).
PerfCounterEn	Especifica um novo parâmetro <parameter_name>, que corresponde ao valor médio do contador de desempenho do sistema <perf_counter_path> durante o período especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos dos contadores de desempenho devem estar em inglês.
PersistentBufferFile	Especifica o arquivo no qual o Zabbix agent 2 deve manter o banco de dados SQLite.
PersistentBufferPeriod	Define a duração do armazenamento de dados quando não há conexão com o server ou proxy.
Plugins.<PluginName>.System.Capacity	Define o limite de verificações por plugin que podem ser executadas simultaneamente.
Plugins.Log.MaxLinesPerSecond	Define o número máximo de novas linhas que o agent enviará por segundo ao Zabbix server ou proxy ao processar verificações ativas `log`, `logrt` e `eventlog`.
Plugins.SystemRun.LogRemoteCommands	Ativa o registro dos comandos shell executados como avisos.
PluginSocket	Especifica o caminho para o named pipe do Windows usado na comunicação com plugins carregáveis.
PluginTimeout	Define o tempo limite, em segundos, para conexões com plugins carregáveis.
ProfilerDir	O diretório no qual os dados de criação de perfis serão gravados.
ProfilerMaxFilesPerProfile	O número máximo de arquivos a serem mantidos para cada tipo de perfil.
ProfilerInterval	O intervalo de gravação dos dados de criação de perfis.
RefreshActiveChecks	Especifica a frequência de atualização da lista de verificações ativas.
Server	Especifica os endereços IP ou nomes DNS dos Zabbix servers/proxies.
ServerActive	Especifica o endereço do Zabbix server/proxy ou a configuração de cluster da qual obter as verificações ativas.
SourceIP	Especifica o endereço IP de origem.
StatusPort	Define a porta na qual o agent aguarda solicitações de status HTTP (`http://localhost:<port>/status`).
Timeout	Especifica quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o Zabbix server ou proxy.
TLSAccept	Especifica o nível de criptografia exigido para conexões de entrada.
TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados de CA de nível superior para verificação do certificado do peer, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado ou a cadeia de certificados do agent, usado nas comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll13	Especifica a string de cifras do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados e PSK.
TLSCipherCert13	Especifica a string de cifras do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificados.
TLSCipherPSK13	Especifica a string de cifras do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.
TLSConnect	Especifica o nível de criptografia exigido para conexões com o Zabbix server ou proxy.
TLSCRLFile	Especifica o caminho completo do arquivo que contém os certificados revogados. Esse parâmetro é usado nas comunicações criptografadas entre componentes do Zabbix.

Parâmetro	Descrição
TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent, usada nas comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	Especifica o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSPSKIdentity	Especifica a string de identidade da chave pré-compartilhada, usada nas comunicações criptografadas com o Zabbix server ou proxy.
TLSServerCertIssuer	Especifica o emissor permitido do certificado do server (proxy).
TLSServerCertSubject	Especifica o assunto permitido do certificado do server (proxy).
UnsafeUserParameters	Permite que todos os caracteres sejam passados nos argumentos dos parâmetros definidos pelo usuário.
UserParameter	Especifica um parâmetro definido pelo usuário para monitoramento.
UserParameterDir	Especifica o caminho de pesquisa padrão para comandos `UserParameter`.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores dos arquivos de configuração fornecidos;
- Os valores são compatíveis com variáveis de ambiente;
- O Zabbix oferece suporte apenas a arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários iniciados com # são compatíveis apenas no início da linha.

Detalhes do parâmetro

Alias

Defina um alias para uma chave de item. Ele 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`.
 Diferentes chaves `Alias` podem fazer referência à mesma chave de item.
 Os aliases podem ser usados em `HostMetadataItem`, mas não no parâmetro `HostnameItem`.

Exemplo 1: Obtendo o uso do arquivo de paginação em porcentagem no server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Agora, a chave abreviada `pg_usage` pode ser usada para obter 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 os parâmetros padrão, bem como usar `cpu.load[percpu,avg15]` para obter dados específicos sobre a carga da CPU.

Exemplo 3: Executando várias regras de low-level discovery que processam 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 daqueles item keys que correspondem a um padrão. O padrão da key é uma expressão curinga que suporta o caractere * para corresponder a qualquer quantidade de quaisquer caracteres.
 Várias regras de correspondência de key podem ser definidas em combinação com `DenyKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um a um de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Veja também: Restricting agent checks.

AllowKeyRegexp

Permite a execução das chaves de item que correspondem a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `DenyKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade. Consulte também: Restringindo verificações do agent.

Exemplo:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Define a frequência de envio dos valores do buffer para o Zabbix server/proxy (em segundos).
Observe que, se o buffer estiver cheio, os dados serão enviados mais cedo.

Padrão: 5
 Intervalo: 1-3600

BufferSize

Define 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 deve ser usado somente se o buffer persistente estiver desativado (`EnablePersistentBuffer=0`).

Padrão: 1000
 Intervalo: 2-65535

ControlSocket

Especifique o socket de controle, usado para enviar comandos em tempo de execução com a opção `-R`.

Padrão: `\\.\pipe\agent.sock`

DebugLevel

Defina 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 (produz muitas informações); - *5* - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

DenyKey

Impede a execução dos item keys que correspondem a um padrão. O padrão da chave é uma expressão curinga que aceita o caractere `*` para corresponder a qualquer quantidade de caracteres.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKey`, `AllowKeyRegexp` e `DenyKeyRegexp`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade.

Consulte também: Restringindo verificações do agent.

Padrão: `DenyKey=system.run[*]`

DenyKeyRegexp

Impede a execução dos item keys que correspondem a um padrão. O padrão da chave é uma expressão regular.
 Várias regras de correspondência de chaves podem ser definidas em conjunto com `AllowKeyRegexp`, `AllowKey` e `DenyKey`. Os parâmetros são processados um por um, de acordo com a ordem em que aparecem. A ordem determina qual regra correspondente terá prioridade.
 Todos os itens `system.run[]` são negados por padrão (`DenyKey=system.run[*]`), a menos que outra regra `system.run[*]` seja especificada.

Consulte também: Restringindo verificações do agent.

EnablePersistentBuffer

Habilita o uso de armazenamento persistente local para itens ativos. Se o armazenamento persistente estiver desabilitado, o buffer de memória será usado.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

EnableProfiler

Habilita a criação periódica de perfis do Go na inicialização do agent. A criação de perfis também pode ser habilitada ou desabilitada em tempo de execução usando a opção `-R`. Enquanto `periodic_prof_execute` estiver coletando perfis, todos os comandos de tempo de execução do profiler serão rejeitados porque ele está ocupado.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

ForceActiveChecksOnStart

Execute 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 - desativado, 1 - ativado

HeartbeatFrequency

Define a frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de active checks.
0 - mensagens de heartbeat desativadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Especifique a interface do host (endereço IP ou nome DNS) usada durante a autoregistration do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente um endereço IP ou DNS. Para mais detalhes, consulte Using DNS as default interface.

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

Especifique um item para determinar a interface do host (endereço IP ou nome DNS) durante o registro automático do host. Esse valor será usado somente se `HostInterface` não estiver definido. Para obter mais detalhes, consulte Usando o DNS como interface padrão.

Durante uma solicitação de registro automático, 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 compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Especifique os metadados usados para identificar ou distinguir o host durante o registro automático do host (agent ativo). `HostMetadata` permite distinguir os hosts além do nome do host.

Se não for definido, o valor será obtido de `HostMetadataItem`.

O agent emitirá um erro e não será iniciado se o valor especificado exceder o limite de 2034 bytes ou não for uma string UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, os valores que forem UTF-8 válidos, mas não forem IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Especifique um item do Zabbix agent para obter metadados do host. Esta opção só é usada quando `HostMetadata` não está definido.

O valor de `HostMetadataItem` é recuperado a cada tentativa de autoregistro e usado somente no processo de autoregistro do host (agent ativo). `HostMetadataItem` permite distinguir entre hosts além do nome do host.

Parâmetros de usuário e aliases são compatíveis. O item `system.run[]` é compatível independentemente das configurações `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

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, os valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e reportados como inválidos.

Metadados com várias linhas não são compatíveis — a saída será truncada na primeira quebra de linha.

Nome do host

Especifique nomes de host exclusivos e sensíveis a maiúsculas e minúsculas. Uma lista delimitada por vírgulas.
 Obrigatório para verificações ativas e deve corresponder aos nomes de host conforme configurados no server. O valor é obtido de HostnameItem se não estiver definido.
 Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para a linha inteira.

Padrão: Definido por HostnameItem

HostnameItem

Especifique um item para obter o nome do host. Esta opção é usada somente quando `Hostname` não está definido. Parâmetros de usuário ou aliases não são compatíveis, mas o item `system.run[]` é compatível independentemente dos valores de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Padrão: `system.hostname`

Include

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração (localizado em `C:\Program Files\Zabbix Agent 2` por padrão, se o Zabbix agent for instalado usando os pacotes do instalador MSI do Windows; localizado na pasta especificada durante a instalação, se o Zabbix agent for instalado como um arquivo zip). Durante a instalação do Zabbix, este parâmetro é definido como `/usr/local/etc`, a menos que você altere esse valor durante a compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_agent2.conf` (por exemplo, `Include=.\zabbix_agent2.d\plugins.d\*`
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê. Portanto, considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter a sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivos:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir somente os arquivos relevantes de um diretório, use o caractere curinga (`*`) para correspondência de padrões.

Exemplo:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Especifique o endereço IP de escuta do agent para conexões provenientes do server/proxy. Uma lista delimitada por vírgulas.
 O primeiro endereço IP é enviado ao server Zabbix, se estiver se conectando a ele, para recuperar a lista de verificações ativas.

Padrão: `0.0.0.0`

ListenPort

Especifique a porta de escuta do agent para conexões do server/proxy.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

Especifique o arquivo de log do agent.

Padrão: `c:\zabbix_agent2.log`
 Obrigatório: Sim, se `LogType` estiver definido como `file`; caso contrário, não

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
0 - desativa 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 reiniciado.

Padrão: 1
 Intervalo: 0-1024

LogType

Especifique 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`

PerfCounter

Defina um novo parâmetro <parameter_name> que seja o valor médio do 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 quiser receber o número médio de interrupções do processador por segundo no último minuto, você 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 do contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos do perfcounter devem estar em inglês. Suportado apenas no **Windows Server 2008/Vista** e versões posteriores.
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, você 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 consultando a seguinte chave do registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

PersistentBufferFile

Especifique o arquivo onde o Zabbix agent 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

Define a duração do armazenamento de dados quando não houver conexão com o server ou proxy. Os dados mais antigos serão perdidos. Os dados de log serão preservados.
 Este parâmetro é usado somente se o buffer persistente estiver habilitado (`EnablePersistentBuffer=1`).

Padrão: `1h`
 Intervalo: 1m-365d

Plugins.<PluginName>.System.Capacity

Define o limite de verificações por plugin <`PluginName`> que podem ser executadas ao mesmo tempo.

Padrão: 1000 Intervalo: 1-1000

Plugins.Log.MaxLinesPerSecond

Define o número máximo de novas linhas que o agent enviará por segundo para o server ou proxy do Zabbix ao processar verificações ativas `log`, `logrt` e `eventlog`. O valor fornecido será substituído pelo parâmetro `maxlines`, informado na chave do item `log`, `logrt` ou `eventlog`.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em `MaxLinesPerSecond` para localizar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Ative o registro dos comandos de shell executados como avisos. Os comandos serão registrados somente se forem executados remotamente. Entradas de log não serão criadas se `system.run[]` for iniciado localmente pelos parâmetros `HostMetadataItem`, `HostInterfaceItem` ou `HostnameItem`.

Padrão: 0
 Valores: 0 - desativado, 1 - ativado

PluginSocket

Especifique o caminho para o pipe nomeado do Windows para comunicações de plugin carregável.

Padrão: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Define o tempo limite para conexões com plugins carregáveis, em segundos.

Padrão: `Timeout`
 Intervalo: 1-30

ProfilerDir

O diretório onde os dados de criação de perfil serão gravados.

Padrão: `c:\zabbix_pprof`

ProfilerMaxFilesPerProfile

O número máximo de arquivos a manter para cada tipo de perfil. O limite é aplicado aos arquivos concluídos após cada despejo de perfil. O perfil de CPU ativo é coletado em um arquivo temporário oculto e não conta para o limite. Os arquivos criados por `periodic_prof_execute` também estão sujeitos ao limite.

Padrão: 10
 Intervalo: 1-100

ProfilerInterval

O intervalo de gravação dos dados de criação de perfil, em segundos.

Padrão: 3600
 Intervalo: 1-86400

RefreshActiveChecks

Especifique a frequência de atualização da lista de verificações ativas, 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

Especifique os endereços IP, opcionalmente em notação CIDR, ou os nomes DNS dos servers/proxies do Zabbix. Uma lista separada por vírgulas.
 Se o parâmetro não for especificado, as verificações passivas serão desativadas e o agent não escutará em nenhuma porta TCP.
 As conexões de entrada serão aceitas somente 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.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Especifique o endereço do server/proxy do Zabbix ou a configuração de cluster para obter verificações ativas.
 Se o parâmetro não for especificado, as verificações ativas serão desativadas.

O endereço do server/proxy é um endereço IP ou nome DNS, com uma porta opcional separada por dois-pontos. Se a porta não for especificada, a porta padrão será usada.
 Endereços IPv6 devem ser colocados entre colchetes se a porta desse host for especificada. Se a porta não for especificada, os colchetes para endereços IPv6 são opcionais.

A configuração de cluster é um ou mais endereços de membros do grupo de server ou proxy separados por ponto e vírgula.

Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula.

A menos que estejam sendo usados grupos de proxy, não deve ser especificado mais de um Zabbix proxy para cada Zabbix server/cluster. Se um Zabbix proxy for especificado, então o Zabbix server/cluster correspondente a 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.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de proxy do Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para vários 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

Especifique o endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy;
- realizar conexões ao executar alguns items (`web.page.get`, `net.tcp.port`, etc.).

StatusPort

Defina a porta na qual o agent escuta as solicitações de status HTTP (`http://localhost:<port>/status`).

Intervalo: 1024-32767

Timeout

Especifique por quanto tempo aguardar (em segundos) para estabelecer a conexão e trocar dados com o proxy ou server do Zabbix.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do server do Zabbix;
- enviar solicitações ao server do Zabbix, incluindo solicitações de configuração de item e dados de item em active checks;
- recuperar dados de log por meio de logfile ou monitoramento do log de eventos do Windows;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- atuar como fallback em cenários em que um server ou proxy mais antigo que a versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações do agent que têm configurações de timeout ajustáveis no frontend (em nível global, de proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

Especifique o nível de criptografia exigido para conexões de entrada.
 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

Especifique o caminho completo do arquivo que contém 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

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll13

Especifique a string de cifras do OpenSSL (TLS 1.3). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Especifique a string de cifra do OpenSSL (TLS 1.3). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em certificado.

Observe que este parâmetro não pode ser usado junto com `TLSAccept=cert,psk`; para conexão por certificado (`TLSConnect=cert`), use `TLSCipherAll13` em vez disso.

TLSCipherPSK13

Especifique a string de cifra do OpenSSL (TLS 1.3). Substitua os critérios padrão de seleção do conjunto de cifras para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Especifique o nível de criptografia exigido para conexões do agent com o 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 estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

Especifique o caminho completo do arquivo que contém os certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

Especifique o caminho completo do arquivo que contém a chave pré-compartilhada do agent, usada para comunicações criptografadas com o server ou proxy do Zabbix.

TLSPSKIdentity

Especifique a string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server ou proxy.

TLSServerCertIssuer

Especifique o emissor do certificado do server (proxy) permitido.

TLSServerCertSubject

Especifique o subject do certificado do server (proxy) permitido.

UnsafeUserParameters

Permite que todos os caracteres sejam passados nos 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

Especifique um parâmetro definido pelo usuário para monitorar. Pode haver vários parâmetros definidos pelo usuário.
Formato: `UserParameter=<key>,<shell command>`
Observe que o comando de shell não deve retornar uma string vazia nem apenas EOL. Os comandos de shell podem ter caminhos relativos, se o parâmetro `UserParameterDir` estiver especificado.

Exemplo:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Especifique o caminho de pesquisa padrão para comandos `UserParameter`. Se usado, o agent alterará seu diretório de trabalho para o diretório especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem usar um prefixo relativo `.\` em vez de um caminho completo.
Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=C:\opt\myscripts
```

Parâmetros de configuração: Plugins

Veja os parâmetros de configuração dos plugins do Zabbix agent 2:

- Plugin Ceph
- Plugin Docker
- Plugin Ember+
- Plugin Memcached
- Plugin Modbus
- Plugin MongoDB
- Plugin MQTT
- Plugin MSSQL
- Plugin MySQL
- Plugin NVIDIA GPU
- Plugin Oracle
- Plugin PostgreSQL
- Plugin Redis
- Plugin S.M.A.R.T.

Princípios de configuração de plugins

Esta página descreve os princípios de configuração de plugins e as práticas recomendadas.

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 de um plugin. Se um plugin usar um arquivo de configuração separado, o caminho para esse arquivo deverá 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 destinos de monitoramento por meio de sessões nomeadas.

Todos os nomes de parâmetros devem atender aos seguintes requisitos:

- Recomenda-se capitalizar os nomes dos 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 verificações ativas que não tenham *Scheduling* intervalo de atualização imediatamente após a reinicialização do agent somente para o plugin Uptime, defina `Plugins.Uptime.System.ForceActiveChecksOnStart=1` no arquivo de configuração. Da mesma forma, para definir um limite personalizado para verificações simultâneas do 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 não for fornecido um valor para esse parâmetro na chave de um item ou nos parâmetros da sessão nomeada, o plugin usará o valor padrão. Se um parâmetro padrão também não estiver definido, serão usados valores padrão codificados.

> **Note:**
> Se a chave de um 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 do 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 uma URI, nome de usuário e/ou senha separadamente.

Nas chaves de item, o primeiro parâmetro pode ser um connString ou uma URI. Se o primeiro parâmetro da chave não corresponder a nenhum nome de sessão, ele será tratado como uma URI. Observe que não há suporte para passar credenciais de URI incorporadas na chave de item; use parâmetros de sessão nomeada em vez disso.

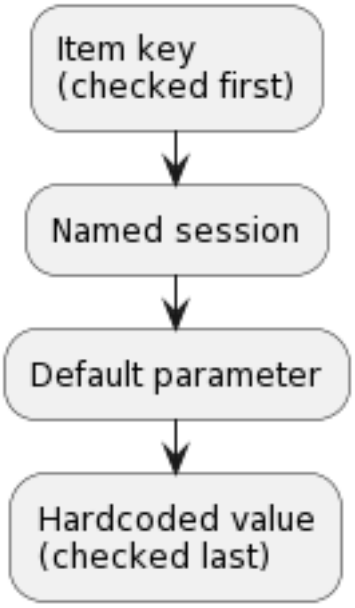
A lista de parâmetros de sessão nomeada disponíveis depende do plugin.

É possível substituir parâmetros de sessão especificando novos valores nos parâmetros da chave de item (consulte exemplo).

Se um parâmetro não estiver 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 valores de parâmetros relacionados à conexão na seguinte ordem:



1. O primeiro parâmetro da chave do item é comparado com os nomes das sessões. Se nenhuma correspondência for encontrada, ele é tratado como um valor real; nesse caso, a etapa 3 será ignorada. Se uma correspondência for encontrada, o valor do parâmetro (normalmente, uma 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, password) estiver vazio, o plugin procurará o parâmetro correspondente da sessão nomeada.
4. Se o parâmetro da sessão também não estiver especificado, será usado o valor definido no parâmetro padrão correspondente.
5. Se nada disso funcionar, 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://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

Como resultado dessa configuração, cada nome de sessão pode ser usado como um connString em uma item key, 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 se conectará 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 ela 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 dessa configuração, o agent se conectará ao PostgreSQL usando os parâmetros:

- URI padrão: `192.0.2.234:5432`
- Nome de usuário padrão: `zabbix`
- Senha padrão: `password`
- Nome do banco de dados da chave do item: `postgres`

Conexões

Alguns plugins suportam a coleta de métricas de várias instâncias simultaneamente. Tanto instâncias locais quanto remotas podem ser monitoradas. Conexões TCP e Unix-socket são suportadas.

É recomendável configurar os plugins para manter as conexões com as instâncias em estado aberto. Os benefícios são a redução da congestão de rede, da latência e do uso de CPU e memória devido ao menor número de conexões. A biblioteca cliente cuida disso.

> **Note:**
> O período de tempo durante o qual as conexões não utilizadas devem permanecer abertas pode ser determinado pelo parâmetro `Plugins.<PluginName>.KeepAlive`. Exemplo: `Plugins.Memcached.KeepAlive`

Plugin Ceph

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Ceph.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `ceph.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Ceph.Default.ApiKey	Especifica a chave de API padrão para conexão com o Ceph.
Plugins.Ceph.Default.Mode	Especifica o modo padrão para conexão com o Ceph.
Plugins.Ceph.Default.URI	Especifica a URI padrão para conexão com o Ceph.
Plugins.Ceph.Default.User	Especifica o nome de usuário padrão para conexão com o Ceph.
Plugins.Ceph.InsecureSkipVerify	Especifica se um cliente HTTP deve verificar a cadeia de certificados do server e o nome do host.
Plugins.Ceph.KeepAlive	Define o tempo máximo de espera antes que conexões de plugin não utilizadas sejam encerradas.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Especifica a chave de API da sessão nomeada para conexão com o Ceph.
Plugins.Ceph.Sessions.<SessionName>.Mode	Especifica o modo da sessão nomeada para conexão com o Ceph.
Plugins.Ceph.Sessions.<SessionName>.URI	Especifica a URI da sessão nomeada para conexão com o Ceph.
Plugins.Ceph.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada para conexão com o Ceph.
Plugins.Ceph.Timeout	Define o tempo limite de execução da requisição.

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 somente em codificação UTF-8 sem BOM;
- Comentários que começam com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Ceph.Default.ApiKey

Especifique a chave de API padrão para conexão com o Ceph.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Ceph.Default.Mode

Especifique o modo padrão para conexão com o Ceph.
 O modo nativo é suportado apenas no Linux e, a partir do Ceph 16. O modo restful não funcionará com a versão 20 do Ceph ou mais recente.
 Observe que o conjunto de credenciais do usuário difere para cada modo e eles não são compatíveis entre si.

Padrão: `restful` (obsoleto)
 Intervalo: native/restful

Plugins.Ceph.Default.URI

Especifique a URI padrão para conexão com o Ceph.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Não deve incluir credenciais incorporadas (elas serão ignoradas).
Deve corresponder ao formato de URI.
Apenas o esquema `https` é suportado; o esquema pode ser omitido.
A porta pode ser omitida (padrão=8003).

Padrão: `https://localhost:8003`

Exemplos:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Especifique o nome de usuário padrão para conectar ao Ceph.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Ceph.InsecureSkipVerify

Especifique se um cliente HTTP deve verificar a cadeia de certificados e o nome do host do server.
 Se *true*, o TLS aceita qualquer certificado apresentado pelo server e qualquer nome de host nesse certificado. Nesse modo, o TLS fica suscetível a ataques man-in-the-middle (use apenas para testes).

Padrão: `false`
 Intervalo: false/true

Plugins.Ceph.KeepAlive

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Especifique a chave de API da sessão nomeada para conexão com o Ceph.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Ceph.Sessions.<SessionName>.Mode

Especifique o modo da sessão nomeada para conexão com o Ceph.
 <SessionName> - define o nome de uma sessão para uso em item keys.
 O modo nativo é suportado apenas no Linux e a partir do Ceph 16. O modo restful não funcionará com a versão 20 do Ceph ou mais recente.
 Observe que o conjunto de credenciais do usuário difere para cada modo e eles não são compatíveis entre si.

Padrão: `restful` (obsoleto)
 Intervalo: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Especifique a URI da sessão nomeada para conexão com o Ceph.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
Não deve incluir credenciais embutidas (elas serão ignoradas).
Deve corresponder ao formato de URI.
 Apenas o esquema `https` é suportado; o esquema pode ser omitido.
 A porta pode ser omitida (padrão=8003).

Padrão: `https://localhost:8003`

Exemplos:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para conexão ao Ceph.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Ceph.Timeout

Defina o tempo limite de execução da solicitação (duração de espera para que uma solicitação seja concluída antes de encerrá-la) em segundos.

Padrão: `global timeout<br>` Intervalo: 1-30

Veja também

- Instruções para configurar plugins
- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

Plugin do Docker

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Docker.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `docker.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Docker.Endpoint	Especifique a localização do socket UNIX do daemon do Docker..
Plugins.Docker.Timeout	Especifique o tempo limite de execução da requisição.

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 que começam com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Docker.Endpoint

Especifique o local do socket UNIX do daemon do Docker.
 Deve conter um esquema (somente `unix://` é suportado).

Padrão: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Especifique o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: `global timeout<br>` Intervalo: 1–30

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin do Memcached

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Memcached.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `memcached.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Memcached.Default.Password	Especifica a senha padrão para conexão com o Memcached.
Plugins.Memcached.Default.Uri	Especifica a URI padrão para conexão com o Memcached.
Plugins.Memcached.Default.User	Especifica o nome de usuário padrão para conexão com o Memcached.
Plugins.Memcached.KeepAlive	Define o tempo máximo de espera antes que conexões não utilizadas do plugin sejam encerradas.
Plugins.Memcached.Sessions.<SessionName>.Password	Especifica a senha da sessão nomeada para conexão com o Memcached.
Plugins.Memcached.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada para conexão com o Memcached.
Plugins.Memcached.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada para conexão com o Memcached.
Plugins.Memcached.Timeout	Define o tempo limite de execução da solicitação.

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 somente em codificação UTF-8 sem BOM;
- Comentários que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Memcached.Default.Password

Especifique a senha padrão para conexão com o Memcached.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Memcached.Default.Uri

Especifique a URI padrão para conexão com o Memcached.
 Usada se nenhum valor for especificado em uma item key ou sessão nomeada.
 As credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato de URI. Esquemas suportados: `tcp`, `unix`.
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=11211).

Padrão: `tcp://localhost:11211`

Exemplos:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Especifique o nome de usuário padrão para se conectar ao Memcached.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Memcached.KeepAlive

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60–900

Plugins.Memcached.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para conexão com o Memcached.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.Memcached.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o Memcached.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 As credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato de URI. Esquemas suportados: `tcp`, `unix`.
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=11211).

Exemplos:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para conexão com o Memcached.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Memcached.Timeout

Define o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: `global timeout`
 Intervalo: 1–30

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin do MongoDB

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin MongoDB.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mongodb.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

O plugin MongoDB é um plugin carregável e está disponível, com descrição completa, no repositório do plugin MongoDB.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Ao se conectar ao MongoDB, o plugin verifica sua configuração em uma ordem específica, que determina qual valor será usado. A ordem, da maior para a menor prioridade, é a seguinte:

1. Parâmetros especificados na URI de conexão (por exemplo, os parâmetros de consulta definidos após `?` em `Plugins.MongoDB.Sessio` têm a maior prioridade).
2. Parâmetros na chave do item do Zabbix agent 2 (por exemplo, mongodb.collection.stats).
3. Parâmetros definidos em uma sessão nomeada (por exemplo, `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions.<SessionName>.Password`).
4. Valores padrão (por exemplo, `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Por exemplo:

- Se você especificar o banco de dados para autenticar uma conexão com o MongoDB server no parâmetro `*.<SessionName>.Uri` (por exemplo, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSo` o plugin considerará que o parâmetro `*.<SessionName>.AuthSource` está definido como `admin`, mesmo que o parâmetro `*.<SessionName>.AuthSource` esteja configurado com outro valor.
- Se você especificar a URI no parâmetro `*.<SessionName>.Uri` (por exemplo, `Plugins.MongoDB.Sessions.<session_name>.Ur` mas fornecer o nome de usuário e a senha na chave do item do Zabbix agent 2, o plugin usará o nome de usuário e a senha da chave do item, mesmo que `Plugins.MongoDB.Sessions.<SessionName>.User` e `Plugins.MongoDB.Sessions.<SessionNam` estejam configurados.

Attention:
Se o esquema `mongodb+srv://` for usado, a URI poderá recuperar parâmetros de um registro DNS TXT. Os parâmetros obtidos desse registro TXT substituem todas as outras configurações, incluindo parâmetros da URI, parâmetros da chave do item do Zabbix Agent 2 e parâmetros da sessão.
 Apenas um registro TXT pode existir para o hostname usado na string de conexão.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique em seu nome)
- Detalhes dos parâmetros

Parameter	Description
Plugins.MongoDB.Default.AuthMechanism	Especifica o mecanismo de autenticação padrão para conectar ao MongoDB server.
Plugins.MongoDB.Default.AuthSource	Especifica o banco de dados padrão para autenticar a conexão com o MongoDB server.
Plugins.MongoDB.Default.Dial	Desativa a descoberta automática dos nós do conjunto de réplicas do MongoDB.
Plugins.MongoDB.Default.Password	Especifica a senha padrão.
Plugins.MongoDB.Default.ReplicaSet	Especifica o nome padrão do conjunto de réplicas, usado para a descoberta automática dos nós do conjunto de réplicas do MongoDB.
Plugins.MongoDB.Default.ReadPreference	Especifica a ordem padrão para conectar aos nós do conjunto de réplicas do MongoDB ou para fallback.

Parameter	Description
Plugins.MongoDB.Default.TLSConnect	Especifica o nível de criptografia exigido para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
Plugins.MongoDB.Default.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados das CA(s) de nível superior para verificação do certificado do par.
Plugins.MongoDB.Default.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados.
Plugins.MongoDB.Default.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do banco de dados.
Plugins.MongoDB.Default.Uri	Especifica a URI padrão.
Plugins.MongoDB.Default.User	Especifica o nome de usuário padrão.
Plugins.MongoDB.KeepAlive	Define o tempo máximo de espera antes que conexões não utilizadas do plugin sejam encerradas.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	Especifica o mecanismo de autenticação da sessão nomeada para conectar ao MongoDB server.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Especifica o banco de dados da sessão nomeada para autenticar a conexão com o MongoDB server.
Plugins.MongoDB.Sessions.<SessionName>.Direct	Desativa a descoberta automática dos nós do conjunto de réplicas do MongoDB para a sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.Password	Especifica a senha da sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Especifica o nome do conjunto de réplicas da sessão nomeada, usado para a descoberta automática dos nós do conjunto de réplicas do MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Especifica o modo de leitura da sessão nomeada para conectar aos nós do conjunto de réplicas do MongoDB ou para fallback.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Especifica o nível de criptografia exigido para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados para a sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados das CA(s) de nível superior para verificação do certificado do par para a sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados para a sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do banco de dados para a sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada.
Plugins.MongoDB.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada.
Plugins.MongoDB.System.Path	Especifica o caminho para o executável do plugin.
Plugins.MongoDB.Timeout	Define o tempo limite de execução da requisição.

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 que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.MongoDB.Default.AuthMechanism

Especifique o mecanismo de autenticação padrão para conexão com o server MongoDB.
 <SessionName> - define o nome de uma sessão para uso em chaves de item.

Intervalo: `SCRAM-SHA-1`, `SCRAM-SHA-256` ou `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Especifique o banco de dados padrão para autenticar a conexão com o server MongoDB.
 <SessionName> - define o nome de uma sessão para uso em chaves de item.

Padrão: `admin`

Se o mecanismo de autenticação `MONGODB-X509` for usado, este parâmetro é definido automaticamente como `$external`; modificá-lo para qualquer outro valor causará um erro ao iniciar.

Plugins.MongoDB.Default.Direct

Desative a descoberta automática dos nós do replica set do MongoDB.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Padrão: `false`
 Intervalo: `true` - conecta-se apenas ao server especificado, ignorando outros membros do replica set, `false` - descobre automaticamente todos os nós no replica set.

Se o esquema `tcp://` (obsoleto) for usado, o parâmetro é definido como `true` por padrão.

Plugins.MongoDB.Default.Password

Especifique a senha padrão.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.MongoDB.Default.ReplicaSet

Especifique o nome padrão do replica set, usado para a descoberta automática de nós do replica set do MongoDB.
 <SessionName> - define o nome de uma sessão para uso em item keys.
 Se este parâmetro for usado com o esquema `mongodb+srv://`, ele se comportará como se vários URLs tivessem sido fornecidos; um servidor DNS pode ter um registro TXT que pode substituir este parâmetro (ou qualquer outro).
 Indisponível se o parâmetro `Plugins.MongoDB.Sessions.<SessionName>.Dire` estiver definido como `true`; o agent falhará ao iniciar.

Plugins.MongoDB.Default.ReadPreference

Especifique a ordem padrão para conexão com os nós do replica set do MongoDB ou o fallback.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.
 Indisponível se o parâmetro `Plugins.MongoDB.Sessions.<SessionName>.Direc` estiver definido como `true`; o agent não será iniciado.

Intervalo: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` ou `NearestMode`

Os valores `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` e `nearest` são preferenciais. Os valores terminados em `Mode` são compatíveis como aliases de compatibilidade legados.

Plugins.MongoDB.Default.TLSConnect

Especifique o nível de criptografia exigido para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 Compatível desde a versão 1.2.1 do plugin.

Intervalo: `required` - exige conexão TLS, `verify_ca` - verifica certificados, `verify_full` - verifica certificados e endereço IP

Plugins.MongoDB.Default.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usados para comunicações criptografadas entre o agent 2 do Zabbix e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso em item keys.

Obrigatório: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.MongoDB.Default.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile` estiver especificado

Plugins.MongoDB.Default.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do banco de dados, usada para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile` estiver especificado

Plugins.MongoDB.Default.Uri

Especifique a URI padrão.
 <SessionName> - define o nome de uma sessão para uso em item keys.
 Deve corresponder ao formato da URI.
 Esquemas suportados: `mongodb://` e `mongodb+srv://`; o esquema `tcp://` está obsoleto e é mantido por compatibilidade com configurações existentes.
 Uma porta pode ser omitida (padrão=27017).
Exemplos: “

Padrão: `tcp://127.0.0.1:27017`

Exemplos:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Especifique o nome de usuário padrão.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.MongoDB.KeepAlive

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60-900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Especifique o mecanismo de autenticação da sessão nomeada para conectar ao server MongoDB.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Intervalo: `SCRAM-SHA-1`, `SCRAM-SHA-256` ou `MONGODB-X509`

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Especifique o banco de dados da sessão nomeada para autenticar a conexão com o server MongoDB.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Padrão: `admin`

Se o mecanismo de autenticação `MONGODB-X509` for usado, este parâmetro é definido automaticamente como `$external`; modificá-lo para qualquer outro valor causará um erro ao iniciar.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Desative a descoberta automática de nós do replica set do MongoDB para a sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso em item keys.

Padrão: `false`
 Intervalo: `true` - conecta-se apenas ao server especificado, ignorando outros membros do replica set, `false` - descobre automaticamente todos os nós no replica set.

Se o esquema `tcp://` (obsoleto) for usado, o parâmetro é definido como `true` por padrão.

Plugins.MongoDB.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Especifique o nome do replica set da sessão nomeada, usado para a descoberta automática dos nós do replica set do MongoDB.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.
 Se este parâmetro for usado com o esquema `mongodb+srv://`, ele se comportará como se vários URLs tivessem sido fornecidos; um servidor DNS pode ter um registro TXT que pode substituir este parâmetro (ou qualquer outro).
 Indisponível se o parâmetro `Plugins.MongoDB.Sessions.<SessionName>.Direct` estiver definido como `true`; o agent falhará ao iniciar.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Especifique a ordem da sessão nomeada para conexão com os nós do replica set do MongoDB ou o fallback.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.
 Indisponível se o parâmetro `Plugins.MongoDB.Sessions.<SessionName` estiver definido como `true`; o agent não será iniciado.

Intervalo: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` ou `NearestMode`

Os valores `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` e `nearest` são preferenciais. Os valores terminados em `Mode` são compatíveis como aliases de compatibilidade legados.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Especifique o nível de criptografia exigido para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados para a sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 Compatível desde a versão 1.2.1 do plugin.

Intervalo: `required` - exige conexão TLS, `verify_ca` - verifica certificados, `verify_full` - verifica certificados e endereço IP

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados CA de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados para a sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso em chaves de item.

Obrigatório: `yes`, se `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados para a sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso em chaves de item.

Obrigatório: yes, se Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile estiver especificado

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do banco de dados, usada para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados para a sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso em chaves de item.

Obrigatório: yes, se Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile for especificado

Plugins.MongoDB.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 Deve corresponder ao formato da URI.
 Esquemas suportados: mongodb:// e mongodb+srv://; o esquema tcp:// está obsoleto e é mantido por compatibilidade com configurações existentes.
 A porta pode ser omitida (padrão=27017).
Exemplos: “

Padrão: tcp://127.0.0.1:27017

Exemplos:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MongoDB.System.Path

Especifique o caminho para o executável do plugin do MongoDB.

Exemplo de uso: Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb

Plugins.MongoDB.Timeout

Define o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: global timeout
 Intervalo: 1-30

Opções

Parameter	Description
-V --version	Exibe a versão do plugin e as informações de licença.
-h --help	Exibe as informações de ajuda (atalho).
-t, --test <item key>	Inicia o plugin para teste (a configuração do plugin é ignorada).

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin Ember+

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Ember+.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, emberplus.conf), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

O plugin Ember+ é um plugin carregável e está disponível, com descrição completa, no repositório do plugin Ember+.

Atualmente, este plugin só está disponível para compilação a partir do código-fonte (tanto para Unix quanto para Windows).

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)

- Detalhes dos parâmetros

Parameter	Description
Plugins.EmberPlus.Default.Uri	Especifica a URI padrão para conexão com Ember+.
Plugins.EmberPlus.KeepAlive	Define o tempo máximo de espera antes que conexões de plugin não utilizadas sejam encerradas.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada para conexão com Ember+.
Plugins.EmberPlus.System.Path	Especifica o caminho para o executável do plugin Ember+.
Plugins.EmberPlus.Timeout	Define o tempo limite de execução da solicitação.

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 que começam com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.EmberPlus.Default.Uri

Especifique a URI padrão para conexão com Ember+.
 O único esquema suportado é `tcp://`. Um esquema pode ser omitido. As credenciais incorporadas serão ignoradas.

Padrão: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60-900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com Ember+.
 O único esquema suportado é `tcp://`. Um esquema pode ser omitido. Credenciais incorporadas serão ignoradas.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Padrão: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Especifique o caminho para o executável do plugin Ember+.

Exemplo: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Defina a duração, em segundos, para aguardar uma resposta do server ao se conectar pela primeira vez e nas operações subsequentes na sessão.

Padrão: `global timeout`
 Intervalo: 1-30

Opções

Parameter	Description
-V --version	Exibe a versão do plugin e as informações de licença.
-h --help	Exibe informações de ajuda (atalho).
-t, --test <item key>	Inicia o plugin para teste (a configuração do plugin é ignorada).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin Modbus

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Modbus.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `modbus.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Especifique o endpoint da sessão nomeada para conexão com o Modbus.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Especifique o ID do escravo da sessão nomeada para conexão com o Modbus.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Defina o tempo limite da sessão nomeada em segundos.

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 que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Especifique o endpoint da sessão nomeada para conexão com o Modbus.
 A string de conexão consiste em um esquema de protocolo, um endereço de host e uma porta ou nome da porta serial e atributos.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Especifique o ID do escravo da sessão nomeada para conexão com o Modbus.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Exemplo: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Observe que esse parâmetro de sessão nomeada é verificado somente se o valor fornecido no parâmetro de ID do escravo da item key estiver vazio.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Defina o tempo limite da sessão nomeada em segundos.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Exemplo: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Se você precisar definir o tempo limite de execução da solicitação (quanto tempo aguardar a conclusão de uma solicitação antes de encerrá-la), use o formulário de configuração do item.

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin MQTT

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin MQTT.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mqtt.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.MQTT.Default.Password	Especifica a senha padrão para conexão com MQTT.
Plugins.MQTT.Default.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da CA de nível superior para verificação do certificado do par.
Plugins.MQTT.Default.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados.
Plugins.MQTT.Default.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do MQTT.
Plugins.MQTT.Default.Topic	Especifica o tópico padrão para a assinatura MQTT.
Plugins.MQTT.Default.Url	Especifica a string de conexão padrão do broker MQTT.
Plugins.MQTT.Default.User	Especifica o nome de usuário padrão para conexão com MQTT.
Plugins.MQTT.Sessions.<Session name>.Password	Especifica a senha da sessão nomeada para conexão com MQTT.
Plugins.MQTT.Sessions.<Session name>.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da CA de nível superior para verificação do certificado do par para a sessão nomeada.
Plugins.MQTT.Sessions.<Session name>.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados para a sessão nomeada.
Plugins.MQTT.Sessions.<Session name>.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do MQTT para a sessão nomeada.
Plugins.MQTT.Sessions.<Session name>.Topic	Especifica o tópico da sessão nomeada para a assinatura MQTT.
Plugins.MQTT.Sessions.<Session name>.Url	Especifica a string de conexão do broker MQTT da sessão nomeada.
Plugins.MQTT.Sessions.<Session name>.User	Especifica o nome de usuário da sessão nomeada para conexão com MQTT.

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 somente em codificação UTF-8 sem BOM;
- Comentários que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.MQTT.Default.Password

Especifique a senha padrão para conexão com MQTT.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.MQTT.Default.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior para a verificação do certificado do par nas comunicações criptografadas entre o agent 2 do Zabbix e o broker MQTT.
 Usado se nenhum valor for especificado em uma sessão nomeada.

Plugins.MQTT.Default.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent ou a 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

Especifique o caminho completo do arquivo que contém 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

Especifique o tópico padrão para a assinatura MQTT.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
 O tópico pode conter curingas (”+”,”#”).

Exemplos:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Especifique a string de conexão padrão do broker MQTT.
 Usado se nenhum valor for especificado em uma item key ou sessão nomeada.
 Não deve incluir parâmetros de consulta.
 Deve corresponder ao formato de URL.
 Esquemas suportados: `tcp` (padrão), `ws`, `tls`
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=1883)

Padrão: `tcp://localhost:1883`

Exemplos:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Especifique o nome de usuário padrão para a conexão com MQTT.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.MQTT.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para conexão com o MQTT.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados CA de nível superior para verificação do certificado do par para a sessão nomeada, usado para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados para a sessão nomeada, usado para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada MQTT para a sessão nomeada, usada para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.Topic

Especifique o tópico da sessão nomeada para a assinatura MQTT.
 <SessionName> - define o nome de uma 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

Especifique a string de conexão do broker MQTT da sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso em chaves de item.
 Não deve incluir parâmetros de consulta.
 Deve corresponder ao formato de URL.
 Esquemas suportados: `tcp` (padrão), `ws`, `tls`
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=1883)

Padrão: `tcp://localhost:1883`

Exemplos:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para conexão ao MQTT.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Se você precisar definir o tempo limite de execução da solicitação (por quanto tempo aguardar a conclusão de uma solicitação antes de encerrá-la), use o formulário de configuração do item.

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

- Instruções para configurar plugins

plugin MSSQL

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin MSSQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mssql.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

O plugin MSSQL é um plugin carregável e está disponível, com descrição completa, no repositório do plugin MSSQL.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.MSSQL.CustomQueriesDir	Especifica o caminho para o diretório que contém arquivos `.sql` definidos pelo usuário com consultas personalizadas que o plugin pode executar.
Plugins.MSSQL.CustomQueriesEnabled	Habilita a execução da item key `mssql.custom.query`.
Plugins.MSSQL.Default.CAConnectPath	Especifica, por padrão, o caminho do arquivo para o certificado de chave pública da autoridade certificadora (CA) que emitiu o certificado do server MSSQL.
Plugins.MSSQL.Default.Database	Especifica, por padrão, o nome do banco de dados.
Plugins.MSSQL.Default.Encrypt	Especifica o nível de criptografia padrão.
Plugins.MSSQL.Default.HostnameInCertificate	Especifica, por padrão, o nome comum (CN) do certificado do server MSSQL.
Plugins.MSSQL.Default.Password	Especifica, por padrão, a senha.
Plugins.MSSQL.Default.TLSMinVersion	Especifica, por padrão, a versão mínima do TLS.
Plugins.MSSQL.Default.TrustServerCertificate	Confia no certificado do server sem validá-lo, por padrão.
Plugins.MSSQL.Default.Uri	Especifica, por padrão, a URI.
Plugins.MSSQL.Default.User	Especifica, por padrão, o nome de usuário.
Plugins.MSSQL.KeepAlive	Especifica o tempo máximo de espera (em segundos) antes que conexões do plugin não utilizadas sejam encerradas.
Plugins.MSSQL.Sessions.<SessionName>.CAConnectPath	Especifica o caminho do arquivo da sessão nomeada para o certificado de chave pública da autoridade certificadora (CA) que emitiu o certificado do server MSSQL.
Plugins.MSSQL.Sessions.<SessionName>.Database	Especifica o nome do banco de dados da sessão nomeada.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Especifica o nível de criptografia da sessão nomeada.
Plugins.MSSQL.Sessions.<SessionName>.HostnameInCertificate	Especifica, para a sessão nomeada, o nome comum (CN) do certificado do server MSSQL.
Plugins.MSSQL.Sessions.<SessionName>.Password	Especifica a senha da sessão nomeada.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Especifica a versão mínima do TLS da sessão nomeada.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Confia no certificado do server da sessão nomeada sem validá-lo.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada.
Plugins.MSSQL.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada.
Plugins.MSSQL.System.Path	Especifica o caminho para o executável do plugin MSSQL.
Plugins.MSSQL.Timeout	Define a duração, em segundos, para aguardar a resposta de um server na conexão inicial e nas operações subsequentes na sessão.

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 somente em codificação UTF-8 sem BOM.
- Comentários que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.MSSQL.CustomQueriesDir

Especifique o caminho para o diretório que contém 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 personalizados não serão refletidas até que o plugin seja reiniciado. O plugin é iniciado e encerrado junto com o Zabbix agent 2.

Padrões:
`/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`)

Plugins.MSSQL.CustomQueriesEnabled

Ativa a execução da chave de item `mssql.custom.query`.
 Se desativado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.

Padrão: `false`
 Intervalo: `true`/`false`

Plugins.MSSQL.Default.CACertPath

Especifique 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

Especifique o nome padrão do banco de dados.

Plugins.MSSQL.Default.Encrypt

Especifique o nível de criptografia padrão.

Intervalo:
true - o envio de dados entre o plugin e o server é criptografado;
false - o envio de dados entre o plugin e o server não é criptografado além do pacote de login;
strict - o envio de dados entre o plugin e o server é criptografado de ponta a ponta usando TDS8;
disable - o envio de dados entre o plugin e o server não é criptografado.

Plugins.MSSQL.Default.HostNameInCertificate

Especifique, por padrão, o nome comum (CN) do certificado do MSSQL server.

Plugins.MSSQL.Default.Password

Especifique a senha padrão para um servidor MSSQL protegido.

Plugins.MSSQL.Default.TLSMinVersion

Especifique a versão mínima do TLS por padrão.

Intervalo: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Confia no certificado do server sem validá-lo por padrão.

Intervalo: `true`, `false`

Plugins.MSSQL.Default.Uri

Especifique a URI padrão para conectar ao server MSSQL.
 As credenciais incorporadas serão ignoradas.
 O único esquema suportado é `sqlserver://`.
 O esquema pode ser omitido.

Padrão: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Especifique o nome de usuário padrão para um servidor MSSQL protegido.

Plugins.MSSQL.KeepAlive

Especifica o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60-900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Especifique o caminho do arquivo da sessão nomeada para o certificado de chave pública da autoridade certificadora (CA) que emitiu o certificado do server MSSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 O certificado deve estar no formato PEM.

Plugins.MSSQL.Sessions.<SessionName>.Database

Especifique o nome do banco de dados da sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Especifique o nível de criptografia da sessão nomeada.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Intervalo:
true - os dados enviados entre o plugin e o server são criptografados;
false - os dados enviados entre o plugin e o server não são criptografados além do pacote de login;
strict - os dados enviados entre o plugin e o server são criptografados de ponta a ponta usando TDS8;
disable - os dados enviados entre o plugin e o server não são criptografados.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Especifique 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

Especifique a senha da sessão nomeada para um servidor MSSQL protegido.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Especifique a versão mínima do TLS da sessão nomeada.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Intervalo: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Confiar no certificado do servidor da sessão nomeada sem validá-lo.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Intervalo: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o server MSSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 As credenciais incorporadas serão ignoradas.
 O único esquema suportado é `sqlserver://`.
 O esquema pode ser omitido.

Padrão: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para um servidor MSSQL protegido.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.MSSQL.System.Path

Especifique o caminho para o executável do plugin MSSQL.
 Configuração global para o plugin MSSQL. Aplicada a todas as conexões.

Exemplo: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Defina a duração, em segundos, para aguardar uma resposta do server ao se conectar pela primeira vez e nas operações subsequentes na sessão.

Padrão: `global timeout`
 Intervalo: 1-30

Opções

Parameter	Description
-V --version	Exibe a versão do plugin e informações de licença.
-h --help	Exibe informações de ajuda (atalho).
-t, --test <item key>	Inicia o plugin para teste (a configuração do plugin é ignorada).

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin MySQL

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin MySQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `mysql.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Mysql.CallTimeout	Especifica o tempo máximo, em segundos, para aguardar a conclusão de uma solicitação.
Plugins.Mysql.CustomQueriesPath	Especifica o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.
Plugins.Mysql.CustomQueriesEnabled	Habilita a execução da chave de item `mysql.custom.query`.
Plugins.Mysql.Default.Password	Especifica a senha padrão.
Plugins.Mysql.Default.TLSConnect	Especifica o nível padrão de criptografia para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
Plugins.Mysql.Default.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da CA de nível superior para verificação do certificado do par.
Plugins.Mysql.Default.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados.
Plugins.Mysql.Default.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do banco de dados.
Plugins.Mysql.Default.Uri	Especifica a URI padrão.
Plugins.Mysql.Default.User	Especifica o nome de usuário padrão.
Plugins.Mysql.KeepAlive	Especifica o tempo máximo de espera (em segundos) antes que conexões não utilizadas do plugin sejam encerradas.
Plugins.Mysql.Sessions.<SessionName>.Password	Especifica a senha da sessão nomeada.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Especifica o nível de criptografia da sessão nomeada para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da CA de nível superior para verificação do certificado do par da sessão nomeada.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados da sessão nomeada.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do banco de dados da sessão nomeada.
Plugins.Mysql.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada.
Plugins.Mysql.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada.
Plugins.Mysql.Timeout	Define o tempo máximo, em segundos, de espera quando uma conexão precisa ser estabelecida.

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 que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Mysql.CallTimeout

Especifique o tempo máximo, em segundos, para aguardar a conclusão de uma solicitação.

Padrão: `global timeout`
 Intervalo: 1–30

Plugins.Mysql.CustomQueriesPath

Especifique o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.

Padrões:
`/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`)

Plugins.Mysql.CustomQueriesEnabled

Habilita a execução da chave de item `mysql.custom.query`.
 Se desabilitado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.

Padrão: `false`
 Intervalo: `true/false`

Plugins.Mysql.Default.Password

Especifique a senha padrão para um server MySQL protegido.

Plugins.Mysql.Default.TLSConnect

Especifique o nível padrão de criptografia para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.

Intervalo:
required - exigir conexão TLS;
verify_ca - verificar certificados;
verify_full - verificar certificados e endereço IP

Plugins.Mysql.Default.TLSCAFile

Especifique o caminho completo do arquivo que contém 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.

Obrigatório: `yes`, se `Plugins.Mysql.Default.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.Mysql.Default.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent ou a cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.

Obrigatório: `yes`, se `Plugins.Mysql.Default.TLSKeyFile` estiver especificado

Plugins.Mysql.Default.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do banco de dados, usada para comunicações criptografadas entre o agent 2 do Zabbix e os bancos de dados monitorados.

Obrigatório: `yes`, se `Plugins.Mysql.Default.TLSCertFile` estiver especificado

Plugins.Mysql.Default.Uri

Especifique a URI padrão para conexão com o server MySQL.
 As credenciais incorporadas serão ignoradas.
 Esquemas suportados: `tcp`, `unix`
 O esquema pode ser omitido.
 A porta pode ser omitida (padrão=3306).

Exemplos:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

Especifique o nome de usuário padrão para um servidor MySQL protegido.

Plugins.Mysql.KeepAlive

Especifique o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60–900

Plugins.Mysql.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para um servidor MySQL protegido.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Especifique o nível de criptografia da sessão nomeada para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Intervalo:
required - exigir conexão TLS;
verify_ca - verificar certificados;
verify_full - verificar certificados e endereço IP

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior para a verificação do certificado do par na sessão nomeada, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent da sessão nomeada ou a cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile` estiver especificado

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do banco de dados da sessão nomeada, usada para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile` estiver especificado

Plugins.Mysql.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o server MySQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 As credenciais incorporadas serão ignoradas.
 Esquemas suportados: `tcp`, `unix`
 O esquema pode ser omitido.
 A porta pode ser omitida (padrão=3306).

Exemplos:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para um servidor MySQL protegido.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Mysql.Timeout

Define o tempo máximo, em segundos, de espera ao estabelecer uma conexão.

Padrão: `global timeout`
 Intervalo: 1–30

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin NVIDIA GPU

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin NVIDIA GPU.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `nvidiagpu.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

O plugin NVIDIA GPU é um plugin carregável e está disponível, com descrição completa, no repositório do plugin NVIDIA GPU.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.NVIDIA.System.Path	Especifica o caminho para o executável do plugin NVIDIA GPU.
Plugins.NVIDIA.Timeout	Especifica o tempo máximo (em segundos) para aguardar uma resposta do server durante as tentativas de conexão e as operações subsequentes na sessão.

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 somente em codificação UTF-8, sem BOM;
- Comentários que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.NVIDIA.System.Path

Especifique o caminho para o executável do plugin da GPU NVIDIA.

Exemplo:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Especifique o tempo máximo (em segundos) para aguardar uma resposta do server durante as tentativas de conexão e as operações subsequentes na sessão.
 O timeout global do tipo item ou o timeout individual do item substituirá este valor se for maior.

Padrão: `global timeout`
 Intervalo: 1–30

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin Oracle

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Oracle.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `oracle.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Oracle.CallTimeout	O tempo máximo de espera, em segundos, para a conclusão de uma solicitação. Atualmente, este parâmetro não tem efeito; em vez disso, é usada a configuração de item timeout.
Plugins.Oracle.ConnectTimeout	Especifique o tempo máximo de espera, em segundos, para que uma conexão seja estabelecida.
Plugins.Oracle.CustomQueriesPath	Especifique o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.
Plugins.Oracle.CustomQueriesEnabled	Habilite a execução da chave de item `oracle.custom.query`.
Plugins.Oracle.Default.Password	Especifique a senha padrão.
Plugins.Oracle.Default.Service	Especifique o nome do serviço padrão.
Plugins.Oracle.Default.URI	Especifique a URI padrão.
Plugins.Oracle.Default.User	Especifique o nome de usuário padrão.
Plugins.Oracle.KeepAlive	Defina o tempo máximo de espera (em segundos) antes que conexões do plugin não utilizadas sejam fechadas.
Plugins.Oracle.ResolveTNS	Especifique como interpretar a string de conexão (ConnString) para conectar ao server Oracle.
Plugins.Oracle.Sessions.<SessionName>.Password	Especifique a senha da sessão nomeada.
Plugins.Oracle.Sessions.<SessionName>.Service	Especifique o nome do serviço da sessão nomeada.
Plugins.Oracle.Sessions.<SessionName>.URI	Especifique a URI da sessão nomeada.
Plugins.Oracle.Sessions.<SessionName>.User	Especifique o nome de usuário da sessão nomeada.

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 somente em codificação UTF-8 sem BOM.
- Comentários que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Oracle.CallTimeout

O tempo máximo de espera, em segundos, para que uma solicitação seja concluída.
 Atualmente, este parâmetro não tem efeito. O tempo máximo de espera é determinado pelo item timeout.

Padrão: `Timeout`[1]
 Intervalo: 1-30

Plugins.Oracle.ConnectTimeout

Especifique o tempo máximo de espera, em segundos, para que uma conexão seja estabelecida.

Padrão: `Timeout`[1]
 Intervalo: 1-30

Plugins.Oracle.CustomQueriesPath

Especifique o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.

Padrões:
`/usr/local/share/zabbix/custom-queries/oracle` - 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`)

Exemplo:

```
/etc/zabbix/oracle/sql
```

Plugins.Oracle.CustomQueriesEnabled

Ativa a execução da chave de item `oracle.custom.query`.
 Se desativado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.

Padrão: `false`
 Intervalo: `true/false`

Plugins.Oracle.Default.Password

Especifique a senha padrão para conexão com o Oracle.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Oracle.Default.Service

Especifique o nome de serviço padrão para conexão com o Oracle (SID não é suportado).
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Oracle.Default.Uri

Especifique a URI padrão para conexão com o Oracle.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
 Credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato da URI.
 Esquemas suportados: `tcp`, `tcps`
 O esquema pode ser omitido.
 A porta pode ser omitida (padrão=1521).
 Também é possível especificar a chave TNS ou o valor TNS como string de conexão. O valor TNS deve ser composto sem espaços em branco.

Exemplos:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Default.User

Especifique o 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

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60-900

Plugins.Oracle.ResolveTNS

Especifique como interpretar a string de conexão (ConnString) para se conectar ao servidor Oracle.
 Se estiver definido como `false`, o esquema e a porta padrão serão adicionados a ConnString (se estiverem ausentes).
 Se a opção estiver definida como `true`, o esquema e a porta padrão serão omitidos (a menos que sejam especificados explicitamente em ConnString), e ConnString será passada ao cliente Oracle como está. Se o cliente Oracle encontrar ConnString no arquivo `tnsnames.ora`, a descrição de conexão encontrada será usada para se conectar ao servidor Oracle.

Padrão: `true`
 Intervalo: `true/false`

Plugins.Oracle.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para conexão com o Oracle.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Plugins.Oracle.Sessions.<SessionName>.Service

Especifique o nome do serviço da sessão nomeada para conexão com o Oracle (SID não é suportado).
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.Oracle.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o Oracle.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.
 As credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato da URI.
 Esquemas suportados: `tcp`, `tcps`
 O esquema pode ser omitido.
 A porta pode ser omitida (padrão=1521).
 Também é possível especificar a chave TNS ou o valor TNS como string de conexão. O valor TNS deve ser composto sem espaços em branco.

Exemplos:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para conexão com o Oracle.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Notas de rodapé

[1] O tempo limite é definido pela opção `Timeout` no arquivo de configuração (`zabbix_agent2.conf`) do agent 2 (UNIX) ou do agent 2 (Windows) do Zabbix.

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin PostgreSQL

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin PostgreSQL.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `postgresql.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

O plugin PostgreSQL é um plugin carregável e está disponível, com descrição completa, no repositório do plugin PostgreSQL.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.PostgreSQL.CallTimeout	Define o tempo máximo de espera, em segundos, para que uma solicitação seja concluída.
Plugins.PostgreSQL.CustomQueriesPath	Especifica o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.
Plugins.PostgreSQL.CustomQueriesEnabled	Habilita a execução da chave de item `postgresql.custom.query`.
Plugins.PostgreSQL.Default.CacheMode	Especifica o modo de cache padrão para a conexão PostgreSQL.
Plugins.PostgreSQL.Default.Database	Especifica o nome padrão do banco de dados.
Plugins.PostgreSQL.Default.Password	Especifica a senha padrão.
Plugins.PostgreSQL.Default.TLSCAFile	Especifica o caminho completo do arquivo que contém o certificado da(s) CA(s) de nível superior para verificação do certificado do par.
Plugins.PostgreSQL.Default.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado PostgreSQL ou a cadeia de certificados.
Plugins.PostgreSQL.Default.TLSConnect	Especifica o nível padrão de criptografia para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
Plugins.PostgreSQL.Default.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada PostgreSQL.
Plugins.PostgreSQL.Default.Uri	Especifica a URI padrão.

Parameter	Description
Plugins.PostgreSQL.Default.User	Especifica o nome de usuário padrão.
Plugins.PostgreSQL.KeepAlive	Define o tempo máximo de espera, em segundos, antes que conexões do plugin não utilizadas sejam encerradas.
Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode	Especifica o modo de cache da sessão nomeada para a conexão PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Especifica o nome do banco de dados da sessão nomeada.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Especifica a senha da sessão nomeada.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Especifica o caminho completo do arquivo que contém o certificado da(s) CA(s) de nível superior para verificação do certificado do par.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado PostgreSQL ou a cadeia de certificados.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Especifica o nível de criptografia da sessão nomeada para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Especifica a URI da sessão nomeada.
Plugins.PostgreSQL.Sessions.<SessionName>.User	Especifica o nome de usuário da sessão nomeada.

|Plugins.PostgreSQL.System.Path|Especifica o caminho para o executável do plugin PostgreSQL.| |Plugins.PostgreSQL.Timeout|Define o tempo limite de execução da solicitação.|

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 somente em codificação UTF-8 sem BOM.
- Comentários que começam com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.PostgreSQL.CallTimeout

Define o tempo máximo de espera, em segundos, para que uma solicitação seja concluída.

Padrão: `global timeout`
 Intervalo: 1–30

Plugins.PostgreSQL.CustomQueriesPath

Especifique o caminho completo do diretório que contém arquivos `.sql` com consultas personalizadas.

Padrões:
`/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`)

Exemplo:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Ativa a execução da chave de item `postgresql.custom.query`.
 Se desativado, nenhuma consulta será carregada do caminho do diretório de consultas personalizadas.

Padrão: `false`
 Intervalo: `true/false`

Plugins.PostgreSQL.Default.CacheMode

Especifique o modo de cache padrão para a conexão PostgreSQL.

Padrão: `prepare`
 Intervalo:
`prepare` - cria prepared statements no server PostgreSQL;
`describe` - usa o anonymous prepared statement para descrever uma statement sem criar uma statement no server.

Observe que `describe` é útil principalmente quando o ambiente não permite prepared statements, como ao usar um connection pooler como o PgBouncer.

Plugins.PostgreSQL.Default.Database

Especifique o 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

Especifique a senha padrão para conexão com o PostgreSQL.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.PostgreSQL.Default.TLSCAFile

Especifique o caminho completo do arquivo que contém o(s) certificado(s) da CA de nível superior para a verificação do certificado do par nas comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 Usado se nenhum valor for especificado em uma sessão nomeada.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Default.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.PostgreSQL.Default.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado PostgreSQL ou a 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.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Default.TLSKeyFile` estiver especificado

Plugins.PostgreSQL.Default.TLSConnect

Especifique o nível padrão de criptografia para as comunicações entre o Zabbix agent 2 e os bancos de dados monitorados.
 Usado se nenhum valor for especificado em uma sessão nomeada.

Intervalo:
`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 certificado.

Tipo de criptografia indefinido significa conexão sem criptografia.

Plugins.PostgreSQL.Default.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do PostgreSQL para comunicações criptografadas entre o agent 2 do Zabbix e os bancos de dados monitorados.
 Usado se nenhum valor for especificado em uma sessão nomeada.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Default.TLSCertFile` estiver especificado

Plugins.PostgreSQL.Default.Uri

Especifique a URI padrão para conexão com o PostgreSQL.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
 Credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato de 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

Especifique o nome de usuário padrão para conectar ao PostgreSQL.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.PostgreSQL.KeepAlive

Defina o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Especifique o modo de cache da sessão nomeada para a conexão PostgreSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Padrão: `prepare`
 Intervalo:
`prepare` - cria prepared statements no server PostgreSQL;
`describe` - usa o prepared statement anônimo para descrever uma instrução sem criar uma instrução no server.

Observe que `describe` é útil principalmente quando o ambiente não permite prepared statements, como ao executar um pooler de conexão como o PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Especifique o banco de dados da sessão nomeada para conexão com o PostgreSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para conexão com o PostgreSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.
 Deve corresponder ao formato da senha.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Especifique o caminho completo do arquivo que contém o certificado da CA de nível superior para verificação do certificado do par em comunicações criptografadas entre o agent 2 do Zabbix e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado PostgreSQL ou a cadeia de certificados para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile` estiver especificado

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Especifique o nível nomeado de criptografia da sessão para comunicações entre o agent 2 do Zabbix e os bancos de dados monitorados.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Intervalo:
`required` - conecta usando TLS como modo de transporte sem verificações de identidade;
`verify_ca` - conecta usando TLS e verifica o certificado;
`verify_full` - conecta usando TLS, verifica o certificado e verifica se a identidade do banco de dados (CN) especificada por DBHost corresponde ao certificado.

Um tipo de criptografia indefinido significa uma conexão sem criptografia.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do PostgreSQL para comunicações criptografadas entre o agent 2 do Zabbix e os bancos de dados monitorados.
 <SessionName> - defina o nome de uma sessão para uso nas chaves de item.

Obrigatório: `yes`, se `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile` estiver especificado.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o PostgreSQL.
 <SessionName> - define o nome de uma sessão para uso em item keys.
 As credenciais incorporadas serão ignoradas.
 Deve corresponder ao formato de 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

Especifique o nome de usuário da sessão nomeada para conexão com o PostgreSQL.
 <SessionName> - define o nome de uma sessão para uso nas chaves de item.

Plugins.PostgreSQL.System.Path

Especifique o caminho para o executável do plugin PostgreSQL.

Obrigatório: `yes`

Exemplo:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Defina o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: `global timeout`
 Intervalo: 1–30

Opções

Parameter	Description
-V --version	Exibe a versão do plugin e as informações de licença.
-h --help	Exibe as informações de ajuda (atalho).
-t, --test <item key>	Inicia o plugin para teste (a configuração do plugin é ignorada).

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin Redis

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin Redis.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `redis.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Redis.Default.Password	Especifica a senha padrão para conexão com o Redis.
Plugins.Redis.Default.TLSConnect	Especifica o nível de criptografia exigido para as comunicações do Zabbix agent 2.
Plugins.Redis.Default.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior.
Plugins.Redis.Default.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent.
Plugins.Redis.Default.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent.
Plugins.Redis.Default.Uri	Especifica a URI padrão para conexão com o Redis.
Plugins.Redis.Default.User	Especifica o nome de usuário padrão para conexão com o Redis.
Plugins.Redis.KeepAlive	Define o tempo máximo de espera (em segundos) antes que conexões não utilizadas do plugin sejam encerradas.
Plugins.Redis.Sessions.<Session name>.Password	Especifica a senha da sessão nomeada para conexão com o Redis.
Plugins.Redis.Sessions.<Session name>.TLSConnect	Especifica o nível de criptografia exigido para as comunicações do Zabbix agent 2 para a sessão nomeada.
Plugins.Redis.Sessions.<Session name>.TLSCAFile	Especifica o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior para a sessão nomeada.
Plugins.Redis.Sessions.<Session name>.TLSCertFile	Especifica o caminho completo do arquivo que contém o certificado do agent para a sessão nomeada.
Plugins.Redis.Sessions.<Session name>.TLSKeyFile	Especifica o caminho completo do arquivo que contém a chave privada do agent para a sessão nomeada.
Plugins.Redis.Sessions.<Session name>.Uri	Especifica a URI da sessão nomeada para conexão com o Redis.
Plugins.Redis.Sessions.<Session name>.User	Especifica o nome de usuário da sessão nomeada para conexão com o Redis.
Plugins.Redis.Timeout	Define o tempo limite de execução da requisição.

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 que começam com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Redis.Default.Password

Especifique a senha padrão para conexão com o Redis.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Plugins.Redis.Default.TLSConnect

Especifique o nível de criptografia exigido para as comunicações do Zabbix agent 2.
 Um valor vazio ou não definido é tratado como sem TLS.

Intervalo: `required` - criptografado, não verificado (apenas para testes), `verify_ca` - criptografado, certificado do servidor, `verify_full` - criptografado, certificado do servidor e nome do servidor verificados com SAN

Plugins.Redis.Default.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior.

Plugins.Redis.Default.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent.

Plugins.Redis.Default.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent.

Plugins.Redis.Default.Uri

Especifique a URI padrão para conexão com o Redis.
 Usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
 Credenciais incorporadas serão ignoradas. Deve corresponder ao formato de URI.
 Esquemas suportados: `tcp`, `unix`
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=6379).

Padrão: `tcp://localhost:6379`

Exemplos:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Especifique o nome de usuário padrão para se conectar ao Redis.
 Usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.

Padrão: `default`

Plugins.Redis.KeepAlive

Define o tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Padrão: 300
 Intervalo: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Especifique a senha da sessão nomeada para conexão com o Redis.
 <SessionName> - defina o nome da sessão a ser usado nas chaves do item.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Especifique o nível de criptografia exigido para as comunicações do Zabbix agent 2 para a sessão nomeada.
 <SessionName> - define o nome da sessão a ser usado nas chaves de item.
 Um valor vazio ou não definido é tratado como ausência de TLS.

Intervalo: `required` - criptografado, não verificado (apenas para testes), `verify_ca` - criptografado, certificado do servidor, `verify_full` - criptografado, certificado do servidor e nome do servidor verificado com SAN

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Especifique o caminho completo do arquivo que contém os certificados da CA de nível superior para a sessão nomeada.
 <SessionName> - define o nome da sessão a ser usado nas chaves de item.

Obrigatório: `yes`, se `Plugins.Redis.Sessions.<SessionName>.TLSConnect` estiver definido como `verify_ca` ou `verify_full`

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Especifique o caminho completo do arquivo que contém o certificado do agent para a sessão nomeada.
 <SessionName> - define o nome da sessão a ser usado nas chaves de item.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Especifique o caminho completo do arquivo que contém a chave privada do agent para a sessão nomeada.
 <SessionName> - define o nome da sessão a ser usado nas chaves de item.

Plugins.Redis.Sessions.<SessionName>.Uri

Especifique a URI da sessão nomeada para conexão com o Redis.
 <SessionName> - define o nome da sessão a ser usado nas chaves de item.
 As credenciais incorporadas serão ignoradas. Deve corresponder ao formato de URI.
 Esquemas suportados: `tcp`, `unix`
 Um esquema pode ser omitido.
 Uma porta pode ser omitida (padrão=6379).

Padrão: `tcp://localhost:6379`

Exemplos:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Especifique o nome de usuário da sessão nomeada para conexão com o Redis.
 <SessionName> - defina o nome da sessão a ser usado nas chaves do item.

Plugins.Redis.Timeout

Defina o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: `global timeout`
 Intervalo: 1–30

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Plugin SMART

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do plugin SMART.

Recomenda-se especificá-los em seu próprio arquivo de configuração (por exemplo, `smart.conf`), em vez de no arquivo de configuração do agent 2. Em seguida, use a diretiva Include para incluir esse arquivo na configuração do Zabbix agent 2.

Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique em seu nome)
- Detalhes dos parâmetros

Parameter	Description
Plugins.Smart.Path	Especifica o caminho para o executável smartctl.
Plugins.Smart.Timeout	Define o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

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 oferece suporte a arquivos de configuração somente em codificação UTF-8 sem BOM;
- Comentários que começam com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Plugins.Smart.Path

Especifique o caminho para o executável smartctl.

Padrão: `smartctl`

Plugins.Smart.Timeout

Define o tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Padrão: `global timeout<br>` Intervalo: 1-30

Veja também

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

Protocolo do plugin do Zabbix agent 2

O protocolo do Zabbix agent 2 é baseado em código, tamanho e modelo de dados.

Código

Type	Size	Comments
Byte	4	Tipo de payload, atualmente apenas JSON é suportado.

Tamanho

Type	Size	Comments
Byte	4	Tamanho da carga útil atual em bytes.

Dados de payload

Type	Size	Comments
Byte	Definido pelo campo *Size*	Dados formatados em JSON.

Definição dos dados de payload

Dados comuns

Estes parâmetros estão presentes em todas as solicitações/respostas:

Name	Type	Comments
id	uint32	Para solicitações - o identificador incremental usado para vincular solicitações às respostas. Exclusivo 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 da solicitação.

Solicitação de log

Uma solicitação enviada por um plugin para gravar uma mensagem de log no arquivo de log do agent.

direction	plugin → agent
response	no

Parâmetros específicos para solicitações de log:

Name	Type	Comments
severity	uint32	A severidade da mensagem (nível de log).
message	string	A mensagem a ser registrada no log.

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.

direction	agent → plugin
response	yes

Parâmetros específicos de solicitações de registro:

Name	Type	Comments
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:

Name	Type	Comments
name	string	O nome do plugin.
metrics	array of strings (optional)	As métricas com descrições, conforme usadas no plugin. Retorna RegisterMetrics(). Ausente se um erro for retornado.
interfaces	uint32 (optional)	A máscara de bits das interfaces suportadas pelo plugin. Ausente se um erro for retornado.
error	string (optional)	Uma mensagem de erro retornada se um plugin não puder ser iniciado. Ausente, se metrics forem retornadas.

Exemplos:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

ou

```
{"id":2,"type":3,"error":"error message"}
```

Iniciar solicitação

Uma solicitação para executar a função Start da interface Runner.

direction	agent → plugin
response	no

A solicitação não possui parâmetros específicos; ela contém apenas parâmetros de common data.

Exemplo:

```
{"id":3,"type":4}
```

Encerrar solicitação

Uma solicitação enviada pelo agent para desligar um plugin.

direction	agent → plugin
response	no

A solicitação não possui parâmetros específicos, ela contém apenas parâmetros de dados comuns.

Exemplo:

```
{"id":3,"type":5}
```

Solicitação de exportação

Uma solicitação para executar a função Export da interface do Exporter.

direction	agent → plugin
response	no

Parâmetros específicos para solicitações de exportação:

Name	Type	Comments
key	string	A chave do plugin.
parameters	array of strings (optional)	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 do Exporter.

direction	plugin → agent
response	n/a

Parâmetros específicos para respostas de exportação:

Name	Type	Comments
value	string (optional)	Valor de resposta da função Export. Ausente, se um erro for retornado.
error	string (optional)	Mensagem de erro se a função Export não tiver sido executada com sucesso. Ausente, se um 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*:

Name	Type	Comments
global_options	JSON object	Objeto JSON contendo opções globais de configuração do agent.
private_options	JSON object (optional)	Objeto JSON contendo opções privadas de configuração do plugin, se fornecidas.

Exemplo:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Validar solicitação

Uma solicitação para executar a função *Validate* da interface *Configurator*.

direction	agent → plugin
response	yes

Parâmetros específicos para solicitações *Validate*:

Name	Type	Comments
private_options	JSON object (optional)	Objeto JSON contendo opções privadas de configuração do plugin, se fornecidas.

Exemplo:

```
{"id":7,"type":9,"private_options":{...}}
```

Validar resposta

Resposta da função *Validate* da interface *Configurator*.

direction	plugin → agent
response	n/a

Parâmetros específicos das respostas de *Validate*:

Name	Type	Comments
error	string (optional)	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":"error message"}
```

Java gateway

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

O suporte nativo ao 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 usa a API de gerenciamento JMX para consultar remotamente a aplicação de interesse. Não é necessário instalar software adicional na aplicação; basta iniciá-la com a opção `-Dcom.sun.management.jmxremote` na linha de comando.

O Java gateway aceita conexões de entrada do Zabbix server ou do proxy e só pode ser usado como um "proxy passivo". Ao contrário do proxy do Zabbix, ele também pode ser usado a partir de um proxy do Zabbix (os proxies do Zabbix não podem ser encadeados). O acesso a cada Java gateway é configurado diretamente no arquivo de configuração do server do Zabbix ou do proxy do Zabbix; portanto, apenas um Java gateway pode ser configurado por Zabbix server ou Zabbix proxy. Se um host tiver itens do tipo *JMX agent* e itens de outros tipos, somente os itens *JMX agent* serão enviados ao Java gateway para obtenção.

Quando um item precisa ser atualizado por meio do Java gateway, o Zabbix server ou o proxy se conecta ao Java gateway e solicita o valor, que o Java gateway, por sua vez, recupera e devolve ao server ou ao proxy. Dessa forma, o Java gateway não armazena nenhum valor em cache.

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

Veja também:

- Parâmetros de configuração: Zabbix Java gateway
- Configuração a partir de pacotes Debian/Ubuntu
- Configuração a partir de pacotes RHEL
- Configuração a partir do código-fonte

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

É recomendável que `StartJavaPollers` seja menor ou igual a `START_POLLERS`; caso contrário, pode haver situações em que não haja threads disponíveis no Java gateway para atender às solicitações recebidas. Nesses casos, o Java gateway usa `ThreadPoolExecutor.CallerRunsPolicy`, o que significa que a thread principal atenderá à solicitação recebida e, temporariamente, não aceitará novas solicitações.

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.

Parâmetros de configuração: Zabbix Java gateway

Se você usar os scripts `startup.sh` e `shutdown.sh` para iniciar o Zabbix Java gateway, então poderá especificar os parâmetros de configuração necessários no arquivo `settings.sh`. Os scripts de inicialização e desligamento carregam o arquivo de configurações e cuidam da conversão das variáveis de shell (listadas na primeira coluna) para propriedades Java (listadas na segunda coluna).

Se você iniciar o Zabbix Java gateway manualmente executando `java` diretamente, então deverá especificar as propriedades Java correspondentes na linha de comando.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	Endereço IP para escutar.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Porta para escutar.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Nome do arquivo PID. Se omitido, o Zabbix Java Gateway é iniciado como um aplicativo de console.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	no			Nome do arquivo de propriedades. Pode ser usado para definir propriedades adicionais usando um formato de chave-valor, de modo que elas não fiquem visíveis na linha de comando ou para sobrescrever as existentes. Por exemplo: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Número de threads de trabalho a iniciar.
TIMEOUT	zabbix.timeout	no	1-30	3	Tempo de espera para operações de rede (em segundos).

Warning:
A porta 10052 não está registrada na IANA.

Configuração 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.

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 Java gateway

Com o Java gateway em execução, você precisa 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 em vez disso.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. No entanto, se você quiser usá-lo, será necessário especificar o número de instâncias pré-fork de Java pollers. Você faz isso da mesma forma que especifica pollers e trappers regulares.

```
StartJavaPollers=5
```

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

Depuração do 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 logging, edite o arquivo:

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

e altere `info` em `level="info"` para debug ou até `trace` (para solução de problemas aprofundada):

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

</configuration>
```

Monitoramento JMX

Consulte a página JMX monitoring para mais detalhes.

Configuração a partir de pacotes RHEL

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

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 Java gateway

Com o Java gateway em execução, você precisa 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 em vez disso.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. No entanto, se você quiser usá-lo, será necessário especificar o número de instâncias pré-fork de Java pollers. Você faz isso da mesma forma que especifica pollers e trappers regulares.

```
StartJavaPollers=5
```

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

Depuração do 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 logging, edite o arquivo:

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

e altere `info` em `level="info"` para debug ou até mesmo `trace` (para solução de problemas aprofundada):

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

</configuration>
```

Monitoramento JMX

Consulte a página JMX monitoring para mais detalhes.

Configuração a partir do código-fonte

Visão geral

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

Visão geral dos arquivos

Se você obteve o Java gateway a partir do código-fonte, deverá ter terminado com uma coleção de scripts de shell, arquivos JAR e arquivos de configuração em `$PREFIX/sbin/zabbix_java`. A função desses arquivos é resumida abaixo.

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

O próprio arquivo JAR do Java gateway.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

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

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

Arquivos de configuração do Logback.

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

Scripts auxiliares para iniciar e interromper o Java gateway.

`settings.sh`

Arquivo de configuração carregado pelos scripts de inicialização e encerramento 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 especificá-la no script `settings.sh`. Consulte a descrição do arquivo de configuração do Java gateway para saber como especificar essa e outras opções.

> **Warning:**
> A porta 10052 não está registrada na IANA.

Depois de se sentir confortável com as configurações, você pode iniciar o Java gateway executando o script `startup.sh`:

```
./startup.sh
```

Da mesma forma, quando não precisar mais do Java gateway, execute o script `shutdown.sh` para interrompê-lo:

```
./shutdown.sh
```

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

Configurando o server para uso com Java gateway

Com o Java gateway em execução, você precisa 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 em vez disso.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. No entanto, se você quiser usá-lo, precisa especificar o número de instâncias pré-forked de Java pollers. Você faz isso da mesma forma que especifica pollers e trappers regulares.

```
StartJavaPollers=5
```

Não se esqueça de reiniciar o server ou o proxy, assim que terminar de configurá-los.

Depuração do Java gateway

Caso haja algum problema com o Java gateway ou a mensagem de erro que você vê sobre um item no frontend não seja descritiva o suficiente, talvez seja útil consultar o arquivo de log do Java gateway.

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

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

Observe que, ao contrário do Zabbix server ou do Zabbix proxy, não é necessário 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 suas necessidades. Consulte o Manual do Logback para mais detalhes.

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

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

Monitoramento JMX

Consulte a página JMX monitoring para mais detalhes.

Remetente

Visão geral O Zabbix sender é um utilitário de linha de comando que pode ser usado para enviar dados de desempenho ao Zabbix server ou proxy para processamento.

O utilitário geralmente é usado em scripts de usuário de longa execução para o envio periódico de dados de disponibilidade e desempenho.

Para enviar resultados diretamente ao Zabbix server ou proxy, um item do tipo *Zabbix trapper* deve ser configurado.

Veja também:

- Protocolo do Zabbix sender
- Biblioteca de vínculo dinâmico do Zabbix sender para Windows
- Biblioteca Python para Zabbix, que possui funcionalidade integrada para atuar como Zabbix sender

Executando o Zabbix sender Um exemplo de execução do Zabbix sender para UNIX:

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

onde:

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

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

O Zabbix sender pode ser usado para enviar vários valores a partir de um arquivo de entrada. Consulte a página de manual do Zabbix sender para obter mais informações.

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

O Zabbix sender aceita strings na codificação UTF-8 (tanto para sistemas semelhantes ao Unix quanto para Windows) sem uma marca de ordem de bytes (BOM) no início do arquivo.

No Windows, o Zabbix sender pode ser executado de forma semelhante:

```
zabbix_sender.exe [options]
```

Os cenários de envio em tempo real do `zabbix_sender` coletarão vários valores passados a ele em rápida sucessão e os enviarão ao server em uma única conexão. Um valor que não esteja separado do valor anterior por mais de 0,2 segundo pode ser colocado na mesma pilha, mas o tempo máximo de consulta ainda é de 1 segundo.

Note:
O Zabbix sender será encerrado se houver uma entrada de parâmetro inválida (que não siga a notação `parameter=value`) no arquivo de configuração especificado.

Executando o Zabbix sender com descoberta em baixo nível Um exemplo de execução do Zabbix sender para enviar um valor em formato JSON para descoberta em 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 em baixo nível deve ter o tipo de item *Zabbix trapper* (neste exemplo, com a chave `trapper.discovery.item`).

Protocolo do Zabbix sender

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 and data length.

Solicitação do Zabbix sender

```
{
    "request": "dados do sender",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "valor de teste"
        }
    ]
}
```

Resposta do server Zabbix

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Solicitação do Zabbix sender com um timestamp

Como alternativa, o Zabbix sender pode enviar uma solicitação com um 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 server Zabbix

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Biblioteca de vínculo dinâmico do Zabbix sender para Windows

Em um ambiente Windows, aplicações podem enviar dados para o server/proxy do Zabbix usando a biblioteca de vínculo dinâmico do Zabbix sender (`zabbix_sender.dll`) em vez de precisar iniciar um processo externo (`zabbix_sender.exe`).

Os arquivos zabbix_sender.h e zabbix_sender.lib são necessários para compilar aplicações de usuário com zabbix_sender.dll.

Como obter

Há duas maneiras de obter zabbix_sender.dll.

1. Baixe os arquivos zabbix_sender.h, zabbix_sender.lib e zabbix_sender.dll como um arquivo ZIP.

Ao escolher as opções de download, certifique-se de selecionar "No encryption" em *Encryption* e "Archive" em *Packaging*. Em seguida, baixe o Zabbix agent (não o Zabbix agent 2).

Os arquivos zabbix_sender.h, zabbix_sender.lib e zabbix_sender.dll estarão dentro do arquivo ZIP baixado no diretório `bin\dev`. Descompacte os arquivos onde precisar.

2. Compile o zabbix_sender.dll a partir do código-fonte (consulte 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 um utilitário simples do Zabbix sender implementado com a biblioteca de vínculo dinâmico do Zabbix sender para ilustrar o uso da biblioteca;
- arquivo zabbix_sender.h com as funções de interface da biblioteca de vínculo dinâmico do Zabbix sender. Este arquivo contém documentação explicando a finalidade de cada função de interface, seus argumentos e valor de retorno.

Obter

Visão geral Zabbix get é um utilitário de linha de comando que pode ser usado para se comunicar com o agent do Zabbix e recuperar as informações necessárias do agent.

O utilitário é normalmente usado para solucionar problemas em agents do Zabbix.

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

Executando o Zabbix get Um exemplo de execução do Zabbix get no 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 website:

```
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 do 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 o 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. Intervalo válido: 1-600 segundos (padrã
-k --key <item key>                     Especifica a chave do item para a qual recuperar o valor
-P --protocol <value>                   Protocolo usado para se comunicar com o agent. Valores:
                                            auto - conecta usando o protocolo JSON, com fallback e nova te
                                            json - conecta usando o protocolo JSON
                                            plaintext - conecta usando o protocolo plaintext, em que apena
-h --help                               Exibe esta mensagem de ajuda
-V --version                            Exibe o número da versão

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

Veja também a página de manual do Zabbix get para mais informações.

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

```
zabbix_get.exe [options]
```

JS

Visão geral `zabbix_js` é uma utilidade de linha de comando que pode ser usada para testar scripts incorporados.

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

Em caso de erros de compilação ou execução, `zabbix_js` imprimirá o erro em stderr e sairá com o código 1.

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

Uso 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 especificado como nome do arquivo, o script será lido da stdi
-i, --input input-file            Especifica o nome do arquivo com as informações de entrada
                                  O conteúdo será passado para o script como um valor
                                  Mutuamente exclusivo com -p
                                  Se "-" for especificado como nome do arquivo, o script será lido da stdi
-p, --param input-param           Especifica o parâmetro de entrada: a variável a ser passada para o scrip
                                  Mutuamente exclusivo com -i
-l, --loglevel log-level          Especifica o nível de log
                                  Intervalo: 0-5
-t, --timeout timeout             Especifica o tempo limite em segundos
                                  Intervalo válido: 1-600 segundos (padrão: 10 segundos)
-h, --help                        Exibe informações de ajuda
-V, --version                     Exibe o número da versão
-w <webdriver url>                Habilita o monitoramento do navegador
```

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

O script (script-file.js):

```
return value;
```

Uso:

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

Saída: example_value

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

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

```
return value;
```

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

```
Exemplo de informações de entrada do arquivo
```

Uso:

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

Saída: Exemplo de informações de entrada do arquivo

Exemplo 3: Executando um script com entrada lida da stdin

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

```
return value;
```

Uso:

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

Entrada (stdin):

```
Exemplo de entrada da stdin
```

Saída: Exemplo de entrada da stdin

Serviço web

Visão geral O Zabbix web service é um processo usado para comunicação com serviços web externos. Atualmente, o Zabbix web service é usado para gerar e enviar relatórios agendados, com planos de adicionar funcionalidades adicionais no futuro.

O server do Zabbix se conecta ao web service via HTTP(S). O Zabbix web service 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 do Zabbix.

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

Para configurar o serviço web do Zabbix, atualize os parâmetros do arquivo de configuração do serviço web.

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

Parâmetros de configuração: Zabbix web service

Visão geral

O web service do Zabbix é um processo usado para comunicação com serviços web externos.

Os parâmetros suportados pelo arquivo de configuração do web service 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.

Parameter	Description
AllowedIP	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS de servers Zabbix e proxies Zabbix.
DebugLevel	O nível de depuração.
IgnoreURLCertErrors	Especifica o tratamento de erros de validação de certificado TLS ao acessar a URL do frontend.
Include	Você pode incluir arquivos individuais ou todos os arquivos de um diretório no arquivo de configuração.
ListenPort	O serviço escutará nesta porta por conexões do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogType	O tipo de saída do log.
Timeout	O tempo máximo (em segundos) gasto na formatação do report em PDF de um dashboard.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo os certificados da CA 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 serviço ou a cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do serviço, usado para comunicações criptografadas entre componentes do Zabbix.

Todos os parâmetros são não obrigatórios, a menos que seja explicitamente indicado 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 que começam com # são suportados apenas no início da linha.

Detalhes do parâmetro

AllowedIP

Uma lista de endereços IP separados por vírgulas, opcionalmente em notação CIDR, ou nomes DNS de servers Zabbix e proxies Zabbix. As conexões de entrada serão aceitas somente dos hosts listados aqui.
Se o suporte a IPv6 estiver habilitado, então `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` são tratados igualmente e `::/0` permitirá qualquer endereço IPv4 ou IPv6. `0.0.0.0/0` pode ser usado para permitir qualquer endereço IPv4.

Exemplo:

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obrigatório: sim

DebugLevel

Especifique o nível de depuração: - *0* - informações básicas sobre a inicialização e a finalização dos processos do Zabbix; - *1* - informações críticas; - *2* - informações de erro; - *3* - avisos; - *4* - para depuração (produz muitas informações); - *5* - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

IgnoreURLCertErrors

Especifica o tratamento de erros de validação de certificado TLS ao acessar a URL do frontend:
0 - não ignorar erros de certificado;
1 - ignorar erros de certificado.

Padrão: 0
 Intervalo: 0-1

Include

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você o altere durante o tempo de compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_web_service.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê; portanto, considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivo:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, ao editar `include/my_specific.conf`, pode ser criado `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir apenas os arquivos relevantes de um diretório, use o curinga (`*`) para correspondência de padrões.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

O serviço escutará nesta porta as conexões provenientes 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 - desativa a rotação automática de log.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação de arquivo falhar, por qualquer motivo, o arquivo de log existente é truncado e reiniciado.

Padrão: 1
 Intervalo: 0-1024

LogType

O tipo da saída de 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`

Tempo limite

O tempo máximo (em segundos) gasto na formatação do PDF report de um dashboard.

Padrão: 10
 Intervalo: 1-30

TLSAccept

Quais conexões de entrada aceitar: - *unencrypted* - aceita conexões sem criptografia (padrão); - *cert* - aceita conexões com TLS e um certificado.

Padrão: `unencrypted`

TLSCAFile

O caminho completo do arquivo que contém os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O caminho completo do arquivo que contém o certificado do serviço ou a cadeia de certificados, usado para comunicações criptografadas com componentes do Zabbix.

TLSKeyFile

O caminho completo do arquivo que contém a chave privada do serviço, usada para comunicações criptografadas entre os componentes do Zabbix.

Adaptador bridge

O adaptador bridge do Zabbix permite que seu server Zabbix se comunique com o aplicativo móvel do Zabbix.

O aplicativo móvel do Zabbix não se conecta diretamente ao seu server Zabbix. Em vez disso, dois componentes transferem dados entre eles:

- Zabbix Bridge - um serviço hospedado centralmente que troca dados com o aplicativo móvel do Zabbix em seu dispositivo móvel.
- Adaptador bridge do Zabbix - um componente que você instala e executa. Ele conecta seu server Zabbix ao Zabbix Bridge.

O adaptador bridge pode ser instalado no mesmo sistema que o seu server Zabbix ou em um sistema diferente, usando pacotes.

Para configurar o adaptador bridge do Zabbix, atualize seu arquivo de configuração (caminho padrão: `/etc/zabbix/zabbix_bridge_adapt`

Para manter o adaptador bridge do Zabbix seguro, consulte Acesso móvel seguro.

Fluxo de dados Ao usar o aplicativo móvel do Zabbix, os dados trafegam da seguinte forma:

1. Você executa uma ação no aplicativo móvel do Zabbix, que envia uma consulta ao Zabbix Bridge.
2. O Zabbix Bridge encaminha a consulta ao adaptador do Zabbix Bridge como uma solicitação JSON-RPC.
3. O adaptador do Zabbix Bridge verifica se o método da API do Zabbix solicitado é permitido. Se o método não for permitido, o adaptador do Bridge rejeita a solicitação. Se o método for permitido, o adaptador do Bridge envia a solicitação à API do Zabbix.
4. A API do Zabbix processa a solicitação e retorna uma resposta ao adaptador do Zabbix Bridge.
5. O adaptador do Zabbix Bridge retorna a resposta ao Zabbix Bridge.
6. O Zabbix Bridge encaminha a resposta ao aplicativo móvel do Zabbix, que exibe os dados no seu dispositivo móvel.

Parâmetros de configuração: adaptador de bridge do Zabbix

Visão geral

Os parâmetros de configuração permitem personalizar o comportamento do adaptador de bridge do Zabbix.

Especifique os valores dos parâmetros no arquivo de configuração (`zabbix_bridge_adapter.conf`). Todos os parâmetros são **opcionais**, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver os detalhes completos de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
AllowedAPIMethods	Especifique quais métodos da API do Zabbix podem ser chamados por meio do adaptador de bridge do Zabbix.
DataDir	Especifique o diretório onde o arquivo de estado do adaptador de bridge e as chaves de criptografia são criados e armazenados.
DebugLevel	Defina o nível de depuração.
Include	Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
ListenIP	Especifique o endereço IP de escuta do adaptador de bridge para conexões HTTP(S) de entrada do server do Zabbix.
ListenHealthPort	Especifique a porta de escuta do adaptador de bridge para conexões HTTP de entrada de verificação de integridade.
ListenPort	Especifique a porta de escuta do adaptador de bridge para conexões HTTP(S) de entrada do server do Zabbix.
LogFile	Especifique o caminho completo para o arquivo de log do adaptador de bridge.
LogFileSize	Defina o tamanho máximo do arquivo de log (em MB).
LogType	Especifique o tipo de saída do log.
PidFile	Especifique o arquivo PID.
TLSConfig.AllowedCN	Especifique uma lista separada por vírgulas de Common Names permitidos dos clientes (somente `mTLS`).
TLSConfig.CAFile	Especifique o caminho completo para um arquivo que contenha o(s) certificado(s) de CA de nível superior usado(s) para verificar os certificados dos clientes no modo `mTLS`.
TLSConfig.CertFile	Especifique o caminho completo para um arquivo que contenha o certificado ou a cadeia de certificados.
TLSConfig.KeyFile	Especifique o caminho completo para um arquivo que contenha a chave privada.
TLSConfig.Mode	Defina o modo TLS para conexões de entrada em `ListenPort`.
ServerId	Especifique o identificador exclusivo do server do Zabbix. **Obrigatório**.
ZabbixAPITLSCAFile	Especifique o caminho completo para um arquivo que contenha o(s) certificado(s) de CA de nível superior usado(s) pelo adaptador de bridge para verificar o certificado da API do Zabbix.
ZabbixAPITLSConnect	Defina o modo TLS para conexões de saída com a API do Zabbix.
ZabbixAPIURL	Especifique a URL do endpoint da API do Zabbix. **Obrigatório**.
ZabbixBridgeURL	Especifique a URL do Zabbix Bridge para recuperar a configuração do Zabbix Bridge (recuperada quando o adaptador de bridge é iniciado).

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores dos arquivos de configuração fornecidos.
- Os valores são compatíveis com variáveis de ambiente.
- O Zabbix oferece suporte apenas a arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários iniciados com # são compatíveis somente no início da linha.

Detalhes dos parâmetros

AllowedAPIMethods

Especifique quais métodos da API do Zabbix podem ser chamados por meio do adaptador Zabbix bridge.
 Para vários métodos, repita o parâmetro, um método por linha.

Cada valor do parâmetro deve ser um nome de método exato (por exemplo, `problem.get`) ou um padrão curinga que substitua apenas o nome completo da API (por exemplo, `*.get`) ou o nome completo do método (por exemplo, `problem.*`).

Qualquer outro valor é inválido e impede a inicialização do adaptador Zabbix bridge. Isso inclui um curinga isolado (`*`), um padrão curinga parcial (`host*` ou `host*.*`) ou um valor vazio (`AllowedAPIMethods=`).

Exemplo:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

O conjunto mínimo de métodos exigido pelo aplicativo móvel do Zabbix:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
```

```
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

Padrão: não definido (todos os métodos da API do Zabbix são negados)

DataDir

Especifique o diretório onde o arquivo de estado do adaptador bridge e as chaves de criptografia são criados e armazenados.
 O diretório deve ter permissões 0700, e todos os arquivos armazenados nele devem ter permissões 0600.

Padrão: `/var/lib/zabbix-bridge-adapter`

DebugLevel

Defina o nível de depuração:

- 0 - informações básicas sobre a inicialização e a interrupção 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

Incluir

Especifique um diretório ou arquivos individuais a serem incluídos no arquivo de configuração.
 Durante a instalação do Zabbix, esse parâmetro é definido como `/usr/local/etc`, a menos que você altere essa definição durante a compilação.

Regras gerais para o parâmetro `Include`:

- O caminho pode ser relativo ao local do arquivo `zabbix_bridge_adapter.conf`.
- Você pode especificar vários parâmetros `Include` em um único arquivo de configuração.
- O Zabbix processa o parâmetro `Include` assim que o lê. Portanto, considere colocá-lo no final do arquivo de configuração.

Regras para arquivos incluídos:

- Cada arquivo deve ser legível.
- Cada arquivo deve ter a sintaxe correta. Caso contrário, o componente não será iniciado.

Regras para incluir diretórios ou padrões de arquivos:

- Não presuma nenhuma ordem de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética).
- Não defina o mesmo parâmetro em mais de um arquivo incluído (por exemplo, para substituir uma configuração geral por uma específica), pois não é possível prever qual valor será usado.
- Fique atento às cópias de backup que alguns editores de texto criam automaticamente (por exemplo, editar `include/my_specific.co` pode criar `include/my_specific.conf.BAK`), pois ambos os arquivos serão incluídos.

Para incluir somente os arquivos relevantes de um diretório, use o caractere curinga (*) para correspondência de padrões.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Especifique o endereço IP de escuta do adaptador de bridge para conexões HTTP(S) de entrada do Zabbix server.
 Se não for especificado, o adaptador de bridge escutará em todos os endereços IP.

ListenHealthPort

Especifique a porta de escuta do adaptador de bridge para conexões HTTP de verificação de integridade recebidas.
 Disponibiliza os endpoints `/health` e `/ready`.
 Se não for especificada, o serviço de integridade não será iniciado.

Padrão: 8081
 Intervalo: 1024-32767

ListenPort

Especifique a porta de escuta do adaptador bridge para conexões HTTP(S) de entrada do Zabbix server.

Padrão: 10054
 Intervalo: 1024-32767

LogFile

Especifique o caminho completo para o arquivo de log do adaptador de bridge.

Obrigatório: **Sim** se LogType estiver definido como `file`

Padrão: `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Defina o tamanho máximo do arquivo de log (em MB).
 0 - desabilita a rotação automática de logs.

Padrão: 1
 Intervalo: 0-1024

LogType

Especifique 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`

PidFile

Especifique o arquivo PID.

Padrão: `/tmp/zabbix_bridge_adapter.pid`

ServerId

Especifique o identificador exclusivo do Zabbix server.
 Disponível em *Relatórios* > *Informações do sistema* na interface de usuário do Zabbix.

Obrigatório: **Sim**

TLSConfig.AllowedCN

Especifique uma lista separada por vírgulas de Common Names de clientes permitidos (somente mTLS).
 Se não especificado, o adaptador de bridge não verifica o Common Name do cliente.

TLSConfig.CAFile

Especifique o caminho completo para um arquivo que contenha o(s) certificado(s) de CA de nível superior usado(s) para verificar certificados de cliente no modo mTLS.

TLSConfig.CertFile

Especifique o caminho completo para um arquivo que contenha o certificado ou a cadeia de certificados.

TLSConfig.KeyFile

Especifique o caminho completo para um arquivo que contenha a chave privada.

TLSConfig.Mode

Defina o modo TLS para conexões de entrada em ListenPort:

- `none` - aceita conexões sem criptografia.
- `tls` - TLS padrão com v1.2 como versão mínima.
- `mtls` - estende o TLS com autenticação mútua.

Padrão: `none`

ZabbixAPITLSCAFile

Especifique o caminho completo de um arquivo que contenha o(s) certificado(s) de CA de nível superior usado(s) pelo adaptador de bridge para verificar o certificado da API do Zabbix.

Obrigatório: **Sim** se ZabbixAPITLSConnect estiver definido como `verify_ca` ou `verify_full`.

ZabbixAPITLSConnect

Defina o modo TLS para conexões de saída com a API do Zabbix:

- `disabled` (ou vazio) - use HTTP simples; ZabbixAPIURL deve usar `http://`.
- `required` - use HTTPS sem verificação do certificado (somente para testes).

- `verify_ca` - use HTTPS e verifique o certificado da API do Zabbix.
- `verify_full` - use HTTPS e verifique o certificado e o nome do host da API do Zabbix.

ZabbixAPIURL

Especifique a URL do endpoint da API do Zabbix.

Obrigatório: **Sim**

Exemplo:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
```

ZabbixBridgeURL

Especifique a URL do Zabbix Bridge para a recuperação da configuração do Zabbix Bridge (recuperada quando o adaptador do bridge é iniciado).

Padrão: `ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration`

MCP server

Visão geral O MCP server do Zabbix é um processo que permite que agentes de IA trabalhem com o Zabbix usando o Model Context Protocol (MCP). O MCP server se comunica com clientes MCP e com a API do Zabbix.

Usando o MCP server, um agente de IA pode:

- Recuperar grupos de hosts, grupos de templates, hosts, templates, itens, regras de LLD, interfaces, macros, triggers, problemas, períodos de manutenção e dados históricos de itens (exceto itens binários);
- Executar atualizações de problemas: adicionar comentários, reconhecer ou cancelar o reconhecimento, alterar a severidade, suprimir ou cancelar a supressão ou fechar problemas;
- Criar, atualizar e excluir períodos de manutenção.

Os tokens da API do Zabbix são usados para autorização; os tokens são enviados com cada solicitação. O acesso é limitado pelas permissões do proprietário do token: somente os dados dos grupos de hosts disponíveis para esse usuário ficam visíveis, e as ações que o usuário não tem permissão para executar são rejeitadas.

O MCP server não mantém estado: nenhuma sessão é mantida entre as solicitações.

> **Note:**
> Embora o MCP server do Zabbix possa funcionar com versões mais antigas ou mais recentes do Zabbix, a compatibilidade é oficialmente suportada somente com a mesma versão do Zabbix do próprio MCP server.

Veja também:

- Parâmetros de configuração: MCP server
- Ferramentas e recursos

Instalação O pacote oficial do zabbix-mcp-server está disponível no repositório do Zabbix.

Para instalar o Zabbix MCP server a partir do código-fonte, compile-o com `make`.

Para configurar o Zabbix MCP server, atualize os parâmetros do arquivo de configuração do MCP server. Pelo menos os parâmetros `FrontendURL` e `AllowedIP` devem ser definidos.

O arquivo de configuração padrão (`zabbix_mcp_server.conf`) está localizado em `/etc/zabbix/`; ele nega as ferramentas que mais frequentemente expõem dados confidenciais. Para habilitar uma ferramenta, remova a linha `DenyTool` correspondente.

Consulte também: Práticas recomendadas de segurança

Execução Para iniciar o servidor MCP, execute:

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

Para verificar o arquivo de configuração e sair sem iniciar o servidor, use a opção `-T`:

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

As seguintes opções de linha de comando são compatíveis:

Opção	Descrição
`-c <file>`	O caminho para o arquivo de configuração. O padrão é `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Valida o arquivo de configuração e sai.
`-h`	Exibe informações de ajuda.
`-V`	Exibe o número da versão.

Quando o servidor MCP é instalado a partir do código-fonte, uma unidade do systemd deve ser criada manualmente, por exemplo:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Conectando um cliente MCP Qualquer cliente MCP que ofereça suporte ao transporte HTTP Streamable pode ser conectado ao servidor MCP. As seguintes configurações são necessárias:

Configuração	Valor
Transporte	HTTP Streamable
URL	`https://<host>:<ListenPort>/mcp`
Cabeçalho	`Authorization: Bearer <Zabbix API token>`, enviado com cada solicitação

Cada usuário deve configurar seu próprio token de API.

Para verificar a configuração, envie uma única solicitação ao servidor MCP:

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

O código de status HTTP 200 e uma lista das ferramentas disponíveis na resposta confirmam que o servidor MCP está acessível e aceita o token; a lista mostra quais ferramentas estão expostas. Outros códigos de status indicam os seguintes problemas:

- 400 - o cabeçalho `Accept` não inclui os dois tipos de mídia exigidos pelo transporte;
- 401 - o cabeçalho `Authorization` com o token está ausente;
- 403 - o endereço do cliente não está listado no parâmetro `AllowedIP`.

Aprovação e acesso às ferramentas Se o cliente MCP for compatível com elicitação de formulários, toda alteração deverá ser aprovada pelo usuário antes de ser enviada ao Zabbix. O cliente MCP mostra ao usuário o que será alterado, e a alteração é aplicada somente se o usuário a confirmar; se o usuário rejeitar ou dispensar a solicitação, nada será alterado no Zabbix. Uma aprovação é válida por cinco minutos e somente para a alteração exata que foi exibida; dentro desse período, a aprovação pode ser reutilizada.

Se o cliente MCP não for compatível com elicitação de formulários (incluindo clientes compatíveis apenas com elicitação por URL), as alterações serão aplicadas imediatamente, sem aprovação. Nesse caso, restrinja as ferramentas de gravação e as permissões da API do Zabbix.

A aprovação é particularmente importante porque a saída das ferramentas pode conter texto não confiável proveniente de sistemas monitorados, que pode influenciar um agent.

Use AllowTool, DenyTool, AllowToolRegexp e DenyToolRegexp para controlar quais ferramentas o servidor MCP expõe. As ferramentas negadas não podem ser chamadas nem listadas, e os recursos associados a elas não podem ser acessados.

Warning:
As regras AllowTool, DenyTool, AllowToolRegexp e DenyToolRegexp limitam o acesso ao MCP, mas não alteram as permissões do token do Zabbix. Se o endereço do frontend for exposto a um agent de IA, o agent poderá tentar enviar chamadas diretas à API. Mantenha o local do frontend do Zabbix e a configuração do servidor MCP inacessíveis ao agent, para que as solicitações precisem passar pelo servidor MCP e por seus controles de segurança.

Práticas recomendadas de segurança Tokens de API

A autenticação e a autorização são gerenciadas inteiramente pelo Zabbix. O cliente MCP envia um token da API do Zabbix com cada solicitação; o MCP server o encaminha sem armazená-lo ou registrá-lo em logs, enquanto o Zabbix valida o token, aplica as permissões e registra as ações no log de auditoria.

Use um token separado, com o mínimo de privilégios e com expiração, para cada agent e usuário. Armazene os tokens em um local seguro e alterne os tokens da API regularmente.

Criptografia TLS

Para evitar que os tokens sejam enviados em texto não criptografado, defina `TLSAccept=cert` e especifique o certificado e a chave (`TLSCertFile` e `TLSKeyFile`) ou execute o MCP server atrás de um proxy reverso confiável que encerre o TLS e defina `TLSAccept=unencrypted`. HTTP não criptografado é aceitável somente em uma rede confiável.

Restrições de acesso

Restrinja o parâmetro `AllowedIP` aos endereços de clientes MCP confiáveis e use as permissões do Zabbix para limitar o que os tokens de API podem fazer.

O MCP server não fornece limitação da taxa de solicitações. Cada solicitação MCP normalmente resulta em uma chamada à API do Zabbix, embora um agent em execução em um loop ainda possa gerar uma carga significativa.

Proteção de dados confidenciais

As chaves de item podem conter credenciais (por exemplo, em itens SSH, JMX, ODBC e HTTP agent), os valores de macro podem conter senhas e strings de comunidade, as interfaces revelam endereços internos e os valores de itens de log ou de texto podem conter quaisquer dados. Por esse motivo, as ferramentas `item_get`, `lld_get`, `interface_get`, `macro_get` e `history_get` são negadas por padrão no arquivo de configuração fornecido.

Para evitar a exposição de valores de macros confidenciais ao provedor de LLM do agent conectado, use macros secretas; seus valores nunca são retornados pela API.

Ferramentas e recursos

Visão geral

O Zabbix MCP server expõe as funcionalidades do Zabbix para clientes MCP como *tools* e *resources*.

Os nomes das tools descrevem a atividade. Se o parâmetro ToolPrefix estiver definido, o prefixo será adicionado a todos os nomes das tools; as regras de acesso às tools correspondem aos nomes simples listados abaixo.

Ferramentas de leitura

Ferramenta	Método da API do Zabbix	Argumentos	Ordenado por	Recurso
`hostgroup_get`	hostgroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://hostgroup/<i`
`templategroup_get`	templategroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://templategrou`
`host_get`	host.get	`hostids`, `groupids`, `search`, `limit`	name	`zabbix://host/<id>`
`template_get`	template.get	`templateids`, `groupids`, `search`, `limit`	name	`zabbix://template/<id`
`item_get`	item.get	`hostids`, `itemids`, `search`, `limit`	name	`zabbix://item/<id>`
`lld_get`	discoveryrule.get	`hostids`, `ruleids`, `limit`	name	-
`interface_get`	hostinterface.get	`hostids`, `interfaceids`, `limit`	ID da interface	-
`macro_get`	usermacro.get	`hostids` ou `global`, `macro_names`, `limit`	macro	-
`problem_get`	problem.get	`eventids`, `groupids`, `hostids`, `severities`, `acknowledged`, `suppressed`, `recent`, `limit`	ID do evento, decrescente	-
`maintenance_get`	maintenance.get	`maintenanceids`, `groupids`, `hostids`, `limit`	name	-
`history_get`	history.get	`itemids`, `time_from`, `time_till`, `type`, `limit`	clock, decrescente	-
`trigger_get`	trigger.get	`triggerids`, `hostids`, `search`, `limit`	description	`zabbix://trigger/<id>`

Observe que:

- `host_get` e `template_get` também retornam os triggers de cada objeto (um conjunto limitado de campos), tags e templates vinculados. Items, regras de descoberta, interfaces e macros são retornados pelas ferramentas correspondentes.
- `history_get` requer um intervalo de tempo não superior a 7 dias e que comece no máximo 7 dias atrás. O argumento `type` deve ser um dos seguintes: `float`, `unsigned`, `character`, `text`, `log` ou `json`, e deve corresponder ao tipo de valor dos items; se o tipo estiver incorreto, será retornado um resultado vazio, que não pode ser distinguido de um item sem dados. O histórico de items binários (tipo de valor 5) não pode ser recuperado por meio do server MCP.
- As ferramentas de leitura solicitam uma lista fixa de campos; todos os campos (`output: extend`) nunca são solicitados.
- Os resultados das ferramentas que têm um recurso contêm o campo `uri`, para que um agent possa prosseguir de uma lista para uma única entidade.

Formato da resposta

Todas as ferramentas de leitura retornam a resposta no mesmo formato:

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Uma resposta é limitada pelo argumento `limit` (padrão: 100, máximo: 1000) e pelo tamanho serializado da resposta (MaxResponseSize, padrão: 64K). A resposta é interrompida quando qualquer um dos limites é atingido. A resposta é truncada somente entre os resultados, nunca dentro de um resultado, e uma resposta truncada indica qual limite foi atingido. Cada ferramenta tem uma ordem de classificação fixa, portanto, uma resposta truncada sempre corresponde ao início do conjunto completo de resultados. A paginação não é compatível; para obter outros resultados, restrinja a consulta.

Ferramentas de problemas

Todas as ferramentas de problemas chamam event.acknowledge com os sinalizadores de ação correspondentes.
O argumento `eventids` é obrigatório; os IDs dos eventos podem ser obtidos com `problem_get`.
Uma mensagem (comentário) pode ser adicionada com qualquer ferramenta de problemas.
O argumento `message` é opcional, exceto para `problem_annotate`, em que é obrigatório; se especificado, a mensagem é registrada na mesma chamada da API do Zabbix.

Tool	Effect	Additional arguments
`problem_acknowledge`	Reconhecer o problema.	-
`problem_unacknowledge`	Remover o reconhecimento.	-
`problem_close`	Fechar o problema manualmente (somente para triggers que permitem o fechamento manual).	-
`problem_annotate`	Registrar uma mensagem.	`message` (obrigatório)
`problem_change_severity`	Alterar a severidade do problema.	`severity` (0-5)
`problem_suppress`	Suprimir o problema até um horário especificado ou indefinidamente.	`suppress_until`
`problem_unsuppress`	Remover a supressão.	-

Não é possível criar ou excluir problemas; todas as ferramentas de problemas atualizam problemas existentes.

Ferramentas de manutenção

Ferramenta	Método da API do Zabbix	Argumentos
`maintenance_create`	maintenance.create	`name`, `groupids` e/ou `hostids`, `active_since`, `active_till`, `timeperiods`, `description`, `tags`, `tags_evaltype`
`maintenance_update`	maintenance.update	`maintenanceid` e qualquer um dos argumentos acima. Os argumentos de lista substituem todo o conjunto de valores; uma lista vazia limpa o conjunto.
`maintenance_delete`	maintenance.delete	`maintenanceids`

Quando um período de manutenção é atualizado, os argumentos de lista (como `hostids`, `groupids` e `timeperiods`) substituem os valores existentes em vez de adicioná-los. Por exemplo, se um período de manutenção abranger os hosts A e B e

`maintenance_update` for chamado com `hostids` contendo apenas o host C, o período de manutenção abrangerá somente o host C.

`maintenance_create` é a única operação que não é idempotente, e `maintenance_delete` é a única operação que não pode ser revertida. Para impedir seu uso, negue essas ferramentas com o parâmetro DenyTool.

Comportamento das ferramentas de gravação

As ferramentas de problemas e manutenção são *ferramentas de gravação*. Elas funcionam da seguinte forma:

- Os argumentos são validados antes de qualquer chamada à API do Zabbix; se a validação falhar, o erro identificará a restrição que não foi atendida.
- Se o cliente oferecer suporte, a alteração será aplicada somente após ser aprovada pelo usuário.
- A alteração é realizada pelo Zabbix usando o token do chamador. O Zabbix verifica as permissões e registra a alteração no log de auditoria associada ao proprietário do token.
- Os erros do Zabbix, incluindo erros de permissão, são retornados inalterados como um objeto de erro com os campos `code`, `message` e `data`.
- As ferramentas de gravação retornam diretamente o resultado do Zabbix; os limites de resposta das ferramentas de leitura não se aplicam.
- As chamadas de gravação nunca são repetidas pelo servidor MCP.

Recursos

Os recursos usam o esquema de URI `zabbix://<kind>/<id>`. Somente `zabbix://about` está incluído na resposta de `resources/list`; os IDs das entidades não podem ser enumerados e são obtidos nos resultados das ferramentas. A resposta de `resources/templates/list` lista os tipos de entidades.

URI	Conteúdo
`zabbix://about`	A URL da API, a URL do frontend, a versão do Zabbix, a identidade à qual o token do chamador corresponde e a lista de ferramentas expostas.
`zabbix://host/<hostid>`	O registro do host sem itens, triggers, interfaces e macros.
`zabbix://hostgroup/<groupid>`	O registro do grupo de hosts sem a lista de membros.
`zabbix://template/<templateid>`	O registro do template sem itens, triggers e regras de descoberta.
`zabbix://templategroup/<groupid>`	O registro do grupo de templates sem a lista de membros.
`zabbix://item/<itemid>`	A definição do item sem o valor atual.
`zabbix://trigger/<triggerid>`	A definição do trigger sem hosts, dependências, funções e tags.

Um recurso cuja ferramenta correspondente tenha acesso negado não existe. Uma entidade inexistente e uma entidade à qual o proprietário do token não tem acesso são relatadas como *recurso não encontrado*; esses casos são deliberadamente indistinguíveis. O recurso `zabbix://about` não é afetado pelas regras de acesso às ferramentas.

Parâmetros de configuração: Zabbix MCP server

Visão geral

O Zabbix MCP server é um processo que permite que agentes de IA trabalhem com o Zabbix usando o Model Context Protocol (MCP).

Os parâmetros compatíveis com o arquivo de configuração do Zabbix MCP server (`zabbix_mcp_server.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 de clientes MCP, separados por vírgulas e opcionalmente na notação CIDR, ou nomes DNS de clientes MCP.
AllowTool	Uma regra curinga que permite uma ferramenta.
AllowToolRegexp	Uma regra de expressão regular que permite ferramentas.
DebugLevel	O nível de depuração.
DenyTool	Uma regra curinga que nega uma ferramenta.
DenyToolRegexp	Uma regra de expressão regular que nega ferramentas.
FrontendURL	O endpoint da API do Zabbix.
IgnoreURLCertErrors	Especifica o tratamento de erros de validação de certificados TLS ao acessar a API do Zabbix.
ListenIP	O endereço IP no qual escutar.
ListenPort	A porta na qual escutar.

Parâmetro	Descrição
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogType	O tipo de saída do log.
MaxResponseSize	O tamanho máximo de uma resposta retornada por um item de leitura.
PidFile	O nome do arquivo PID.
Timeout	O tempo máximo (em segundos) de espera por uma resposta da API do Zabbix.
TLSAccept	Quais conexões de entrada aceitar.
TLSCertFile	O caminho completo de um arquivo que contém o certificado do server ou a cadeia de certificados.
TLSKeyFile	O caminho completo de um arquivo que contém a chave privada do server.
ToolPrefix	Um prefixo adicionado aos nomes de todas as ferramentas.

Todos os parâmetros são opcionais, a menos que seja explicitamente indicado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores dos arquivos de configuração fornecidos;
- Os valores são compatíveis com variáveis de ambiente no formato `${NAME}`;
- O Zabbix oferece suporte apenas a arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários iniciados com # são compatíveis apenas no início da linha.

Detalhes dos parâmetros

AllowedIP

Uma lista de endereços IP delimitados por vírgulas, opcionalmente na notação CIDR, ou nomes DNS de clientes MCP. As conexões de entrada serão aceitas somente dos hosts listados aqui.

Exemplo:

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

Obrigatório: sim

AllowTool

Uma regra curinga que permite uma ferramenta. O parâmetro pode ser especificado várias vezes.
 Os padrões curinga usam a sintaxe de chave de item, na qual * corresponde a quaisquer caracteres. A regra corresponde ao nome simples da ferramenta e ignora ToolPrefix.
 Todas as regras `AllowTool`, `DenyTool`, `AllowToolRegexp` e `DenyToolRegexp` formam uma única lista na ordem em que são especificadas; a primeira regra correspondente determina o resultado. Uma ferramenta que não corresponde a nenhuma regra é permitida.

Exemplo:

```
AllowTool=host_get
```

AllowToolRegexp

Uma regra de expressão regular que permite ferramentas. O parâmetro pode ser especificado várias vezes.
 As expressões regulares usam a sintaxe RE2 e correspondem a qualquer parte do nome simples da ferramenta, a menos que estejam ancoradas.

Exemplo:

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

Especifique o nível de depuração: - *0* - informações básicas sobre a inicialização e a interrupção 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

DenyTool

Uma regra curinga que nega uma ferramenta. O parâmetro pode ser especificado várias vezes.
 Uma ferramenta negada não é incluída na lista de ferramentas e não pode ser chamada. Consulte AllowTool para obter detalhes.

Exemplo:

```
DenyTool=maintenance_delete
```

DenyToolRegexp

Uma regra de expressão regular que nega ferramentas. O parâmetro pode ser especificado várias vezes.
 Consulte AllowTool-Regexp para obter detalhes.

Exemplo:

`DenyToolRegexp=^problem_(close|suppress)$`

FrontendURL

O endpoint da API do Zabbix: a URL do frontend do Zabbix terminando com `api_jsonrpc.php`.

Exemplo:

`FrontendURL=https://zabbix.example.com/api_jsonrpc.php`

Obrigatório: sim

IgnoreURLCertErrors

Especifica o tratamento de erros de validação de certificados TLS ao acessar a API do Zabbix:
 false - não ignorar erros de certificado;
 true - ignorar erros de certificado (usar somente em um ambiente de desenvolvimento).

Padrão: `false`

ListenIP

O endereço IP no qual escutar.

Padrão: `0.0.0.0`

ListenPort

A porta na qual escutar. O caminho do endpoint MCP é `/mcp`.

Padrão: 8443
 Intervalo: 1024-32767

LogFile

O nome do arquivo de log.

Padrão: `/tmp/zabbix_mcp_server.log`

Obrigatório: Não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desabilita a rotação automática de logs.

Padrão: 1
 Intervalo: 0-1024

LogType

O tipo da 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`

MaxResponseSize

O tamanho máximo de uma resposta retornada por uma ferramenta de leitura. O limite que for atingido primeiro, entre este limite e o argumento `limit` da ferramenta, interrompe a resposta.
 Suporta sufixos de unidade.

Padrão: 64K
 Intervalo: 16K-256K

PidFile

O nome do arquivo PID.

Padrão: `/tmp/zabbix_mcp_server.pid`

Tempo limite

O tempo máximo (em segundos) de espera por uma resposta da API do Zabbix.

Padrão: 10
 Intervalo: 3-30

TLSAccept

Quais conexões de entrada aceitar: - *unencrypted* - aceitar conexões sem criptografia (padrão); - *cert* - aceitar conexões com TLS e um certificado.

Padrão: `unencrypted`

TLSCertFile

O caminho completo do arquivo que contém o certificado do server ou a cadeia de certificados.

Obrigatório: Sim, se `TLSAccept` estiver definido como `cert`; caso contrário, não

TLSKeyFile

O caminho completo do arquivo que contém a chave privada do server.

Obrigatório: Sim, se `TLSAccept` estiver definido como `cert`; caso contrário, não

ToolPrefix

Um prefixo adicionado aos nomes de todas as ferramentas, por exemplo, para evitar conflitos de nomes com ferramentas de outros servidores MCP. Por exemplo, com `ToolPrefix=zbx_`, a ferramenta `host_get` é exposta como `zbx_host_get`.
 Caracteres permitidos: letras, dígitos, sublinhado (`_`), hífen (`-`) e ponto (`.`).
 As regras de acesso às ferramentas correspondem aos nomes das ferramentas sem o prefixo.

Exemplo:

`ToolPrefix=zbx_`

Variáveis de ambiente

Visão geral As variáveis de ambiente permitem configurar os componentes do Zabbix sem codificar valores diretamente 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 cenários.

No caso mais simples, ao definir o valor do parâmetro de configuração DebugLevel do Zabbix server como uma variável de ambiente, você pode usá-lo 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` comandos em tempo de execução 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.
- 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 os 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, por meio do arquivo `/proc/<PID>/environ`).

Sintaxe As variáveis de ambiente devem usar a seguinte sintaxe: `${alphanumerics/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 correspondem à sintaxe exigida ou que são combinadas com um valor regular serão tratadas como valores regulares, o que pode gerar erros.

Sintaxe correta da variável:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Sintaxe incorreta da 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 componentes do Zabbix.

Exemplo 1: Configurando e testando o Zabbix agent

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 Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agen
```

3. Inicie o agent com as variáveis de ambiente:

```
ZBX_HOSTNAME="Novo Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agen
```

Ao usar comandos de runtime (por exemplo, para aumentar o nível de log do agent), todas as variáveis de ambiente usadas anteriormente devem ser especificadas:

```
ZBX_HOSTNAME="Novo Zabbix agent" 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 de runtime; se as variáveis de ambiente forem omitidas, serão usados os valores padrão dos parâmetros de configuração.

Como alternativa, depois de 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 definidas incorretamente.

```
export ZBX_HOSTNAME="Novo Zabbix agent"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

Exemplo 2: Configurando o Zabbix agent para um container

Se você estiver criando e configurando sua própria imagem personalizada para componentes do Zabbix (por exemplo, o Zabbix agent), 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. Depois de 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 de runtime (por exemplo, para aumentar o nível de log do agent), acesse o shell do container e execute o comando de runtime:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

> **Attention:**
> O comando de runtime `userparameter_reload` não oferece suporte ao recarregamento de variáveis de ambiente. Durante o recarregamento, as variáveis são ignoradas e somente os parâmetros com valores regulares são recarregados.

Configuração

Visão geral A configuração no Zabbix envolve especificar quais hosts e sistemas monitorar, definir quais dados coletar e estabelecer como as notificações são entregues quando ocorrem problemas.

Use a barra lateral para navegar pelos aspectos relevantes da configuração.

Principais tarefas de configuração

- Configurar um novo host ou grupo de hosts → Hosts and host groups
- Definir quais dados coletar dos hosts → Items
- Criar regras para detectar problemas → Problem detection with triggers
- Configurar como as notificações são entregues → Notifications upon events
- Exibir dados de monitoramento por meio de dashboards, gráficos e mapas → Visualization
- Reutilizar configurações em vários hosts → Templates and template groups, Templates out of the box
- Controlar o acesso e as permissões dos usuários → Users and user groups

Melhores práticas de configuração As escolhas de configuração afetam o desempenho e a manutenibilidade. Embora cada instalação do Zabbix seja diferente, estas melhores práticas podem ser usadas como base para ajustes conforme necessário.

Host separado para cada instância de monitoramento

Um host no Zabbix é uma entidade lógica, e não uma máquina ou dispositivo físico. Crie um host separado para cada instância independente, como uma máquina virtual, banco de dados, contêiner ou switch de rede.

Essa abordagem oferece:

1. Dados de monitoramento mais organizados, com items, triggers e notificações de alerta separados para cada host.
2. Controle de acesso de usuários mais granular por meio de funções de usuário alinhado ao princípio do menor privilégio.

Use templates and mass update for similar hosts

Para hosts semelhantes, como *Network switch 1* e *Network switch 2*, você pode criar um template com as métricas necessárias (*Network switch template*) e agrupar os hosts em um grupo de hosts (*Network switches*).

Ao adicionar novos switches, você pode usar a atualização em massa em *Data Collection* > *Hosts* para vincular o template em vez de clonar hosts manualmente.

Configure itens master e dependentes para minimizar requisições

Para minimizar requisições à entidade de destino, configure um item master para coletar dados em uma única requisição e, em seguida, use itens dependentes com pré-processamento para extrair valores específicos.

Por exemplo, o item master pode coletar uma resposta JSON ou XML com várias métricas ou executar uma consulta ao banco de dados retornando várias colunas (por exemplo, número de conexões abertas, conexões abortadas, número máximo permitido de conexões simultâneas e total acumulado de conexões desde a inicialização), e os itens dependentes irão analisar e armazenar cada valor separadamente.

Descarte o histórico do item master após a coleta e mantenha apenas os dados dos itens dependentes.

Implante um proxy do Zabbix quando necessário

Se todos os seus hosts estiverem na mesma rede local que o server do Zabbix, sem preocupações relacionadas à escalabilidade ou ao desempenho, talvez você não precise de um proxy. Em ambientes maiores ou mais complexos, um proxy permite uma distribuição de carga mais equilibrada.

Adicione um proxy do Zabbix quando:

1. Monitorar vários hosts usando diferentes métodos de coleta de métricas atrás de um firewall. O proxy coleta dados dos hosts e os encaminha para o server do Zabbix, reduzindo o número de portas abertas no firewall.
2. Monitorar locais remotos, filiais ou redes. Os proxies do Zabbix em locais remotos continuam coletando dados durante interrupções de rede e enviam os dados quando a conexão é restabelecida.
3. Gerenciar implantações em grande escala. Se você tiver muitos hosts e/ou coletar um grande número de valores por segundo, implante um proxy para reduzir a carga no server e melhorar o desempenho.

Configurar a recuperação independente de macro secreta

Você pode querer usar macros de usuário secret como texto secreto ou como macros de cofre secreto.

Por padrão, os valores de macros secretas são recuperados pelo Zabbix server e propagados para os proxies. Para maior segurança com macros de cofre secreto, configure os valores das macros para serem recuperados independentemente pelo Zabbix server e pelos proxies.

Configurando um template

Visão geral

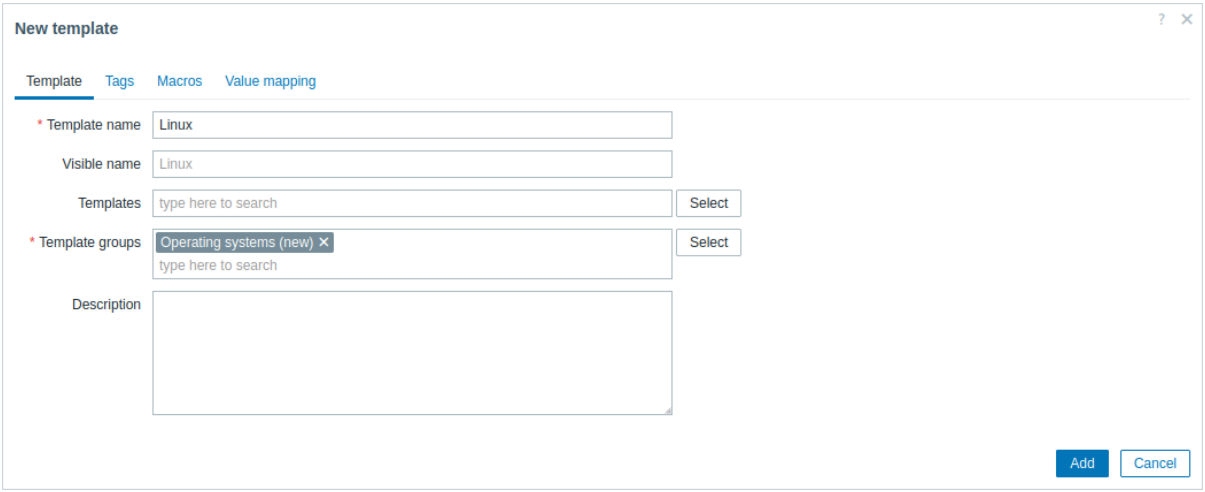
A configuração de um template requer que você primeiro crie um template definindo seus parâmetros gerais e, em seguida, adicione entidades (itens, triggers, gráficos, etc.) a ele.

Criando um template

Para criar um template, faça o seguinte:

1. Acesse *Coleta de dados* > *Templates*.
2. Clique em *Criar template*.
3. Edite os atributos do template.

A aba **Template** contém os atributos gerais do template.



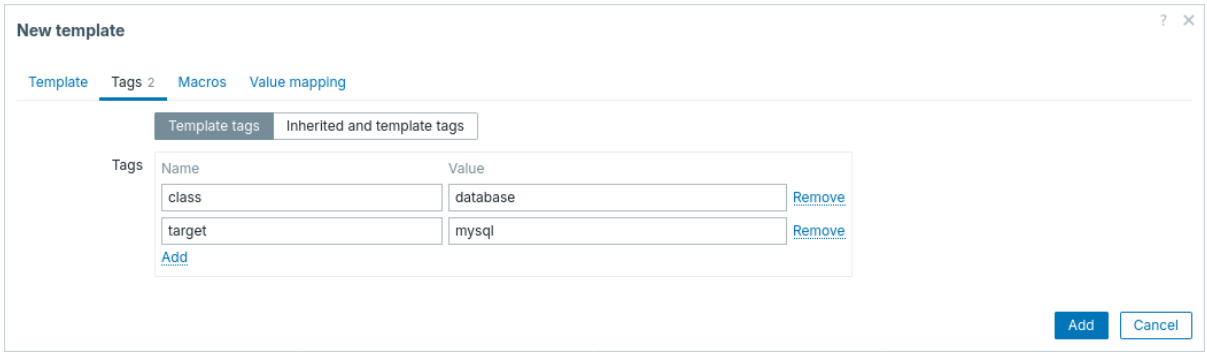
Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Atributos do template:

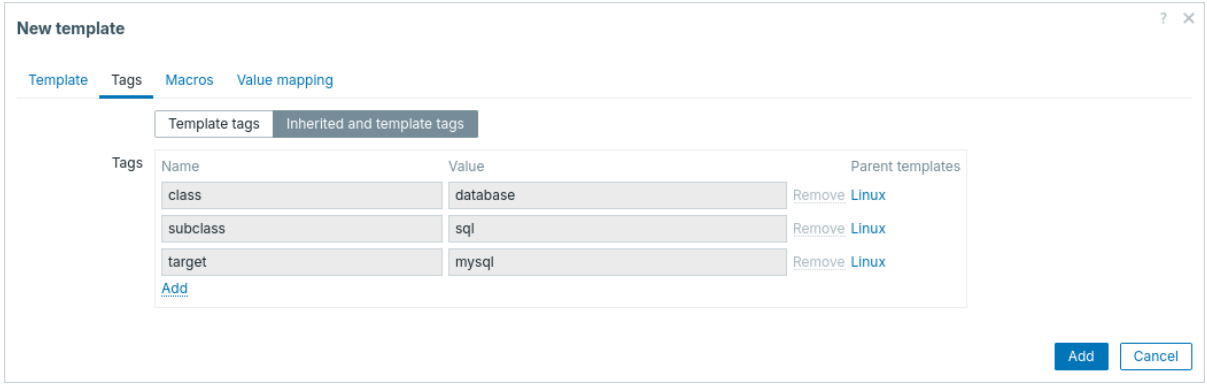
Parâmetro	Descrição
Nome do template	Nome exclusivo do template. São permitidos caracteres alfanuméricos, espaços, pontos, hífens e sublinhados. Espaços no início e no fim não são permitidos.
Nome visível	Se você definir este nome, ele será o nome exibido em listas, mapas etc.

Parâmetro	Descrição
Templates	Vincule um ou mais templates a este template. Todas as entidades (itens, triggers etc.) serão herdadas dos templates vinculados. Para vincular um novo template, digite o nome dele no campo *Templates* — isso exibirá uma lista suspensa com os templates correspondentes. Como alternativa, clique em *Selecionar* para abrir a janela pop-up *Templates*, que inicialmente não exibe nenhum template. Para preencher a lista, digite o nome de um grupo de templates no campo *Grupo de templates* (e selecione um na lista sugerida) ou clique no botão *Selecionar* ao lado do campo *Grupo de templates* para abrir a janela pop-up *Grupos de templates*. Depois que você escolher um grupo de templates, a janela pop-up *Templates* será atualizada para exibir 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 (itens, triggers etc.); *Desvincular e limpar* - desvincula o template e remove todas as suas entidades (itens, 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; exibidos somente ao atualizar templates existentes (templates prontos fornecidos pelo Zabbix, templates importados ou templates modificados por meio da API de Template), se a configuração do template contiver essas informações. Não pode ser modificado no frontend do Zabbix. Para templates prontos, a versão é exibida da seguinte forma: versão principal do Zabbix, delimitador (”-”), número da revisão (incrementado a cada nova versão do template e redefinido 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 receberão as tags com os valores inseridos aqui.



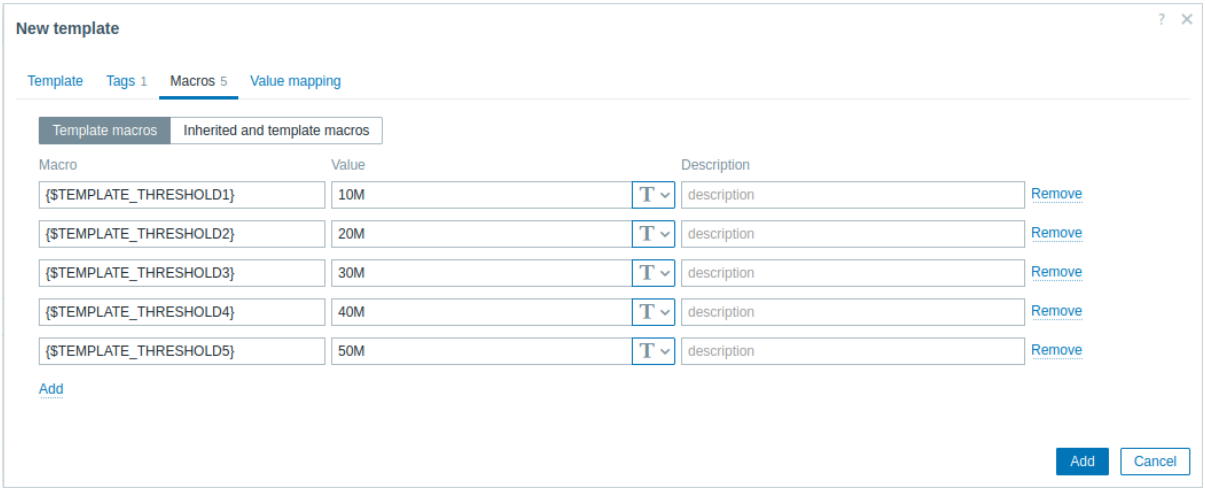
Se você selecionar a opção *Tags herdadas e do template*, também poderá visualizar as tags dos templates vinculados que serão herdadas por este template.



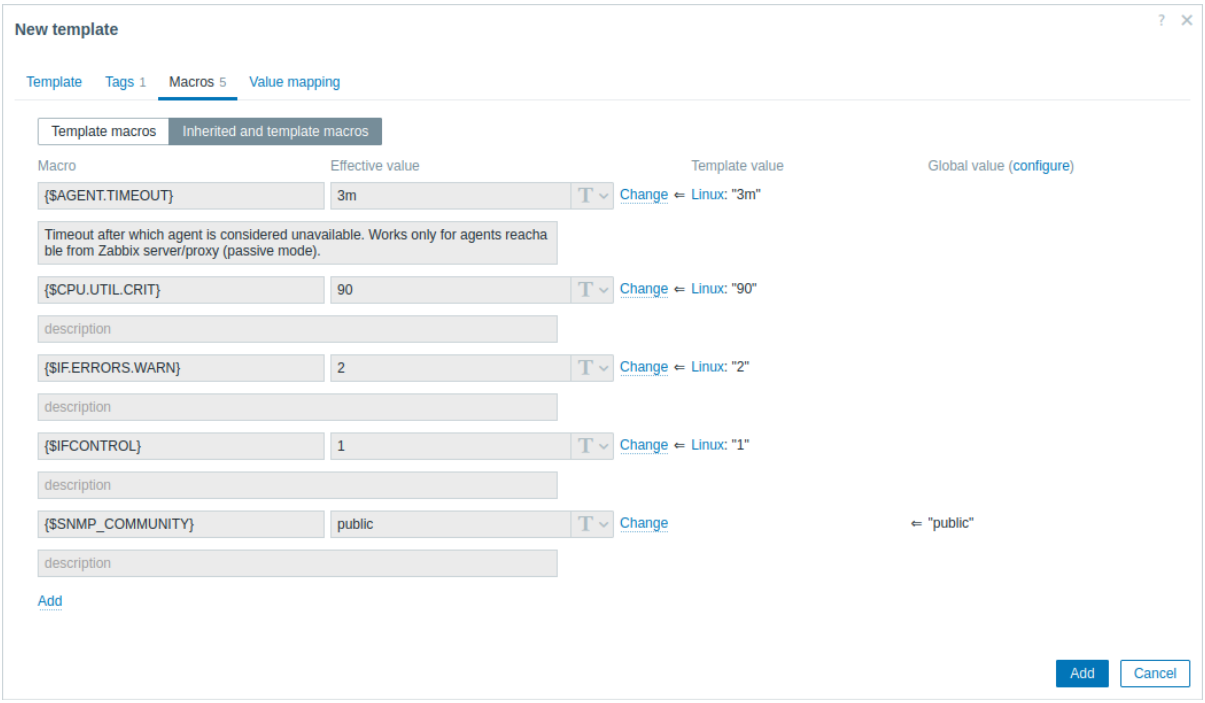
Para sua conveniência, são fornecidos links para os respectivos templates.

Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são compatíveis com 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 as macros dos templates vinculados e as macros globais que serão herdadas por este template, bem como os valores para os quais as macros serão resolvidas.



Para sua conveniência, são fornecidos links para os respectivos templates, bem como um link para a configuração de macros globais. Também é possível editar uma macro de um template vinculado ou uma 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 dos itens 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 (itens, 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 possa ser distinguido do original.
Delete	Exclui o template; as entidades do template (itens, triggers etc.) permanecem nos 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, graphs

Attention:
Os items precisam ser adicionados a um template primeiro. Triggers e graphs não podem ser adicionados sem o item correspondente.

Você pode adicionar items existentes a um template ou criar novos items.

Para adicionar items ao template:

1. Vá para *Data collection* > *Hosts* (ou *Templates*).
2. Clique em *Items* na linha do host/template desejado.
3. Marque as caixas de seleção dos items que você deseja adicionar ao template.
4. Clique em *Copy* abaixo da lista de items.
5. Selecione o template (ou grupo de templates) para o qual os items devem ser copiados e clique em *Copy*.
 Todos os items selecionados devem ser copiados para o template.

A adição de triggers e graphs é feita de forma semelhante (a partir da lista de triggers e graphs, 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 *Data collection* > *Templates*, faça o seguinte:

1. Clique em *Dashboards* na linha do template.
2. Configure um dashboard seguindo as orientações 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 têm um rótulo diferente. Isso ocorre porque os dashboards de template exibem dados apenas do host ao qual o template está vinculado. Por exemplo, os parâmetros *Host groups*, *Exclude host groups* e *Hosts* no widget *Problems* não estão disponíveis, o parâmetro *Host groups* no widget *Host availability* não está disponível, e o parâmetro *Show hosts in maintenance* é renomeado para *Show data in maintenance*, etc. Para mais informações sobre a disponibilidade de parâmetros nos 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 *Data collection* > *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.

Configurando um grupo de template

Visão geral

Os grupos de templates são usados para o agrupamento lógico de templates e para a atribuição de permissões de usuário a eles.

Cada template deve ter pelo menos um grupo de templates 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 host 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 e atribuindo a esse grupo de usuários permissões para o grupo de templates que contém esse template.

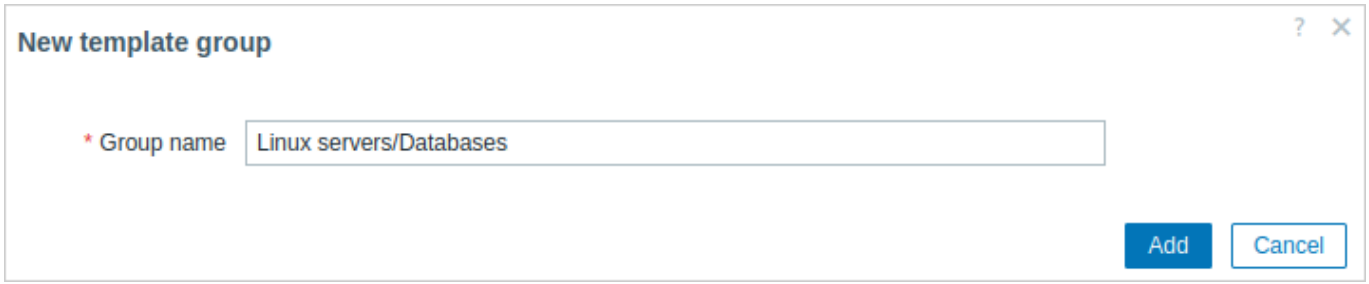
Configuração

Attention:
Somente usuários Super admin podem criar grupos de template.

Há duas opções para criar um grupo de template no frontend do Zabbix.

Opção um:

1. Vá para *Data collection* > *Template groups*.
2. Clique em *Create template group* no canto superior direito da tela.
3. Informe o nome do grupo no formulário.



Opção dois: ao configurar um template, informe um nome de grupo inexistente no campo *Template groups*.

Depois que o grupo de template for criado, você pode clicar no nome do template na lista em *Data collection* > *Template groups* para editar o nome do grupo, clonar o grupo ou excluir o grupo.

Excluir um grupo de template remove apenas o grupo lógico, não os templates do grupo. Não é possível excluir um grupo de template que seja o único grupo de qualquer template existente.

Criando subgrupos de template

Um subgrupo de template (ou grupo de template aninhado) é um filho do grupo de template pai que o contém.

Um subgrupo é criado usando a barra '/' no campo de entrada do nome do grupo para denotar sua relação com o(s) grupo(s) pai(s). Por exemplo:

- inserir `Servidores Linux/Bancos de Dados` cria o subgrupo `Servidores Linux/Bancos de Dados` do grupo pai `Servidores Linux`.
- inserir `Servidores Linux/Bancos de Dados/MySQL/Tokyo` cria o respectivo subgrupo dentro dos grupos pais aninhados `Servidores Linux`, `Servidores Linux/Bancos de Dados`, `Servidores Linux/Bancos de Dados/MySQL`.

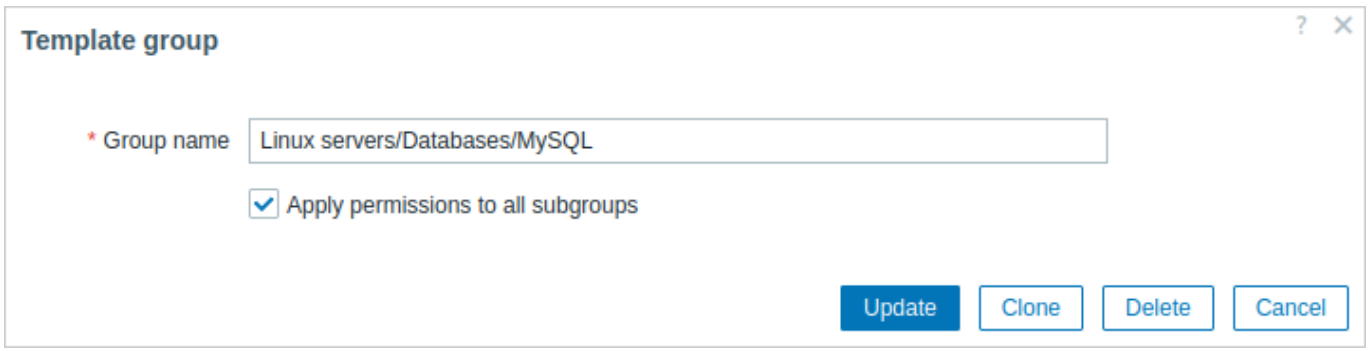
Ao criar um subgrupo, não é permitido usar barras no início ou no final, nem várias barras seguidas. Não há suporte para escape de '/'.

Não é necessário criar nenhum grupo de template pai antes de criar um subgrupo. Você pode escolher se deseja começar criando um subgrupo (por exemplo, `Servidores Linux/Bancos de Dados`) ou qualquer grupo de template pai (em nosso exemplo, `Servidores Linux`). Se você começar criando um subgrupo, o(s) grupo(s) de template pai **não** será(ão) criado(s) automaticamente.

Permissões para grupos de template

- Ao criar um subgrupo em um grupo de template pai existente (por exemplo, ao criar `Linux servers/Databases` quando `Linux servers` já existe), as permissões do grupo de usuários para o subgrupo são herdadas do pai.
- Ao criar um grupo de template pai para um subgrupo existente (por exemplo, ao criar `Linux servers` quando `Linux servers/Databases` já existe), nenhuma permissão para o pai é definida.

Ao editar qualquer grupo de template, você também pode definir uma opção adicional, *Aplicar permissões a todos os subgrupos*.



Marcar esta caixa de seleção e clicar em *Update* aplicará o mesmo nível de permissões a todos os subgrupos atuais e futuros do grupo de template que está sendo editado.

Assim, se algum grupo de usuários tiver recebido permissões diferentes 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 do grupo que está sendo editado.

Observe que essa opção não é salva no banco de dados e substituirá as permissões existentes. Quaisquer alterações feitas por meio dessa opção só podem ser revertidas manualmente.

Vincular/desvincular

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. Acesse *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 será exibida; role para baixo para selecionar. Como alternativa, clique em *Selecionar* ao lado do campo e selecione um ou vários templates na lista exibida em uma janela pop-up.
4. Clique em *Adicionar/Atualizar* no formulário de atributos do host.

Agora, o host terá todas as entidades do template. Isso inclui itens, triggers, gráficos, regras de descoberta de baixo nível, cenários web e dashboards.

Attention:
A vinculação de vários templates ao mesmo host falhará se esses templates contiverem itens com a mesma chave de item. Além disso, como triggers e gráficos usam itens, eles também não poderão ser vinculados a um único host a partir de vários templates se forem usadas chaves de item idênticas.

Quando entidades (itens, triggers etc.) são adicionadas a partir do template:

- entidades idênticas que já existiam no host são atualizadas como entidades do template, e **todas as 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 somente no host permanecem inalteradas.

Nas listas, todas as entidades do template passam a ter como prefixo o nome do template, indicando que pertencem a esse template específico. O próprio nome do template (exibido em cinza) é um link que permite acessar a lista dessas entidades no nível do template.

Note:
Para alguns itens, como verificações externas, verificações do 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 itens serão adicionados sem uma interface. Se você adicionar uma interface de host posteriormente, ela não será atribuída automaticamente aos itens já existentes. Para atribuir a interface de host recém-adicionada a todos os itens do template de uma só vez, desvincule o template do host e vincule-o novamente. Para preservar o histórico dos itens, use a opção *Desvincular*; não use *Desvincular e limpar*.

Se alguma entidade não tiver o nome do template como prefixo, isso significa que ela já existia no host antes e não foi adicionada pelo template.

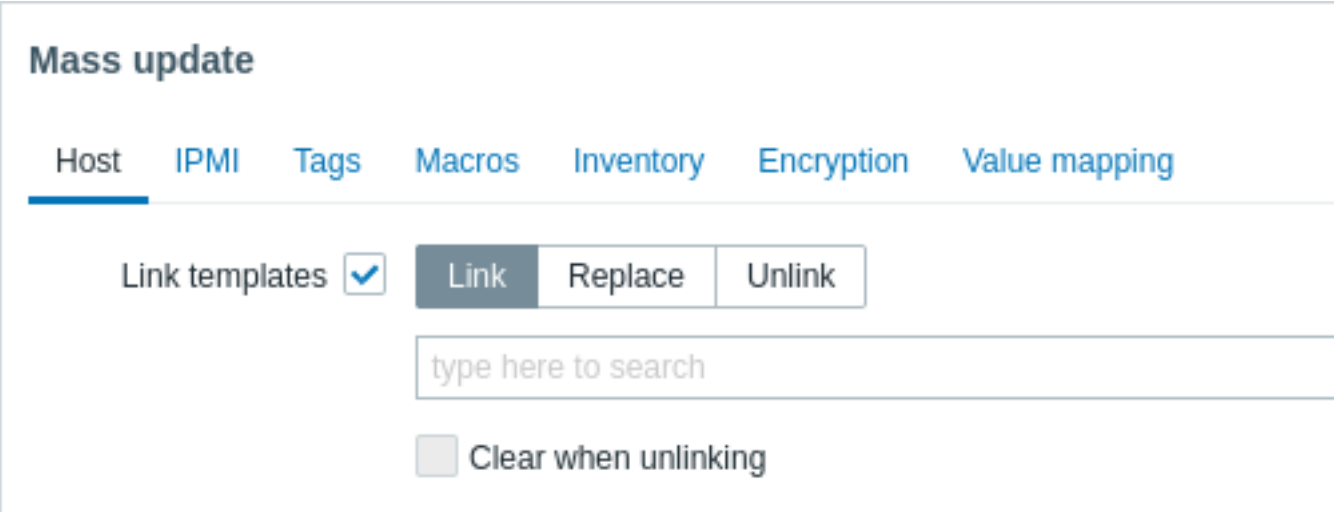
Critérios de unicidade de entidades

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 a vinculação de templates de vários hosts, em *Coleta de dados* > *Hosts*, selecione alguns hosts marcando suas caixas de seleção. Em seguida, clique em **Atualização em massa** abaixo da lista e selecione *Vincular templates*:



Para vincular templates adicionais, comece a digitar o nome do template no campo de preenchimento automático até que uma lista suspensa apareça, oferecendo os templates correspondentes. Basta rolar para baixo para selecionar o template a ser vinculado.

A opção *Substituir* permite vincular um novo template e, ao mesmo tempo, desvincular qualquer template que tenha sido vinculado anteriormente aos hosts. A opção *Desvincular* permite especificar quais templates devem ser desvinculados. A opção *Limpar ao desvincular* permite não apenas desvincular templates vinculados anteriormente, mas também remover todas as entidades herdadas deles (itens, triggers etc.).

Note:
O Zabbix oferece um conjunto considerável de templates predefinidos. Você pode usá-los como referência, mas evite usá-los sem alterações em produção, pois eles podem conter itens demais e coletar dados com muita frequência. Se 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. Acesse *Data collection* > *Hosts*.
2. Clique no host desejado e localize o campo *Templates*.
3. Clique em *Unlink* ou *Unlink and clear* ao lado do template para desvinculá-lo.
4. Clique em *Update* no formulário de atributos do host.

A opção *Unlink* removerá apenas a associação com o template, mantendo todas as suas entidades no host. Isso inclui items, triggers, gráficos, regras de descoberta em baixo nível e cenários web, mas exclui dashboards. Observe que value maps e tags herdadas de templates vinculados também serão removidos.

A opção *Unlink and clear* removerá tanto a associação com o template quanto todas as suas entidades (items, triggers etc.).

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. Localize o campo *Templates*.
3. Clique em *Select* para abrir a janela pop-up *Templates*.
4. Na janela pop-up, escolha os templates necessários e, em seguida, clique em *Select* para adicionar os templates à lista.
5. Clique em *Add* ou *Update* 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, graphs, regras de descoberta de baixo nível e cenários web, mas exclui dashboards. No entanto, os dashboards dos templates vinculados serão herdados pelos hosts.

Para desvincular qualquer um dos templates vinculados, clique em *Unlink* ou *Unlink and clear* no formulário de configuração do template e, em seguida, clique em *Update*.

A opção *Unlink* simplesmente removerá a associação com o template vinculado, sem remover nenhuma de suas entidades (items, triggers etc.).

A opção *Unlink and clear* removerá tanto a associação com o template vinculado quanto todas as suas entidades (items, triggers etc.).

Atualização em massa

Visão geral

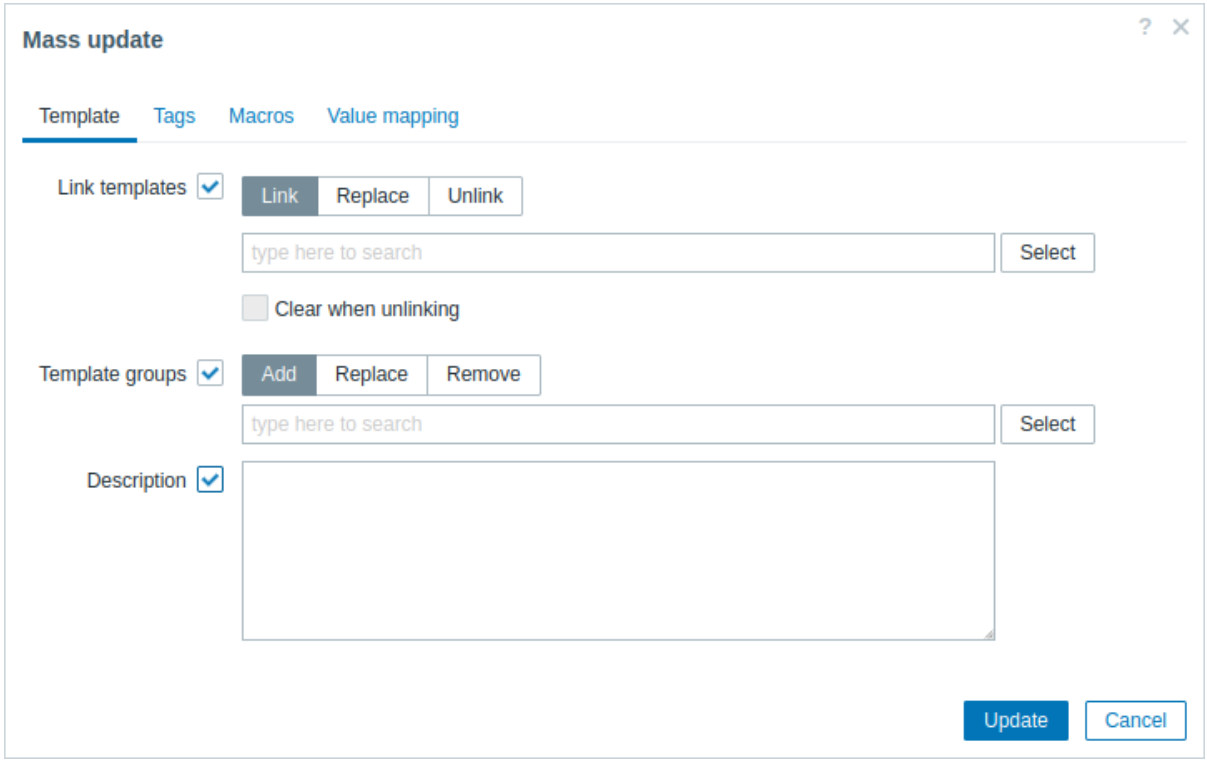
Às vezes, você pode querer alterar algum atributo para vários templates de uma só vez. Em vez de abrir cada template individualmente para edição, você pode usar a função de atualização em massa para isso.

Usando atualização em massa

Para atualizar em massa alguns templates, faça o seguinte:

1. Marque as caixas de seleção antes dos templates que você deseja atualizar na lista de templates.
2. Clique em *Atualização em massa* abaixo da lista.
3. Navegue até a aba com os atributos necessários (*Template*, *Tags*, *Macros* ou *Mapeamento de valores*).
4. Marque as caixas de seleção de qualquer atributo que deseja atualizar e insira um novo valor para eles.

A aba **Template** contém opções gerais de atualização em massa de templates.



As opções a seguir estão disponíveis ao selecionar o botão correspondente para a atualização de *Vincular templates*:

- *Vincular* - especifique quais templates adicionais devem ser vinculados;
- *Substituir* - especifique quais templates devem ser vinculados e, ao mesmo tempo, desvincule todos os templates vinculados anteriormente;
- *Desvincular* - especifique quais templates devem ser desvinculados.

Para especificar os templates a serem vinculados/desvinculados, comece a digitar o nome do template no campo de preenchimento automático até que uma lista suspensa seja exibida com os templates correspondentes. Basta rolar para baixo para selecionar os templates necessários.

A opção *Limpar ao desvincular* permite não apenas desvincular os templates vinculados anteriormente, mas também remover todas as entidades herdadas deles (itens, triggers, gráficos etc.).

As opções a seguir estão disponíveis ao selecionar o botão correspondente para a atualização de *Grupos de templates*:

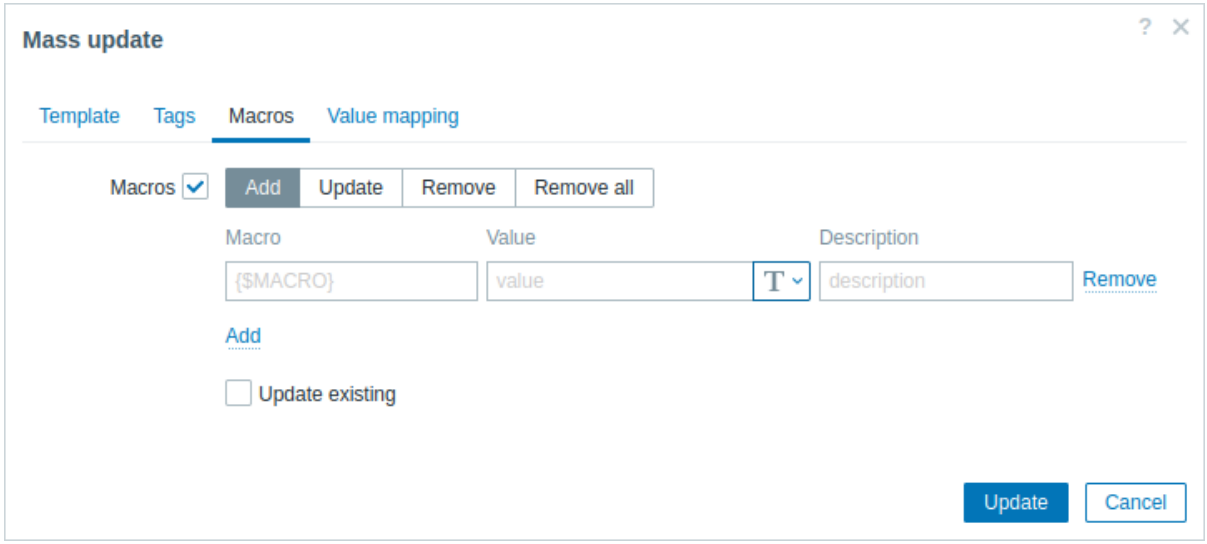
- *Adicionar* - especifique grupos de templates adicionais entre os existentes ou insira grupos de templates completamente novos para os templates;
- *Substituir* - remova o template de todos os grupos de templates existentes e substitua-os pelo(s) grupo(s) especificado(s) neste campo (grupos de templates existentes ou novos);
- *Remover* - remova grupos de templates específicos dos templates.

Esses campos usam preenchimento automático: ao começar a digitar neles, uma lista suspensa com os grupos de templates correspondentes será exibida. Se o grupo de templates for novo, ele também aparecerá na lista suspensa 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.

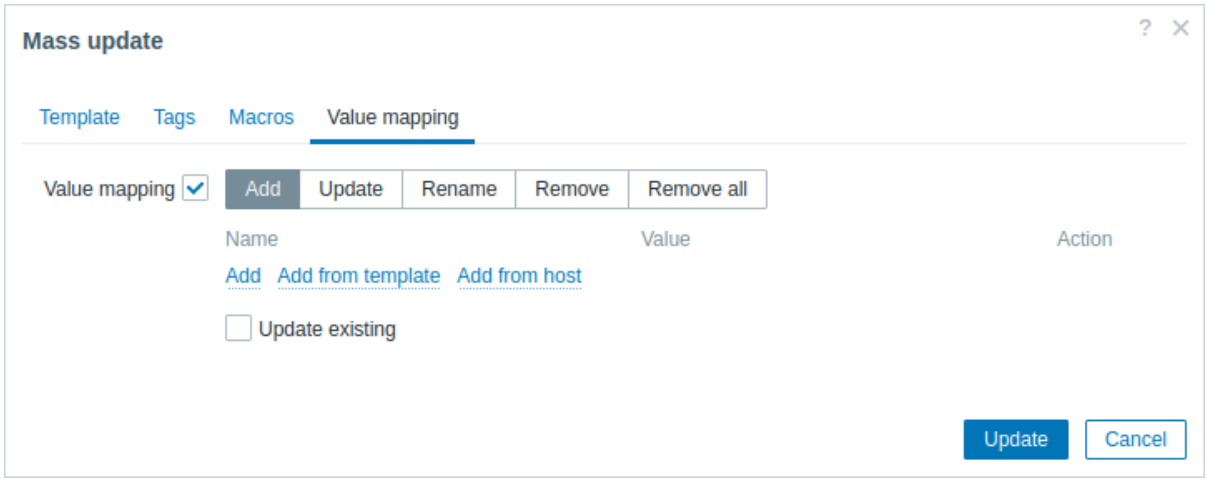
A aba **Macros** permite atualizar em massa as macros no nível do template.



As seguintes opções estão disponíveis ao selecionar o botão correspondente para a atualização das macros:

- *Adicionar* - especifica macros de usuário adicionais para os templates. Se a caixa de seleção *Atualizar existentes* estiver marcada, o valor, o tipo e a descrição do nome da macro especificada serão atualizados. Se não estiver marcada, caso já exista uma macro com esse nome no(s) template(s), ela não será atualizada.
- *Atualizar* - substitui os valores, tipos e descrições das macros especificadas nesta lista. Se a caixa de seleção *Adicionar ausentes* estiver marcada, a macro que ainda não existia em um template será adicionada como uma nova macro. Se não estiver marcada, somente as macros que já existem em um template serão atualizadas.
- *Remover* - remove as macros especificadas dos templates. Se a caixa de seleção *Exceto as selecionadas* estiver marcada, todas as macros, exceto as especificadas na lista, serão removidas. Se não estiver marcada, somente as macros especificadas na lista serão removidas.
- *Remover todas* - remove todas as macros de usuário dos templates. Se a caixa de seleção *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.



Estão disponíveis botões com as seguintes opções para atualizar o mapa 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, somente 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 os selecionados*, todos os mapas de valores serão removidos **exceto** os especificados.
- *Remover todos* - remove todos os mapas de valores dos templates. Se a caixa de seleção *Confirmo a remoção de todos os mapas de valores* não estiver marcada, uma nova janela pop-up será aberta solicitando a confirmação da remoção.

As opções *Adicionar do template* e *Adicionar do host* estão disponíveis para operações de adição/atualização de mapeamento de

valores. Elas permitem selecionar mapeamentos de valores de um template ou de um host, respectivamente.

Quando terminar todas as alterações necessárias, clique em *Update*. Os atributos serão atualizados de acordo para todos os templates selecionados.

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.

Assistente de host

Visão geral

O Host Wizard 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:

- Selecionar um template
- Criar ou selecionar um host
- Instalar o Zabbix agent ou agent 2
- Adicionar uma interface do host
- Aplicar configuração adicional ao seu alvo de monitoramento ou ao host do Zabbix (se exigido pelo template)

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

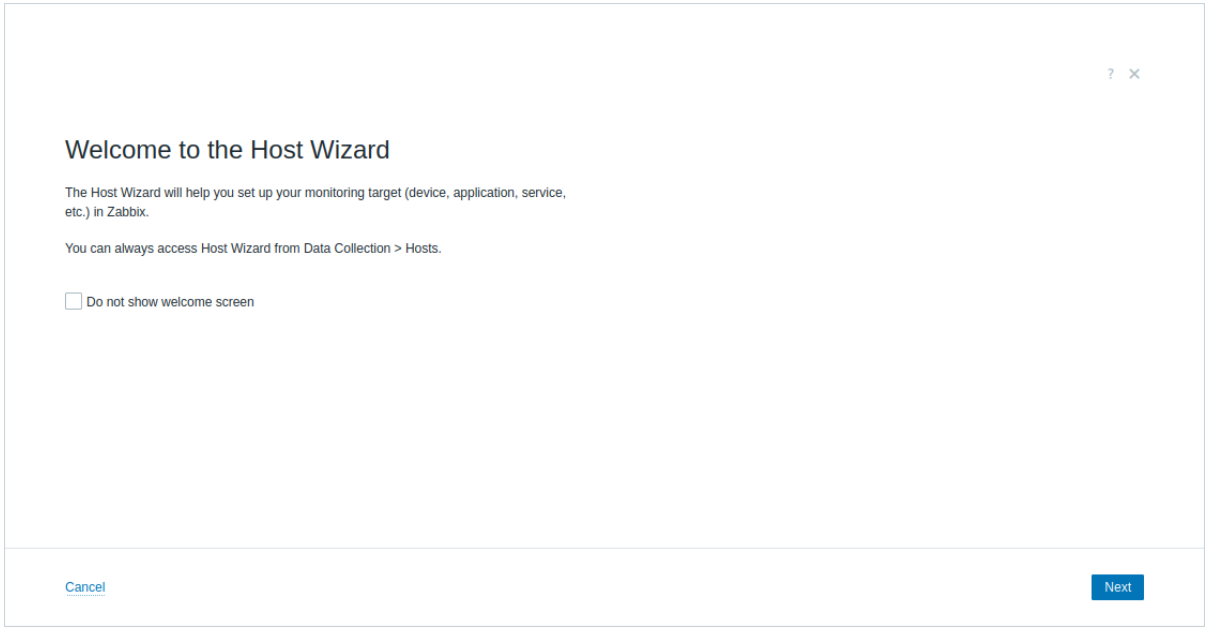
1. Acesse: *Data collection* > *Hosts*.
2. Clique em *Host Wizard* no canto superior direito da tela.
3. Siga as instruções no assistente.

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

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

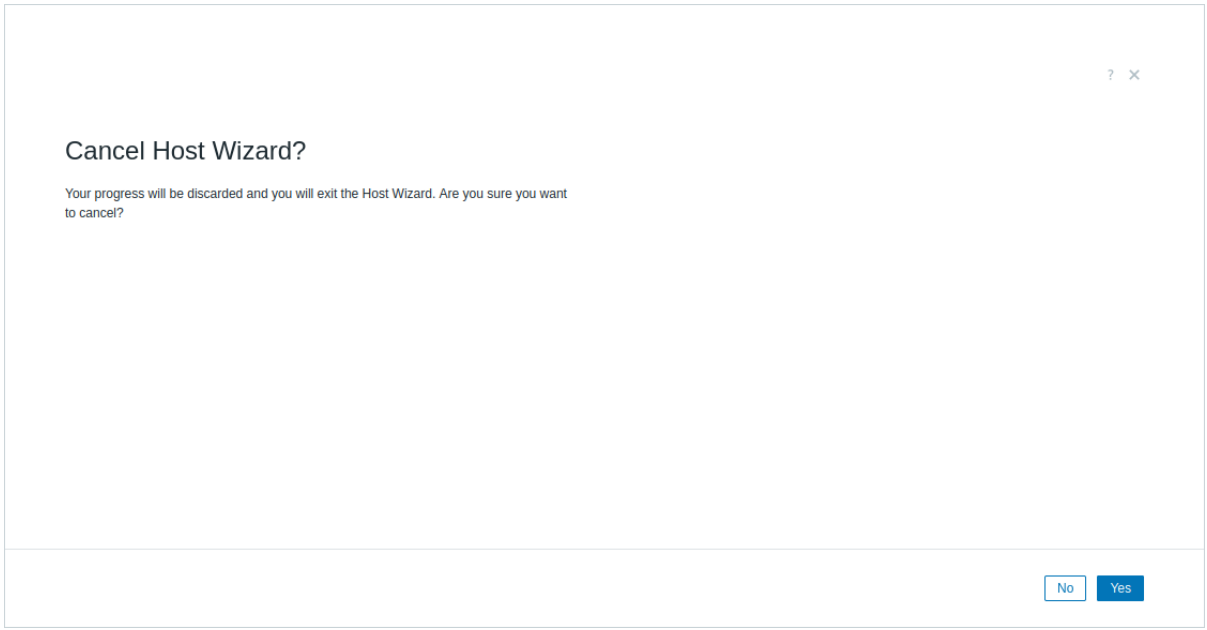
Bem-vindo ao Host Wizard

Quando você inicia o Host Wizard, uma tela de boas-vindas é exibida:



Para pular esta tela em sessões futuras, você pode marcar a caixa de seleção *Do not show welcome screen* e clicar em *Next*.

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



Clicar em *Yes* sairá do Host Wizard sem salvar seu progresso. Clicar em *No*, Esc ou no ícone de fechar retornará à última etapa.

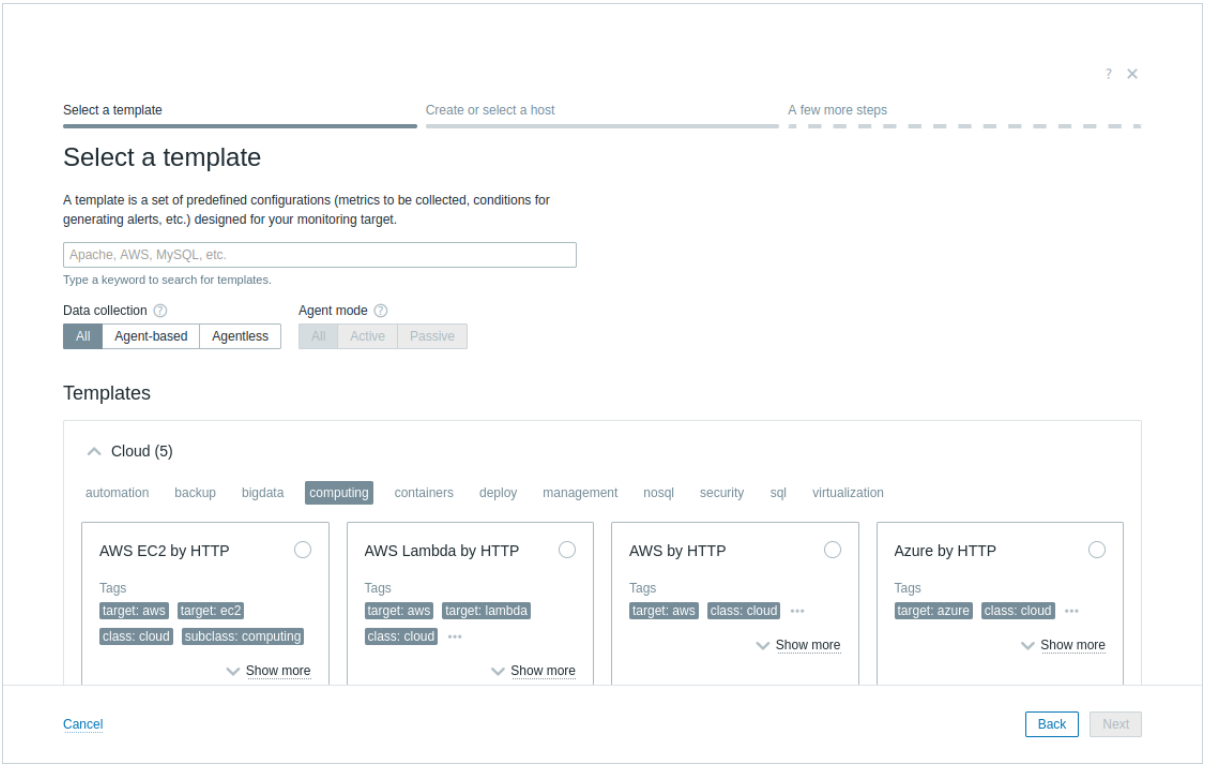
Selecione um template

A primeira etapa para configurar o seu alvo de monitoramento é selecionar um template, que é um conjunto de configurações predefinidas (métricas a serem coletadas, condições para gerar alertas etc.) desenvolvido para o seu alvo de monitoramento.

Os templates prontos para uso do Zabbix oferecem uma variedade de configurações de monitoramento predefinidas para sistemas operacionais, aplicações, 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, automação, backup), que são baseadas em tags de template.
- Pesquisar templates por palavras-chave no nome do template ou nos nomes e valores das tags de template.
- Filtrar templates pelo método de coleta de dados (baseado em agent ou sem agent; os templates baseados em agent incluem pelo menos um item de Zabbix agent).
- Filtrar templates pelo modo do agent (ativo ou passivo; os templates ativos incluem pelo menos um item de Zabbix agent (ativo), enquanto os templates passivos incluem pelo menos um item passivo; para obter detalhes, consulte Verificações passivas e ativas do agent).
- Ao usar o Host Wizard para configurar um host existente, você também pode filtrar os templates já vinculados ao host.



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

O template selecionado determinará as próximas etapas no Host Wizard. Por exemplo, se você selecionar o template *MySQL by Zabbix agent*, que usa o Zabbix agent para a coleta de dados, o assistente orientará você durante o processo de instalação do agent.

Criar ou selecionar um host

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

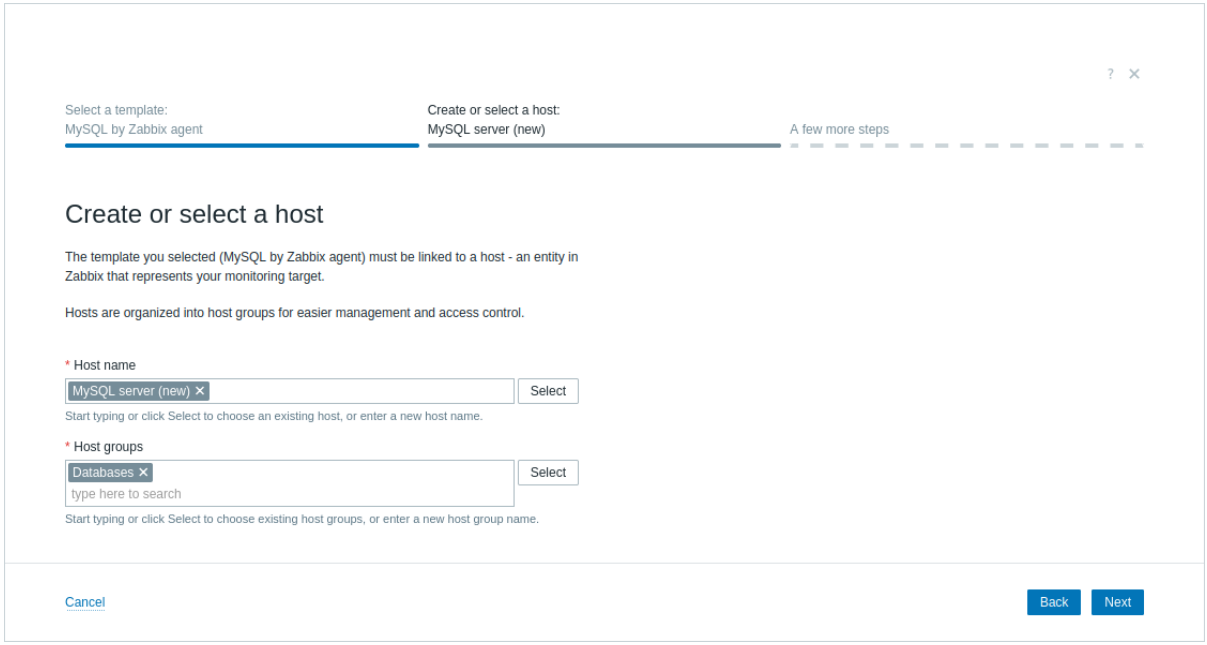
Cada host também deve pertencer a pelo menos um grupo de hosts, usado para organizar hosts e atribuir permissões de usuário a eles.

Nesta etapa, você pode:

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

Note:
Selecionar um host existente não remove nem substitui sua configuração atual, a menos que isso seja explicitamente indicado nas etapas posteriores. Hosts descobertos não podem ser selecionados.

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



Instalar o Zabbix agent

Os templates baseados em agent (como *MySQL by Zabbix agent*) exigem a instalação do Zabbix agent ou agent 2 - um processo implantado no seu alvo de monitoramento para monitorar ativamente recursos e aplicações locais (se exigido pelo template).

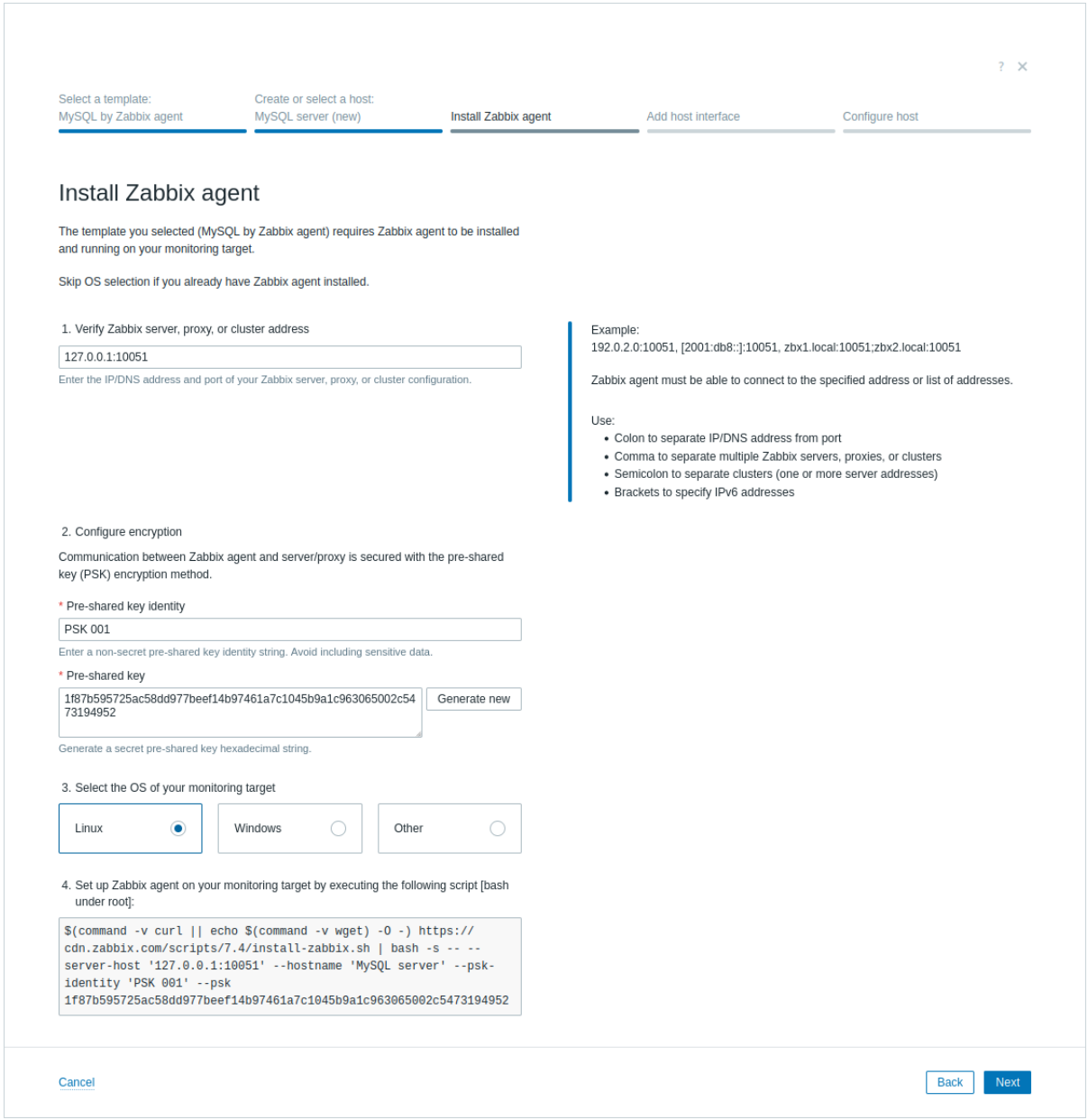
Nesta etapa:

- Verifique o endereço do Zabbix server, proxy ou cluster (por exemplo, 192.0.2.1:10051). Esse valor será usado para configurar os parâmetros Server e ServerActive do Zabbix agent.
- Insira uma identidade de chave pré-compartilhada nova, 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 com chave pré-compartilhada (PSK) para o host do Zabbix (por exemplo, *MySQL server*) e o Zabbix agent.

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

- Selecione o sistema operacional da máquina que hospeda o seu alvo de monitoramento.
- Instale o Zabbix agent nesse sistema executando o script fornecido nele 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 sistema operacional e executar o script fornecido nesse sistema. Depois de instalar o Zabbix agent, retorne ao Assistente de host.

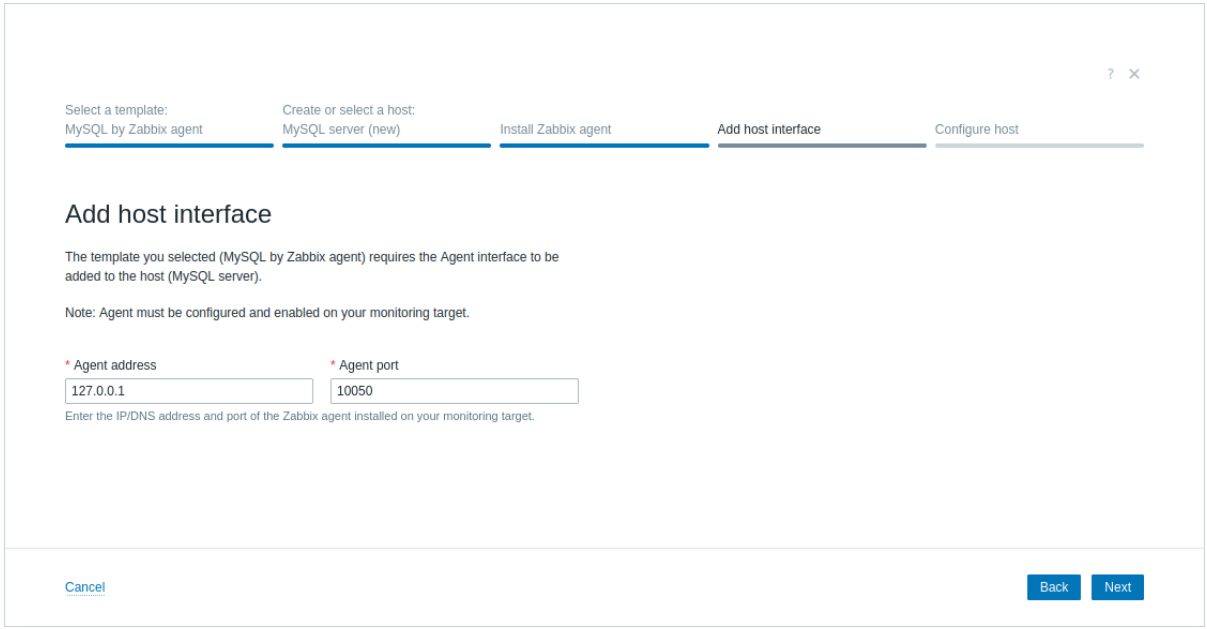


Adicionar interface do host

A host interface define como o server Zabbix 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, informe os detalhes da interface do host exigidos pelo template selecionado.

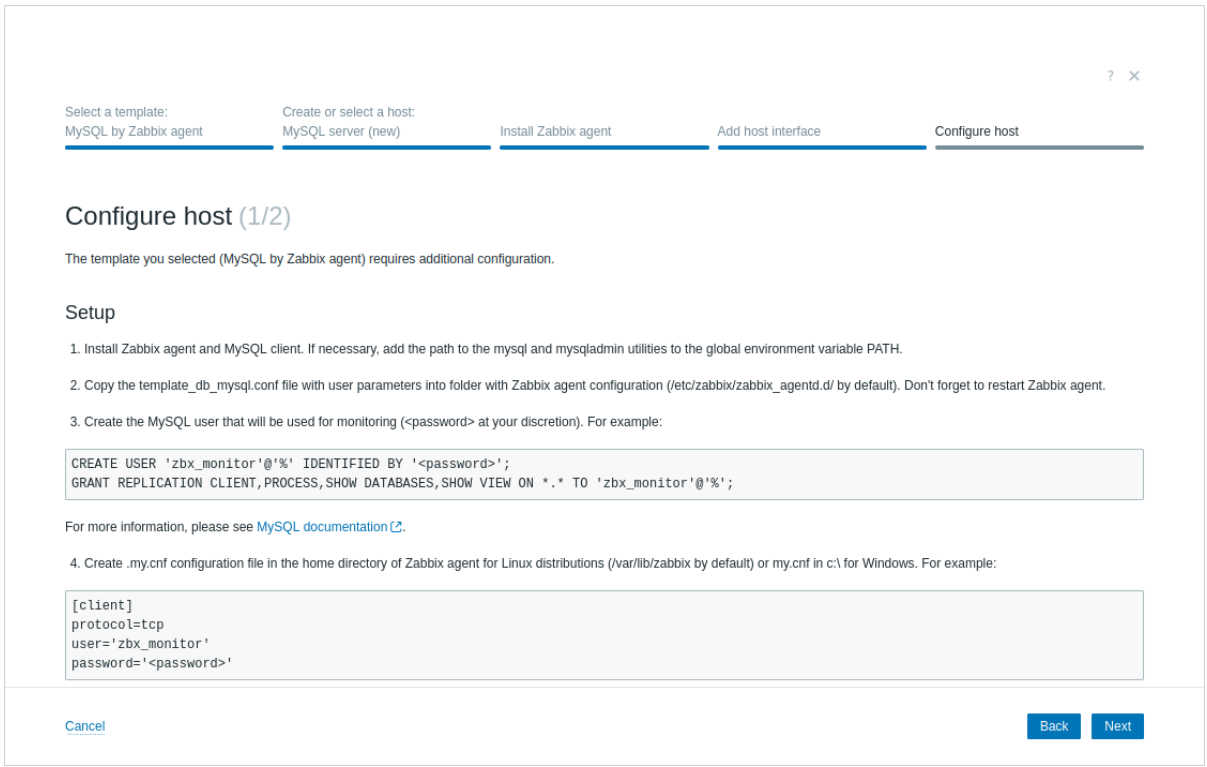
Por exemplo, você pode usar o endereço padrão do agent (`127.0.0.1`) e a porta (10050) se o server Zabbix, o agent Zabbix e o server MySQL estiverem todos em execução localmente na mesma máquina.



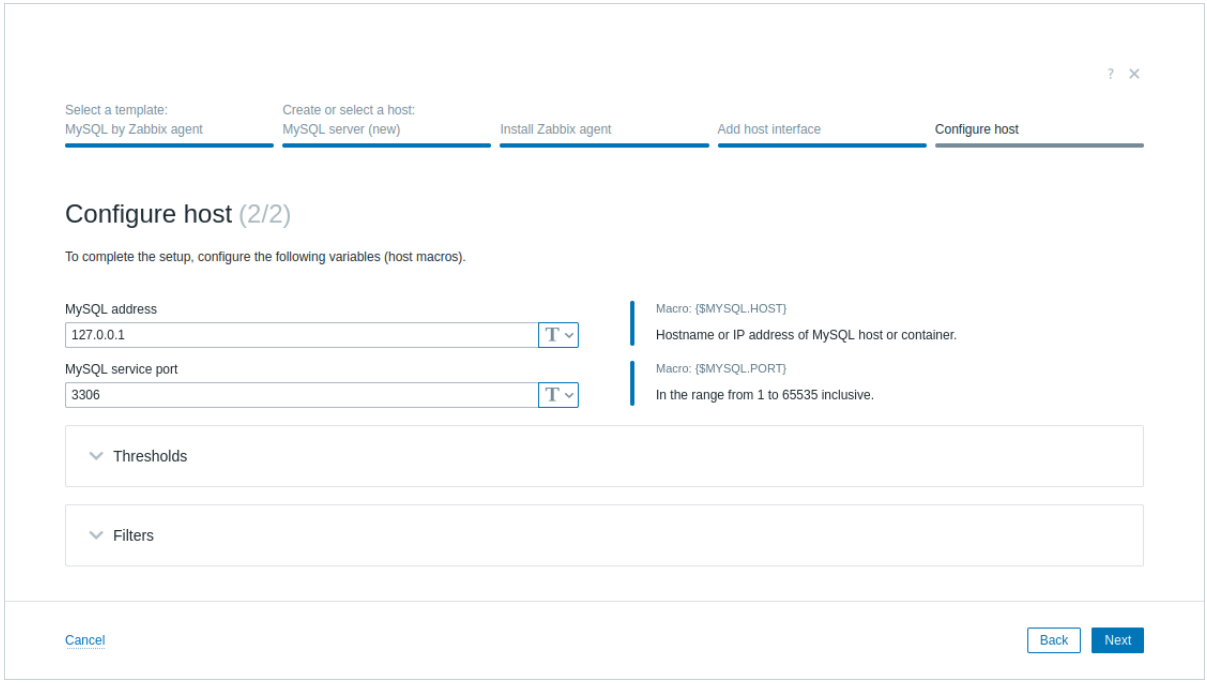
Configurar host

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

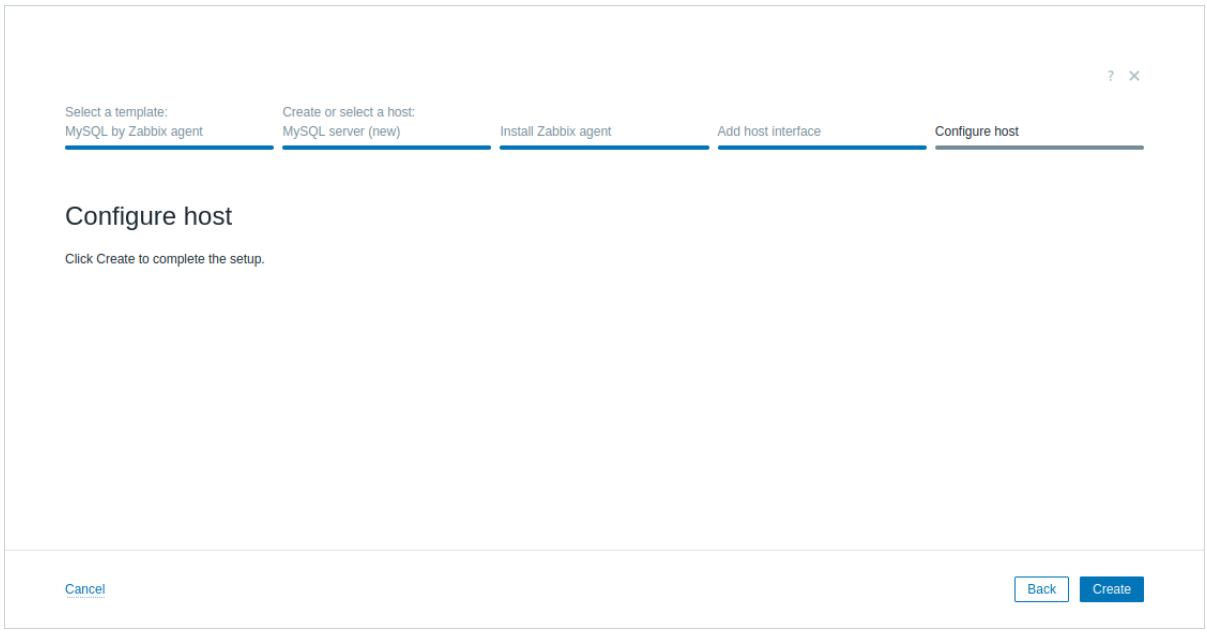
- Configuração manual do seu destino de monitoramento (por exemplo, habilitar serviços específicos, criar usuários de serviço ou conceder permissões):



- Definição de macros de host - variáveis que controlam o comportamento dos itens, as configurações de conexão, as credenciais de autenticação e outros parâmetros:



Depois de concluir todas as etapas de configuração necessárias, clique em *Criar* para adicionar o host ao Zabbix (ou em *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.

Configurando um host

Visão geral

Para configurar um host no frontend do Zabbix, faça o seguinte:

1. Vá para *Coleta de dados* > *Hosts* ou *Monitoramento* > *Hosts*.
2. Clique em *Criar host* no canto superior direito da tela (ou no nome do host para editar um host existente).
3. Insira os parâmetros do host no formulário.

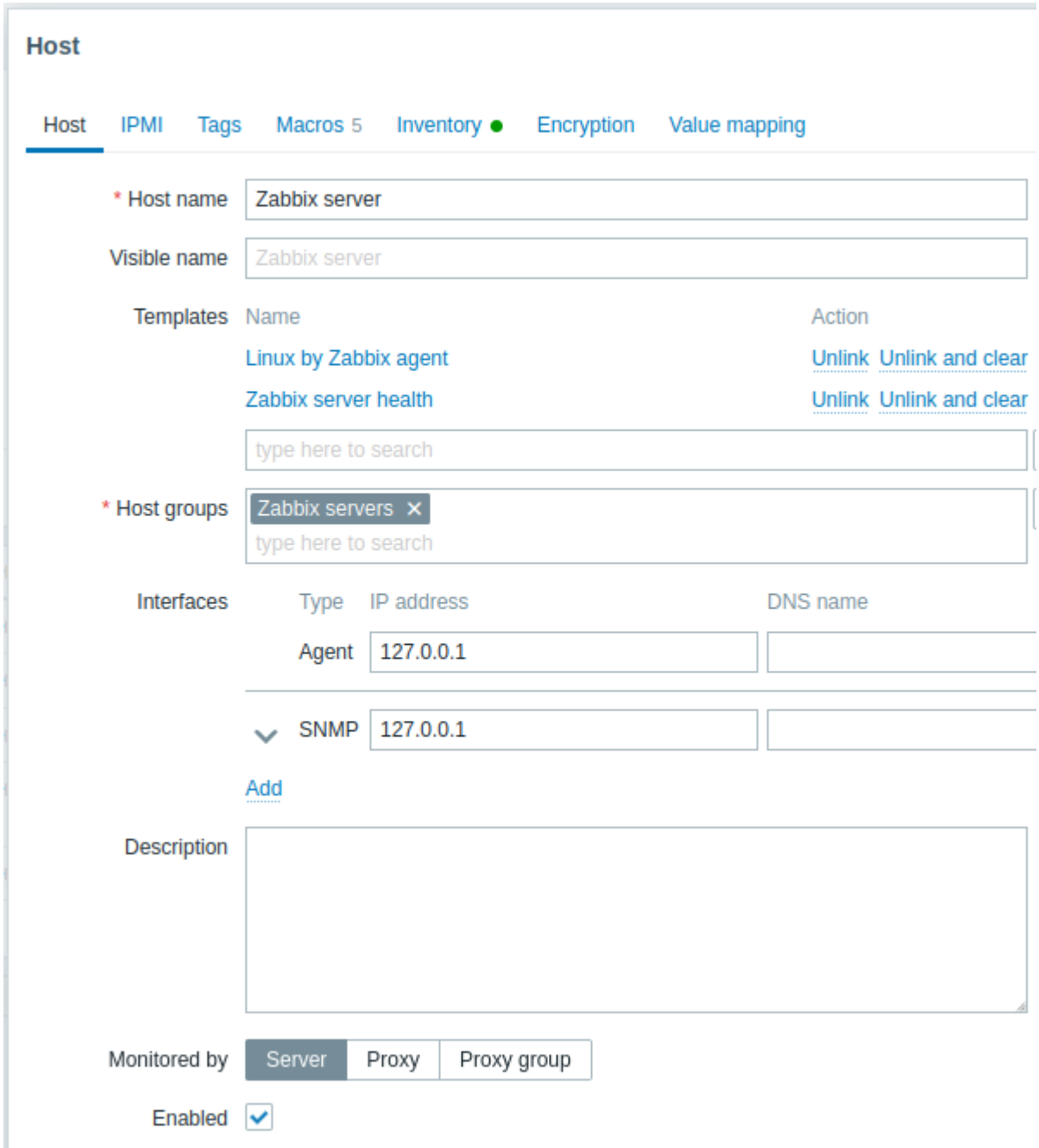
Você também pode usar o botão *Clonar* no formulário de configuração de um host existente para criar um novo host. Esse host terá todas as propriedades do host existente, incluindo templates vinculados, entidades (items, triggers etc.) desses templates, bem como as entidades diretamente associadas ao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

Observe que, quando um host é clonado, ele manterá todas as entidades do template exatamente como estão originalmente no template. Quaisquer alterações feitas nessas entidades no nível do host existente (como alteração do intervalo do item, modificação de expressão regular ou adição de protótipos à regra de descoberta de baixo nível) não serão clonadas para o novo host; em vez disso, elas permanecerão como estão no template.

Como alternativa, você pode usar o Host Wizard 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 de entrada obrigatórios são marcados com um asterisco vermelho.

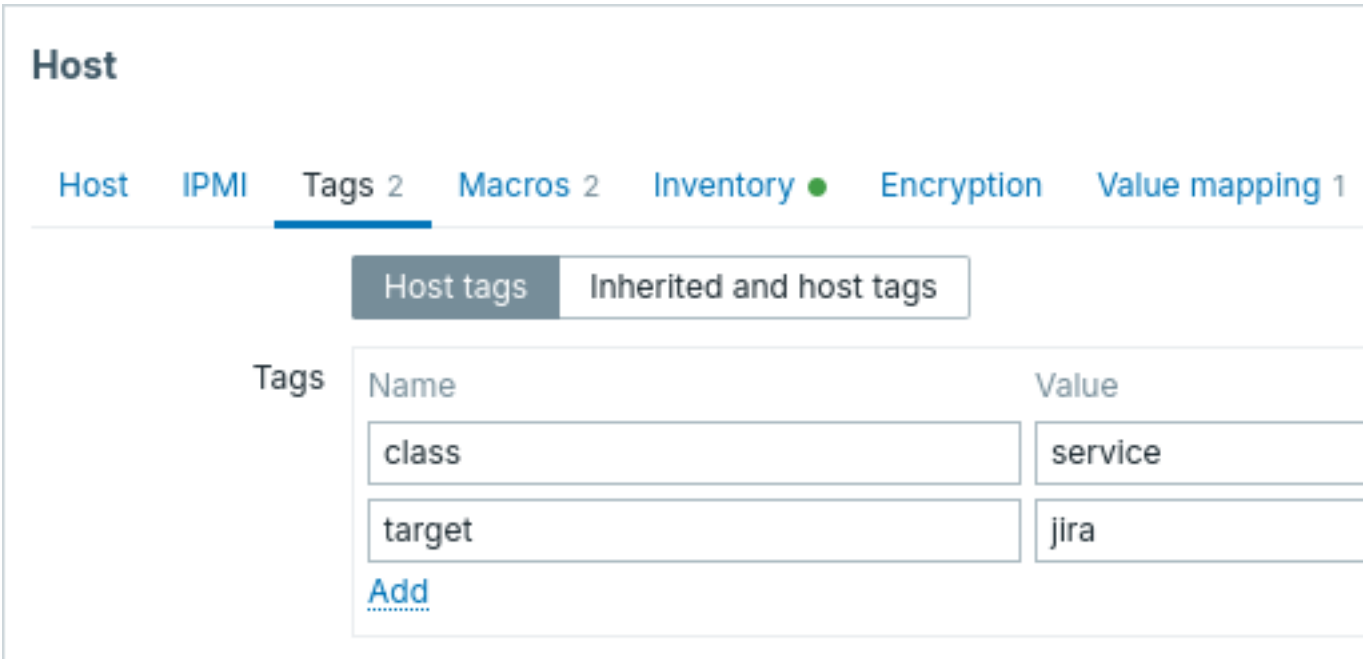
Parâmetro	Descrição
Nome do host	Insira um nome exclusivo para o host. São permitidos caracteres alfanuméricos, espaços, pontos, hífens e sublinhados. No entanto, espaços no início e no fim não são permitidos. *Observação:* com o Zabbix agent em execução no host que você está configurando, o parâmetro *Hostname* do arquivo de configuração do agent deve ter o mesmo valor do nome do host inserido aqui. O nome definido no parâmetro é necessário para o processamento de verificações ativas.
Nome visível	Insira um nome visível exclusivo para o host. Se você definir esse nome, ele será exibido em listas, mapas etc., em vez do nome técnico do host. Esse atributo oferece 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 entrada de texto. Uma lista de templates correspondentes será exibida; role para baixo para selecionar. Como alternativa, clique em *Selecionar* ao lado do campo e selecione os templates na lista exibida em uma janela pop-up. Todos os templates selecionados serão vinculados ao host quando o formulário de configuração do host for salvo ou atualizado. Para desvincular um template, use uma das duas opções no bloco *Templates vinculados*: *Desvincular* - desvincula o template, mas preserva suas entidades (items, triggers etc.); *Desvincular e limpar* - desvincula o template e remove todas as suas entidades (items, triggers etc.). Os nomes dos templates listados são links clicáveis que levam ao formulário de configuração do template.
Grupos de hosts	Selecione os grupos de hosts aos quais o host pertence. Um host deve pertencer a pelo menos um grupo de hosts. Um novo grupo pode ser criado e vinculado ao host adicionando um nome de grupo inexistente.
Interfaces	Vários tipos de interface de host são compatíveis: *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*. *Observação:* as interfaces 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 traps SNMP. Durante a correspondência, somente o ”IP” ou ”DNS” selecionado na interface do host é usado. Consulte Configurando o monitoramento SNMP para obter 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 botão correspondente informará ao Zabbix server o que deve ser usado 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 Zabbix agent, 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 proxies** - o host é monitorado por um grupo de proxies.
Proxy	O nome do proxy atribuído é exibido (somente se o Zabbix server tiver atribuído um proxy do grupo de proxies selecionado). Esse campo é exibido somente se o host for monitorado por um grupo de proxies.
Habilitado	Quando a caixa de seleção está marcada, o host está habilitado e pronto para monitoramento. Quando a caixa de seleção não está marcada, o host está desabilitado e não é monitorado: Para solicitações de dados passivas iniciadas pelo Zabbix server/proxy (por exemplo, Zabbix agent, agent SNMP, verificações simples), o monitoramento é desabilitado após a sincronização da configuração. Os triggers e as ações vinculados ao host também são desabilitados somente após o recarregamento da configuração. Para verificações ativas do Zabbix agent, o monitoramento é interrompido dentro do intervalo de tempo (aproximadamente 5 segundos) em que o Zabbix agent recebe a informação de que o host foi desabilitado. Durante esse breve intervalo, o host continuará coletando dados localmente para as verificações ativas e tentará enviá-los ao server/proxy; no entanto, como o host está marcado como *Desabilitado*, o server/proxy rejeitará os dados. Quando você desabilita o host, seus items são imediatamente removidos do cache de histórico (exceto pelos últimos valores, que são mantidos para os logs).

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

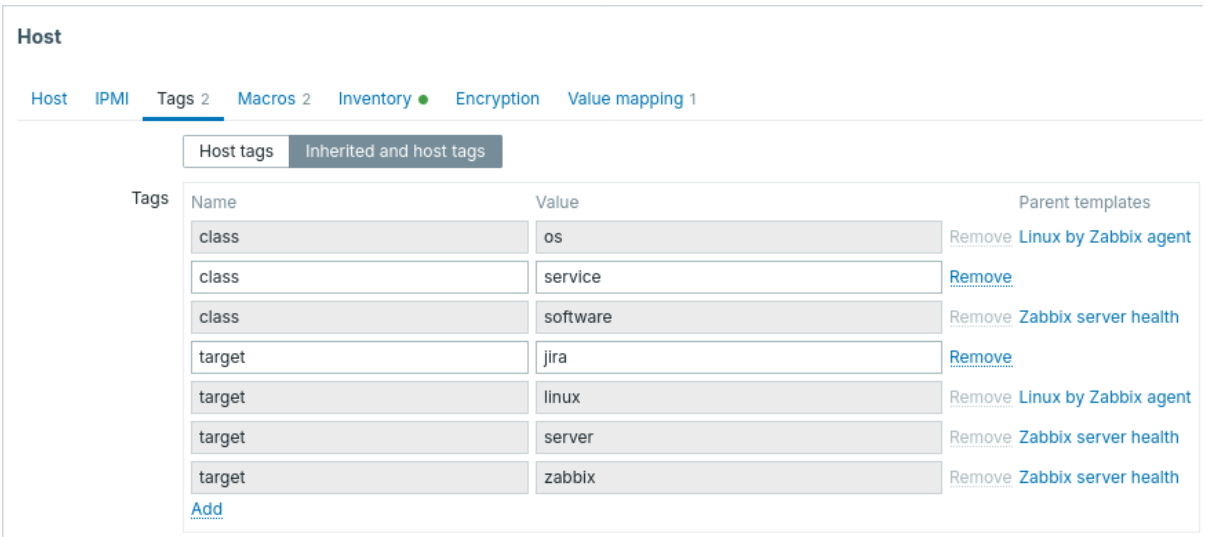
Parâmetro	Descrição
Algoritmo de autenticação	Selecione o algoritmo de autenticação.
Nível de privilégio	Selecione o nível de privilégio.
Nome de usuário	Nome de usuário para autenticação. Macros de usuário podem ser usadas.
Senha	Senha para autenticação. Macros de usuário podem ser usadas.

A aba **Tags** permite definir tags no nível do host. Todos os problemas desse host receberão as tags com os valores inseridos aqui.



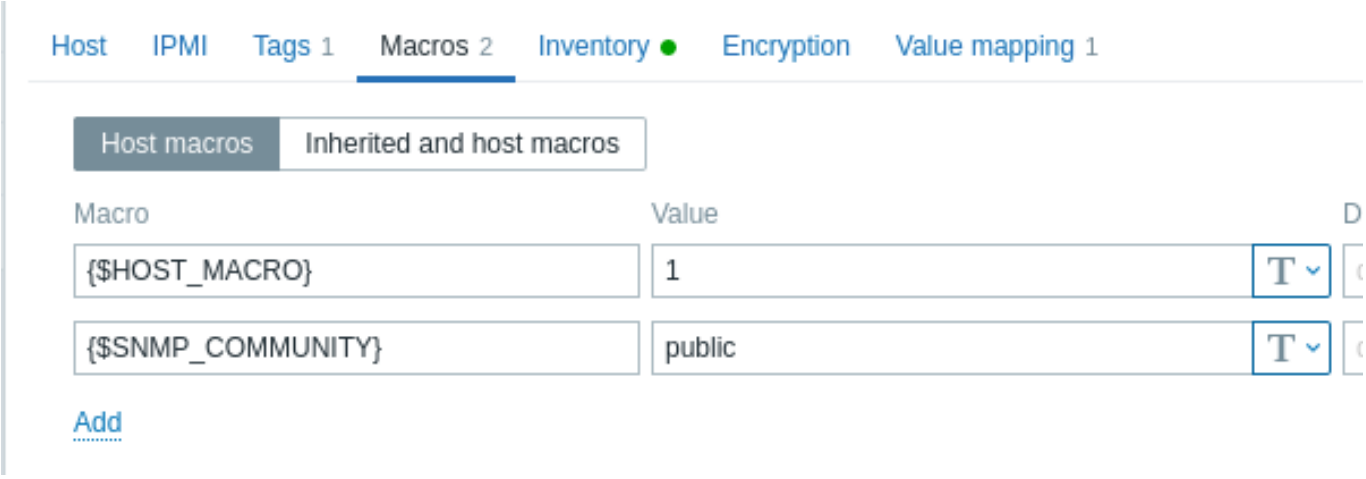
Macros de usuário, macros `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` e `{HOST.ID}` são compatíveis com tags.

Você também pode visualizar aqui as tags no nível do template se selecionar a opção *Tags herdadas e do host*. É nesse local que todas as tags de usuário definidas para o host são exibidas, assim como sua origem.

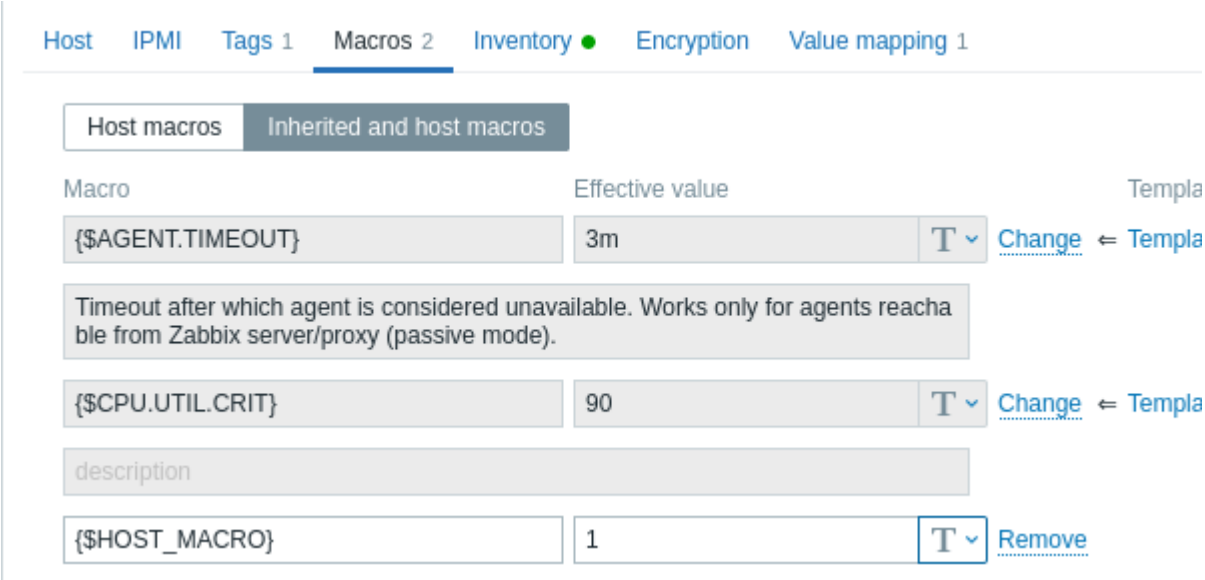


Para sua conveniência, são fornecidos links para os templates correspondentes.

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.

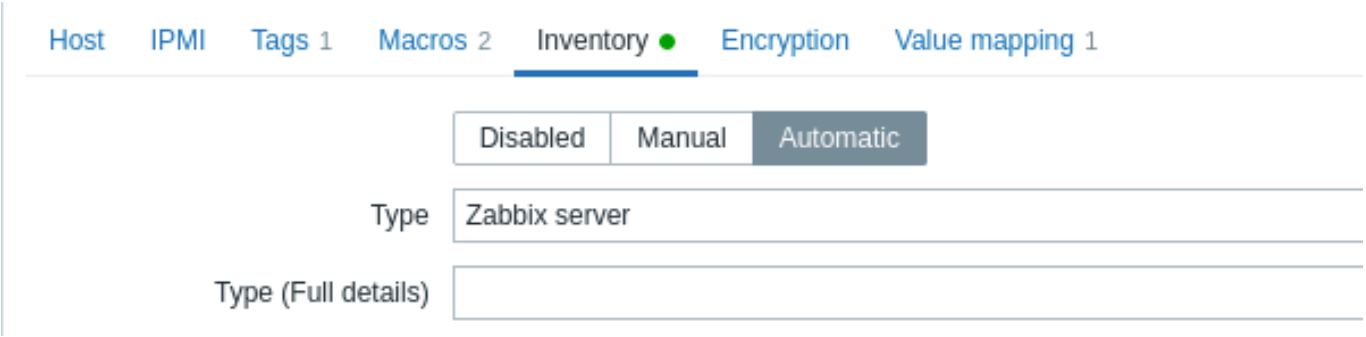


Você também pode visualizar aqui as macros de usuário no nível do template e globais se selecionar a opção *Macros herdadas e do host*. É nesse local que todas as macros de usuário definidas para o host são exibidas, juntamente com o valor para o qual são resolvidas e sua origem.



Para sua conveniência, são fornecidos links para os templates correspondentes e para a configuração de macros globais. 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 do host. Você também pode optar por habilitar o preenchimento de inventário *Automático* ou desabilitar o preenchimento de inventário para este host.



Se o inventário estiver habilitado (manual ou automático), um ponto verde será exibido ao lado do nome da aba.

Criptografia

A aba **Criptografia** permite que você exija conexões criptografadas com o host.

Parameter	Description
Connections to host	Como o server ou proxy do Zabbix se conecta ao agent do Zabbix em um host: sem criptografia (padrão), usando PSK (chave pré-compartilhada) ou certificado.

Parameter	Description
Connections from host	Selecione que tipo de conexões são permitidas a partir do host (ou seja, do agent do Zabbix e do Zabbix sender). Vários tipos de conexão podem ser selecionados ao mesmo tempo (útil para testes e para alternar para outro tipo de conexão). O padrão é "No encryption".
Issuer	Emissor permitido do certificado. O certificado é primeiro validado com a CA (autoridade certificadora). Se ele for válido e assinado pela CA, então o campo *Issuer* pode ser usado para restringir ainda mais a CA permitida. Este campo foi criado para ser usado se sua instalação do Zabbix utilizar certificados de várias CAs. Se este campo estiver vazio, qualquer CA será aceita.
Subject	Subject permitido do certificado. O certificado é primeiro validado com a CA. Se ele for válido e assinado pela CA, então o campo *Subject* pode ser usado para permitir apenas um valor da string *Subject*. Se este campo estiver vazio, qualquer certificado válido assinado pela CA configurada será aceito.
PSK identity	String de identidade da chave pré-compartilhada. Não coloque informações sensíveis na identidade da PSK, pois ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK usar.
PSK	Chave pré-compartilhada (string hexadecimal). 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.

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 template. Portanto, mesmo que um único usuário precise de acesso a um único host, isso é concedido adicionando o usuário a um grupo de usuários e atribuindo a esse grupo de usuários permissões para o grupo de hosts que contém esse host.

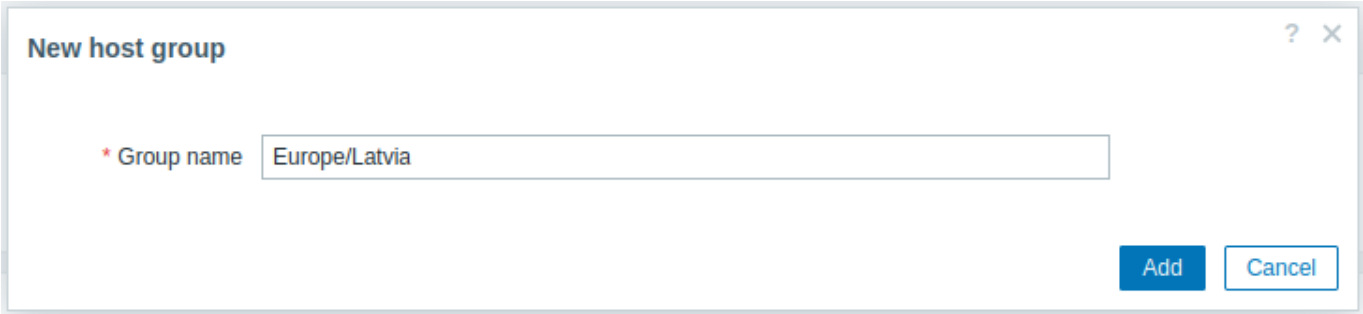
Configuração

> **Attention:**
> Somente usuários Super admin podem criar grupos de host.

Há duas opções para criar um grupo de host no frontend do Zabbix.

Opção um:

1. Acesse *Data collection* > *Host groups*.
2. Clique em *Create host group* no canto superior direito da tela.
3. Informe o nome do grupo no formulário.



Opção dois: ao configurar um host, informe um nome de grupo inexistente no campo *Host groups*.

Depois que o grupo de host for criado, você pode clicar no nome do grupo na lista em *Data collection* > *Host groups* para editar o nome do grupo, clonar o grupo ou excluí-lo.

A exclusão de um grupo de host remove 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`.

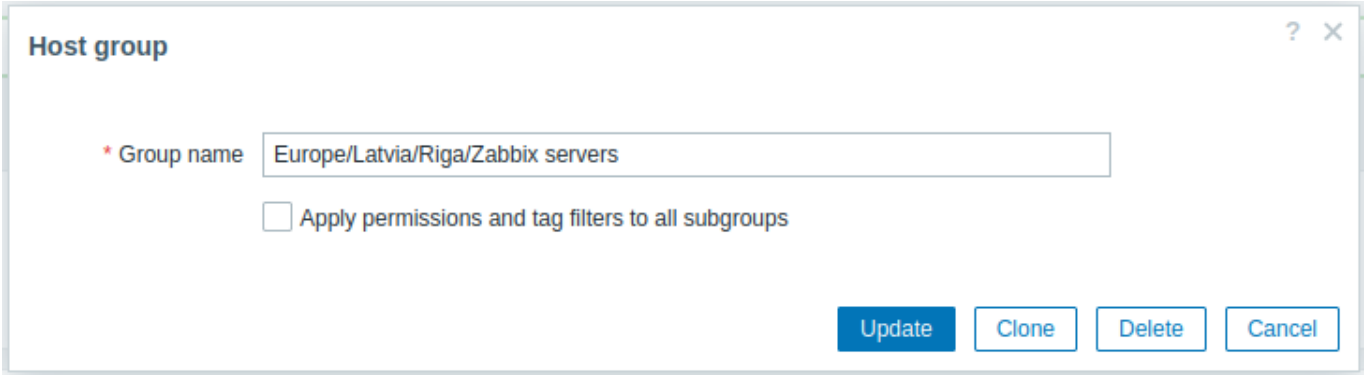
Ao criar um subgrupo, não é permitido usar barras à esquerda ou à direita, ou várias barras seguidas. O escape de '/' não é suportado.

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 em um grupo de hosts pai existente (por exemplo, ao criar `Europe/Latvia` quando `Europe` já existe), as permissões do user group para o subgrupo são herdadas do pai.
- Ao criar um grupo de hosts pai para um subgrupo existente (por exemplo, ao criar `Europe` quando `Europe/Latvia` já existe), nenhuma permissão para o pai é definida.

Ao editar qualquer grupo de hosts, você também pode definir uma opção adicional, *Aplicar permissões e filtros de tag a todos os subgrupos*.



Marcar esta caixa de seleção e clicar em *Update* aplicará o mesmo nível de permissões e filtros de tag a todos os subgrupos atuais e futuros do grupo de hosts que está sendo editado.

Portanto, se algum user group tiver recebido permissions diferentes 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 tag do grupo que está sendo editado.

Observe que essa opção não é salva no banco de dados e substituirá as permissões existentes. Quaisquer alterações feitas por meio dessa opção só podem ser revertidas manualmente.

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 o tipo de dispositivo, número de série, localização, pessoa responsável, URLs etc. - dados que preencherão 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 isso, ao configurar um host, o modo de inventário na aba *Inventory* deve ser definido como *Automatic*.

Em seguida, você pode configurar itens do host para preencher qualquer campo do inventário do host com o valor correspondente, indicando o campo de destino por meio do atributo correspondente (chamado *Item will populate host inventory field*) na configuração do item.

Itens especialmente úteis para a coleta automatizada de dados de inventário:

- system.hw.chassis[full|type|vendor|model|serial] - o padrão é [full], são necessárias permissões de root
- 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 uma expressão regular
- 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.

Por padrão, o modo de inventário para novos hosts é selecionado com base na configuração *Default host inventory mode* em *Administration > General > Other*.

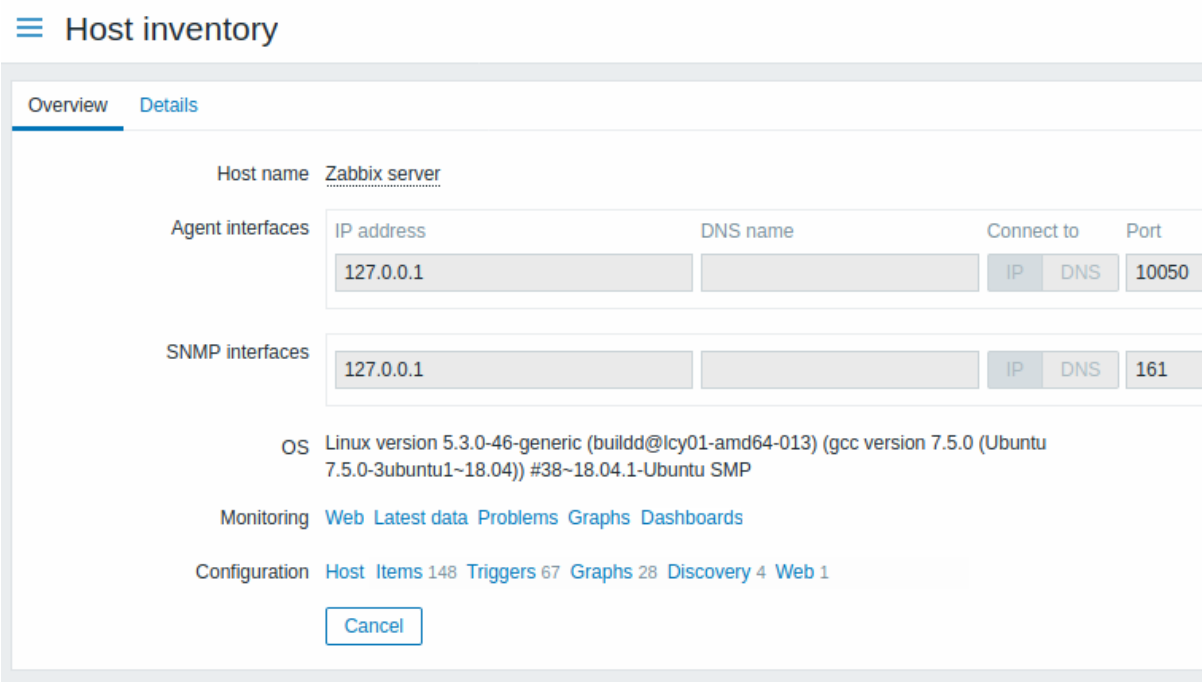
Para hosts adicionados por descoberta de rede ou por ações de registro automático, é possível definir uma operação *Set host inventory mode*, selecionando o modo manual ou automático. Essa operação substitui a configuração *Default host inventory mode*.

Visão geral do inventário

Os detalhes de todos os dados de inventário existentes estão disponíveis no menu *Inventory*.

Em *Inventory > Overview*, você pode obter a contagem de hosts por vários campos do inventário.

Em *Inventory > Hosts*, você pode ver todos os hosts que possuem informações de inventário. Ao clicar no nome do host, os detalhes do inventário serão exibidos em um formulário.



A aba **Overview** mostra:

Parameter	Description
Host name	Nome do host. Ao clicar no nome, é aberto 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.
Visible name	Nome visível do host (se definido).
Host (Agent, SNMP, JMX, IPMI) interfaces	Este bloco fornece detalhes das interfaces configuradas para o host.
OS	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).
Description	Descrição do host.
Monitoring	Links para as seções de monitoramento com dados deste host: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Links para as seções de configuração deste host: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. A quantidade de entidades configuradas é □□□□ada após cada link.

A aba **Details** mostra todos os campos de inventário que estão preenchidos (não estão vazios).

Macros de inventário

Há macros de inventário do host `{INVENTORY.*}` disponíveis para uso em notificações, por exemplo:

"O server em {INVENTORY.LOCATION1} está com um problema, a pessoa responsável é {INVENTORY.CONTACT1}, o número de telefone é {INVENTORY.POC.PRIMARY.PHONE.A1}."

Para mais detalhes, consulte a página Macros suportadas.

Atualização em massa

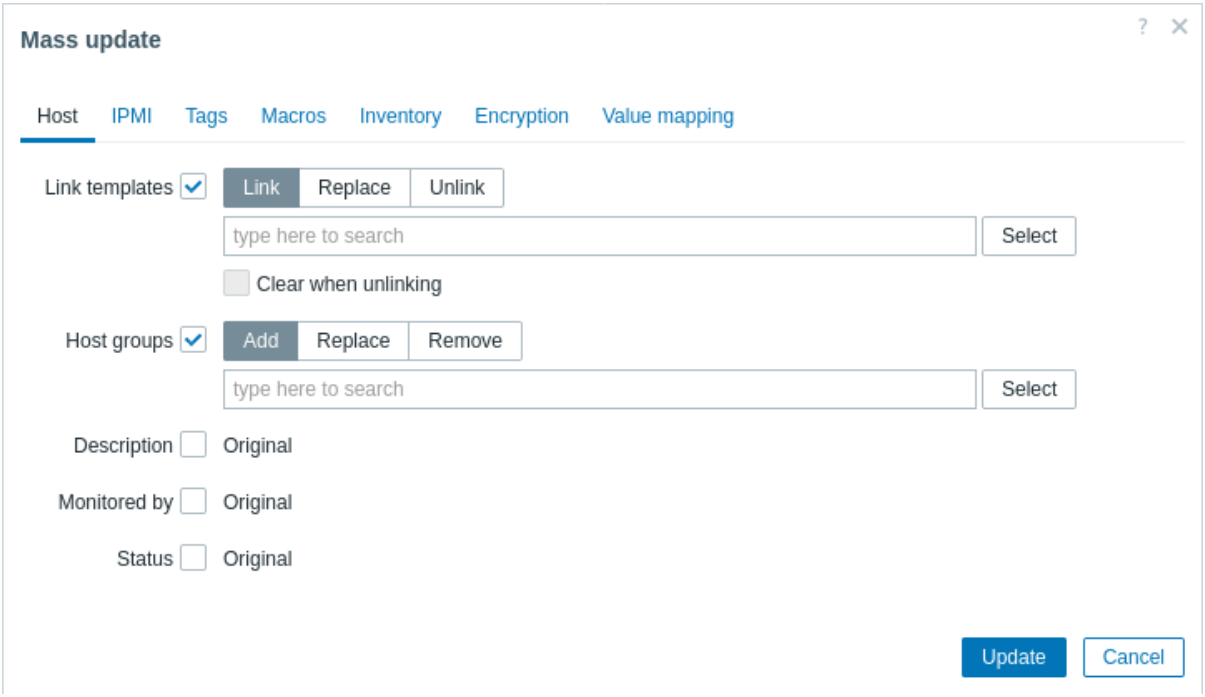
Visão geral

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 a atualização em massa

Para atualizar vários hosts em massa, faça o seguinte:

- Marque as caixas de seleção antes dos hosts que deseja atualizar na lista de hosts
- Clique em *Atualização em massa* abaixo da lista
- Navegue até a aba com os atributos necessários (*Host*, *IPMI*, *Tags*, *Macros*, *Inventário*, *Criptografia* ou *Mapeamento de valores*)
- Marque as caixas de seleção de qualquer atributo que deseja atualizar e insira um novo valor para eles



As seguintes opções estão disponíveis ao selecionar o botão correspondente para atualizar o vínculo com **template**:

- *Vincular* - especifique quais templates adicionais devem ser vinculados
- *Substituir* - especifique quais templates devem ser vinculados, desvinculando qualquer template que já estivesse vinculado aos hosts
- *Desvincular* - especifique quais templates devem ser desvinculados

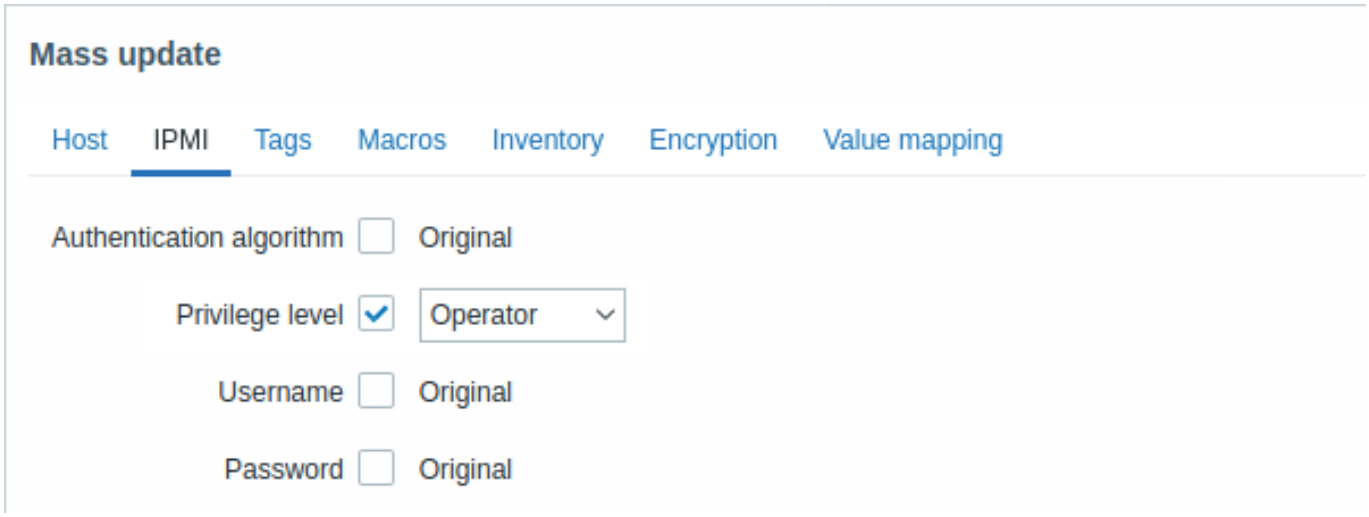
Para especificar os templates a serem vinculados/desvinculados, comece a digitar o nome do template no campo de preenchimento automático até que uma lista suspensa apareça oferecendo os templates correspondentes. Basta rolar para baixo para selecionar o template necessário.

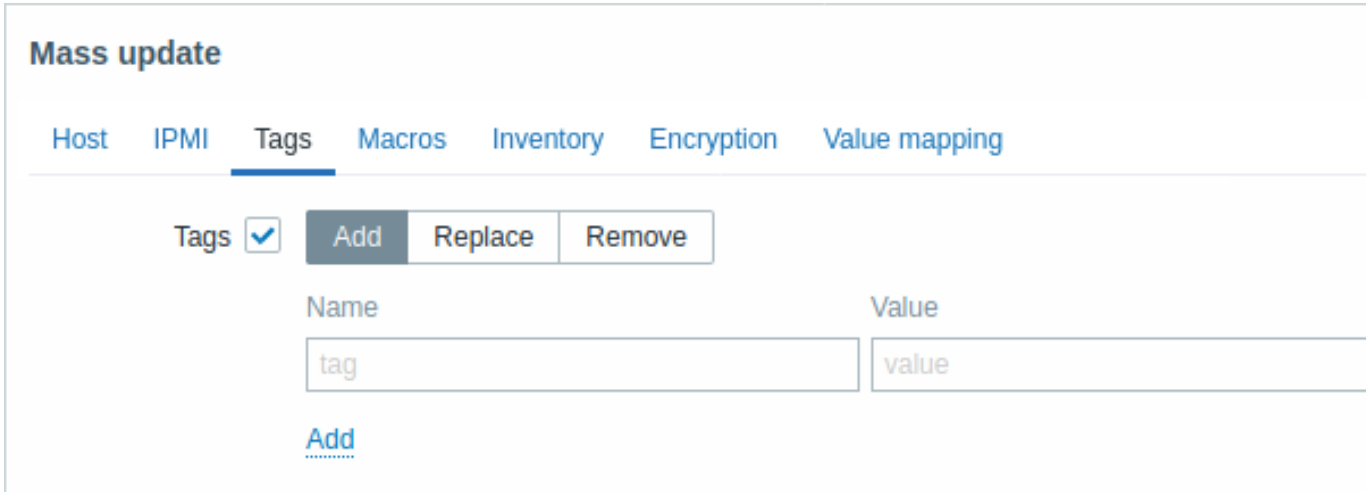
A opção *Limpar ao desvincular* permite não apenas desvincular templates vinculados anteriormente, mas também remover todas as entidades herdadas deles (items, triggers, gráficos etc.).

As seguintes opções estão disponíveis ao selecionar o botão correspondente para atualizar o **grupo de hosts**:

- *Adicionar* - especifique grupos de hosts adicionais entre os existentes ou insira grupos de hosts completamente novos para os hosts;
- *Substituir* - remova o host de todos os grupos de hosts existentes e substitua-os pelo(s) especificado(s) neste campo (grupos de hosts existentes ou novos);
- *Remover* - remova grupos de hosts específicos dos hosts.

Esses campos usam preenchimento automático: ao começar a digitar neles, uma lista suspensa de grupos de hosts correspondentes será exibida. Se o grupo de hosts for novo, ele também aparecerá na lista suspensa e será indicado por *(novo)* após a string. Basta rolar para baixo para selecionar.

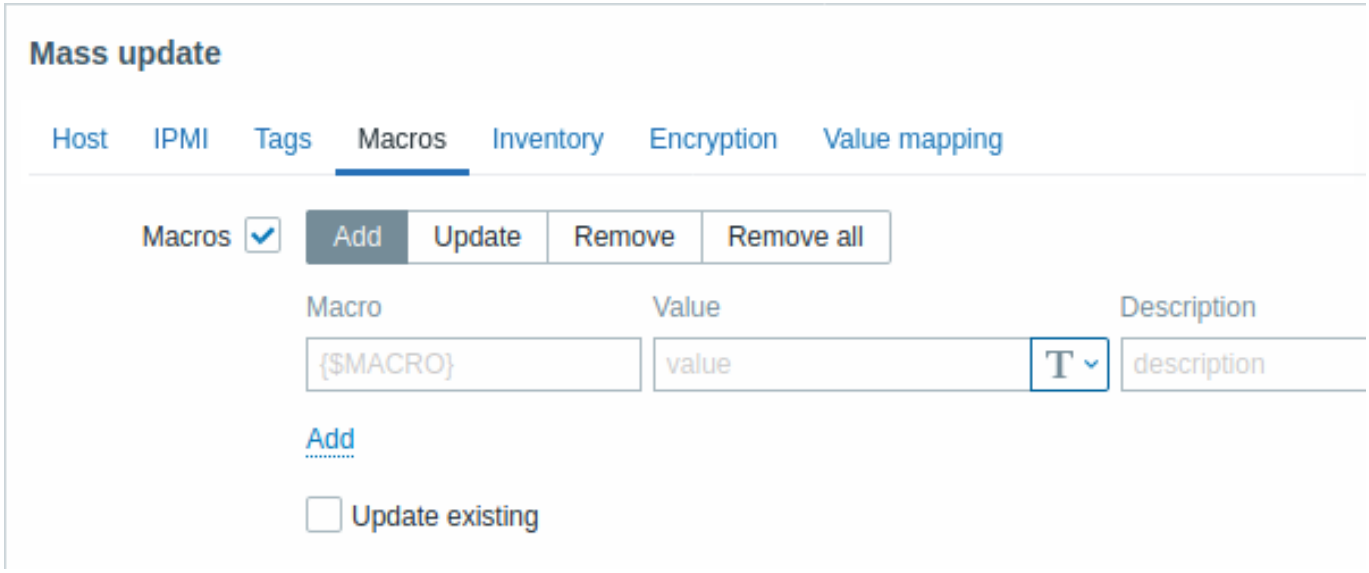




As seguintes opções estão disponíveis ao selecionar o botão correspondente para atualizar tags:

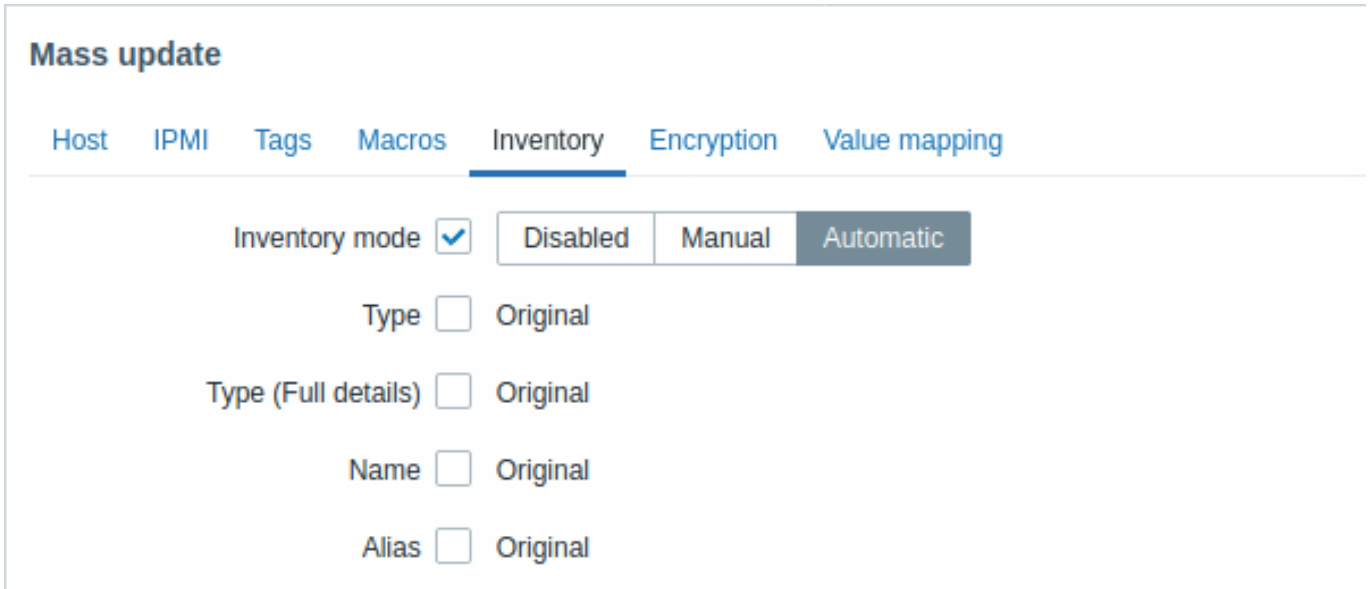
- *Adicionar* - adicione novas tags aos hosts;
- *Substituir* - remova todas as tags existentes do host e substitua-as pela(s) especificada(s) abaixo;
- *Remover* - remova as tags especificadas dos hosts.

Macros de usuário, macros `{INVENTORY.*}`, macros `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` e `{HOST.ID}` são compatíveis com tags. Observe que tags com o mesmo nome, mas valores diferentes, não são consideradas 'duplicatas' e podem ser adicionadas ao mesmo host.

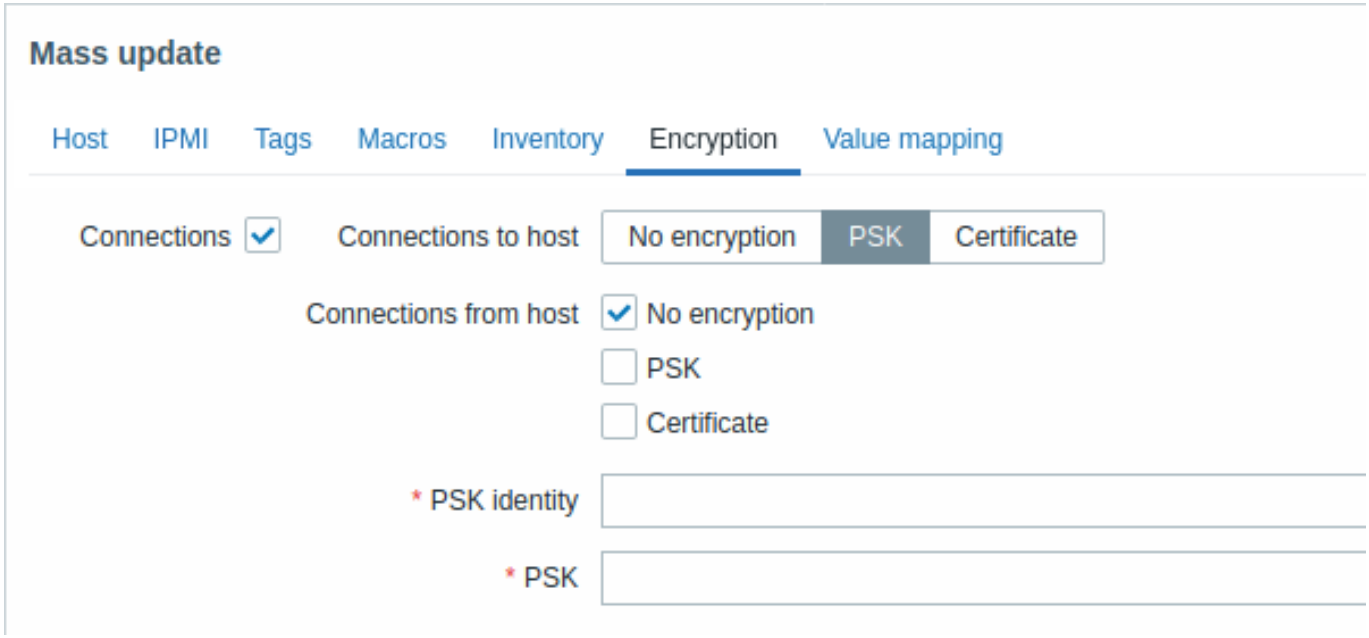


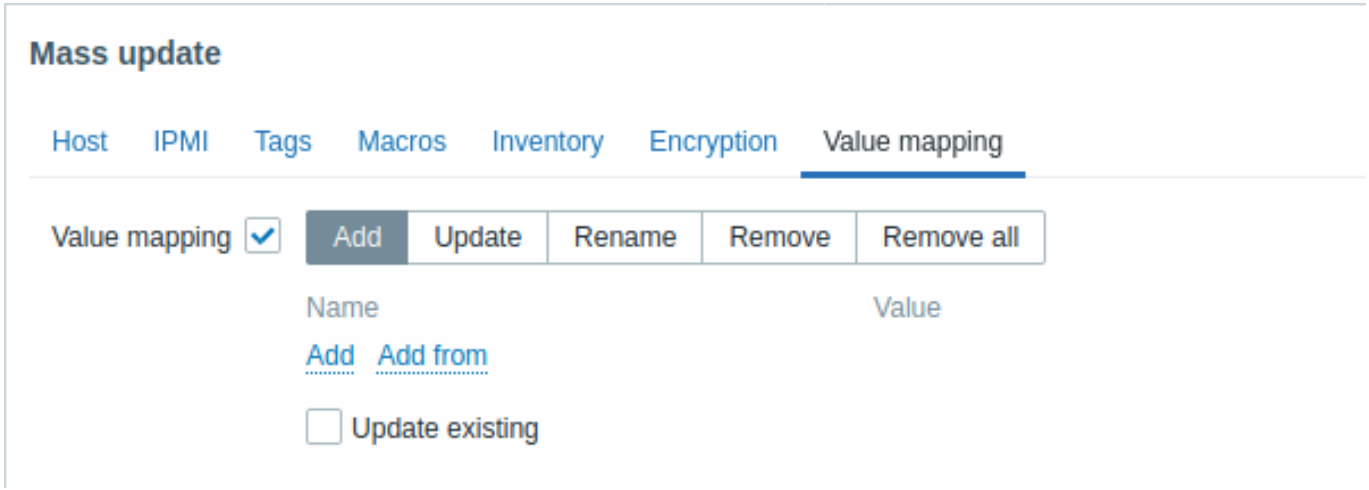
As seguintes opções estão disponíveis ao selecionar o botão correspondente para atualizar macros:

- *Adicionar* - especifique macros de usuário adicionais para os hosts. Se a caixa de seleção *Atualizar existentes* estiver marcada, o valor, o tipo e a descrição do nome da macro especificada serão atualizados. Se não estiver marcada, se já existir uma macro com esse nome no(s) host(s), ela não será atualizada.
- *Atualizar* - substitua os valores, tipos e descrições das macros especificadas nesta lista. Se a caixa de seleção *Adicionar ausentes* estiver marcada, a macro que não existia anteriormente em um host será adicionada como uma nova macro. Se não estiver marcada, somente as macros que já existem em um host serão atualizadas.
- *Remover* - remova as macros especificadas dos hosts. Se a caixa de seleção *Exceto as selecionadas* estiver marcada, todas as macros, exceto as especificadas na lista, serão removidas. Se não estiver marcada, somente as macros especificadas na lista serão removidas.
- *Remover todas* - remova todas as macros de usuário dos hosts. Se a caixa de seleção *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 estar definido como 'Manual' ou 'Automático'.





Botões com as seguintes opções estão disponíveis para atualizar o mapeamento de valores:

- *Adicionar* - adicione mapas de valores aos hosts. 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* - atualize os mapas de valores existentes. Se você marcar *Adicionar ausentes*, um mapa de valores que não existia anteriormente em um host será adicionado como um novo mapa de valores. Caso contrário, somente os mapas de valores que já existem em um host serão atualizados.
- *Renomear* - atribua um novo nome a um mapa de valores existente
- *Remover* - remova os mapas de valores especificados dos hosts. Se você marcar *Exceto as selecionadas*, todos os mapas de valores serão removidos **exceto** os especificados.
- *Remover todos* - remova todos os mapas de valores dos hosts. 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.

Quando terminar todas as alterações necessárias, clique em *Atualizar*. Os atributos serão atualizados de acordo para todos os hosts selecionados.

Itens

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.

Criando um item

Visão geral

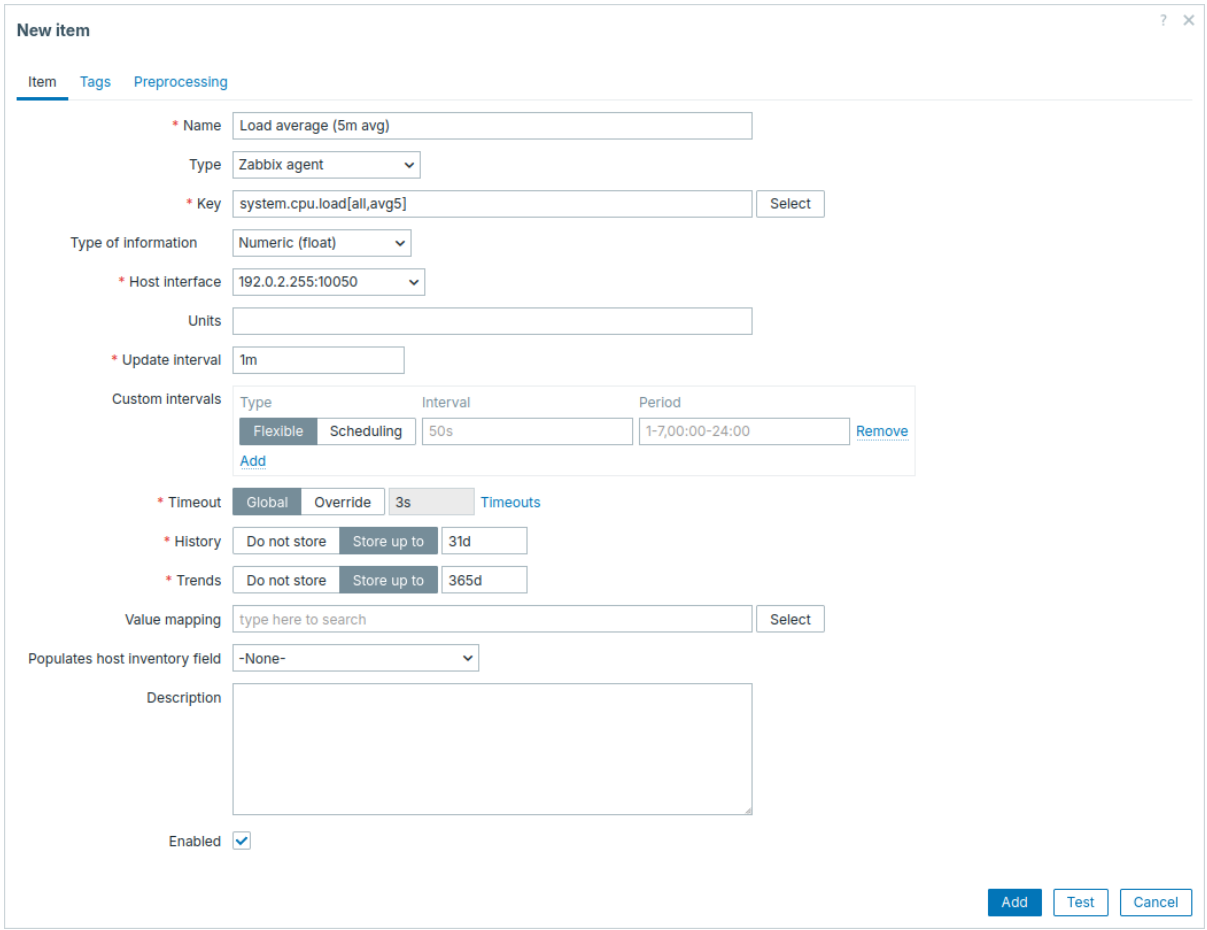
Para criar um item no frontend do Zabbix, faça o seguinte:

1. Acesse *Coleta de dados* > *Hosts*.
2. Clique em *Itens* na linha do host.
3. Clique em *Criar item* no canto superior direito da tela.
4. Insira os parâmetros do item no formulário.

Como alternativa, você pode criar um item abrindo um item existente, clicando no botão *Clonar* e salvando-o com um nome diferente.

Configuração

A aba **Item** contém atributos gerais do item.



Todos os campos de preenchimento obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Name	Nome do item. Macros de usuário são compatíveis.
Type	Tipo de item. Consulte as seções individuais de tipos de item.
Key	Chave do item (até 2048 caracteres). As chaves de item compatíveis podem ser encontradas nas seções individuais de tipos de item. A chave deve ser exclusiva em um único host. Se o tipo de chave for 'Zabbix agent', 'Zabbix agent (active)' ou 'Simple check', o valor da chave deverá ser compatível com o Zabbix agent ou o Zabbix server. Consulte também o formato correto da chave.
Type of information	O tipo de dados usado para validar o valor do item e armazená-lo no banco de dados, após quaisquer conversões: **Numeric (unsigned)** - inteiro sem sinal de 64 bits; **Numeric (float)** - número de ponto flutuante de 64 bits; **Character** - texto curto; **Log** - texto longo com ou sem propriedades de log (timestamp, origem, severidade, ID do evento de log); **Text** - texto longo; **Binary** - número binário (compatível somente com itens dependentes); **JSON** - dados JSON estruturados, armazenados nativamente no banco de dados (não compatível com itens calculados). Itens que retornam valores com o tipo de dados binário ou JSON não são compatíveis com fórmulas de itens calculados ou expressões de trigger. Consulte limites de dados dos itens para conhecer os limites de armazenamento por tipo de dados e backend de banco de dados. Para itens que retornam dados em um único formato específico, um tipo de dados correspondente é selecionado automaticamente — por exemplo, `system.cpu.load` e `system.cpu.util` retornam um número de ponto flutuante de 64 bits, portanto *Numeric (float)* é selecionado automaticamente.
Host interface	Selecione a interface do host. Este campo está disponível ao editar um item no nível do host.
Units	Se um símbolo de unidade for definido, o Zabbix aplicará pós-processamento ao valor recebido do item e o exibirá com o sufixo de unidade especificado. Símbolos de unidade compatíveis 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 diferentes de zero (881764 → 10d 4h 56m) **uptime** - tempo decorrido no formato hh:mm:ss ou N dias, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** - timestamp Unix, formatado como aaaa.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); para uma formatação correta, o valor recebido do item deve ser *Numeric (unsigned)*. Para outras unidades (como Hz, W etc.), se o valor recebido exceder 1000, ele será dividido por 1000 e exibido com o prefixo correspondente (5000 → 5 KHz, 881764 → 881.76 KW). Se o símbolo da unidade for prefixado com `!`, a conversão e a aplicação de prefixos serão desativadas (1024 !B → 1024 B, 61 !s → 61 s). Para obter mais exemplos e detalhes sobre símbolos de unidade e conversão de unidades, consulte Sufixos de valores de itens.

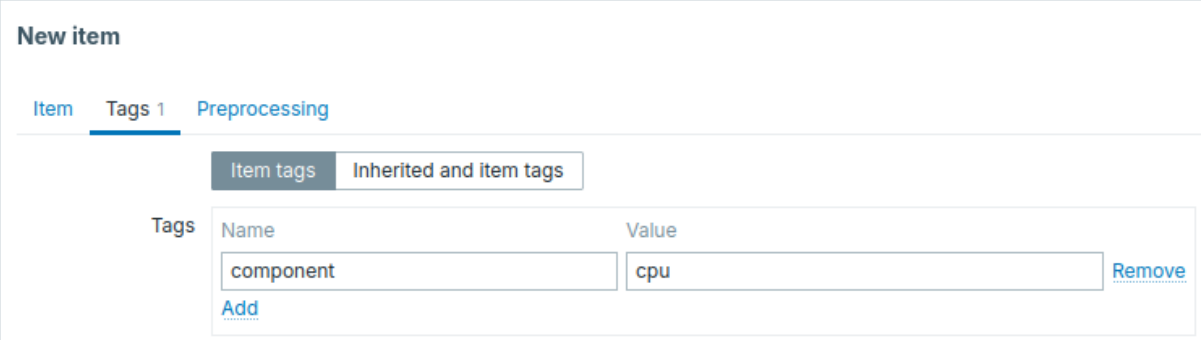
Parâmetro	Descrição
Update interval	Recupere um novo valor para este item a cada N segundos. O intervalo máximo permitido é de 86400 segundos (1 dia). Sufixos de tempo são compatíveis, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são compatíveis. Uma única macro deve preencher todo o campo. Não há suporte para várias macros em um campo nem para macros misturadas com texto. *Observação*: o intervalo de atualização só pode ser definido como '0' se existirem intervalos personalizados com um valor diferente de zero. Se definido como '0' e existir um intervalo personalizado (flexível ou agendado) com um valor diferente de zero, o item será consultado durante a duração do intervalo personalizado. *Observe* que a primeira consulta do item após ele se tornar ativo ou após a alteração do intervalo de atualização pode ocorrer antes do valor configurado. Novos itens serão verificados em até 60 segundos após sua criação, a menos que tenham um intervalo de atualização Agendado ou Flexível e *Update interval* esteja definido como 0. Um item passivo existente pode ser consultado imediatamente para obter seu valor pressionando o botão *Execute now* button.
Custom intervals	Você pode criar regras personalizadas para verificar o item: **Flexible** - cria uma exceção ao *Update interval* (intervalo com frequência diferente). **Scheduling** - cria uma agenda de consulta personalizada. Para obter informações detalhadas, consulte Intervalos personalizados. Sufixos de tempo são compatíveis no campo *Interval*, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são compatíveis. Uma única macro deve preencher todo o campo. Não há suporte para várias macros em um campo nem para macros misturadas com texto.
Timeout	Defina o tempo limite da verificação do item (disponível para tipos de item compatíveis). Selecione a opção de tempo limite: **Global** - o tempo limite do proxy/global é usado (exibido no campo *Timeout* desativado). **Override** - um tempo limite personalizado é usado (definido no campo *Timeout*; intervalo permitido: 1 - 600s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são compatíveis. Clicar no link *Timeouts* permite configurar os tempos limite do proxy ou os tempos limite globais (se um proxy não for usado). Observe que o link *Timeouts* fica visível somente para usuários do tipo *Super admin* com permissões para as seções do frontend *Administration* > *General* ou *Administration* > *Proxies*.
History	Selecione uma das opções: **Do not store** - o histórico do item não é armazenado. Útil para itens mestre quando somente itens dependentes precisam manter o histórico. Esta configuração não pode ser substituída pelas configurações globais do housekeeper. **Store up to** - especifique por quanto tempo o histórico detalhado será mantido no banco de dados (de 1 hora a 25 anos). Dados mais antigos serão removidos pelo housekeeper. Armazenado em segundos. Sufixos de tempo são compatíveis, por exemplo, 2h, 1d. Macros de usuário são compatíveis. O valor de *Store up to* pode ser substituído globalmente em *Administration* > *Housekeeping*. Se existir uma configuração global de substituição, um ícone de informação laranja será exibido. Ao posicionar o mouse sobre ele, uma mensagem de aviso será exibida, por exemplo, *Overridden by global housekeeper settings (1d)*. É recomendável manter os valores registrados pelo menor tempo possível para reduzir o tamanho do histórico de valores no banco de dados. Em vez de armazenar um histórico longo de valores, você pode armazenar dados de tendências por mais tempo. Consulte também Histórico e tendências.

Parâmetro	Descrição
Trends	Selecione uma das opções: **Do not store** - as tendências não são armazenadas. Esta configuração não pode ser substituída pelas configurações globais do housekeeper. **Store up to** - especifique por quanto tempo o histórico agregado (mínimo, máximo, média e contagem por hora) será mantido no banco de dados (de 1 dia a 25 anos). Dados mais antigos serão removidos pelo housekeeper. Armazenado em segundos. Sufixos de tempo são compatíveis, por exemplo, 24h, 1d. Macros de usuário são compatíveis. O valor de *Store up to* pode ser substituído globalmente em *Administration* > *Housekeeping*. Se existir uma configuração global de substituição, um ícone de informação laranja será exibido. Ao posicionar o mouse sobre ele, uma mensagem de aviso será exibida, por exemplo, *Overridden by global housekeeper settings (7d)*. *Observação:* a manutenção de tendências não está disponível para dados não numéricos — character, log e text. Consulte também Histórico e tendências.
Value mapping	Aplique o mapeamento de valores a este item. O mapeamento de valores não altera os valores recebidos; ele serve apenas para exibir os dados. Funciona com itens *Numeric(unsigned)*, *Numeric(float)* e *Character*. Por exemplo, ”Windows service states”.
Log time format	Disponível somente para itens do tipo **Log**. Placeholders compatíveis: **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 no horário Unix, representando 1º de janeiro de 1970. Por exemplo, considere a seguinte linha do arquivo de log do Zabbix agent: ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” Ela começa com seis posições de caracteres para o PID, seguidas pela data, hora e o restante da mensagem. O formato de hora do log para essa linha seria ”ppppp:yyyyMMdd:hhmmss”. Observe que os caracteres ”p” e ”:” são placeholders e podem ser quaisquer caracteres, exceto ”yMdhms”.
Populates host inventory field	Você pode selecionar um campo do inventário do host que será preenchido pelo valor do item. Isso funcionará se o preenchimento automático do inventário estiver habilitado para o host e com itens *Numeric(unsigned)*, *Numeric(float)*, *Character* e *Text*.
Description	Insira uma descrição para o item. Macros de usuário são compatíveis.
Enabled	Marque a caixa de seleção para habilitar o item, permitindo que ele seja processado. Quando você desabilita o item, ele é removido imediatamente do cache de histórico (exceto pelo último valor, que é mantido para logs).
Latest data	Clique no link para visualizar os dados mais recentes do item. Este link está disponível somente ao editar um item já existente.

Note:
Os campos específicos do tipo de item são descritos nas páginas correspondentes.

Note:
Ao editar no nível do host um item existente no nível do template, vários campos são somente leitura. Você pode usar o link no cabeçalho do formulário para acessar o nível do template e editá-los lá, lembrando que as alterações no nível do template modificarão o item para todos os hosts aos quais o template está vinculado.

A aba **Tags** permite definir tags no nível do item.



Pré-processamento do valor do item

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

Teste de item

Attention:
Para realizar o teste de item, certifique-se de que a hora do sistema no server e no proxy esteja sincronizada. No caso de a hora do server estar atrasada, o teste de item pode retornar a mensagem de erro "The task has been expired." No entanto, definir fusos horários diferentes no server e no proxy não afetará o resultado do teste.

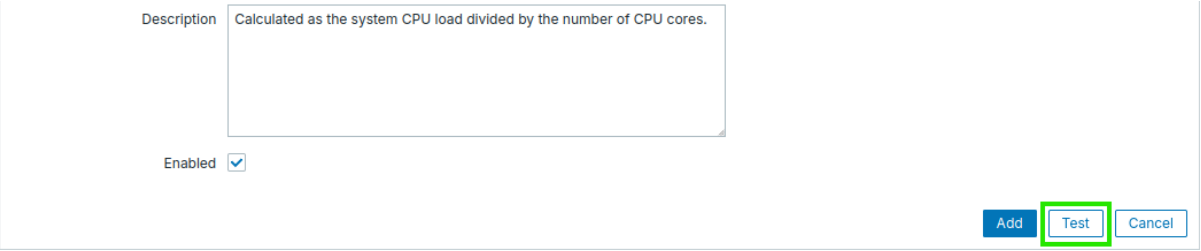
É possível testar um item e, se estiver configurado corretamente, obter um valor real como retorno. O teste pode ocorrer até mesmo antes de um item ser salvo.

O teste está disponível para itens de host e template, protótipos de item e regras de descoberta em baixo nível. O teste não está disponível para itens ativos.

O teste de item está disponível para os seguintes tipos de item passivos:

- Zabbix agent
- SNMP agent (v1, v2, v3)
- IPMI agent
- SSH checks
- Telnet checks
- JMX agent
- Simple checks (exceto itens `icmpping*`, `vmware.*`)
- Zabbix internal
- Calculated items
- External checks
- Database monitor
- HTTP agent
- Script
- Browser

Para testar um item, clique no botão *Test* na parte inferior do formulário de configuração do item. Observe que o botão *Test* ficará desativado para itens que não podem ser testados (como verificações ativas, simple checks excluídos).



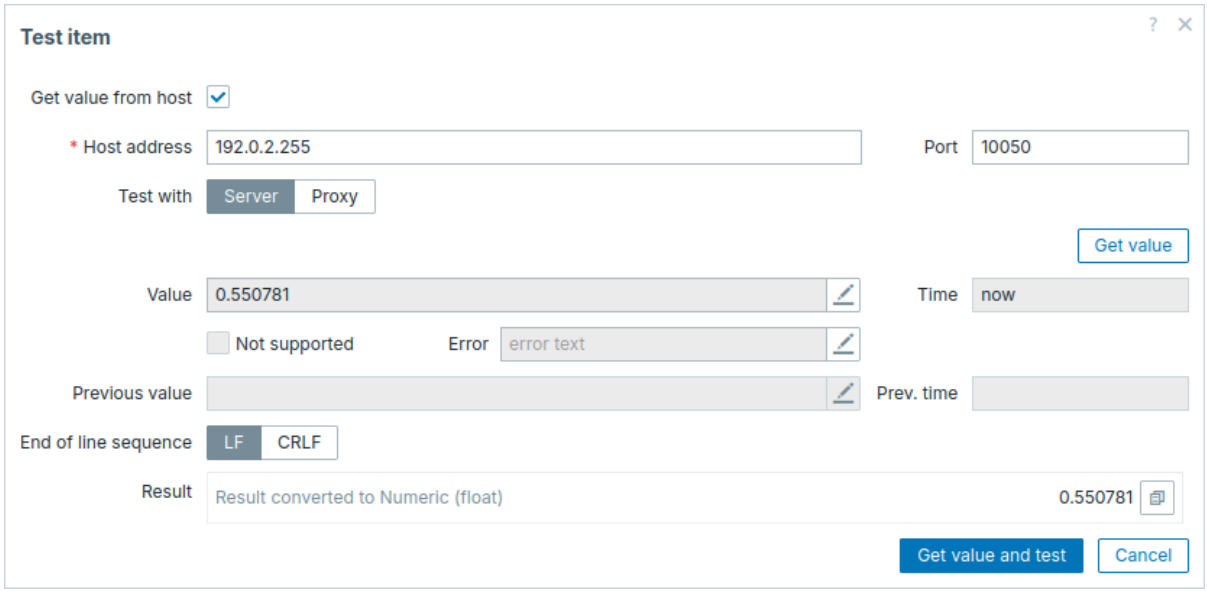
O formulário de teste de item possui campos para os parâmetros obrigatórios do host (endereço do host, porta, testar com server/proxy (nome do proxy)) e detalhes específicos do item (como comunidade SNMPv2 ou credenciais de segurança SNMPv3). Esses campos são sensíveis ao contexto:

- Os valores são preenchidos previamente quando possível, ou seja, para itens que exigem um agent, usando as informações da interface de agent selecionada do host.
- Os valores precisam ser preenchidos manualmente para itens de template.
- Valores de macros em texto simples são resolvidos.

- Campos em que o valor (ou parte do valor) seja um segredo ou uma macro Vault ficam vazios e precisam ser informados 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 desativados quando não são necessários no contexto do tipo de item (por exemplo, o campo de endereço do host e o campo de proxy são desativados para calculated items)

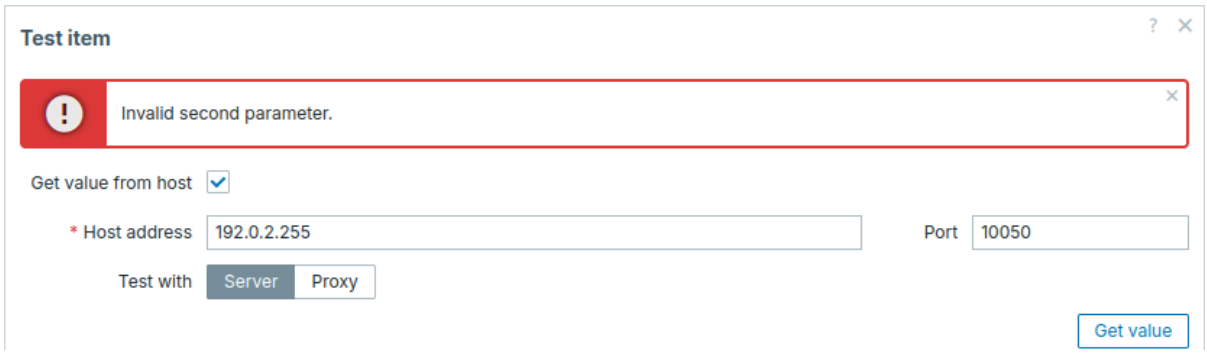
Para testar o item, clique em *Get value*. Se o valor for recuperado com sucesso, ele preencherá o campo *Value*, movendo o valor atual (se houver) para o campo *Previous value* e também calculando o campo *Prev. time*, ou seja, a diferença de tempo entre os dois valores (cliques), além de tentar detectar uma sequência EOL e alternar para CRLF se detectar "\n\r" no valor recuperado.

Clique em *Get value and test* para testar o preprocessing.



Os valores recuperados de um host e os resultados do teste são truncados para um tamanho máximo de 512KB quando enviados ao frontend. Se um resultado de teste for truncado, um ícone de aviso será exibido; passe o mouse sobre ele para ver os detalhes. Se um valor for truncado, ele poderá falhar na validação do tipo de dado em testes consecutivos que comparem valores grandes (por exemplo, JSON); nesses casos, uma mensagem de erro será exibida. Observe que dados maiores que 512KB ainda são processados integralmente pelo Zabbix server.

Se a configuração estiver incorreta, será exibida uma mensagem de erro descrevendo a possível causa.



Um valor recuperado com sucesso de um host também pode ser usado para testar etapas de preprocessing.

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 itens.
Update	Atualizar as propriedades de um item.
Clone	Criar outro item com base nas propriedades do item atual.
Execute now	Executar imediatamente uma verificação para obter um novo valor do item. Compatível apenas com verificações **passivas** (consulte mais detalhes). *Observação*: 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 do item

Os limites de dados do item dependem do tipo de dado do item e do backend do banco de dados.

Valores numéricos (unsigned) são armazenados como estão. Se um valor de ponto flutuante for recebido, apenas a parte inteira será armazenada (por exemplo, 1.23 → 1).

Valores numéricos (float) são armazenados como estão e suportam aproximadamente 15 a 17 dígitos de precisão, com um intervalo de aproximadamente -1.79E+308 a 1.79E+308. Valores em notação científica também são suportados (por exemplo, 1.23E+7, 1e308, 1.1E-4).

Valores de texto são truncados antes de serem armazenados para corresponder ao limite do tipo de valor do banco de dados:

Database	Type of information		
	Character	**Log**	**Text**
MySQL	255 characters	65536 bytes	65536 bytes
PostgreSQL	255 characters	65536 characters	65536 characters
SQLite (only Zabbix proxy)	255 characters	65536 characters	65536 characters

Valores binários são limitados a 16MiB (16777216 bytes). Valores que excedem esse limite são descartados, e um erro correspondente é exibido no frontend.

Valores com o tipo de dado JSON são limitados a 128MiB (134217728 bytes). Valores que excedem esse limite são descartados, e um erro correspondente é exibido no frontend. Para MySQL/MariaDB, considere configurar partições diárias para a tabela `history_json` (usada para armazenar valores JSON), pois ela pode crescer rapidamente e bloquear o housekeeper por um período prolongado. Se o seu caso de uso envolver valores JSON grandes (1MiB ou mais), veja como configurar seu sistema para suporte a valores JSON grandes.

Quando o Zabbix grava qualquer dado de item no banco de dados, ele usa consultas INSERT e as registra (se `DebugLevel` estiver definido como 4 ou 5). Entradas de log grandes são truncadas para 64KB.

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

Timeout do item

O timeout do item especifica quanto tempo o Zabbix deve aguardar antes de abortar a verificação como falha.

Quando o timeout é atingido, a verificação é abortada mesmo que a obtenção dos dados ainda não tenha sido concluída. Se os dados forem recebidos parcialmente, o item se torna unsupported e uma mensagem de erro é registrada (por exemplo, se os dados forem coletados com sucesso para apenas um de vários OIDs em uma verificação SNMP).

Para muitos tipos de item, você pode definir timeouts de item **flexíveis**:

- por item individual
- por tipo de item (no nível de proxy)
- por tipo de item (no nível global global)

Um timeout personalizado *por item individual* é útil se você quiser especificar um timeout maior para um item específico, mantendo timeouts baixos para os outros itens.

Precedência do timeout

1. O timeout de item individual substitui qualquer outro timeout.
2. Timeouts no nível de proxy substituem timeouts globais.

O timeout da configuração do server ou do proxy do Zabbix **não** tem papel nas verificações para as quais timeouts flexíveis estão definidos.

Suporte a timeout flexível

Timeouts flexíveis de item são suportados para estes tipos de item:

- Zabbix agent (tanto verificações passivas quanto ativas)
- SNMP agent (exceto verificações SNMP legadas[1])
- Simple check (exceto `icmpping*` e itens VMware[2])
- SSH agent
- Telnet agent
- External check
- Database monitor
- Script
- HTTP agent
- Browser

[1] Para verificações SNMP legadas (polling de OID único), aplicam-se as configurações de timeout do server ou do proxy.
 [2] Para itens `icmpping*`, o valor de timeout é especificado diretamente na chave do item; além disso, aplica-se um timeout máximo fixo de 600 segundos. Os itens de monitoramento VMware usam seu próprio parâmetro VMwareTimeout da configuração do server ou do proxy.

Items não suportados

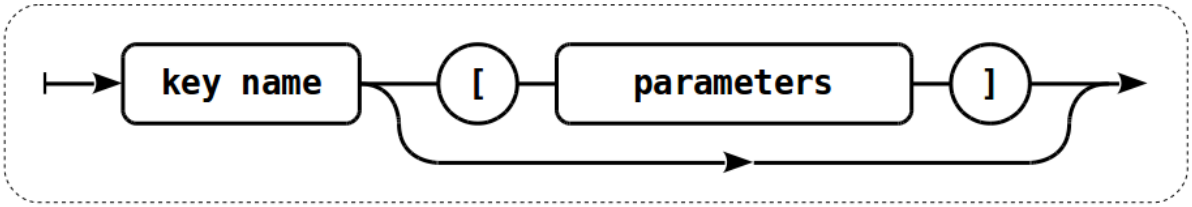
Itens não suportados são relatados como tendo o status *Não suportado* e ainda são verificados novamente no seu *Intervalo de atualização* padrão.

Um item se torna não suportado se seu valor não puder ser recuperado por algum motivo (por exemplo, erro de conexão, nenhum poller configurado para processar o item). Além disso, itens que não recebem dados históricos permanecem no status *Não suportado*. Este é o comportamento esperado—um item muda para o status *Normal* somente quando recebe novos dados históricos válidos.

Assim como os problemas, os itens não suportados são reavaliados apenas quando novos dados são recebidos—mesmo que não haja mais dados históricos disponíveis para esse item. Em outras palavras, itens e triggers mudam seu estado apenas ao receber novos dados. No entanto, se uma expressão de trigger incluir uma função de data e hora e/ou nodata(), a trigger será recalculada a cada 30 segundos (veja tempo de cálculo das triggers para detalhes).

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; em seguida, há a opção de incluir parâmetros ou não - conforme mostrado pelas duas linhas que podem ser seguidas.

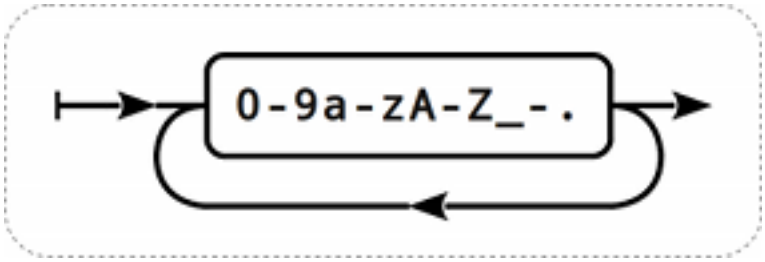
Nome da chave

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.

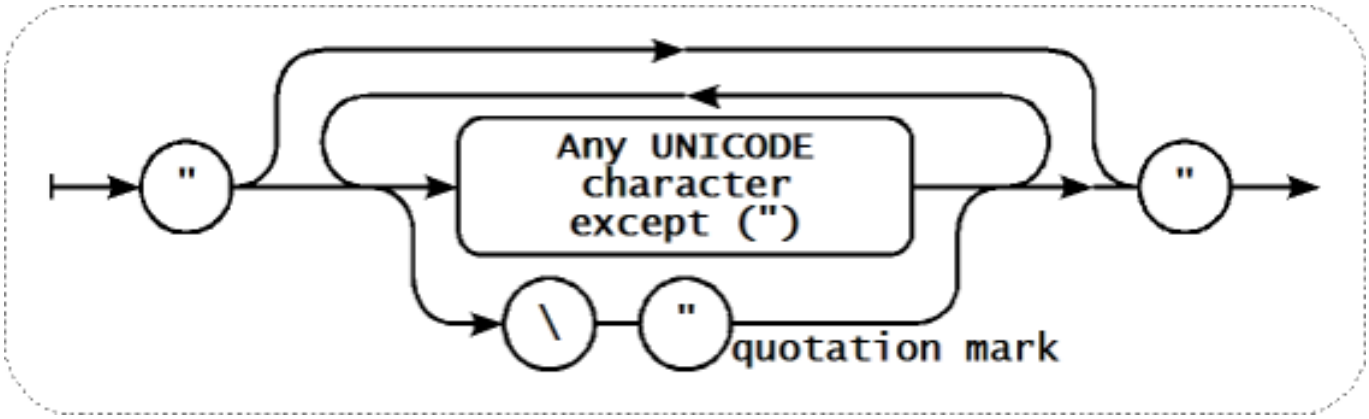
Cada parâmetro de chave pode ser uma string entre aspas, uma string sem aspas ou um array.

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.

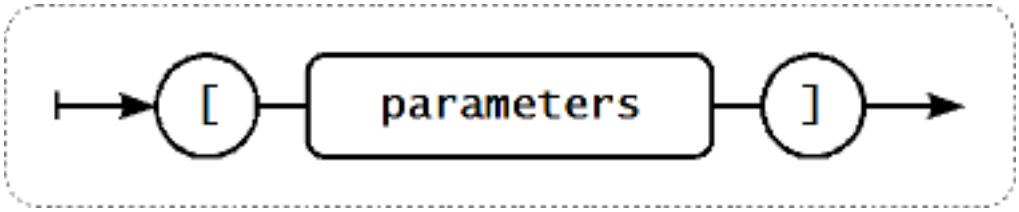
Attention:
Arrays de parâmetros multinível, por exemplo, `[a,[b,[c,d]],e]`, não são permitidos.

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.



Intervalos personalizados

Visão geral

É possível criar regras personalizadas relacionadas 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 em uma sequência de horários.

Intervalos flexíveis

Os intervalos flexíveis permitem redefinir o intervalo de atualização padrão para períodos específicos. Um intervalo flexível é definido com *Interval* e *Period*, em que:

- *Interval* - o intervalo de atualização para o período especificado. Os sufixos de tempo são compatíveis, por exemplo, 30s, 1m, 2h, 1d.
- *Period* - o período em que o intervalo flexível está ativo (consulte os períodos de tempo para obter 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 será usado.

Observe que, se o intervalo flexível for igual à duração do período, o item será verificado exatamente uma vez. Se o intervalo flexível for maior que o período, o item poderá ser verificado uma vez ou poderá não ser verificado (portanto, essa 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 a coleta será retomada de acordo com o *Update interval* padrão assim que o período terminar. Exemplos:

Interval	Period	Description
10	1-5,09:00-18:00	O item será verificado a cada 10 segundos durante o horário de trabalho.
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 diariamente às 12:00. Observe que isso era usado como uma solução alternativa para verificações agendadas, sendo recomendado usar intervalos de agendamento para essas verificações.

Intervalos de agendamento

Os intervalos de agendamento são usados para verificar itens 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 agenda 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, então o filtro corresponderá a um intervalo `<from> - <from>`. Se `<from>` também for omitido, então 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 correspondem.

Embora as definições de filtro sejam opcionais, pelo menos um filtro deve ser usado. Um filtro deve ter um intervalo ou o valor de *<step>* definido.

Um filtro vazio corresponde a '0' se nenhum filtro de nível inferior estiver definido, ou a todos os valores possíveis caso contrário. Por exemplo, se o filtro de hora for omitido, então apenas a hora '0' corresponderá, desde que os filtros de minutos e segundos também sejam omitidos; caso contrário, um filtro de hora vazio corresponderá a todos os valores de hora.

Os valores válidos de `<from>` e `<to>` para seus respectivos prefixos de filtro são:

Prefixo	Descrição	*<from>*	*<to>*
md	Dias do mês	1-31	1-31
wd	Dias da semana	1-7	1-7
h	Horas	0-23	0-23
m	Minutos	0-59	0-59
s	Segundos	0-59	0-59

O valor `<from>` deve ser menor ou igual ao valor `<to>`. O valor `<step>` deve ser maior ou igual a 1 e menor ou igual a `<to> - <from>`.

Valores de um único dígito para dias do mês, horas, minutos e segundos podem ser prefixados com 0. Por exemplo, `md01-31` e `h/02` são intervalos válidos, mas `md01-031` e `wd01-07` não são.

No Zabbix frontend, vários intervalos de agendamento são inseridos em linhas separadas. Na API do Zabbix, eles são concatenados em uma única string com 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 9:00.

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 or m/30	de hora em hora em hh:00 e hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	a cada cinco minutos
wd1-5h9	toda segunda a sexta às 9:00
wd1-5h9-18	toda segunda a sexta às 9:00,10:00,...,18:00
h9,10,11 or h9-11	todos os dias às 9:00, 10:00 e 11:00
md1h9m30	todo dia 1 de cada mês às 9:30
md1wd1h9m30	todo dia 1 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 isto 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"
```

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 executadas pelo Zabbix server ou proxy (se os itens forem monitorados por proxy).

Esse recurso oferece suporte a vários casos de uso, como:

- converter bytes em bits (por exemplo, multiplicando os valores de tráfego de rede por ”8”);
- calcular estatísticas por segundo para valores que aumentam incrementalmente;
- aplicar expressões regulares para extrair ou modificar valores;
- executar scripts personalizados nos valores;
- descartar valores inalterados para otimizar o armazenamento no banco de dados.

Uma ou mais etapas de pré-processamento podem ser configuradas para um item. Essas etapas são executadas na ordem em que são configuradas.

Attention:
Se uma etapa de pré-processamento falhar, um item se tornará não suportado. Isso pode ser evitado com o tratamento de erros *Custom on fail* (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 o recebimento de um valor de log do agent.

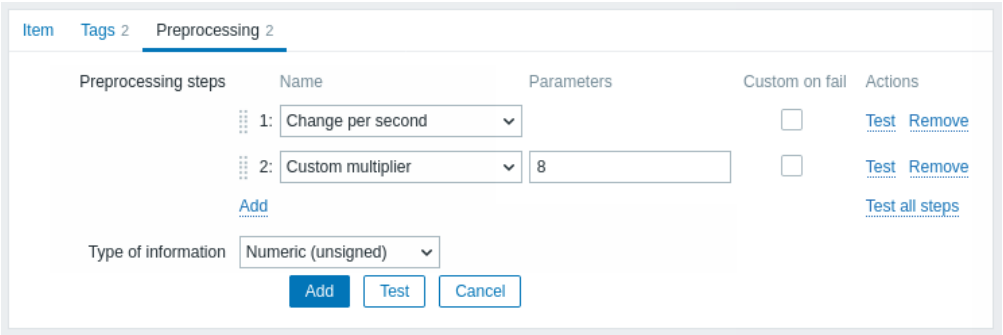
Todos os valores passados para o pré-processamento são inicialmente tratados como strings. A conversão para o tipo de valor desejado (conforme definido na configuração do item) é realizada ao final do pipeline de pré-processamento. No entanto, etapas específicas de pré-processamento podem acionar conversões mais cedo, 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 aba **Pré-processamento** do formulário de configuração do item.



Clique em *Adicionar* para selecionar uma transformação compatível.

O campo *Tipo de informação* é exibido na parte inferior da aba quando pelo menos uma etapa de pré-processamento é definida. Se necessário, é possível alterar o tipo de informação sem sair da aba *Pré-processamento*. Consulte Criando um item para obter uma descrição detalhada dos parâmetros.

Transformações compatíveis

Todas as transformações compatíveis estão listadas abaixo.
Clique no nome da transformação para ver todos os detalhes sobre ela.

Nome	Descrição	Tipo
Expressão regular	Compara o valor com a expressão regular e substitui pelo resultado necessário.	Texto
Substituir	Localiza a string de pesquisa e a substitui por outra (ou por nada).	
Remover espaços	Remove os caracteres especificados do início e do fim do valor.	
Remover espaços à direita	Remove os caracteres especificados do fim do valor.	

Nome	Descrição	Tipo
Remover espaços à esquerda	Remove os caracteres especificados do início do valor.	
XML XPath	Extrai um valor ou fragmento de dados XML usando a funcionalidade XPath.	Dados estruturados
JSON Path	Extrai um valor ou fragmento de JSON usando a funcionalidade JSONPath.	
CSV para JSON	Converte dados de um arquivo CSV para o formato JSON.	
XML para JSON	Converte dados no formato XML para JSON.	
Valor de SNMP walk	Extrai um valor pelo nome de OID/MIB especificado e aplica opções de formatação.	SNMP
SNMP walk para JSON	Converte valores SNMP para JSON.	
Valor de SNMP get	Aplica opções de formatação ao valor de SNMP get.	
Multiplicador personalizado	Multiplica o valor pelo valor inteiro ou de ponto flutuante especificado.	Aritmética
Alteração simples	Calcula a diferença entre o valor atual e o anterior.	Alteração
Alteração por segundo	Calcula a velocidade da alteração 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 código JavaScript.	Scripts personalizados
No intervalo	Define um intervalo no qual o valor deve estar.	Validação
Corresponde à expressão regular	Especifica uma expressão regular à qual o valor deve corresponder.	
Não corresponde à expressão regular	Especifica uma expressão regular à qual o valor não deve corresponder.	
Verificar erro em JSON	Verifica uma mensagem de erro no nível da aplicação localizada em JSONPath.	
Verificar erro em XML	Verifica uma mensagem de erro no nível da aplicação localizada em XPath.	
Verificar erro usando uma expressão regular	Verifica uma mensagem de erro no nível da aplicação usando uma expressão regular.	
Verificar valor não compatível	Verifica se nenhum valor do item pôde ser recuperado.	
Descartar se inalterado	Descarta um valor se ele não tiver sido alterado.	Limitação
Descartar se inalterado com heartbeat	Descarta um valor se ele não tiver sido alterado durante o período definido.	
Padrão do Prometheus	Usa a consulta a seguir para extrair os dados necessários das métricas do Prometheus.	Prometheus
Prometheus para JSON	Converte as métricas necessárias do Prometheus para JSON.	

Se a caixa de seleção *Personalizado em caso de falha* estiver habilitada para a transformação selecionada, você poderá marcá-la e especificar opções personalizadas de tratamento de erros: *Descartar valor* (descartar o valor), *Definir valor como* (definir um valor especificado) ou *Definir erro como* (definir uma mensagem de erro especificada).
Se uma etapa de pré-processamento falhar, o item não se tornará não compatível quando a opção *Descartar valor* ou *Definir valor como* estiver selecionada.

Observe que, para as etapas de pré-processamento *Alteração* e *Limitação*, o Zabbix precisa memorizar o último valor para calcular/comparar o novo valor conforme necessário.
Esses valores anteriores são gerenciados pelo gerenciador de pré-processamento.
Se o Zabbix server ou o proxy for reiniciado, ou se houver qualquer alteração nas etapas de pré-processamento, o último valor do item correspondente será redefinido, resultando no seguinte:

- para as etapas *Alteração simples* e *Alteração por segundo*: o próximo valor será ignorado porque não há um valor anterior a partir do qual calcular a alteração;
- para as etapas *Descartar se inalterado* e *Descartar se inalterado com heartbeat*: o próximo valor nunca será descartado, mesmo que devesse ser descartado de acordo com as regras de descarte.

Expressão regular

Corresponde o valor à expressão regular e substitui-o pela saída necessária.

Parâmetros:

- **pattern** - a expressão regular;

- **output** - o template de formatação da 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 fará com que o item fique sem suporte;
- A expressão regular permite a extração de no máximo 10 grupos capturados com a sequência \N;
- Consulte a seção expressões regulares para ver alguns exemplos existentes.

Substituir

Encontre a string de pesquisa e substitua-a por outra (ou por nada).

Parâmetros:

- **string de pesquisa** - a string a ser encontrada e substituída, diferencia maiúsculas de minúsculas (obrigatório);
- **substituição** - a string pela qual a string de pesquisa será substituída. A string de substituição também pode estar vazia, permitindo efetivamente que a string de pesquisa seja excluída 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, retornos 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, retornos 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

Extraia um valor ou fragmento de dados XML usando a funcionalidade XPath.

Comentários:

- Para que esta opção funcione, o Zabbix server (ou o Zabbix proxy) deve ser compilado com suporte à libxml;
- Namespaces não são compatíveis.

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></doc
```

JSON Path

Extraia um valor ou fragmento de dados JSON usando a funcionalidade JSONPath.

CSV para JSON

Converta dados de arquivos CSV para o formato JSON.

Para mais informações, consulte: pré-processamento de CSV para JSON.

XML para JSON

Converta dados no formato XML para JSON.

Para obter mais informações, consulte: Regras de serialização.

Valor de SNMP walk

Extraia o valor pelo OID/nome MIB especificado e aplique as opções de formatação:

- **Inalterado** - retorna a sequência hexadecimal como uma sequência hexadecimal não escapada (*observe* que as dicas de exibição ainda são aplicadas);
- **UTF-8 de hex-STRING** - converte a sequência hexadecimal em uma sequência UTF-8;
- **MAC de hex-STRING** - valida a sequência hexadecimal como um endereço MAC e retorna uma sequência de endereço MAC adequada (em que `' '` é substituído por `':'`);

- **Inteiro de BITS** - converte os primeiros 8 bytes de uma sequência de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, ”1A 2B 3C 4D”) em um inteiro sem sinal de 64 bits. Em sequências de bits com mais de 8 bytes, os bytes subsequentes serão ignorados.

SNMP walk para JSON

Converta valores SNMP em JSON.

Especifique um nome de campo no JSON e o caminho OID SNMP correspondente. Os valores dos campos serão preenchidos com os valores do caminho OID SNMP especificado.

Comentários:

- Estão disponíveis opções de formatação de valores semelhantes às da etapa *Valor do SNMP walk*;
- Você pode usar esta etapa de pré-processamento para descoberta de OID SNMP.

Obter valor SNMP

Aplique opções de formatação ao valor obtido via SNMP:

- **UTF-8 de Hex-STRING** - converte uma sequência hexadecimal em uma sequência UTF-8;
- **MAC de Hex-STRING** - valida a sequência hexadecimal como um endereço MAC e retorna uma sequência de endereço MAC adequada (em que `' '` é substituído por `':'`);
- **Inteiro de BITS** - converte os primeiros 8 bytes de uma sequência de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, ”1A 2B 3C 4D”) em um inteiro sem sinal de 64 bits. Em sequências de bits com mais de 8 bytes, os bytes subsequentes serão ignorados.

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. para 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 *Numeric (unsigned)*, 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 *Change per second* para itens cujo tipo de informação esteja definido como *Numeric (unsigned)* e o valor calculado resultante for, na verdade, um número de ponto flutuante, o valor calculado ainda será aceito como correto, com a parte decimal sendo truncada e o valor armazenado como um inteiro;
- Compatível com: 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.

Alteração simples

Calcule a diferença entre o valor atual e o valor anterior.

Comentários:

- Esta etapa pode ser útil para medir um valor que cresce constantemente;
- Avaliada como **value**-**prev_value**, em que *value* é o valor atual; *prev_value* é o valor recebido anteriormente;
- É permitida apenas uma operação de alteração por item (”Alteração simples” ou ”Alteração por segundo”);
- Se o valor atual for menor que o valor anterior, o Zabbix descarta essa diferença (não armazena nada) e aguarda outro valor.

Alteração por segundo

Calcule 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 (ponto flutuante)*, mesmo que os valores brutos recebidos sejam inteiros. Isso é especialmente relevante para números pequenos, nos quais a parte decimal é importante. Se os valores de ponto flutuante forem grandes e puderem exceder o comprimento do campo ’float’, fazendo com que o valor inteiro seja perdido, é recomendável usar *Numérico (sem sinal)* e, assim, remover apenas a parte decimal.
- Avaliado como (**value**-**prev_value**)/(**time**-**prev_time**), em que *value* - o valor atual; *prev_value* - o valor recebido anteriormente; *time* - o timestamp atual; *prev_time* - 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 descartará essa diferença (não armazenará nada) e aguardará outro valor. Isso ajuda a funcionar corretamente com, por exemplo, um contador SNMP de 32 bits que sofreu um rollover (estouro).

Booleano para decimal

Converte o valor do formato booleano para decimal.

Comentários:

- A representação textual é convertida para 0 ou 1. Assim, 'TRUE' é armazenado como 1 e 'FALSE' é armazenado como 0. Todos os valores são correspondidos sem distinção entre maiúsculas e minúsculas. 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.

Octal para decimal

Converta o valor do formato octal para decimal.

Hexadecimal para decimal

Converte o valor do formato hexadecimal para decimal.

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 no qual um valor deve estar especificando valores mínimo/máximo (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.

Corresponde à expressão regular

Especifique uma expressão regular à qual um valor deve corresponder.

Não corresponde à expressão regular

Especifique uma expressão regular à qual um valor não deve corresponder.

Verificar erro no JSON

Verifique se há uma mensagem de erro no nível da aplicação localizada no JSONPath. Interrompa o processamento se a operação for bem-sucedida e a mensagem não estiver vazia; caso contrário, continue o processamento com o valor que estava disponível antes desta etapa de pré-processamento.

Comentários:

- Esses erros de serviços externos são relatados ao usuário como estão, sem adicionar informações sobre a etapa de pré-processamento;
- Nenhum erro será relatado caso não seja possível analisar um JSON inválido.

Verificar se há um erro no XML

Verifique se há uma mensagem de erro no nível da aplicação localizada no XPath.
Interrompa o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continue o processamento com o valor que estava presente antes desta etapa de pré-processamento.

Comentários:

- Esses erros de serviços externos são relatados ao usuário como estão, sem adicionar informações sobre a etapa de pré-processamento;
- Nenhum erro será relatado caso não seja possível analisar um XML inválido.

Verificar erros usando uma expressão regular

Verifique uma mensagem de erro no nível da aplicação usando uma expressão regular. Interrompa o processamento se houver correspondência e a mensagem não estiver vazia; caso contrário, continue o processamento com o valor anterior a esta etapa de pré-processamento.

Parâmetros:

- **pattern** - a expressão regular;
- **output** - o template de formatação da 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 sobre a etapa de pré-processamento.

Verificar valor não suportado

Verifique se nenhum valor do item pôde ser obtido. Especifique como a falha deve ser processada, com base na inspeção da mensagem de erro retornada.

Parâmetros:

- **scope** - selecione o escopo do processamento do erro:
any error - qualquer erro;
error matches - somente o erro que corresponde à expressão regular especificada em *pattern*;
error does not match - somente o erro que não corresponde à expressão regular especificada em *pattern*
- **pattern** - a expressão regular usada para corresponder ao erro. Se *any error* estiver selecionado no parâmetro de escopo, este campo não será exibido. Quando exibido, este campo é obrigatório.

Comentários:

- Normalmente, a ausência de um valor ou a falha ao obtê-lo faria com que o item se tornasse não suportado. Esta etapa de pré-processamento permite modificar esse comportamento.
- Esta etapa de pré-processamento verifica somente se nenhum valor do item pôde ser obtido. Ela não verifica, por exemplo, se o tipo do valor obtido (por exemplo, string) corresponde ao tipo de informação do item (por exemplo, numérico); para obter detalhes, consulte Exemplos de pré-processamento. Se houver uma incompatibilidade de tipos, o item ainda poderá se tornar não suportado após a execução de todas as etapas de pré-processamento. Para verificar uma incompatibilidade de tipos, você pode usar, por exemplo, a etapa de pré-processamento *Multiplicador personalizado*; consulte Exemplos de pré-processamento.
- Há suporte para grupos de expressões regulares de captura 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;
- Há suporte para várias etapas *Verificar valor não suportado*, na ordem especificada. Uma etapa para *any error* será colocada automaticamente como a última etapa deste 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 valores inalterados com heartbeat

Descarta um valor se ele não tiver sido alterado dentro do período definido (em segundos).

Comentários:

- São aceitos valores inteiros positivos para especificar os segundos (mínimo de 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 e, consequentemente, nenhum problema para os triggers relacionados será criado ou resolvido. As funções funcionarão somente com base nos dados efetivamente salvos no banco de dados. Como os dados de tendência são criados com base nos dados do banco de dados, se nenhum valor for salvo durante uma hora, também não haverá dados de tendência para essa hora.
- Somente 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 Prometheus checks para mais detalhes.

Prometheus para JSON

Converta as métricas Prometheus necessárias para JSON.

Veja Prometheus checks 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.

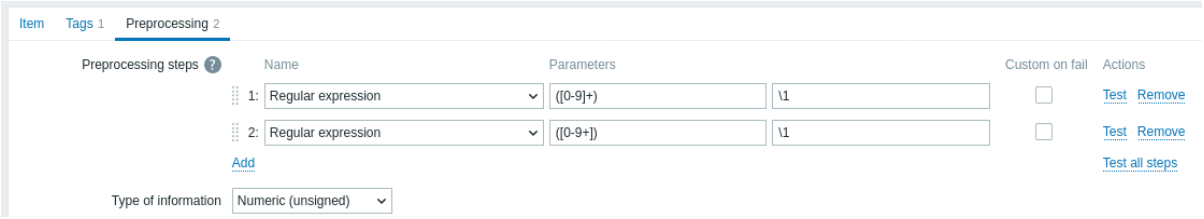
Testando

Veja testando pré-processamento.

Teste de pré-processamento

Teste

Testar 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.

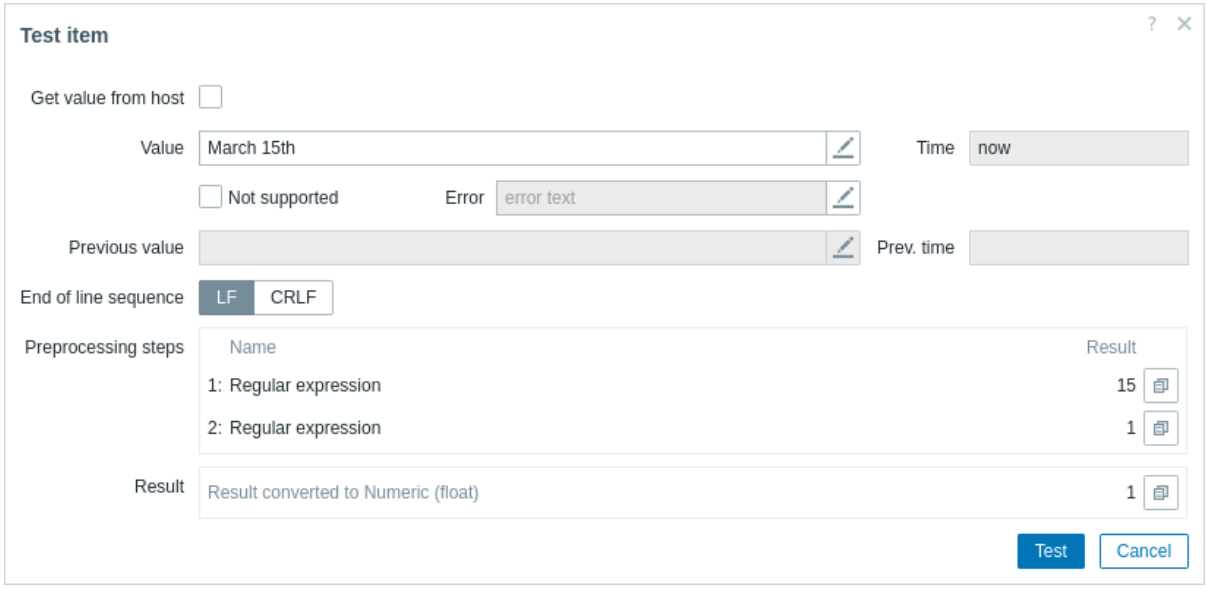


É possível testar:

- com base em um valor hipotético
- com base em um valor real de um host

Cada etapa de pré-processamento pode ser testada individualmente, assim como todas as etapas podem ser testadas em conjunto. Quando você clica no botão *Test* ou *Test all steps*, respectivamente, no bloco Actions, uma janela de teste é aberta.

Testando valor hipotético



Parameter	Description
Get value from host	Se você quiser testar um valor hipotético, deixe esta caixa de seleção desmarcada. Veja também: Testando valor real.
Value	Digite o valor de entrada a ser testado. Clicar no campo do parâmetro ou no botão de visualizar/editar abrirá uma janela de área de texto para inserir o valor ou o bloco de código.
Not supported	Marque esta caixa de seleção para testar um valor não suportado. Esta opção é útil para testar a etapa de pré-processamento *Check for not supported value*.
Error	Digite o texto do erro. Este campo é habilitado quando *Get value from host* está desmarcado, mas *Not supported* está marcado. Se *Get value from host* estiver marcado, este campo será preenchido com a mensagem de erro real (somente leitura) do host.
Time	O horário do valor de entrada é exibido: `now` (somente leitura).
Previous value	Digite um valor de entrada anterior para comparação. Apenas para as etapas de pré-processamento *Change* e *Throttling*.
Previous time	Digite o horário do valor de entrada anterior para comparação. Apenas para as etapas de pré-processamento *Change* e *Throttling*. O valor padrão é baseado no valor do campo 'Update interval' do item (se for '1m', então este campo é preenchido com `now-1m`). Se nada for especificado ou se o usuário não tiver acesso ao host, o padrão será `now-30s`.
Macros	Se forem usadas macros, elas serão listadas junto com seus valores. Os valores podem ser editados para fins de teste, mas as alterações serão salvas apenas no contexto do teste.
End of line sequence	Selecione a sequência de fim de linha para valores de entrada multilinha: **LF** - sequência LF (line feed) **CRLF** - sequência CRLF (carriage-return line-feed).
Preprocessing steps	As etapas de pré-processamento são listadas; o resultado do teste é exibido para cada etapa após clicar no botão *Test*. Os resultados do teste são truncados para um tamanho máximo de 512KB quando enviados ao frontend. Os resultados do teste 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 integralmente 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. No caso de "Custom on fail" ser especificado para a etapa e essa ação ser executada, uma nova linha aparece logo após a linha da etapa de teste de pré-processamento, mostrando qual ação foi realizada e qual resultado ela produziu (erro ou valor).

Parameter	Description
Result	O resultado final do teste das etapas de pré-processamento é exibido em todos os casos em que todas as etapas são testadas juntas (quando você clica no botão *Test all steps*). O tipo de conversão para o tipo de valor do item também é exibido, por exemplo `Result converted to Numeric (unsigned)`. Os resultados do teste são truncados para um tamanho máximo de 512KB quando enviados ao 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 integralmente pelo Zabbix server.

Clique em *Test* para ver o resultado após cada etapa de pré-processamento.

Os valores de teste são armazenados entre sessões de teste, seja para etapas individuais ou para todas as etapas, permitindo que o usuário altere as etapas de pré-processamento ou a configuração do item e depois retorne à janela de teste sem precisar reinsira as informações. No entanto, os valores são perdidos ao atualizar a página.

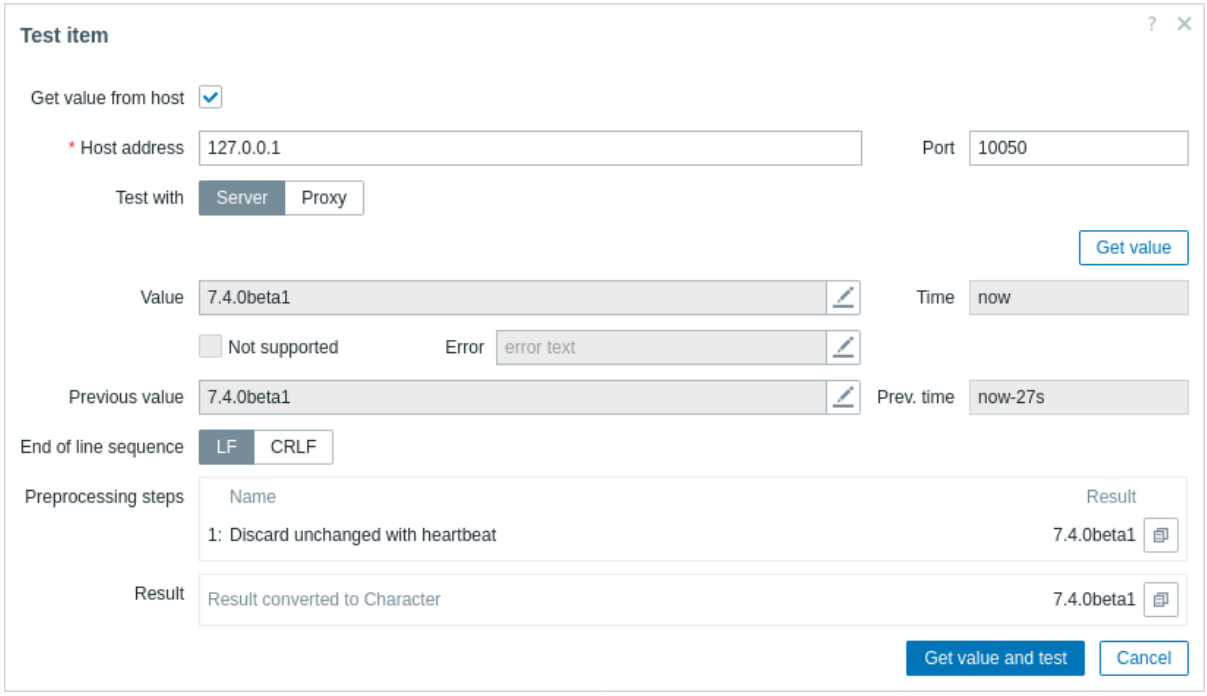
O teste é realizado pelo Zabbix server. O frontend envia uma solicitação correspondente ao server e aguarda o resultado. A solicitação contém o valor de entrada e as etapas de pré-processamento (com macros de usuário expandidas). Para as etapas *Change* e *Throttling*, um valor e horário anteriores opcionais podem ser especificados. O server responde com os resultados de 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 um 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 os detalhes específicos do item (como a comunidade SNMPv2 ou as credenciais de segurança SNMPv3). Esses campos levam o contexto em consideração:
  - Os valores são preenchidos previamente quando possível, ou seja, para itens que exigem um agent, usando as informações da interface de agent selecionada do host
  - Os valores devem ser preenchidos manualmente para itens de template
  - Os valores de macros de texto simples são resolvidos
  - Quando o valor do campo (ou parte dele) é um segredo ou uma macro do Vault, o campo ficará vazio e deverá ser preenchido manualmente. Se algum 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 ficam desabilitados quando não são necessários no contexto do tipo de item (por exemplo, os campos de endereço do host e proxy ficam desabilitados para itens calculados)
- Clique em *Obter valor e testar* para testar o pré-processamento



Se você tiver especificado um mapeamento de valores no formulário de configuração do item (o campo *Mostrar valor*), a caixa de diálogo de teste do item exibirá outra linha após o resultado final, chamada *Resultado com o mapa de valores aplicado*.

Parâmetros específicos para obter um valor real de um host:

Parâmetro	Descrição
Obter valor do host	Marque esta caixa de seleção para obter um valor real do host.
Endereço do host	Insira o endereço do host. Este campo é preenchido automaticamente com o endereço da interface do item no host.
Porta	Insira a porta do host. Este campo é preenchido automaticamente com a porta da interface do item no host.
Campos adicionais para interfaces SNMP (versão SNMP, comunidade SNMP, nome do contexto etc.)	Consulte Configurando o 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 item no host.
Proxy	Especifique o proxy se o host for monitorado por um proxy. Este campo é preenchido automaticamente com o proxy do host (se houver).
Valor	Valor recuperado do host. Clicar no campo do parâmetro ou no botão de visualização/edição abrirá uma janela de área de texto com o valor ou bloco de código. Os valores são truncados para um tamanho máximo de 512 KB somente no frontend. Se um resultado for truncado, um ícone de aviso será exibido. A descrição do aviso será exibida ao passar o mouse sobre o ícone. Observe que os dados maiores que 512 KB ainda são processados integralmente pelo Zabbix server.

Para os demais parâmetros, consulte Testando um valor hipotético acima.

Detalhes de pré-processamento

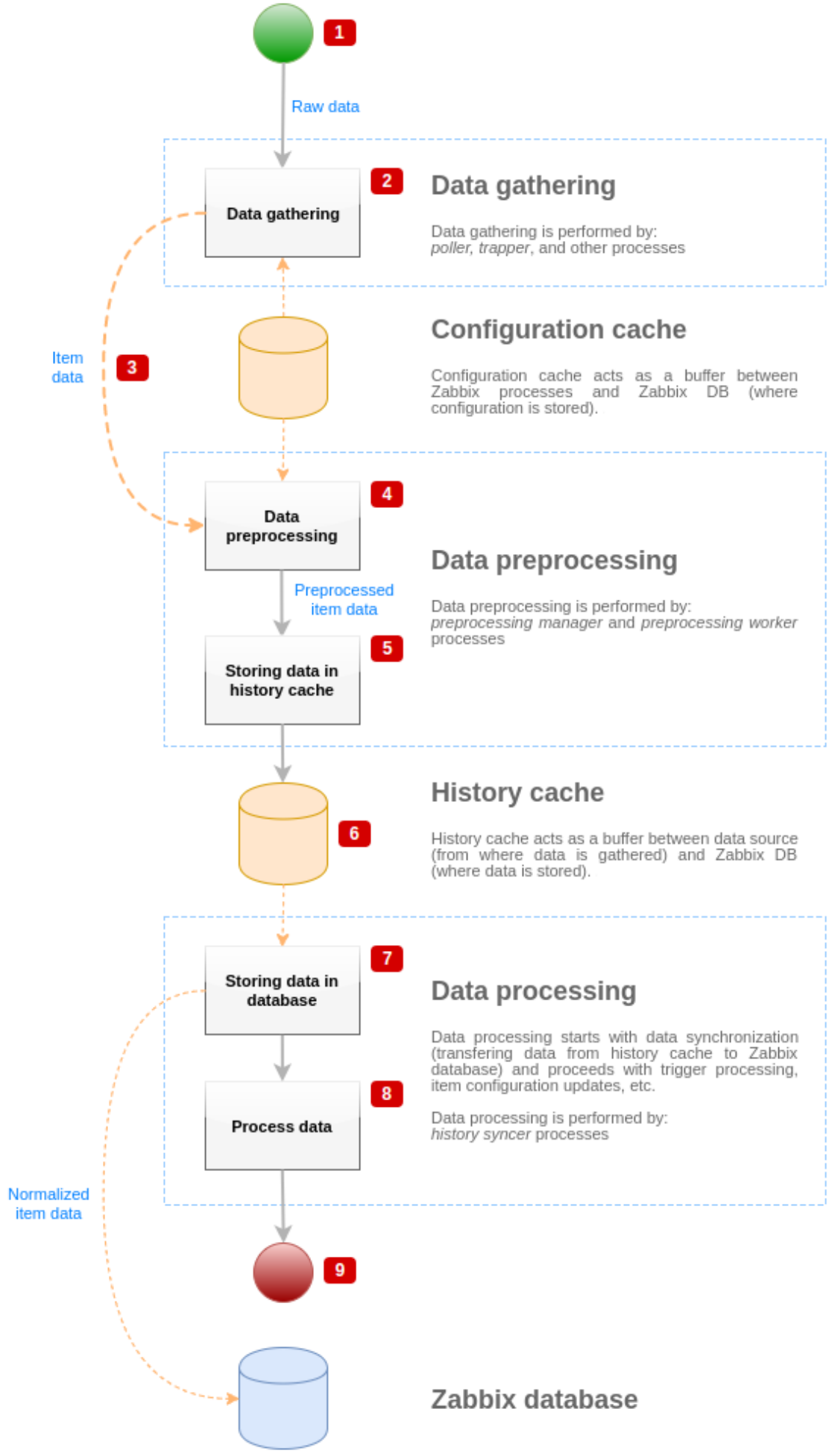
Visão geral

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 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 Zabbix server ou o Zabbix proxy (para os itens monitorados pelo proxy) executa as etapas de pré-processamento.

Processamento do valor do item

Para visualizar o fluxo de dados desde a fonte de dados até o banco de dados do Zabbix, podemos usar o seguinte diagrama simplificado:


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

O diagrama acima mostra apenas processos, objetos e ações relacionadas ao processamento do valor do item de forma **simplificada**. O diagrama não mostra mudanças condicionais de direção, tratamento de erros ou loops. O cache de dados local do gerenciador de pré-processamento também não é mostrado porque não afeta o fluxo de dados diretamente. O objetivo deste diagrama é mostrar os processos envolvidos no processamento do valor do item e a forma como eles interagem.

- A coleta de dados começa com dados brutos de uma fonte de dados. Neste ponto, os dados contêm apenas ID, timestamp e valor (podem ser múltiplos valores também).
- Não importa qual tipo de coletor de dados é usado, a ideia é a mesma para checagens ativas ou passivas, para items trapper, etc., pois isso apenas muda o formato dos dados e o iniciador da comunicação (ou o coletor de dados está esperando por uma conexão e dados, ou o coletor de dados inicia a comunicação e solicita os dados). Os dados brutos são validados, a configuração do item é recuperada do cache de configuração (os dados são enriquecidos com os dados de configuração).
- Um mecanismo IPC baseado em socket é usado para passar dados dos coletores de dados para o gerenciador de pré-processamento. Neste ponto, o coletor de dados continua a coletar dados sem esperar pela resposta do gerenciador de pré-processamento.
- O pré-processamento dos dados é realizado. Isso inclui a execução das etapas de pré-processamento e o processamento de items dependentes.

Note:
Um item pode mudar seu estado para NÃO SUPORTADO enquanto o pré-processamento é realizado se qualquer uma das etapas de pré-processamento falhar.

- Os dados de histórico do cache de dados local 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.

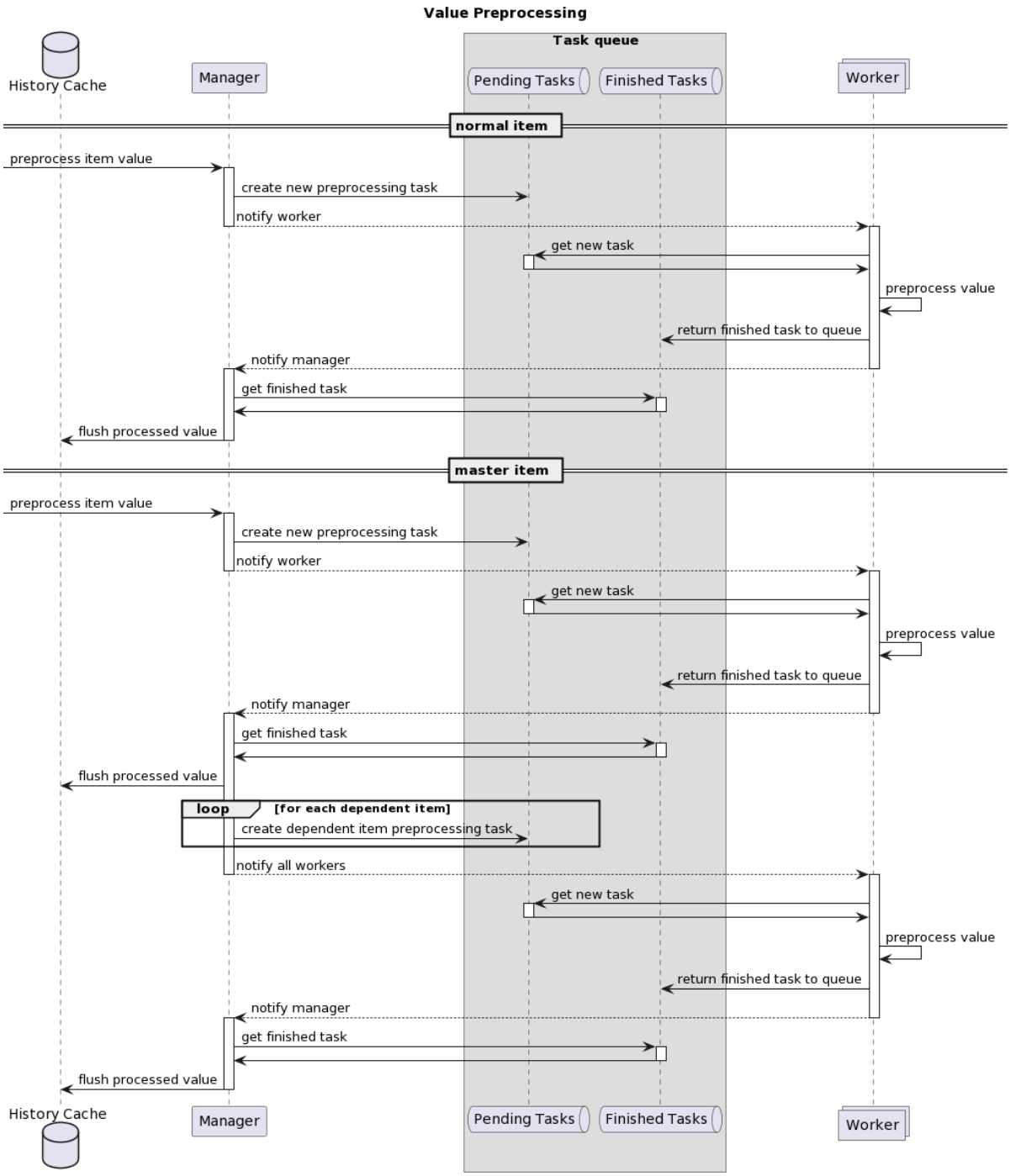
Note:
Um item pode mudar seu estado para NÃO SUPORTADO se a normalização dos dados falhar (por exemplo, quando um valor textual não pode ser convertido para número).

- Os dados coletados estão sendo processados - triggers são verificadas, a configuração do item é atualizada se o item se tornar NÃO SUPORTADO, etc.
- Isso é considerado o fim do fluxo de dados do ponto de vista do processamento 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")]`)

Workers de pré-processamento

O arquivo de configuração do Zabbix server permite que os usuários definam a quantidade de threads de worker de pré-processamento. O parâmetro de configuração StartPreprocessors deve ser usado para definir o número de instâncias de workers de pré-processamento pré-iniciadas, que deve ser pelo menos igual ao número de núcleos de CPU disponíveis.

Se as tarefas de pré-processamento não forem limitadas pela CPU e envolverem solicitações frequentes de rede, é recomendável configurar workers adicionais. O número ideal de workers de pré-processamento pode ser determinado por muitos fatores, incluindo a quantidade de items ”pré-processáveis” (items que exigem a execução de qualquer etapa de pré-processamento), a quantidade de processos de coleta de dados, a média de etapas de pré-processamento por item etc. A quantidade insuficiente de workers pode levar a alto uso de memória. Para solucionar problemas de uso excessivo de memória na sua instalação do Zabbix, consulte Profiling excessive memory usage with tcmalloc.

Mas, assumindo que não haja operações pesadas de pré-processamento, como a análise de grandes blocos XML/JSON, o número de workers de pré-processamento pode corresponder ao número total de coletores de dados. Dessa forma, na maior parte do tempo (exceto nos casos em que os dados do coletor chegam em lote), haverá pelo menos um worker de pré-processamento desocupado para os dados coletados.

Warning:
Processos demais de coleta de dados (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers), juntamente com o IPMI manager, SNMP trapper e workers de pré-processamento, podem esgotar o limite de descritores de arquivo por processo para o preprocessing manager.

 Esgotar o limite de descritores de arquivo por processo fará com que o Zabbix server pare, normalmente logo após a inicialização, mas às vezes levando mais tempo. 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 arquivo esteja definido em um valor 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.

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.

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:

Operand	Description

`"<text>"`
`'<text>'`
	Constante de texto.
	
	Exemplo:
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	Constante numérica com suporte a notação científica. Exemplo: 123
`<jsonpath starting with $>`	Valor referenciado pelo JSONPath a partir do nó raiz do documento de entrada; apenas caminhos definitivos são suportados. Exemplo: `$.object.name`
`<jsonpath starting with @>`	Valor referenciado pelo JSONPath a partir do objeto/elemento atual; apenas caminhos definitivos são suportados. Exemplo: `@.name`

Operadores suportados: |Operator|Type|Description|Result| |-|---|---|---| |−|Binary|Subtração|Number| |+|Binary|Adição|Number| |/|Binary|Divisão|Number| |*|Binary|Multiplicação|Number| |==|Binary|Igualdade|Boolean (1/0)| |!=|Binary|Desigualdade|Boolean (1/0)| |<|Binary|Menor que|Boolean (1/0)| |<=|Binary|Menor ou igual a|Boolean (1/0)| |>|Binary|Maior que|Boolean (1/0)| |>=|Binary|Maior ou igual a|Boolean (1/0)| |=~|Binary|Corresponde à expressão regular|Boolean (1/0)| |!|Unary|NÃO booleano|Boolean (1/0)| ||||Binary|OU booleano|Boolean (1/0)| |&&|Binary|E booleano|Boolean (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

Função	Descrição	Entrada	Saída
`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": "Ditados do Século",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Espada de Honra",
      "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": "O Senhor dos Anéis",
```

```
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Entrega no dia seguinte na cidade local",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Impressão e encadernação de livro no formato A5",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Vários métodos de restauração",
      "active": false,
      "methods": [
        {
          "description": "Limpeza química",
          "price": 46
        },
        {
          "description": "Prensagem de páginas danificadas pela umidade",
          "price": 24.5
        },
        {
          "description": "Reencadernação de livro rasgado",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "A loja está fechada",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }

JSONPath	Type	Result
`$.books[1].title`	definite	Espada de Honra
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Ditados do Século”]
`$.books[1]['author', "title"]`	indefinite	[”Espada de Honra”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Espada de Honra”] Note: Esta consulta mostra que operações aritméticas podem ser usadas em consultas; ela pode ser simplificada para `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Espada de Honra”, ”O Senhor dos Anéis”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Ditados do Século”, ”Moby Dick”, ”O Senhor dos Anéis”]
`$.books[?(@.id != 2)].title`	indefinite	[”Ditados do Século”, ”Moby Dick”, ”O Senhor dos Anéis”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Ditados do Século”, ”Espada de Honra”, ”O Senhor dos Anéis”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”O Senhor dos Anéis”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Ditados do Século”, ”O Senhor dos Anéis”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Espada de Honra”, ”O Senhor dos Anéis”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Espada de Honra”,”Moby Dick”,”O Senhor dos Anéis”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Ditados do Século”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Espada de Honra”
		},
		{
		”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”: ”O Senhor dos Anéis”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Impressão e encadernação de livro no formato A5”, ”Reencadernação de livro rasgado”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Espada de Honra
`$..tags.first().length()`	definite	5 Note: `$..tags` é um caminho indefinido, então ele retorna um array de elementos correspondentes, ou seja, `[["a", "b", "c", "d", "e" ]]`; `first()` retorna o primeiro elemento, ou seja, `["a", "b", "c", "d", "e"]`; `length()` calcula o comprimento do elemento, ou seja,5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Note: Uma consulta sem correspondência retorna NULL para caminhos definidos e indefinidos.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Note: Constantes de texto devem ser usadas em comparações de valores booleanos.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Note: Constantes de texto devem ser usadas em comparações de valores booleanos.
`$.services[?(@.servicegroup==1002)].description`	definite	test

Escape de caracteres especiais dos valores de macro LLD em JSONPath

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

- apenas os caracteres barra invertida (\) e aspas duplas (") são considerados para escape;
- se o valor da macro resolvido contiver esses caracteres, cada um deles é escapado com uma barra invertida;
- se eles já estiverem escapados com uma barra invertida, isso não é considerado como escape e tanto a barra invertida quanto os caracteres especiais seguintes são escapados novamente.

Por exemplo:

JSONPath	LLD macro value	After substitution
$.[?(@.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 conter caracteres especiais deve ser colocada entre aspas duplas:

JSONPath	LLD macro value	After substitution	Result
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Expressão JSONPath inválida**

Quando usada no caminho, a macro que pode conter caracteres especiais deve ser colocada entre colchetes **e** aspas duplas:

JSONPath	LLD macro value	After substitution	Result
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Expressão JSONPath inválida**

Pré-processamento de 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 de JavaScript

O pré-processamento de JavaScript é feito invocando uma função JavaScript com um único parâmetro 'value' e um corpo de 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 de 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 por meio do método toString() (se isso falhar, então o erro é retornado como um valor de string), com algumas exceções:

- retornar um valor undefined resultará em um erro;
- retornar um valor null fará com que o valor de entrada seja descartado, de forma semelhante ao pré-processamento 'Discard value' na ação 'Custom on fail'.

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 um erro. Um limite de heap de 512 megabytes é aplicado.

O bytecode da etapa de pré-processamento de JavaScript é armazenado em cache e reutilizado na próxima vez em que a etapa for aplicada. 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 e superior).

O pré-processamento de JavaScript é implementado com o mecanismo JavaScript Duktape.

> **Note:**
> As definições de métodos de objetos JavaScript integrados não podem ser redefinidas em scripts de pré-processamento de item.

Veja também: Objetos JavaScript adicionais e funções globais

Usando macros em scripts

É possível usar macros de usuário (assim como macros LLD no contexto de descoberta de baixo nível) no 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 precisarão ser inseridos manualmente.

> **Note:**
> O contexto é ignorado quando uma macro é substituída por seu valor. O valor da macro é inserido no código como está, não é possível adicionar escape adicional antes de colocar o valor no código JavaScript. 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 o status do Apache em JSON

Converta o valor recebido de um item Zabbix agent web.page.get (por exemplo, web.page.get[http://127.0.0.1:80/server-status?auto]) em um objeto JSON.

```
// Converter o status do Apache em JSON

// Dividir o valor em substrings e colocar essas substrings em um array
var lines = value.split('\n');
```

```
// Criar um objeto vazio "output"
var output = {};

// Criar 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
};

// Adicionar 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;

// Analisar 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]++;
    }
}

// Adicionar dados dos 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['.']
};

// Retornar a 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
    }
}
```

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 um identificador cURL, permitindo realizar solicitações HTTP simples. Os erros são lançados como exceções.

Attention:
A inicialização de vários objetos `HttpRequest` é limitada a 10 por execução de script.

Método	Descrição
`addHeader(value)`	Adiciona um campo de cabeçalho HTTP. Esse campo é usado em todas as solicitações 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 (incluindo caracteres especiais e nomes de cabeçalho).
`clearHeader()`	Limpa o cabeçalho HTTP. Se nenhum campo de cabeçalho estiver definido, `HttpRequest` definirá Content-Type como application/json se os dados enviados estiverem formatados como JSON; caso contrário, como text/plain.
`connect(url)`	Envia uma solicitação HTTP CONNECT para a URL e retorna a resposta.
`customRequest(method, url, data)`	Permite especificar qualquer método HTTP no primeiro parâmetro. Envia a solicitação do método para a URL com um payload *data* opcional e retorna a resposta.
`delete(url, data)`	Envia uma solicitação HTTP DELETE para a URL com um payload *data* opcional e retorna a resposta.
`getHeaders(<asArray>)`	Retorna o objeto com os campos de cabeçalho HTTP recebidos. O parâmetro `asArray` pode ser definido como "true" (por exemplo, `getHeaders(true)`), "false" ou permanecer 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 de campos de vários cabeçalhos com o mesmo nome. Se não for definido ou for definido como "false", os valores dos campos de cabeçalho HTTP recebidos serão retornados como strings.
`get(url, data)`	Envia uma solicitação HTTP GET para a URL com um payload *data* opcional e retorna a resposta.
`head(url)`	Envia uma solicitação HTTP HEAD para a URL e retorna a resposta.
`options(url)`	Envia uma solicitação HTTP OPTIONS para a URL e retorna a resposta.
`patch(url, data)`	Envia uma solicitação HTTP PATCH para a URL com um payload *data* opcional e retorna a resposta.
`put(url, data)`	Envia uma solicitação HTTP PUT para a URL com um payload *data* opcional e retorna a resposta.
`post(url, data)`	Envia uma solicitação HTTP POST para a URL com um payload *data* opcional e retorna a resposta.
`getStatus()`	Retorna o código de status da última solicitação HTTP.
`setProxy(proxy)`	Define o proxy HTTP como o valor "proxy". Se esse 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'.
	O sinalizador 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 solicitação HTTP TRACE para a URL com um payload *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 um hash HMAC como uma string formatada em hexadecimal; `md5` e `sha256` são suportados como `hash type`; os parâmetros `key` e `data` suportam dados binários.

  Exemplos:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` - retorna a assinatura calculada (assinatura RSA com SHA-256) como uma string, onde:
 hash - apenas `sha256` é permitido, caso contrário, um erro será lançado.
 key - a chave privada. Deve 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 a partir de uma macro de usuário/macro secreta/vault.

  data - os dados que serão assinados. Pode ser uma string (dados binários também são suportados) ou buffer (Uint8Array/ArrayBuffer).

  Exemplo:

  - `sign('sha256',key,data)`

  OpenSSL ou GnuTLS é usado para calcular as assinaturas. Se o Zabbix foi compilado sem nenhuma dessas bibliotecas de criptografia, um erro será lançado ('missing OpenSSL or GnuTLS library').

Objetos JavaScript do item do 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 ao ser criado e encerrando-a ao ser destruído. Um único script pode suportar até quatro objetos `Browser`.

Para construir um objeto `Browser`, use a sintaxe `new Browser(options)`. O parâmetro `options` (*objeto JSON*) especifica as opções do navegador, geralmente o resultado do método de opções do WebDriver (por exemplo, `Browser.chromeOptions()`).

Os seguintes métodos são suportados com o objeto `Browser`.

Método	Descrição
`navigate(url)`	Navega para a URL especificada. Parâmetros: `url` - (string) URL para navegar.
`getUrl()`	Retorna uma string da URL da página aberta.
`getPageSource()`	Retorna uma string do código-fonte da página aberta.
`findElement(strategy, selector)`	Retorna um objeto `Element` com um elemento na página aberta (ou retorna `null` se nenhum elemento corresponder à `strategy` e ao `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.

Método	Descrição
`findElements(strategy, target)`	Retorna um array de objetos `Element` com vários elementos na página aberta (ou retorna um array vazio se nenhum elemento corresponder à estratégia de localização e ao target). 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`.
`addCookie(cookie)`	Define cookies. Parâmetros: `cookie` - (objeto `Cookie`) Cookie a ser definido.
`getScreenshot()`	Retorna uma string codificada em base64 representando uma imagem da viewport do navegador.
`setScreenSize(x,y)`	Define o tamanho da viewport do navegador. Parâmetros: `x` - (string) Largura em pixels; `y` - (string) Altura em pixels.
`setScriptTimeout(timeout)`	Define o tempo limite de carregamento do script. Parâmetros: `timeout` - (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 do `Browser` falhar; contém uma propriedade adicional `browser` com um objeto `Browser` que lançou este `BrowserError`.
- `WebdriverError` - derivado de `BrowserError`; contém as mesmas propriedades do objeto `BrowserError`, que indicam se o erro foi gerado em resposta a um erro na resposta do WebDriver.

Element

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 web e fornece métodos para interagir com ele.

Os seguintes métodos são suportados com o objeto `Element`.

Method	Description
`getAttribute(name)`	Retorna uma string com o valor do atributo do elemento (ou retorna `null` se o atributo especificado não for encontrado). Parameters: `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). Parameters: `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. Parameters: `keys` - (string) Teclas a serem enviadas.

Cookie

O objeto `Cookie` é retornado pelo método `getCookies()` do objeto `Browser` e passado para o método `addCookie()`.

Embora o objeto `Cookie` não tenha nenhum método, ele pode conter as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cookie.
value	string	Valor do cookie.
path	string	Caminho para o qual o cookie é válido. O padrão é `"/"` se omitido ao adicionar um cookie.
domain	string	Domínio para o qual o cookie é visível. O padrão é o domínio da URL do documento ativo do contexto de navegação atual da sessão, se omitido ao adicionar um cookie.
secure	boolean	Indica se o cookie é seguro. O padrão é `false` se omitido ao adicionar um cookie.
httpOnly	boolean	Indica se o cookie é HTTP-only. O padrão é `false` se omitido ao adicionar um cookie.
expiry	integer	Tempo de expiração do cookie (em segundos desde a Época Unix). Não deve ser definido se omitido ao adicionar um cookie.
sameSite	string	O atributo `sameSite` do cookie, que controla se o cookie deve ser restrito a um contexto de primeira parte ou mesmo site. Pode ser definido como `"Lax"` ou `"Strict"`. O padrão é `"None"` se omitido ao adicionar um cookie.

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.

Pré-processamento de CSV para JSON

Visão geral

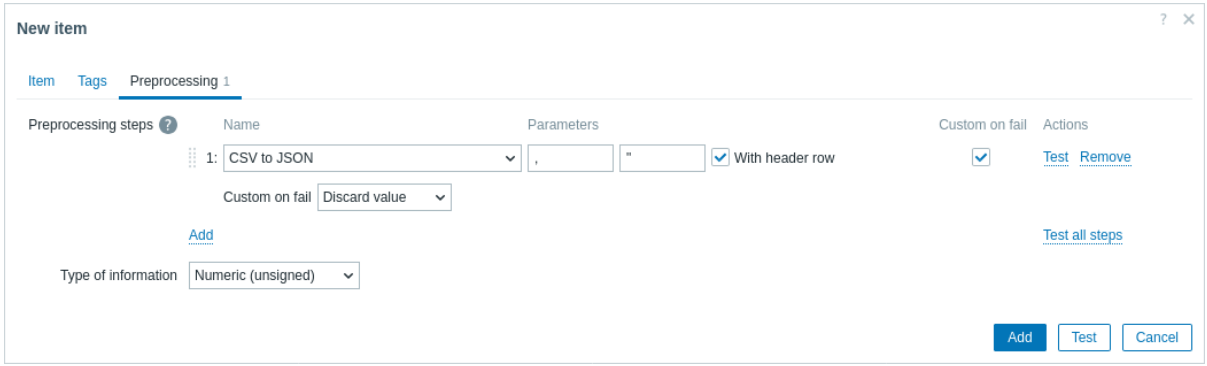
Nesta etapa de pré-processamento, é possível converter dados de arquivos CSV para o formato JSON. É compatível com:

- itens (protótipos de item)
- regras de descoberta de baixo nível

Configuração

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

1. Acesse a aba *Preprocessing* na configuração de item/discovery rule.
2. Clique em *Add*.
3. Selecione a opção *CSV to 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 nenhum delimitador for obtido da linha "Sep=", uma vírgula será usada como separador.

O segundo parâmetro opcional permite definir um símbolo de aspas.

Se a caixa de seleção *With header row* estiver marcada, os valores da linha de cabeçalho serão interpretados como nomes de colunas (consulte Processamento do cabeçalho para obter mais informações).

Se a caixa de seleção *Custom on fail* estiver marcada, o item não ficará sem suporte caso uma etapa de pré-processamento falhe. Além disso, é possível definir opções personalizadas de tratamento de erros: 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"
    }
]
```

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 compatíveis e parâmetros obrigatórios.

Os seguintes tipos de item são oferecidos atualmente pelo Zabbix:

- Zabbix agent
  - Monitoramento de arquivos de log
- Verificação simples
  - Monitoramento do VMware
- SNMP agent
- Trap SNMP
- Zabbix interno
- Zabbix trapper
- Verificação externa
- Monitor de banco de dados
- HTTP agent
  - Verificação do Prometheus
- IPMI agent
- SSH agent
- Telnet agent
- JMX agent
- Item calculado
- Item dependente
- Item de script
- Item de navegador
- Consulta de telemetria

Os detalhes de todos os tipos de item estão incluídos nas subpáginas desta seção. Embora os tipos de item ofereçam muitas opções para coleta de dados, há opções adicionais por meio de parâmetros de usuário ou módulos carregáveis.

Algumas verificações são realizadas somente pelo Zabbix server (como monitoramento sem agent), enquanto outras exigem o Zabbix agent ou até mesmo o Zabbix Java gateway (com monitoramento JMX).

> **Attention:**
> Se um determinado tipo de item exigir uma interface específica (por exemplo, uma verificação de IPMI exige uma interface IPMI no host), essa interface deverá existir na definição do host.

Várias interfaces podem ser definidas na definição do host: Zabbix agent, SNMP agent, JMX e IPMI. Se um item puder usar mais de uma interface, ele pesquisará as interfaces disponíveis do host (na ordem: Agent→SNMP→JMX→IPMI) para encontrar a primeira apropriada à qual será vinculado.

Todos os itens que retornam texto (tipos de informação caractere, log e texto) também podem retornar somente espaços em branco, quando aplicável, definindo o valor retornado como uma string vazia (compatível desde a versão 2.0).

Zabbix agent

Visão geral

Esta seção fornece detalhes sobre as chaves de item que usam comunicação com o agent do Zabbix para coleta de dados.

Há 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 Zabbix agent também são suportadas pelo Zabbix agent 2, da nova geração. Veja as chaves de item adicionais que você pode usar somente com o agent 2.

Chaves de item compatíveis

As chaves de item que você pode usar com o Zabbix agent estão listadas abaixo.

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

Chave de item	Descrição	Grupo de itens
kernel.maxfiles	O número máximo de arquivos abertos compatível com o SO.	Kernel
kernel.maxproc	O número máximo de processos compatível com o SO.	
kernel.openfiles	O número de descritores de arquivo 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 sofre rotação.	
logrt.count	A contagem de linhas correspondentes em um arquivo de log monitorado que sofre rotação.	
modbus.get	Lê dados do 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.get	A lista de interfaces de rede com informações detalhadas.	
net.if.in	As estatísticas do tráfego de entrada em uma interface de rede.	
net.if.out	As estatísticas do tráfego de saída em uma interface de rede.	
net.if.total	A soma das estatísticas do tráfego de entrada e de 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 estabelecer 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.	

Chave de item	Descrição	Grupo de itens
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 pelo processo.	Processos
proc.get	A lista de processos do SO e seus parâmetros.	
proc.mem	A memória usada pelo processo, em bytes.	
proc.num	O número de processos.	
sensor	Leitura do sensor de hardware.	Sensores
system.boottime	A hora 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 lista de dispositivos PCI ou USB.	
system.hw.macaddr	A lista 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 de software.	
system.sw.os	As informações do sistema operacional.	
system.sw.os.get	Informações detalhadas sobre o sistema operacional (versão, tipo, nome da distribuição, versão menor e principal etc.).	
system.sw.packages	A lista de pacotes instalados.	
system.sw.packages.get	Uma lista 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 como porcentagem do total.	
system.uname	A 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 seus tipos.	Sistemas de arquivos virtuais
vfs.dev.get	A lista de dispositivos de bloco com informações detalhadas.	
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	A soma de verificação do arquivo, calculada pelo algoritmo UNIX cksum.	
vfs.file.contents	Recupera o conteúdo de um arquivo.	
vfs.file.exists	Verifica se o arquivo existe.	
vfs.file.get	Retorna informações sobre um arquivo.	
vfs.file.md5sum	A soma de verificação 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 as permissões UNIX.	
vfs.file.regexp	Recupera uma string do arquivo.	
vfs.file.regmatch	Localiza uma string no arquivo.	
vfs.file.size	O tamanho do arquivo.	
vfs.file.time	As informações de data e hora do arquivo.	
vfs.fs.discovery	A lista de sistemas de arquivos montados com seus tipos e opções de montagem.	
vfs.fs.get	A lista de sistemas de arquivos montados com seus tipos e opções de montagem e, no modo *full*, o espaço em disco disponível e as estatísticas de inode.	
vfs.fs.inode	O número ou a porcentagem de inodes.	
vfs.fs.size	O espaço em disco em bytes ou como porcentagem do total.	

Chave de item	Descrição	Grupo de itens
vm.memory.size	O tamanho da memória em bytes ou como porcentagem do total.	Memória virtual
web.page.get	Obtém o conteúdo de uma página da web.	Monitoramento da web
web.page.perf	O tempo de carregamento de uma página da web completa.	
web.page.regexp	Localiza uma string na página da 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 Zabbix agent (Zabbix agent ou Zabbix agent 2).	
agent.version	A versão do Zabbix agent.	
zabbix.stats	Retorna um conjunto de métricas internas do Zabbix server ou proxy. Se `ip` e `port` forem especificados, as métricas serão recuperadas da instância remota; caso contrário, da instância local.	
zabbix.stats (queue)	Retorna o número de itens monitorados na fila que estão atrasados no Zabbix server ou proxy. Se `ip` e `port` forem especificados, as métricas serão recuperadas da instância remota; caso contrário, da instância local.	

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 retornado: *Log*.
 Consulte as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo e o nome de um arquivo de log;
- **regexp** - uma expressão regular que descreve o padrão necessário;
- **encoding** - o identificador da página de código identificador;
- **maxlines** - o número máximo de novas linhas por segundo que o agent enviará ao Zabbix server ou proxy. Este parâmetro substitui o valor de `MaxLinesPerSecond` em `zabbix_agentd.conf`;
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente);

- **output** - um template opcional de formatação da saída. A sequência de escape **\0** é substituída pela parte correspondente do texto (do primeiro caractere onde a correspondência começa até o caractere onde a correspondência termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou por uma string vazia se N exceder o número de grupos capturados);
- **maxdelay** - o atraso máximo em segundos. Tipo: float. Valores:
  - 0 - (padrão) nunca ignorar linhas do arquivo de log;
  - > 0.0 - ignorar linhas mais antigas para obter as linhas mais recentes analisadas dentro de ”maxdelay” segundos. Leia as observações sobre maxdelay antes de usá-lo!
- **options** - opções adicionais:
mtime-noreread - registros não exclusivos, reler somente se o tamanho do arquivo mudar (ignorar a alteração do horário de modificação). (Esse parâmetro está obsoleto desde 5.0.2, porque agora mtime é ignorado.)
- **persistent dir** (somente no zabbix_agentd em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também observações adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como uma verificação ativa.
- Se o arquivo estiver ausente ou as permissões não permitirem acesso, o item ficará sem suporte.
- Se `output` for deixado em branco - a linha inteira que contém o texto correspondente será retornada. Observe que todos os tipos globais de expressão regular, exceto *Matches regular expression*, 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 corr
```

Exemplo de uso do parâmetro `output` para reescrever um registro de log antes de enviá-lo ao 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 que descreve o padrão necessário;
- **encoding** - o identificador da página de código identificador;
- **maxproclines** - o número máximo de novas linhas por segundo que o agent analisará (não pode exceder 10000). O valor padrão é 10*`MaxLinesPerSecond` em `zabbix_agentd.conf`.
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente).
- **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 observações sobre maxdelay antes de usá-lo!
- **options** - opções adicionais:
mtime-noreread - registros não exclusivos, relê somente se o tamanho do arquivo mudar (ignora a alteração do horário de modificação). (Esse parâmetro está obsoleto desde 5.0.2, porque agora mtime é ignorado.)
- **persistent dir** (somente em `zabbix_agentd` em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também observações 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 acesso, o item passa a 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 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 completo (por exemplo, `/path/to/agent` corresponderá a `zabbix_agentd.log`).
- **regexp** - uma expressão regular que descreve 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 ao Zabbix server ou proxy. Este parâmetro substitui o valor de `MaxLinesPerSecond` em `zabbix_agentd.conf`.
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente).
- **output** - um template opcional de formatação da saída. A sequência de escape **\0** é substituída pela parte correspondente do texto (do primeiro caractere em que a correspondência começa até o caractere em que a correspondência termina), enquanto uma sequência de escape **\N** (em que N=1...9) é substituída pelo N-ésimo grupo correspondente (ou por 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 obter as linhas mais recentes analisadas dentro de "maxdelay" segundos. Leia as observações 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*. Nesse caso, *maxdelay* deve ser 0 ou não especificado; veja as observações sobre copytruncate.
  - *mtime-reread* - registros não exclusivos, reler se o tempo de modificação ou o tamanho mudar (usado por padrão se nenhuma opção `mtime-*` for explicitamente definida).
  - *mtime-noreread* - registros não exclusivos, reler somente se o tamanho mudar (ignora a alteração do tempo de modificação).
- **persistent dir** (somente em `zabbix_agentd` em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também observações adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como uma verificação ativa.
- A rotação de log é baseada no último horário de modificação dos arquivos.
- Observe que `logrt` foi projetado para funcionar com um arquivo de log atualmente ativo, com vários outros arquivos inativos correspondentes rotacionados. Se, por exemplo, um diretório tiver muitos arquivos de log ativos, um item `logrt` separado deve ser criado para cada um. Caso contrário, se um item `logrt` coletar muitos arquivos, isso pode levar ao esgotamento de memória e a uma falha do monitoramento.
- Se `output` for deixado em branco - a linha inteira contendo o texto correspondente será retornada. Observe que todos os tipos de expressão regular global, exceto *Matches regular expression*, sempre retornam a linha inteira correspondente e o parâmetro `output` é ignorado.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.
- No parâmetro `file regexp`, o caminho do diretório de log e a expressão regular do arquivo de log devem ser separados pelo separador de diretório correto:
  - No Windows, o separador deve ser uma barra invertida (\); barras normais podem ser toleradas em outras posições, exceto na que separa o caminho do diretório de log e a expressão regular do arquivo de log (veja os 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 de um registro de log:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #este ite
```

Exemplo de uso do parâmetro `output` para reescrever um registro de log antes de enviá-lo ao server:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 A contagem de linhas correspondentes em um arquivo de log monitorado que é rotacionado.
 Valor de retorno: *Integer*.
 Consulte 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 completo (por exemplo, `/path/to/agent` corresponderá a `zabbix_agentd.log`).
- **regexp** - uma expressão regular que descreve o padrão necessário.
- **encoding** - o identificador da página de código identificador.
- **maxproclines** - o número máximo de novas linhas por segundo que o agent analisará (não pode exceder 10000). O valor padrão é 10*MaxLinesPerSecond em `zabbix_agentd.conf`.
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente).
- **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 observações 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`. Nesse caso, `maxdelay` deve ser 0 ou não especificado; consulte as observações de `copytruncate`.
  - *mtime-reread* - registros não exclusivos, reler se o tempo de modificação ou o tamanho mudar (padrão).
  - *mtime-noreread* - registros não exclusivos, reler apenas se o tamanho mudar (ignora a alteração do tempo de modificação).
- **persistent dir** (somente em `zabbix_agentd` em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Consulte também observações 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.
- A rotação de log é baseada no último horário de modificação dos arquivos.
- No parâmetro `file regexp`, o caminho do diretório de log e a expressão regular do arquivo de log devem ser separados pelo separador de diretório correto:
  - No Windows, o separador deve ser uma barra invertida (\); barras normais podem ser toleradas em outras posições, exceto na que separa o caminho do diretório de log e a expressão regular do arquivo de log (veja os exemplos abaixo).
  - Em outros sistemas, deve ser uma barra normal (/).

Exemplos para Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item contará o número de corresp
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item contará o
```

Exemplos para outros sistemas:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #este item contará o número de correspondên
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #este item contará o núme
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Lê dados Modbus.
 Valor de retorno: *string JSON*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **endpoint** - o endpoint definido como `protocol://connection_string`;
- **slave id** - o ID do slave;
- **function** - a função Modbus;
- **address** - o endereço do primeiro registrador, coil ou input;
- **count** - o número de registros a serem lidos;
- **type** - o tipo de dados;
- **endianness** - a configuração de endianness;
- **offset** - o número de registradores, começando em `address`, cujos resultados serão descartados.

No Windows, este item segue caminhos UNC no parâmetro `endpoint`, o que pode representar um risco de segurança.

Veja uma descrição detalhada dos parâmetros.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Verifica o status de um serviço DNS.
 Valores de retorno: - 0 - a resolução DNS falhou (o servidor DNS não respondeu ou retornou um erro); - 1 - a resolução DNS foi bem-sucedida.
 Consulte as plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para usar o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é `SOA`);
- **timeout** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o tempo limite da solicitação, em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o número de tentativas da 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` está definido como *PTR*), você pode fornecer o nome DNS nos formatos invertido e não invertido (veja os exemplos abaixo). Observe que, quando o registro PTR é solicitado, o nome DNS na verdade é um endereço IP.
- Nomes de domínio internacionalizados não são suportados; use nomes codificados em IDNA.

Exemplos:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 - o serviço está indisponível; segundos - a quantidade de segundos gastos aguardando uma resposta do serviço).
 Consulte as plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para usar o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é `SOA`);
- **timeout** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o tempo limite da solicitação, em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que esteja usando Zabbix agent 2) - o número de tentativas da 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 sistemas operacionais), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Para consultas DNS reversas (quando `type` está definido como *PTR*), você pode fornecer o nome DNS nos formatos invertido e não invertido (consulte os exemplos abaixo). Observe que, quando um 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.
- 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 necessário.
 Consulte as plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é `SOA`);
- **timeout** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o tempo limite da solicitação em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o número de tentativas da 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 o Zabbix agent no Windows, Zabbix agent 2 em todos os SO), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Para consultas DNS reversas (quando `type` está definido como *PTR*), você pode fornecer o nome DNS em formato invertido ou não invertido (consulte os exemplos abaixo). Observe que, quando o registro PTR é solicitado, o nome DNS é na verdade um endereço IP.
- Nomes de domínio internacionalizados não são suportados; use nomes codificados em IDNA.

Exemplos:

```
net.dns.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, Windows.

net.if.get[<regexp>]

 A lista de interfaces de rede com informações detalhadas.
 Usado para descoberta de baixo nível.
 Este item é uma versão estendida do item do Zabbix agent `net.if.discovery`.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux.

Parâmetros:

- **regexp** - uma expressão regular que corresponde aos nomes das interfaces de rede.

Comentários:

- Retorna um objeto JSON com interfaces de rede cujos nomes correspondem à expressão regular fornecida (veja mais detalhes).

net.if.in[if,<mode>]

 As estatísticas de tráfego de entrada em uma interface de rede.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. São necessários privilégios de root 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 do 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 key com a etapa de pré-processamento *Change per second* para obter as estatísticas de bytes por segundo.
- O modo *dropped* é suportado apenas em Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Os modos *overruns*, *frame*, *compressed*, *multicast* são suportados apenas em Linux.
- No HP-UX, este item não fornece detalhes sobre interfaces de loopback (por exemplo, lo0).

Exemplos:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 As estatísticas de tráfego de saída em uma interface de rede.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Privilégios de root são necessários no NetBSD.

Parâmetros:

- **if** - nome da interface de rede (Unix); descrição completa da interface de rede ou endereço IPv4; ou, se entre chaves, GUID da interface de rede (Windows);
- **mode** - valores possíveis:
  - *bytes* - número de bytes (padrão);
  - *packets* - número de pacotes;
  - *errors* - número de erros;
  - *dropped* - número de pacotes descartados;
  - *overruns (fifo)* - número de erros do buffer FIFO;
  - *collisions (colls)* - número de colisões detectadas na interface;
  - *carrier* - número de perdas de carrier detectadas pelo driver do dispositivo;
  - *compressed* - número de pacotes compactados 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 em Linux, HP-UX.
- Os modos *overruns*, *collision*, *carrier*, *compressed* são suportados apenas em 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 retornado: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Privilégios de root são necessários no NetBSD.

Parâmetros:

- **if** - nome da interface de rede (Unix); descrição completa da interface de rede ou endereço IPv4; ou, se entre chaves, GUID da interface de rede (Windows);
- **mode** - valores possíveis:
  - *bytes* - número de bytes (padrão);
  - *packets* - número de pacotes;
  - *errors* - número de erros;
  - *dropped* - número de pacotes descartados;
  - *overruns (fifo)* - o número de erros do buffer FIFO;
  - *collisions (colls)* - o número de colisões detectadas na interface;
  - *compressed* - o 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 em Linux, HP-UX. Pacotes descartados são suportados somente se ambos `net.if.in` e `net.if.out` funcionarem para pacotes descartados na sua plataforma.
- Os modos *overruns*, *collision*, *compressed* são suportados apenas em Linux.
- No HP-UX, este item não fornece detalhes sobre interfaces de loopback (por exemplo, lo0).

Exemplos:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Verifica se esta porta TCP está no estado LISTEN.
 Valores de retorno: 0 - não está no estado LISTEN; 1 - está no estado LISTEN.
 Plataformas suportadas: Linux, FreeBSD, Solaris, MacOS X, Windows.

Parâmetro:

- **port** - número da porta TCP

Em kernels Linux 2.6.14 e superiores, as informações sobre sockets TCP em escuta são obtidas da interface NETLINK do kernel, se possível. Caso contrário, as informações são recuperadas dos arquivos `/proc/net/tcp` e `/roc/net/tcp6`.

Exemplo:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Verifica se é possível estabelecer uma conexão TCP com a porta especificada.
 Valores de retorno: 0 - não é possível conectar; 1 - é possível conectar.
 Consulte as plataformas suportadas.

Parâmetros:

- **ip** - o endereço IP ou nome DNS (o padrão é `127.0.0.1`);
- **port** - o número da porta.

Comentários:

- Para testes simples de desempenho TCP, use `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Essas verificações podem resultar em mensagens adicionais nos arquivos de log dos daemons do sistema (sessões SMTP e SSH sendo registradas normalmente).

Exemplo:

```
net.tcp.port[,80] #este item pode ser usado para testar a disponibilidade do web server em execução na por
```

net.tcp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e aceitando conexões TCP.
 Valores de retorno: 0 - o serviço está inativo; 1 - o serviço está em execução.
 Consulte 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, é usado o número da porta padrão do serviço).

Comentários:

- Essas verificações podem resultar em mensagens adicionais nos arquivos de log do daemon do sistema (normalmente, sessões SMTP e SSH são registradas).
- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) não é suportada no momento. 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 telnet procura um prompt de login (':' no final).

Exemplo:

```
net.tcp.service[ftp,,45] #this item can be used to test the availability of FTP server on TCP port 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço TCP.
 Valores de retorno: *Float* (0 - o serviço está indisponível; segundos - a quantidade 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 (o padrão é `127.0.0.1`);
- **port** - o número da porta (por padrão, é usado o número da porta padrão do serviço).

Comentários:

- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) não é suportada no momento. Como alternativa, use `net.tcp.service.perf[tcp,<ip>,<port>]` para verificações como essas.
- A verificação telnet procura um prompt de login (':' no final).

Exemplo:

```
net.tcp.service.perf[ssh] #this item can be used to test the speed of initial response from the SSH server
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Retorna o número de sockets TCP que correspondem aos parâmetros.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **laddr** - o endereço IPv4/6 local ou sub-rede CIDR;
- **lport** - o número da porta local ou nome do serviço;
- **raddr** - o endereço IPv4/6 remoto ou sub-rede CIDR;
- **rport** - o número da porta remota ou nome do serviço;
- **state** - o estado da conexão (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Exemplo:

```
net.tcp.socket.count[,80,,,established] #o número de conexões para a porta TCP local 80 no estado establis
```

net.udp.listen[port]

 Verifica se esta porta UDP está no estado LISTEN.
 Valores de retorno: 0 - não está no estado LISTEN; 1 - está no estado LISTEN.
 Plataformas suportadas: Linux, FreeBSD, Solaris, MacOS X.

Parâmetro:

- **port** - número da porta UDP

Exemplo:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e respondendo a solicitações UDP.
 Valores de retorno: 0 - o serviço está inativo; 1 - o serviço está em execução.
 Consulte plataformas suportadas.

Parâmetros:

- **service** - *ntp* (consulte detalhes);
- **ip** - o endereço IP ou nome DNS (o padrão é `127.0.0.1`);
- **port** - o número da porta (por padrão, é usado o número da porta padrão do serviço).

Exemplo:

```
net.udp.service[ntp,,45] #this item can be used to test the availability of NTP service on UDP port 45
```

net.udp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço UDP.
 Valores de retorno: *Float* (0 - o serviço está inativo; segundos - a quantidade de segundos gastos aguardando uma resposta do serviço).
 Veja plataformas suportadas.

Parâmetros:

- **service** - *ntp* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (o padrão é `127.0.0.1`);
- **port** - o número da porta (por padrão, é usado o número da porta padrão do serviço).

Exemplo:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Retorna o número de sockets UDP que correspondem aos parâmetros.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **laddr** - o endereço IPv4/6 local ou sub-rede CIDR;
- **lport** - o número da porta local ou nome do serviço;
- **raddr** - o endereço IPv4/6 remoto ou sub-rede CIDR;
- **rport** - o número da porta remota ou nome do serviço;
- **state** - o estado da conexão (*established*, *unconn*).

Exemplo:

```
net.udp.socket.count[,,,,established] #retorna o número de sockets UDP no estado conectado
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 A porcentagem de utilização de CPU do processo.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, Solaris[6].

Parâmetros:

- **name** - o nome do processo (padrão é *all processes*);
- **user** - o nome do usuário (padrão é *all users*);
- **type** - o tipo de utilização de CPU: *total* (padrão), *user* ou *system*;
- **cmdline** - filtra 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 somente no Solaris.

Comentários:

- O valor retornado é baseado na porcentagem de utilização de um único núcleo de CPU. Por exemplo, a utilização de CPU de um processo que usa totalmente dois núcleos é 200%.
- Os dados de utilização de 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 durante as últimas 24 horas são removidas do coletor.
- Ao definir o parâmetro `zone` como *current* (ou o padrão), caso o agent tenha sido compilado em um Solaris sem suporte a zonas, mas esteja sendo executado em uma versão mais recente do Solaris com suporte a zonas, o agent retornará NOTSUPPORTED (o agent não pode limitar os resultados apenas à zona atual). No entanto, *all* é suportado nesse caso.

Exemplos:

```
proc.cpu.util[,root] #Utilização de CPU de todos os processos em execução sob o usuário "root"
proc.cpu.util[zabbix_server,zabbix] #Utilização de CPU de todos os processos zabbix_server em execução sob
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 A lista de processos do sistema operacional e seus parâmetros. Pode ser usado para descoberta em 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** - filtra pela linha de comando (é uma expressão regular). Este parâmetro não é suportado no Windows; em outras plataformas, não é suportado se mode estiver definido como 'summary'.

- **mode** - valores possíveis:
  - *process* (padrão);
  - *thread* (não suportado no NetBSD);
  - *summary*. Veja uma lista de parâmetros de processo retornados para cada modo e sistema operacional.

Comentários:

- Se um valor não puder ser obtido, por exemplo, devido a um erro (o processo já terminou, falta de permissões, falha na chamada de sistema), −1 será retornado.
- Veja as notas sobre a seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

Exemplos:

```
proc.get[zabbix_server,zabbix,,process] #lista de todos os processos zabbix_server 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 *all processes*);
- **user** - o nome do usuário (padrão é *all users*);
- **mode** - valores possíveis: *avg*, *max*, *min* ou *sum* (padrão);
- **cmdline** - filtra pela linha de comando (é uma expressão regular);
- **memtype** - o tipo de memória usado 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 altos e irreais.

Veja as notas sobre a seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

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.mem[,,,apache2]`), um processo extra será contabilizado, pois o agent contará a si mesmo.

Exemplos:

```
proc.mem[,root] #a memória usada por todos os processos em execução sob o usuário "root"
proc.mem[zabbix_server,zabbix] #a memória usada por todos os processos zabbix_server em execução sob o usu
proc.mem[,oracle,max,oracleZABBIX] #a memória usada pelo processo que consome mais memória em execução sob
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 O número de processos.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parâmetros:

- **name** - o nome do processo (o padrão é *all processes*).
- **user** - o nome do usuário (o padrão é *all users*).
- **state** - valores possíveis:
  - *all* (padrão);
  - *disk* - sono ininterruptível;
  - *run* - em execução;
  - *sleep* - sono interrompível;
  - *trace* - parado;
  - *zomb* - zumbi.
- **cmdline** - filtra 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 no Linux, FreeBSD, OpenBSD, NetBSD.
- No Windows, apenas os parâmetros `name` e `user` são suportados.
- 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 contabilizado, pois o agent contará a si mesmo.

- Ao definir o parâmetro zone como `current` (ou o padrão), caso o agent tenha sido compilado em um Solaris sem suporte a zonas, mas esteja sendo executado em um Solaris mais recente com suporte a zonas, o agent retornará NOTSUPPORTED (o agent não pode limitar os resultados apenas à zona atual). No entanto, *all* é suportado nesse caso.
- Consulte as notas sobre a seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

Exemplos:

```
proc.num[,mysql] #o número de processos em execução sob o usuário mysql
proc.num[apache2,www-data] #o número de processos apache2 em execução sob o usuário www-data
proc.num[,oracle,sleep,oracleZABBIX] #o número de processos no estado sleep em execução sob Oracle, tendo
```

sensor[device,sensor,<mode>]

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

Parâmetros:

- **device** - o nome do dispositivo;
- **sensor** - o nome do sensor;
- **mode** - valores possíveis: *avg*, *max* ou *min*.

Comentários:

- Lê `/proc/sys/dev/sensors` no Linux 2.4.
- Lê `/sys/class/hwmon` no Linux 2.6+. Veja uma descrição mais detalhada do item sensor no Linux.
- Lê o MIB `hw.sensors` no OpenBSD.
- No Linux 2.4, se você especificar mode, então device e sensor são tratados como expressões regulares.
- No Linux 2.6+ e OpenBSD, ou se mode for omitido, device e sensor são tratados como texto literal.

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, Windows.

Parâmetro:

- **type** - valores possíveis: *online* (padrão) ou *max*

O parâmetro de tipo *max* é suportado apenas em Linux, FreeBSD, Solaris, MacOS X.

Exemplo:

```
system.cpu.num
```

system.cpu.switches

 A contagem de trocas de contexto.
 Valor de retorno: *Inteiro*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 A porcentagem de utilização da CPU.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parâmetros:

- **cpu** - *<número da CPU>* ou *all* (padrão);
- **type** - valores possíveis: *user* (padrão), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *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 retornado: *String*.
 Veja plataformas suportadas.

Parâmetros:

- **type** - valores possíveis:
  - *netbios* (padrão no Windows);
  - *host* (padrão no Linux);
  - *shorthost* (retorna a parte do hostname antes do primeiro ponto, uma string completa para nomes sem pontos);
  - *fqdn* (retorna o Fully Qualified Domain Name).
- **transform** - valores possíveis:
  - *none* (padrão);
  - *lower* (converte para minúsculas).

O valor é obtido a partir de `nodename` na 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 informada em `interface` (*all* lista todas as interfaces).
- Se `format` for especificado como *short*, os nomes das interfaces 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;
  - *local* - o tempo no formato 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Deve ser usado como uma verificação passiva para o Zabbix agent; no Zabbix agent 2, pode ser configurado como uma verificação ativa.

Exemplo:

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

system.run[command,<mode>]

 Executa o comando especificado no host.
 Valor de retorno: resultado *Text* do comando ou 1 - com `mode` como *nowait* (independentemente do resultado do comando).
 Veja supported platforms.

Parâmetros:

- **command** - comando para execução;
- **mode** - valores possíveis:
  - *wait* - aguarda o término da execução (padrão);
  - *nowait* - não aguarda.

Comentários:

- Este item está desativado por padrão. Saiba como habilitá-los.
- O valor de retorno do item é a saída padrão juntamente com a saída de erro padrão produzida pelo comando. Exit code checking não é executada.
- Para ser processado corretamente, o valor de retorno do comando deve ser do tipo de dados `text`. Um resultado vazio também é permitido.
- O valor de retorno é limitado a 16MB (incluindo espaços em branco no final, que são truncados); database limits também se aplicam.
- Veja também: Command execution.

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 a 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 Virtual Memory Manager (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 page faults e atividade de paginação:
  - *fi* - páginas de arquivo carregadas por segundo (float);
  - *fo* - páginas de arquivo descarregadas por segundo (float);
  - *pi* - páginas paginadas da área de paginação (float);
  - *po* - páginas paginadas para a área de paginação (float);
  - *fr* - páginas liberadas (substituição de páginas) (float);
  - *sr* - páginas examinadas pelo algoritmo de substituição de páginas (float).
- **faults,<type>** - taxa de traps e interrupções:
  - *in* - interrupções de dispositivo (float);
  - *sy* - chamadas de sistema (float);
  - *cs* - trocas de contexto de threads do kernel (float).
- **cpu,<type>** - detalhamento do uso percentual do tempo de processador:
  - *us* - tempo de usuário (float);
  - *sy* - tempo de sistema (float);
  - *id* - tempo ocioso (float);
  - *wa* - tempo ocioso durante o qual o sistema tinha solicitações pendentes de I/O de disco/NFS (float);
  - *pc* - número de processadores físicos consumidos (float);
  - *ec* - percentual da capacidade contratada consumida (float);
  - *lbusy* - indica o percentual de utilização do(s) processador(es) lógico(s) que ocorreu durante a execução no nível de usuário e sistema (float);
  - *app* - indica os processadores físicos disponíveis no pool compartilhado (float).
- **disk,<type>** - estatísticas de disco:
  - *bps* - indica a quantidade de dados transferidos (lidos ou gravados) para a unidade, em bytes por segundo (integer);
  - *tps* - indica o número de transferências por segundo emitidas para o disco/fita físico (float).

Comentários:

- Observe as seguintes limitações nestes itens:
  - `system.stat[cpu,app]` - suportado apenas em AIX LPAR do tipo ”Shared”;
  - `system.stat[cpu,ec]` - suportado em AIX LPAR dos tipos ”Shared” e ”Dedicated” (”Dedicated” sempre retorna 100 (percent));
  - `system.stat[cpu,lbusy]` - suportado apenas em AIX LPAR do tipo ”Shared”;
  - `system.stat[cpu,pc]` - suportado em AIX LPAR dos tipos ”Shared” e ”Dedicated”;
  - `system.stat[ent]` - suportado em AIX LPAR dos tipos ”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*

As informações de info são obtidas de (observe 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: *Text*.
 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 informada (*all* lista todos eles).
- Gerenciadores de pacotes suportados (comando executado):
  - pkg (`dpkg --get-selections`);
  - pkgtool (`ls /var/log/packages`);
  - rpm (`rpm -qa`);
  - pacman (`pacman -Q`);
  - portage.
- Se `format` for especificado como *full*, os pacotes são agrupados por gerenciadores de pacotes (cada gerenciador em uma linha separada, começando com seu nome entre colchetes).
- Se `format` for especificado como *short*, os pacotes não são agrupados e são listados em uma única linha.

Exemplo:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Uma listagem detalhada dos pacotes instalados.
 Valor de retorno: *JSON string*.
 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 sem formatação 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** - especifique o dispositivo usado para swap (somente Linux) ou *all* (padrão);
- **type** - valores possíveis:
  - *count* (número de swapins, padrão em plataformas não Linux);
  - *sectors* (setores trocados para a memória);
  - *pages* (páginas trocadas para a memória, 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 device não tiver sido especificado.
- O parâmetro de tipo *sectors* é suportado somente 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** - especifique o dispositivo usado para swap (somente Linux) ou *all* (padrão);
- **type** - valores possíveis:
  - *count* (número de swapouts, padrão em plataformas não Linux);
  - *sectors* (setores gravados em swap);
  - *pages* (páginas gravadas em swap, padrão no Linux).

Comentários:

- A origem destas 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 device não tiver sido especificado.
- O parâmetro de tipo *sectors* é suportado apenas no Linux.

Exemplo:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 O tamanho do espaço de swap em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parâmetros:

- **device** - especifique o dispositivo usado para swap (somente 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 total de swap);
  - *used* (espaço de swap usado).

Comentários:

- Observe que *pfree*, *pused* não são suportados no Windows se o tamanho do swap for 0.

- Se device não for especificado, o Zabbix agent considerará 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 dos dispositivos de swap (ao contrário 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, Windows.

Na configuração do item, use as unidades **s** ou **uptime** para obter valores legíveis.

system.users.num

 O número de usuários conectados.
 Valor de retorno: *Integer*.
 Veja 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.get[<mode>,<regexp>]

 A lista de dispositivos de bloco com informações detalhadas. Usado para descoberta em baixo nível.
 Valor de retorno: *string JSON*.
 Plataformas suportadas: Linux, AIX.

Parâmetros:

- **mode** - valores possíveis:
  - *disks* (padrão) - retorna a lista de discos;
  - *disk_stats* - retorna estatísticas para cada disco;
  - *devices* (somente Linux) - retorna a lista de dispositivos de bloco (discos e partições);
  - *device_stats* (somente Linux) - retorna estatísticas para cada dispositivo (discos e partições).
- **regexp** - uma expressão regular usada para filtrar dispositivos de bloco pelo nome.

vfs.dev.read[<device>,<type>,<mode>]

 As estatísticas de leitura de disco.
 Valor retornado: - *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: sectors, operations, bytes por segundo, respectivamente);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*. Este parâmetro é suportado somente com `type` em: sps, ops, bps.

Comentários:

- Se estiver usando um intervalo de atualização de três horas ou mais[2], este item sempre retornará 0.
- Os parâmetros de tipo *sectors* e *sps* são suportados somente no Linux.
- O parâmetro de tipo *ops* é suportado somente no Linux e no FreeBSD.
- O parâmetro de tipo *bps* é suportado somente no FreeBSD.
- O parâmetro de tipo *bytes* é suportado somente no FreeBSD, Solaris, AIX, OpenBSD.
- O parâmetro `mode` é suportado somente no Linux, FreeBSD.

- Você pode usar nomes relativos de dispositivos (por exemplo, `sda`) assim como um prefixo opcional `/dev/` (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 gravação 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: sectors, operations, bytes por segundo, respectivamente);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*. Este parâmetro é suportado somente com `type` em: sps, ops, bps.

Comentários:

- Se estiver usando um intervalo de atualização de três horas ou mais[2], este item sempre retornará 0.
- Os parâmetros de tipo *sectors* e *sps* são suportados somente no Linux.
- O parâmetro de tipo *ops* é suportado somente no Linux e FreeBSD.
- O parâmetro de tipo *bps* é suportado somente no FreeBSD.
- O parâmetro de tipo *bytes* é suportado somente no FreeBSD, Solaris, AIX, OpenBSD.
- O parâmetro `mode` é suportado somente no Linux, FreeBSD.
- Você pode usar nomes relativos de dispositivos (por exemplo, `sda`) assim como um prefixo opcional `/dev/` (por exemplo, `/dev/sda`).
- Volumes lógicos LVM são suportados.
- Os valores padrão do parâmetro `type` para diferentes sistemas operacionais:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* e *bps* nas plataformas suportadas são limitados a 1024 dispositivos (1023 individuais e um para *all*).

Exemplo:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 A contagem de entradas do diretório.
 Valor retornado: *Integer*.
 Consulte plataformas compatíveis.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser incluída; inclui todas se estiver vazia (valor padrão);
- **regex excl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser excluída; não exclui nenhuma se estiver vazia (valor padrão);
- **types incl** - tipos de entradas de diretório a serem contados. 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 colocados entre aspas.
- **types excl** - tipos de entradas de diretório (consulte `types incl`) que NÃO devem ser contados. Se algum tipo de entrada estiver presente em `types incl` e `types excl`, as entradas de diretório desse tipo NÃO serão contadas.
- **max depth** - a profundidade máxima dos subdiretórios a serem percorridos:
  - **-1** (padrão) - ilimitada;
  - **0** - não entra nos subdiretórios.
- **min size** - o tamanho mínimo (em bytes) para que o arquivo seja contado. Arquivos menores não serão contados. Podem ser usados sufixos de memória.
- **max size** - o tamanho máximo (em bytes) para que o arquivo seja contado. Arquivos maiores não serão contados. Podem ser usados sufixos de memória.
- **min age** - a idade mínima (em segundos) da entrada de diretório para que seja contada. Entradas mais recentes não serão contadas. Podem ser usados sufixos de tempo.
- **max age** - a idade máxima (em segundos) da entrada de diretório para que seja contada. Entradas com essa idade ou mais antigas não serão contadas (data de modificação). Podem ser usados sufixos de tempo.
- **regex excl dir** - uma expressão regular que descreve o padrão de nome do diretório a ser excluído. Todo o conteúdo do diretório será excluído (diferentemente de `regex_excl`).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode representar um risco de segurança.
- Variáveis de ambiente, como `%APP_HOME%`, `$HOME` e `%TEMP%`, não são compatíveis.
- Os pseudo-diretórios "." e ".." nunca são contados.
- Links simbólicos nunca são seguidos durante a travessia de diretórios.
- Tanto `regex incl` quanto `regex excl` são aplicados a arquivos e diretórios ao calcular a contagem de entradas, mas são ignorados ao selecionar os subdiretórios a serem percorridos (se `regex incl` for `(?i)^.+\.zip$` e `max depth` não estiver definido, todos os subdiretórios serão percorridos, mas somente os arquivos do tipo zip serão contados).
- O tempo de execução é limitado pelo valor de timeout na configuração do agent (3 segundos). Como a travessia de diretórios grandes pode levar mais tempo, nenhum dado será retornado e o item se tornará sem suporte. Uma contagem parcial não será retornada.
- Ao filtrar por tamanho, somente arquivos regulares têm tamanhos significativos. No Linux e no BSD, os diretórios também têm tamanhos diferentes de zero (normalmente alguns KB). Os dispositivos têm tamanho zero; por exemplo, o tamanho de `/dev/sda1` não reflete o tamanho da partição correspondente. Portanto, ao usar `<min_size>` e `<max_size>`, recomenda-se especificar `<types_incl>` como "*file*" para evitar resultados inesperados.

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 retornado: *string JSON*.
 Consulte plataformas compatíveis.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser incluída; inclui todas se estiver vazia (valor padrão);
- **regex excl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser excluída; não exclui nenhuma se estiver vazia (valor padrão);
- **types incl** - tipos de entradas de diretório a serem listados, 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 colocados entre aspas.

- **types excl** - tipos de entradas de diretório (consulte `types incl`) que NÃO devem ser listados. Se algum tipo de entrada estiver presente em `types incl` e `types excl`, as entradas de diretório desse tipo NÃO serão listadas.
- **max depth** - a profundidade máxima dos subdiretórios a serem percorridos:
  - **-1** (padrão) - ilimitada;
  - **0** - não entra em subdiretórios.
- **min size** - o tamanho mínimo (em bytes) para que o arquivo seja listado. Arquivos menores não serão listados. Sufixos de memória podem ser usados.
- **max size** - o tamanho máximo (em bytes) para que o arquivo seja 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 com essa idade ou mais antigas não serão listadas (data de modificação). Sufixos de tempo podem ser usados.
- **regex excl dir** - uma expressão regular que descreve o padrão de nome do diretório a ser excluído. Todo o conteúdo do diretório será excluído (diferentemente de `regex excl`).

Comentários:

- No Windows, este item segue caminhos UNC, que podem representar um risco de segurança.
- Variáveis de ambiente, como `%APP_HOME%`, `$HOME` e `%TEMP%`, não são compatíveis.
- Os pseudo-diretórios "." e ".." nunca são listados.
- Links simbólicos nunca são seguidos durante a navegação pelos diretórios.
- Tanto `regex incl` quanto `regex excl` são aplicadas a arquivos e diretórios ao gerar a lista de entradas, mas são ignoradas ao selecionar subdiretórios para percorrer (se `regex incl` for `(?i)^.+\.zip$` e `max depth` não estiver definido, todos os subdiretórios serão percorridos, mas somente os arquivos do tipo zip serão contabilizados).
- O tempo de execução é limitado pelo valor de timeout na configuração do agent. Como a navegação por diretórios grandes pode levar mais tempo do que isso, nenhum dado será retornado e o item se tornará sem suporte. Uma lista parcial não será retornada.
- Ao filtrar por tamanho, somente arquivos regulares têm tamanhos significativos. No Linux e no BSD, os diretórios também têm tamanhos diferentes de zero (normalmente alguns KB). Os dispositivos têm tamanho zero; por exemplo, o tamanho de `/dev/sda1` não reflete o tamanho da partição correspondente. Portanto, ao usar `min size` e `max size`, recomenda-se especificar `types incl` como "*file*" para evitar resultados inesperados.

Exemplos:

```
vfs.dir.get[/dev] #recupera a lista de dispositivos em /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #recupera a lista de arquivos em um diretório temporár
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 O tamanho do diretório (em bytes).
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows. O item pode funcionar em outras plataformas do tipo UNIX.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a incluir; inclui tudo se estiver vazio (valor padrão);
- **regex excl** - uma expressão regular que descreve o padrão de nome da entidade (arquivo, diretório, link simbólico) a excluir; não exclui nada se estiver vazio (valor padrão);
- **mode** - valores possíveis:
  - *apparent* (padrão) - obtém tamanhos aparentes dos arquivos em vez do uso em disco (age como `du -sb dir`);
  - *disk* - obtém o uso em disco (age como `du -s -B1 dir`). Ao contrário do comando du, o item vfs.dir.size considera arquivos ocultos ao calcular o tamanho do diretório (age como `du -sb .[^.]* *` dentro de dir).
- **max depth** - a profundidade máxima de subdiretórios a percorrer:
  - **-1** (padrão) - ilimitada;
  - **0** - não desce para subdiretórios.
- **regex excl dir** - uma expressão regular que descreve o padrão de nome do diretório a excluir. Todo o conteúdo do diretório será excluído (em contraste com `regex excl`)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- Somente diretórios com pelo menos permissão de leitura para o usuário *zabbix* são calculados. Para diretórios com apenas permissão de leitura, o tamanho do próprio diretório é calculado. Diretórios com permissões de leitura e execução são calculados incluindo o conteúdo.
- Em diretórios grandes ou unidades lentas, este item pode exceder o tempo limite devido à configuração de 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 que contêm 'log' em seus nomes
vfs.dir.size[/tmp,log,^.+\.old$] #calcula o tamanho de todos os arquivos em /tmp que contêm 'log' em seus
```

vfs.file.cksum[file,<mode>]

 A soma de verificação do arquivo, calculada pelo algoritmo UNIX cksum.
 Valor de retorno: *Integer* - com `mode` como *crc32*, *String* - com `mode` como *md5*, *sha256*.
 Veja supported platforms.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - *crc32* (padrão), *md5* ou *sha256*.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.cksum[/etc/passwd]
```

Exemplos de valores retornados (crc32/md5/sha256 respectivamente):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Recupera o conteúdo de um arquivo[7].
 Valor de retorno: *Text*.
 Veja supported platforms.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **encoding** - o identificador da página de código identifier.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode representar um risco de segurança.
- O valor de retorno é limitado a 16 MB (incluindo espaços em branco finais, que são truncados); database limits 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.
 Consulte as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **types incl** - a lista de tipos de arquivo a incluir, valores possíveis:
  - *file* (arquivo regular, padrão (se `types_excl` não estiver definido));
  - *dir* (diretório);
  - *sym* (link simbólico);
  - *sock* (socket);
  - *bdev* (dispositivo de bloco);
  - *cdev* (dispositivo de caractere);
  - *fifo* (FIFO);
  - *dev* (sinônimo de ”bdev,cdev”);
  - *all* (todos os tipos mencionados, padrão se `types_excl` estiver definido).
- **types excl** - a lista de tipos de arquivo a excluir, veja `types_incl` para os valores possíveis (por padrão, nenhum tipo é excluído)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- Vários tipos devem ser separados por vírgula e o conjunto inteiro deve estar entre aspas "".
- Se o mesmo tipo estiver em ambos <types_incl> e <types_excl>, os arquivos desse tipo serão excluídos.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplos:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 Retorna informações sobre um arquivo.
 Valor de retorno: *string JSON*.
 Veja plataformas suportadas.

Parâmetro:

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

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- Tipos de arquivo suportados em sistemas do tipo UNIX: 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

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Exemplo de valor retornado:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Recupera o proprietário de um arquivo.
 Valor de retorno: *String*.
 Consulte plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **ownertype** - *user* (padrão) ou *group* (somente Unix);
- **resulttype** - *name* (padrão) ou *id*; para id - retorna uid/gid no Unix, SID no Windows.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplos:

```
vfs.file.owner[/tmp/zabbix_server.log] #retorna o proprietário do arquivo /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #retorna o ID do proprietário do arquivo /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 Retorna uma string de 4 dígitos contendo o número octal com as permissões UNIX.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux. O item pode funcionar em outras plataformas 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 que contém 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 que descreve o padrão necessário;
- **encoding** - o identificador da página de código identifier;
- **start line** - o número da primeira linha a ser pesquisada (por padrão, a primeira linha do arquivo);
- **end line** - o número da última linha a ser pesquisada (por padrão, a última linha do arquivo);
- **output** - um template opcional de formatação de saída. A sequência de escape **\0** é substituída pela parte correspondente do texto (do primeiro caractere em que a correspondência começa até o caractere em que a correspondência termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou por uma string vazia se N exceder o número de grupos capturados).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode representar um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.
- Apenas a primeira linha correspondente é retornada.
- Uma string vazia é retornada se nenhuma linha corresponder à expressão.
- A marca de ordem de bytes (BOM) é excluída da saída.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.

Exemplos:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] → obtendo o ID do usuário *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Encontrar uma string no arquivo[7].
 Valores de retorno: 0 - correspondência não encontrada; 1 - encontrada.
 Veja plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **regexp** - uma expressão regular que descreve o padrão necessário;
- **encoding** - o identificador da página de código;
- **start line** - o número da primeira linha a ser pesquisada (primeira linha do arquivo por padrão);
- **end line** - o número da última linha a ser pesquisada (última linha do arquivo por padrão).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.
- A marca de ordem de bytes (BOM) é ignorada.

Exemplo:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 O tamanho do arquivo (em bytes).
 Valor de retorno: *Integer*.
 Veja supported platforms.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - valores possíveis:
  - *bytes* (padrão);

    - *lines* (linhas em branco também são contadas).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode representar um risco de segurança.
- O arquivo deve ter permissões de leitura para o usuário *zabbix*.
- O limite de tamanho do arquivo depende do large file support.

Exemplo:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Informações sobre o tempo do arquivo.
 Valor de retorno: *Integer* (timestamp Unix).
 Consulte as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - valores possíveis:
    - *modify* (padrão) - a última vez em que o conteúdo do arquivo foi modificado (padrão);
    - *access* - a última vez em que o arquivo foi lido;
    - *change* - a última vez em que as propriedades do arquivo foram alteradas.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode representar um risco de segurança.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 A lista de sistemas de arquivos montados com seu tipo e opções de montagem. Usado para descoberta de baixo nível.
 Valor de retorno: *string JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get[<mode>,<mountpoint>]

 A lista de sistemas de arquivos montados com seu tipo e opções de montagem e, no modo *full*, o espaço em disco disponível e as estatísticas de inode. Pode ser usada para descoberta de baixo nível.
 Valor retornado: *string JSON*.
 Plataformas compatíveis: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parâmetros:

- **mode** - valores possíveis:
    - *full* (padrão) - propriedades do sistema de arquivos com estatísticas de espaço em disco e inode;
    - *short* - somente as propriedades do sistema de arquivos, sem estatísticas de espaço em disco e inode;
- **mountpoint** - o ponto de montagem para o qual os dados devem ser retornados (por exemplo, `/var` ou `C:`). Se não for especificado, todos os sistemas de arquivos montados serão retornados.

Comentários:

- No modo *short*, somente o nome, o tipo e as opções de montagem do sistema de arquivos são retornados (no Windows: nome, tipo, rótulo e tipo de unidade). Os sistemas de arquivos não são consultados para obter estatísticas; em sistemas semelhantes ao Unix, somente a lista de sistemas de arquivos montados é lida (por exemplo, `/proc/mounts` no Linux), portanto, a verificação não aciona a montagem automática e não é afetada por sistemas de arquivos de rede que não respondem. Isso torna o modo *short* adequado para descoberta de baixo nível.
- O valor de **mountpoint** deve corresponder exatamente à propriedade `fsname`. No Windows, a comparação não diferencia maiúsculas de minúsculas; especifique o ponto de montagem sem a barra invertida final (por exemplo, `C:`).
- Se **mountpoint** for especificado, os dados serão retornados somente para esse ponto de montagem. Se nenhum sistema de arquivos montado corresponder a ele, uma matriz vazia (`[]`) será retornada.
- A saída pode conter várias entradas com o mesmo `fsname` se mais de um sistema de arquivos estiver montado no mesmo diretório.
- No modo *full*, o espaço em disco disponível e as estatísticas de inode serão retornados somente se o usuário do Zabbix agent tiver permissão de travessia em todos os diretórios no caminho do ponto de montagem; caso contrário, o sistema de arquivos será omitido da saída. Isso não se aplica ao modo *short*, que pode, portanto, listar sistemas de arquivos ausentes na saída do modo *full*.

- As propriedades `fslabel` e `fsdrivetype` são retornadas somente no Windows; as propriedades `inodes` e `options` não são retornadas no Windows.
- Sistemas de arquivos com a contagem de inode igual a zero, o que pode ocorrer em sistemas de arquivos com inodes dinâmicos (por exemplo, btrfs), também são reportados.
- Consulte também: Descoberta de sistemas de arquivos montados.

Exemplos:

```
vfs.fs.get #todos os sistemas de arquivos montados com estatísticas; igual a vfs.fs.get[full]
vfs.fs.get[short] #todos os sistemas de arquivos montados sem estatísticas
vfs.fs.get[full,/var] #estatísticas somente para o ponto de montagem /var
vfs.fs.get[short,C:] #somente propriedades do volume C: (Windows)
```

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:
  - *total* - quantidade total de memória;
  - tipos de memória específicos da plataforma: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - estimativas em nível de usuário de quanta memória está sendo usada 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>,<redirect_limit>]

 Obtém o conteúdo de uma página da web.
 Valor retornado: código-fonte da página da web como texto (incluindo cabeçalhos).
 Consulte plataformas compatíveis.

Parâmetros:

- **host** - o nome do host ou a URL (como `scheme://host:port/path`, em que apenas *host* é obrigatório). Esquemas de URL permitidos: *http*, *https*[4]. A ausência de um esquema será tratada como *http*. Se uma URL for especificada, `path` e `port` deverão estar vazios. A especificação de nome de usuário/senha ao conectar-se a servers que exigem autenticação, por exemplo: `http://user:password@www.example.com`, só é possível com suporte ao cURL [4]. Punycode é compatível com nomes de host.
- **path** - o caminho para um documento HTML (o padrão é /);
- **port** - o número da porta (o padrão é 80 para HTTP)
- **redirect_limit** - o número máximo de redirecionamentos a seguir [4], de 0 a 1000 (o padrão é 0 - os redirecionamentos não são seguidos).

Comentários:

- Este item se torna incompatível se o recurso especificado em `host` não existir ou estiver indisponível, ou se `redirect_limit` for maior que 0 e o número de redirecionamentos excedê-lo.
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. No entanto, para um endereço IPv6, o Zabbix agent deve ser compilado com o suporte a IPv6 habilitado.
- Se os redirecionamentos forem seguidos, o valor retornado conterá os cabeçalhos de cada resposta de redirecionamento, seguidos pelos cabeçalhos e pelo corpo da resposta final.

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.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 O tempo de carregamento de uma página web completa (em segundos).
 Valor retornado: *Float*.
 Consulte plataformas compatíveis.

Parâmetros:

- **host** - o nome do host ou a URL (como `scheme://host:port/path`, em que apenas *host* é obrigatório). Esquemas de URL permitidos: *http*, *https*[4]. A ausência de um esquema será tratada como *http*. Se uma URL for especificada, `path` e `port` deverão estar vazios. A especificação de nome de usuário/senha ao conectar-se a servers que exigem autenticação, por exemplo: `http://user:password@www.example.com`, só é possível com suporte a cURL [4]. Punycode é compatível com nomes de host.
- **path** - o caminho para um documento HTML (o padrão é /);
- **port** - o número da porta (o padrão é 80 para HTTP)
- **redirect_limit** - o número máximo de redirecionamentos a seguir [4], de 0 a 1000 (o padrão é 0 - os redirecionamentos não são seguidos).

Comentários:

- Este item se torna incompatível se o recurso especificado em `host` não existir ou estiver indisponível, ou se `redirect_limit` for maior que 0 e o número de redirecionamentos excedê-lo.
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. No entanto, para um endereço IPv6, o Zabbix agent deve ser compilado com o suporte a IPv6 habilitado.
- Se os redirecionamentos forem seguidos, o tempo de carregamento incluirá o tempo gasto em todos os redirecionamentos.

Exemplos:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Encontra uma string na página da web.
 Valor de retorno: a string correspondente ou, conforme especificado pelo parâmetro opcional `output`.
 Consulte plataformas compatíveis.

Parâmetros:

- **host** - o nome do host ou a URL (como `scheme://host:port/path`, em que apenas *host* é obrigatório). Esquemas de URL permitidos: *http*, *https*[4]. Se o esquema não for especificado, será considerado *http*. Se uma URL for especificada, `path` e `port` deverão estar vazios. A especificação de nome de usuário/senha ao conectar-se a servers que exigem autenticação, por exemplo: `http://user:password@www.example.com`, só é possível com suporte a cURL [4]. Punycode é compatível com 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 que descreve o padrão necessário.
- **length** - o número máximo de caracteres a serem retornados;
- **output** - um template opcional para formatação da saída. A sequência de escape **\0** é substituída pela parte correspondente do texto (do primeiro caractere em que a correspondência começa até o caractere em que termina), enquanto uma sequência de escape **\N** (em que N=1...9) é substituída pelo N-ésimo grupo correspondente (ou por uma string vazia se N exceder o número de grupos capturados).
- **redirect_limit** - o número máximo de redirecionamentos a seguir [4], de 0 a 1000 (o padrão é 0 - os redirecionamentos não são seguidos).

Comentários:

- Este item se torna incompatível se o recurso especificado em `host` não existir ou estiver indisponível, ou se `redirect_limit` for maior que 0 e o número de redirecionamentos excedê-lo.
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. No entanto, para um endereço IPv6, o Zabbix agent deve ser compilado com o suporte a IPv6 habilitado.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.
- Se os redirecionamentos forem seguidos, os cabeçalhos de cada resposta de redirecionamento também serão pesquisados.

Exemplos:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 Os metadados do host do agent.
 Valor de retorno: *String*.
 Consulte supported platforms.

Retorna o valor dos parâmetros `HostMetadata` ou `HostMetadataItem`, ou uma string vazia se nenhum estiver definido.

agent.hostname

 O nome do host do agent.
 Valor de retorno: *String*.
 Consulte supported platforms.

Retorna:

- Como verificação passiva - o nome do primeiro host listado no parâmetro Hostname do arquivo de configuração do agent;
- Como verificação ativa - o nome do hostname atual.

agent.ping

 A verificação de disponibilidade do agent.
 Valor de retorno: Nothing - indisponível; 1 - disponível.
 Consulte supported platforms.

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 server ou proxy do Zabbix. Se `ip` e `port` forem especificados, as métricas são obtidas da instância remota; caso contrário, da instância local.
 Valor de retorno: *string JSON*.
 Consulte as plataformas suportadas.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede de 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, consulte Monitoramento remoto de estatísticas do Zabbix.
- Observe que a solicitação de estatísticas só será aceita a partir dos endereços listados no parâmetro `StatsAllowedIP` do server/proxy na instância de destino.

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

 Retorna o número de itens monitorados na fila que estão atrasados no server ou proxy do Zabbix. Se `ip` e `port` forem especificados, as métricas são obtidas da instância remota; caso contrário, da instância local.
 Valor de retorno: *string JSON*.
 Veja as plataformas suportadas.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede de servers/proxies a serem consultados remotamente (o padrão é `127.0.0.1`);
- **port** - a porta do server/proxy a ser consultado remotamente (o padrão é 10051)
- **queue** - constante (deve ser usada como está)
- **from** - atrasado por pelo menos (o padrão é 6 segundos)
- **to** - atrasado por no máximo (o padrão é infinito)

Observe que a solicitação de stats só será aceita a partir dos endereços listados no parâmetro 'StatsAllowedIP' de server/proxy na instância de destino.

Notas de rodapé

[1] Uma observação específica do Linux.
O Zabbix agent deve ter acesso somente leitura ao sistema de arquivos `/proc`.
Os patches do kernel de www.grsecurity.org limitam os direitos de acesso de usuários sem privilégios.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: o Zabbix agent 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, quando usam um intervalo de atualização de 3 horas ou mais, sempre retornarão '0'.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: se o valor padrão `all` for usado como primeiro parâmetro, a chave retornará estatísticas resumidas, incluindo todos os dispositivos de bloco, como sda e sdb, e suas partições (sda1, sda2, sdb3...), além de vários dispositivos (RAID MD) baseados nesses dispositivos de bloco/partições e volumes lógicos (LVM) baseados nesses dispositivos de bloco/partições.
Nesses casos, os valores retornados devem ser considerados apenas valores relativos (dinâmicos ao longo do tempo), e não valores absolutos.

[4] HTTPS e o seguimento de redirecionamentos (`redirect_limit` maior que 0) são compatíveis somente quando o Zabbix agent é compilado com suporte a cURL; caso contrário, o item se torna incompatível. O Zabbix agent 2 não requer cURL para nenhum dos dois.

[5] Os valores `bytes` e `errors` não são compatíveis com interfaces de loopback em sistemas Solaris até, inclusive, o Solaris 10 6/06, pois as estatísticas de bytes, erros e utilização não são armazenadas e/ou relatadas pelo kernel.
No entanto, se você estiver monitorando um sistema Solaris via net-snmp, os valores poderão ser retornados, pois o net-snmp contém código legado do cmu-snmp, datado de 1997, que, ao não conseguir ler os valores de bytes das estatísticas da interface, retorna o contador de pacotes (que existe nas interfaces de loopback) multiplicado por um valor arbitrário de 308.
Isso pressupõe que o comprimento médio de um pacote seja de 308 octetos, uma estimativa bastante imprecisa, pois o limite de MTU nos sistemas Solaris para interfaces de loopback é de 8892 bytes.
Não se deve presumir que esses valores estejam corretos ou mesmo que sejam aproximadamente precisos.
São estimativas aproximadas.
O Zabbix agent não faz suposições, 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 tal como 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 quiser restringir o acesso a arquivos específicos que contenham informações confidenciais, execute o Zabbix agent com 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 compatíveis (identificadores de página de código) pode ser encontrada na documentação da libiconv (GNU Project) ou na documentação do Microsoft Windows SDK para "Code Page Identifiers". Observe que, às vezes, a Microsoft marca algumas páginas de código como "disponíveis apenas para aplicativos gerenciados", o que significa que elas são expostas somente por meio do runtime do .NET e não estão disponíveis pelas APIs nativas do Win32. O Zabbix agent implementa sua própria lógica de conversão de codificação; portanto, essas páginas de código são compatíveis com o Zabbix agent mesmo quando as funções nativas do Windows não as fornecem.

Se nenhuma codificação for especificada no parâmetro `encoding`, as seguintes estratégias de resolução serão aplicadas:

- Se a codificação não for especificada (ou for uma string vazia), será considerado UTF-8 e os dados serão processados "como estão";
- Análise de BOM - aplicável aos itens `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 a BOM não estiver presente, a resolução padrão (consulte acima) será aplicada.

Solução de problemas de itens do agent

No caso de verificações passivas, para evitar que o item não receba nenhum valor porque a solicitação do server ao agent atinge o tempo limite primeiro, observe o seguinte:

- Quando a versão do agent é mais antiga que a versão do server, o valor de `Timeout` na configuração do item (ou o tempo limite global) pode precisar ser maior que o valor de `Timeout` no arquivo de configuração do agent.
- Quando a versão do agent é 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.

Zabbix agent 2

O Zabbix agent 2 suporta todas as chaves de item suportadas pelo Zabbix agent em Unix e Windows. Esta página fornece detalhes sobre as chaves de item adicionais, que você pode usar somente com o Zabbix agent 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 de item	Descrição	Plugin
ceph.df.details	O uso de dados do cluster e a distribuição entre pools.	Ceph (carregável)
ceph.osd.stats	Estatísticas agregadas e por OSD.	
ceph.osd.discovery	A lista de OSDs descobertos.	
ceph.osd.dump	Os limites de uso e os 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.	

Chave de item	Descrição	Plugin
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 do Docker está ativo ou não.	
ember.get	Retorna o resultado do dispositivo solicitado.	Ember+ (carregável)
memcached.ping	Testa se uma conexão está ativa ou não.	Memcached
memcached.stats	Obtém a saída do comando STATS.	
mongodb.cfg.discovery	Retorna uma lista de servidores de configuração descobertos.	MongoDB (carregável)
mongodb.collection.stats	Retorna várias estatísticas de armazenamento para uma coleção específica.	
mongodb.collections.discovery	Retorna uma lista de coleções descobertas.	
mongodb.collections.usage	Retorna as estatísticas de uso das coleções.	
mongodb.connpool.stats	Retorna informações sobre as conexões de saída abertas da instância atual do banco de dados para outros membros do cluster fragmentado ou do conjunto de réplicas.	
mongodb.db.stats	Retorna as estatísticas que refletem o estado de um determinado banco de dados.	
mongodb.db.discovery	Retorna uma lista de bancos de dados descobertos.	
mongodb.jumbo_chunks.count	Retorna a contagem de jumbo chunks.	
mongodb.oplog.stats	Retorna o status do conjunto de réplicas, usando dados coletados do oplog.	
mongodb.ping	Testa se uma conexão está ativa ou não.	
mongodb.rs.config	Retorna a configuração atual do conjunto de réplicas.	
mongodb.rs.status	Retorna o status do conjunto de réplicas do ponto de vista do membro em que 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 (carregável)
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 participam de um grupo de disponibilidade Always On e réplica (primária ou secundária) e que estão localizados no server ao qual a conexão foi estabelecida.	
mssql.mirroring.get	Retorna informações de espelhamento.	
mssql.nonlocal.db.get	Retorna bancos de dados que participam de um grupo de disponibilidade Always On e réplica (primária ou secundária) localizados em outros servers (o banco de dados não é local à 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 quorum.	
mssql.quorum.member.get	Retorna os membros do quorum.	
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 globais de status.	
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.	Network
nvml.device.count	O número de dispositivos GPU.	NVIDIA GPU (carregável)

Chave de item	Descrição	Plugin
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 sobre erros de memória ECC do dispositivo GPU.	
nvml.device.errors.register	Informações sobre erros 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.	
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 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 discos do Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Retorna a lista de grupos de discos ASM.	
oracle.archive.info	As estatísticas dos logs de archive.	
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 data files.	
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 no control file.	
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 a 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 de autovacuum.	PostgreSQL (carregável)
pgsql.archive	As informações sobre arquivos arquivados.	
pgsql.bgwriter	O número combinado de checkpoints para o cluster de banco de dados, detalhado por tipo de checkpoint.	
pgsql.cache.hit	A taxa de acerto do cache de buffers 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 com bloat 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.	

Chave de item	Descrição	Plugin
pgsql.dbstat.sum	Os dados resumidos de 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 consultas por tempo de execução.	
pgsql.replication.count	O número de servidores standby.	
pgsql.replication.process	O atraso de flush, o atraso de write e o atraso de replay para cada processo de envio.	
pgsql.replication.process.discovery	A descoberta do nome do processo de replicação.	
pgsql.replication.recovery_role	O status de recuperação.	
pgsql.replication.status	O status da replicação.	
pgsql.replication_lag.b	O atraso de replicação em bytes.	
pgsql.replication_lag.sec	O atraso de replicação em segundos.	
pgsql.uptime	O tempo de atividade do PostgreSQL em milissegundos.	
pgsql.version	Retorna a versão do PostgreSQL.	
pgsql.wal.stat	As estatísticas do WAL.	
redis.config	Obtém os parâmetros de configuração de uma instância do 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 do slow log desde que o Redis foi iniciado.	
smart.attribute.discovery	Retorna uma lista de atributos de dispositivos S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Retorna uma lista de dispositivos S.M.A.R.T.	
smart.disk.get	Retorna todas as propriedades disponíveis dos dispositivos S.M.A.R.T.	
systemd.unit.get	Retorna todas as propriedades de uma unidade systemd.	Systemd
systemd.unit.info	Informações da unidade systemd.	
systemd.unit.discovery	A lista de unidades systemd e seus detalhes.	
web.certificate.get	Valida os certificados e retorna os detalhes do certificado.	Web certificates

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

ceph.df.details[connString,<user>,<apikey>]

 O uso de dados do cluster e a distribuição entre pools.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Estatísticas agregadas e por OSD.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 A lista de OSDs descobertos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Os limites de uso e status dos OSDs.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.ping[connString,<user>,<apikey>]

 Testa se uma conexão com o Ceph pode ser estabelecida.
 Valor de retorno: *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração); *1* - a conexão foi bem-sucedida.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 A lista de pools descobertos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.status[connString,<user>,<apikey>]

 O status geral do cluster.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

docker.container_info[<ID>,<info>]

 Informações de baixo nível sobre um container.
 Valor de retorno: A saída da chamada de API ContainerInspect serializada como JSON.

Parâmetros:

- **ID** - o ID ou nome do container;
- **info** - a quantidade de informações retornadas. Valores suportados: *short* (padrão) ou *full*.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.container_stats[<ID>]

 As estatísticas de uso de recursos do container.
 Valor de retorno: A saída da chamada de API ContainerStats e a porcentagem de uso 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.cfg.discovery[connString,<user>,<password>]

 Retorna uma lista de servidores de configuração descobertos.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

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 fragmentado ou conjunto de réplicas.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Retorna as estatísticas que refletem o estado do sistema 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: depende do conteúdo do tópico. Se forem usados curingas, 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 a porta padrão (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 um active check (tipo de item "Zabbix agent (active)");
- Certificados de criptografia TLS podem ser usados ao salvá-los em um local padrão (por exemplo, o 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, seção *Consultas personalizadas*.

mysql.db.discovery[connString,<user>,<password>]

 Retorna a lista de bancos de dados MySQL. Usado para descoberta de baixo nível.
 Valor de retorno: O resultado da consulta SQL ”show databases” no formato JSON LLD.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.db.size[connString,<user>,<password>,<database name>]

 O tamanho do banco de dados em bytes.
 Valor de retorno: Resultado da consulta SQL ”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” para um banco de dados específico em bytes.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL;
- **database name** - o nome do banco de dados.

mysql.get_status_variables[connString,<user>,<password>]

 Valores das variáveis de status globais.
 Valor de retorno: Resultado da consulta SQL ”show global status” 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` ou `SHOW REPLICA STATUS` no formato JSON de LLD.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 O status da replicação.
 Valor de retorno: resultado da consulta SQL `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS` em formato JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL;
- **master host** - o nome do host mestre de replicação. Se nenhum for encontrado, um erro será retornado. Se este parâmetro não for especificado, todos os hosts serão retornados.

mysql.version[connString,<user>,<password>]

 A versão do MySQL.
 Valor de retorno: *String* (com a versão da instância do MySQL).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Executa uma consulta DNS e retorna informações detalhadas do registro DNS.
 Este item é uma versão estendida do item `net.dns.record` do agent do Zabbix, com suporte a mais tipos de registro e flags personalizáveis.
 Valores de retorno: *JSON object*

Parâmetros:

- **ip** - o endereço IP do servidor DNS (deixe em branco para usar 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 da solicitação, em segundos (o padrão é 1 segundo);
- **count** - o número de tentativas da solicitação (o padrão é 2);
- **protocol** - o protocolo usado para realizar consultas DNS: *udp* (padrão) ou *tcp*;
- **flags** - um ou mais argumentos separados por vírgula a serem passados para uma consulta.

Comentários:

- Os valores possíveis para `type` são: *A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*, *TALINK*, *CDS*, *CDNSKEY*, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY*. Observe que os valores devem estar somente em maiúsculas; valores em minúsculas ou em caixa mista não são suportados.
- Para consultas DNS reversas (quando `type` está definido como *PTR*), você pode fornecer o nome DNS nos formatos invertido e não invertido (veja os exemplos abaixo). Observe que, quando um registro PTR é solicitado, o nome DNS na verdade é 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 exigem *edns0*. Observe que os valores devem estar somente em minúsculas; valores em maiúsculas ou em caixa mista não são suportados.
- Nomes de domínio internacionalizados não são suportados; use nomes codificados em IDNA.
- A saída é um objeto contendo informações do registro DNS com base nos parâmetros fornecidos (veja mais detalhes).

Exemplos:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 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 discos 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 são ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.

- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle;
- **diskgroup** - o nome do grupo de discos ASM a ser consultado.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de grupos de discos 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 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 especificado 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **database** - o nome do banco de dados a ser consultado.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 O resultado de uma consulta personalizada.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada, e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta acrescentar 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 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 a extensão;
- **args** - um ou vários argumentos separados por vírgula para passar à 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 em JSON novamente.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Retorna as estatísticas dos arquivos de dados.
 Valor retornado: *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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta anexar uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de bancos de dados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta acrescentar 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 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 retornado: *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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 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 levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

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

 As informações dos Bancos de Dados Plugáveis (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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte para acrescentar 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **database** - o nome do banco de dados a ser consultado.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de PDBs. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada, e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.

- **user** - o nome de usuário do Oracle; suporta acrescentar 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 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta acrescentar 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do 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 está interrompida (se houver qualquer erro apresentado, incluindo problemas de AUTH e de 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta acrescentar 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do 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 retornado: *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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.

- **user** - o nome de usuário do Oracle; suporta acrescentar 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle.

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

 As informações do arquivo de log do arquivo de controle.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada, e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 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 Área Global do Sistema (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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle;
- **lockMaxTime** - a duração máxima do bloqueio da sessão, em segundos, para contar a sessão como bloqueada por tempo prolongado. Padrão: 600 segundos.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Retorna um conjunto de valores de métricas do sistema.
 Valor retornado: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - a 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://"), uma porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e o valor será tratado como URI de qualquer forma;
  - nome da sessão - se esse 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 iniciais são ignorados). Nesse caso, a opção ResolveTNS não será considerada.
- **user** - o nome de usuário do Oracle; permite acrescentar 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 do Oracle;
- **duration** - o intervalo de captura (em segundos) dos valores de 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 levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 iniciais ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta anexar uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle;
- **tablespace** - nome do tablespace a ser consultado. Padrão (se deixado vazio e `type` estiver definido):
  - "TEMP" (se `type` estiver definido como "TEMPORARY");
  - "USERS" (se `type` estiver definido como "PERMANENT").
- **type** - o tipo do tablespace a ser consultado. Padrão (se `tablespace` estiver definido): "PERMANENT".
- **conname** - nome do contêiner para o qual as informações são necessárias.

Se `tablespace`, `type` ou `conname` forem omitidos, o item retornará estatísticas de tablespace para todos os contêineres correspondentes (incluindo PDBs e CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Retorna uma lista de tablespaces. Usado para low-level discovery.
 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 schema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será levada em consideração e, ainda assim, será tratada como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; oferece suporte à 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do 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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **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 e não deve conter 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 server de banco de dados.
 Valor retornado: *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 levada em consideração e, ainda assim, será tratado como URI;
  - nome da sessão - se esse 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 colchete de abertura "(" (espaços à esquerda ignorados). Nesse caso, a opção ResolveTNS não será levada em consideração.
- **user** - o nome de usuário do Oracle; suporta acrescentar 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 espaço à direita).
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle.

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

 O número de workers de autovacuum.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 Informações sobre arquivos arquivados.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` será usado).

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

 O número combinado de checkpoints para o cluster de banco de dados, discriminado por tipo de checkpoint.
 Valor de retorno: *JSON object*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 A taxa de acerto do cache de buffer do PostgreSQL.
 Valor de retorno: *Float*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 Retorna conexões por tipo.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Retorna o resultado de uma consulta personalizada.
 Valor de retorno: *JSON object*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database de postgresql.conf é usado);
- **queryName** - o nome de uma consulta personalizada, deve corresponder ao nome do arquivo SQL sem a 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 (número de transações).
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 O número de tabelas inchadas por banco de dados.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - a URI ou nome da sessão;

- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` será usado).

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

 A lista de bancos de dados PostgreSQL. Usado para low-level discovery.
 Valor de retorno: *JSON object*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`).

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

 O tamanho do banco de dados em bytes.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 Coleta as estatísticas por banco de dados. Usado para descoberta em baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`).

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

 Os dados resumidos de todos os bancos de dados em um cluster.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 Informações sobre locks concedidos por banco de dados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`).

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

 A idade do XID mais antigo (número de transações).
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

pgsql.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 AUTH e de configuração).

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`).

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

 Consulta métricas por tempo de execução.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`);
- **time period** - 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** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf`).

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

 Retorna a versão do PostgreSQL.
 Valor de retorno: *String*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

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

 As estatísticas de WAL.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - a URI ou o nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` é usado).

redis.config[connString,<password>,<pattern>]

 Obtém os parâmetros de configuração de uma instância Redis que correspondem ao padrão.
 Valor de retorno: *JSON* - se um padrão de estilo glob foi usado; valor único - se um padrão não 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).

Comentário:

- Aviso de segurança: `redis.config` executa o comando Redis `CONFIG GET` e pode retornar parâmetros de configuração sensíveis (por exemplo, `requirepass`, `masterauth`, configurações relacionadas a TLS e outros segredos). Usar o padrão padrão (`*`) ou padrões amplos pode expor esses segredos a qualquer pessoa com permissão para visualizar valores de item no Zabbix.

redis.info[connString,<password>,<section>]

 Obtém a saída do comando INFO.
 Valor de retorno: *JSON* - a saída é serializada como JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis;
- **section** - a seção de informações (*default* por padrão).

redis.ping[connString,<password>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTH e configuração).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis.

redis.slowlog.count[connString,<password>]

 O número de entradas de log lento desde que o Redis foi iniciado.
 Valor de retorno: *Inteiro*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis.

smart.attribute.discovery

 Retorna uma lista de atributos de 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}`;
- Tipos de unidade HDD, SSD e NVME são suportados. As unidades podem estar sozinhas ou combinadas em um RAID. `{#NAME}` terá um complemento no 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: *JSON object*.

Parâmetro:

- **type** - especifica um valor a ser verificado para 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, `{#RAIDTYPE}` ficará vazio. `{#NAME}` terá um complemento no caso de RAID, por exemplo: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.get[<path>,<raid type>]

 Retorna todas as propriedades disponíveis dos dispositivos S.M.A.R.T.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **path** - o caminho do disco, a macro `{#PATH}` pode ser usada como valor;
- **raid_type** - o tipo de RAID, a macro `{#RAID}` pode ser usada como valor

Comentários:

- Tipos de unidade HDD, SSD e NVME são suportados. As unidades podem estar sozinhas ou combinadas em um RAID;
- Os dados incluem a versão do smartctl e os argumentos da chamada, além de campos adicionais:
disk_name - contém o nome com os complementos necessários para a descoberta de RAID, por exemplo: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - contém o tipo de disco HDD, SSD ou NVME, por exemplo: `{"disk_type": "ssd"}`;
- Se nenhum parâmetro for especificado, o item retornará informações sobre todos os discos.

systemd.unit.get[unit name,<interface>]

 Retorna todas as propriedades de uma unit do systemd.
 Valor de retorno: *JSON object*.

Parâmetros:

- **unit name** - o nome da unit (talvez você queira usar a macro `{#UNIT.NAME}` no item prototype para descobrir o nome);
- **interface** - o tipo de interface da unit, 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[unit name,<property>,<interface>]

 Informações de uma unidade do systemd.
 Valor retornado: *String*.

Parâmetros:

- **unit name** - o nome da unidade (você pode usar a macro `{#UNIT.NAME}` no protótipo de item para descobrir o nome);
- **property** - propriedade da unidade (por exemplo, ActiveState (padrão), LoadState, Description);
- **interface** - tipo de interface da unidade (por exemplo, Unit (padrão), Socket, Service).

Comentários:

- Este item é compatível somente com a plataforma Linux;
- Este item permite recuperar uma propriedade específica de um tipo específico de interface, conforme descrito na API dbus.

Exemplos:

```
systemd.unit.info["{#UNIT.NAME}"] #coleta informações sobre o estado ativo (active, reloading, inactive, f
systemd.unit.info["{#UNIT.NAME}",LoadState] #coleta informações sobre o estado de carregamento das unidade
systemd.unit.info[mysqld.service,Id] #recupera o nome técnico do serviço (mysqld.service)
systemd.unit.info[mysqld.service,Description] #recupera a descrição do serviço (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #recupera a última vez em que o serviço entrou no e
systemd.unit.info[dbus.socket,NConnections,Socket] #coleta o número de conexões desta unidade de socket
```

systemd.unit.discovery[<type>]

 Lista de unidades systemd e seus detalhes. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **type** - valores possíveis: *all*, *automount*, *device*, *mount*, *path*, *service* (padrão), *socket*, *swap*, *target*.

Este item é suportado apenas na plataforma Linux.

web.certificate.get[hostname,<port>,<address>]

 Valida os certificados e retorna os detalhes do certificado.
 Valor de retorno: *JSON object*.

Parâmetro:

- **hostname** - pode ser IP ou DNS.
Pode conter o esquema da URL (*apenas https*), o caminho (ele será ignorado) e a porta.
Se uma porta for fornecida tanto no primeiro quanto no segundo parâmetro, os valores devem corresponder.
Se address (o 3º parâmetro) for especificado, o hostname será usado apenas para SNI e verificação de hostname;
- **port** - o número da porta (padrão é 443 para HTTPS);
- **address** - pode ser IP ou DNS. Se especificado, será usado para a conexão, e o hostname (o 1º parâmetro) será usado para SNI e verificação de host. Caso o 1º parâmetro seja um IP e o 3º parâmetro seja DNS, o 1º parâmetro será usado para a conexão, e o 3º parâmetro será usado para SNI e verificação de host.

Comentários:

- Este item se tornará 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 da 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 subject do certificado.
    - *public_key_algorithm*: o algoritmo usado para a chave pública (por exemplo, `RSA`).
    - *alternative_names*: nomes alternativos do subject (se presentes), caso contrário `null`.
  - result: contém o resultado da validação.
    - *value*: o status da validação (veja os valores possíveis abaixo).
    - *message*: mensagem detalhada da validação (por exemplo, `"certificate verified successfully"`).
  - *sha1_fingerprint*: a impressão digital SHA-1 do certificado.
  - *sha256_fingerprint*: a impressão digital SHA-256 do certificado.
- O campo `$.result.value` indica o resultado da validação do certificado. Valores possíveis incluem:
  - *valid* - o certificado é válido e confiável.
  - *valid-but-self-signed* - o certificado é válido, mas autoassinado, o que significa que seu subject corresponde ao seu issuer.
  - *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"
}
```


Windows Zabbix agent

Visão geral

Os items do agent Zabbix para Windows são apresentados em duas listas:

- Itens compartilhados - as chaves de item que são compartilhadas com o agent Zabbix para UNIX;
- Itens específicos do Windows - as chaves de item que são suportadas **apenas** no Windows.

Observe que todas as chaves de item suportadas pelo agent Zabbix no Windows também são suportadas pela nova geração do agent Zabbix 2. Veja as chaves de item adicionais que você pode usar apenas com o agent 2.

Veja também: Permissões mínimas para itens do Windows

Itens compartilhados

A tabela abaixo lista itens do Zabbix agent com suporte no Windows e compartilhados com o Zabbix agent UNIX:

- A chave do item é um link para os detalhes completos do item do Zabbix agent UNIX
- Comentários do item relevantes para Windows estão incluídos

Item key	Description	Item group
log	O monitoramento de um arquivo de log. Este item não é suportado para o Windows Event Log. O parâmetro `persistent_dir` não é suportado no Windows.	Log monitoring
log.count	A contagem de linhas correspondentes em um arquivo de log monitorado. Este item não é suportado para o Windows Event Log. O parâmetro `persistent_dir` não é suportado no Windows.	
logrt	O monitoramento de um arquivo de log rotacionado. Este item não é suportado para o Windows Event Log. O parâmetro `persistent_dir` não é suportado no Windows.	
logrt.count	A contagem de linhas correspondentes em um arquivo de log monitorado e rotacionado. Este item não é suportado para o Windows Event Log. O parâmetro `persistent_dir` não é suportado no Windows.	

Item key	Description	Item group
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 Zabbix agent 2.	Network
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 Zabbix agent 2.	
net.dns.record	Executa uma consulta DNS. Os parâmetros `ip`, `timeout` e `count` são ignorados no Windows, a menos que esteja usando o Zabbix agent 2.	
net.if.discovery	A lista de interfaces de rede. Algumas versões do Windows (por exemplo, Server 2008) podem exigir a instalação das atualizações mais recentes para oferecer suporte a 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 no Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Nomes de interface com vários bytes no Windows são suportados. Você pode obter descrições de interfaces de rede no Windows com 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 no Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Nomes de interface com vários bytes no Windows são suportados. Você pode obter descrições de interfaces de rede no Windows com itens net.if.discovery ou net.if.list.	
net.if.total	A soma das estatísticas de tráfego de entrada e saída em uma interface de rede. No Windows, o item obtém valores de contadores de 64 bits, se disponíveis. Os contadores de estatísticas de interface de 64 bits foram introduzidos no Windows Vista e no Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Nomes de interface com vários bytes no Windows são suportados. Você pode obter descrições de interfaces de rede no Windows com 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 estabelecer 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 Zabbix agent 2.	
net.tcp.service.perf	Verifica o desempenho de um serviço TCP. A verificação de LDAP e HTTPS no Windows é suportada apenas pelo Zabbix agent 2.	
net.tcp.socket.count	Retorna o número de sockets TCP que correspondem aos parâmetros. Este item é suportado no Linux pelo Zabbix agent, mas no Windows é suportado apenas pelo Zabbix agent 2 em Windows de 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 Zabbix agent, mas no Windows é suportado apenas pelo Zabbix agent 2 em Windows de 64 bits.	
proc.get	A lista de processos do sistema operacional e seus parâmetros. O parâmetro `cmdline` não é suportado no Windows.	Processes
proc.num	O número de processos. No Windows, apenas os parâmetros `name` e `user` são suportados.	
system.cpu.discovery	A lista de CPUs/núcleos de CPU detectados.	System

Item key	Description	Item group
system.cpu.load	A carga da CPU. Quando um processo coletor é iniciado no Zabbix agent, os seguintes contadores de desempenho são inicializados e depois 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 nas versões anteriores era o contador *Processor Time* (consulte 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 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.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 Zabbix agent, mas no Windows é suportado apenas pelo Zabbix agent 2. Observe que esta chave pode relatar tamanho/porcentagem incorretos do espaço de swap em plataformas Windows virtualizadas (VMware ESXi, VirtualBox). Nesse caso, você pode usar a chave `perf_counter[\700(_Total)\702]` para obter a porcentagem correta do espaço de swap.	
system.uname	Identificação do sistema. No Windows, o valor deste item é obtido das classes WMI `Win32_OperatingSystem` e `Win32_Processor`. O nome do SO (incluindo a edição) pode ser traduzido para o idioma de exibição do usuário. Em algumas versões do Windows, ele contém símbolos de marca registrada e espaços extras.	
system.uptime	O tempo de atividade do sistema em segundos.	
vfs.dir.count	A contagem de entradas do diretório. No Windows, links simbólicos de diretório são ignorados e links físicos são contados apenas uma vez.	Virtual file systems
vfs.dir.get	A lista de entradas do diretório. No Windows, links simbólicos de diretório são ignorados e links físicos são contados apenas uma vez.	
vfs.dir.size	O tamanho do diretório. No Windows, qualquer link simbólico é ignorado e links físicos são considerados apenas uma vez.	
vfs.file.cksum	A soma de verificação do arquivo, calculada pelo algoritmo UNIX cksum.	
vfs.file.contents	Recupera o conteúdo de um arquivo.	
vfs.file.exists	Verifica se o arquivo existe. No Windows, as aspas duplas precisam ser escapadas com barra invertida '\' e toda a chave do item deve estar entre aspas duplas ao usar o utilitário de linha de comando para chamar `zabbix_get.exe` ou `agent2`. Observe que o item pode ficar sem suporte 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	A soma de verificação MD5 do arquivo.	
vfs.file.owner	Recupera o proprietário de um arquivo.	
vfs.file.regexp	Recupera uma string no arquivo.	
vfs.file.regmatch	Localiza uma string no arquivo.	
vfs.file.size	O tamanho do arquivo.	

Item key	Description	Item group
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.	Virtual memory
web.page.get	Obtém o conteúdo de uma página web.	Web monitoring
web.page.perf	O tempo de carregamento de uma página web completa.	
web.page.regexp	Localiza 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 Zabbix agent (Zabbix agent ou Zabbix agent 2).	
agent.version	A versão do Zabbix agent.	
zabbix.stats	Retorna remotamente um conjunto de métricas internas do Zabbix server ou proxy.	
zabbix.stats (queue)	Retorna remotamente o número de itens monitorados na fila que estão atrasados no Zabbix server ou proxy.	

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.	
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 desem-penho
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>]

 O monitoramento do log de eventos.
 Valor de retorno: *Log*.

Parâmetros:

- **name** - o nome do canal do log de eventos (*Log Name* na interface gráfica do Event Viewer);
- **regexp** - uma expressão regular que descreve o padrão necessário (sensível a maiúsculas e minúsculas);
- **severity** - uma expressão regular que descreve a severidade (insensível a maiúsculas e minúsculas). Este parâmetro aceita uma expressão regular baseada nos seguintes valores: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (em execução no Windows Vista ou mais recente).
- **source** - uma expressão regular que descreve o identificador da origem (insensível a maiúsculas e minúsculas);
- **eventid** - uma expressão regular que descreve o(s) identificador(es) do evento (sensível a maiúsculas e minúsculas);
- **maxlines** - o número máximo de novas linhas por segundo que o agent enviará ao Zabbix server ou proxy. Este parâmetro substitui o valor de `MaxLinesPerSecond` em `zabbix_agentd.conf`.
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente).

Comentários:

- O item deve ser configurado como uma verificação ativa;
- O agent não consegue enviar eventos do log ”Forwarded events”;
- O Windows Eventing 6.0 é suportado;
- Selecionar um tipo de informação diferente de Log para este item resultará na perda do carimbo de data/hora local, bem como das informações de severidade e origem 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] #here a custom regular expression named `TWOSHORT` is referenced (defined as
```

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 (*Log Name* na interface gráfica do Event Viewer);
- **regexp** - uma expressão regular que descreve o padrão necessário (sensível a maiúsculas e minúsculas);
- **severity** - uma expressão regular que descreve a severidade (insensível a maiúsculas e minúsculas). Este parâmetro aceita uma expressão regular baseada nos seguintes valores: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (em execução no Windows Vista ou mais recente).
- **source** - uma expressão regular que descreve o identificador da origem (insensível a maiúsculas e minúsculas);
- **eventid** - uma expressão regular que descreve o(s) identificador(es) do evento (sensível a maiúsculas e minúsculas);
- **maxproclines** - o número máximo de novas linhas por segundo que o agent analisará (não pode exceder 10000). O valor padrão é 10*`MaxLinesPerSecond` em `zabbix_agentd.conf`.
- **mode** - valores possíveis:
  - *all* (padrão);
  - *skip* - ignora o processamento de dados mais antigos (afeta apenas items criados recentemente).

Comentários:

- O item deve ser configurado como uma verificação ativa;
- O agent não consegue enviar eventos do log ”Forwarded events”;
- O Windows Eventing 6.0 é suportado;
- Selecionar um tipo de informação diferente de Log para este item resultará na perda do timestamp local, bem como das informações de severidade e origem do log;
- Consulte 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: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **counter** - o caminho para o contador;
- **interval** - os últimos N segundos para armazenar o valor médio. O `interval` deve estar entre 1 e 900 segundos (inclusive) e o valor padrão é 1.

Comentários:

- Este item segue caminhos UNC, o que pode representar um risco de segurança;
- `interval` é usado para contadores que exigem mais de uma amostra (como utilização de CPU), portanto a verificação retorna um valor médio dos últimos "interval" segundos a cada execução;
- O Performance Monitor pode ser usado para obter a lista de contadores disponíveis.
- Veja também: Windows performance counters.

perf_counter_en[counter,<interval>]

 O valor de qualquer contador de desempenho do Windows em inglês.
 Valor de retorno: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **counter** - o caminho para o contador em inglês;
- **interval** - os últimos N segundos para armazenar o valor médio. O `interval` deve estar entre 1 e 900 segundos (inclusive) e o valor padrão é 1.

Comentários:

- Este item segue caminhos UNC, o que pode representar um risco de segurança;
- `interval` é usado para contadores que exigem mais de uma amostra (como utilização de CPU), portanto a verificação retorna um valor médio dos últimos "interval" segundos a cada execução;
- Este item é suportado apenas no **Windows Server 2008/Vista** e versões posteriores;
- Você pode encontrar a lista de strings em inglês visualizando a seguinte chave do 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 objetos 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[process,<attribute>,<type>]

 Várias informações sobre um ou mais processos específicos.
 Valor de retorno: *Float*.

Parâmetros:

- **process** - o nome do processo;
- **attribute** - o atributo do processo solicitado;
- **type** - o tipo de representação (faz sentido quando existe mais de um processo com o mesmo nome)

Comentários:

- Os seguintes `attributes` são suportados:
  - *vmsize* (padrão) - tamanho da memória virtual do processo em Kbytes;
  - *wkset* - tamanho do conjunto de trabalho do processo (quantidade de memória física usada pelo processo) em Kbytes;
  - *pf* - número de page faults;
  - *ktime* - tempo de kernel do processo em milissegundos;
  - *utime* - tempo de usuário do processo em milissegundos;
  - *io_read_b* - número de bytes lidos pelo processo durante operações de I/O;
  - *io_read_op* - número de operações de leitura realizadas pelo processo;
  - *io_write_b* - número de bytes gravados pelo processo durante operações de I/O;
  - *io_write_op* - número de operações de gravação realizadas pelo processo;
  - *io_other_b* - número de bytes transferidos pelo processo durante operações diferentes de leitura e gravação;
  - *io_other_op* - número de operações de I/O realizadas pelo processo, diferentes de leitura e gravação;
  - *gdiobj* - número de objetos GDI usados pelo processo;
  - *userobj* - número de objetos USER usados pelo processo.
- Os `types` válidos são:
  - *avg* (padrão) - valor médio para todos os processos chamados <process>;
  - *min* - valor mínimo entre todos os processos chamados <process>;
  - *max* - valor máximo entre todos os processos chamados <process>;
  - *sum* - soma dos valores de todos os processos chamados <process>.
- Em um sistema de 64 bits, é necessário um Zabbix agent de 64 bits para que este item funcione corretamente.

Exemplos:

```
proc_info[iexplore.exe,wkset,sum] #recupera a quantidade de memória física usada por todos os processos do
proc_info[iexplore.exe,pf,avg] #recupera o número médio de page faults dos processos do Internet Explorer
```

registry.data[key,<value name>]

 Retorna os dados do 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 estar entre aspas duplas.

Exemplos:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #retorna os dados d
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #retorna os dados do
```

registry.get[key,<mode>,<name regexp>]

 A lista de valores ou chaves do Windows Registry localizados na chave informada.
 Valor de retorno: *JSON object*.

Parâmetros:

- **key** - a chave do registry, incluindo a chave raiz; abreviações da chave raiz (por exemplo, HKLM) são permitidas (consulte os comentários de `registry.data[]` para ver a lista completa de abreviações);
- **mode** - valores possíveis:
values (padrão) ou *keys*;
- **name regexp** - descobre apenas valores cujos nomes correspondem à regexp (padrão - descobre todos os valores). Permitido somente com *values* como *mode*.

Chaves com espaços devem estar entre aspas duplas.

Exemplos:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
O JSON incluirá detalhes da chave, da última subchave, do nome do valor, do tipo do valor e dos dados do v
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #retorna os dados de todos o
O JSON incluirá detalhes da chave, da última subchave, do nome do valor, do tipo do valor e dos dados do v
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #retorna todas as subchaves de
O JSON incluirá detalhes da chave e da última subchave.
```

service.discovery

 A lista de serviços do Windows. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

service.info[service,<param>]

 Informações sobre um serviço.
 Valor de retorno: - *Integer* - com `param` como *state*, *startup*;

 Especificamente para *state*: - 0 - em execução - 1 - pausado - 2 - aguardando início - 3 - aguardando pausa - 4 - aguardando continuação - 5 - aguardando parada - 6 - parado - 7 - desconhecido - 255 - serviço inexistente

 Especificamente para *startup*: - 0 - automático - 1 - automático com atraso - 2 - manual - 3 - desativado - 4 - desconhecido - 5 - início automático por trigger - 6 - início automático com atraso por trigger - 7 - início manual por trigger - *String* - com `param` como *displayname*, *path*, *user*; - *Text* - com `param` como *description*.

Parâmetros:

- **service** - um nome de serviço real ou seu nome de exibição, como visto no snap-in Serviços do MMC;
- **param** - *state* (padrão), *displayname*, *path*, *user*, *startup* ou *description*.

Comentários:

- Itens 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] - o tipo de inicialização do serviço EventLog
```

services[<type>,<state>,<exclude>]

 A listagem de serviços.
 Valor de retorno: *0* - se vazio; *Texto* - a lista de serviços separados por 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 atual de memória comprometida para o sistema ou agent Zabbix, o que for menor.

Exemplo:

```
vm.vmemory.size[pavailable] #retorna a memória virtual disponível, em porcentagem
```

wmi.get[<namespace>,<query>]

 Executa uma consulta WMI e retorna o primeiro objeto selecionado.
 Valor de retorno: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **namespace** - o namespace WMI;
- **query** - a consulta WMI retornando um único objeto.

As consultas WMI são realizadas com WQL.

Exemplo:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #retorna o statu
```

wmi.getall[<namespace>,<query>]

 Executa uma consulta WMI e retorna a resposta completa. 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 executadas com WQL.
- O preprocessing com 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%'] #returns status inf
```

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.

Etapa 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 aba *General*, você deve ver um campo chamado *Service name*. O valor a seguir é o nome que você usará ao configurar um item para monitoramento. Por exemplo, se você quisesse monitorar o serviço "workstation", então o nome do seu serviço poderia 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 que seja baseado no mapa de valores *Windows service state* configurado nos templates mencionados.

Observe que ambos os templates mencionados possuem uma regra de descoberta configurada que irá descobrir serviços automaticamente. Se você não quiser isso, você pode desabilitar a regra de descoberta no nível do host assim que o template for vinculado ao host.

Descoberta de serviços do Windows

A descoberta de baixo nível fornece uma maneira de criar automaticamente items, triggers e gráficos para diferentes entidades em um computador. O Zabbix pode iniciar automaticamente o monitoramento dos serviços do Windows em sua máquina, sem a necessidade de saber o nome exato de um serviço ou criar items para cada serviço manualmente. Um filtro pode ser usado para gerar items, triggers e gráficos reais apenas para os serviços de interesse.

Monitoramento de arquivo de log

Visão geral

O Zabbix pode ser usado para monitoramento centralizado e análise de arquivos de log com ou sem suporte à rotação de logs.

As 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:

- Zabbix agent em execução no host.
- item de monitoramento de log configurado.

Attention:
O limite de tamanho de um arquivo de log monitorado depende do suporte a arquivos grandes.

Configuração

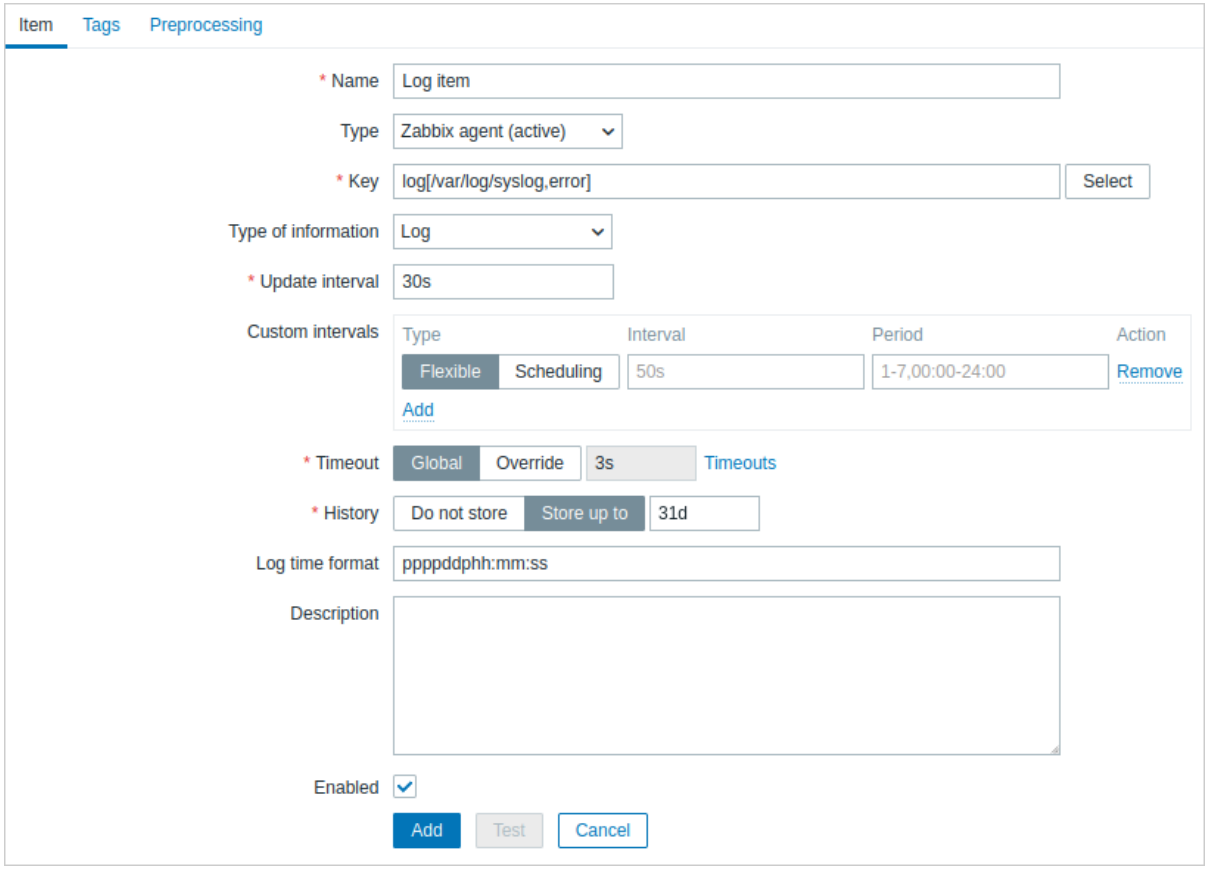
Verificar parâmetros do agent

Certifique-se de que, no arquivo de configuração do agent:

- O parâmetro `Hostname` corresponde ao nome do host no frontend.
- Os servers no parâmetro `ServerActive` estão especificados para o processamento de verificações ativas.

Configuração do item

Configure um item de monitoramento de logs.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Especificamente para itens de monitoramento de logs, informe:

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 'unsupported'. **log.count[]** ou **logrt.count[]**: Essas duas chaves de item permitem retornar somente o número de linhas correspondentes. Consulte a seção de chaves de item do Zabbix agent compatíveis para obter detalhes sobre o uso dessas chaves de item e seus parâmetros.
Tipo de informação	Preenchido automaticamente: Para itens `log[]` ou `logrt[]` - *Log*; Para itens `log.count[]` ou `logrt.count[]` - *Numeric (unsigned)*. Se estiver usando opcionalmente o parâmetro `output`, você poderá selecionar manualmente um tipo de informação apropriado diferente de *Log*. Observe que escolher um tipo de informação diferente de Log fará com que o timestamp local seja perdido.
Intervalo de atualização (em segundos)	O parâmetro define com que frequência o Zabbix agent verificará se houve alterações no arquivo de log. Defini-lo como 1 segundo garantirá que você obtenha novos registros assim que possível.

Formato de hora do log	Neste campo, você pode especificar opcionalmente o padrão para analisar o timestamp da linha de log. Placeholders compatíveis: * **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 no horário Unix, representando `January 1, 1970`. Por exemplo, considere a seguinte linha do arquivo de log do Zabbix agent: `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` Ela começa com seis posições de caracteres para o PID, seguidas pela data, hora e o restante da mensagem. O formato de hora do log para essa linha seria "ppppp: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 rastreamento do tamanho de um log monitorado e da última hora de modificação (para `logrt`) em dois contadores. Além disso:
  - O agent também usa internamente números de inode (em UNIX/GNU/Linux), índices de arquivo (no 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, assume-se que os sistemas de arquivos onde os arquivos de log estão armazenados informam números de inode, que podem ser usados para rastrear arquivos.
  - No Microsoft Windows, o Zabbix agent determina o tipo de sistema de arquivos no qual os arquivos de log residem e usa:
    * Em sistemas de arquivos NTFS, índices de arquivo de 64 bits.
    * Em sistemas de arquivos ReFS (somente a partir do Microsoft Windows Server 2012), IDs de arquivo de 128 bits.
    * Em sistemas de arquivos em que os índices de arquivo mudam (por exemplo, FAT32, exFAT), é usado um algoritmo de fallback para adotar uma abordagem sensata em condições incertas quando a rotação do arquivo de log resulta em vários arquivos de log com a mesma hora da última modificação.
  - Os números de inode, índices de arquivo e somas MD5 são coletados internamente pelo Zabbix agent. Eles não são transmitidos ao Zabbix server e são perdidos quando o Zabbix agent é interrompido.
  - Não modifique a hora da 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 qualquer um dos casos, o Zabbix pode tratar o arquivo como um arquivo diferente e lê-lo novamente desde o início, o que pode gerar alertas duplicados.
  - Se houver vários arquivos de log correspondentes para o item `logrt[]` e o Zabbix agent estiver acompanhando o mais recente deles e esse arquivo de log mais recente for excluído, uma mensagem de aviso `"there are no files matching "<regexp mask>" in "<directory>"` é registrada. O Zabbix agent ignora arquivos de log com hora de modificação menor que a hora de modificação mais recente vista 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 a última hora de modificação (o contador de tempo) são armazenados no banco de dados do Zabbix e são 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 itens que estavam anteriormente desativados ou não suportados. No entanto, se o agent tiver recebido um contador de tamanho diferente de zero do server, mas o item `logrt[]` ou `logrt.count[]` não conseguir encontrar arquivos correspondentes, o contador de tamanho é redefinido para 0 para analisar desde o início caso os arquivos apareçam mais tarde.
- Sempre que o arquivo de log ficar menor que o contador de tamanho do log conhecido pelo agent, o contador é redefinido para zero e o agent começa a ler o arquivo de log desde o início, levando em conta o contador de tempo.
- Se houver vários arquivos correspondentes com a mesma hora da última modificação no diretório, então o agent tenta analisar corretamente todos os arquivos de log com a mesma hora 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 a mesma hora da última modificação, o agent os processará em ordem lexicográfica decrescente. Assim, para alguns esquemas de rotação, os arquivos de log serão analisados e reportados em sua ordem original. Para outros esquemas de rotação, a ordem original dos arquivos de log não será respeitada, o que pode levar ao reporte de registros de arquivos de

log correspondentes em ordem alterada (o problema não ocorre se os arquivos de log tiverem horas de última modificação diferentes).

- O Zabbix agent processa novos registros de um arquivo de log uma vez a cada *Update interval* segundos.
- O Zabbix agent não envia mais do que `maxlines` de um arquivo de log por segundo. O limite evita 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 *Update interval* de 1 segundo, por padrão o agent analisará no máximo 200 registros de arquivo de log e enviará no máximo 20 registros correspondentes ao Zabbix server em uma verificação. Ao aumentar `MaxLinesPerSecond` no arquivo de configuração do agent ou definir o parâmetro `maxlines` na chave do item, o limite pode ser aumentado para até 10000 registros de arquivo de log analisados e 1000 registros correspondentes enviados ao Zabbix server em uma verificação. Se o *Update interval* estiver definido como 2 segundos, os limites para uma verificação seriam definidos 2 vezes mais altos do que com *Update interval* 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 não relacionados a log nele. Portanto, para que os valores de `maxlines` sejam enviados em uma única conexão (e não em várias conexões), o parâmetro `BufferSize` do Zabbix agent deve ser pelo menos `maxlines` x 2. O Zabbix agent pode enviar dados durante a coleta de log e, assim, liberar o buffer, enquanto o Zabbix agent 2 interromperá a coleta de log até que os dados sejam enviados e o buffer seja liberado, o que é feito de forma assíncrona.
- Na ausência de itens de log, todo o tamanho do buffer do agent é usado para valores não relacionados a log. Quando valores de log chegam, eles substituem os valores não relacionados a log mais antigos conforme necessário, até o limite designado de 50%.
- Para registros de arquivo de log maiores que 256 kB, apenas os primeiros 256 kB são comparados com a expressão regular e o restante do registro é ignorado. No entanto, se o Zabbix agent 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.
- Observação 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 se é o primeiro símbolo do nome do arquivo.
- Expressões regulares para `logrt` são suportadas apenas no nome do arquivo; a correspondência por expressão regular no diretório não é suportada.
- Em plataformas UNIX, um item `logrt[]` torna-se NOTSUPPORTED se um diretório onde se espera encontrar os arquivos de log não existir.
- No Microsoft Windows, se um diretório não existir, o item não se tornará NOTSUPPORTED (por exemplo, se o diretório estiver escrito incorretamente 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 Zabbix agent, mas não tornam o item NOTSUPPORTED.
- O arquivo de log do Zabbix agent pode ser útil para descobrir por que um item `log[]` ou `logrt[]` se tornou NOTSUPPORTED. O Zabbix pode monitorar o arquivo de log do seu agent, exceto quando estiver em `DebugLevel=4` ou `DebugLevel=5`.
- A busca por um ponto de interrogação usando uma expressão regular, por exemplo `\?`, pode resultar em falsos positivos se o arquivo de texto contiver símbolos NUL, pois eles são substituídos por `?` pelo Zabbix para continuar processando a linha até o caractere de nova linha.

Extraindo a parte correspondente da expressão regular

Às vezes, podemos querer extrair apenas o valor relevante de um arquivo de destino, em vez de retornar a linha inteira quando uma correspondência de expressão regular é encontrada.

Os itens de log podem extrair valores desejados das linhas correspondentes. Isso é feito por meio do parâmetro adicional **output** nos itens `log` e `logrt`.

O uso do parâmetro 'output' permite indicar o "grupo de captura" da correspondência que pode ser de nosso interesse.

Por exemplo:

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

deve permitir retornar a contagem de entradas 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
```

Somente o número será retornado, pois `\1` faz referência ao primeiro e único grupo de captura: `([0-9]+)`.

Além disso, 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 itens de log permite ignorar algumas linhas mais antigas dos arquivos de log para que as linhas mais

recentes sejam analisadas dentro dos maxdelay segundos.

> **Warning:**
> Especificar maxdelay > 0 pode levar a **ignorar registros importantes do arquivo de log e perder alertas**. Use-o com cuidado, por sua conta e risco, somente quando necessário.

Por padrão, os itens de monitoramento de log seguem todas as novas linhas que aparecem nos arquivos de log. No entanto, há 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 um 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 analisadas diligentemente e as linhas correspondentes serão enviadas ao server conforme configurado nos itens log e logrt.

A proteção integrada contra sobrecarga consiste em um parâmetro configurável maxlines (protege o server contra muitas linhas de log correspondentes recebidas) e um limite de 10*maxlines (protege a CPU e a E/S do host contra sobrecarga pelo agent em uma única verificação). Ainda assim, há 2 problemas com a proteção integrada. Primeiro, um grande número de mensagens potencialmente pouco informativas é reportado ao server e consome espaço no banco de dados. Segundo, devido ao número limitado de linhas analisadas por segundo, o agent pode ficar atrasado em relação aos registros de log mais recentes por horas. Muito provavelmente, você preferirá ser informado mais cedo sobre a situação atual nos arquivos de log em vez de vasculhar 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 seriam necessários para analisar todos os registros restantes em um arquivo de log.

Se o atraso não exceder maxdelay, então o agent prossegue com a análise do arquivo de log normalmente.

Se o atraso for maior que maxdelay, então o agent **ignora um trecho de um arquivo de log "saltando" sobre ele** para uma nova posição estimada, de modo que as linhas restantes possam ser analisadas dentro de maxdelay segundos.

Observe que o agent nem sequer lê as linhas ignoradas para o buffer, mas calcula uma posição aproximada para saltar 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 "jump", o agent ajusta a posição no arquivo para o início de uma linha de log, que pode estar mais adiante no arquivo ou antes.

Dependendo de como a velocidade de crescimento se compara com a velocidade de análise do arquivo de log, você pode não ver "jumps", "jumps" raros ou frequentes, "jumps" grandes ou pequenos, ou até mesmo um "jump" pequeno em cada verificação. Flutuações na carga do sistema e na latência da rede também afetam o cálculo do atraso e, portanto, o "jump" para frente para acompanhar o parâmetro maxdelay.

Não é recomendável definir maxdelay < update interval (isso pode resultar em "jumps" pequenos e frequentes).

Notas sobre o tratamento da rotação de arquivos de log com copytruncate

logrt com a opção copytruncate pressupõe que arquivos de log diferentes tenham registros diferentes (ao menos seus timestamps sejam diferentes), portanto os somatórios MD5 dos blocos iniciais (até os primeiros 512 bytes) serão diferentes. Dois arquivos com os mesmos somatórios 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 cópias de arquivos de log sem relatar duplicatas. No entanto, situações como gerar várias cópias do arquivo de log com o mesmo timestamp, rotação de arquivos de log com mais frequência do que o intervalo de atualização do item logrt[], ou reinicializações frequentes do agent não são recomendadas. O agent tenta lidar com todas essas situações de forma razoável, mas bons resultados não podem ser garantidos em todas as circunstâncias.

Notas sobre arquivos persistentes para itens de log*[]

Finalidade dos arquivos persistentes

Quando o Zabbix agent é iniciado, ele recebe uma lista de verificações ativas do Zabbix server ou proxy. Para métricas log*[], ele recebe o tamanho do log processado e o horário da modificação para determinar onde iniciar o monitoramento do arquivo de log. Dependendo do tamanho real do arquivo de log e do horário da modificação informados pelo sistema de arquivos, o agent decide se deve continuar o monitoramento do arquivo de log a partir do tamanho do log processado ou reanalisar 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 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 do item `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 Zabbix agent.

O principal caso de uso é o monitoramento de 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 Zabbix agent o analisa e envia o tamanho do log processado e o horário da modificação para o server. Na cópia passiva, o arquivo de log permanece inalterado, bem atrás da cópia ativa. Mais tarde, o sistema operacional e o Zabbix agent são reinicializados a partir da cópia passiva. O tamanho do log processado e o horário da modificação que o Zabbix agent 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 separação do espelho do sistema de arquivos, o agent restaura seu estado a partir do arquivo persistente.

Operação do agent com arquivo persistente

Na inicialização, o Zabbix agent não sabe nada sobre arquivos persistentes. Somente após receber uma lista de verificações ativas do Zabbix server (proxy), o agent vê que alguns itens de log devem ser respaldados por arquivos persistentes em diretórios especificados.

Durante a operação do agent, os arquivos persistentes são abertos para gravação (com `fopen(filename, "w")`) e sobrescritos com os dados mais recentes. A chance de perder dados do arquivo persistente se a sobrescrita e a divisão do espelhamento do sistema de arquivos acontecerem ao mesmo tempo é muito pequena; não há tratamento especial para isso. A gravação no arquivo persistente NÃO é seguida por sincronização forçada com o meio de armazenamento (`fsync()` não é chamado).

A sobrescrita com os dados mais recentes é feita após o envio bem-sucedido do registro correspondente do arquivo de log ou dos metadados (tamanho do log processado e tempo de modificação) ao Zabbix server. Isso pode acontecer com a mesma frequência de cada verificação do item, se o arquivo de log continuar mudando.

Nenhuma ação especial é executada durante o encerramento do agent.

Após receber uma lista de verificações ativas, o agent marca arquivos persistentes obsoletos para remoção. Um arquivo persistente torna-se obsoleto se:

1. O item de log correspondente não estiver mais sendo 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 no estado NOTSUPPORTED não são incluídos na lista de verificações ativas, mas eles podem se tornar SUPPORTED mais tarde e seus arquivos persistentes serão úteis.

Se o agent for interrompido antes de 24 horas expirarem, então os arquivos obsoletos não serão excluídos, pois o Zabbix agent não estará mais recebendo informações sobre sua localização do Zabbix server.

> **Warning:**
> Reconfigurar o `persistent_dir` de um item de log de volta para o antigo local de `persistent_dir` enquanto o agent estiver interrompido, sem que o usuário exclua o antigo arquivo persistente, fará com que o estado do agent seja restaurado a partir do antigo arquivo persistente, resultando em mensagens perdidas ou alertas falsos.

Nome e localização dos arquivos persistentes

O Zabbix agent distingue 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]` fará com que o item `logrt[/home/zabbix/test.log,,,10]` seja excluído da lista de verificações ativas do agent e que o item `logrt[/home/zabbix/test.log,,,20]` seja criado (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 do sistema de arquivos, pontos de montagem e opções de montagem, além da configuração de espelhamento do 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 o Zabbix agent não tiver permissões suficientes para criar, gravar, ler ou excluir arquivos, o item de log se tornará NOTSUPPORTED.

Se as permissões de acesso aos arquivos de armazenamento persistente forem removidas durante a operação do agent ou ocorrerem outros erros (por exemplo, disco cheio), os erros serão registrados no arquivo de log do agent, mas o item de log não se tornará NOTSUPPORTED.

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, então os primeiros 50 registros serão enviados em um lote, o arquivo persistente será atualizado, depois os 20 registros restantes serão enviados (talvez com algum atraso, quando mais dados forem acumulados) no 2º lote, e o arquivo persistente será atualizado novamente.

Ações se a comunicação falhar entre agent e server

Cada linha correspondente de um item log[] e logrt[] e o resultado de cada verificação de item log.count[] e logrt.count[] requer uma posição 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 as posições do buffer ficam livres novamente.

Enquanto houver posições livres na área designada de log no buffer de envio do agent e a comunicação falhar entre agent e server (ou proxy), os resultados do monitoramento de log são acumulados no buffer de envio. Isso ajuda a atenuar falhas de comunicação de curta duração.

Durante falhas de comunicação mais longas, todas as posições de log ficam ocupadas e as seguintes ações são executadas:

- As verificações dos itens log[] e logrt[] são interrompidas. Quando a comunicação for restaurada e houver posições livres disponíveis no buffer, as verificações serão retomadas a partir da posição anterior. Nenhuma linha correspondente é perdida; elas apenas serão reportadas mais tarde.
- As verificações log.count[] e logrt.count[] são interrompidas se maxdelay = 0 (padrão). O comportamento é semelhante ao dos itens log[] e logrt[] descritos 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 posições livres no buffer, a verificação é interrompida. Quando a comunicação for restaurada, o agent conta as mesmas 100 linhas correspondentes e também 70 novas linhas correspondentes. Agora o agent envia count = 170 como se elas tivessem sido encontradas em uma única verificação.
- As verificações log.count[] e logrt.count[] com maxdelay > 0: se não houve ”jump” durante a verificação, o comportamento é semelhante ao descrito acima. Se ocorreu um ”jump” sobre linhas do arquivo de log, então a posição após o ”jump” é mantida e o resultado contado é descartado. Assim, o agent tenta acompanhar um arquivo de log em crescimento mesmo em caso de falha de comunicação.

Tratamento de erros de compilação e de 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 entrará no estado NOTSUPPORTED com uma mensagem de erro. Para continuar monitorando o item de log, a expressão regular deverá 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 permanecerá com suporte e o monitoramento continuará. O erro de execução será registrado no arquivo de log do Zabbix agent (sem o registro do arquivo de log).

A taxa de registro é limitada a um erro de execução por verificação, permitindo que o Zabbix agent monitore seu próprio arquivo de log. Por exemplo, se 10 registros forem analisados e 3 falharem com um erro de execução de expressão regular, um registro será produzido no log do agent.

Exceção: se MaxLinesPerSecond=1 e update interval=1 (permitindo que apenas um registro seja analisado por verificação), os erros de execução de expressões regulares não serão registrados.

O zabbix_agentd registra a chave do item em caso de erro de execução; o zabbix_agent2 registra o ID do item para ajudar a identificar qual item de log apresenta erros de execução. Recomenda-se reprojetar a expressão regular em caso de erros de execução.

Verificação 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.

Checks suportados

As chaves de item são listadas sem parâmetros opcionais e informações adicionais. Clique na chave de item para ver os detalhes completos.

Veja também chaves de item de monitoramento VMware.

Item key	Description
icmpping	A acessibilidade do host por ping ICMP.
icmppingloss	A porcentagem de pacotes perdidos.
icmppingretry	A acessibilidade do host por ping ICMP com tentativas de repetição.
icmppingsec	O tempo de resposta do ping ICMP.
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>]

 A acessibilidade do host por ping ICMP.
 Valor de retorno: *0* - o ping ICMP falha; *1* - o ping ICMP é bem-sucedido.

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 estiver vazio (valor padrão), respostas redirecionadas são tratadas como host de destino indisponível; se definido como *allow_redirect*, respostas redirecionadas são tratadas como host de destino disponível.

Consulte também a tabela de valores padrão.

Comentário:

- Se pelo menos um dos pacotes definidos pelo parâmetro `packets` for retornado, o item retorna 1. Se nenhum dos pacotes for retornado, o item retorna 0.

Exemplo:

```
icmpping[,4] #Se pelo menos um dos quatro pacotes for retornado, o item retorna 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 até que o número máximo de tentativas definido no parâmetro `retries` seja atingido. 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 em 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 necessária a versão 2.6.1 ou superior do OpenLDAP.

Processamento de timeout

Timeouts flexíveis de item, embora suportados para verificações simples, não se aplicam a itens `icmpping*` e VMware. Veja Suporte a timeout flexível.

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 com o qual o Zabbix server/proxy é executado, 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

Padrões, limites e descrição dos valores dos parâmetros de verificação ICMP:

Parâmetro	Unidade	Descrição	Opção do fping	Padrões definidos por		Limites permitidos pelo Zabbix	
				fping	**Zabbix**	**mín.**	**máx.**
packets	número	Número de pacotes de solicitação enviados a um destino	-C		3	1	10000
interval	milissegundos	Tempo de espera entre pacotes sucessivos enviados a 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 de espera após o envio do último pacote, afetado pela opção `-C` **fping v4.x** - tempo limite individual para cada pacote	-t	**fping v3.x** - 500 **fping v4.x** e mais recentes - herdado da opção `-p`, mas não superior a 2000		50	ilimitado
retries	número	Número de vezes que será feita uma tentativa de executar ping em um destino, sem incluir a primeira tentativa	-r	3	1	0	ilimitado
backoff factor	número	Número pelo qual o tempo de espera é multiplicado a cada solicitação sucessiva	-B	1.5	1.0	1.0	5.0

Os padrões podem variar ligeiramente dependendo da plataforma e da versão.

Além disso, o Zabbix usa as opções do fping `-i interval ms` (não confunda com o parâmetro `interval` do item mencionado na tabela acima, que corresponde à opção `-p` do fping) e `-S source IP address` (ou `-I` em versões mais antigas do fping). Essas opções são detectadas automaticamente executando verificações com diferentes combinações de opções. O Zabbix tenta detectar o valor mínimo em milissegundos permitido pelo fping para a opção `-i`, testando três valores: 0, 1 e 10. O primeiro valor que funcionar será usado nas verificações ICMP subsequentes. Esse processo é executado individualmente por cada processo ICMP pinger.

As opções do fping detectadas automaticamente são invalidadas a cada hora e detectadas novamente na próxima tentativa de executar uma verificação ICMP. Defina DebugLevel>=4 para visualizar os detalhes desse processo no arquivo de log do server ou do proxy.

O Zabbix grava em um arquivo temporário os endereços IP que devem ser verificados por qualquer uma das chaves `icmpping*` e, em seguida, passa esse arquivo ao fping. Se os itens tiverem parâmetros de chave diferentes, somente aqueles com parâmetros de chave idênticos serão gravados em um único arquivo. Todos os endereços IP gravados no mesmo arquivo serão verificados

pelo fping em paralelo; portanto, o processo ICMP pinger do Zabbix gastará uma quantidade fixa de tempo, independentemente do número de endereços IP no arquivo.

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.

agent SNMP

Visão geral

Talvez você queira usar o monitoramento SNMP em dispositivos como impressoras, switches de rede, roteadores ou UPS, que normalmente são habilitados para SNMP e nos quais seria impraticável tentar configurar sistemas operacionais completos e agents do Zabbix.

Para conseguir recuperar dados fornecidos por agents SNMP nesses dispositivos, o server do Zabbix deve ser configurado inicialmente com suporte a SNMP, especificando a flag `--with-net-snmp`. Também é recomendável 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 executadas somente pelo protocolo UDP.

Os daemons do server e do proxy do 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, elas são úteis para identificar dispositivos SNMP individuais para os quais as requisições combinadas devem ser desativadas.

O server/proxy do Zabbix tentará novamente até 5 vezes para items SNMP `walk` e `get`. O mecanismo de repetição não se aplica a falhas de resolução de DNS.

Para verificações SNMP legadas (OID único numérico ou string), o server/proxy do Zabbix tentará pelo menos mais 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 único para cada dispositivo.

Warning:
O RFC3414 exige que os dispositivos SNMPv3 persistam seus engineBoots. Alguns dispositivos não fazem isso, o que faz com que suas mensagens SNMP sejam descartadas como desatualizadas após uma reinicialização. 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 sondagem a cada vez, reduzindo o tráfego de rede. Se um EngineID não puder ser reutilizado, será realizada uma nova tentativa com sondagem para descobrir o novo EngineID.

Configurando o monitoramento SNMP

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

Etapa 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 você deve ter instalado como parte da instalação do Zabbix) ou uma ferramenta equivalente:

```
snmpwalk -v 2c -c public <host IP> .
```

Como 2c aqui representa a versão do SNMP, você também pode substituí-lo por 1, para indicar a Versão 1 do SNMP no dispositivo.

Isso deve fornecer uma lista de strings SNMP e seu último valor. Se isso não acontecer, é possível que a "community" SNMP seja diferente do padrão `public`; nesse caso, você precisará descobrir qual é.

Em seguida, você pode percorrer a lista até encontrar a string que deseja monitorar, por exemplo: se você quisesse monitorar os bytes recebidos pelo seu switch na porta 3, 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 de `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
```

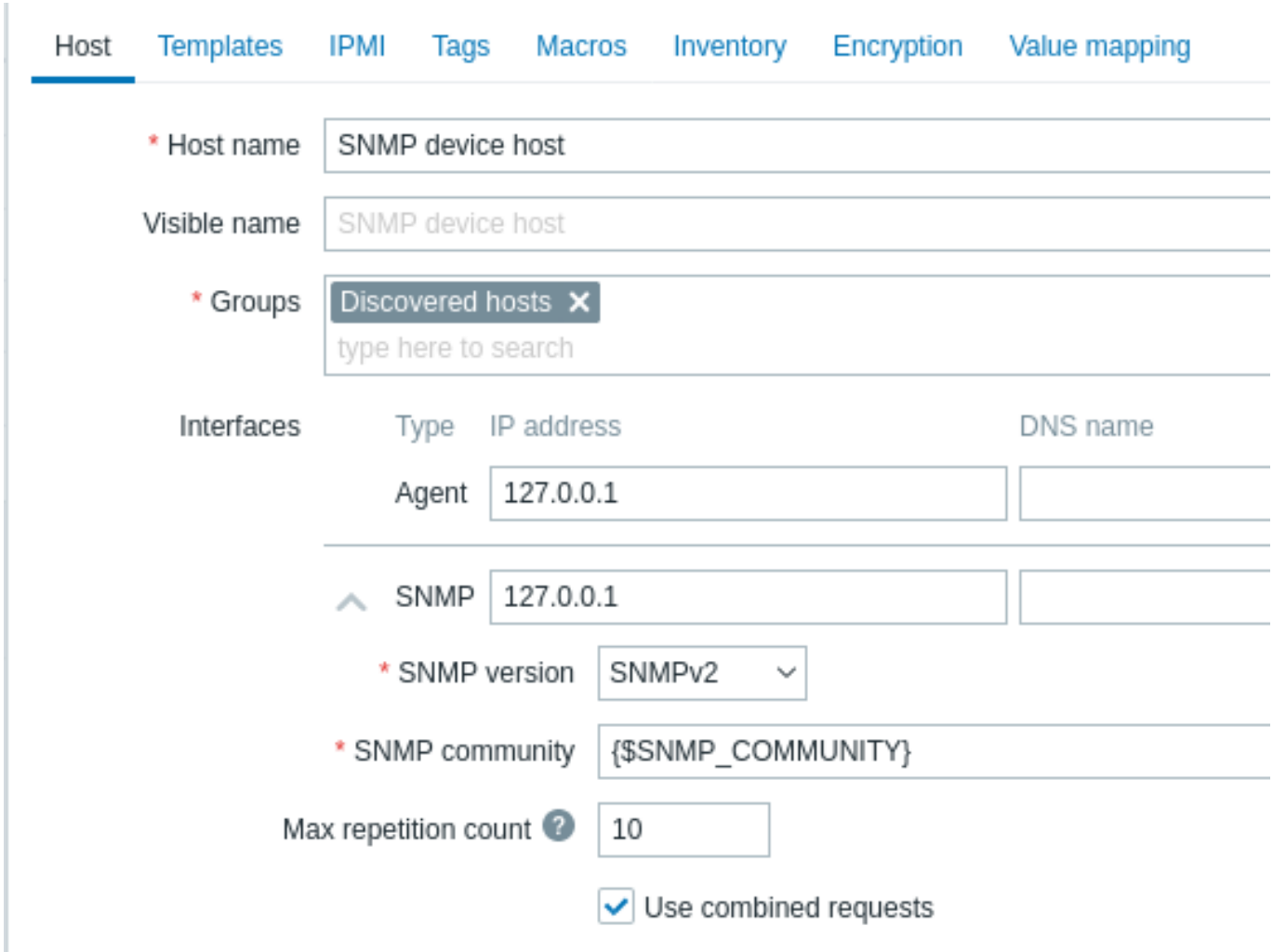
Novamente, o último número no OID é o número da porta.

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.

Etapa 2

Crie um host correspondente a um dispositivo.



Adicione uma interface SNMP para o host:

1. Informe o endereço IP/nome DNS e o número da porta.
2. Selecione a *versão SNMP* no menu suspenso.
3. Adicione as credenciais da interface de acordo com a versão SNMP selecionada:
    - SNMPv1, v2 exigem apenas a community (geralmente 'public').
    - SNMPv3 exige opções mais específicas; consulte SNMPv3.
4. Especifique o valor máximo de repetição (padrão: 10) para solicitações bulk nativas de SNMP (GetBulkRequest-PDUs); apenas para itens `discovery[]` e `walk[]` em SNMPv2 e v3. Observe que definir esse valor muito alto pode causar timeout na verificação do agent SNMP.
5. Marque a caixa de seleção *Usar solicitações combinadas* para permitir o processamento combinado de solicitações SNMP (não relacionado às solicitações bulk nativas "walk" e "get" do SNMP).

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 *Add* para salvar o host.

SNMPv3

Os seguintes parâmetros são obrigatórios para SNMPv3:

- *Context name* - Informe o nome do contexto para identificar o item na sub-rede SNMP.
Macros de usuário são resolvidas neste campo.
- *Security name* - Informe o nome de segurança.
Macros de usuário são resolvidas neste campo.
- *Security level* - Selecione o nível de segurança:
    - *noAuthNoPriv* - nenhum protocolo de autenticação nem de privacidade é usado;
    - *AuthNoPriv* - o protocolo de autenticação é usado, o protocolo de privacidade não é;
    - *AuthPriv* - os protocolos de autenticação e de privacidade são usados.
- *Authentication protocol* - Selecione o protocolo de autenticação: *MD5*, *SHA1*; com net-snmp 5.8 e versões mais recentes, *SHA224*, *SHA256*, *SHA384* ou *SHA512*.
- *Authentication passphrase* - Informe a frase secreta de autenticação.
Macros de usuário são resolvidas neste campo.
- *Privacy protocol* - Selecione o protocolo de privacidade: *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) ou *AES256C* (Cisco).
Consulte as observações sobre suporte ao protocolo de privacidade.
- *Privacy passphrase* - Informe a frase secreta de privacidade.
Macros de usuário são resolvidas neste campo.

No caso de credenciais SNMPv3 incorretas (security name, authentication protocol/passphrase, privacy protocol):

- O Zabbix recebe um ERROR do net-snmp, exceto no caso de *Privacy passphrase* incorreta, em que o Zabbix recebe um erro TIMEOUT do net-snmp.
- A disponibilidade da interface SNMP mudará para vermelho (indisponível).

Note:
Alterações em *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* ou *Privacy passphrase*, feitas sem alterar o *Security name*, normalmente são aplicadas automaticamente quando a interface SNMPv3 correspondente é atualizada no Zabbix. Nos casos em que *Security name* também é alterado, todos os parâmetros serão atualizados imediatamente.

Suporte a 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 novos (por exemplo, RHEL9), o suporte a DES é removido do pacote `net-snmp`.
- Os protocolos de criptografia AES192 e mais fortes não são suportados nativamente 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 oferece suporte a AES192+, use uma das opções a seguir:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Se a saída contiver `--enable-blumenthal-aes`, o AES192+ é suportado.

Observe que `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`, o AES192+ não é suportado. Se a saída contiver `No hostname specified.`, o AES192+ não é suportado.

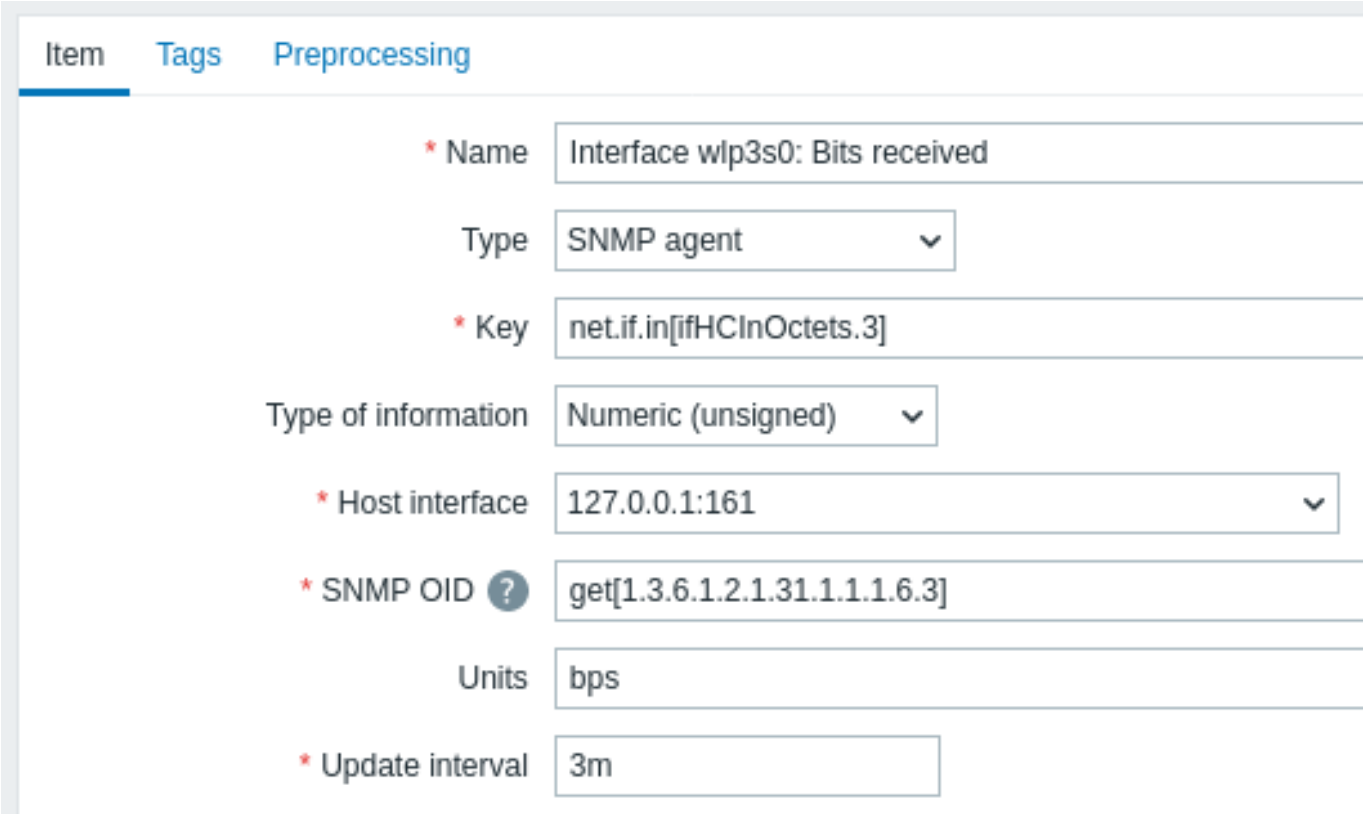
Se a sua biblioteca net-snmp não oferecer suporte aos protocolos AES192 e superiores, recompile o net-snmp com a opção `--enable-blumenthal-aes` e, em seguida, recompile o Zabbix server especificando a opção `--with-net-snmp=/home/user/yourcust`

Etapa 3

Crie um item para monitoramento.

Agora, volte ao Zabbix e clique em *Items* para o host SNMP criado anteriormente. Dependendo de você ter usado ou não um template ao criar o host, haverá uma lista de itens SNMP associados ao host ou apenas uma lista vazia. Vamos considerar que você criará o item manualmente usando as informações obtidas com snmpwalk e snmpget. Portanto, clique em *Create item*.

Preencha os parâmetros obrigatórios no formulário do novo item:



Parameter	Description
Name	Insira o nome do item.
Type	Selecione **SNMP agent** aqui.
Key	Insira uma chave significativa.
Host interface	Certifique-se de selecionar a interface SNMP, por exemplo, do seu switch/roteador.

Parameter	Description
SNMP OID	Use um dos formatos compatíveis para inserir o(s) valor(es) de 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]`. Essa opção usa solicitações bulk nativas do SNMP (GetBulkRequest-PDUs) de forma assíncrona. As configurações de tempo limite para este item podem ser definidas no formulário de configuração do item. Considere definir um valor baixo de tempo limite para evitar longos atrasos se o dispositivo estiver inacessível, pois serão feitas até 5 tentativas caso as anteriores atinjam o tempo limite ou falhem (por exemplo, um tempo limite de 3 segundos pode resultar em uma espera de 15 segundos). Você pode usar este item como item mestre, com itens dependentes que extraem dados do item mestre usando o pré-processamento. É possível especificar vários OIDs em um único snmp walk, como `walk[OID1,OID2,...]`, para processar os OIDs de forma assíncrona, um por vez. Se a solicitação bulk não retornar resultados, será feita uma tentativa de recuperar um único registro sem uma solicitação bulk. Nomes de MIB são aceitos como parâmetros; portanto, `walk[1.3.6.1.2.1.2.2.1.2]` e `walk[ifDescr]` retornarão a mesma saída. Se vários OIDs/MIBs forem especificados, por exemplo, `walk[ifDescr,ifType,ifPhysAddress]`, a saída será uma lista concatenada. Solicitações GetBulk são usadas com interfaces SNMPv2 e v3, e GetNext com interfaces SNMPv1; o número máximo de repetições para solicitações bulk é configurado no nível da interface. O parâmetro de número máximo de repetições afeta as solicitações bulk ao determinar o número máximo de OIDs retornados em uma única resposta bulk. Um valor maior resulta em respostas bulk maiores, reduzindo o número de transmissões necessárias. No entanto, alguns dispositivos podem não aceitar 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 na 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 tempo limite para este item podem ser definidas no formulário de configuração do item. Considere definir um valor baixo de tempo limite para evitar longos atrasos se o dispositivo estiver inacessível, pois serão feitas até 5 tentativas caso as anteriores atinjam o tempo limite ou falhem (por exemplo, um tempo limite de 3 segundos pode resultar em uma 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 tempo limite da verificação do item será igual ao valor definido no arquivo de configuração do server. É **recomendável** usar itens `walk[OID]` e `get[OID]` para obter melhor desempenho. Todos os itens `walk[OID]` e `get[OID]` são executados 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 simultaneidade máxima de verificações assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller). O número de pollers SNMP assíncronos é definido pelo parâmetro StartSNMPPollers. Observe que, para estatísticas de tráfego de rede retornadas por qualquer um dos métodos, uma etapa *Change per second* deve ser adicionada na aba *Preprocessing*; caso contrário, você obterá o valor cumulativo do dispositivo SNMP em vez da alteração mais recente.

Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Agora, salve o item e acesse *Monitoring* > *Latest data* para visualizar os 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

Solicitações SNMP bulk nativas

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

Uma solicitação GetBulk no SNMP executa várias solicitações GetNext e retorna o resultado em uma única resposta. Isso pode ser usado tanto para itens SNMP regulares quanto para descoberta SNMP, a fim de minimizar as viagens de ida e volta pela rede.

O item SNMP **walk[OID1,OID2,...]** pode ser usado como o item mestre que coleta dados em uma única solicitação, com itens dependentes que analisam a resposta conforme necessário usando o pré-processamento.

Observe que o uso de solicitações SNMP bulk nativas não está relacionado à opção de combinar solicitações SNMP, que é a forma própria do Zabbix de combinar várias solicitações SNMP (consulte a próxima seção).

Serão realizadas até cinco tentativas para itens SNMP bulk, a fim de evitar falhas caso um dos pacotes seja perdido. O tempo limite para itens SNMP com `get` e `walk` (definido no formulário de configuração do item) é aplicado a toda a sessão. O tempo limite é aplicado independentemente de os dados serem recuperados completamente; se os dados forem recebidos parcialmente (por exemplo, se os dados forem coletados com sucesso para apenas um de vários OIDs), o item ficará sem suporte com a mensagem ”Only partial data received”. Se o tempo limite for atingido, uma nova tentativa será realizada, o tempo limite será redefinido e a última solicitação será reenviada, permitindo que a sessão continue a partir da última solicitação caso um único pacote seja perdido ou chegue tarde demais. Considere definir um valor baixo para o tempo limite, a fim de evitar longos atrasos quando o dispositivo estiver inacessível, pois serão realizadas até 5 tentativas se as anteriores expirarem ou falharem (por exemplo, um tempo limite de 3 segundos pode resultar em uma espera de 15 segundos).

Funcionamento interno do processamento combinado

O server e o proxy do Zabbix podem consultar dispositivos SNMP em busca de vários valores em uma única solicitação. Isso afeta vários tipos de itens SNMP:

- itens SNMP regulares
- itens SNMP com índices dinâmicos
- regras de descoberta de baixo nível SNMP

Todos os itens SNMP em uma única interface com parâmetros idênticos são programados para serem consultados ao mesmo tempo. Os dois primeiros tipos de itens são obtidos pelos pollers em lotes de, no máximo, 128 itens, enquanto as regras de descoberta de baixo nível são processadas individualmente, como antes.

Em um nível inferior, há dois tipos de operações realizadas para consultar valores: obter vários objetos especificados e percorrer uma árvore de OIDs.

Para ”obter”, um GetRequest-PDU é usado com, no máximo, 128 associações de variáveis. Para ”percorrer”, um GetNextRequest-PDU é usado para SNMPv1, e um 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:

- os itens SNMP regulares se beneficiam das melhorias de ”obter”;
- os itens SNMP com índices dinâmicos se beneficiam das melhorias de ”obter” e ”percorrer”: ”obter” é usado para a verificação do índice e ”percorrer” para criar o cache;
- as regras de descoberta de baixo nível SNMP se beneficiam das melhorias de ”percorrer”.

No entanto, há um problema técnico: nem todos os dispositivos são capazes de retornar 128 valores por solicitação. Alguns sempre retornam uma resposta adequada, mas outros respondem com um erro ”tooBig(1)” ou não respondem quando a resposta potencial ultrapassa determinado limite.

Para encontrar um número ideal de objetos a serem consultados para um determinado dispositivo, o Zabbix usa a estratégia a seguir. Ele começa de forma cautelosa, consultando 1 valor por solicitação. Se isso for bem-sucedido, consulta 2 valores por solicitação. Se isso também for bem-sucedido, consulta 3 valores por solicitação e continua de forma semelhante, multiplicando o número de objetos consultados por 1,5, resultando na seguinte sequência de tamanhos de solicitação: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

No entanto, quando um dispositivo se recusa a fornecer uma resposta adequada (por exemplo, para 42 variáveis), o Zabbix faz duas coisas.

Primeiro, para o lote de itens atual, ele reduz pela metade o número de objetos em uma única solicitação e consulta 21 variáveis. Se o dispositivo estiver ativo, a consulta deverá funcionar na grande maioria dos casos, pois já se sabe que 28 variáveis funcionam e 21 é significativamente menos que isso. No entanto, se isso ainda falhar, o Zabbix volta a consultar os valores um por um. Se ainda falhar nesse ponto, o dispositivo definitivamente não está respondendo, e o tamanho da solicitação não é o problema.

A segunda coisa que o Zabbix faz para os lotes de itens subsequentes é começar com o último número de variáveis bem-sucedido (28 em nosso exemplo) e continuar incrementando os tamanhos das solicitações em 1 até atingir o limite. Por exemplo, supondo que o maior tamanho de resposta seja de 32 variáveis, as solicitações subsequentes terão os tamanhos 29, 30, 31, 32 e 33. A última solicitação falhará, e o Zabbix nunca mais emitirá uma solicitação de tamanho 33. A partir desse ponto, o Zabbix consultará no máximo 32 variáveis para esse dispositivo.

Se consultas grandes falharem com esse número de variáveis, isso pode significar uma de duas coisas. Não é possível conhecer os critérios exatos usados por um dispositivo para limitar o tamanho da resposta, mas tentamos fazer uma aproximação usando o número de variáveis. A primeira possibilidade é que esse número de variáveis esteja próximo do limite real de tamanho da resposta do dispositivo no caso geral: às vezes, a resposta é menor que o limite e, às vezes, é maior. A segunda possibilidade é que um pacote UDP em qualquer uma das direções simplesmente tenha sido perdido. Por esses motivos, quando o Zabbix recebe uma consulta com falha, ele reduz o número máximo de variáveis a serem tentadas para se aproximar mais da faixa confortável do dispositivo, mas somente até duas vezes.

No exemplo acima, se uma consulta com 32 variáveis falhar, o Zabbix reduzirá a quantidade para 31. Se isso também falhar, o Zabbix reduzirá a quantidade para 30. No entanto, o Zabbix não reduzirá a quantidade para menos de 30, pois presumirá que falhas adicionais são causadas pela perda de pacotes UDP, e não pelo limite do dispositivo.

Se, no entanto, um dispositivo não conseguir processar corretamente solicitações combinadas por outros motivos e a heurística descrita acima não funcionar, haverá uma configuração *Use combined requests* para cada interface, que permite desativar as solicitações combinadas para esse dispositivo.

Se as solicitações combinadas causarem respostas parciais ou malformadas que resultem em cálculos incorretos por segundo (delta) (por exemplo, picos aparentes nos contadores de interface), desative *Use combined requests* na interface afetada para forçar consultas separadas por item; isso geralmente evita picos falsos. Como alternativa, considere usar itens assíncronos `get[]` ou `walk[]`, que são executados de forma assíncrona e não estão sujeitos ao agrupamento por interface de *Use combined requests* — eles podem ser usados no lugar das verificações de OID síncronas legadas para evitar problemas relacionados a solicitações combinadas. Procure entradas no log do server/proxy semelhantes à mostrada na seção Visão geral para ajudar a identificar os dispositivos afetados.

Além disso, se a interface ficar indisponível com frequência, pode ser necessário aumentar o parâmetro `UnavailableDelay` nos arquivos de configuração do server do Zabbix ou do proxy do Zabbix para reduzir a frequência das solicitações. Os itens podem se tornar sem suporte se dados parciais forem recebidos durante a descoberta ou ao percorrer OIDs.

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 não é possível confiar totalmente que o número de índice permanecerá sempre o mesmo.

Os números de índice podem ser dinâmicos — eles podem mudar ao longo do tempo e, como consequência, seu item pode parar de funcionar.

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 a **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 recuperação de dados no item.
index	Método de processamento. Atualmente, um método é suportado: **index** - pesquisar o índice e anexá-lo 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 busca. Diferencia maiúsculas de minúsculas.

Exemplo

Obtendo o uso de memória do processo *apache*.

Se estiver usando esta sintaxe de 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 recebermos o valor no qual estamos interessados:

```
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 do índice, mesmo que uma correspondência pudesse ser encontrada antes.
Isso é feito caso outro item venha a se referir ao mesmo OID base posteriormente — o Zabbix pesquisará 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 tiver mudado, o valor é solicitado.
Se tiver mudado, o cache é reconstruído — cada poller que encontrar um índice alterado percorrerá novamente a tabela SNMP de índice.

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.

Special OID	Identifier	Description
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. Essa string deve incluir o nome do fabricante, o nome do produto e a versão da interface de hardware.

Special OID	Identifier	Description
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.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	O número de pacotes unicast da 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 ou multicast da sub-rede) entregues a um protocolo de camada superior.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	O número de pacotes de entrada que foram escolhidos para serem descartados, embora nenhum erro tenha sido detectado, para impedir que fossem entregues a um protocolo de camada superior. Um possível motivo para descartar esse pacote poderia ser liberar espaço em buffer.
ifInErrors	1.3.6.1.2.1.2.2.1.14	O número de pacotes de entrada que continham erros, impedindo que fossem entregues 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 para fora da 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 os 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 os que foram descartados ou não enviados.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	O número de pacotes de saída que foram escolhidos para serem descartados, embora nenhum erro tenha sido detectado, para impedir que fossem transmitidos. Um possível motivo para descartar esse pacote poderia ser liberar espaço em buffer.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	O número de pacotes de saída 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).

Arquivos MIB

Introdução

MIB significa Management Information Base (Base de Informações de Gerenciamento). Os arquivos MIB permitem o uso de representações textuais de OIDs (Object Identifiers, identificadores de objetos). É possível usar OIDs brutos ao monitorar dispositivos SNMP com o Zabbix, mas, se você se sentir mais confortável usando uma representação textual, será necessário instalar os 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.

SNMP trap

Visão geral

Receber traps SNMP é o oposto de consultar dispositivos habilitados para SNMP.

Nesse caso, as informações são enviadas de um dispositivo habilitado para SNMP para `snmptrapd` e são coletadas ou "capturadas" pelo Zabbix server ou Zabbix proxy a partir de um arquivo.

Normalmente, os traps são enviados quando ocorre alguma mudança de condição, e o agent se conecta 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 breves que ocorrem entre os intervalos de consulta e que podem passar despercebidos pelos dados consultados.

O recebimento de traps SNMP no Zabbix foi projetado para funcionar com `snmptrapd` e um dos mecanismos de passagem dos 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 com script Bash, porque Perl e SNMPTT muitas vezes estão ausentes em distribuições modernas e exigem uma configuração mais complexa. No entanto, essa solução usa um script configurado como `traphandle`. Para melhor desempenho em sistemas de produção, use a solução com Perl embutido (seja um script com a opção `do perl` ou SNMPTT).

O fluxo de trabalho para receber um trap:

1. `snmptrapd` recebe um trap.
2. `snmptrapd` passa o trap para o script receptor (Bash, Perl) ou SNMPTT.
3. O receptor analisa, formata e grava o trap em um arquivo.
4. O Zabbix SNMP trapper lê e analisa o arquivo de traps.
5. Para cada trap, o Zabbix encontra todos os itens SNMP trapper com interfaces de host que correspondem ao endereço do trap recebido. Observe que, durante a correspondência, é usado apenas o *IP* ou *DNS* selecionado na interface do host.
6. Para cada item encontrado, o trap é comparado com a regexp em `snmptrap[regexp]`. O trap é definido como o valor de **todos** os itens correspondentes. Se nenhum item correspondente for encontrado e houver um item `snmptrap.fallback`, o trap será definido como o valor desse item.
7. Se o trap não for definido como valor de nenhum item, o Zabbix, por padrão, registra o trap não correspondente. (Isso é configurado por *Log unmatched SNMP traps* em *Administration* > *General* > *Other*.)

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:

Etapa 1: Criar uma interface SNMP para o host

1. Em *Coleta de dados* > *Hosts*, crie um host ou edite o existente.
2. No campo *Interfaces*, selecione o tipo de interface *SNMP*.
3. Informe o endereço IP/nome DNS e o número da porta.
4. Selecione a *versão SNMP* no menu suspenso: (*SNMPv1*, *SNMPv2*, ou *SNMPv3*) e adicione as credenciais da interface de acordo com a versão SNMP selecionada.

O endereço de cada trap recebido será comparado com os endereços IP e DNS de todas as interfaces SNMP para encontrar os hosts correspondentes.

Etapa 2: Configurar o item

1. Para o host, crie um item ou edite o existente.
2. No campo *Key*, use uma das chaves de trap SNMP:
   - `snmptrap[regexp]` - captura todos os traps SNMP que correspondem à expressão regular especificada no parâmetro `regexp`; se `regexp` não for especificado, captura qualquer trap.
 Macros de usuário e expressões regulares globais são suportadas no parâmetro.
 Valor de retorno: trap SNMP.
 Este item pode ser definido somente para interfaces SNMP.
   - `snmptrap.fallback` - captura todos os traps SNMP que não foram capturados por nenhum dos itens `snmptrap[]` para essa interface.
 Valor de retorno: trap SNMP.
 Este item pode ser definido somente para interfaces SNMP.

> **Note:**
> A correspondência de expressões regulares em várias linhas não é suportada no momento.

3. Defina *Tipo de informação* como *Log* para que os timestamps sejam analisados. Outros formatos, como *Numérico*, também são aceitáveis, mas podem exigir um manipulador de trap personalizado.

Configurando o monitoramento de traps SNMP

Configurando o Zabbix server/proxy

Para ler os traps, o Zabbix server ou proxy deve ser configurado para iniciar o processo SNMP trapper e apontar para o arquivo de traps que está sendo gravado pelo SNMPTT ou por um receptor de traps 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 do systemd `PrivateTmp` for usado, é improvável que esse arquivo funcione em `/tmp`.

Configurando o receptor de traps Bash

Requisitos: apenas snmptrapd.

Um script de receptor de traps Bash pode ser usado para encaminhar traps para o Zabbix server a partir do snmptrapd usando o arquivo trapper. Para configurá-lo, adicione a opção `traphandle` ao arquivo de configuração do snmptrapd (`snmptrapd.conf`), veja o exemplo.

> **Note:**
> Pode ser necessário reiniciar o snmptrapd para que as alterações em sua configuração sejam aplicadas.

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 encaminhar traps para o server Zabbix diretamente do snmptrapd. Para configurá-lo:

- Adicione o script Perl ao arquivo de configuração do snmptrapd (`snmptrapd.conf`), por exemplo:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Configure o receptor, por exemplo:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

> **Note:**
> Talvez seja necessário reiniciar o snmptrapd para que as alterações em sua configuração sejam aplicadas.

> **Note:**
> Se o nome do script não estiver entre aspas, o snmptrapd recusará 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 obter o melhor desempenho, o SNMPTT deve ser configurado como um daemon usando `snmptthandler-embedded` para encaminhar os traps para ele. Veja as instruções para configurar o SNMPTT.

Quando o SNMPTT estiver configurado para receber os traps, configure `snmptt.ini`:

1. Habilite o uso do módulo Perl do pacote `NET-SNMP`:

```
net_snmp_perl_enable = 1
```

2. Registre os traps no arquivo de traps que será lido pelo Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. Defina o formato de data e hora:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

> **Warning:**
> O pacote `net-snmp-perl` foi removido no RHEL 8.0-8.2; foi adicionado novamente no RHEL 8.3. Para mais informações, consulte os known issues.

Agora formate os traps para que o Zabbix os reconheça (edite `snmptt.conf`):

1. Cada instrução `FORMAT` deve começar com `ZBXTRAP [address]`, em que `[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 de trap SNMP abaixo.

> **Attention:**
> Não use traps desconhecidos - o Zabbix não conseguirá reconhecê-los. Traps desconhecidos podem ser tratados definindo um evento geral em `snmptt.conf`:

> `EVENT general .* "General event" Normal`

Formato de trap SNMP

Todos os receivers de trap Perl personalizados e a configuração de trap do SNMPTT devem formatar o trap da seguinte maneira:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 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.
- `[address]` - endereço IP usado para localizar o host para este trap.

Observe que `ZBXTRAP` e `[address]` serão removidos da mensagem durante o processamento. Se o trap for formatado de outra maneira, 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 oferece suporte a arquivos grandes para arquivos do 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 somente depois excluí-lo, para que nenhum trap seja perdido:

1. O Zabbix abre o arquivo de trap no último local conhecido e vai para a etapa 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 nenhum arquivo aberto, o Zabbix redefine o último local e vai para a etapa 1.
3. O Zabbix lê os dados do arquivo atualmente aberto e define o novo local.
4. Os novos dados são analisados. Se este era o arquivo rotacionado, o arquivo é fechado e volta para a etapa 2.
5. Se não houver novos dados, o Zabbix dorme por 1 segundo e volta para a etapa 2.

Sistema de arquivos

Devido à implementação do arquivo de trap, o Zabbix precisa que o sistema de arquivos ofereça suporte a inodes para diferenciar arquivos (as informações são obtidas por uma chamada `stat()`).

Exemplos de configuração usando diferentes versões do protocolo SNMP

Este exemplo usa o snmptrapd e um script receptor Bash para encaminhar traps ao Zabbix server.

Configuração:

1. Configure o Zabbix para iniciar o SNMP trapper e defina o arquivo de traps. 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/temp
```

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 a `snmtrapd.conf` (consulte o exemplo funcional):

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Pode ser necessário reiniciar o snmptrapd para que ele carregue as alterações na configuração.

4. Crie um item SNMP TEST (tenha em mente os 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, configuraremos o `snmptrapd` para a versão escolhida do protocolo SNMP e enviaremos 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. O mesmo valor deve ser definido nos remetentes de traps SNMP.

Observe que, embora ainda seja amplamente usado em ambientes de produção, o SNMPv2 não oferece criptografia nem autenticação real do remetente. Os dados são enviados em texto simples e, portanto, essas versões do protocolo devem ser usadas somente 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 usada atualmente, pois não oferece suporte a contadores de 64 bits e é considerada um protocolo legado.

Para permitir o recebimento de traps SNMPv1 ou SNMPv2, adicione a seguinte linha a `snmptrapd.conf`. Substitua `secret` pela community string SNMP configurada nos remetentes de traps SNMP:

```
authCommunity log,execute,net secret
```

Em seguida, podemos enviar um trap de teste usando `snmptrap`. Neste exemplo, usaremos o OID comum "link up":

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

O SNMPv3 resolve problemas de segurança do SNMPv1/v2 e fornece autenticação e criptografia. Você pode usar os métodos de autenticação MD5 ou várias opções SHA, além de DES ou várias opções AES como cifra.

Configure o `snmptrapd` para aceitar traps SNMPv3 adicionando as seguintes linhas a `snmptrapd.conf`:

```
##### Cria um usuário USM SNMPv3 com autenticação e privacidade
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### Permite que o usuário registre traps e execute scripts de tratamento de traps
authUser log,execute <securityName>
```

Por exemplo:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Observe a palavra-chave "execute", que permite executar scripts para este modelo de segurança do usuário.

Para testar a configuração de traps SNMPv3, envie um trap de teste usando o seguinte comando:

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
Se quiser usar métodos de criptografia fortes, como AES192 ou AES256, use o net-snmp a partir da versão 5.8. Talvez seja necessário recompilá-lo com a opção `configure: --enable-blumenthal-aes`. Versões mais antigas do net-snmp não oferecem suporte a AES192/AES256. Consulte também: Strong Authentication or Encryption.

Verificação

Em ambos os exemplos, você verá linhas semelhantes no arquivo `/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)

Zabbix interno

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 pelo proxy, independentemente do status de manutenção do host.

Para usar este item, selecione 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 compatíveis

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

Chave de item	Descrição
zabbix[apm,]	As estatísticas dos dados de telemetria recebidos pelo coletor de APM.
zabbix[boottime]	O horário 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 itens habilitados (compatíveis e não compatíveis) no host.
zabbix[host„items_unsupported]	O número de itens não compatíveis habilitados no host.
zabbix[host„maintenance]	O status atual de manutenção do host.
zabbix[host,active_agent,available]	A disponibilidade das verificações do agent ativo no host.
zabbix[host,discovery,interfaces]	Os detalhes de todas as interfaces configuradas do host no Zabbix frontend.
zabbix[host„available]	A disponibilidade da interface principal de um determinado tipo de verificação no host.
zabbix[hosts]	O número de hosts monitorados.
zabbix[items]	O número de itens habilitados (compatíveis e não compatíveis).
zabbix[items_unsupported]	O número de itens não compatíveis.
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 determinado processo do Zabbix ou um grupo de processos (identificado por <type> e <mode>) passou em <state>.
zabbix[proxy„]	As informações sobre o Zabbix proxy.
zabbix[proxy,discovery]	A lista de Zabbix proxies.
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.
zabbix[proxy group„proxies]	A lista de Zabbix proxies em um grupo de proxies.
zabbix[proxy group„state]	O estado de um grupo de proxies.

Chave de item	Descrição
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 alterações de estado entre os modos de buffer em disco e em memória desde a inicialização.
zabbix[proxy_buffer,state,current]	Retorna o estado de operação atual no qual 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 itens monitorados na fila que estão atrasados em 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 esperados por segundo.
zabbix[stats,,]	Retorna as métricas internas de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas serão obtidas da instância remota; caso contrário, da instância local.
zabbix[stats,,,queue,,]	Retorna as métricas internas da fila de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas serão obtidas da instância remota; caso contrário, da instância local.
zabbix[tcache,,]	As estatísticas de eficácia do cache da função de tendências do Zabbix.
zabbix[triggers]	O número de triggers habilitados no banco de dados do Zabbix, com todos os itens 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 eficácia 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 a disponibilidade do cache de gravação do Zabbix.

Detalhes da chave do item

- Parâmetros sem colchetes angulares são obrigatórios e devem ser usados *como estão* (por exemplo, ”host” e ”available” em `zabbix[host,<type>,available]`).
- Parâmetros com colchetes angulares **< >** devem ser substituídos por um valor válido. Se um parâmetro tiver um valor padrão, ele pode ser omitido.
- Valores para itens e parâmetros de item rotulados como ”não suportado em proxy” só podem ser recuperados se o host for monitorado pelo server. Da mesma forma, valores ”não suportados em server” só podem ser recuperados se o host for monitorado por proxy.

zabbix[apm,<type>]

 As estatísticas dos dados de telemetria recebidos pelo coletor APM:

- *requests* - o número de solicitações recebidas pelo coletor APM (contador):
  - *accepted* - o número de solicitações aceitas para processamento;
  - *dropped* - o número de solicitações descartadas;
- *rows* - o número de linhas de dados de telemetria gravadas na fonte de dados, por tipo de dado (contador).

Valor de retorno: *objeto JSON*.

Parâmetros:

- **type** - *traces*, *metrics* ou *logs*. Se não for especificado, serão retornadas as estatísticas de todos os tipos.

Comentários:

- Este item não é compatível com o Zabbix server.

Exemplo de valores de retorno para `zabbix[apm]`:

```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
```

```
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```

zabbix[boottime]

 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 retornado: *objeto JSON*.

Comentários:

- Este item pode ser usado na descoberta de baixo nível.
- Os campos retornados correspondem principalmente às propriedades do objeto de nó de alta disponibilidade.
- Campos adicionais:
  - `db_timestamp` - hora atual do server (timestamp Unix) incluída na resposta
  - `lastaccess_age` - segundos desde o último heartbeat do nó (db_timestamp - lastaccess)
  - `status` - estado do nó:
    * 0 - em espera
    * 1 - interrompido manualmente
    * 2 - indisponível
    * 3 - ativo

Exemplo de valores retornados:

```
[
  {
    "id": "ckvupihk70001z8mkpw5cg0u3",
    "name": "zabbix-prod-01",
    "status": 3,
    "address": "10.0.4.12:10051",
    "port": 10051,
    "lastaccess": 1756115995,
    "db_timestamp": 1756116000,
    "lastaccess_age": 5
  },
  {
    "id": "ckvx2a9k70004b1nq2hz9d7f",
    "name": "zabbix-standby-02",
    "status": 0,
    "address": "10.0.4.13:10051",
    "port": 10051,
    "lastaccess": 1756115550,
    "db_timestamp": 1756116000,
    "lastaccess_age": 450
  },
  {
    "id": "ckw0bq3l70007y4r1a0m5kz8",
    "name": "zabbix-backup-eu1",
    "status": 1,
    "address": "backup.example.com:10051",
    "port": 10051,
    "lastaccess": 1756105080,
    "db_timestamp": 1756116000,
    "lastaccess_age": 10920
  }
]
```

zabbix[connector_queue]

 A contagem de valores enfileirados na fila do conector.
 Valor de retorno: *Inteiro*.

zabbix[discovery_queue]

 A contagem de verificações de rede enfileiradas na fila de descoberta.
 Valor de retorno: *Inteiro*.

zabbix[host,,items]

 O número de items habilitados (suportados e não suportados) no host.
 Valor de retorno: *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 está 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 tipo específico de verificações no host.
 Valores retornados: *0* - indisponível; *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 inalcançabilidade/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 processo específico do Zabbix ou um grupo de processos (identificado por `<type>` e `<mode>`) passou em `<state>`. Ela é calculada somente para o último minuto.
 Valor retornado: *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*, *apm manager*, *apm worker*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode**:
  - *avg* - valor médio de 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-fork; por exemplo, se houver 4 trappers em execução, o valor estará entre 1 e 4);
- **state**:
  - *busy* - o processo está ocupado, por exemplo, processando uma solicitação (padrão);
  - *idle* - o processo está ocioso, sem realizar nenhuma atividade.

Comentários:

- Se <mode> for o número de um processo do Zabbix que não está em execução (por exemplo, com 5 pollers em execução, se <mode> for especificado como 6), esse item ficará sem suporte.
- Os valores mínimo e máximo referem-se à porcentagem de uso de um único processo. Assim, 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 informam o que estão fazendo na memória compartilhada, e o processo de self-monitoring resume esses dados a cada segundo. As alterações de estado (busy/idle) são registradas quando ocorrem; portanto, um processo que se torna ocupado é registrado como tal e não altera nem atualiza o estado até ficar ocioso. Isso garante que até mesmo processos completamente travados sejam registrados corretamente como 100% ocupados.
- Atualmente, ”busy” significa ”não está dormindo”, mas, no futuro, estados adicionais poderão ser introduzidos - aguardando bloqueios, executando consultas ao banco de dados etc. Observe que pollers assíncronos são considerados ocupados quando atingem 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] #o tempo médio que os processos poller passaram realizando alguma atividad
zabbix[process,"icmp pinger",max,busy] #o tempo máximo que qualquer processo ICMP pinger passou realizando
zabbix[process,"history syncer",2,busy] #o tempo que o processo history syncer número 2 passou realizando
zabbix[process,trapper,count] #a quantidade de processos trapper atualmente em execução
```

zabbix[proxy,<name>,<param>]

 As informações sobre o proxy do Zabbix.
 Valor retornado: *Integer*.

Parâmetros:

- **name** - o nome do proxy;
- **param**:
  - *lastaccess* - o timestamp da última mensagem de heartbeat recebida do proxy;
  - *delay* - há quanto tempo os valores coletados não são enviados; calculado como ”atraso do proxy” + (”hora atual do server” - ”lastaccess do proxy”), em que ”atraso do proxy” é a diferença entre a hora atual do proxy e o timestamp do valor não enviado mais antigo no proxy.

Comentários:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxies.
- 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, compactação, versão, última vez visto, contagem de hosts, contagem de itens, valores necessários por segundo (vps), status da versão (atual/desatualizada/não compatível), tempos limite por tipo de item, nome do grupo de proxies (se o proxy pertencer a um grupo), estado (desconhecido/offline/online).
 Valor retornado: *objeto JSON*.

Comentário:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxies.

Exemplo de valores retornados:


```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
```

```
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```

zabbix[proxy group,<name>,available]

 O número de proxies online em um grupo de proxies.
 Valor retornado: *Integer*.

Parâmetros:

- **name** - o nome do grupo de proxies.

Comentário:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxies.

zabbix[proxy group,<name>,pavailable]

 A porcentagem de proxies online em um grupo de proxies.
 Valor retornado: *Float*.

Parâmetros:

- **name** - o nome do grupo de proxies.

Comentário:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxies.

zabbix[proxy group,<name>,proxies]

 A lista de proxies do Zabbix em um grupo de proxies, com nome, modo, criptografia, compactação, versão, última vez visto, contagem de hosts, contagem de itens, valores necessários por segundo (vps), status da versão (atual/desatualizada/não compatível), tempos limite, nome do grupo de proxies e estado (desconhecido/offline/online).
 Valor retornado: *JSON*.

Parâmetros:

- **name** - o nome do grupo de proxies.

Comentário:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxies.

zabbix[proxy group,<name>,state]

 O estado de um grupo de proxy.
 Valor retornado: *0* - desconhecido; *1* - offline; *2* - em recuperação; *3* - online; *4* - degradando.

Parâmetros:

- **name** - o nome do grupo de proxy.

Comentário:

- Este item não é compatível com hosts monitorados por um proxy ou 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 necessário de proxies online. Os dados em tempo real incluem o estado do grupo de proxy (consulte os comentários para obter detalhes), o número de proxies online e a porcentagem de proxies online.
 Valor de retorno: *JSON*.

Comentários:

- Este item não é compatível com hosts monitorados por um proxy ou grupo de proxy.
- Este item não retorna proxies sem grupo.
- Se houver um valor inválido para `failover_delay` ou `min_online`, um valor especial −1 será informado para indicar isso. Valores inválidos podem ocorrer quando macros são usadas para a configuração e não podem ser expandidas para um valor válido.
- O estado do grupo de proxy é informado como um número inteiro: *0* - desconhecido; *1* - offline; *2* - recuperação; *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 retornados: *Integer* (para o tamanho); *Float* (para a 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 o erro 'Proxy memory buffer is disabled' quando o buffer de memória está desabilitado;
- Este item não é compatível com o 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 funcionamento atual em que os novos dados estão sendo armazenados.
 Valores de retorno: *0* - disco; *1* - memória.

Comentário:

- 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 retornados: *Integer*.

Comentário:

- Este item não é compatível com o Zabbix server.

zabbix[queue,<from>,<to>]

 O número de items monitorados na fila que estão atrasados por pelo menos <from> segundos, mas menos que <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 proxy do Zabbix não tem conhecimento dos períodos de manutenção; veja Cálculo de filas durante a manutenção para detalhes.

zabbix[rcache,<cache>,<mode>]

 As estatísticas de disponibilidade do cache de configuração do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **cache** - *buffer*;
- **mode**:
  - *total* - o tamanho total do buffer;
  - *free* - o tamanho do buffer livre;
  - *pfree* - a porcentagem de buffer livre;
  - *used* - o tamanho do buffer usado;
  - *pused* - a porcentagem de buffer usado.

zabbix[requiredperformance]

 O desempenho necessário do server Zabbix ou do proxy Zabbix, em novos valores por segundo esperados.
 Valor de retorno: *Float*.

Comentários:

- Correlaciona-se aproximadamente com "Required server performance, new values per second" em *Reports* > *System information*.

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

 Retorna as métricas internas de um server ou proxy do Zabbix. Se <ip> e <port> forem especificados, as métricas serão obtidas da instância remota; caso contrário, da instância local.
 Valores de retorno: *objeto JSON*.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (o padrão é 127.0.0.1);
- **port** - a porta do server/proxy a ser consultado remotamente (o padrão é 10051).

Comentários:

- A solicitação de estatísticas só será aceita a partir dos endereços listados no parâmetro `StatsAllowedIP` server/proxy da instância de destino.
- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, consulte Monitoramento remoto das estatísticas do Zabbix.

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

 Retorna as métricas internas da fila (veja `zabbix[queue,<from>,<to>]`) de um server ou proxy do Zabbix. Se <ip> e <port> forem especificados, as métricas serão obtidas da instância remota; caso contrário, da instância local.
 Valores de retorno: *objeto JSON*.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (padrão é `127.0.0.1`);
- **port** - a porta do server/proxy a ser consultado remotamente (padrão é 10051);
- **from** - atrasado em pelo menos (padrão é 6 segundos);
- **to** - atrasado em no máximo (padrão é infinito).

Comentários:

- A solicitação de stats só será aceita a partir dos endereços listados no parâmetro `StatsAllowedIP` server/proxy na instância de destino.
- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, consulte Monitoramento remoto das stats do Zabbix.

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

zabbix[tcache,cache,<parameter>]

 As estatísticas de eficácia do cache da função de tendências do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **parameter**:
  - *all* - total de solicitações do cache (padrão);
  - *hits* - acertos no cache;
  - *phits* - porcentagem de acertos no cache;
  - *misses* - falhas no cache;
  - *pmisses* - porcentagem de falhas no cache;
  - *items* - o número de itens em cache;
  - *requests* - o número de solicitações em cache;
  - *pitems* - porcentagem de itens em cache em relação a itens em cache + solicitações. Uma porcentagem baixa provavelmente significa que o tamanho do cache pode ser reduzido.
  - *paccessed* - porcentagem de entradas acessadas nas últimas 24 horas.

Comentários:

- Este item não é suportado no proxy do 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 de buffer livre;
  - *used* - o tamanho do buffer usado;
  - *pused* - a porcentagem de buffer usado.

Comentários:

- Este item não é suportado no proxy do Zabbix.

zabbix[vcache,cache,<parameter>]

 As estatísticas de eficácia do cache de valores do Zabbix.
 Valor de retorno: *Integer*. Se <parameter> estiver definido como *mode*, o valor de retorno será: *0* - modo normal; *1* - modo de baixo consumo de memória.

Parâmetros:

- **parameter**:
  - *requests* - o número total de solicitações;
  - *hits* - o número de acertos no cache (valores de histórico obtidos do cache);

    - *misses* - o número de falhas no cache (valores de histórico obtidos do banco de dados);
    - *mode* - o modo de operação do cache de valores.

Comentários:

- Depois que o modo de baixo consumo de memória for ativado, o cache de valores permanecerá nesse estado por 24 horas, mesmo que o problema que acionou esse modo seja resolvido antes.
- Você pode usar 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 a disponibilidade do cache de gravação do Zabbix.
 Valores de retorno: *Integer* (para número/tamanho); *Float* (para porcentagem).

Parâmetros:

- **cache** - *values*, *history*, *index* ou *trend*;
- **mode**:
    - com o parâmetro de cache *values*:
        * *all* - o número total de valores processados pelo server/proxy do Zabbix, exceto itens sem suporte (contador) (padrão);
        * *float* - o número de valores float processados (contador);
        * *uint* - o número de valores inteiros sem sinal processados (contador);
        * *str* - o número de valores de caractere/string processados (contador);
        * *log* - o número de valores de log processados (contador);
        * *text* - o número de valores de texto processados (contador);
        * *bin* - o número de valores binários processados (contador);
        * *json* - o número de valores JSON processados (contador);
        * *not supported* - o número de vezes em que o processamento do item resultou em o item se tornar sem suporte ou permanecer nesse estado (contador);
    - com os parâmetros de cache *history*, *index*, *trend*:
        * *pfree* - a porcentagem de buffer livre (padrão);
        * *total* - o tamanho total do buffer;
        * *free* - o tamanho do buffer livre;
        * *used* - o tamanho do buffer usado;
        * *pused* - a porcentagem do buffer usado.

Comentários:

- O parâmetro de cache `trend` não é suportado com o Zabbix proxy.
- O cache de histórico é usado para armazenar valores de item. Um número baixo indica problemas de desempenho no lado do banco de dados.
- O cache de índice de histórico é usado para indexar os valores armazenados no cache de histórico.
- Depois que o cache de histórico é preenchido e então limpo, o cache de índice de histórico ainda manterá alguns dados. Esse comportamento é esperado e ajuda o sistema a funcionar com mais eficiência, evitando o processamento extra necessário para redimensionar a memória constantemente.
- O cache de tendência armazena o agregado da hora atual para todos os itens que recebem dados.

- Você pode usar a chave `zabbix[wcache,values]` com a etapa de pré-processamento *Change per second* para obter estatísticas de valores por segundo.

Zabbix trapper

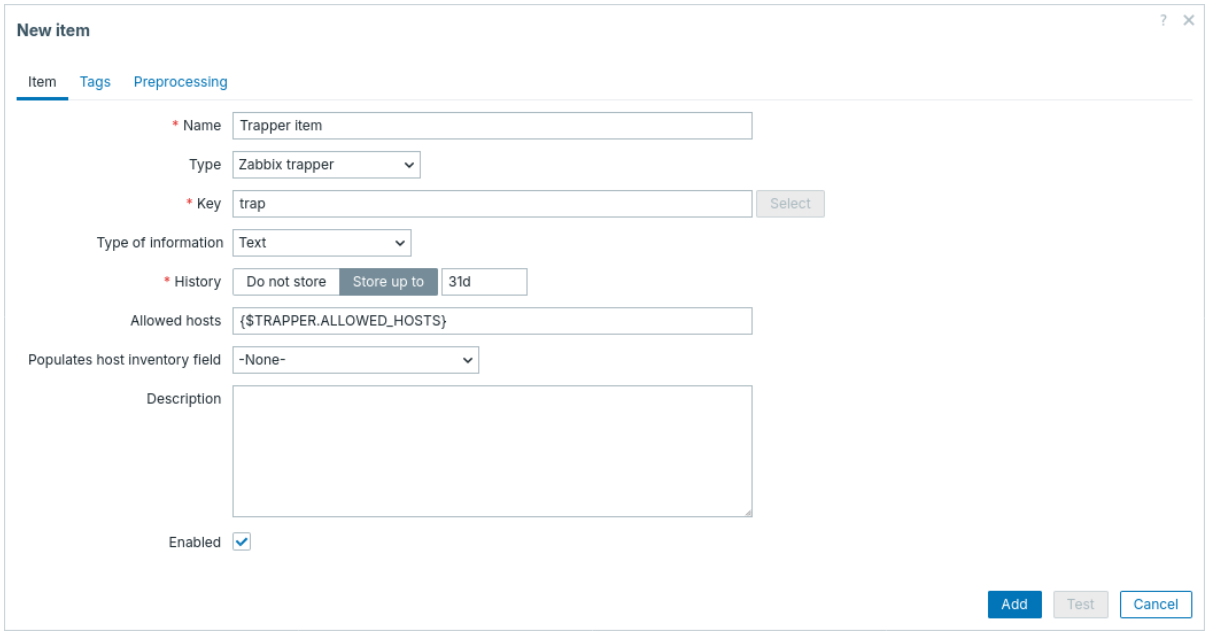
Visão geral

Itens trapper aceitam dados recebidos em vez de consultá-los. Isso é útil para quaisquer dados que você queira enviar ao Zabbix.

Configuração

Para configurar um item trapper:

1. Acesse *Coleta de dados* > *Hosts*.
2. Clique em *Items* na linha do host.
3. Clique em *Criar item*.
4. Insira os parâmetros do item no formulário de configuração.



Todos os campos obrigatórios são marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens trapper são:

Type	Selecione ”Zabbix trapper”.
Key	Digite uma key que será usada para reconhecer o item ao enviar dados para o server Zabbix.
Type of information	Selecione o tipo de informação que corresponderá ao formato dos dados que serão enviados.
Allowed hosts	Lista de endereços IP separados por vírgula (opcionalmente em notação CIDR) ou nomes DNS. O valor padrão é a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}`. Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Você pode inserir os endereços diretamente no campo ou alterar o valor da macro. Se deixado em branco, conexões de entrada serão recusadas de todos os hosts. Se especificado, conexões de entrada serão aceitas somente 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’ serão 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 estão obsoletos pela 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, talvez seja necessário aguardar até 60 segundos após salvar o item até que o server Zabbix processe as alterações a partir de uma atualização do cache de configuração.

Enviando dados

O envio de dados para o Zabbix server ou proxy é possível usando a utilidade 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 <server IP address> -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 *Zabbix sender protocol*.

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;

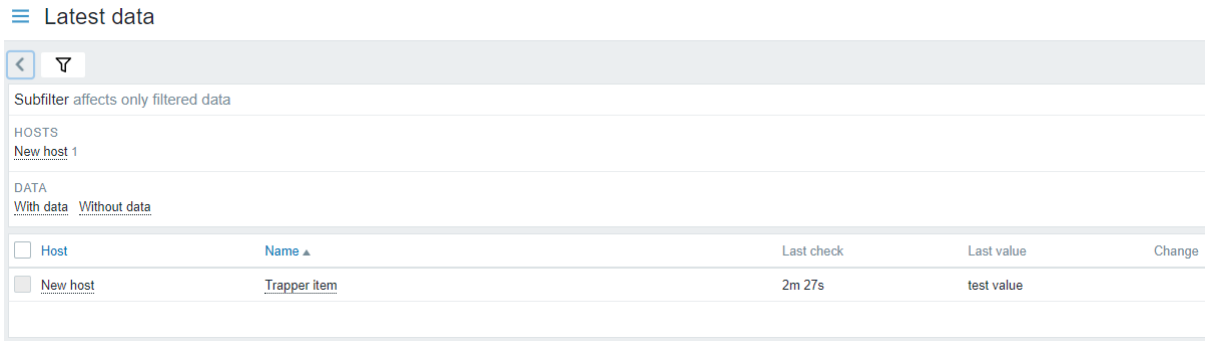
- o item não existe ou ainda não está incluído no cache de configuração do server;
- o item está desabilitado ou seu tipo é diferente de Zabbix trapper ou HTTP agent (com trapping habilitado);
- o IP ou DNS do usuário não está definido na lista de *Allowed hosts* do item;
- outro item possui um valor 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 forem enviados, você pode navegar até *Monitoring* > *Latest data* 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 no tempo do valor.

Verificação externa

Visão geral

Uma verificação externa é uma verificação executada pelo Zabbix server por meio da execução de um script de 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[<parameter1>,<parameter2>,...]`

Onde:

ARGUMENT	DEFINITION
script	Nome de um script de shell ou de um binário.
parameter(s)	Parâmetros opcionais da linha de comando.

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 com o mesmo usuário do Zabbix server/proxy, portanto, quaisquer permissões de acesso ou variáveis de ambiente devem ser tratadas em um script wrapper, se necessário. As permissões no comando também devem permitir que esse usuário o execute. Somente comandos no diretório especificado estão disponíveis para execução.

Warning:
Não use verificações externas em excesso, pois cada script exige que o Zabbix server/proxy inicie um processo fork, e a execução de muitos scripts pode reduzir 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}"]`

Supondo que o host esteja configurado para usar 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 (caractere, log ou tipo de informação de texto) não ficará não suportado em caso de saída de erro padrão.

O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam.

Se o script solicitado não for encontrado ou o Zabbix server/proxy não tiver permissões para executá-lo, o item ficará não suportado e uma mensagem de erro correspondente será exibida.

Em caso de timeout, o item ficará não suportado, uma mensagem de erro correspondente será exibida e o processo criado para o script será encerrado.

Monitor de banco de dados

Visão geral

O tipo de item *Database monitor* no frontend do Zabbix é usado para monitoramento ODBC (verificações ODBC).

ODBC é uma API de middleware da linguagem de programação C para acessar sistemas de gerenciamento de banco de dados (DBMS). O conceito de ODBC foi desenvolvido pela Microsoft e posteriormente portado para outras plataformas.

O Zabbix pode consultar qualquer banco de dados com suporte a 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 comumente usadas.

> **Attention:**
> Veja também: known issues para verificações ODBC.

Instalando o unixODBC

A forma recomendada de instalar o unixODBC é usar os repositórios de pacotes padrão do sistema operacional Linux. Nas distribuições Linux mais populares, o unixODBC já está incluído por padrão no repositório de pacotes. 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 preferência:

```
##### 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 a ODBC, o Zabbix deve ser compilado com a seguinte opção de configuração:

> `--with-unixodbc[=ARG] # Use ODBC driver against unixODBC package.`

Instalando drivers unixODBC

O driver de banco de dados unixODBC deve ser instalado para o banco de dados que será monitorado. Para uma lista de bancos de dados e drivers suportados, consulte a página inicial do unixODBC: http://www.unixodbc.org/drivers.html.

> **Note:**
> Em algumas distribuições Linux, os drivers de banco de dados estão incluídos nos repositórios de pacotes.

MySQL

Para instalar o driver de banco de dados MySQL unixODBC, use o gerenciador de pacotes para o sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install odbc-mariadb

##### Para sistemas baseados em RedHat/Fedora:
dnf install mariadb-connector-odbc

##### Para sistemas baseados em SUSE:
zypper install mariadb-connector-odbc
```

Para instalar o driver de banco de dados sem um gerenciador de pacotes, consulte a documentação do MySQL para `mysql-connector-odbc`, ou a documentação do MariaDB para `mariadb-connector-odbc`.

PostgreSQL

Para instalar o driver de banco de dados 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, use o gerenciador de pacotes para o sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install tdsodbc

##### Para sistemas baseados em RedHat/Fedora (pacotes EPEL: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### Para sistemas baseados em SUSE:
zypper install libtdsodbc0
```

Para instalar o driver de banco de dados sem um gerenciador de pacotes, consulte o Guia do usuário do FreeTDS.

Configurando o unixODBC

Para configurar o unixODBC, você deve editar os arquivos `odbcinst.ini` e `odbc.ini`. Você pode verificar a localização desses arquivos executando o seguinte comando:

```
odbcinst -j
```

O resultado do comando deve conter informações semelhantes ao seguinte:

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
```

odbcinst.ini

O arquivo `odbcinst.ini` lista os drivers de banco de dados ODBC instalados. Se `odbcinst.ini` estiver ausente, é necessário criá-lo manualmente.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parâmetro	Descrição
TEST_MYSQL	Nome do driver do banco de dados.
Description	Descrição do driver do banco de dados.
Driver	Localização da biblioteca do driver do banco de dados.
FileUsage	Determina se o driver do banco de dados suporta 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.
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.

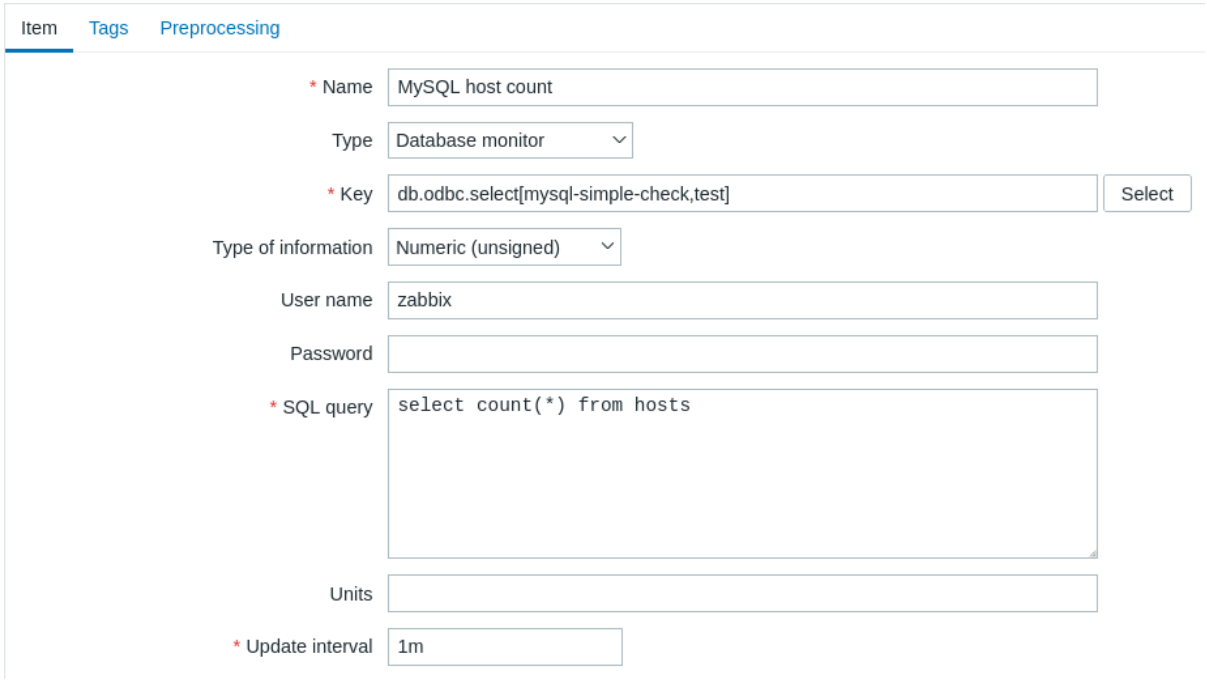
Testando a conexão ODBC

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 de entrada obrigatórios são marcados com um asterisco vermelho.

Para itens de monitoramento de banco de dados, você deve especificar:

Type	Selecione "Database monitor" aqui.
Key	Digite 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.
User name	Digite o nome de usuário do banco de dados (até 255 caracteres). Este parâmetro é opcional se o nome de usuário do banco de dados estiver especificado no arquivo `odbc.ini`. Se uma string de conexão for usada e o campo *User name* não estiver vazio, ela será anexada à string de conexão como `UID=<user>`.
Password	Digite a senha do usuário do banco de dados (até 255 caracteres). Este parâmetro é opcional se a senha estiver especificada no arquivo `odbc.ini`. Se uma string de conexão for usada e o campo *Password* não estiver vazio, ela será anexada à string de conexão como `PWD=<password>`. Caracteres especiais são suportados neste campo. Para todos os bancos de dados, exceto Oracle, senhas contendo `}` devem ser colocadas entre chaves `{}`. A senha será anexada à string de conexão após o nome de usuário, por exemplo, como `UID=<username>;PWD=P?;)*word`. Para testar a string resultante, você pode executar o seguinte comando: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Digite a consulta SQL. Observe que, com `db.odbc.select[]`, a consulta deve retornar apenas um valor.
Type of information	Selecione aqui o tipo de informação que será retornado pela consulta. Se o tipo de informação for selecionado incorretamente, o item ficará sem suporte.

Notas importantes

- Os itens de monitoramento de banco de dados se tornarão sem suporte se nenhum processo *odbc poller* for iniciado na configuração do server ou do 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 timeout de login do ODBC e timeout de execução da consulta. Observe que essas configurações de timeout 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 RDBMS que a processará. A sintaxe da solicitação para um procedimento armazenado deve começar 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[<unique short description>,<dsn>,<connection string>]

 Transforma o resultado da consulta SQL em um array JSON.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **unique short description** - uma breve descrição única para identificar o item (para uso em triggers, etc.);
- **dsn** - o nome da fonte de dados (conforme especificado em `odbc.ini`);
- **connection string** - a string de conexão (pode conter argumentos específicos do driver).

Comentários:

- Embora `dsn` e `connection string` sejam parâmetros opcionais, pelo menos um deles é obrigatório; se ambos estiverem 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 única chamada do 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 na descoberta de baixo nível.

Exemplo:

```
# Conexão para o driver ODBC MySQL 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<descrição curta exclusiva>,<dsn>,<string de conexão>]

 Transforma o resultado da consulta SQL em um array JSON, usado para descoberta de baixo nível. Os nomes das colunas do resultado da consulta são transformados em nomes de macros de descoberta de baixo nível emparelhados com os valores dos campos descobertos. Essas macros podem ser usadas na criação de protótipos de item, trigger, etc.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **descrição curta exclusiva** - uma descrição curta exclusiva para identificar o item (para uso em triggers, etc.);
- **dsn** - o nome da fonte de dados (conforme especificado em `odbc.ini`);
- **string de conexão** - a string de conexão (pode conter argumentos específicos do driver).

Comentários:

- Embora `dsn` e `string de conexão` sejam parâmetros opcionais, pelo menos um deles é obrigatório; se ambos forem definidos, `dsn` será ignorado.

Mensagens de erro

As mensagens de erro do ODBC são estruturadas em campos para fornecer informações detalhadas. Por exemplo, uma mensagem de erro pode ser assim:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "`Cannot execute ODBC query`" - Mensagem do Zabbix
- "`[SQL_ERROR]`" - Código de retorno do ODBC
- "`[42601]`" - SQLState
- "`[7]`" - Código de erro nativo

- "[ERROR: syntax error at or near ";"; Error while executing the query]" - Mensagem de erro nativa

Observe que o comprimento da mensagem de erro é limitado a 2048 bytes, portanto, a mensagem pode ser truncada. Se houver mais de um registro de diagnóstico ODBC, o Zabbix tenta concatená-los (separados por |) até o limite de comprimento permitir.

agent HTTP

Visão geral

Este tipo de item permite a coleta de dados usando o protocolo HTTP/HTTPS. Também é possível fazer trapping usando o utilitário Zabbix sender ou o protocolo do Zabbix sender (para enviar dados ao Zabbix server ou proxy), ou usando o método de API `history.push` (para enviar dados ao Zabbix server).

As verificações de item HTTP são executadas pelo Zabbix server. No entanto, quando os hosts são monitorados por um Zabbix proxy, as verificações de item HTTP são executadas pelo proxy.

As verificações de item HTTP não exigem nenhum agent em execução no host monitorado.

O HTTP agent suporta HTTP e HTTPS. O Zabbix seguirá redirecionamentos opcionalmente (consulte a opção *Follow redirects* abaixo). O número máximo de redirecionamentos é definido de forma fixa como 10 (usando a opção `CURLOPT_MAXREDIRS` do cURL).

> **Attention:**
> O Zabbix server/proxy deve ser configurado inicialmente com suporte a cURL (libcurl).

As verificações HTTP são executadas de forma assíncrona - não é necessário receber a resposta de uma solicitação antes que outras verificações sejam iniciadas. A resolução de DNS também é assíncrona.

A concorrência máxima de verificações assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller).

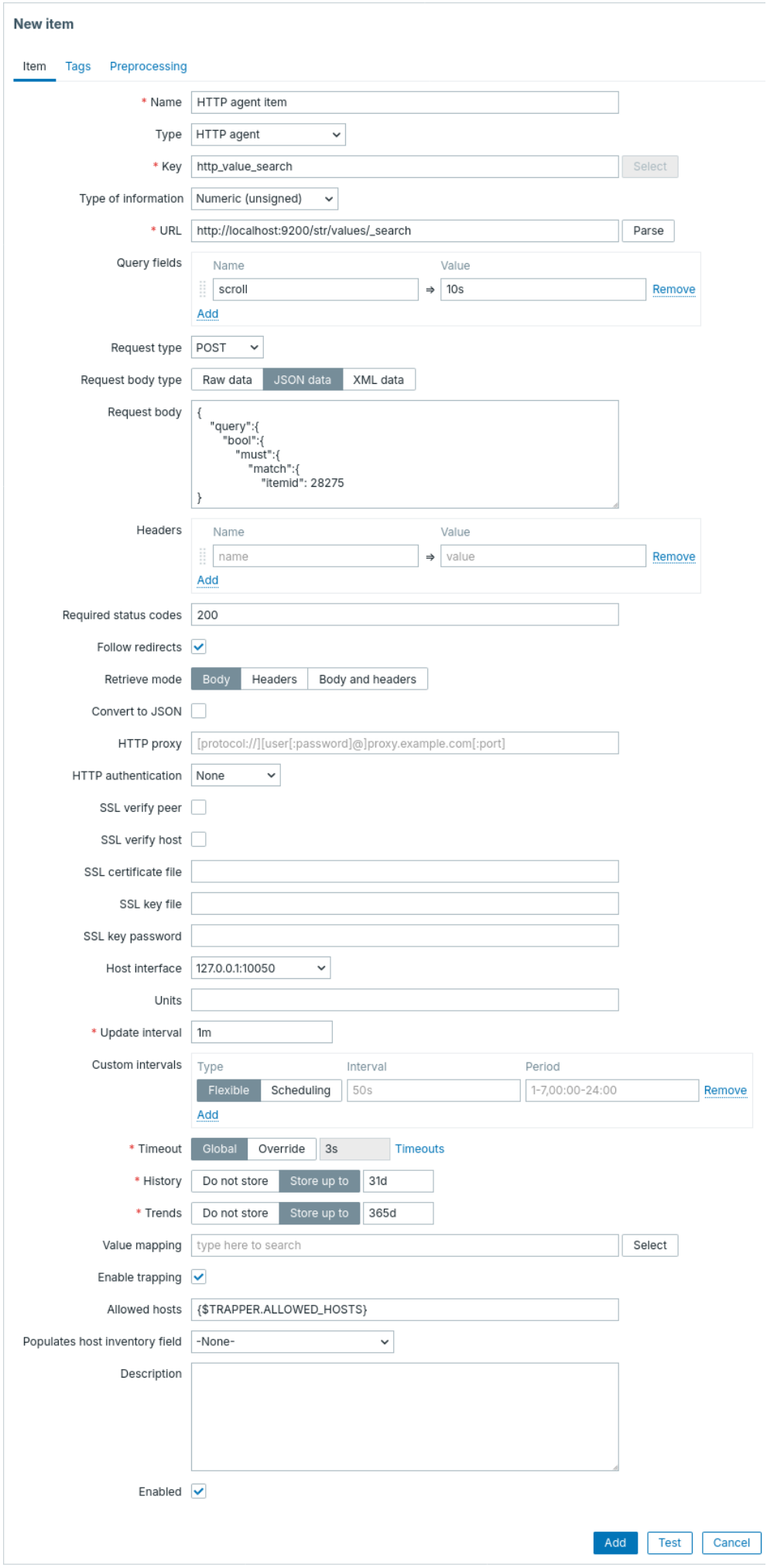
O número de pollers assíncronos do HTTP agent é definido pelo parâmetro StartHTTPAgentPollers.

O recurso de conexões persistentes do cURL foi adicionado às verificações do HTTP agent desde o Zabbix 7.0.

Configuração

Para configurar um item HTTP:

1. Acesse *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.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

Todos os campos de entrada obrigatórios são 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 exclusiva para o item.

URL	URL à qual se conectar e da qual recuperar dados. Por exemplo:
	`https://www.example.com`
	`http://www.example.com/download`
	Os nomes de domínio podem ser especificados usando caracteres Unicode. Eles são convertidos automaticamente para punycode ASCII ao executar a verificação HTTP.
	O botão *Analisar* pode ser usado para separar campos de consulta opcionais (como `?key=value&limit=10`) da URL, movendo os atributos e valores para *Campos de consulta* para codificação automática da URL.
	Por motivos de segurança, não inclua nomes de usuário ou senhas reais nas URLs. Use macros de usuário para evitar a exposição das credenciais.
	Limitado a 2048 caracteres.
	Macros compatíveis: `{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 `CURLOPT_URL` do cURL.

Campos de consulta	Variáveis para a URL (consulte acima). Especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Os valores provenientes de macros são resolvidos e, em seguida, codificados automaticamente na URL. Macros compatíveis: `{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 `CURLOPT_URL` do cURL.
Tipo de solicitação	Selecione o tipo de método da solicitação: *GET*, *POST*, *PUT* ou *HEAD*.
Tipo do corpo da solicitação	Selecione o tipo do corpo da solicitação: **Dados brutos** - corpo de solicitação HTTP personalizado; as macros são substituídas, mas nenhuma codificação é realizada. **Dados JSON** - corpo de solicitação HTTP no formato JSON. As macros podem ser usadas como string, número, true e false; as macros usadas como strings devem estar entre aspas duplas. Os valores provenientes de macros são resolvidos e escapados automaticamente. Se "Content-Type" não for especificado nos cabeçalhos, o padrão será "Content-Type: application/json". **Dados XML** - corpo de solicitação HTTP no formato XML. As macros podem ser usadas como nó de texto, atributo ou seção CDATA. Os valores provenientes de macros são resolvidos e escapados automaticamente em um nó de texto e atributo. Se "Content-Type" não for especificado nos cabeçalhos, o padrão será "Content-Type: application/xml".
Corpo da solicitação	Insira o corpo da solicitação. Macros compatíveis: `{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 solicitação. Especificados como pares de atributo e valor. Macros compatíveis: `{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 `CURLOPT_HTTPHEADER` do cURL.
Códigos de status obrigatórios	Lista dos códigos de status HTTP esperados. Se o Zabbix obtiver um código que não esteja na lista, o item ficará sem suporte. Se estiver vazia, nenhuma verificação será realizada. Por exemplo: 200,201,210−299. Macros compatíveis na lista: macros de usuário, macros de descoberta de baixo nível. Isso usa a opção `CURLINFO_RESPONSE_CODE` do cURL.
Seguir redirecionamentos	Marque a caixa de seleção para seguir redirecionamentos HTTP. Isso define a opção `CURLOPT_FOLLOWLOCATION` do cURL.

Parâmetro	Descrição
Modo de recuperação	Selecione a parte da resposta que deve ser recuperada: **Corpo** - somente o corpo **Cabeçalhos** - somente os cabeçalhos **Corpo e cabeçalhos** - corpo e cabeçalhos
Converter para JSON	Os cabeçalhos são salvos como pares de atributo e valor sob a chave "header" se *Modo de recuperação* não estiver definido como *Corpo*. Se 'Content-Type: application/json' for encontrado, o corpo será salvo como um objeto; caso contrário, será armazenado como string, por exemplo: `{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }`
HTTP proxy	Você pode especificar um HTTP proxy a ser usado, utilizando o formato `[protocol://][username[:password]@]proxy.example.com[:port]`. O prefixo opcional `protocol://` pode ser usado para especificar protocolos de proxy alternativos (por exemplo, `https`, `socks4`, `socks5`; consulte a documentação; o suporte ao prefixo de protocolo foi adicionado no cURL 7.21.7). Quando nenhum protocolo é especificado, o proxy será tratado como um HTTP proxy. Se você especificar o protocolo incorreto, a conexão falhará e o item ficará sem suporte. Por padrão, a porta 1080 será usada. Se especificado, o proxy substituirá variáveis de ambiente relacionadas a proxy, como `http_proxy` e `HTTPS_PROXY`. Se não for especificado, o proxy não substituirá as variáveis de ambiente relacionadas a proxy. O valor inserido é passado "como está"; nenhuma verificação de validade é realizada. Observe que somente a autenticação simples é compatível com HTTP proxy. Macros compatíveis: `{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 `CURLOPT_PROXY` do cURL.
Autenticação HTTP	Selecione a opção de autenticação: **Nenhuma** - nenhuma autenticação é usada; **Básica** - é usada autenticação básica; **NTLM** - é usada autenticação NTLM (Windows NT LAN Manager); **Kerberos** - é usada autenticação Kerberos (consulte também: Configurando o Kerberos com o Zabbix); **Digest** - é usada autenticação Digest. Isso define a opção `CURLOPT_HTTPAUTH` do cURL.
Nome de usuário	Insira o nome de usuário (até 255 caracteres). Este campo está disponível quando *Autenticação HTTP* está definida como *Básica*, *NTLM*, *Kerberos* ou *Digest*. Macros de usuário e macros de descoberta de baixo nível são compatíveis.
Senha	Insira a senha do usuário (até 255 caracteres). Este campo está disponível quando *Autenticação HTTP* está definida como *Básica*, *NTLM*, *Kerberos* ou *Digest*. Macros de usuário e macros de descoberta de baixo nível são compatíveis.
Verificar peer SSL	Marque a caixa de seleção para verificar o certificado SSL do web server. O certificado do server é verificado em relação aos certificados da autoridade certificadora (CA) confiável no local de CA de todo o sistema. Você pode substituir o local dos arquivos de CA usando o parâmetro de configuração SSLCALocation do Zabbix server ou proxy. Isso define a opção `CURLOPT_SSL_VERIFYPEER` do cURL.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo *Common Name* ou *Subject Alternate Name* do certificado do web server corresponde. Isso define a opção `CURLOPT_SSL_VERIFYHOST` do cURL.

Parâmetro	Descrição
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* vazio. Se a chave estiver criptografada, especifique a senha no campo *Senha da chave SSL*. O diretório que contém esse arquivo é especificado pelo parâmetro de configuração `SSLCertLocation` do Zabbix server ou proxy. Macros compatíveis: `{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 `CURLOPT_SSLCERT` do cURL.
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 esse arquivo é especificado pelo parâmetro de configuração `SSLKeyLocation` do Zabbix server ou proxy. Macros compatíveis: `{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 `CURLOPT_SSLKEY` do cURL.
Senha da chave SSL	Senha do arquivo de chave privada SSL. Macros compatíveis: macros de usuário, macros de descoberta de baixo nível. Isso define a opção `CURLOPT_KEYPASSWD` do cURL.
Tempo limite	O Zabbix não gastará mais do que o tempo definido processando a URL (1-600 segundos). Na prática, esse parâmetro define o tempo máximo para estabelecer uma conexão com a URL e o tempo máximo para realizar uma solicitação HTTP. Portanto, o Zabbix não gastará mais do que 2 x *Tempo limite* segundos em uma verificação. Isso define a opção `CURLOPT_TIMEOUT` do cURL. Para obter mais informações sobre o parâmetro *Tempo limite*, consulte atributos gerais do item.
Ativar trapping	Com esta caixa de seleção marcada, o item também funcionará como um item trapper e aceitará dados enviados ao Zabbix server ou proxy usando o utilitário Zabbix sender ou o protocolo do Zabbix sender, ou aceitará dados enviados ao Zabbix server usando o método de API `history.push`. Para obter mais informações sobre o envio de dados, consulte: Itens trapper.
Hosts permitidos	Visível somente se a caixa de seleção *Ativar trapping* estiver marcada. Lista de endereços IP separados por vírgulas (opcionalmente na notação CIDR) ou nomes DNS. O valor padrão é a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}`. Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Você pode inserir os endereços diretamente no campo ou alterar o valor da macro. Se deixado vazio, as conexões recebidas serão negadas de todos os hosts. Se especificado, as conexões recebidas serão aceitas somente dos hosts listados aqui. Se o suporte a IPv6 estiver ativado, `127.0.0.1`, `::127.0.0.1` e `::ffff:127.0.0.1` serão 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 compatíveis, mas foram descontinuados pela 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 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 *HTTP proxy* estiver vazio, outra forma de usar um HTTP proxy 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 um comando shell: *# man curl*.

Attention:
[1] O Zabbix é compatível somente com arquivos de certificado e chave privada no formato PEM. Caso os dados do certificado e da chave privada estejam em um arquivo no formato PKCS #12 (geralmente com a extensão `*.p12` ou `*.pfx`), você poderá gerar o arquivo PEM 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 a 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 requisições POST simples para recuperar dados de serviços como o 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,
```

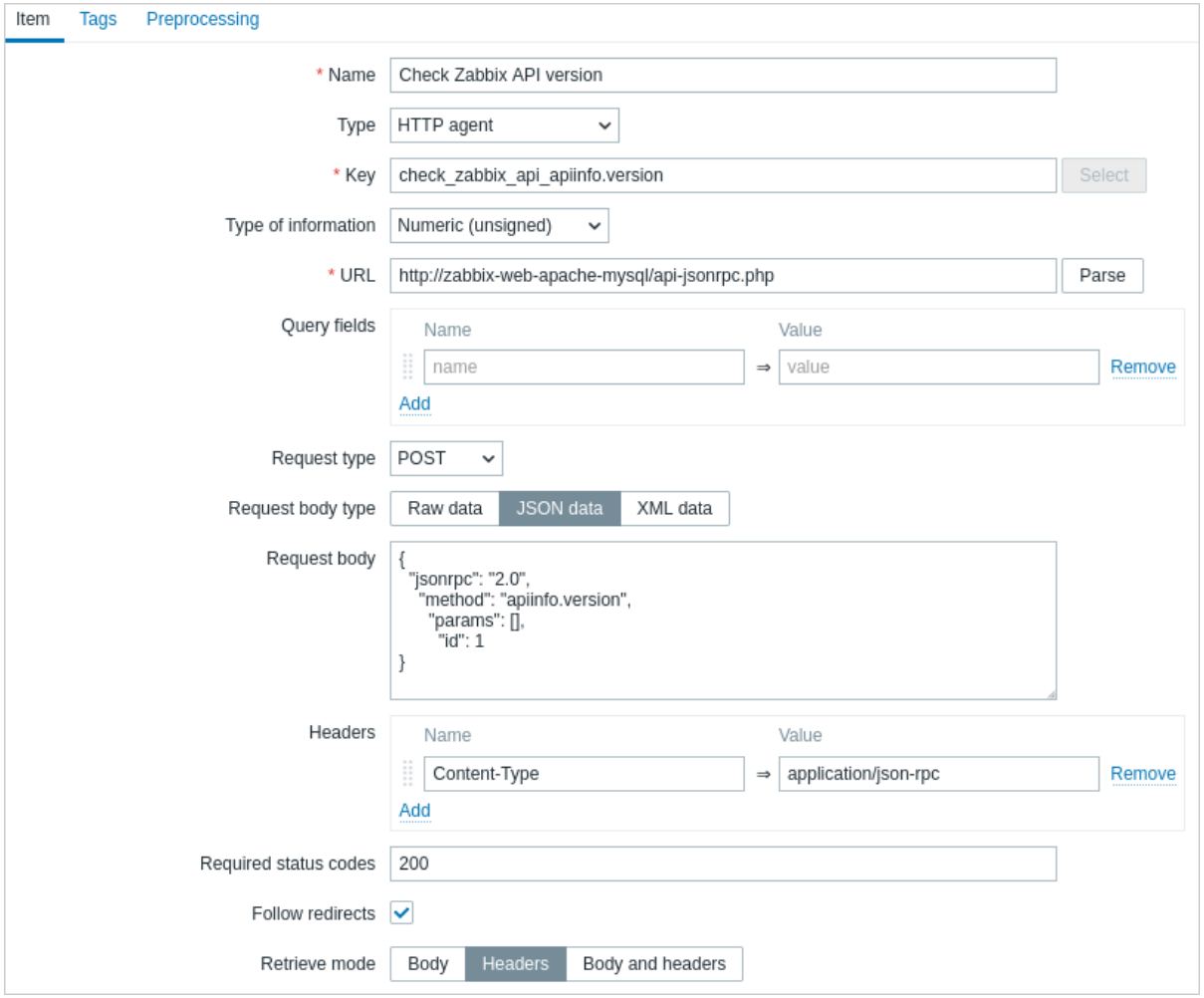
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_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

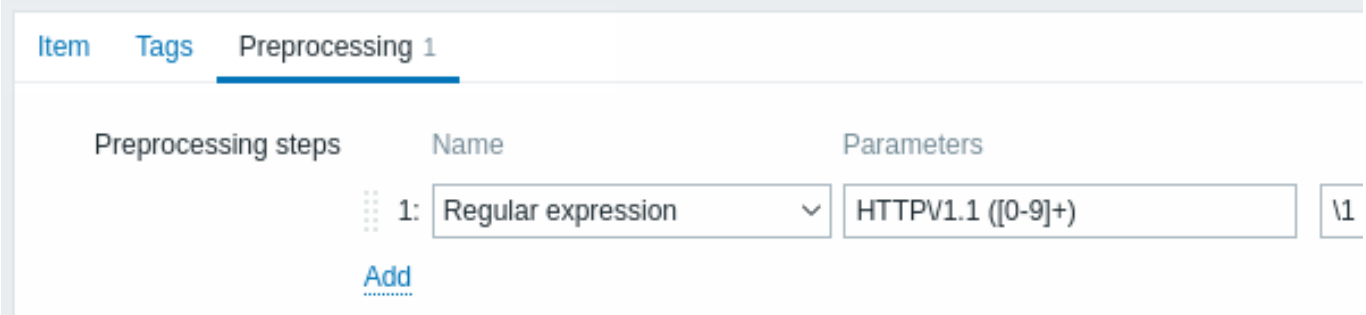
Verificando se a Zabbix API está ativa, usando apiinfo.version.

- Configuração do item:

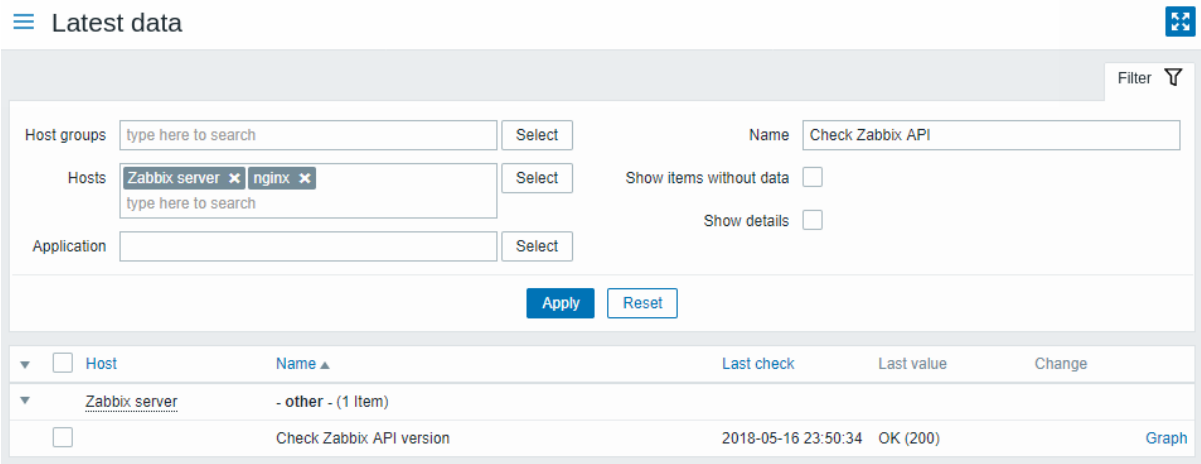


Observe o uso do método POST com dados JSON, definindo cabeçalhos da requisição e solicitando o retorno apenas dos cabeçalhos:

- Pré-processamento do valor do item com expressão regular para obter o código HTTP:



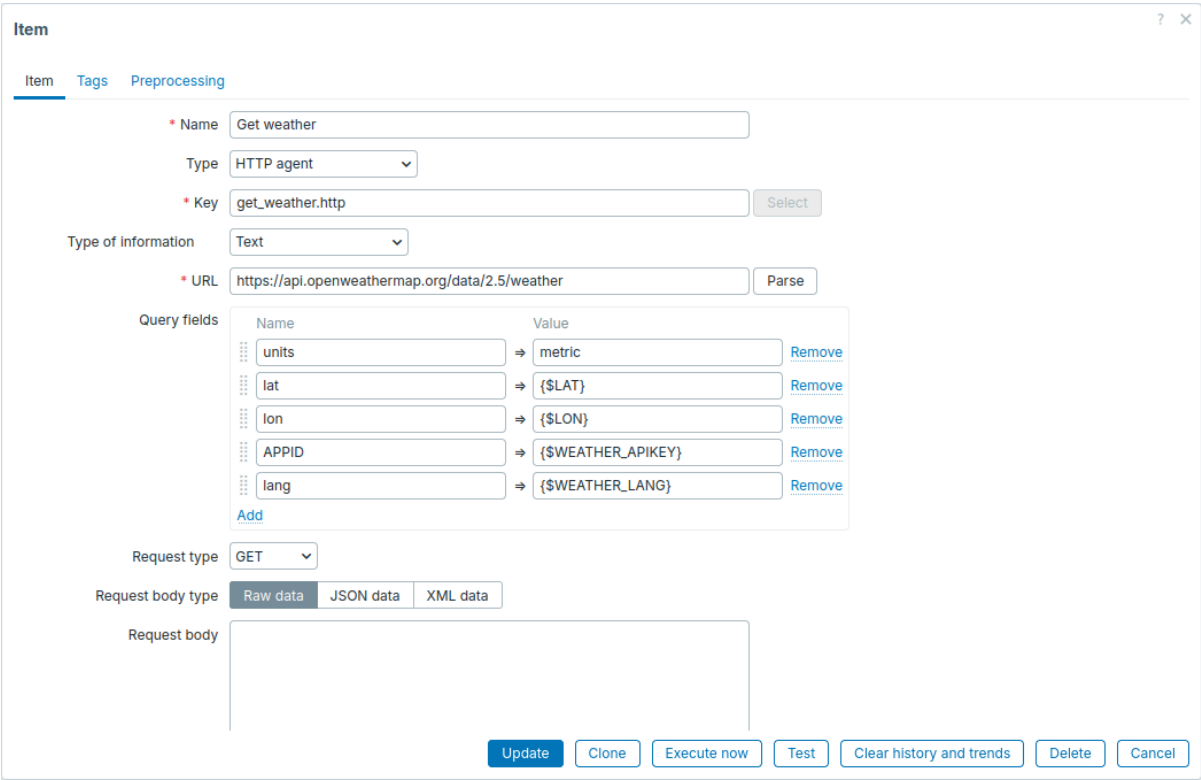
- Verificando o resultado em *Latest data*:



Exemplo 4

Recuperando informações meteorológicas por meio da conexão com o serviço público Openweathermap.

- Configure um item mestre para coleta de dados em lote em um único JSON:



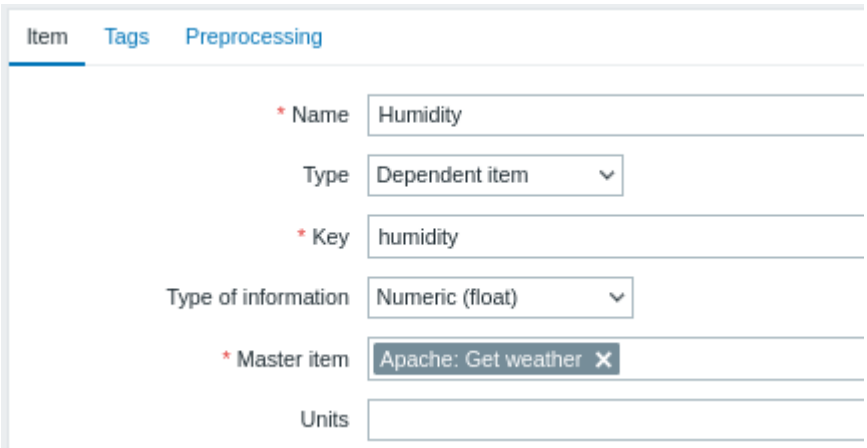
Observe o uso de macros nos campos de consulta. Consulte a Openweathermap API para saber como preenchê-los.

Exemplo de JSON retornado na resposta do 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
    }
}
```

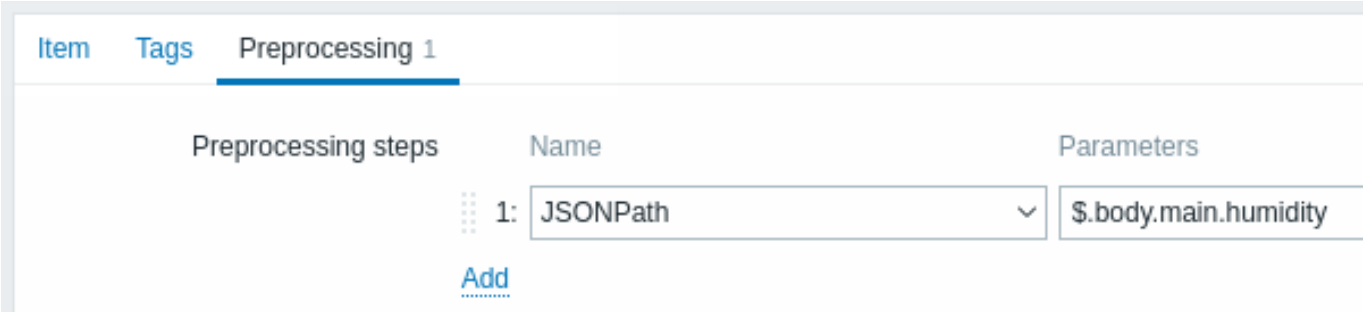

A próxima tarefa é configurar itens dependentes que extraem dados do JSON.

- Configure um item dependente de exemplo para umidade:

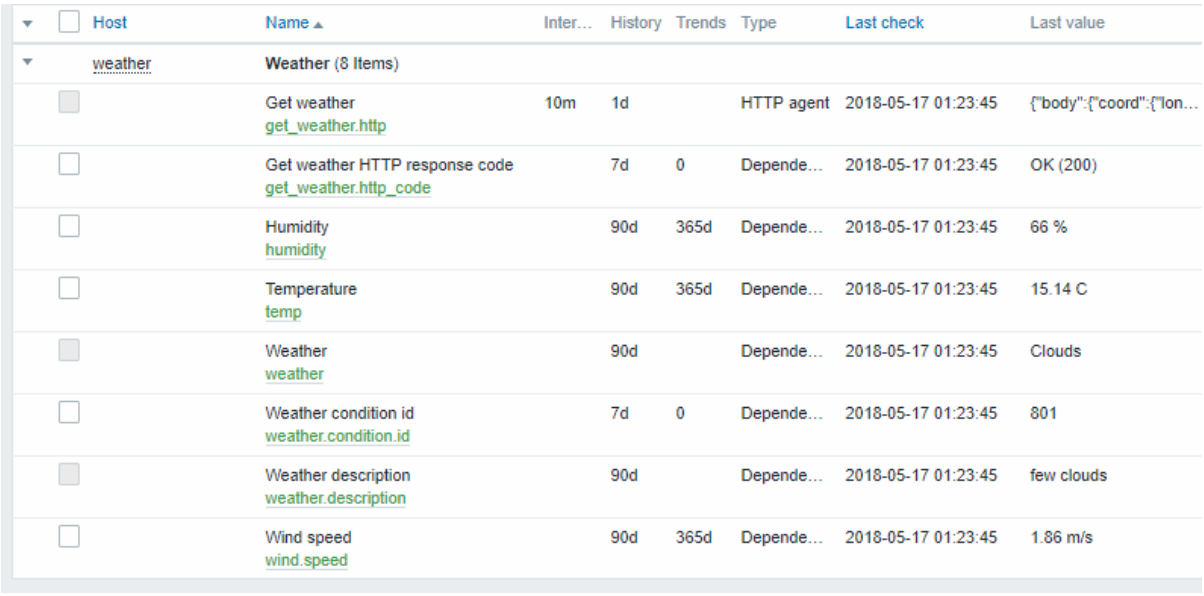


Outras métricas meteorológicas, como 'Temperature', são adicionadas da mesma forma.

- Exemplo de pré-processamento do valor de um item dependente com JSONPath:



- Verifique o resultado dos dados meteorológicos em *Latest data*:

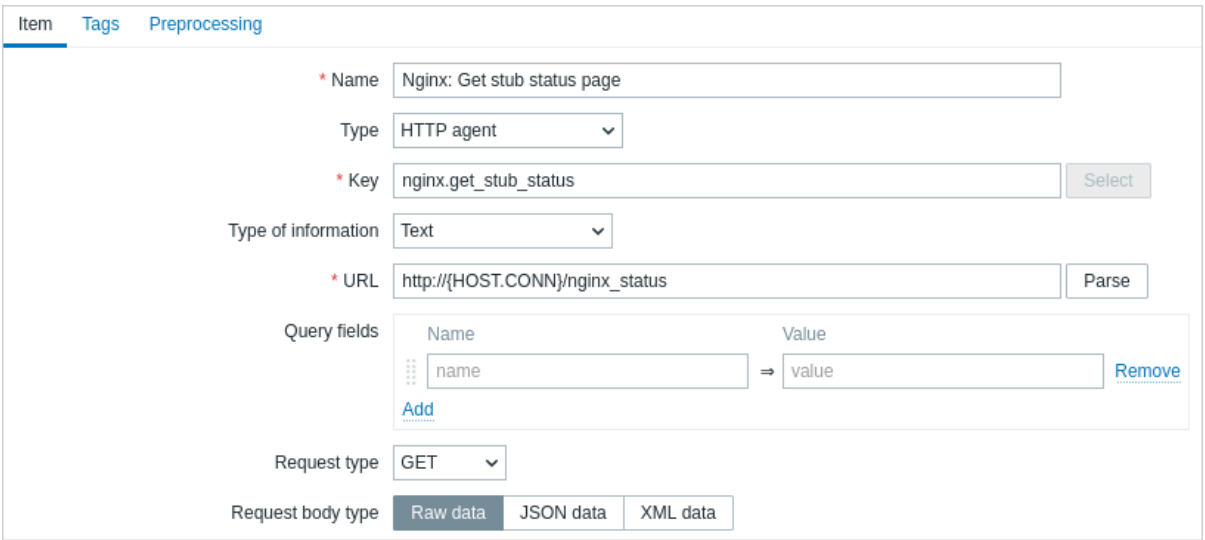


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

Exemplo 5

Conectando-se à página de status do NGINX e obtendo suas métricas em massa.

- Configure o NGINX seguindo o guia oficial.
- Configure um item mestre para a coleta de dados em massa:

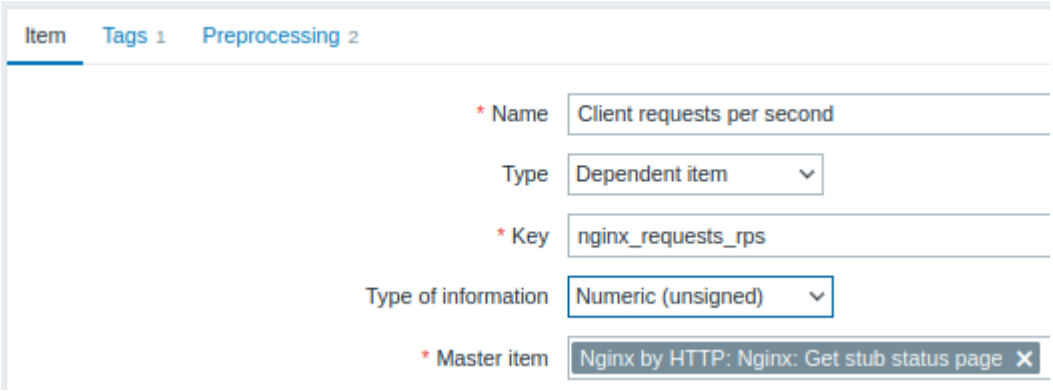


Exemplo de saída do status stub do NGINX:

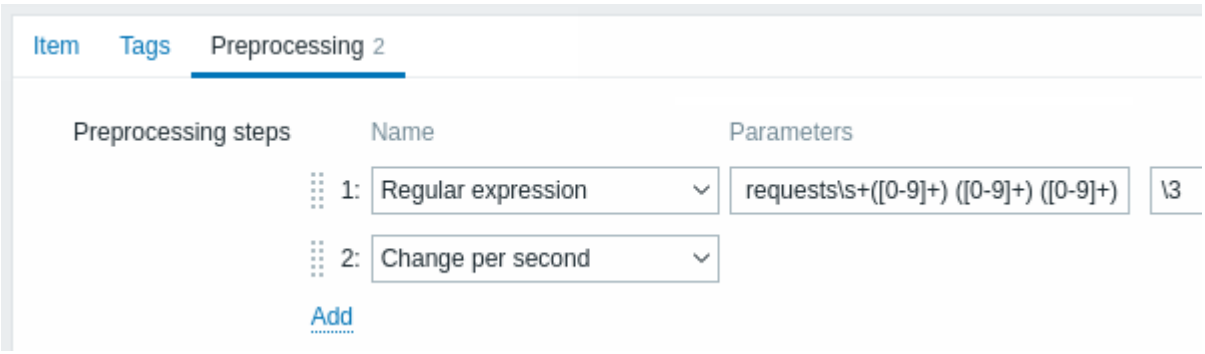
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

A próxima tarefa é configurar itens dependentes que extraiam os dados.

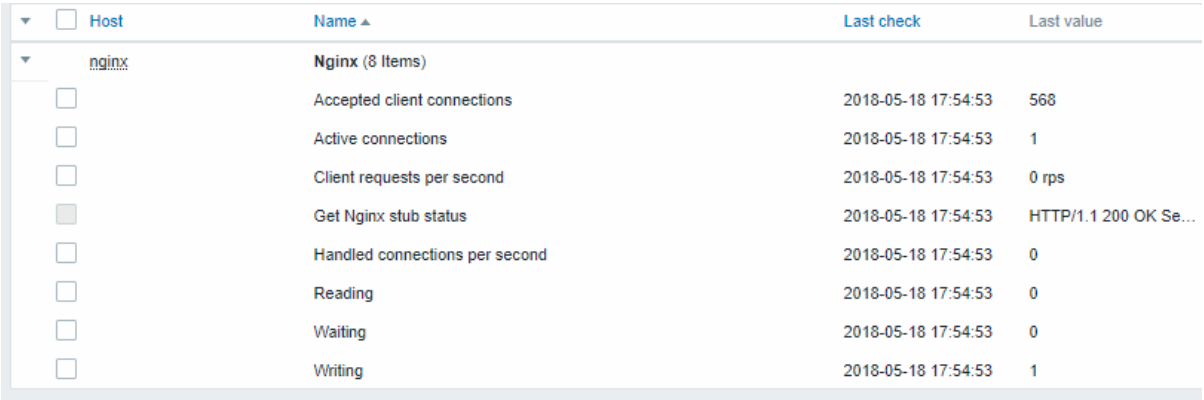
- Configure um item dependente de exemplo para solicitações por segundo:



- Exemplo de pré-processamento do valor do item dependente com a expressão regular `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Verifique o resultado completo do módulo stub em *Latest data*:



Verificação 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 do Prometheus para consultar os dados necessários a partir das métricas coletadas pelo item mestre

Há duas opções de pré-processamento de dados do Prometheus:

- *Padrão do Prometheus* - usado em itens normais para consultar dados do Prometheus
- *Prometheus para JSON* - usado em itens normais e para descoberta de baixo nível. Nesse caso, os dados do Prometheus consultados 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

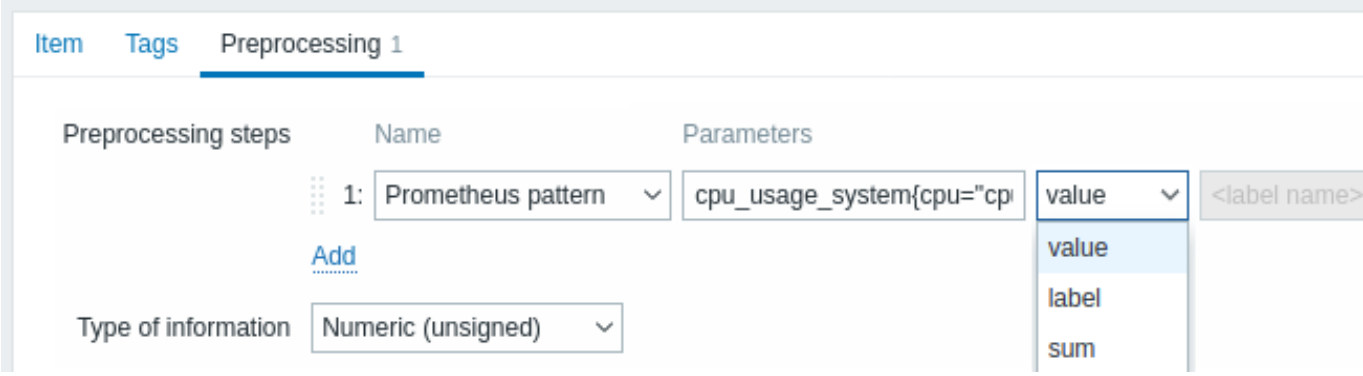
pré-processamento, os dados do Prometheus analisados são armazenados em cache e indexados pela primeira condição `<label>==<value>` na etapa de pré-processamento *Prometheus pattern*. Esse cache é reutilizado ao processar outros itens dependentes neste lote. Para um desempenho ideal, o primeiro label deve ser aquele com o maior número de 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 será 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 use uma etapa de pré-processamento do Prometheus:

1. Insira os parâmetros gerais do item dependente no formulário de configuração.
2. Acesse a aba *Preprocessing*.
3. Selecione uma opção de pré-processamento do Prometheus (*Prometheus pattern* ou *Prometheus to JSON*).



Os parâmetros a seguir são específicos da opção de pré-processamento *Prometheus pattern*:

Parâmetro	Descrição	Exemplos
Pattern	Para definir o padrão de dados necessário, você pode usar uma linguagem de consulta semelhante à linguagem de consulta do Prometheus (consulte a tabela de comparação), por exemplo: <metric name> - selecionar pelo nome da métrica {__name__="<metric name>"} - selecionar pelo nome da métrica {__name__=~"<regex>"} - selecionar pelo nome da métrica correspondente a uma expressão regular {<label name>="<label value>",...} - selecionar pelo nome do label {<label name>=~"<regex>",...} - selecionar pelo nome do label correspondente a uma expressão regular {__name__=~".*"}==<value> - selecionar pelo valor da métrica Ou uma combinação dos itens acima: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> O valor do label pode ser qualquer sequência de caracteres UTF-8, mas os caracteres de barra invertida, aspas duplas e avanço de linha devem ser escapados como \\, \" e \n, respectivamente; os demais caracteres não devem ser escapados.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=".*"}* *cpu_usage_system{cpu="cpu-total",host="* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

Parâmetro	Descrição	Exemplos
Result processing	Especifique se deseja retornar o valor, o label ou aplicar a função apropriada (se o padrão corresponder a várias linhas e o resultado precisar ser agregado): **value** - retornar o valor da métrica (erro se várias linhas corresponderem) **label** - retornar o valor do label especificado no campo *Label* (erro se várias métricas corresponderem) **sum** - retornar a soma dos valores **min** - retornar o valor mínimo **max** - retornar o valor máximo **avg** - retornar o valor médio **count** - retornar a quantidade de valores Este campo está disponível somente para a opção *Prometheus pattern*.	Consulte também os exemplos de uso dos parâmetros abaixo.
Output	Defina o nome do label (opcional). Nesse caso, será retornado o valor correspondente ao nome do label. Este campo está disponível somente para a opção *Prometheus pattern*, se 'Label' estiver selecionado no campo *Result processing*.	

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:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - selecione 'value'

2. Você também pode ter interesse no valor **médio** de todos os parâmetros `node_disk_usage_bytes`:

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - selecione 'avg'

3. Embora o Prometheus ofereça suporte somente a dados numéricos, é comum usar uma solução alternativa para retornar também a descrição textual relevante. Isso pode ser feito com um filtro e especificando o label. Assim, para retornar o valor do label '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:

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - selecione 'label'
- *Label* - especifique 'color'

O filtro (baseado no valor numérico '1') corresponderá à linha apropriada, enquanto o label retornará a descrição do status de integridade (atualmente 'green', mas potencialmente também 'red' ou 'yellow').

Prometheus para JSON

Os dados do Prometheus podem ser usados para descoberta de baixo nível. Nesse caso, são necessários dados no formato JSON, e a opção de pré-processamento *Prometheus to JSON* retornará exatamente isso.

Para mais detalhes, consulte Discovery using Prometheus data.

Comparação da linguagem de consulta

A tabela a seguir lista as diferenças e semelhanças entre PromQL e a linguagem de consulta de pré-processamento do Zabbix para Prometheus.

Seletor de vetor instantâneo do PromQL		Zabbix Prometheus preprocessing
Diferenças		
Destino da consulta	Prometheus server	Texto simples no formato de exposição do Prometheus
Retorna	Vetor instantâneo	Valor de métrica ou rótulo (padrão Prometheus) Array de métricas para valor único em JSON (Prometheus to JSON)
Operadores de correspondência de rótulos	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Expressão regular usada na correspondência de rótulo ou nome de métrica	RE2	PCRE
Operadores de comparação	Consulte a lista	Apenas **==** (igual) é suportado para filtragem de valor
Semelhanças		
Seleção por nome de métrica que é igual a uma string	<metric name> ou {__name__="<metric name>"}	<metric name> ou {__name__="<metric name>"}
Seleção por nome de métrica que corresponde a uma expressão regular	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Seleção por valor de <label name> que é igual a uma string	{<label name>="<label value>",...}	{<label name>="<label value>",...}
Seleção por valor de <label name> que corresponde a uma expressão regular	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
Seleção por valor que é igual a uma string	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

IPMI agent

Visão geral

Você pode monitorar a integridade e a disponibilidade de dispositivos com Intelligent Platform Management Interface (IPMI) no Zabbix. Para realizar verificações de IPMI, o Zabbix server deve ser inicialmente configurado com suporte a IPMI.

IPMI é uma interface padronizada para o gerenciamento remoto "lights-out" ou "out-of-band" de sistemas de computador. Ela permite monitorar diretamente o status do hardware por meio das chamadas placas de gerenciamento "out-of-band", independentemente do sistema operacional ou de a máquina estar ligada.

O monitoramento de IPMI do Zabbix funciona somente para dispositivos com suporte a IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP etc.).

Um processo gerenciador de IPMI agenda as verificações de IPMI por meio dos pollers de IPMI. Um host é sempre consultado por apenas um poller de IPMI por vez, reduzindo o número de conexões abertas com os controladores BMC. Assim, é seguro aumentar o número de pollers de IPMI sem se preocupar com a sobrecarga do controlador BMC. O processo gerenciador de IPMI é iniciado automaticamente quando pelo menos um poller de IPMI é iniciado.

Consulte também os problemas conhecidos relacionados às verificações de 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 de IPMI, portanto quaisquer items de 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
```

Remova o comentário e defina a quantidade de pollers, por exemplo, para 3, de modo que fique assim:

`StartIPMIPollers=3`

Salve o arquivo e reinicie o zabbix_server depois disso.

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 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 há mais atualizações planejadas para a especificação IPMI).

O primeiro parâmetro a ser analisado é "reading_type". Use a "Tabela 42-1, Event/Reading Type Code Ranges" das especificações para decodificar o código "reading_type". A maioria dos sensores em nosso exemplo tem "reading_type:0x1", o que significa sensor 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 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 deslocamento 00h significa "PowerOff/Power Down". Em outras palavras, se o bit menos significativo for 1, então o server está desligado. Para testar esse bit, a função `bitand` com máscara '1' pode ser usada. A expressão do trigger pode ser como

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

para avisar sobre um server desligado.

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.

SSH agent

Visão geral

As verificações SSH são realizadas como monitoramento sem agent. O Zabbix agent não é necessário para verificações SSH.

Para realizar verificações SSH, o Zabbix server 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, recomenda-se libssh 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 itens SSH, são necessárias algumas alterações na configuração do server.

Abra o arquivo de configuração do Zabbix server (`zabbix_server.conf`) como `root` e procure a seguinte linha:

```
##### SSHKeyLocation=
```

Remova o comentário e defina o caminho completo para a pasta onde as chaves pública e privada serão armazenadas:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Salve o arquivo e reinicie o Zabbix server em seguida.

O caminho `/home/zabbix` aqui é o diretório home da conta de usuário *zabbix*, e `.ssh` é um diretório onde, por padrão, as chaves pública e privada 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 ocorre com 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 `SSHKeyLocation` do Zabbix server mencionado acima.

> **Note:**
> As etapas a seguir podem ser ignoradas se a conta *zabbix* tiver sido adicionada manualmente de acordo com a seção de instalação. Nesse caso, o diretório home da conta *zabbix* provavelmente já é `/home/zabbix`.

Para alterar o diretório home da conta de usuário *zabbix*, todos os processos em execução que o estejam utilizando precisam ser interrompidos:

```
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 tenha existido 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 esteja seguro, comandos adicionais podem ser executados para definir permissões no diretório home:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Os processos que foram interrompidos anteriormente agora podem ser iniciados novamente:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Agora, as etapas para gerar as chaves pública e privada podem ser executadas com os seguintes comandos (para melhor legibilidade, os prompts de comando foram 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):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |        .        |
##### |       o         |
##### | .      o        |
##### |+      . S       |
##### |.+    o =        |
##### |E .    * =       |
##### |=o . ..* .       |
```

```
##### |... oo.o+        |
##### +-----------------+
```

Note:
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 talvez não sejam 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 comandos a seguir, o arquivo da 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 foram 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>
##### Agora tente fazer login na máquina com "ssh 'root@10.10.10.10'",
##### e verifique se apenas a(s) chave(s) desejada(s) foram adicionadas.
```

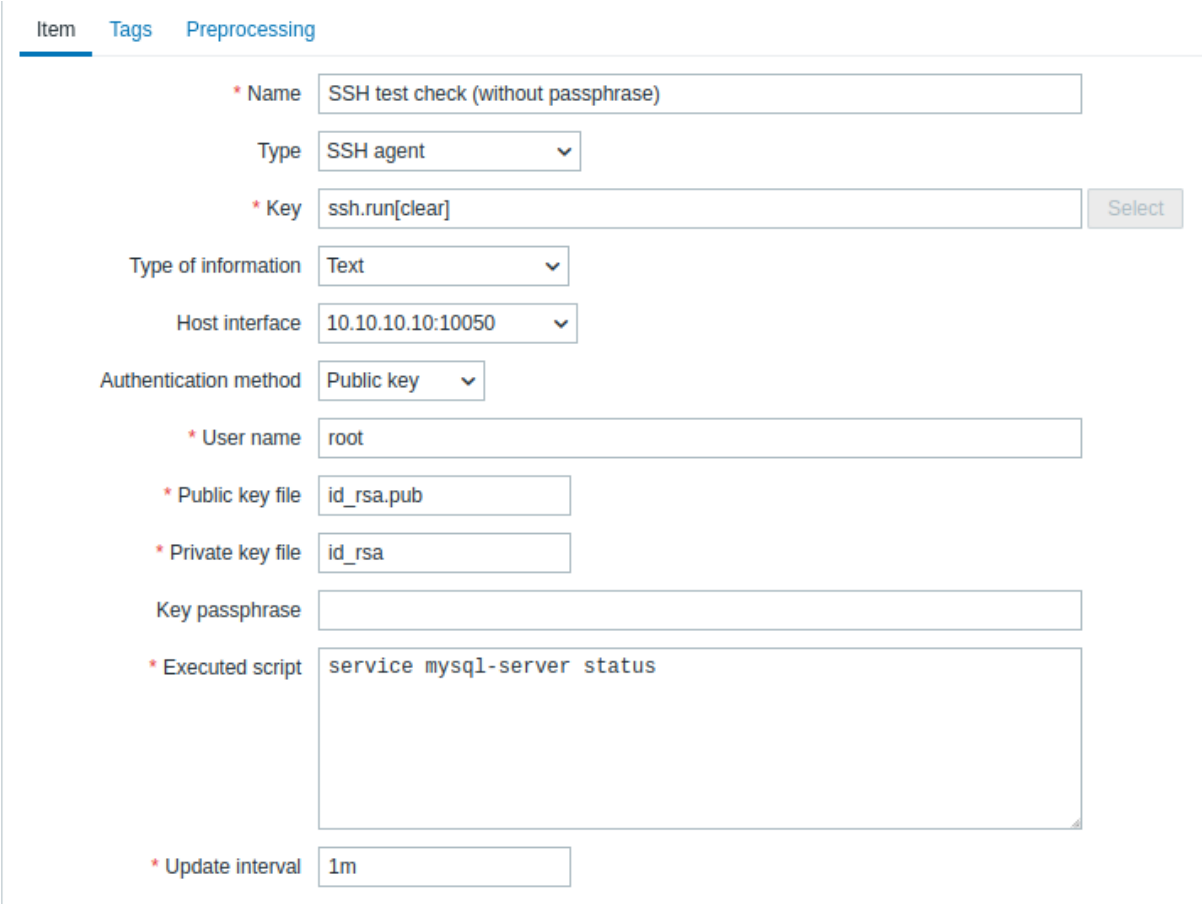
Agora é possível verificar o login SSH usando a chave privada padrão (`/home/zabbix/.ssh/id_rsa`) para a conta de usuário *zabbix*:

```
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

Os comandos que devem ser executados devem ser inseridos no campo *Script executado* na configuração do item. Vários comandos podem ser executados um após o outro, inserindo-os em uma nova linha. Nesse caso, os valores retornados também serão formatados em várias linhas.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens SSH são:

Parâmetro	Descrição	Comentários
Tipo	Selecione **SSH agent** aqui.	

Parâmetro	Descrição	Comentários

Chave	Chave exclusiva (por host) do item no formato `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	**unique short description** é obrigatória e deve ser exclusiva para cada item SSH por host.
		
		A porta padrão é 22, não a porta especificada na interface à qual este item está atribuído.
		
		ssh options permite passar opções SSH adicionais no formato `key1=value1;key2=value2,value3`. Vários valores para uma chave podem ser passados separados por vírgula (nesse caso, o parâmetro deve ser colocado entre aspas); várias chaves de opção podem ser passadas separadas por ponto e vírgula.
		
		As seguintes chaves de opção são compatíveis: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. O suporte à chave e ao valor da opção depende da biblioteca SSH (por exemplo, `PubkeyAcceptedKeyTypes` é compatível somente com libssh); se uma opção não for compatível, um erro será retornado e o item ficará sem suporte.
		
		Observe que o sinal "+" para adicionar configurações de cifras e "!" para desabilitar configurações de cifras específicas (como no GnuTLS e no OpenSSL) não são compatíveis.
		
		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 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 métodos "Password" ou "Public key".	
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 "Public key". Obrigatório.	Exemplo: *id_rsa.pub* - nome de arquivo padrão da chave pública gerada pelo comando ssh-keygen.
Arquivo de chave privada	Nome do arquivo da chave privada se o *Método de autenticação* for "Public key". Obrigatório.	Exemplo: *id_rsa* - nome de arquivo padrão da chave privada.

Parâmetro	Descrição	Comentários
Senha ou *Frase secreta da chave*	Senha (até 255 caracteres) para autenticação ou Frase secreta **se** ela tiver sido usada para a chave privada.	Deixe o campo *Frase secreta da chave* vazio se nenhuma frase secreta tiver sido usada. Consulte também os problemas conhecidos relacionados ao uso de frases secretas.

Script executado	Comando(s) shell executado(s) usando uma sessão SSH remota.	O valor retornado pelos comandos shell executados é limitado a 16 MB (incluindo espaços em branco finais, que são truncados); os limites do banco de dados também se aplicam.
		
		Observe que a biblioteca libssh2 pode truncar scripts executáveis para aproximadamente 32 kB.
		
		Exemplos:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Exemplo (para o subsistema NETCONF):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Telnet agent

Visão geral

As verificações Telnet são realizadas como monitoramento sem agent. O agent Zabbix não é necessário para verificações Telnet.

Campos configuráveis

O(s) comando(s) real(is) a ser(em) executado(s) devem ser colocados no campo **Script executado** na configuração do item.
 Vários comandos podem ser executados um após o outro, colocando-os em uma nova linha. Neste caso, o valor retornado também será formatado como multilinhas.

Caracteres suportados com os quais o prompt do shell pode terminar:

- $
- #
- >
- %

Note:
Uma linha de prompt do telnet que termina com um desses caracteres será removida do valor retornado, mas apenas para o primeiro comando na lista de comandos, ou seja, apenas no início da sessão telnet.

Chave	Descrição
telnet.run[<descrição curta única>,<ip>,<porta>,<codificação>]	Executa um comando em um dispositivo remoto usando conexão telnet

Attention:
Se uma verificação telnet retornar um valor com caracteres não-ASCII e em codificação diferente de UTF8, então o parâmetro *<codificação>* da chave deve ser especificado corretamente. Veja a página codificação de valores retornados para mais detalhes.

agente 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.

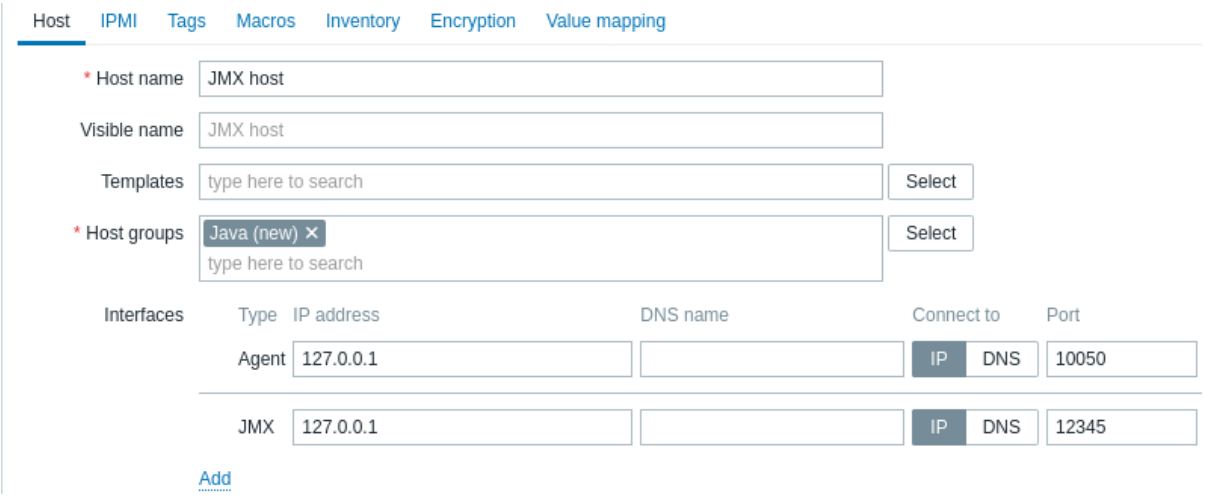
Veja Monitoramento e Gerenciamento usando JMX para uma descrição detalhada.

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.

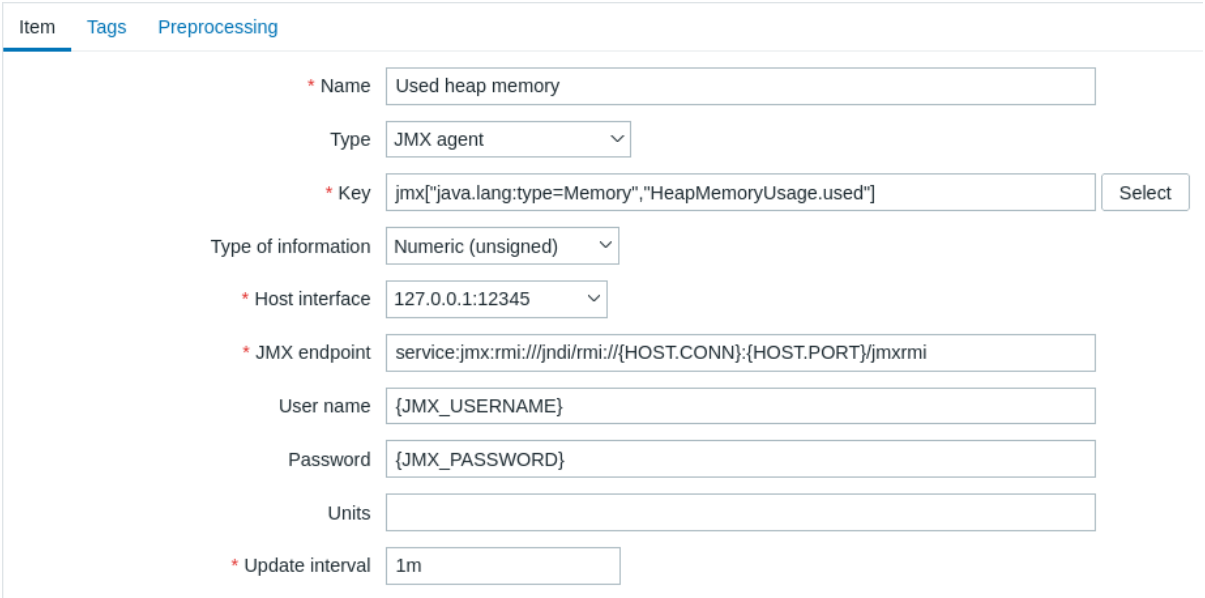


Todos os campos obrigatórios são marcados com um asterisco vermelho.

Adicionando item JMX agent

Para cada contador JMX de seu interesse, adicione um item **JMX agent** associado a essa interface.

A chave na captura de tela abaixo diz `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens JMX são:

Type	Defina **JMX agent** aqui.
Key	A chave do item `jmx[]` contém três parâmetros: **object name** - o nome de objeto de um MBean **attribute name** - o nome de um atributo de MBean com nomes opcionais de campos de dados compostos separados por pontos **unique short description** - uma descrição exclusiva que permite vários itens JMX com o mesmo nome de objeto e nome de atributo no host (opcional) Veja abaixo mais detalhes sobre as chaves de item JMX. Você pode descobrir MBeans e atributos de MBean usando um item `jmx.discovery[]` de low-level discovery.
JMX endpoint	Você pode especificar um JMX endpoint personalizado. Certifique-se de que os parâmetros de conexão do JMX endpoint correspondam à interface JMX. Isso pode ser feito usando macros `{HOST.*}` como no JMX endpoint padrão. As macros `{HOST.*}` e macros de usuário são suportadas.

User name	Especifique o nome de usuário (até 255 caracteres), se você tiver configurado autenticação em sua aplicação Java. Macros de usuário são suportadas.
Password	Especifique a senha (até 255 caracteres), se você tiver configurado autenticação em sua aplicação Java. Macros de usuário são suportadas.

Se você quiser monitorar um contador Boolean que seja ”true” ou ”false”, especifique o tipo de informação como ”Numeric (unsigned)” e selecione a etapa de pré-processamento ”Boolean to decimal” na aba Preprocessing. O server armazenará valores Boolean 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 sobre o formato da chave do item section.

Na verdade, isso é tudo o que há para saber. 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. Se não, então você pode fazer isso de acordo com a documentação.
- O server Zabbix e o Java gateway estão instalados com o prefixo `/usr/local/`
- O JBoss já está instalado em `/opt/jboss-eap-6.4/` e está sendo executado em modo standalone
- Vamos assumir que todos esses componentes funcionam no mesmo host
- Firewall e SELinux estão desativados (ou configurados adequadamente)

Vamos fazer algumas configurações simples em `zabbix_server.conf`:

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

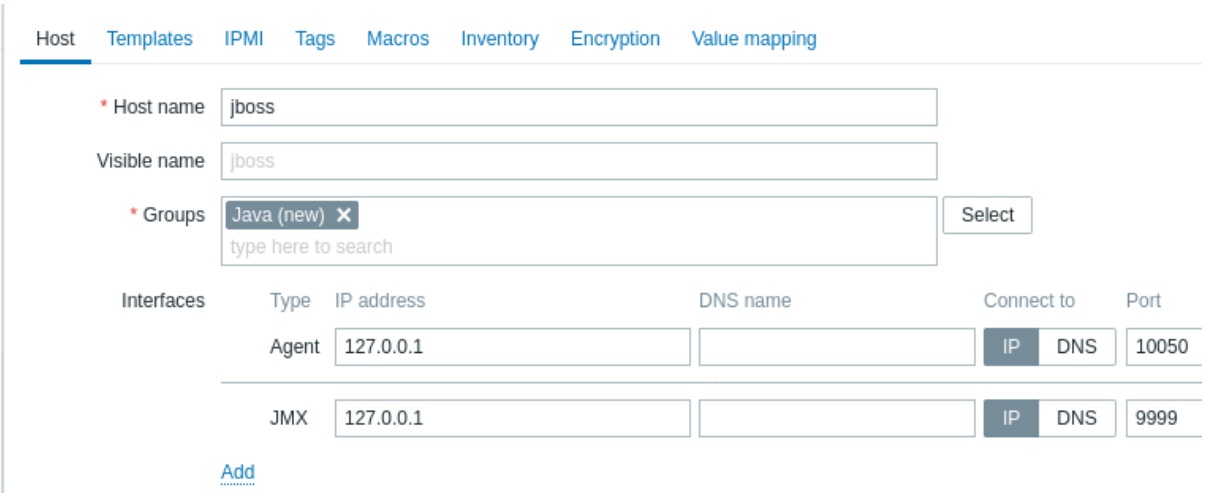
E no arquivo de configuração `zabbix_java/settings.sh` (ou `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Verifique se o JBoss escuta sua porta padrão de gerenciamento:

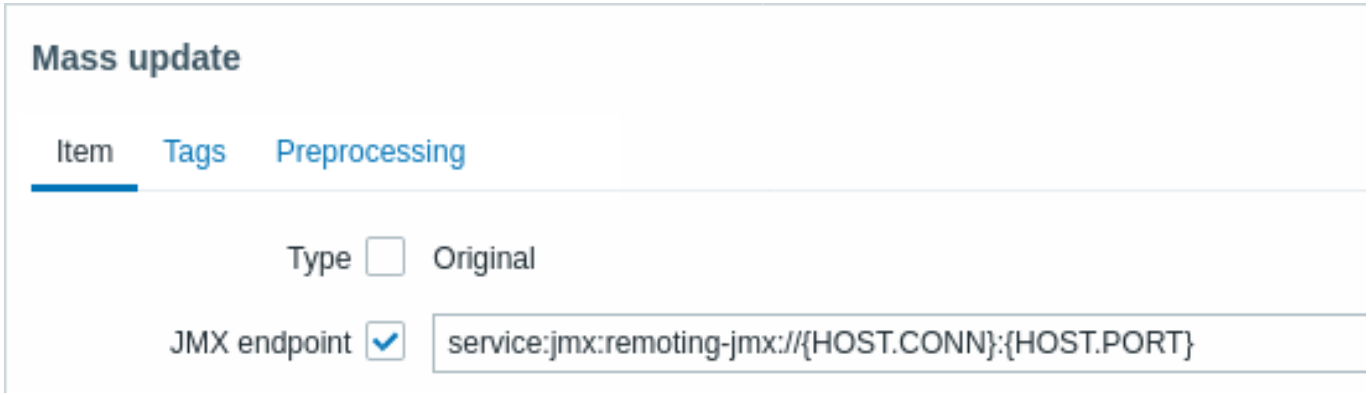
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

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 de RMI, podemos atualizar em massa o parâmetro de endpoint JMX para os items em nosso template JMX de acordo com isso:

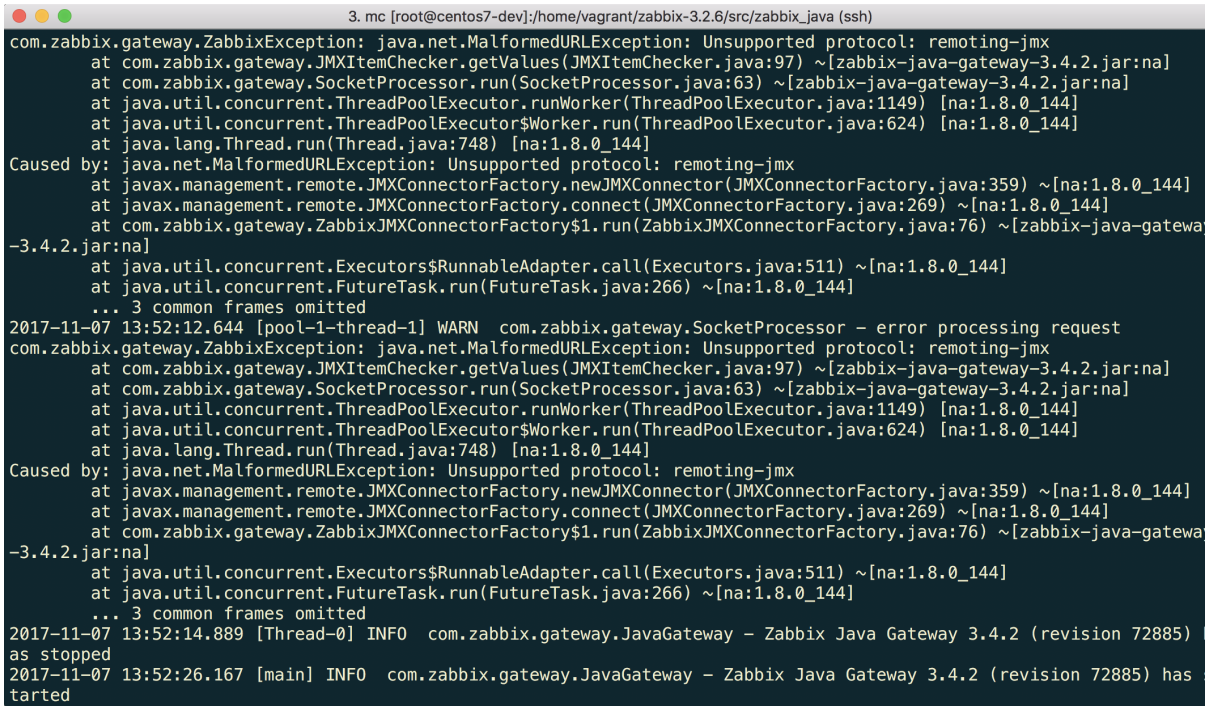
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Vamos atualizar o cache de configuração:

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

Observe que você pode encontrar um erro primeiro.



"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 corretamente, ver que os dados de monitoramento JMX começam a chegar ao Zabbix (veja também: Latest data).

Item calculado

Visão geral

Este tipo de item permite realizar cálculos sobre as métricas que já existem. Um item calculado não coleta dados por conta própria.

Por exemplo, você pode:

- Contar dois valores.
- Calcular a média de vários valores de item.
- Calcular o valor total de um grupo de itens.

O valor resultante é armazenado no banco de dados do Zabbix como qualquer outro item; tanto os valores de histórico quanto os de tendência são armazenados, e gráficos podem ser gerados.

O cálculo é baseado em uma fórmula. A sintaxe da fórmula é compartilhada com as expressões de trigger.

Na verdade, é possível manter os triggers simples colocando a lógica complexa de transformação de dados em um item calculado e, em seguida, referenciar o item calculado em um trigger usando a chave do item.

Há duas possibilidades para especificar a fonte de dados dos cálculos:

1. Especificar os itens na fórmula.
2. Fazer agregações de dados com base em um filtro que seleciona os itens.

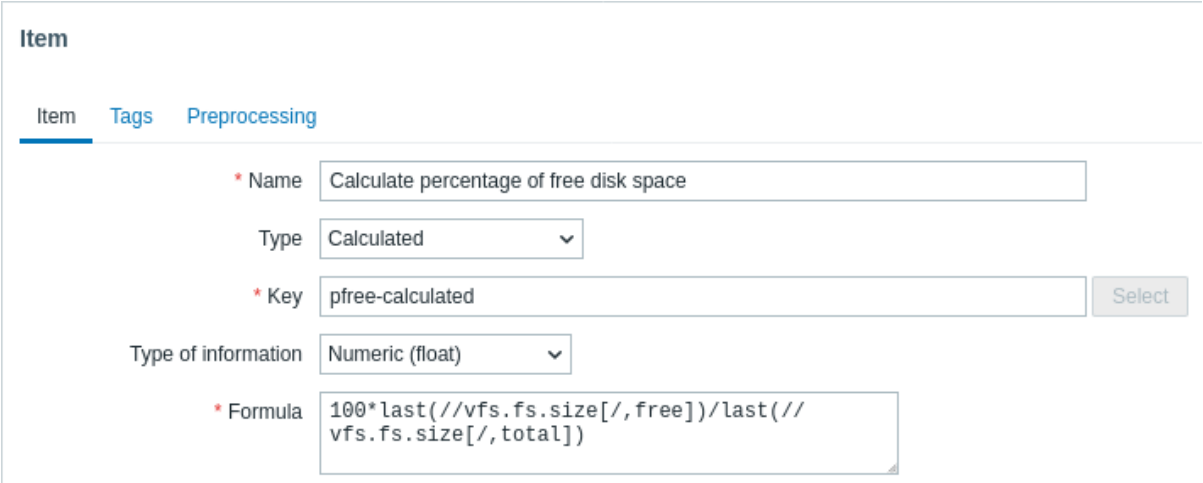
A segunda opção permite calcular médias de grupos, totais etc.

Todos os cálculos são realizados pelo Zabbix server e tratados pelos processos `history poller`; proxy ou agents não estão envolvidos. Observe que o uso extensivo de itens calculados com períodos longos pode afetar o desempenho do server.

Observe que os itens calculados podem acessar o histórico de outros itens, incluindo itens aos quais o usuário que está criando o item calculado não tem acesso. Se esse comportamento não for desejado, você pode desabilitar os itens calculados definindo o parâmetro de configuração `StartHistoryPollers` como 0.

Campos configuráveis

Para usar itens calculados, escolha o tipo de item **Calculated**.



A **key** é um identificador exclusivo do item (por host). Você pode criar qualquer nome de key usando os símbolos compatíveis. A key permite referenciar o item calculado em triggers.

A definição do cálculo deve ser inserida no campo **Formula**.

Não há conexão entre a fórmula e a key. Os parâmetros da key não são usados na fórmula de nenhuma maneira.

Diferentemente das expressões de trigger, o Zabbix processa itens calculados de acordo com o intervalo de atualização do item, e não ao receber um novo valor.

Observe que, se você selecionar *Numeric (unsigned)* como tipo de informação e o resultado do cálculo for um valor de ponto flutuante, ele será truncado para um número inteiro.

Fórmula de cálculo simples

A sintaxe de uma fórmula simples é:

`function(/host/key,<parameter1>,<parameter2>,...)`

onde:

Elemento	Descrição
`function`	Uma das funções suportadas: last, min, max, avg, count etc.
`host`	host do item usado para o cálculo. O host atual pode ser omitido (ou seja, como em `function(//key,parameter,...)`).
`key`	Chave do item usada para o cálculo. Itens que retornam valores com o tipo de dados binário ou JSON não são suportados. Se você alterar a chave do item de um item referenciado, a chave deverá ser atualizada manualmente na fórmula. Todos os itens referenciados por funções de histórico devem existir e estar coletando dados.
`parameter(s)`	Parâmetros da função, se necessário. Sufixos de tempo e sufixos de tamanho de memória são suportados.

Uma fórmula mais complexa pode usar uma combinação de funções, operadores e parênteses.

Veja exemplos de fórmulas de item calculado:

- (1) Tráfego total na interface
- (2) Percentual de espaço livre em disco
- (3) Percentual de tráfego de entrada
- (4) Número médio de valores processados

Algumas variáveis podem ser passadas para a fórmula com macros de usuário.
Observe que as macros de usuário:

- serão resolvidas - se usadas para referenciar um parâmetro de função, parâmetro de filtro de item ou uma constante.
- **não** serão resolvidas - se estiverem referenciando uma função, nome do host, chave do item, parâmetro da chave do item ou operador.

A comparação com strings é permitida.

Se não houver dados recentes no cache e não houver um período de consulta definido na função, o Zabbix, por padrão, retrocederá até uma semana no passado para consultar o banco de dados em busca de valores históricos.

Um item calculado pode se tornar não suportado em vários casos:

- item referenciado
  - não é encontrado
  - está desabilitado ou pertence a um host desabilitado
  - não é suportado (exceto com a função nodata() e operadores com valores desconhecidos)
- não há dados para calcular uma função
- divisão por zero
- sintaxe incorreta usada

Agregações de dados

As agregações de dados são baseadas em um filtro que seleciona múltiplos itens. As agregações de dados também são um tipo de item calculado, porém, você usa uma sintaxe ligeiramente diferente na fórmula.

Para obter agregações, 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.

Como alternativa, 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.

Veja exemplos de cálculos de agregação:

- (5) Tráfego de entrada total para o host
- (6) Espaço total em disco para o grupo
- (7) Carga média do processador para o grupo
- (8) Carga média do processador para o grupo e tag
- (9) Número médio de valores processados
- (10) Número médio de consultas para o grupo
- (11) Número total de verificações DNS bem-sucedidas
- (12) Número total de itens não suportados para o grupo
- (13) Estatísticas de interface para o grupo

Macros de usuário e macros de descoberta de baixo nível são suportadas em:

- parâmetros de chave de item
- parâmetros de função
- condições de filtro (nome do grupo de hosts e nome da tag)
- constantes da expressão

Um cálculo de agregação 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 itens calculados

(1) Tráfego total na interface

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Calcula a largura de banda total na eth0.

(2) Percentual de espaço livre em disco

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

O Zabbix obterá os valores mais recentes de espaço livre e espaço total em disco (em /) e calculará o percentual de acordo com a fórmula fornecida.

(3) Percentual do tráfego de entrada

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Calcula o percentual do tráfego de entrada em relação ao tráfego total.

(4) Número médio de valores processados

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Calcula uma média de 10 minutos do número de valores processados pelo Zabbix.

(5) Tráfego total de entrada para o host

```
sum(last_foreach(/host/net.if.in[*]))
```

Calcula o total de todos os itens que correspondem a `net.if.in[*]` no host.

(6) Espaço total em disco para o grupo

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Calcula o espaço total em disco para todos os `MySQL Servers`.

(7) Carga média do processador para o grupo

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Calcula a carga média do processador para todos os `MySQL Servers`.

(8) Carga média do processador por grupo e tag

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Calcula a carga média da CPU em todos os hosts de vários grupos de hosts que possuem tags específicas.

(9) Número médio de consultas para o grupo

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Calcula a média de 5 minutos do número de consultas por segundo para `MySQL Servers`.

(10) Número total de verificações DNS bem-sucedidas

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Calcule o total de verificações DNS bem-sucedidas em todos os hosts. O formato de item mostrado é `net.dns[192.0.2.1,example.com,A]` como exemplo de uma possível key.

Observe que os curingas devem corresponder ao número de parâmetros na key (aqui `net.dns` tem três parâmetros: ip, name, type).

(11) Número total de itens não suportados para o grupo

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Calcula o número total de itens não suportados para `Zabbix servers`.

(12) Estatísticas de interface para o grupo

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Calcule as estatísticas de interface para o grupo `video`.

Exemplos de sintaxe correta/incorreta

Expressões (incluindo chamadas de função) não podem ser usadas como parâmetros de histórico, tendência ou foreach function. No entanto, essas funções podem ser usadas nos parâmetros de outras funções (não históricas).

Expression	Example
Válida	`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)))`
Inválida	`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))
```

(em que `resptime[*]` é um exemplo de um padrão de chave de item que corresponde às chaves `resptime[1]`, `resptime[2]` etc., usado para obter o tempo de resposta de vários nós ativos do cluster), não é possível garantir que ambas as partes da equação sempre terão o mesmo conjunto de valores. Enquanto uma parte da expressão é avaliada, um novo valor para o período solicitado pode chegar e, nesse caso, a outra parte da expressão terá um conjunto diferente de valores.

item dependente

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.

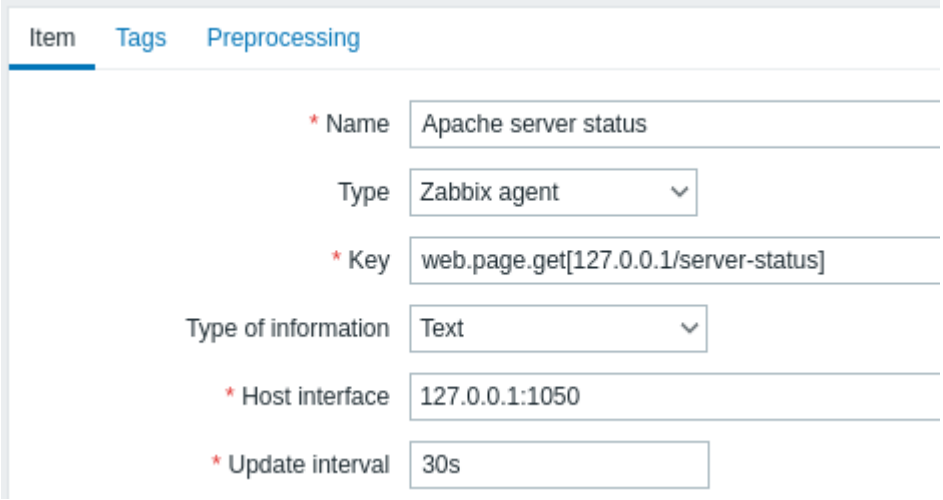
Limitações

- 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 do seu item mestre para obter dados. Por isso, o **item mestre** deve ser configurado (ou existir) primeiro:

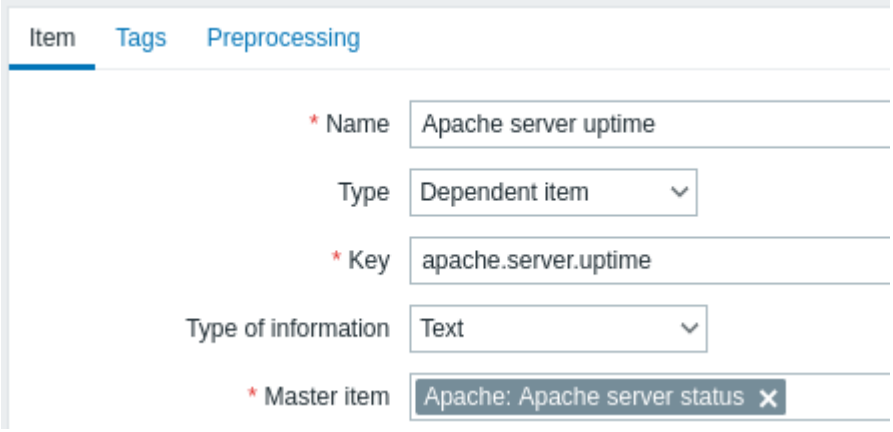
1. Vá para *Coleta de dados* > *Hosts*.
2. Clique em *Items* na linha do host.
3. Clique em *Criar item*.
4. Insira os parâmetros do item no formulário.



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Clique em *Adicionar* para salvar o item mestre.

Depois, você pode configurar um **item dependente**.



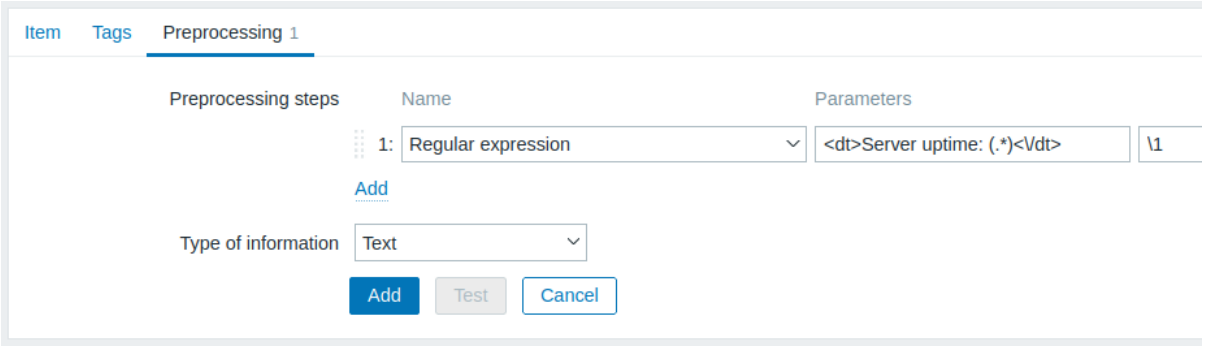
Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens dependentes são:

Type	Selecione **Item dependente** aqui.

Key	Digite uma chave que será usada para reconhecer o item.
Master item	Selecione o item mestre. O valor do item mestre será usado para preencher o valor do item dependente.
Type of information	Selecione o tipo de informação que corresponderá ao formato dos dados que serão armazenados.

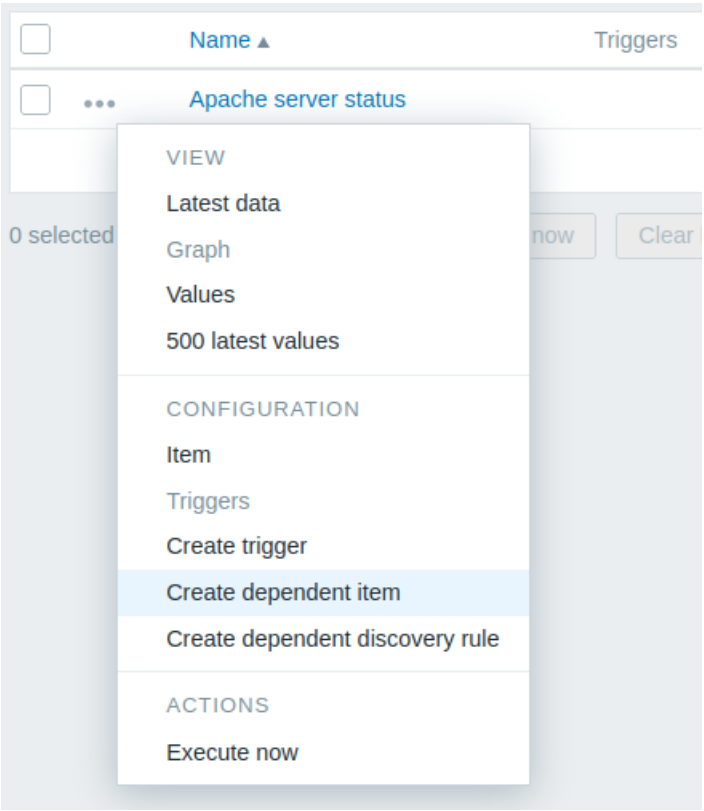
Você pode usar o pré-processamento do valor do item para extrair a parte necessária do valor do item mestre.



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.

Item 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;
```

Item de navegador

Visão geral

Os itens Browser permitem monitorar sites e aplicações web complexos usando um browser.

> **Attention:**
> O suporte a itens Browser está atualmente em caráter experimental.

Os itens Browser coletam dados executando um código JavaScript definido pelo usuário e recuperando dados via HTTP/HTTPS. Este item pode simular ações relacionadas ao browser, como clicar, inserir texto, navegar por páginas da web e outras interações do usuário com sites ou aplicações web.

Além do script, é possível especificar uma lista opcional de parâmetros (pares de nome e valor) e timeout.

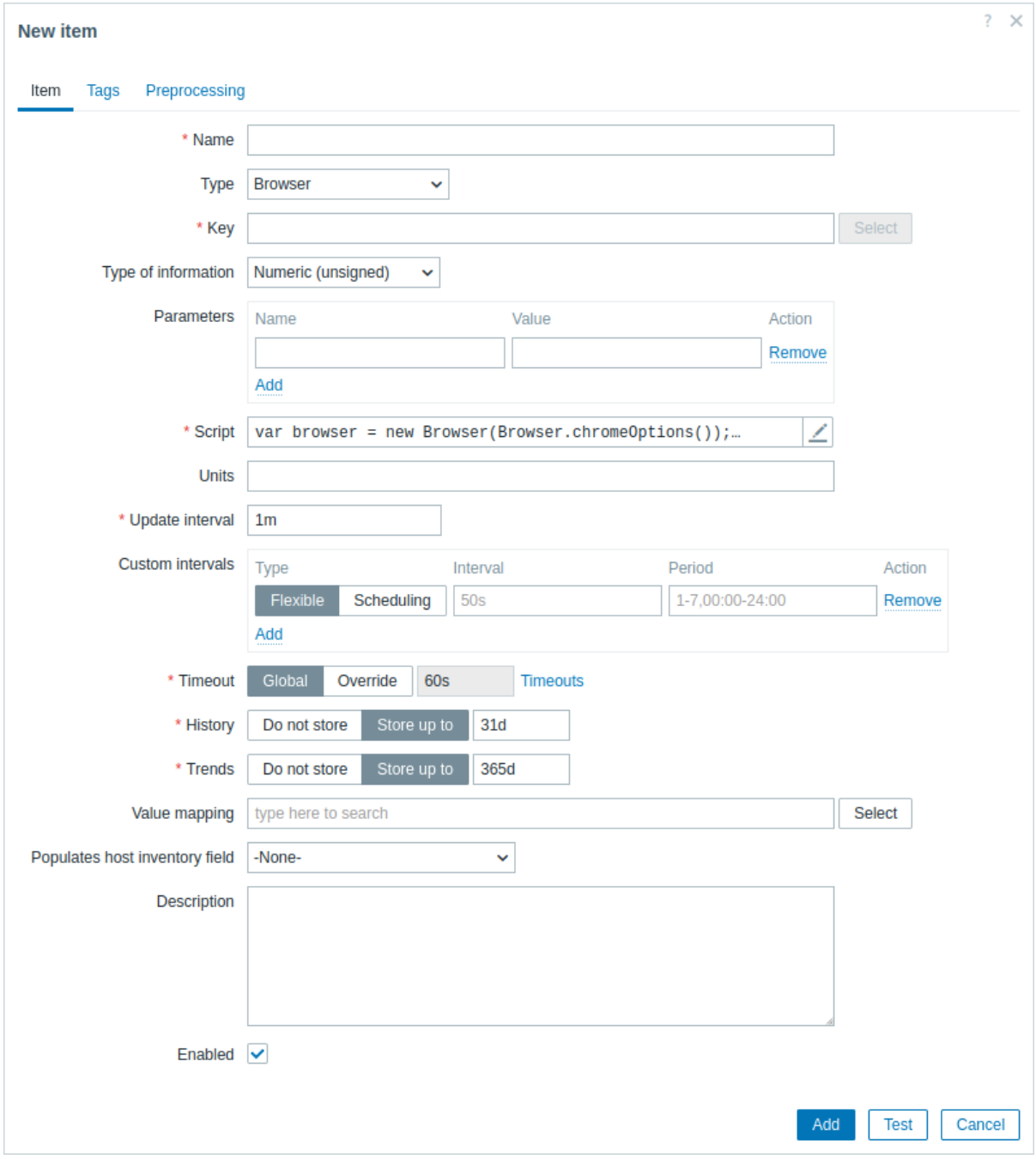
> **Attention:**
> O item implementa parcialmente o padrão W3C WebDriver com o Selenium Server ou um WebDriver simples (por exemplo, ChromeDriver) como endpoint de teste web. Para que o item funcione, defina o parâmetro `WebDriverURL` no arquivo de configuração do Zabbix server/proxy (se estiver usando ChromeDriver, consulte Considerações de segurança). Para melhor desempenho, considere usar um server dedicado para o ambiente de teste web.

As verificações de itens Browser são executadas e processadas pelos browser pollers do Zabbix server ou proxy. Se necessário, você pode ajustar o número de instâncias pré-forked de browser pollers no parâmetro `StartBrowserPollers` do arquivo de configuração do Zabbix server/proxy.

Para monitorar sites e aplicações web complexos, o template Website by Browser está disponível como um template pronto para uso.

Configuração

No campo *Tipo* do formulário de configuração do item, selecione Navegador e preencha os campos obrigatórios.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens do tipo Navegador são:

Campo	Descrição
Chave	Insira uma chave exclusiva que será usada para identificar o item.
Parâmetros	Especifique as variáveis a serem passadas para o script como pares de atributo e valor. Macros de usuário são suportadas. Para ver quais macros internas são suportadas, faça uma busca por "item do tipo Navegador" na tabela de macros suportadas.
Script	Insira o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. Este código deve fornecer a lógica para retornar o valor da métrica. O código tem acesso a todos os parâmetros, 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 Monitor websites with Browser items.

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);
}
```

Consulta de telemetria

Este tipo de item permite recuperar ou agregar dados de telemetria. Os dados de telemetria são usados para o monitoramento do desempenho de aplicações (APM), ajudando você a visualizar como as solicitações percorrem suas aplicações e serviços, medir o desempenho deles e encontrar a causa de operações lentas ou com falha.

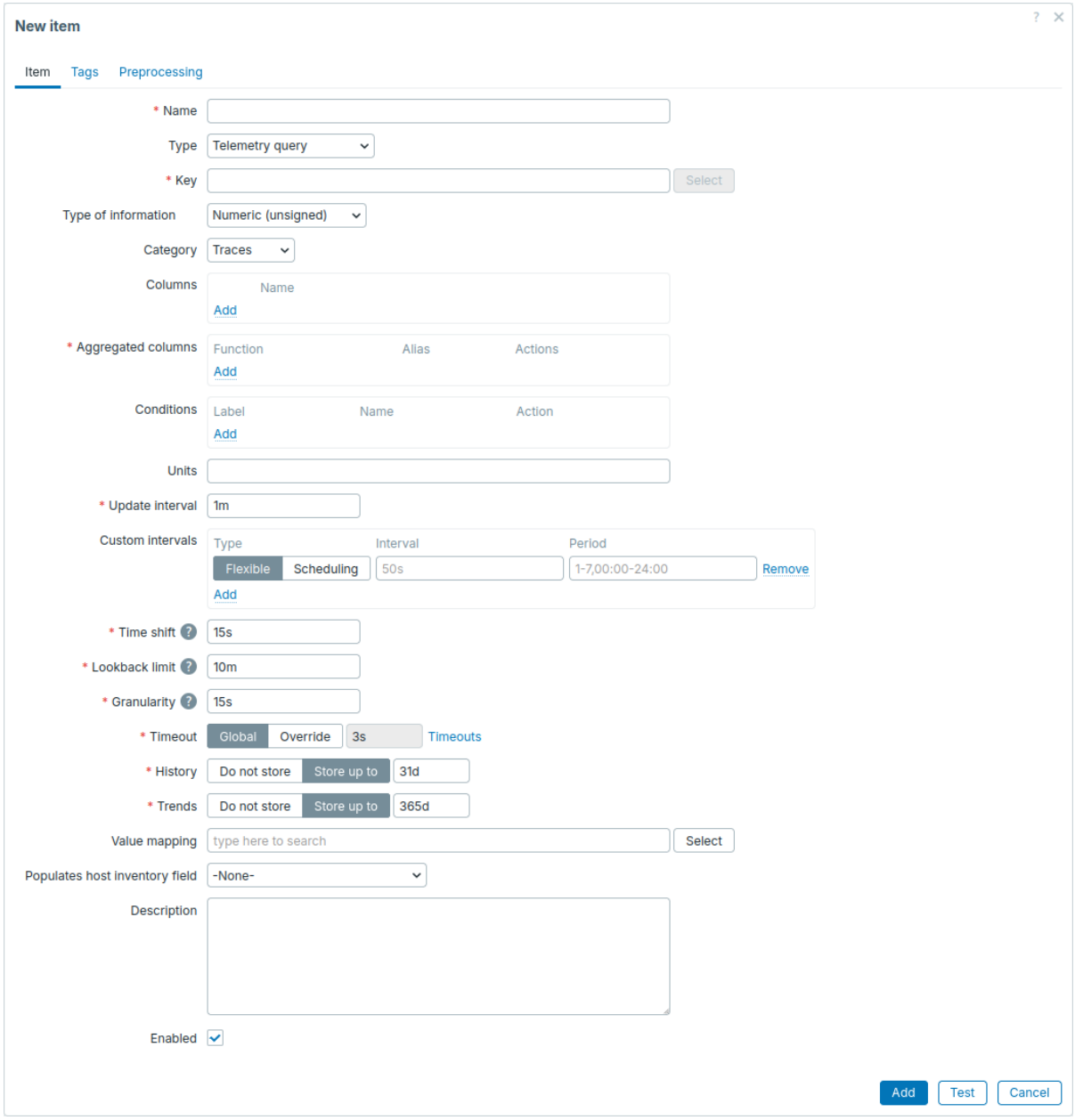
Os dados de telemetria são divididos em três tipos: traces (o caminho percorrido por uma solicitação através de suas aplicações e serviços), métricas (medições, como o tempo de resposta) e logs (registros de texto de eventos). Ao usar o item de consulta de telemetria, você pode, por exemplo, encontrar todos os traces nos quais um serviço retornou um erro, obter o tempo médio de resposta a partir das suas métricas ou pesquisar nos seus logs uma mensagem de erro específica.

Ao configurar um item de consulta de telemetria, você define um conjunto de instruções (uma consulta) que informa ao Zabbix quais dados de telemetria solicitar, como filtrá-los e como agregá-los. Quando a verificação do item é executada, o Zabbix envia essa consulta à fonte de dados de telemetria e recupera o resultado com base na consulta.

Para usar este item, especifique a fonte de dados de telemetria e defina o parâmetro `StartTelemetryQueryPollers` na configuração do Zabbix server ou do Zabbix proxy, dependendo de qual deles monitora o host do item.

Configuração

Ao criar um item, defina seu tipo como *Telemetry query*:



Parâmetro	Descrição
Name	Consulte os parâmetros comuns a todos os itens.
Type	Selecione *Telemetry query*.
Key	Consulte os parâmetros comuns a todos os itens.
Type of information	
Category	Selecione o tipo de signal que deseja consultar: *Traces*, *Metrics* ou *Logs*.
Metric points	Selecione o tipo de metric point que deseja consultar: *Sum*, *Gauge*, *Histogram* ou *Exponential histogram*. Este parâmetro está disponível quando *Category* está definido como *Metrics*.
Columns	Adicione as colunas de dados de telemetria que deseja consultar. Para colunas que contêm pares chave-valor, informe também o nome da chave. Os registros dessas colunas serão incluídos no valor do item como estão.

Parâmetro	Descrição
Aggregated columns	Adicione as colunas de dados de telemetria que deseja consultar e como calcular um valor a partir de seus registros: **Function** - selecione a função a ser aplicada aos registros da coluna: *min*, *max*, *avg*, *count*, *sum* ou *percentile*. **Percentage** - informe o percentil a ser calculado a partir dos registros da coluna (se *Function* estiver definido como *percentile*). **Column** - selecione a coluna a partir da qual o cálculo será feito (se *Function* não estiver definido como *count*). **Alias** - informe um nome exclusivo para o valor calculado, usado para identificá-lo no valor do item.
Type of calculation	Selecione como avaliar as condições definidas no parâmetro *Conditions* abaixo: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo *Type* e *Attribute key* são combinadas com *Or*. **And** - todas as condições devem ser atendidas. **Or** - pelo menos uma condição deve ser atendida. **Custom expression** - as condições devem ser atendidas conforme especificado na fórmula, por exemplo, `A and (B or C)`. Este parâmetro está disponível quando você define duas ou mais condições.
Conditions	Adicione condições para filtrar registros de colunas de dados de telemetria: **Type** - selecione a coluna de dados de telemetria à qual a condição será aplicada. **Attribute key** - informe a chave à qual a condição será aplicada (se *Type* estiver definido como uma coluna com pares chave-valor). **Operator** - selecione a comparação a ser usada (consulte abaixo) entre o valor do registro e *Attribute value*. **Attribute value** - informe o valor com o qual o valor do registro será comparado (se *Operator* não estiver definido como *Exists*). Se *Type* estiver definido como uma coluna de valor único, a condição será aplicada ao registro da coluna; se a coluna contiver uma matriz de valores, a condição corresponderá ao registro se pelo menos um valor da matriz atender à condição. Se *Type* estiver definido como uma coluna que contém pares chave-valor, a condição será aplicada ao valor da chave especificada em *Attribute key*; se a coluna contiver uma matriz de pares chave-valor, a condição corresponderá ao registro se pelo menos um par com o *Attribute key* especificado atender à condição. Para colunas de valor único, você pode usar estes operadores: **Equals** - corresponde a registros iguais a *Attribute value* (diferencia maiúsculas de minúsculas). **Does not equal** - corresponde a registros diferentes de *Attribute value* (diferencia maiúsculas de minúsculas). **Contains** - corresponde a registros que contêm *Attribute value* (diferencia maiúsculas de minúsculas). **Does not contain** - corresponde a registros que não contêm *Attribute value* (diferencia maiúsculas de minúsculas). Para colunas que contêm pares chave-valor, você pode usar estes operadores: **Equals** - corresponde a registros nos quais *Attribute key* existe e seu valor é igual a *Attribute value* (diferencia maiúsculas de minúsculas). **Does not equal** - corresponde a registros nos quais *Attribute key* existe e seu valor é diferente de *Attribute value* (diferencia maiúsculas de minúsculas). **Exists** - corresponde a registros nos quais *Attribute key* existe (diferencia maiúsculas de minúsculas).
Units *Update interval* *Custom intervals*	Consulte os parâmetros comuns a todos os itens.

Parâmetro	Descrição
Time shift	Informe quanto tempo para trás, antes do horário atual, deve terminar o intervalo de tempo consultado pelo item. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou macros de usuário. Cada consulta solicita dados para um intervalo de tempo. Normalmente, o intervalo começa quando terminou o intervalo de tempo da consulta anterior e termina no ponto definido por *Time shift*. Se o intervalo de tempo da consulta anterior terminou depois do horário atual menos *Granularity*, o intervalo de tempo da consulta atual começa no horário atual menos *Granularity*, sobrepondo-se assim ao intervalo de tempo da consulta anterior. Se não houver uma consulta anterior a partir da qual continuar, o intervalo começa no ponto definido por *Lookback limit*. Se o host deste item passar a ser monitorado por uma entidade diferente (por exemplo, um proxy diferente de um proxy group), o item poderá não saber mais quando a consulta anterior terminou. Isso pode causar valores duplicados.
Lookback limit	Informe quanto tempo para trás, antes do horário atual, o item deve consultar dados quando não houver uma consulta anterior a partir da qual continuar (por exemplo, na primeira execução do item) ou quando a consulta anterior for mais antiga que *Update interval* (por exemplo, porque o Zabbix server estava indisponível). Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou macros de usuário. Se a fonte de dados de telemetria contiver uma grande quantidade de dados (por exemplo, mais de um milhão de registros por hora), um *Lookback limit* alto poderá reduzir significativamente o desempenho quando o item o utilizar.
Granularity	Informe a duração de cada parte em que o intervalo de tempo completo da consulta será dividido. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou macros de usuário.
Timeout *History* *Trends* *Value mapping* *Populates host inventory field* *Description* *Enabled*	Consulte os parâmetros comuns a todos os itens.

Colunas de dados de telemetria compatíveis

Esta seção lista as colunas compatíveis com estes parâmetros do item: *Columns*, *Aggregated columns* e *Conditions*. Essas são as colunas usadas pelo ClickHouse para armazenar dados de telemetria, seguindo o esquema do exporter do ClickHouse do OpenTelemetry.

Selecione a *Category*:

- Traces
- Metrics: Sum
- Metrics: Gauge
- Metrics: Histogram
- Metrics: Exponential histogram
- Logs

Traces

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
Duração	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Timestamp	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Métricas: Soma

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Métricas: Gauge

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
Atributos	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Métricas: Histograma

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Métricas: histograma exponencial

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Logs

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓

Coluna	*Colunas*	*Colunas agregadas*	*Condições*
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

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 as tarefas do housekeeper

Quaisquer dados mais antigos serão removidos pelo housekeeper.

A recomendação geral é manter o histórico pelo menor número possível de dias, para não sobrecarregar o banco de dados com muitos valores históricos.

Em vez de manter um histórico longo, você pode manter os 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 os dados de histórico em comparação com os dados de tendências consultando a página de dimensionamento do banco de dados.

Mesmo mantendo 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 estiver definido como '0', o item atualizará apenas itens dependentes e o inventário. Nenhuma função de trigger será avaliada, porque a avaliação de trigger se baseia apenas em dados de histórico.

Note:
Como forma alternativa de preservar o histórico, considere usar a funcionalidade de exportação de histórico dos 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. Qualquer dado mais antigo será removido pelo housekeeper.

O servidor Zabbix acumula dados de tendências em tempo de execução no cache de tendências, à medida que os dados chegam. O servidor grava as tendências da **hora anterior** de cada item no banco de dados (onde o frontend pode encontrá-las) nas seguintes situações:

- o servidor recebe o primeiro valor da hora atual do item
- faltam 5 minutos ou menos para o final da hora atual e ainda não há valores da hora atual do item
- o servidor é parado

Para ver as tendências em um gráfico, você precisa esperar pelo menos até o início da próxima hora (se o item for atualizado com frequência) e no máximo até o final da próxima hora (se o item for atualizado raramente), o que é no máximo 2 horas.

Quando o servidor grava o cache de tendências e já existem tendências no banco de dados para essa hora (por exemplo, o servidor foi reiniciado no meio da hora), o servidor precisa usar instruções de atualização em vez de simples inserções. Portanto, em uma instalação maior, se for necessário reiniciar, é desejável parar o servidor no final de uma hora e iniciar no início da próxima hora para evitar sobreposição de dados de tendências.

As tabelas de histórico não participam da geração de tendências de forma alguma.

Attention:
Se as tendências forem definidas como '0', o servidor Zabbix não calcula nem armazena tendências.

Note:
As tendências são calculadas e armazenadas com o mesmo tipo de dado dos valores originais. Como resultado, os cálculos de valor médio de valores do tipo de dado sem sinal são arredondados e quanto menor o intervalo de valores, menos preciso será o resultado. Por exemplo, se o item tiver valores 0 e 1, o valor médio será 0, não 0,5.
Além disso, reiniciar o servidor pode resultar em perda de precisão nos cálculos do valor médio do tipo de dado sem sinal para a hora atual.

Parâmetros do usuário

Visão geral

Às vezes, você pode querer executar uma verificação do agent que não vem predefinida com o Zabbix. É aí que os parâmetros de usuário ajudam.

Você pode escrever um comando que recupere os dados de que precisa 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=<key>,<command>
```

Como você pode ver, um parâmetro de usuário também contém uma key. A key será necessária ao configurar um item. Informe uma key de sua escolha que seja fácil de referenciar (ela 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
```

Então, ao configurar um item, informe a key para referenciar o comando do parâmetro de usuário que você deseja executar.

Parâmetros de usuário são comandos executados pelo Zabbix agent. Observe que até 16 MB 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 key. Dessa 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
```

Parameter	Description
Key	Key exclusivo do item. O [*] define que essa key aceita parâmetros dentro dos colchetes. Os parâmetros são informados ao configurar o item.
Command	Comando a ser executado para avaliar o valor da key. *Somente para parâmetros de usuário flexíveis*: Você pode usar referências posicionais $1...$9 no comando para se referir ao respectivo parâmetro na key do item. O Zabbix analisa os parâmetros entre [ ] da key 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. As referências posicionais são interpretadas independentemente de estarem entre aspas duplas (") ou aspas simples (''). Para usar referências posicionais sem alteração, especifique um cifrão duplo - por exemplo, awk '{print $$2}'. Nesse caso, $$2 realmente se tornará $2 ao executar o comando.

Attention:
As referências posicionais com o sinal $ são procuradas e substituídas pelo Zabbix agent somente para parâmetros de usuário flexíveis. Para parâmetros de usuário simples, esse processamento de referência é ignorado e, portanto, não é necessário nenhum escape do sinal $.

Attention:
Certos caracteres não são permitidos em parâmetros de usuário por padrão. Para a lista completa de caracteres, consulte UnsafeUserParameters.

Exemplo 1

Algo muito simples:

```
UserParameter=ping[*],echo $1
```

Podemos definir um número ilimitado de 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 de texto) não ficará não suportado em caso de uma saída de erro padrão.

O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam.

Parâmetros de usuário que retornam texto (caractere, log ou tipo de informação de texto) também podem retornar um espaço em branco. Em caso de resultado inválido, o item ficará não suportado.

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 user parameter.

Etapa 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 server MySQL:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Quando executado, o comando retorna o número total de consultas SQL.

Etapa 2

Adicione o comando ao arquivo de configuração do agent do Zabbix (`zabbix_agentd.conf`):

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions é um identificador exclusivo. Ele pode ser qualquer identificador de chave válido, por exemplo, *queries*.

Teste este parâmetro usando o agent do Zabbix com a flag `-t` (se estiver sendo executado como root, observe, no entanto, que o agent pode ter permissões diferentes quando iniciado como daemon):

```
zabbix_agentd -t mysql.questions
```

Etapa 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 de usar o comando de controle em tempo de execução.

Teste o parâmetro usando o utilitário `zabbix_get`.

Etapa 4

Adicione um novo item com `Key=mysql.questions` ao host monitorado. O tipo do item deve ser *Zabbix Agent* ou *Zabbix Agent (active)*.

Observe que o tipo dos valores retornados deve ser configurado corretamente no server Zabbix. Caso contrário, o Zabbix não os aceitará.

Contadores de desempenho do Windows

Visão geral

Você pode monitorar com eficiência 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 obter mais informações sobre o uso dessa chave ou de sua equivalente somente em inglês, `perf_counter_en`, consulte Chaves de item específicas do Windows.

Para obter uma lista completa dos contadores de desempenho disponíveis para monitoramento, execute:

```
typeperf -qx
```

Você também pode usar a descoberta de baixo nível para descobrir várias instâncias de objetos dos contadores de desempenho do Windows e automatizar a criação de itens `perf_counter` para vários objetos de instância.

Representação numérica

O Windows mantém representações numéricas (índices) para nomes de objetos e contadores de desempenho. O Zabbix oferece suporte a 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 recomendável usá-las, a menos que você possa garantir que seus índices numéricos correspondam às strings corretas em hosts específicos. Se precisar criar itens portáteis que funcionem em diferentes hosts com várias versões localizadas do Windows, você poderá usar a chave `perf_counter_en` ou o parâmetro de configuração `PerfCounterEn`, que permitem usar nomes em inglês independentemente da localidade do sistema.

Para descobrir os equivalentes numéricos, execute `regedit` e localize *Counter* em `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

A entrada do registro contém informações como estas:

```
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 a cada parte de string do contador de desempenho, como em `\System\% Processor Time`:

```
System → 2
% Processor Time → 6
```

Em seguida, você pode usar esses números para representar o caminho numericamente:

```
\2\6
```

Parâmetros de contadores de desempenho

Você pode configurar alguns parâmetros PerfCounter para o monitoramento dos contadores de desempenho do Windows.

Por exemplo, você pode adicioná-los ao arquivo de configuração do Zabbix agent:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
ou
PerfCounter=UserPerfCounter2,"\4\24",30
```

Com esses parâmetros configurados, você pode simplesmente usar `UserPerfCounter1` ou `UserPerfCounter2` como as chaves para criar os respectivos itens.

Lembre-se de reiniciar o Zabbix agent após fazer alterações no arquivo de configuração.

Atualização em massa

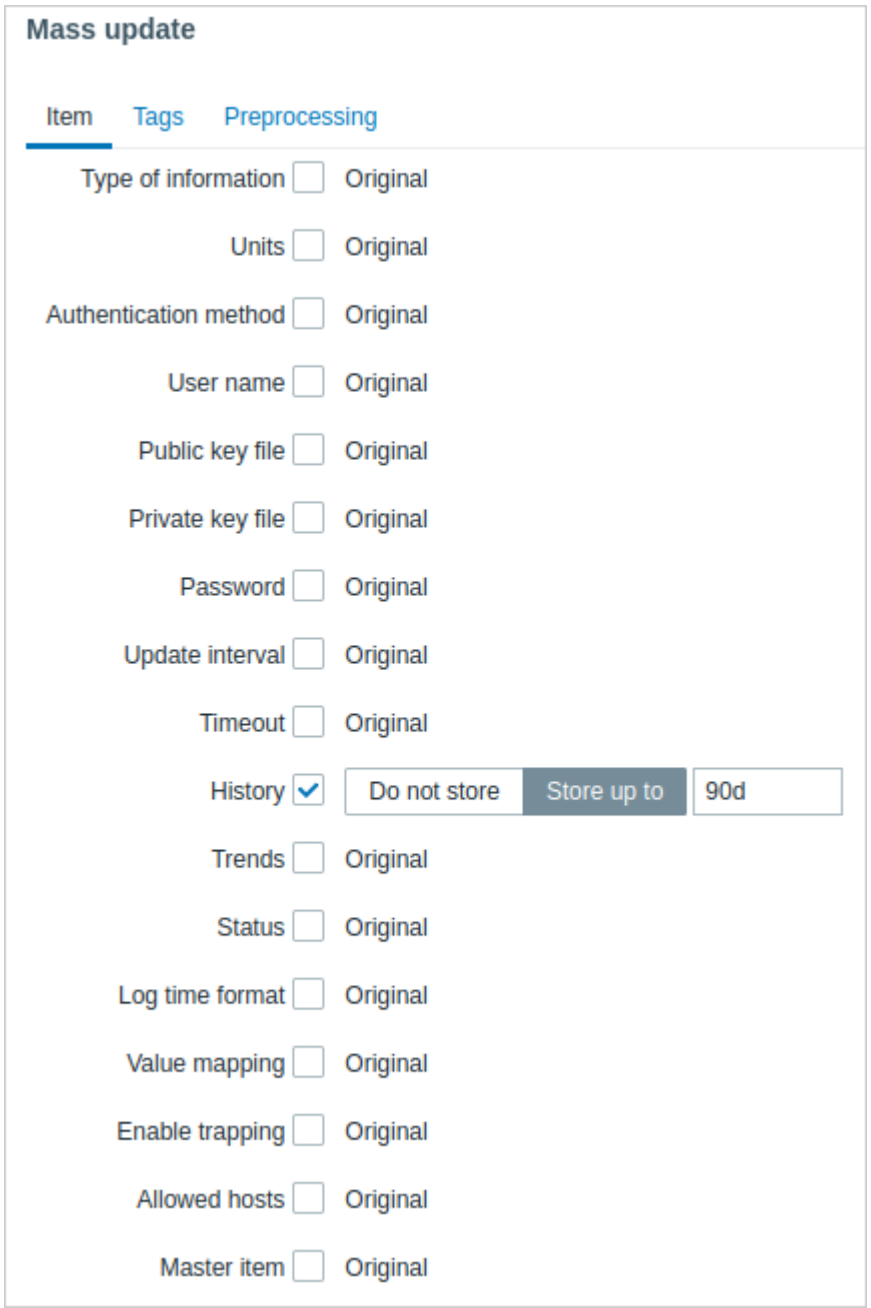
Visão geral

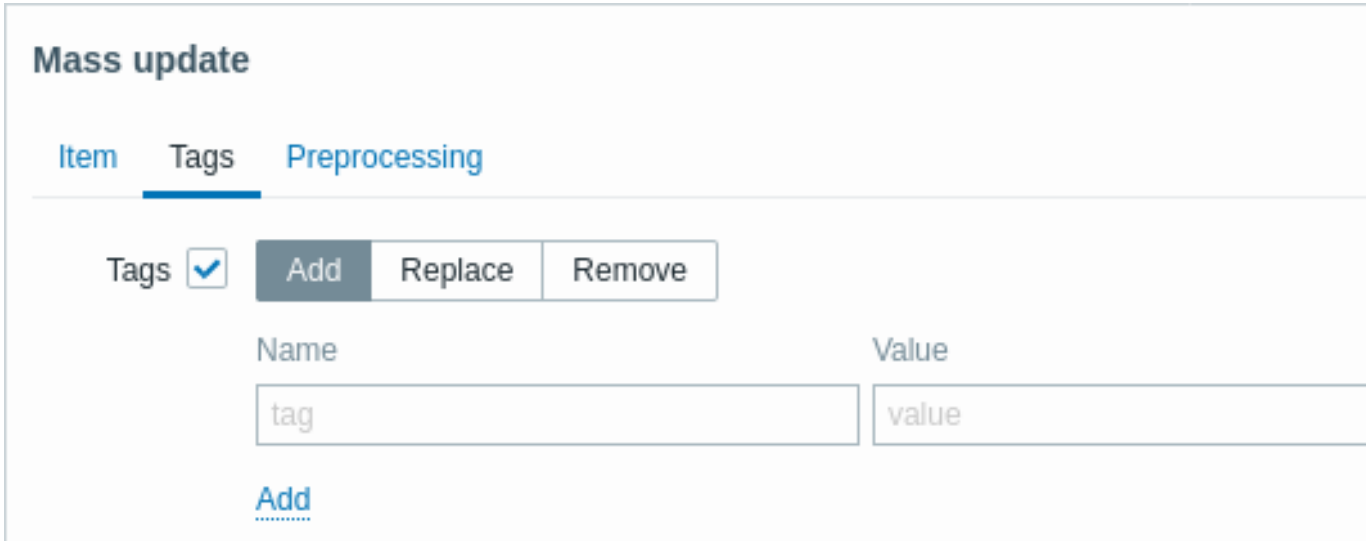
Às vezes, você pode querer alterar algum atributo para vários items de uma vez. Em vez de abrir cada item individualmente para edição, você pode usar a função de atualização em massa para isso.

Usando a atualização em massa

Para atualizar vários itens em massa, faça o seguinte:

- Marque as caixas de seleção dos itens que deseja atualizar na lista
- Clique em *Atualização 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 que deseja atualizar
- Insira novos valores para os atributos

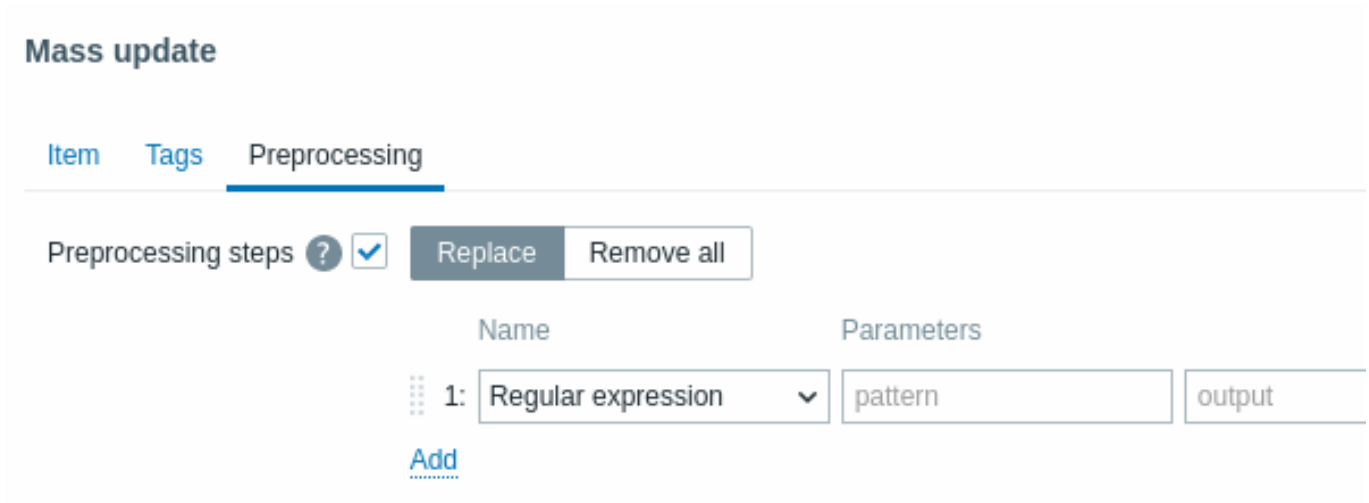




A opção *Tags* permite:

- *Adicionar* - adicionar as tags especificadas aos itens (tags com o mesmo nome, mas com 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 itens

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



Marque a caixa de seleção ao lado de *Etapas de pré-processamento* para ativar a atualização em massa das etapas de pré-processamento.

A atualização em massa de *Pré-processamento* permite:

- *Substituir* - substituir as etapas de pré-processamento existentes nos itens pelas etapas de pré-processamento especificadas abaixo
- *Remover tudo* - remover todas as etapas de pré-processamento dos itens

Quando terminar, clique em *Atualizar*.

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*.

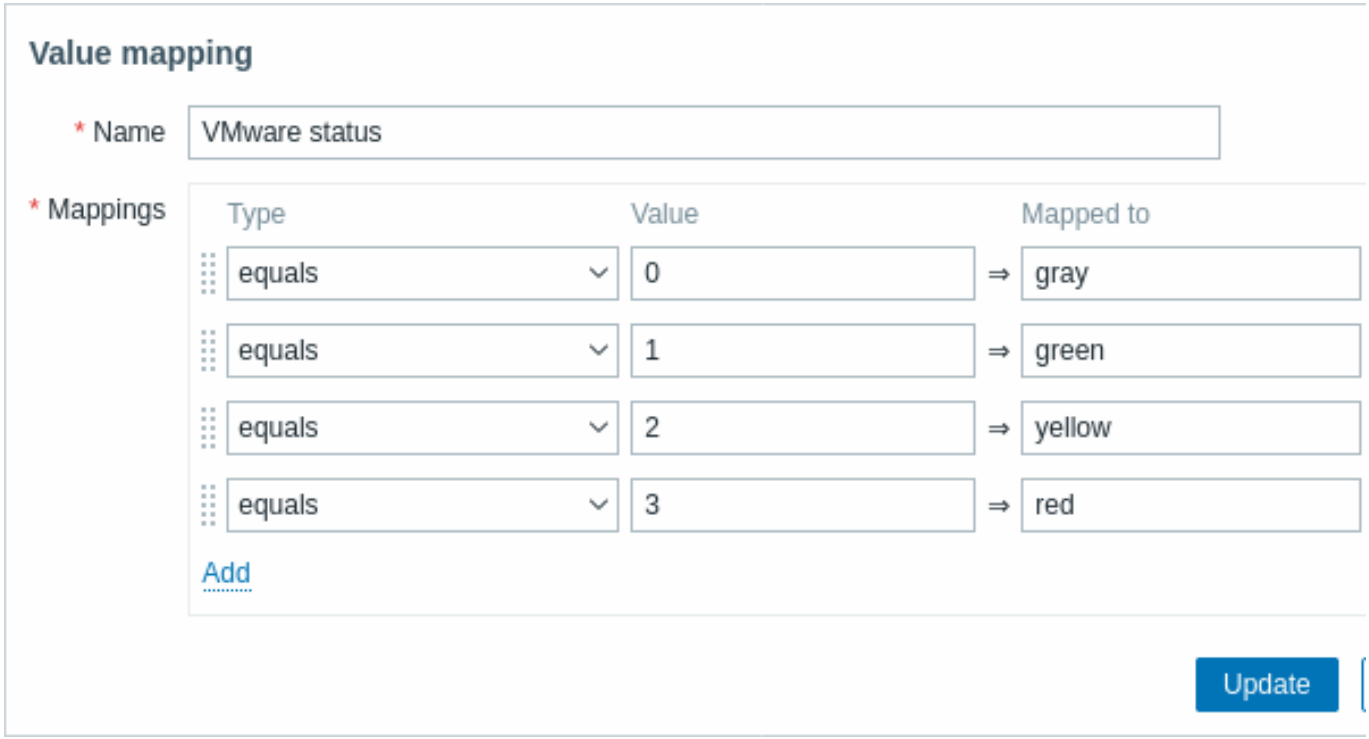
Os mapeamentos de valores são exportados/importados com o respectivo template ou host. Eles também podem ser atualizados em massa usando os formulários de atualização em massa de host e template.

Configuração

Para configurar um mapeamento de valores, siga estas etapas:

1. Abra o formulário de configuração do host ou do template.

2. Na aba *Value mapping*, clique em *Add* para adicionar um novo mapeamento de valores ou clique no nome de um mapeamento existente para editá-lo.



Parâmetros de um mapeamento de valores:

Parameter	Description
Name	Nome exclusivo para o conjunto de mapeamentos de valores.
Mappings	Regras individuais para mapear valores numéricos/de string para representações de string. O mapeamento é aplicado na ordem das regras, que pode ser alterada arrastando-as.
Type	Tipo de mapeamento: **equals** - valores iguais serão mapeados; **is greater than or equals** - valores iguais ou maiores serão mapeados; **is less than or equals** - valores iguais ou menores serão mapeados; **in range** - valores dentro do intervalo serão mapeados; o intervalo é expresso como <number1>-<number2> ou <number>; vários intervalos são compatíveis (por exemplo, 1-10,101-110,201); **regexp** - valores correspondentes à expressão regular serão mapeados (expressões regulares globais não são compatíveis); **default** - todos os valores restantes serão mapeados, exceto aqueles com mapeamentos específicos. Para mapear intervalos, somente tipos de valores numéricos (*is greater than or equals*, *is less than or equals*, *in range*) são compatíveis.
Value	Valor recebido (pode conter um intervalo ou uma expressão regular, dependendo do tipo de mapeamento).
Mapped to	Representação em string (de até 64 caracteres) para o valor recebido.

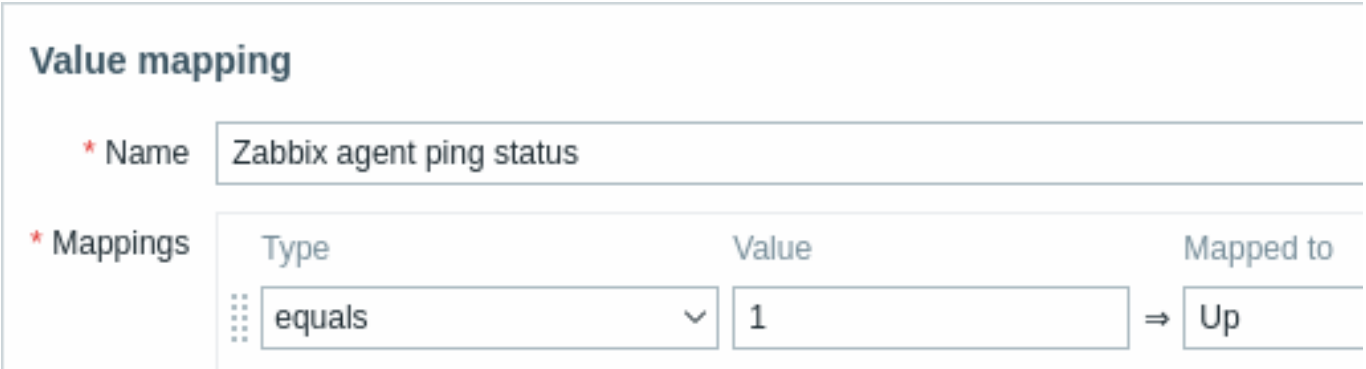
Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Ao visualizar o mapeamento de valores na lista, somente os três primeiros mapeamentos ficam visíveis; três pontos indicam que existem mais mapeamentos.



Exemplo de mapeamento de valores

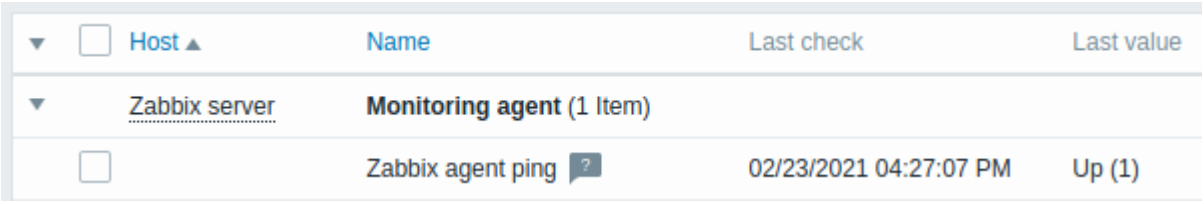
Um dos itens de agent predefinidos, *Zabbix agent ping*, usa um mapeamento de valores no nível do template, "Zabbix agent ping status", para exibir seus valores.



No formulário de configuração do item, você pode encontrar uma referência a esse mapeamento de valores no campo *Value mapping*:

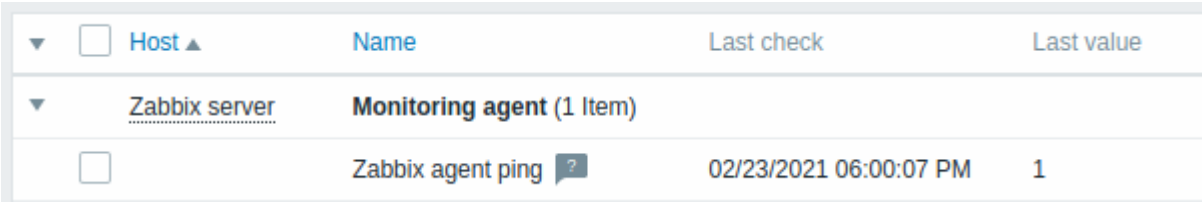


Esse mapeamento é usado na seção *Monitoring* > *Latest data* para exibir "Up" (com o valor bruto entre parênteses).



Note:
Na seção *Latest data*, os valores exibidos são reduzidos a 20 caracteres. Se o mapeamento de valores for usado, essa redução não será aplicada ao valor mapeado, mas somente ao valor bruto (exibido entre parênteses).

Sem um mapeamento de valores predefinido, você veria apenas "1", o que pode ser difícil de entender.



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

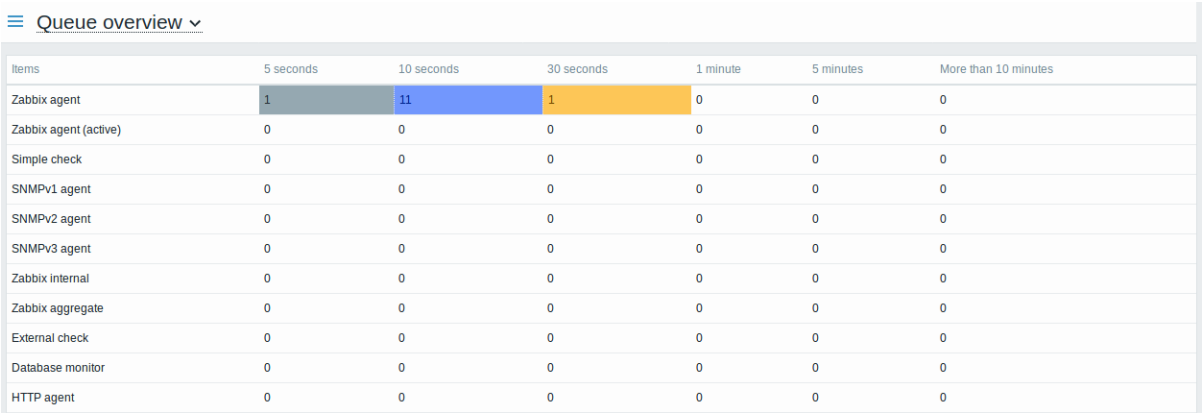
As estatísticas mostradas pela fila são um bom indicador do desempenho do server Zabbix.

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.

Leitura da fila

Para ler a fila, acesse *Administration* > *Queue* > *Queue overview*.

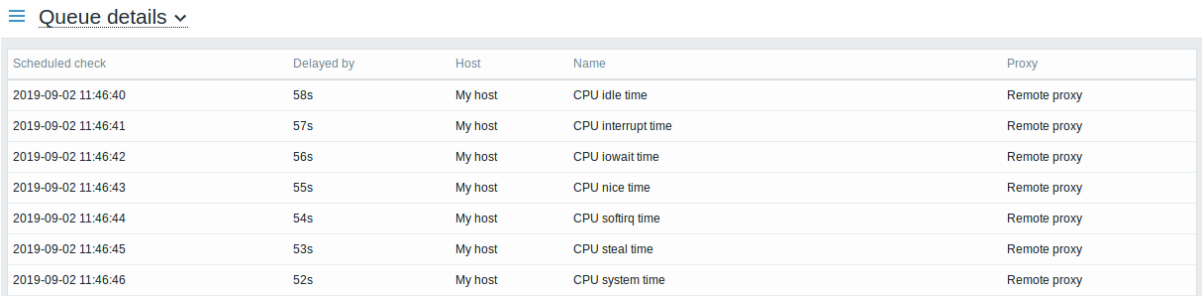

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

A imagem aqui está geralmente ”ok”, portanto podemos presumir que o server está funcionando corretamente.

A fila mostra alguns itens aguardando há até 30 segundos. Seria ótimo saber quais são esses itens.

Para fazer isso, selecione *Queue details* no menu suspenso do título. Agora você pode ver uma lista desses itens atrasados.

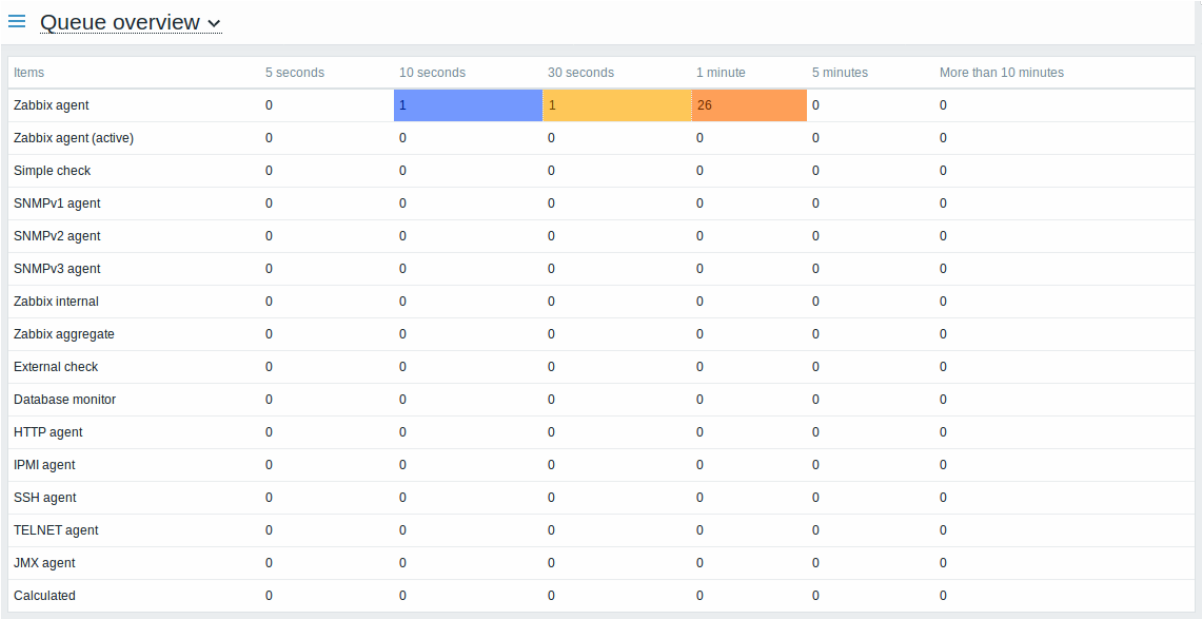

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

Com esses detalhes, talvez seja possível descobrir por que esses itens podem estar atrasados.

Com um ou dois itens atrasados, talvez não haja motivo para preocupação. Eles podem ser atualizados em um segundo. No entanto, se você perceber que vários itens estão atrasando por muito tempo, pode haver um problema mais sério.

Veja também: Alinhamento de fusos horários ao usar intervalos de agendamento.



Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

Item de fila

Um item interno especial `zabbix[queue,<from>,<to>]` pode ser usado para monitorar a integridade da fila no Zabbix. Ele retornará o número de itens atrasados pelo período definido. Para obter mais informações, consulte Itens internos.

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` pode ser especificado na configuração do server do Zabbix.

Dois itens internos são compatíveis com o monitoramento do cache de valores: `zabbix[vcache,buffer,<mode>]` e `zabbix[vcache,cache,<parameter>]`. Consulte mais detalhes em itens internos.

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

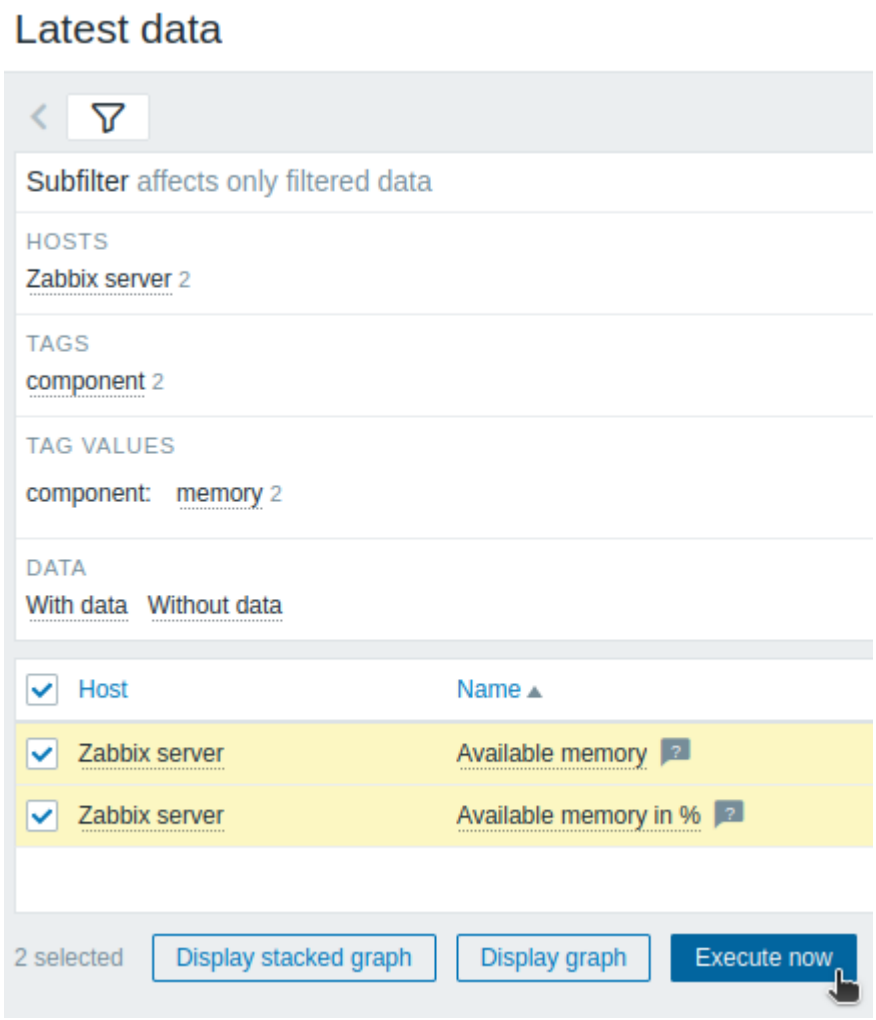
Você pode executar verificações passivas a partir de:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items* ou *Discovery*
- O menu de item
- Ao editar itens 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 itens ou regras LLD que estão sendo criados ou que foram criados recentemente, você pode usar a opção Test em vez disso.

Por exemplo, para executar uma verificação passiva em *Monitoring* > *Latest data*:

1. Selecione itens; você pode executar a verificação para vários itens ao mesmo tempo.
2. Clique em *Execute now*.



Note:
Em *Latest data*, os usuários podem *Execute now* em hosts para os quais tenham permissão de read-write, ou permissão de read e a ação *Invoke "Execute now" on read-only hosts* action habilitada para sua função.

Restringindo verificações do agent

Visão geral

Você pode controlar quais chaves de item o Zabbix agent ou agent 2 tem permissão ou é impedido de usar ao executar verificações de item, comandos remotos ou scripts.

Para isso, use estes parâmetros de configuração do agent para definir regras de □□□□□□/negação:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`
- `AllowKeyRegexp=<pattern>`
- `DenyKeyRegexp=<pattern>`

O <pattern> deve conter uma única chave de item.
 Para `AllowKey` e `DenyKey`, o <pattern> suporta curingas (*). O curinga corresponde a qualquer número de quaisquer caracteres na sua posição e pode ser usado para corresponder a chaves de item ou parâmetros (por exemplo, `vfs.file.*[*]`).
 Para `AllowKeyRegexp` e `DenyKeyRegexp`, o <pattern> suporta expressões regulares. O padrão de expressão regular pode ser usado para corresponder a chaves de item e suas strings de parâmetros. Veja também exemplos de escape correto.
 Observe que o Zabbix agent e o Zabbix agent 2 podem oferecer suporte a sintaxes diferentes de expressões regulares.

Attention:
Para melhorar a segurança, é recomendável usar chaves de item exatas em vez de curingas para `AllowKey` e `DenyKey`. Também é recomendável usar os padrões de expressão regular mais específicos possíveis para `AllowKeyRegexp` e `DenyKeyRegexp`. Para detalhes, consulte Protegendo regras de permissão/negação.

Ao contrário de outros parâmetros de configuração do agent, você pode especificar um número ilimitado de parâmetros AllowKey, DenyKey, AllowKeyRegexp ou DenyKeyRegexp.

Notas importantes

- Todos os itens system.run são desativados por padrão (mesmo quando DenyKey e DenyKeyRegexp estão vazios), como se DenyKey=system.run[*] ou DenyKeyRegexp=^system\.run\[.*\]$ tivesse sido definido como a última regra. Por isso, você pode permitir itens específicos system.run sem negar explicitamente outros itens system.run.
- Se possível, use AllowKey/AllowKeyRegexp para permitir apenas os itens necessários e negar todo o restante. Algumas chaves podem ser exploradas para ler arquivos não intencionais por meio de path traversal (por exemplo, vfs.file.contents["../../../../etc/passwd"]), e novas versões do Zabbix agent podem introduzir chaves não cobertas pelas suas regras DenyKey/DenyKeyRegexp.
- A configuração de AllowKey, DenyKey, AllowKeyRegexp e DenyKeyRegexp não afeta os parâmetros do agent HostnameItem, HostMetadataItem ou HostInterfaceItem.
- Itens negados tornam-se não suportados sem qualquer indicação ou mensagem de erro; por exemplo:
  - O parâmetro de linha de comando --print (-p) do Zabbix agent não exibirá as chaves de item negadas.
  - O parâmetro de linha de comando --test (-t) do Zabbix agent retornará "Unsupported item key." para chaves de item negadas.
  - Quando o registro em log está ativado (LogRemoteCommands=1), o arquivo de log do Zabbix agent não registrará os comandos remotos negados.

Ordem das regras de □□□□□□/negação

Você pode especificar um número ilimitado de regras AllowKey, DenyKey, AllowKeyRegexp ou DenyKeyRegexp, embora a ordem delas seja importante.

- As regras são avaliadas uma a uma, de cima para baixo.
- Quando uma item key corresponde a uma regra, ela é permitida ou negada, e a avaliação das regras é interrompida.

Por exemplo, ao avaliar vfs.file.contents[/etc/passwd], as regras são processadas da seguinte forma:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # O padrão da item key não corresponde; o agent prossegue para
AllowKey=vfs.file.contents[/etc/passwd]     # O padrão da item key corresponde; o agent permite a verifica
DenyKey=vfs.file.*[*]                       # O agent ignora a regra, pois a avaliação já foi interrompida
```

A seguinte ordem de regras negará a verificação do item:

```
DenyKey=vfs.file.*[*]                       # O padrão da item key corresponde; o agent nega a verificação
AllowKey=vfs.file.contents[/etc/passwd]     # O agent ignora a regra, pois a avaliação já foi interrompida
AllowKey=vfs.file.contents[/tmp/app.log]    # O agent ignora a regra, pois a avaliação já foi interrompida
```

Exemplos

Os exemplos a seguir mostram padrões comuns de configuração para AllowKey, DenyKey, AllowKeyRegexp e DenyKeyRegexp.

Permitindo verificações e comandos específicos

Permita apenas duas verificações de item vfs.file e dois comandos system.run:

usando AllowKey e DenyKey:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

usando AllowKeyRegexp e DenyKeyRegexp:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Definir DenyKey=system.run[*] ou DenyKeyRegexp=^system\.run\[.*\]$ é desnecessário, porque todos os outros comandos system.run são negados por padrão.

Permitindo scripts

Permita que o Zabbix agent execute scripts em hosts por meio de todos os métodos disponíveis:

- Scripts globais que podem ser executados no frontend ou via API (este método sempre usa a chave `system.run[myscript.sh]`)
- Comandos remotos de operações de ação (este método sempre usa a chave `system.run[myscript.sh,nowait]`)
- Itens `system.run` do Zabbix agent com o script, por exemplo:
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=system.run[myscript.sh,*]
```

ou

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Para controlar o parâmetro `wait/nowait`, você deve definir uma regra diferente. Por exemplo, você pode permitir apenas itens `system.run[myscript.sh,wait]`, excluindo assim outros métodos:

```
AllowKey=system.run[myscript.sh,wait]
```

ou

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Protegendo regras allow/deny

Estes exemplos mostram como proteger regras `AllowKey/AllowKeyRegexp` ou `DenyKey/DenyKeyRegexp` excessivamente permissivas.

Considere as seguintes regras:

usando `AllowKey` e `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

usando `AllowKeyRegexp` e `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
No Windows, você deve escapar os espaços no caminho usando um acento circunflexo (`^`) para `AllowKey` ou `DenyKey` e `\s` para `AllowKeyRegexp` ou `DenyKeyRegexp`.

Essas regras contêm um curinga (`*`) (ou `.*` para padrões baseados em expressão regular), que pode ser usado indevidamente:

- O script `test.bat` pode ser executado com quaisquer argumentos, inclusive os não intencionais.
- O padrão `vfs.file.*` (`^vfs\.file\..*$`) corresponde a chaves de item com e sem parâmetros; no entanto, todos os itens `vfs.file` exigem parâmetros.
- O padrão `system.cpu.load[*]` (`^system\.cpu\.load\[.*\]$`) corresponde apenas a chaves de item com parâmetros; no entanto, os itens `system.cpu.load` não exigem parâmetros.

Para proteger essas regras, permita explicitamente a execução de `test.bat` apenas com argumentos específicos e negue padrões corretos de chaves de item; por exemplo:

usando `AllowKey` e `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

usando `AllowKeyRegexp` e `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

Você pode testar as regras executando os seguintes comandos, que retornarão ZBX_NOTSUPPORTED.

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

Exemplos de padrões

As tabelas a seguir mostram como os padrões de chave de item são correspondidos:

- Uma chave corresponde ao padrão somente se atender a **todas** as condições na coluna *Matches*.
- Os parâmetros devem estar totalmente delimitados por colchetes (por exemplo, `vfs.file.contents[*` e `vfs.file.contents*utf` são padrões inválidos).
- Os exemplos ilustram apenas a correspondência de padrões; os itens reais exigem parâmetros.

Para `AllowKey` e `DenyKey`:

Pattern	Matches	Examples
`*`	Qualquer chave, com ou sem parâmetros	

`vfs.file.*`	A chave começa com `vfs.file.`
	Com ou sem parâmetros
		Matches:
		`vfs.file.size`
		`vfs.file.contents`
		`vfs.file.contents[]`
		`vfs.file.size[/var/log/app.log]`

`vfs.*.contents`	A chave começa com `vfs.`
	A chave termina com `.contents`
	Sem parâmetros
		Matches:
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Does not match:
		`vfs.contents`
		`vfs.file.contents[]`

`vfs.file.*[*]`	A chave começa com `vfs.file.`
	Parâmetros quaisquer ou vazios
		Matches
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Does not match:
		`vfs.file.get.custom`

`vfs.file.contents`	A chave é `vfs.file.contents`
	Sem parâmetros
		Matches:
		`vfs.file.contents`
		
		Does not match:
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	A chave é `vfs.file.contents[]`
	Parâmetros vazios
		Matches:
		`vfs.file.contents[]`
		
		Does not match:
		`vfs.file.contents`

`vfs.file.contents[*]`	A chave é `vfs.file.contents`
	Parâmetros quaisquer ou vazios
		Matches:
		`vfs.file.contents[/path/to/file]`
		
		Does not match:
		`vfs.file.contents`

`vfs.file.contents[/etc/passwd,*]`	A chave é `vfs.file.contents`
	O primeiro parâmetro é `/etc/passwd`
	Segundo parâmetro qualquer ou vazio
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

Pattern	Matches	Examples
vfs.file.contents[*passwd*]	A chave é vfs.file.contents Parâmetros quaisquer, com pelo menos um contendo passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	A chave é vfs.file.contents O primeiro parâmetro contém passwd Segundo parâmetro qualquer ou vazio	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	A chave é vfs.file.contents O primeiro parâmetro é /etc/passwd O segundo parâmetro é utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Para AllowKeyRegexp e DenyKeyRegexp:

Pattern	Matches	Examples
^.*$	Qualquer chave, com ou sem parâmetros	
^vfs\.file\..*$	A chave começa com vfs.file. Com ou sem parâmetros	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	A chave começa com vfs. A chave termina com .contents Sem parâmetros	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	A chave começa com vfs.file. Parâmetros quaisquer ou vazios	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	A chave é vfs.file.contents Sem parâmetros	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[\]$	A chave é vfs.file.contents[] Parâmetros vazios	Matches: vfs.file.contents[] Does not match: vfs.file.contents
^vfs\.file\.contents\[.*\]$	A chave é vfs.file.contents Parâmetros quaisquer ou vazios	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents

Pattern	Matches	Examples
`^vfs\.file\.contents\[/etc/passwd,.*\]$`	Apenas vfs.file.contents O primeiro parâmetro é /etc/passwd Segundo parâmetro qualquer ou vazio	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
`^vfs\.file\.contents\[.*passwd.*\]$`	Apenas vfs.file.contents Parâmetros quaisquer, com pelo menos um contendo passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
`^vfs\.file\.contents\[.*passwd.*,.*\]$`	Apenas vfs.file.contents O primeiro parâmetro contém passwd Segundo parâmetro qualquer ou vazio	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
`^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$`	Apenas vfs.file.contents O primeiro parâmetro é /etc/passwd O segundo parâmetro é utf8 ou windows-1252	Matches: vfs.file.contents[/etc/passwd,utf8] vfs.file.contents[/etc/passwd,windows-1252] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Detecção de problemas com triggers

A detecção de problemas no Zabbix é baseada em **triggers**.

Um trigger descreve a condição do problema, permitindo que o Zabbix reaja caso o problema ocorra.

O fluxo completo de detecção de problemas é:

1. Certifique-se de ter um item coletando dados.
2. Defina o trigger para este item. Sem um trigger, o Zabbix apenas coleta dados, mas nunca reage a eles.
3. Observe os problemas detectados:
    - Em *Monitoring* > *Problems*
    - No widget *Problems*
4. Receba notificações, execute comandos remotos (se definidos).

Observe que os templates oficiais do Zabbix contêm triggers predefinidos. Veja o guia para aplicar template para monitor Linux.

O que é um ”trigger”?

Um trigger descreve a *condição de problema*, por meio da expressão do trigger.

A condição de problema, quando ocorre na vida real, é quando o trigger ”dispara” (um problema é criado no Zabbix).

A condição de problema pode ser baseada no valor mais recente, no valor médio, em uma string descoberta e em muitas outras funções. Triggers não podem ser criados para items que retornam valores com o tipo de dado binário ou JSON data type.

Uma expressão de trigger simples calcula a função (por exemplo, `max()` para o valor máximo) para os dados especificados entre parênteses (normalmente item de dados e janela de tempo) e, em seguida, compara esse cálculo a um limite.

Por exemplo:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Esse trigger dispara se o espaço em disco livre, medido ao longo de 5 minutos, estiver constantemente abaixo de 10 GB.

Um trigger pode ter os seguintes estados:

Estado do trigger	Descrição	No Zabbix
OK	A expressão do trigger foi avaliada como "0" (ou FALSO)	Nenhum problema criado O problema é resolvido (se existia)
Problema	A expressão do trigger foi avaliada como "1" (ou VERDADEIRO)	O problema é criado

Às vezes, a avaliação da expressão do trigger não é possível por algum motivo. Veja Estado de expressão desconhecida.

Outros recursos:

- Exemplos de trigger
- Configurando um trigger
- Melhores práticas

Frequência de cálculo

Um trigger é recalculado toda vez que o servidor Zabbix 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 aquela que recebeu o novo valor).

Além disso, um trigger é recalculado a cada 30 segundos (pelo history syncer) se a expressão contiver:

- Funções de Data e hora
- Função nodata()

Triggers contendo apenas funções de tendência são avaliados uma vez por menor período de tempo na expressão. Enquanto muitas funções de trigger usam dados de histórico do item, funções de tendência usam dados de tendência.

Exemplos de trigger

Esta página é uma coleção de exemplos de trigger.

Os exemplos estão organizados por caso de uso:

- host indisponível
- proxy indisponível
- cluster SMTP indisponível
- host inacessível
- reinicialização inesperada
- alterações em arquivos importantes
- alterações na consulta DNS
- software não correspondente em diferentes hosts
- relógios dessincronizados
- agent desatualizado
- interface oscilando
- alto tráfego de entrada
- baixo espaço em disco
- baixo espaço em disco (limiar dinâmico)
- alta carga de CPU
- comparação de carga de CPU
- comparação de carga de CPU de longo prazo
- string contendo caracteres especiais

Detectar host indisponível

```
max(/host/zabbix[host,agent,available],5m)=0
```

Este trigger dispara se o agent Zabbix no host estiver indisponível por 5 minutos.
 Função: `max`
 Item de dados: `zabbix[host,agent,available]`

Alternativa:

```
nodata(/host/agent.ping,5m)=1
```

Este trigger dispara se nenhum dado for recebido do agent Zabbix por 5 minutos.
 Função: `nodata`
 Item de dados: `agent.ping`

Detectar proxy indisponível

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Este trigger é acionado se os dados do proxy Zabbix estiverem atrasados em relação ao horário do servidor Zabbix em 1 minuto.
 Função: fuzzytime
 Item de dados: zabbix[proxy,{$PROXY_NAME},lastaccess]

Detectar cluster SMTP indisponível

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Este trigger dispara se ambos os servidores SMTP estiverem inativos.
 Função: last
 Item de dados: net.tcp.service

Detectar host inacessível

```
count(/host/icmpping,30m,,"0")>5
```

Este trigger dispara se o host estiver inacessível por ping mais de 5 vezes nos últimos 30 minutos.
 Função: count
 Item de dados: icmpping

Detectar reinicialização inesperada

```
change(/host/system.uptime)<0
```

Este trigger dispara se uma alteração negativa no valor do tempo de atividade do sistema for detectada (indicando uma reinicialização).
 Função: change
 Item de dados: system.uptime

Detectar alterações em arquivos importantes

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Este trigger dispara se /etc/passwd foi alterado. A expressão é verdadeira quando o checksum anterior 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.
 Função: last
 Item de dados: vfs.file.cksum

Detectar alterações na consulta DNS

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Observe as aspas ao redor do segundo operando.

Este trigger é acionado se o resultado da consulta não for igual ao que ela normalmente retorna:

```
example.com             MX       0 mail.example.com
```

Função: last
 Data item: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], com macros definidas como:

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Detectar software diferente em hosts diferentes

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Este trigger dispara se a versão do Ubuntu for diferente em hosts diferentes. Observe como os operandos aqui são funções que retornam strings.
 Função: last
 Item de dados: vfs.file.contents

Detectar relógios dessincronizados

```
fuzzytime(/host/system.localtime,10s)=0
```

O trigger dispara se a hora local do cliente e a hora do server Zabbix diferirem em mais de 10 segundos.
 Função: fuzzytime
 Item de dados: system.localtime

Observe que system.localtime deve ser configurado como uma verificação passiva para o agent Zabbix; no Zabbix agent 2, ele pode ser configurado como uma verificação ativa.

Detectar agent desatualizado

```
find(/host/agent.version,,"like","beta")=1
```

Este trigger dispara se o agent Zabbix tiver uma versão beta. O agent Zabbix precisa ser atualizado.
 Função: find
 Item de dados: agent.version

Detectar interface instável

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Este trigger dispara se o estado operacional (up/down/unknown) da eth0 mudou mais de 5 vezes em uma hora.
 Função: changecount
 Item de dados: vfs.file.contents

Detectar tráfego de entrada alto

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Este trigger dispara se o número de bytes recebidos em eth0 nos últimos cinco minutos foi sempre superior a 100 kilobytes. Alguém provavelmente está baixando um arquivo grande.
 Função: min
 Item de dados: net.if.in[eth0,bytes]

Detectar pouco espaço em disco

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

O trigger dispara se o espaço livre em disco estiver consistentemente (5 minutos) abaixo de 10 GB.

Você também pode definir uma expressão de recuperação:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

O problema é **resolvido** apenas quando o espaço livre em disco está consistentemente (10 minutos) acima de 40 GB.
 Função: min
 Item de dados: vfs.fs.size

Detectar pouco espaço em disco (limite dinâmico)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

O trigger dispara se o armazenamento livre (em unidades de alocação) cair abaixo de 10 por cento. Observe o valor de outro item sendo usado para obter um limite de trigger adaptativo, aplicável ao armazenamento descoberto de vários tamanhos. Função: last

Detectar alta carga de CPU

```
last(/host/system.cpu.load[all,avg1])>5
```

O trigger dispara quando a carga média do processador está acima de 5 por um minuto.

Variações:

```
min(/host/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

```
min(/host/system.cpu.load[all,avg1],5m)>2 and not (dayofweek()=7 and time()>230000) and not (dayofweek()=1
```

Esses triggers analisam 5 minutos de dados e disparam apenas se a carga da CPU nunca estiver abaixo de 2. Além disso, esses triggers disparam:

- apenas durante a noite (00:00-06:00)
- a qualquer momento, exceto por 2 horas na mudança da semana (domingo, 23:00 - segunda-feira, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Este trigger dispara se a carga do processador estiver muito alta em pelo menos dois dos três hosts.
 Funções: last, min, dayofweek, time
 Item de dados: system.cpu.load

Comparar cargas de CPU

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

O trigger dispara se a carga média de hoje ultrapassar a carga média da mesma hora de ontem (usando o deslocamento de tempo como now-1d) mais de duas vezes.
 Função: avg
 Item de dados: system.cpu.load

Comparar cargas de CPU de longo prazo

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Este trigger dispara se a carga de CPU no host aumentou mais de 10% no último mês
 Função: trendavg
 Item de dados: system.cpu.load

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:

```
A carga do servidor {HOST.HOST} aumentou em {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.
```

Também é útil permitir o fechamento manual na configuração do trigger para esse tipo de problema.

Detectar string contendo caracteres especiais

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

O trigger dispara se o conteúdo de `/tmp/hello` for igual à string definida em `{$HELLO_MACRO}`:

```
{$HELLO_MACRO} = \" //hello ?\"
```

Alternativamente, você pode comparar diretamente com a string:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Observe como os caracteres especiais (\\ e ") são escapados quando a string é comparada diretamente.

Função: `last`
 Item de dados: `vfs.file.contents`

Configurando um trigger

Esta página descreve como configurar um trigger no frontend do Zabbix.

Observe que os templates oficiais do Zabbix contêm triggers predefinidos.
Consulte o guia para aplicar o template a monitor Linux.
Também é possível atualizar em massa triggers configurados.

Outros recursos:

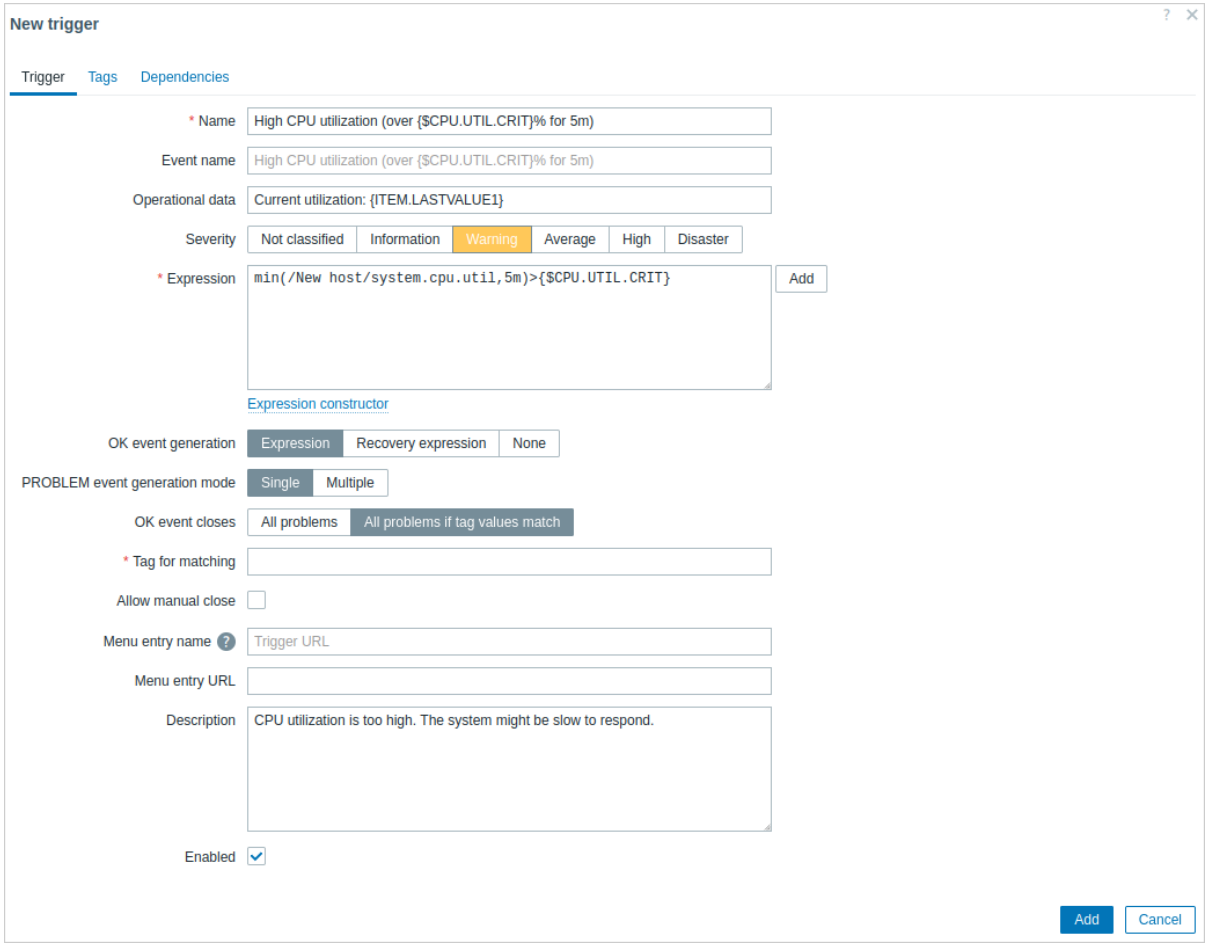
- Detecção de problemas com triggers (introdução geral)
- Exemplos de triggers (por caso de uso)
- Expressão e funções de trigger (detalhes de sintaxe)
- Boas práticas

Configuração

Para configurar um trigger, faça o seguinte:

1. Vá para *Coleta de dados* > *Hosts*.
2. Clique em *Triggers* na linha do host.
3. Clique em *Criar trigger* à direita (ou no nome do trigger para editar um trigger existente).
4. Insira os parâmetros do trigger no formulário.

A aba **Trigger** contém todos os atributos essenciais do trigger.

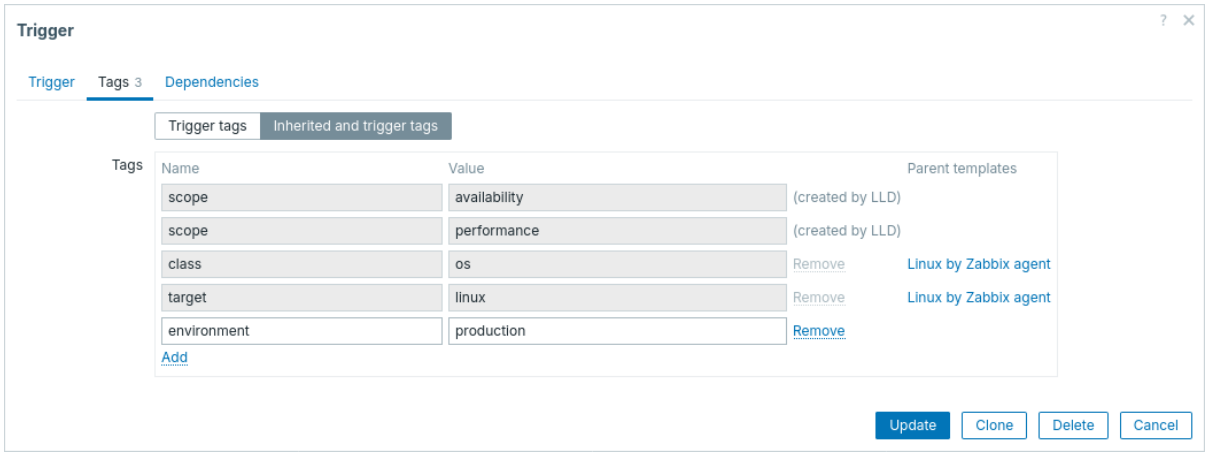


Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parameter	Description
Name	Nome do trigger. Os macros suportados são: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*} e macros de usuário {$MACRO}. As macros **$1, $2...$9** podem ser usadas para referenciar a primeira, segunda...nona constante da expressão. *Nota*: as macros $1-$9 serão resolvidas corretamente se estiverem se referindo a constantes em expressões relativamente simples e diretas. Por exemplo, o nome "Processor load above $1 on {HOST.NAME}" será alterado automaticamente para "Processor load above 5 on New host" se a expressão for last(/New host/system.cpu.load[percpu,avg1])>5.
Event name	Quando definido, este nome será usado para criar o nome do evento de problema. Por padrão, o nome do evento é o mesmo que o nome do trigger. O nome do evento pode ser usado para criar alertas significativos contendo dados do problema (veja o exemplo). O mesmo conjunto de macros é suportado como no nome do trigger, além das macros de expressão {TIME}, {TIMESTAMP} e {?EXPRESSION}.

Parameter	Description
Operational data	Insira alguma string com macros para exibir dados dinâmicos em tempo real em *Monitoramento* > *Problemas*. Como alternativa, deixe este campo vazio para exibir os valores mais recentes de todos os items da expressão do trigger. O mesmo conjunto de macros é suportado como no nome do trigger, com a capacidade de resolução dinâmica. Por exemplo: {ITEM.VALUE<1-9>} é resolvido para os valores do item no momento em que o estado do trigger é alterado (problema criado, resolvido, fechado manualmente ou fechado por correlação). {ITEM.LASTVALUE<1-9>} é resolvido para os valores mais recentes do item. Observe que fechar um problema manualmente não produz um novo valor, portanto ambas as macros ainda mostrarão o valor do momento do problema. Observe também que ambas as macros são resolvidas como *UNKNOWN* se o valor mais recente for mais antigo que *Max history display period* (veja *Administration > General*).
Severity	Defina a severity necessária do trigger clicando nos botões.
Expression	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 depois que todas as condições incluídas na expressão são atendidas, ou seja, a expressão é avaliada como TRUE. O problema será resolvido assim que a expressão for avaliada como FALSE, a menos que condições adicionais de recuperação sejam especificadas em *Recovery expression*.
OK event generation	Opções de geração de evento OK: **Expression** - eventos OK são gerados com base na mesma expressão dos eventos de problema; **Recovery expression** - eventos OK são gerados se a expressão do problema for avaliada como FALSE e a expressão de recuperação for avaliada como TRUE; **None** - nesse caso, o trigger nunca retornará ao estado OK por conta própria.
Recovery expression	Expressão lógica opcional que define condições adicionais que precisam ser atendidas antes que o problema seja resolvido, depois que a expressão original do problema já tiver sido avaliada como FALSE. A expressão de recuperação é útil para a histerese do trigger. Não é **possível** resolver um problema apenas pela expressão de recuperação se a expressão do problema ainda for TRUE. Este campo só está disponível se 'Recovery expression' estiver selecionado para *OK event generation*.
PROBLEM event generation mode	Modo de geração de eventos de problema: **Single** - um único evento é gerado quando um trigger entra no estado 'Problem' pela primeira vez; **Multiple** - um evento é gerado em *cada* avaliação 'Problem' do trigger.
OK event closes	Selecione se o evento OK fecha: **All problems** - todos os problemas deste trigger; **All problems if tag values match** - apenas os problemas do trigger com valores de tag de evento correspondentes.
Tag for matching	Insira o nome da tag do evento a ser usado para correlação de eventos. Este campo é exibido se 'All problems if tag values match' estiver selecionado para a propriedade *OK event closes* e é obrigatório nesse caso.
Allow manual close	Marque para permitir o fechamento manual dos eventos de problema gerados por este trigger. O fechamento manual é possível ao reconhecer eventos de problema.
Menu entry name	Se não estiver vazio, o nome inserido aqui (até 64 caracteres) é usado em vários locais do frontend como rótulo para a URL do trigger especificada no parâmetro *Menu entry URL*. Se estiver vazio, o nome padrão *Trigger URL* é usado. O mesmo conjunto de macros é suportado como no nome do trigger, além de {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}.
Menu entry URL	Se não estiver vazio, a URL inserida aqui (até 2048 caracteres) fica 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 do dashboard *Problemas*. O mesmo conjunto de macros é suportado como no nome do trigger, além de {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}. Observação: macros de usuário com valores secretos não serão resolvidas na URL.
Description	Campo de texto usado para fornecer mais informações sobre este trigger. Pode conter instruções para corrigir um problema específico, dados de contato da equipe responsável etc. O mesmo conjunto de macros é suportado como no nome do trigger.
Enabled	Desmarcar esta caixa desativará o trigger, se necessário. Os problemas de um trigger desativado 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 *Inherited and trigger tags* permite visualizar as 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" nem exibe tags no nível do host.

Parameter	Description
Name/Value	Defina tags personalizadas para marcar eventos de trigger. Tags são um par de nome e valor de tag. Você pode usar apenas o nome ou combiná-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 em baixo nível e funções de macro funções com `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` são suportadas em tags de evento. Macros de descoberta em baixo nível podem ser usadas dentro do contexto da macro. A macro {TRIGGER.ID} é suportada nos valores de tag do trigger. Isso 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 truncado 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 **Dependencies** contém todas as dependências do trigger.

Clique em *Add* para adicionar uma nova dependência.

> **Note:**
> Você também pode configurar um trigger abrindo um existente, clicando no botão *Clone* e depois salvando com um nome diferente.

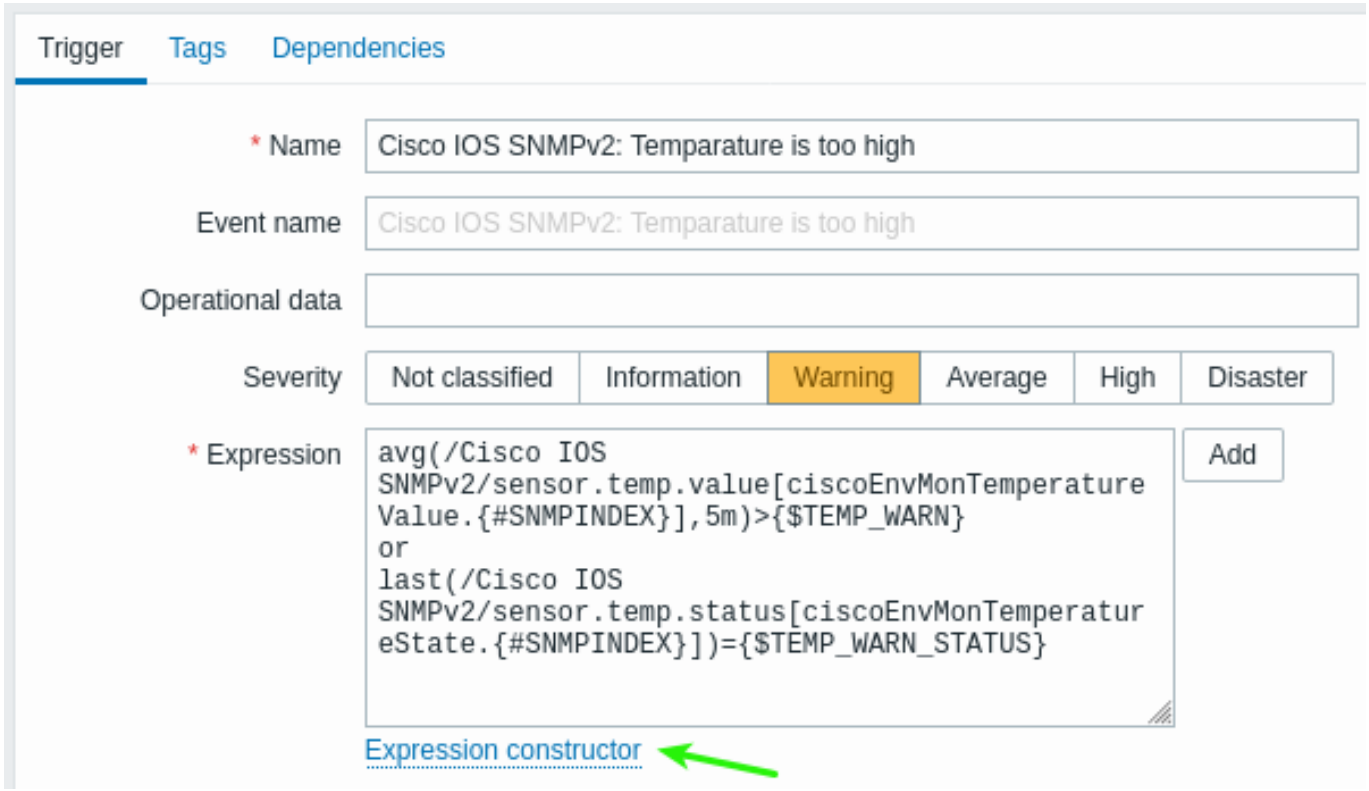
Teste de expressão de trigger

É possível testar qual seria o resultado da expressão dependendo do valor recebido.

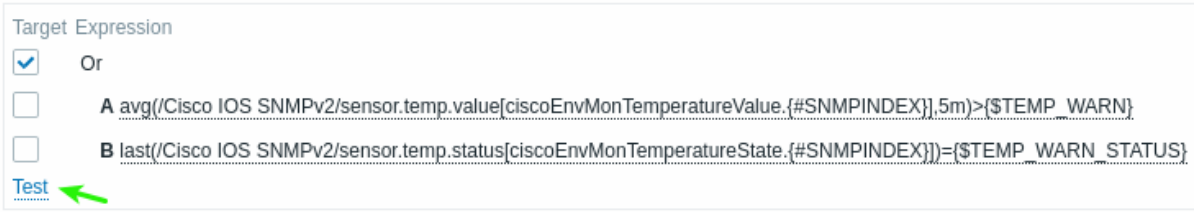
A expressão a seguir de um template oficial é usada como exemplo:

```
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}
```

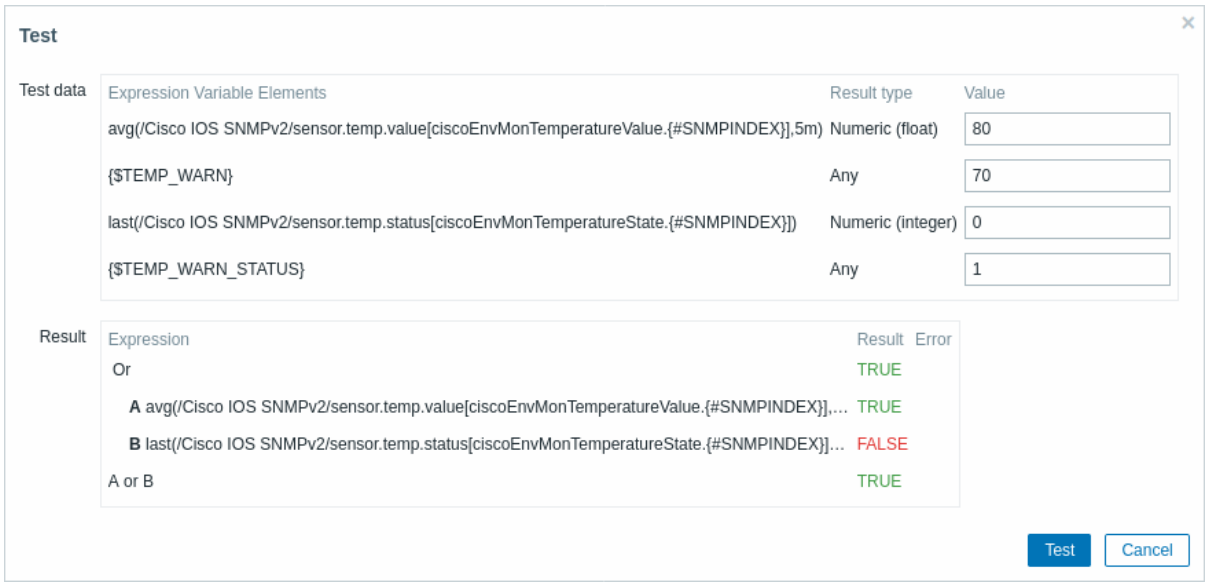
Para testar a expressão, clique em *Construtor de expressão* abaixo do campo de expressão.



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, assim como da expressão completa, pode ser visto.

”TRUE” significa que a expressão especificada está correta. Neste caso 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 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 ocorreu um problema.

Expressão de trigger e funções

Visão geral

Esta página descreve a sintaxe da expressão do trigger e detalhes de avaliação.

A sintaxe de uma expressão simples é:

```
function(/host/item,time_period)<operator><constant>
```

Nesta expressão, o primeiro operando (à esquerda do operador) é uma função com seus parâmetros entre parênteses (normalmente o item de dados e o período de tempo).

A função é usada para analisar os dados recebidos durante o período de tempo definido, resultando em um valor calculado.

Este valor é então comparado ao segundo operando, usando o operador. Neste exemplo, o segundo operando é uma constante, mas também pode ser outra função.

Por exemplo:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

Este trigger dispara se o número de bytes recebidos em `eth0` durante os últimos cinco minutos foi sempre superior a 100 kilobytes. Neste caso, a expressão é verdadeira e um problema é criado.

As expressões de trigger são extremamente flexíveis. Em expressões mais complexas, várias funções, operadores e constantes podem ser combinados.

Veja também:

- Exemplos de triggers
- Detecção de problemas com triggers (introdução geral)

Funções

As funções permitem analisar valores coletados, por exemplo, calcular a média ou localizar uma string específica.

Clique no grupo de funções correspondente para ver mais detalhes.

Grupo de funções	Funções
Funções de agregação	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 bit a bit	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, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
Funções de tendências	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

Grupo de funções	Funções
Funções de operadores	between, in
Funções preditivas	forecast, timeleft
Funções de string	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Exceto quando indicado de outra forma, estas funções são compatíveis com:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções Foreach são compatíveis somente com cálculos de agregação.

Normalmente, as funções retornam valores numéricos para comparação. Ao retornar strings, a comparação é possível com os operadores **=** e **<>** (consulte o exemplo Detectar software não correspondente em hosts diferentes).

Parâmetros de função

Os parâmetros de função permitem especificar:

- a chave do item (como `/host/key`) para funções que fazem referência ao histórico de um item do host
- o período de tempo (e outros parâmetros específicos da função)
- outras expressões

Chave do item

O item referenciado deve estar em um estado suportado (exceto para a função **nodata()**, que também é calculada para itens não suportados).

A omissão do nome do host no primeiro parâmetro (ou seja, como em `function(//key,parameter,...)`) só é suportada em determinados 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 usado no caso do campo *Nome do evento* e do elemento de mapa "Trigger" para se referir a um item específico na expressão do trigger. Quando o nome do host é omitido ou substituído por `{HOST.HOST}` nesses contextos, a referência aponta para o primeiro item na expressão do trigger ou para o primeiro item no gráfico. Fora desses contextos suportados, omitir o nome do host em expressões de trigger resultará em erro. Veja o exemplo Comparar cargas de CPU de longo prazo para uma ilustração do uso de barra dupla em macros de nome de evento.

Período de tempo

Os parâmetros específicos da função são colocados após a chave do item e separados dela por uma vírgula.

A maioria das funções numéricas aceita um **período de tempo** como parâmetro. Ele permite especificar o intervalo no qual estamos interessados. Ele pode ser especificado como um período de tempo (30s, 10m, 1h) ou como um intervalo de valores (#5 - para os cinco valores mais recentes).

Você pode usar segundos ou sufixos de tempo para indicar o período de tempo. Quando precedido por um símbolo de hash, o parâmetro tem um significado diferente:

Expression	Description
sum(/host/key,**10m**)	Soma dos valores dos últimos 10 minutos.
sum(/host/key,**#10**)	Soma dos dez últimos valores.

Os parâmetros com um símbolo de hash têm um significado diferente com a função **last**: eles indicam o N-ésimo valor anterior. Portanto, considerando os valores 30, 70, 20, 60, 50 (do mais recente ao menos recente):

- `last(/host/key,#2)` retornaria '70'
- `last(/host/key,#5)` retornaria '50'

O período de tempo é medido até "agora" - em que "agora" é o horário do recálculo mais recente do trigger (consulte Frequência de cálculo); "agora" não é o horário atual do server.

O período de tempo especifica:

- Todos os valores entre "agora - período de tempo" e "agora" (ou, com o deslocamento de tempo aplicado, entre "agora - deslocamento de tempo - período de tempo" e "agora - deslocamento de tempo").
- O número especificado de valores do passado, até "agora".

Se não existirem dados para o período de avaliação especificado, o trigger ou item calculado que usa essa função ficará sem suporte.

Observe que:

- Se apenas uma função (que faz referência ao histórico de dados) for usada no trigger, "agora" será sempre o valor recebido mais recentemente. Por exemplo, se o último valor tiver sido recebido há uma hora, o período de tempo será considerado até o valor mais recente recebido há uma hora.
- Um novo trigger é calculado assim que o primeiro valor é recebido (funções de histórico); ele será calculado em até 30 segundos para as funções de data e hora e nodata(). Assim, o trigger será calculado mesmo que o período de tempo definido (por exemplo, uma hora) ainda não tenha transcorrido desde a criação do trigger. O trigger também será calculado após o primeiro valor, mesmo que o intervalo de tempo tenha sido definido, por exemplo, como os dez valores mais recentes.

Deslocamento de tempo

Um deslocamento de tempo opcional é compatível com a contagem de tempo ou de valores como parâmetro da função. Esse parâmetro permite referenciar dados de um período no passado.

O deslocamento de tempo começa com `now`, que especifica o horário atual, e é seguido por `+N<time unit>` ou `-N<time unit>`, para adicionar ou subtrair N unidades de tempo.

Por exemplo, `avg(/host/key,1h:now-1d)` retornará o valor médio de uma hora de um dia atrás.

Attention:
O deslocamento de tempo especificado em meses (M) e anos (y) é compatível apenas com funções de tendência. Outras funções são compatíveis com segundos (s), minutos (m), horas (h), dias (d) e semanas (w).

Deslocamento de tempo com períodos absolutos

Períodos absolutos são compatíveis com o parâmetro de deslocamento de tempo, por exemplo, da meia-noite à meia-noite para um dia, de segunda-feira a domingo para uma semana e do primeiro ao último dia do mês para um mês.

O deslocamento de tempo para períodos absolutos começa com `now`, que especifica o horário atual, e é seguido por qualquer número de operações de tempo: `/<time unit>`, que define o início e o fim da unidade de tempo, por exemplo, da meia-noite à meia-noite para um dia; `+N<time unit>` ou `-N<time unit>`, 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.

Parameter	Description
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)`~~

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.

Operadores

Os seguintes operadores são suportados para triggers **(em ordem decrescente de prioridade de execução)**:

Prioridade	Operador	Definição	**Notas para valores desconhecidos**	Forçar conversão do operando para float [1]
1	**-**	Menos unário	**-**Desconhecido → Desconhecido	Sim
2	**not**	NOT lógico	**not** Desconhecido → Desconhecido	Sim
3	*	Multiplicação	0 * Desconhecido → Desconhecido (sim, Desconhecido, não 0 - para não perder Desconhecido em operações aritméticas) 1.2 * Desconhecido → Desconhecido	Sim
	/	Divisão	Desconhecido / 0 → erro Desconhecido / 1.2 → Desconhecido 0.0 / Desconhecido → Desconhecido	Sim
4	+	Soma aritmética	1.2 + Desconhecido → Desconhecido	Sim
	-	Subtração aritmética	1.2 - Desconhecido → Desconhecido	Sim
5	<	Menor que. O operador é definido como: A<B ⇔ (A<B-0.000001)	1.2 < Desconhecido → Desconhecido	Sim
	<=	Menor ou igual a. O operador é definido como: A<=B ⇔ (A≤B+0.000001)	Desconhecido <= Desconhecido → Desconhecido	Sim
	>	Maior que. O operador é definido como: A>B ⇔ (A>B+0.000001)		Sim
	>=	Maior ou igual a. O operador é definido como: A>=B ⇔ (A≥B-0.000001)		Sim
6	=	Igual a. O operador é definido como: A=B ⇔ (A≥B-0.000001) e (A≤B+0.000001)		Não [1]

Prioridade	Operador	Definição	**Notas para valores desconhecidos**	Forçar conversão do operando para float [1]
	<>	Diferente de. O operador é definido como: A<>B ⇔ (A<B-0.000001) ou (A>B+0.000001)		Não [1]
7	**and**	AND lógico	0 **and** Desconhecido → 0 1 **and** Desconhecido → Desconhecido Desconhecido **and** Desconhecido → Desconhecido	Sim
8	**or**	OR lógico	1 **or** Desconhecido → 1 0 **or** Desconhecido → Desconhecido Desconhecido **or** Desconhecido → Desconhecido	Sim

[1] O operando string ainda é convertido para numérico se:

- outro operando for numérico
- operador diferente de **=** ou **<>** for usado em um operando

(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 **-** unário e **not**, têm associatividade da esquerda para a direita. **-** unário e **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'.

Estado de expressão desconhecido

É possível que um operando desconhecido apareça em uma expressão de trigger quando:

- um item sem suporte é usado
- a avaliação de uma função, para um item com suporte, resulta em erro

Nesse caso, a expressão de trigger geralmente é avaliada como `Unknown` (pois não pode ser avaliada)

É possível receber notificações sobre triggers desconhecidos.

Exceções

Apesar de um operando desconhecido, as expressões de trigger podem ser avaliadas para um resultado conhecido (Problem/OK) em alguns casos:

- A função `nodata()` é avaliada independentemente de o item referenciado ter suporte ou não.
- Expressões com AND/OR podem ser avaliadas para um resultado conhecido em dois casos:
  - **Caso 1**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" é avaliada para um resultado conhecido ('1' ou "Problem"),
  - **Caso 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" é avaliada para um resultado conhecido ('0' ou "OK").

- Se a avaliação da função para um item com suporte resultar em erro, o valor da função se torna `Unknown` e ele participa como operando desconhecido em avaliações posteriores da expressão.

Operandos desconhecidos podem "desaparecer" apenas em expressões lógicas, conforme descrito acima. Em expressões aritméticas, operandos desconhecidos sempre levam a `Unknown` (exceto divisão por 0).

Attention:
O estado de expressão desconhecido não altera o estado do trigger (Problem/OK). Portanto, se o estado do trigger era "Problem" (veja o Caso 1), ele permanece no estado de problema mesmo que a parte conhecida seja resolvida ('1' se torne '0'), porque a expressão agora é avaliada como `Unknown` e isso não altera o estado do trigger.

Se uma expressão de trigger com vários itens sem suporte for avaliada como `Unknown`, a mensagem de erro no frontend se refere ao último item sem suporte avaliado.

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.

Funções agregadas

Exceto quando indicado de outra forma, todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Fórmulas de item calculado
- 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))` (somente em fórmulas de item calculado; para usar em triggers, referencie o item calculado pela sua key)

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Function	Description
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 "cauda" da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido.
mad	O desvio absoluto mediano 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 populacional nos valores coletados dentro do período de avaliação definido.
stddevsamp	O desvio padrão amostral 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 populacional dos valores coletados dentro do período de avaliação definido.
varsamp	A variância amostral dos valores coletados dentro do período de avaliação definido.

Parâmetros comuns

- `/host/key` é um primeiro parâmetro obrigatório comum para as funções que fazem referência ao histórico do item do host.
- `(sec|#num)<:time shift>` é um segundo parâmetro comum para as funções que fazem referência ao histórico do item do host, em que:
  - **sec** - período máximo de avaliação em segundos (podem ser usados sufixos de tempo).
  - **#num** - intervalo máximo de avaliação nos valores coletados mais recentes (se precedido por um símbolo de cerquilha).
  - **time shift** (opcional) permite mover o ponto de avaliação para trás no tempo. Consulte 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 entre aspas

avg(/host/key,(sec|#num)<:time shift>)

O valor médio 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.

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) #calcula o valor médio da última hora até agora
avg(/host/key,#5) #calcula o valor médio dos cinco valores mais recentes
avg(/host/key,1h:now-1d) #calcula o valor médio de uma hora, de 25 horas atrás até 24 horas atrás a partir
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #calcula a carga média do processador
avg(/host/proc.num,5m)>300 #trigger se o número médio de processos nos últimos 5 minutos tiver ficado acim
```

bucket_percentile(item filter,time period,percentage)

Calcula o percentil a partir dos buckets de um histograma.

Parâmetros:

- **item filter** - veja item filter;
- **time period** - veja time period;
- **percentage** - porcentagem (0-100).

Comentários:

- Suportado apenas em itens calculados;
- Esta função é um alias para `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Exemplos:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #calcula o tempo de resposta do 95º pe
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #calcula a latência da API no 99
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

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 item filter;
- **time period** - veja time period;
- **operator** (deve estar entre aspas duplas). `operators` suportados:
eq - igual
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like - corresponde se contiver o padrão (sensível a maiúsculas e minúsculas)
bitand - AND bit a bit
regexp - corresponde à expressão regular fornecida em `pattern`, com sensibilidade a maiúsculas e minúsculas
iregexp - corresponde à expressão regular fornecida em `pattern`, sem sensibilidade a maiúsculas e minúsculas

- **pattern** - o padrão necessário (argumentos de string devem estar entre aspas duplas); suportado se *operator* for especificado no terceiro parâmetro.

Comentários:

- Usar **count()** com uma função foreach relacionada ao histórico (max_foreach, avg_foreach, etc.) pode gerar impacto de desempenho, enquanto usar **exists_foreach()**, que funciona apenas com dados de configuração, não terá esse efeito.
- Parâmetros opcionais *operator* ou *pattern* não podem ficar vazios após uma vírgula; eles só podem ser totalmente omitidos.
- Com *bitand* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por '/': **number_to_compare_with/mask**. count() calcula o "AND bit a bit" do valor com a *mask* e compara o resultado com *number_to_compare_with*. Se o resultado do "AND bit a bit" for igual a *number_to_compare_with*, o valor é contado.
Se *number_to_compare_with* e *mask* forem iguais, apenas a *mask* precisa ser especificada (sem '/').
- Com *regexp* ou *iregexp* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência com regexp, valores float sempre serão representados com 4 dígitos decimais após '.'. Observe também que, para números grandes, a diferença entre a representação decimal (armazenada no banco de dados) e a binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
count(max_foreach(/*/net.if.in[*],1h)) #calcula o número de items net.if.in que receberam dados na última
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #calcula o número de sistemas de arquivos com mais de
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Calcula o φ-quantil a partir dos buckets de um histograma.
 Compatível com a função foreach: *bucket_rate_foreach*.

Parâmetros:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - pares inseridos manualmente (>=2) de parâmetros ou a resposta de bucket_rate_foreach.

Comentários:

- Compatível apenas em itens calculados;
- Corresponde funcionalmente a 'histogram_quantile' do PromQL;
- Retorna -1 se os valores do último bucket 'Infinity' ("+*inf*") forem iguais a 0.

Exemplos:

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

A contagem de items existentes na configuração que correspondem aos critérios do filtro.
 Tipo de valor suportado: *Integer*.

Parâmetro:

- **item filter** - critérios para seleção de item, permite referenciar por grupo de hosts, host, chave de item e tags. Caracteres curinga são suportados. Veja item filter para mais detalhes.

Comentários:

- Suportado apenas em items calculados;
- Funciona como um alias para a função *count(exists_foreach(item_filter))*.

Exemplos:

```
item_count(/*/agent.ping?[group="Host group 1"]) #calcula o número de hosts com o item agent.ping em "Host
```

kurtosis(/host/key,(sec|#num)<:deslocamento de tempo>)

A "caudalidade" da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido. Veja também: Kurtosis.
 Tipos de valor suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplos:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigger on sharp spikes of CPU utilization over 5 minutes
```

mad(/host/key,(sec|#num)<:deslocamento de tempo>)

O desvio absoluto mediano nos valores coletados dentro do período de avaliação definido. Veja também: Desvio absoluto mediano.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplos:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

O valor mais alto 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.

Exemplos:

```
max(/host/key,1h) - min(/host/key,1h) #calcula a diferença entre os valores máximo e mínimo na última hora
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #retorna o maior uso de disco entre tod
```

min(/host/key,(sec|#num)<:deslocamento de tempo>)

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.

Exemplos:

```
max(/host/key,1h) - min(/host/key,1h) #calcula a diferença entre os valores máximo e mínimo na última hora
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #retorn
```

skewness(/host/key,(sec|#num)<:time shift>)

A assimetria da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido. Veja também: Skewness.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplos:

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigger on sudden negative drops indicating crashes/restarts
```

stddevpop(/host/key,(sec|#num)<:time shift>)

O desvio padrão populacional 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.

Exemplos:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #trigger se o uso de CPU oscilar muito (o desvio padrão popu
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

O desvio padrão amostral dos valores coletados dentro do período de avaliação definido. Veja também: Desvio padrão.
 Tipos de valor 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.

Exemplos:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #trigger se o uso de CPU oscilar muito (o desvio padrão amo
```

sum(/host/key,(sec|#num)<:time shift>)

A soma dos valores coletados 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.

Exemplos:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #calcula o tráfego total de rede de entrada par
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #calcula o espaço total em disco para t
sum(last_foreach(/*/net.dns[*,*,*])) #calcula o número total de verificações DNS bem-sucedidas
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

A soma dos quadrados nos valores coletados dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplos:

```
sumofsquares(/host/calculated.net.if.rate,1m) #a soma dos quadrados da taxa de tráfego de rede (entrada vs
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #a soma dos quadrados dos valores mais
```

varpop(/host/key,(sec|#num)<:time shift>)

A variância populacional dos valores coletados dentro do período de avaliação definido. Veja também: Variância.
 Tipos de valor suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplos:

```
varpop(/host/system.cpu.util[,user],10m)>50 #trigger se o uso de CPU oscilar muito (a variância estiver ac
```

varsamp(/host/key,(sec|#num)<:time shift>)

A variância amostral dos valores coletados dentro do período de avaliação definido. Veja também: Variância.
 Tipos de valor 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.

Exemplos:

```
varsamp(/host/system.cpu.util[,user],10m)>50 #trigger se o uso de CPU oscilar muito (a variância amostral
```

Veja todas as funções suportadas.

Funções foreach

Visão geral

As funções Foreach são usadas em cálculos agregados para retornar um valor agregado para cada item selecionado pelo **filtro de item** usado. Um array de valores é retornado.

Por exemplo, a função `avg_foreach` retornará um array de valores, em que cada valor é o valor de histórico *médio* do item selecionado, durante o intervalo de tempo especificado.

Attention:
As funções Foreach são suportadas apenas em fórmulas de item calculado. Elas não podem ser invocadas em chaves de item simples nem usadas diretamente para construir expressões de trigger. Em expressões de trigger, um item calculado pode ser usado referenciando sua chave.

O filtro de item faz parte da sintaxe usada pelas funções foreach. O uso de curingas é □□□□□ado no filtro de item, portanto os itens necessários podem ser selecionados de forma bastante flexível.

Funções suportadas

Function	Description
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()`, em que "limite superior do bucket" é o valor do parâmetro da chave do item definido pelo parâmetro <parameter number>.
count_foreach	Retorna o número de valores para cada item.
exists_foreach	Retorna '1' para cada item habilitado.

Function	Description
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 os detalhes abaixo) e `período de tempo`:

```
foreach_function(item filter,time period)
```

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 do item
  - "<text>" - constante de string, com o caractere de escape \ para escapar " e \
- operadores de comparação de string sensíveis a maiúsculas e minúsculas: =, <>
- operadores lógicos: `and`, `or`, `not`
- agrupamento com parênteses: ( )

A citação de constantes de string é obrigatória. Apenas a comparação de string completa e sensível a maiúsculas e minúsculas é suportada.

Warning:
Ao especificar tags no filtro (ou seja, `tag="tagname:value"`), o dois-pontos ":" é usado como delimitador. Tudo após ele é considerado o valor da tag. Portanto, atualmente não é suportado especificar um nome de tag contendo ":".

Exemplos

Um filtro complexo pode ser usado, referenciando a chave do item, grupo de host e tags, conforme ilustrado pelos exemplos:

Exemplo de sintaxe	Descrição
`/host/key[abc,*]`	Corresponde a itens semelhantes neste host.
`/*/key`	Corresponde ao mesmo item de qualquer host.

Exemplo de sintaxe	Descrição
`/*/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 parâmetro **segundo** permite especificar o período de tempo para a agregação. O período de tempo só pode ser expresso como tempo; a quantidade de valores (prefixada com **#**) não é compatível.

Os símbolos de unidade compatíveis podem ser usados neste parâmetro por conveniência, por exemplo, `5m` (cinco minutos) em vez de `300s` (300 segundos) ou `1d` (um dia) em vez de 86400 (86400 segundos).

Para a função `last_foreach`, o período de tempo é um parâmetro opcional (compatível desde o Zabbix 7.0) e pode ser omitido:

```
last_foreach(/*/key?[group="host group"])
```

O período de tempo não é compatível com a função `exists_foreach`.

Parâmetros adicionais

bucket_rate_foreach

Um terceiro parâmetro opcional é suportado pela função `bucket_rate_foreach`:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

onde `<parameter number>` é a posição do valor "bucket" na chave do item. Por exemplo, se o valor "bucket" em `myItem[aaa,0.2]` for `0.2`, então sua posição é 2.

O valor padrão de <parameter number> é 1.

count_foreach

Terceiro e quarto parâmetros opcionais são suportados pela função `count_foreach`:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Onde:

- **operator** é o operador condicional para os valores do item (deve estar entre aspas duplas). Operadores suportados:
eq - igual
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like - corresponde se contiver o padrão (sensível a maiúsculas e minúsculas)
bitand - AND bit a bit
regexp - corresponde, com sensibilidade a maiúsculas e minúsculas, à expressão regular fornecida em `pattern`
iregexp - corresponde, sem sensibilidade a maiúsculas e minúsculas, à expressão regular fornecida em `pattern`.
- **pattern** é o padrão obrigatório (argumentos de string devem estar entre aspas duplas); é suportado se *operator* for especificado no terceiro parâmetro.

Comentários:

- Os parâmetros opcionais *operator* ou *pattern* não podem ser deixados em branco após uma vírgula; eles devem ser totalmente omitidos.
- Com *bitand* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por '/': **number_to_compare_with/mask**. `count_foreach()` calcula o "AND bit a bit" do valor com a *mask* e compara o resultado com *number_to_compare_with*. Se o resultado do "AND bit a bit" for igual a *number_to_compare_with*, o valor é contado.
Se *number_to_compare_with* e *mask* forem iguais, apenas a *mask* precisa ser especificada (sem '/').
- Com *regexp* ou *iregexp* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência de regexp, valores de ponto flutuante sempre serão representados com 4 dígitos decimais após '.'. Observe também que, para números grandes, a diferença

entre a representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Consulte cálculos agregados para mais detalhes e exemplos de 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 históricos.

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.

Funções bitwise

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Function	Description
bitand	O valor de ”AND bit a bit” de um valor de item e uma máscara.
bitlshift	O deslocamento bit a bit para a esquerda de um valor de item.
bitnot	O valor de ”NOT bit a bit” de um valor de item.
bitor	O valor de ”OR bit a bit” de um valor de item e uma máscara.
bitrshift	O deslocamento bit a bit para a direita de um valor de item.
bitxor	O valor de ”OR exclusivo bit a bit” de um valor de item e uma máscara.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Expressões são aceitas como parâmetros
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >

bitand(value,mask)

O valor de ”E bit a bit” de um valor de item e máscara.
 Tipos de 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(value,bits to shift)

O deslocamento de bits para a esquerda de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **bits to shift** (obrigatório) - o número de bits a serem deslocados.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitnot(value)

O valor de ”bitwise NOT” de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetro:

- **value** - o valor a ser verificado.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitor(value,mask)

O valor de ”OU bit a bit” de um valor de item e máscara.
 Tipos de 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.

bitrshift(value,bits to shift)

O deslocamento de bits para a direita de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **bits to shift** (obrigatório) - o número de bits a serem deslocados.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitxor(value,mask)

O valor de ”OU exclusivo bit a bit” de um valor de item e máscara.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **mask** (obrigatório) - um inteiro sem sinal de 64 bits (0 - 18446744073709551615).

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

Veja todas as funções suportadas.

Funções de data e hora

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

Attention:
As funções de data e hora não podem ser usadas sozinhas na expressão; pelo menos uma função de outro grupo, referenciando o item do host, deve ser incluída na expressão (exceto a função nodata()). Para informações detalhadas sobre como as funções de data e hora funcionam dentro das expressões, consulte Tempo de cálculo.

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Function	Description
date	A data atual no formato YYYYMMDD.
dayofmonth	O dia do mês no intervalo de 1 a 31.
dayofweek	O dia da semana no intervalo de 1 a 7.
now	A quantidade de segundos desde a Época (00:00:00 UTC, 1º 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, 1 de janeiro de 1970).

Exemplo:

`now()<1640998800`

time

A hora atual no formato HHMMSS.

Exemplo (apenas noturno, 00:00-06:00):

`time()<060000`

Veja todas as funções suportadas.

Funções de histórico

Todas as funções listadas aqui são compatíveis com:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver todos os detalhes.

Função	Descrição
change	A diferença entre o valor anterior e o mais recente.
changecount	O número de alterações 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 exclusivos dentro do período de avaliação definido.
find	Localiza uma correspondência de valor dentro do período de avaliação definido.
first	O primeiro valor (o mais antigo) dentro do período de avaliação definido.
firstclock	O timestamp do primeiro valor (o mais antigo) dentro do período de avaliação definido.
fuzzytime	Verifica quanto o horário do agent passivo difere do horário do Zabbix server/proxy.
jsonpath	Retorna uma matriz com os resultados da consulta JSONPath.
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 mais recente do item de log.
monodec	Verifica se houve uma diminuição monótona nos valores.
monoinc	Verifica se houve um aumento monótono nos valores.
nodata	Verifica se nenhum dado foi recebido.
percentile	O percentil P de um período, em que P (porcentagem) é especificado pelo terceiro parâmetro.
rate	A taxa média por segundo do aumento de um contador que cresce monotonicamente dentro do período definido.
xmlxpath	Retorna uma matriz com os resultados da consulta XML XPath.

Parâmetros comuns

- `/host/key` é um primeiro parâmetro obrigatório comum para as funções que fazem referência ao histórico do item do host.
- `(sec|#num)<:time shift>` é um segundo parâmetro comum para as funções que fazem referência ao histórico do item do host, em que:
  - **sec** - período máximo de avaliação em segundos (podem ser usados sufixos de tempo), ou
  - **#num** - intervalo máximo de avaliação em valores coletados mais recentes (se precedido por um símbolo de cerquilha).
  - **time shift** (opcional) permite mover o ponto de avaliação para trás no tempo. Consulte 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

change(/host/key)

A quantidade de diferença entre o valor anterior e o mais recente.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.
 Para strings retorna: 0 - valores são iguais; 1 - valores diferem.

Parâmetros: veja parâmetros comuns.

Comentários:

- A diferença numérica será calculada, conforme visto nestes valores de exemplo recebidos (’valor anterior’ e ’valor mais recente’ = diferença):
’1’ e ’5’ = +4
’3’ e ’1’ = −2
’0’ e ’-2.5’ = −2.5.
- Em triggers, útil para capturar picos repentinos (ou quedas), reinicializações de contadores, transições de estado numérico.
- Veja também: abs para comparação.

Exemplos:

```
change(/host/system.uptime)<0 #a alteração no tempo de atividade do sistema foi negativa desde o último va
change(/host/system.cpu.load[all,avg1])>2 #a carga da CPU (por um minuto) aumentou em mais de 2 desde o úl
change(/host/vfs.fs.size[/,free])<-1G #o espaço livre em disco caiu mais de 1 GB entre as verificações
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

O número de alterações entre valores adjacentes dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve estar entre aspas duplas) - valores possíveis: *all* - conta todas as alterações (padrão); *dec* - conta apenas as diminuições; *inc* - conta apenas os aumentos

Comentários:

- Para tipos de valor não numéricos, o parâmetro *mode* é ignorado.
- Em triggers, útil para capturar mudanças frequentes de estado (indicando instabilidade).

Exemplos:

```
changecount(/host/icmpping,10m)>5 #o status do ping mudou mais de 5 vezes em 10 minutos
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5 #o estado operacional da eth0 m
changecount(/host/proc.num[httpd],15m)>10 #o número de processos httpd mudou mais de 10 vezes em 15 minuto
changecount(/host/key,#10,"inc") #o número de aumentos de valor (em relação ao valor adjacente) entre os ú
changecount(/host/key,24h,"dec") #o número de diminuições de valor (em relação ao valor adjacente) nas últ
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

O número de valores dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **operator** (deve estar entre aspas duplas) Operadores `supported`:
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 (sensível a maiúsculas e minúsculas)
bitand - AND bit a bit
regexp - corresponde, com sensibilidade a maiúsculas e minúsculas, à expressão regular fornecida em `pattern`
iregexp - corresponde, sem sensibilidade a maiúsculas e minúsculas, à expressão regular fornecida em `pattern`
- **pattern** - o padrão necessário (argumentos de string devem estar 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 de regexp, valores float sempre serão representados com 4 dígitos decimais após '.'. Observe também que, para números grandes, a diferença entre a representação decimal (armazenada no banco de dados) e a binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
count(/host/icmpping,30m,,"0")>5 #ping falhou mais de 5 vezes em 30 minutos
count(/host/key,10m,"like","error") #o número de valores 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 valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **operator** (deve estar 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 (sensível a maiúsculas e minúsculas)
bitand - AND bit a bit
regexp - correspondência sensível a maiúsculas e minúsculas da expressão regular fornecida em `pattern`
iregexp - correspondência insensível a maiúsculas e minúsculas da expressão regular fornecida em `pattern`

- **pattern** - o padrão necessário (argumentos de string devem estar entre aspas duplas)

Comentários:

- Itens Float correspondem com a precisão de 2.22e-16.
- *like* não é suportado como operator para valores integer.
- *like* e *bitand* não são suportados como operators para valores float.
- Para valores string, text e log, apenas os operators *eq*, *ne*, *like*, *regexp* e *iregexp* são suportados.
- Com *bitand* como operator, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por '/': **number_to_compare_with/mask**. countunique() calcula "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 operator, 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 de regexp, valores float sempre serão representados com 4 dígitos decimais após '.'. Observe também que, para números grandes, a diferença entre a representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
countunique(/host/key,10m) #o número de valores únicos nos últimos 10 minutos até agora
countunique(/host/key,10m,"like","error") #o número de valores únicos nos últimos 10 minutos até agora que
countunique(/host/key,10m,,12) #o número de valores únicos nos últimos 10 minutos até agora que são iguais
countunique(/host/key,10m,"gt",12) #o número de valores únicos nos últimos 10 minutos até agora que são ma
countunique(/host/key,#10,"gt",12) #o número de valores únicos nos últimos 10 valores até agora que são ma
countunique(/host/key,10m:now-1d,"gt",12) #o número de valores únicos entre 24 horas e 10 minutos e 24 hor
countunique(/host/key,10m,"bitand","6/7") #o número de valores únicos nos últimos 10 minutos até agora que
countunique(/host/key,10m:now-1d) #o número de valores únicos entre 24 horas e 10 minutos e 24 horas atrás
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Encontra uma correspondência de valor dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.
 Retorna: 1 - encontrado; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **sec** ou **#num** (opcional) - usa o valor mais recente por padrão, se não for especificado
- **operator** (deve estar 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` (sensível a maiúsculas e minúsculas)
bitand - AND bit a bit
regexp - correspondência sensível a maiúsculas e minúsculas com a expressão regular fornecida em `pattern`
iregexp - correspondência insensível a maiúsculas e minúsculas com a 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 operator para valores integer.
- *like* e *bitand* não são suportados como operators para valores float.
- Para valores string, text e log, apenas os operators *eq*, *ne*, *like*, *regexp* e *iregexp* são suportados.
- Com *regexp* ou *iregexp* como operator, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@').
 No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global.

Exemplos:

```
find(/host/key,10m,"like","error") #encontra um valor que contém 'error' nos últimos 10 minutos até agora
find(/host/agent.version,,"like","beta")=1 #o Zabbix agent tem versão beta, deve ser atualizado
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #foi encontrado um erro interno do web serv
```

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().

Exemplos:

```
firstclock(/host/key,1h) #recupera o carimbo de data/hora do valor mais antigo dentro da última hora
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 server/proxy do Zabbix.
 Tipos de valor suportados: *Float*, *Integer*.
 Retorna: 1 - a diferença entre o valor do item passivo (como timestamp) e o timestamp do server/proxy do Zabbix (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 a hora local está sincronizada com a hora local do server do Zabbix. *Observe* que 'system.localtime' deve ser configurado como uma verificação passiva para o agent do Zabbix; no Zabbix agent 2, ele pode ser configurado como uma verificação ativa.
- Também pode ser usado com a chave `vfs.file.time[/path/file,modify]` para verificar se o arquivo não recebe atualizações há muito tempo.
- Esta função não é recomendada para uso em expressões complexas de trigger (com vários items envolvidos), porque pode causar resultados inesperados (a diferença de tempo será medida com a métrica mais recente), por exemplo, em `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Exemplos:

```
fuzzytime(/host/system.localtime,5m)=0 #a hora local do cliente difere da hora do server/proxy do Zabbix e
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #a hora local do cliente d
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Retorna uma matriz de resultados de consultas JSONPath.
 Tipos de valor compatíveis: *String*, *Text*, *Log*.

Parâmetros:

- Consulte os parâmetros comuns
- **pattern** - o padrão obrigatório (os argumentos de string devem estar entre aspas duplas)

Comentários:

- Esta função é usada com funções de agregação.
- Uma consulta JSONPath é executada em cada valor de histórico selecionado; cada resultado bem-sucedido e não vazio da consulta é adicionado à matriz. Se o resultado de uma consulta para um único valor de histórico contiver várias correspondências, cada correspondência será adicionada à matriz separadamente.
- A função falha com um erro se:
  - O formato de `pattern` for inválido.
  - Algum dos valores de histórico contiver JSON inválido ou a consulta JSONPath falhar para esse valor.
- A função retorna uma matriz vazia se:

- Nenhum valor de histórico tiver sido selecionado.
- Todos os valores de histórico selecionados contiverem JSON inválido.

Exemplo:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

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 hash (#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().

Exemplos:

```
last(/host/key) #recupera o último valor
last(/host/key,#2) #recupera o valor anterior
last(/host/key,#1)<>last(/host/key,#2) #os dois últimos valores são diferentes
```

lastclock(/host/key,<#num<:time shift»)

O carimbo de data/hora do Nº valor mais recente dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns;
- **#num** (opcional) - o Nº valor mais recente.

A função falha com um erro se nenhum dado tiver sido coletado no período fornecido ou se nenhum valor Nth tiver sido coletado.

Veja também firstclock().

Exemplos:

```
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
now()-lastclock(/host/heartbeat.item)>300 #mais de 5 minutos se passaram desde que o item de heartbeat foi
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #o intervalo
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #o va
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Verifica se o ID do evento da última entrada de log corresponde a uma expressão regular.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - não corresponde; 1 - corresponde.

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente
- **pattern** (opcional) - a expressão regular que descreve o padrão necessário, no estilo Perl Compatible Regular Expression (PCRE) (os argumentos de string devem estar entre aspas duplas)

Exemplos:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #uma entrada de log com ID correspondente a "4625" (auten
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #uma entrada de log com ID correspondente a "6008" ou
```

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 - Verboso).

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.

Exemplos:

```
logseverity(/host/log[/var/log/syslog],10m)>3 #uma entrada de log com severidade acima de "3" encontrada
logseverity(/host/eventlog[System],10m)=4 #uma entrada de log com severidade igual a "Erro" encontrada
```

logsource(/host/key,<#num<:time shift»,<pattern>)

Verifica se a origem do último registro de log corresponde a uma expressão regular.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - não corresponde; 1 - corresponde.

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente
- **pattern** (opcional) - a expressão regular que descreve o padrão necessário, no estilo Perl Compatible Regular Expression (PCRE) (os argumentos de string devem estar entre aspas duplas)

Normalmente usado para logs de eventos do Windows.

Exemplos:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #uma entrada de log com origem correspondente a "M
logsource(/host/eventlog[System],,"Service Control Manager")=1 #uma entrada de log com origem correspondem
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

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 N-ésimo valor tiver sido coletado;
  - a mensagem de log não contiver um valor de timestamp.

Exemplos:

```
logtimestamp(/host/key) #recupera o timestamp da mensagem de log mais recente
logtimestamp(/host/key,#2) #recupera o timestamp da 2ª mensagem de log mais recente
logtimestamp(/host/key,#3:now-1d) #recupera o timestamp da 3ª mensagem de log mais recente de um dia atrás
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se houve uma diminuição monótona nos valores.
 Tipos de valores suportados: *Integer*.
 Retorna: 1 - se todos os elementos no período de tempo diminuíram continuamente; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve estar entre aspas duplas) - *weak* (todo valor é menor ou igual ao anterior; padrão) ou *strict* (todo valor diminuiu)

Exemplos:

```
monodec(/host/system.swap.size[all,free],60s) + monodec(/host2/system.swap.size[all,free],60s) + monodec(/
monodec(/host/proc.num[nginx],10m,"strict")=1 #o número de processos nginx diminuiu monotonamente em 10 mi
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se houve um aumento monótono nos valores.
 Tipos de valores suportados: *Integer*.
 Retorna: 1 - se todos os elementos no período de tempo aumentarem continuamente; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve estar entre aspas duplas) - *weak* (todo valor é maior ou igual ao anterior; padrão) ou *strict* (todo valor aumentou)

Exemplos:

```
monoinc(/host/system.localtime,#3,"strict")=0 #o tempo local do sistema não aumentou consistentemente
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #dispara se a contagem de arquivos estagnar por
```

nodata(/host/key,sec,<mode>)

Verifica se nenhum dado foi recebido.
 Tipos de valor compatíveis: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Retorna: 1 - se nenhum dado tiver sido recebido durante o período definido; 0 - caso contrário.

Parâmetros:

- Consulte os parâmetros comuns
- **sec** - o período não deve ser inferior a 30 segundos, pois o processo history syncer calcula esta função somente a cada 30 segundos; `nodata(/host/key,0)` não é permitido
- **mode** - se definido como *strict* (entre aspas duplas), esta função não será sensível à disponibilidade do proxy (consulte os comentários para obter detalhes)

Comentários:

- por padrão, os triggers 'nodata' monitorados pelo proxy são sensíveis à disponibilidade do proxy - se o proxy ficar indisponível, os triggers 'nodata' não serão acionados imediatamente após o restabelecimento da conexão, mas ignorarão os dados referentes ao período de atraso. Observe que, para proxies passivos, a supressão será ativada se a conexão for restabelecida após mais de 15 segundos e não menos de 2 segundos depois. Para proxies ativos, a supressão será ativada se a conexão for restabelecida após mais de 15 segundos. Para desativar a sensibilidade à disponibilidade do proxy, use o terceiro parâmetro, por exemplo: `nodata(/host/key,5m,"strict")`; nesse caso, a função será acionada assim que o período de avaliação (cinco minutos) sem dados tiver transcorrido.
- Esta função exibirá um erro se, dentro do período do primeiro parâmetro:
- não houver dados e o Zabbix server tiver sido reiniciado
- não houver dados e a manutenção tiver sido concluída
- não houver dados e o item tiver sido adicionado ou reativado.
- Os erros são exibidos na coluna *Info* da lista de triggers.
- Esta função pode não funcionar corretamente se houver diferenças de horário entre o Zabbix server, o proxy e o agent. Consulte também: Requisito de sincronização de horário.

Exemplo:

```
nodata(/host/agent.ping,5m)=1 #trigger se nenhum dado tiver sido recebido do Zabbix agent por 5 minutos
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

O percentil P de um período, onde P (percentual) é especificado pelo terceiro parâmetro.
 Tipos de valor suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns
- **percentage** - um número de ponto flutuante entre 0 e 100 (inclusive) com até 4 dígitos após o ponto decimal

Exemplos:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #o percentil 95 da entrada de rede (bytes/seg) em 1
percentile(/host/system.cpu.util,5m,95)>80 #o percentil 95 da porcentagem de utilização da CPU no tempo de
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #a maioria das medições de latência está abaixo de
percentile(/host/net.if.in[eth0,bytes],1h,50) #calcula o percentil 50 (o valor mediano) do tráfego de rede
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #calcula o
```

rate(/host/key,sec<:time shift>)

O aumento médio por segundo em um contador monotonicamente crescente dentro do período de tempo definido.
 Tipos de valores suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns

Corresponde funcionalmente ao 'rate' do PromQL.

Exemplos:

```
rate(/host/key,30s) #se o aumento monotônico em 30 segundos for 20, esta função retorna 0,67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #a taxa de tráfego de entrada na interface eth0 excedeu 500 KB
rate(/host/app.requests.count,1m)>100 #o contador de requisições aumentou em >100 requisições por segundo
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Retorna uma matriz de resultados de consultas XPath XML.
 Tipos de valor compatíveis: *String*, *Text*, *Log*.

Parâmetros:

- Consulte os parâmetros comuns
- **pattern** - o padrão obrigatório (os argumentos de string devem estar entre aspas duplas)

Comentários:

- Esta função é usada com funções de agregação.
- Uma consulta XPath XML é executada em cada valor de histórico selecionado; cada resultado bem-sucedido e não vazio é adicionado à matriz. Se o resultado da consulta para um único valor de histórico contiver várias correspondências, cada correspondência será adicionada à matriz separadamente.
- A função falha com um erro se:
  - O formato de `pattern` for inválido.
  - O Zabbix tiver sido compilado sem suporte a XML.
  - Algum dos valores de histórico contiver XML inválido ou a consulta XPath falhar para esse valor.
- A função retornará uma matriz vazia se:
  - Nenhum valor de histórico tiver sido selecionado.
  - Todos os valores de histórico selecionados contiverem XML inválido.

Exemplo:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

Veja todas as funções suportadas.

Funções de trend

As funções de trend, em contraste com as funções de history, usam dados de trend para os cálculos.

Trends armazenam valores agregados por hora. As funções de trend usam essas médias horárias e, portanto, são úteis para análises de longo prazo.

Os resultados das funções de trend são armazenados em cache, de modo que várias chamadas para a mesma função com os mesmos parâmetros buscam informações no banco de dados apenas uma vez. O cache de funções de trend é controlado pelo parâmetro de servidor TrendFunctionCacheSize.

Use `zabbix[tcache,cache,paccessed]` para estimar quanto do cache está realmente em uso. Um `paccessed` baixo, junto com `pitems` baixo e `pmisses` baixo, geralmente significa que o tamanho do cache pode ser reduzido.

Triggers que fazem referência **apenas** a funções de trend 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 trend e de history (ou data e hora e/ou nodata()), ele é 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.

Function	Description
baselinedev	Retorna o número de desvios (pelo algoritmo stddevpop) entre o último período de dados e os mesmos períodos de dados nas temporadas anteriores.
baselinewma	Calcula a linha de base fazendo a 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 trend dentro do período de tempo definido.

Function	Description
trendcount	O número de valores de history recuperados com sucesso usados para calcular o valor de trend dentro do período de tempo definido.
trendmax	O valor máximo entre os valores de trend dentro do período de tempo definido.
trendmin	O valor mínimo entre os valores de trend 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 `((the number of anomaly values)/(total number of values))`.
trendsum	A soma dos valores de trend dentro do período de tempo definido.

Parâmetros comuns

- `/host/key` é um parâmetro obrigatório comum como primeiro parâmetro
- `período de tempo:deslocamento de tempo` é um parâmetro comum como segundo parâmetro, onde:
  - **período de tempo** - o período de tempo (mínimo '1h'), definido como <N><unidade de tempo> onde `N` - o número de unidades de tempo, `unidade de tempo` - h (hora), d (dia), w (semana), M (mês) ou y (ano).
  - **deslocamento de tempo** - o deslocamento do período de tempo (veja exemplos de funções)

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função
- Os parâmetros `/host/key` e `período de tempo:deslocamento de tempo` nunca devem ser colocados entre aspas

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Retorna o número de desvios (pelo algoritmo stddevpop) entre o último período de dados e os mesmos períodos de dados nas temporadas anteriores.

Parâmetros:

- Veja parâmetros comuns
- **data period** - o período de coleta de dados dentro de uma temporada, definido como <N><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 unidade de calendário que define uma temporada (h, d, w, M, y); não pode ser menor que o período de dados
- **num seasons** - o número de temporadas a serem avaliadas

Exemplos:

```
baselinedev(/host/key,1d:now/d,"M",6) #calculando o número de desvios padrão (população) entre o dia anter
baselinedev(/host/key,1h:now/h,"d",10)  #calculando o número de desvios padrão da população entre a hora a
```

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 ('seasons') usando o algoritmo de média móvel ponderada.

Parâmetros:

- Veja parâmetros comuns
- **data period** - o período de coleta de dados dentro de uma season, definido como <N><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 season
Time shift - o deslocamento do período de tempo, define o final do período de coleta de dados em seasons (veja exemplos)
- **season unit** - a unidade de calendário que define uma season (h, d, w, M, y); não pode ser menor que o data period
- **num seasons** - o número de seasons a serem avaliadas

Exemplos:

```
baselinewma(/host/key,1h:now/h,"d",3) #calculando a linha de base da mesma hora do dia nos últimos 3 dias
baselinewma(/host/key,2h:now/h,"d",3) #calculando a linha de base do mesmo período de duas horas nos últim
baselinewma(/host/key,1d:now/d,"M",4) #calculando a linha de base do mesmo dia do mês que 'ontem' nos 4 me
```

trendavg(/host/key,time period:time shift)

A média dos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendavg(/host/key,1h:now/h) #a média da hora anterior (por exemplo, 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #a média de duas horas atrás (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #a média de três horas atrás (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #a média do mês anterior há um ano
```

trendcount(/host/key,time period:time shift)

O número de valores de histórico recuperados com sucesso usados para calcular o valor de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendcount(/host/key,1h:now/h) #a contagem de valores para a hora anterior (por exemplo, 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #a contagem de valores para duas horas atrás (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #a contagem de valores para três horas atrás (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #a contagem de valores para o mês anterior há um ano
```

trendmax(/host/key,time period:time shift)

O máximo nos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendmax(/host/key,1h:now/h) #o máximo para a hora anterior (por exemplo, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcula a diferença entre os valores máximo
trendmax(/host/key,1h:now/h-1h) #o máximo para duas horas atrás (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #o máximo para três horas atrás (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #o máximo para o mês anterior há um ano
```

trendmin(/host/key,time period:time shift)

O mínimo nos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendmin(/host/key,1h:now/h) #o mínimo para a hora anterior (por exemplo, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcula a diferença entre os valores máximo
trendmin(/host/key,1h:now/h-1h) #o mínimo para duas horas atrás (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #o mínimo para três horas atrás (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #o mínimo para o mês anterior há um ano
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Retorna a taxa de anomalias durante o período de detecção - um valor decimal entre 0 e 1 que é `((o número de valores anômalos)/(número total de valores))`.

Parâmetros:

- Veja parâmetros comuns
- **eval period** - o período de tempo que deve ser decomposto (mínimo '1h'), definido como <N><unidade de tempo> onde
`N` - o número de unidades de tempo
`unidade de tempo` - h (hora), d (dia), w (semana), M (mês) ou y (ano)
- **detection period** - o período de tempo antes do final do eval period para o qual as anomalias são calculadas (mínimo '1h', não pode ser maior que 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 (”estação”) 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
```

Veja todas as funções suportadas.

Funções matemáticas

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções matemáticas são suportadas com tipos de valor float e integer, salvo indicação em contrário.

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Function	Description
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	A arcotangente de um valor como um ângulo, expresso em radianos.
atan2	A arcotangente 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 maior ou igual mais próximo.
cos	O cosseno de um valor, em que o valor é um ângulo expresso em radianos.
cosh	O cosseno hiperbólico de um valor.
cot	A cotangente de um valor, em que 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 menor ou igual mais próximo.
log	O logaritmo natural.
log10	O logaritmo decimal.
max	O maior valor dos valores do item referenciado.

Function	Description
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.
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, em que o valor é um ângulo expresso em radianos.
sinh	O seno hiperbólico de um valor, em que 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 opcionais da função (ou partes do parâmetro) são indicados por < >

abs(value)

O valor absoluto (a partir de 0) de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o valor absoluto de '3' ou '-3' será '3'.

Exemplo:

```
abs(last(/host/key))>10
```

acos(value)

O arco cosseno de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar entre -1 e 1. Por exemplo, o arco cosseno de um valor '0.5' será '2.0943951'.

Exemplo:

```
acos(last(/host/key))
```

asin(value)

O arco seno de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar entre -1 e 1. Por exemplo, o arco seno de um valor '0.5' será '-0.523598776'.

Exemplo:

```
asin(last(/host/key))
```

atan(value)

O arco-tangente de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o arco-tangente de um valor ’1’ será ’0.785398163’.

Exemplo:

```
atan(last(/host/key))
```

atan2(value,abscissa)

O arco-tangente das coordenadas de ordenada (valor) e abscissa especificadas como um ângulo, expresso em radianos.

Parâmetros:

- **value** - o valor a ser verificado;
- **abscissa** - o valor da abscissa.

Por exemplo, o arco-tangente das coordenadas de ordenada e abscissa de um valor ’1’ será ’2.21429744’.

Exemplo:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

O valor médio dos valores dos items referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

A raiz cúbica de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, a raiz cúbica de ’64’ será ’4’, de ’63’ será ’3.97905721’.

Exemplo:

```
cbrt(last(/host/key))
```

ceil(value)

Arredonda o valor para o inteiro mais próximo, maior ou igual.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, ’2.4’ será arredondado para ’3’. Veja também floor().

Exemplo:

```
ceil(last(/host/key))
```

cos(value)

O cosseno de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cosseno de um valor ’1’ será ’0.54030230586’.

Exemplo:

```
cos(last(/host/key))
```

cosh(value)

O cosseno hiperbólico de um valor. Retorna o valor como um número real, não como notação científica.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cosseno hiperbólico de um valor ’1’ será ’1.54308063482’.

Exemplo:

```
cosh(last(/host/key))
```

cot(value)

O cotangente de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cotangente de um valor ’1’ será ’0.54030230586’.

Exemplo:

```
cot(last(/host/key))
```

degrees(value)

Converte um valor de radianos para graus.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor ’1’ convertido para graus será ’57.2957795’.

Exemplo:

```
degrees(last(/host/key))
```

e

O número de Euler (2.718281828459045).

Exemplo:

```
e()
```

exp(value)

O número de Euler elevado à potência de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o número de Euler elevado à potência de um valor ’2’ será ’7.38905609893065’.

Exemplo:

```
exp(last(/host/key))
```

expm1(value)

O número de Euler elevado a um valor menos 1.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o número de Euler elevado ao valor ’2’ menos 1 será ’6.38905609893065’.

Exemplo:

```
expm1(last(/host/key))
```

floor(value)

Arredonda o valor para baixo para o inteiro menor ou igual mais próximo.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, ’2.6’ será arredondado para baixo para ’2’. Veja também ceil().

Exemplo:

```
floor(last(/host/key))
```

log(value)

O logaritmo natural.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o logaritmo natural de um valor ’2’ será ’0.69314718055994529’.

Exemplo:

```
log(last(/host/key))
```

log10(value)

O logaritmo decimal.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o logaritmo decimal de um valor ’5’ será ’0.69897000433’.

Exemplo:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

O maior valor dos valores de item referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<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,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

O resto da divisão.

Parâmetros:

- **value** - o valor a ser verificado;
- **denominator** - o denominador da divisão.

Por exemplo, o resto da divisão de um valor ’5’ com o denominador 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 3ª potência de um valor ’2’ será ’8’.

Exemplo:

```
power(last(/host/key),3)
```

radians(value)

Converte um valor de graus para radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor ’1’ convertido para radianos será ’0.0174532925’.

Exemplo:

```
radians(last(/host/key))
```

rand

Retorna um valor inteiro aleatório. Um número pseudo-aleatório gerado usando o tempo como semente (suficiente para fins matemáticos, mas não para criptografia).

Exemplo:

```
rand()
```

round(value,decimal places)

Arredonda o valor para casas decimais.

Parâmetros:

- **value** - o valor a ser verificado;
- **decimal places** - especifica as casas decimais para arredondamento (0 também é possível).

Por exemplo, um valor ’2.5482’ arredondado para 2 casas decimais será ’2.55’.

Exemplo:

```
round(last(/host/key),2)
```

signum(value)

Retorna ’-1’ se o valor for negativo, ’0’ se o valor for zero, ’1’ se o valor for positivo.

Parâmetro:

- **value** - o valor a ser verificado.

Exemplo:

```
signum(last(/host/key))
```

sin(value)

O seno de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o seno de um valor ’1’ será ’0.8414709848’.

Exemplo:

```
sin(last(/host/key))
```

sinh(value)

O seno hiperbólico de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o seno hiperbólico de um valor ’1’ será ’1.17520119364’.

Exemplo:

```
sinh(last(/host/key))
```

sqrt(value)

A raiz quadrada de um valor.
 Esta função 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,1h),avg(/host2/key2,1h))
```

tan(value)

A tangente de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, a tangente de um valor ’1’ será ’1.55740772465’.

Exemplo:

```
tan(last(/host/key))
```

truncate(value,decimal places)

Trunca o valor para casas decimais.

Parâmetros:

- **value** - o valor a ser verificado;
- **decimal places** - especifica as casas decimais para truncar (0 também é possível).

Por exemplo, um valor ’2.5482’ truncado para 2 casas decimais será ’2.54’.

Exemplo:

```
truncate(last(/host/key),2)
```

Veja todas as funções suportadas.

Funções de operador

Todas as funções listadas aqui são compatíveis com:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver todos os detalhes.

Function	Description
between	Verifica se o valor pertence ao intervalo especificado.
in	Verifica se o valor é igual a pelo menos um dos valores listados.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula

- Expressões são aceitas como parâmetros

between(value,min,max)

Verifica se o valor pertence ao intervalo fornecido.
 Tipos de valores suportados: *Integer*, *Float*.
 Retorna: 1 - dentro do intervalo; 0 - caso contrário.

Parâmetros:

- **value** - o valor a ser verificado;
- **min** - o valor mínimo;
- **max** - o valor máximo.

Exemplo:

```
between(last(/host/key),1,10)=1 #trigger se o valor estiver entre 1 e 10
```

in(value,value1,value2,...valueN)

Verifica se o valor é igual a pelo menos um dos valores listados.
 Tipos de valores suportados: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Retorna: 1 - se igual; 0 - caso contrário.

Parâmetros:

- **value** - o valor a ser verificado;
- **valueX** - valores listados (valores de string devem estar entre aspas duplas).

O valor é comparado aos valores listados como números, se todos esses valores puderem ser convertidos para numéricos; caso contrário, são comparados como strings.

Exemplo:

```
in(last(/host/key),5,10)=1 #trigger se o último valor for igual a 5 ou 10
in("text",last(/host/key),last(/host/key,#2))=1 #trigger se "text" for igual a qualquer um dos 2 últimos v
```

Veja todas as funções suportadas.

Funções preditivas

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Function	Description
forecast	O valor futuro, máximo, mínimo, delta ou média do item.
timeleft	O tempo, em segundos, necessário para um item atingir o limite especificado.

Parâmetros comuns

- `/host/key` é um primeiro parâmetro obrigatório comum para as funções que fazem referência ao histórico do item do host.
- `(sec|#num)<:time shift>` é um segundo parâmetro comum para as funções que fazem referência ao histórico do item do host, em que:
  - **sec** - período máximo de avaliação em segundos (podem ser usados sufixos de tempo), ou
  - **#num** - intervalo máximo de avaliação nos valores coletados mais recentes (se precedido por um símbolo de hash).
  - **time shift** (opcional) - permite mover o ponto de avaliação para trás no tempo. Consulte mais detalhes sobre como especificar o deslocamento de tempo.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por uma vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função
- Os parâmetros `/host/key` e `(sec|#num)<:deslocamento de tempo>` nunca devem ser colocados entre aspas

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

O valor futuro, máximo, mínimo, delta ou médio do item.
 Tipos de valor compatíveis: *Float*, *Integer*.

Parâmetros:

- Consulte os parâmetros comuns;
- **time** - o horizonte da previsão em segundos (por exemplo, 1, 60) (sufixos de tempo podem ser usados (por exemplo, 30s, 1m)); valores negativos são compatíveis;
- **fit** (opcional; deve estar entre aspas duplas) - a função usada para ajustar os dados históricos. Ajustes compatíveis:
linear - função linear (padrão)
polynomialN - polinômio 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 estar entre aspas duplas) - a saída solicitada. Modos compatíveis:
value - valor (padrão)
max - máximo
min - mínimo
delta - *max*-*min*
avg - média
Observe que *value* estima o valor do item no momento `now` + `time`; *max*, *min*, *delta* e *avg* analisam a estimativa do valor do item no intervalo entre `now` e `now` + `time`.

Comentários:

- Se o valor a ser retornado for maior que 1.7976931348623158E+308 ou menor que -1.7976931348623158E+308, o valor retornado será limitado a 1.7976931348623158E+308 ou -1.7976931348623158E+308, respectivamente;
- Torna-se incompatível somente se for usado incorretamente na expressão (tipo de item incorreto, parâmetros inválidos); caso contrário, retorna -1 em caso de erros;
- Consulte também informações adicionais sobre funções de trigger preditivas.

Exemplos:

```
forecast(/host/key,#10,1h) #prever o valor do item em uma hora com base nos últimos 10 valores
forecast(/host/key,1h,30m) #prever o valor do item em 30 minutos com base nos dados da última hora
forecast(/host/key,1h:now-1d,12h) #prever o valor do item em 12 horas com base em uma hora de dados de um
forecast(/host/key,1h,10m,"exponential") #prever o valor do item em 10 minutos com base nos dados da últim
forecast(/host/key,1h,2h,"polynomial3","max") #prever o valor máximo que o item pode atingir nas próximas
forecast(/host/key,#2,-20m) #estimar o valor do item 20 minutos atrás com base nos dois últimos valores (i
```

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 compatíveis: *Float*, *Integer*.

Parâmetros:

- Consulte os parâmetros comuns;
- **threshold** - o valor a ser atingido (podem ser usados sufixos de unidade);
- **fit** (opcional; deve estar entre aspas duplas) - consulte forecast().

Comentários:

- Se o valor a ser retornado for maior que 1.7976931348623158E+308, o valor retornado será limitado a 1.7976931348623158E+308;
- Retorna 1.7976931348623158E+308 se o limite não puder ser atingido;
- Torna-se sem suporte somente quando usado incorretamente na expressão (tipo de item incorreto, parâmetros inválidos); caso contrário, retorna -1 em caso de erros;
- Consulte também informações adicionais sobre funções de trigger preditivas.

Exemplos:

```
timeleft(/host/key,#10,0) #o tempo até que o valor do item atinja zero com base nos últimos 10 valores
timeleft(/host/key,1h,100) #o tempo até que o valor do item atinja 100 com base nos dados da última hora
timeleft(/host/key,1h:now-1d,100) #o tempo até que o valor do item atinja 100 com base em uma hora de dado
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.

Funções preditivas de trigger

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 implementadas por meio de funções preditivas de trigger.

Funções

Antes de definir um trigger, é necessário definir o que é um estado de problema e quanto tempo é necessário para agir. Em seguida, existem duas maneiras de configurar um trigger que sinalize 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

Primeiramente, você deve especificar o período histórico que o Zabbix deve analisar para gerar a previsão. Você faz isso de uma maneira familiar, por meio do parâmetro `período de tempo` e de um deslocamento de tempo opcional, assim como faz com as funções **avg**, **count**, **delta**, **max**, **min** e **sum**.

Horizonte de previsão

(**forecast** apenas)
 O parâmetro `time` especifica até onde no futuro o Zabbix deve extrapolar as dependências encontradas nos dados históricos. Não importa se você usa `time_shift` ou não, `time` é sempre contado a partir do momento atual.

Limite a ser atingido

(**timeleft** apenas)
 O parâmetro `threshold` especifica um valor que o item analisado deve atingir, não importa se de cima para baixo ou de baixo para cima. Uma vez que determinamos f(t) (veja abaixo), devemos resolver a equação f(t) = `threshold` e retornar a raiz que está mais próxima de agora e à direita de agora 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 deve ser usar previsões como complemento ao diagnóstico de problemas comum, não como substituição.[1]

Funções de ajuste

O `fit` padrão é a função *linear*. Mas se o seu sistema monitorado for mais complicado, você tem mais opções para escolher.

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$
exponential	$x = a*exp(b*t)$
logarithmic	$x = a + b*log(t)$
power	$x = a*t^b$

Modos

(**forecast** apenas)
 Toda vez que uma função de trigger é avaliada, ela obtém dados do período de histórico especificado e ajusta uma função especificada aos dados. Assim, se os dados forem ligeiramente diferentes, a função ajustada será ligeiramente diferente. Se simplesmente calcularmos o valor da função ajustada em um momento especificado no futuro, você não saberá nada sobre como o item analisado deve se comportar entre agora e aquele momento no futuro. Para algumas opções de `fit` (como *polynomial*) um valor simples do futuro pode ser enganoso.

`mode`	**forecast** resultado
value	f(agora + `time`)
max	$max_{agora <= t <= agora + time}$ f(t)
min	$min_{agora <= t <= agora + time}$ f(t)
delta	*max - min*
avg	média de f(t) (agora <= t <= agora + `time`) de acordo com a definição

Detalhes

Para evitar cálculos com números muito grandes, consideramos o timestamp do primeiro valor no período especificado mais 1 ns como um novo tempo zero (o tempo de época atual está na ordem de 10^9, a época ao quadrado é 10^{18}, a precisão dupla é cerca de 10^{-16}). 1 ns é adicionado para fornecer todos os valores de tempo positivos para os ajustes *logarítmico* e *potência*, que envolvem

o cálculo de log(t). O deslocamento de tempo não afeta os ajustes *linear*, *polinomial*, *exponencial* (além de cálculos mais fáceis e precisos), mas altera a forma das funções *logarítmica* e *potência*.

Possíveis erros

As funções retornam -1 nas seguintes situações:

- o período de avaliação especificado não contém dados;
- o resultado da operação matemática não está definido[3];
- complicações numéricas (infelizmente, para alguns conjuntos de dados de entrada, o intervalo e a precisão do formato de ponto flutuante de dupla precisão tornam-se insuficientes)[4].

Note:
Nenhum aviso ou erro é sinalizado se o ajuste escolhido descrever mal os dados fornecidos ou se houver poucos dados para uma previsão precisa.

Exemplos e tratamento de erros

Para receber um aviso quando você estiver prestes a ficar sem espaço livre em disco no seu host, você pode usar uma expressão de trigger como esta:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

No entanto, o código de erro -1 pode entrar em ação e colocar seu trigger em 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 complicada [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

A situação é um pouco mais difícil com **forecast**. Antes de tudo, -1 pode ou não colocar o trigger em estado de problema, dependendo de você ter uma expressão como `forecast(/host/item,(...))<...` ou como `forecast(/host/item,(...))>...`

Além disso, -1 pode ser uma previsão válida se for normal para o valor do item ser negativo. Mas a probabilidade dessa situação no mundo real é insignificante (veja como o operador **=** funciona). Portanto, adicione `... or forecast(/host/item,(...))=-1` ou `... and forecast(/host/item,(...))<>-1` se você quiser ou não quiser tratar -1 como um problema, respectivamente.

Notas de rodapé

[1] Por exemplo, um trigger simples como `timeleft(/host/item,1h,X) < 1h` pode entrar em estado de problema quando o valor do item se aproxima de X e, em seguida, se recuperar repentinamente assim que o valor X for atingido. Se o problema for o valor do item estar abaixo de X, use: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Se o problema for o valor do item estar acima de X, use: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] O grau do polinômio pode ser de 1 a 6, *polynomial1* é equivalente a *linear*. No entanto, use polinômios de grau mais alto com cautela. Se o período de avaliação contiver menos pontos do que o necessário para determinar os coeficientes do polinômio, o grau do polinômio será reduzido (por exemplo, *polynomial5* é solicitado, mas há apenas 4 pontos, portanto *polynomial3* será ajustado).

[3] Por exemplo, ajustar funções *exponential* ou *power* envolve calcular log() dos valores do item. Se os dados contiverem zeros ou números negativos, você receberá um erro, pois log() é definido apenas para valores positivos.

[4] Para ajustes *linear*, *exponential*, *logarithmic* e *power* todos os cálculos necessários podem ser escritos explicitamente. Para *polynomial* apenas *value* pode ser calculado sem etapas adicionais. Calcular *avg* envolve computar a antiderivada do polinômio (analiticamente). Calcular *max*, *min* e *delta* envolve computar a derivada do polinômio (analiticamente) e encontrar suas raízes (numericamente). Resolver f(t) = 0 envolve encontrar as raízes do polinômio (numericamente).

[5] Mas, neste caso, -1 pode fazer com que seu trigger se recupere do estado de problema. Para estar totalmente protegido, use: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free], or {TRIGGER.VALUE}=1)`

Funções de string

Todas as funções listadas aqui são compatíveis com:

- Expressões de trigger
- Fórmulas de item calculado
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver todos os detalhes.

Função	Descrição
ascii	O código ASCII do caractere mais à esquerda do valor.
bitlength	O comprimento do valor em bits.
bytelength	O comprimento do valor em bytes.
char	Retorna o caractere interpretando o valor como um código ASCII.
concat	A string resultante da concatenação dos valores dos itens referenciados ou dos valores constantes.
contains	Retorna 1 se o valor contiver a substring especificada por `search value`; caso contrário, retorna 0.
insert	Insere os caracteres ou espaços especificados na string de caracteres, começando na posição especificada da string.
jsonpath	Retorna o resultado de 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 de caractere especificada por 'start'. Esta função é um alias da função `substring()`.
repeat	Repete uma string.
replace	Localiza o padrão no valor e o substitui pela substituição.
right	Retorna os caracteres mais à direita do valor.
rtrim	Remove os caracteres especificados do final da string.
substring	Retorna uma substring de N caracteres começando na posição de caractere especificada por 'start'.
trim	Remove os caracteres especificados do início e do final da string.
xmlxpath	Retorna o resultado de 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 opcionais da função (ou partes do parâmetro) são indicados por < >

ascii(value)

O código ASCII do caractere mais à esquerda do valor.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor como 'Abc' retornará '65' (código ASCII para 'A').

Exemplo:

```
ascii(last(/host/key))
```

bitlength(value)

O comprimento do valor em bits.
 Tipos de 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 valores suportados: *String*, *Text*, *Log*, *Integer*.

Parâmetro:

- **value** - o valor a ser verificado

Exemplo:

`bytelength(last(/host/key))`

char(value)

Retorna o caractere interpretando o valor como código ASCII.
 Tipos de valor suportados: *Integer*.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar no intervalo de 0-255. Por exemplo, um valor como '65' (interpretado como código ASCII) retornará 'A'.

Exemplo:

`char(last(/host/key))`

concat(<valor1>,<valor2>,...)

A string resultante da concatenação dos valores dos items referenciados ou valores constantes.
 Tipos de valores suportados: *String*, *Text*, *Log*, *Float*, *Integer*.

Parâmetro:

- **valorX** - o valor retornado por uma das funções de histórico ou um valor constante (string, inteiro ou número 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])
```

contains(value, search value)

Retorna 1 se o valor contiver a substring especificada por `search value`; caso contrário, retorna 0.| Tipos de valor compatíveis: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **search value** - o valor a ser pesquisado.

Comentários:

- A pesquisa diferencia maiúsculas de minúsculas.
- Quando o valor é válido e `search value` está vazio, a função retorna 1.

Por exemplo, a função retorna 1 se o valor contiver 'Zabbix'.

Exemplo:

`contains(last(/host/key),"Zabbix")`

insert(value,start,length,replacement)

Insere caracteres ou espaços especificados na string de caracteres começando na posição especificada na string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **start** - posição inicial;
- **length** - posições a serem substituídas;
- **replacement** - string de substituição.

Por exemplo, um valor como 'Zabbbix' será substituído por 'Zabbix' se 'bb' (posição inicial 3, posições a 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 estar entre aspas);
- **default** - o valor opcional de fallback se a consulta JSONPath não retornar dados. Observe que em outros erros, a falha é retornada (por exemplo, ”unsupported construct”).

Exemplo:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Retorna os caracteres mais à esquerda do valor.
 Tipos de valores suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de caracteres a serem retornados.

Por exemplo, você pode retornar ’Zab’ de ’Zabbix’ especificando 3 caracteres mais à esquerda para retornar. Veja também right().

Exemplo:

```
left(last(/host/key),3) #retorna três caracteres mais à esquerda
```

length(value)

O comprimento do valor em caracteres.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetro:

- **value** - o valor a ser verificado.

Exemplos:

```
length(last(/host/key)) #o comprimento do valor mais recente
length(last(/host/key,#3)) #o comprimento do terceiro valor mais recente
length(last(/host/key,#1:now-1d)) #o comprimento do valor mais recente 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’.
Esta função é um alias da função `substring()`.
 Tipos de valor compatíveis: *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 o número de posições a serem retornadas for 4.

Exemplo:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Repete uma string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de vezes a repetir.

Exemplo:

```
repeat(last(/host/key),2) #repete o valor duas vezes
```

replace(value,pattern,replacement)

Procura o padrão no valor e substitui pelo valor de substituição. Todas as ocorrências do padrão serão substituídas.
 Tipos de 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 valores suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de caracteres a serem retornados.

Por exemplo, você pode retornar 'bix' de 'Zabbix' especificando 3 caracteres mais à direita para retornar. Veja também left().

Exemplo:

```
right(last(/host/key),3) #retorna três caracteres mais à direita
```

rtrim(value,<chars>)

Remove os caracteres especificados do final da string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **chars** (opcional) - especifica os caracteres a serem removidos.

Espaços em branco são removidos à direita por padrão (se nenhum caractere opcional for especificado). Veja também: ltrim(), trim().

Exemplos:

```
rtrim(last(/host/key)) #remove espaços em branco do final da string
rtrim(last(/host/key),"x") #remove qualquer 'x' do final da string
rtrim(last(/host/key),"x ") #remove qualquer 'x' e espaço do final da string
```

substring(value,start,length)

Retorna uma substring de N caracteres, começando na posição de caractere especificada por 'start'.
 Tipos de valor compatíveis: *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 o número de posições a serem retornadas for 4.

Exemplo:

```
substring(last(/host/key),2,4)="abbi"
```

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.

Por padrão, os espaços em branco são removidos de ambos os lados (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, é retornada falha (por exemplo, ”invalid expression”).

Exemplo:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Veja todas as funções suportadas.

Melhores práticas para triggers

Triggers são uma ferramenta poderosa, mas também podem gerar ruído de alertas indesejado. Para ver mais sinais reais e menos ruído, siga estas dicas:

1. Dessensibilize triggers. Em vez de alertar com base no último valor (muito alto/muito baixo), analise a média de um período mais longo usando funções como **avg**, **min** e **max**.
2. Considere usar a função **percentile** (defina-a para 95% ou 5%) em triggers, se você quiser evitar alertas causados por picos e quedas aleatórios.
3. Use hysteresis para evitar o flapping de trigger — mudanças frequentes do estado do trigger (OK ↔ Problem). Um continuum para o estado de problema pode ser definido adicionando uma expressão de recuperação (uma condição separada para a resolução do problema).
4. Use dependencies de trigger para evitar alertas que não estejam relacionados à causa raiz.
5. Use severity de trigger para alertar apenas sobre os problemas mais graves.
6. Defina janelas de maintenance.

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 uma *Expressão de recuperação* ao configurar um trigger.

Neste caso, a recuperação do problema ocorre em duas etapas:

- Primeiro, a expressão de problema (temperatura acima de 20°C) deve ser avaliada como FALSA
- Segundo, a expressão de recuperação (temperatura abaixo de 15°C) deve ser avaliada como VERDADEIRA

A expressão de recuperação é avaliada somente após o evento de problema ser resolvido. A expressão de recuperação ser VERDADEIRA sozinha não resolve um problema se a expressão de problema ainda for VERDADEIRA!

Exemplo

A temperatura na sala do servidor está muito alta.

Expressão do problema:

```
last(/server/temp)>20
```

Expressão de recuperação:

```
last(/server/temp)<=15
```

> **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" (o que indica um estado de "Problema") ao avaliar a expressão.

Dependências de trigger

As dependências de trigger podem ser usadas para evitar alertas que não estão relacionados à causa raiz.

Veja todas as melhores práticas.

Visão geral

Às vezes, a disponibilidade de um host depende de outro. Um server que está atrás de um roteador ficará inacessível se o roteador parar de funcionar. Com triggers configurados para ambos, você poderá receber notificações sobre dois hosts indisponíveis, embora apenas o roteador seja o responsável.

É nesse caso que alguma dependência entre hosts pode ser útil. Quando uma dependência é configurada, as notificações dos dependentes podem ser retidas, enviando-se apenas a notificação sobre o problema raiz.

Embora o Zabbix não ofereça suporte direto a dependências entre hosts, elas podem ser definidas por meio de um método mais flexível: dependências entre triggers. Um trigger pode depender de um ou mais triggers.

Em nosso exemplo simples, abrimos o formulário de configuração do trigger do server e definimos que ele depende do trigger correspondente do roteador. Com essa dependência, o trigger do server não mudará de estado enquanto o trigger do qual depende estiver no estado 'PROBLEM' e, portanto, nenhuma ação dependente será executada e nenhuma notificação será enviada.

Se tanto o server quanto o roteador estiverem indisponíveis e for definida uma dependência entre o trigger "server indisponível" e o trigger "roteador indisponível", o Zabbix não executará ações para o trigger dependente.

Enquanto o trigger pai estiver no estado PROBLEM, seus dependentes poderão informar valores que não são confiáveis. Portanto, os triggers dependentes não serão reavaliados até que o trigger pai (o roteador no exemplo acima):

- volte do estado 'PROBLEM' para o estado '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 pelo valor de um item que não esteja envolvido no trigger dependente;
- seja desabilitado, tenha um item desabilitado ou um host com item desabilitado

Em todos os casos mencionados acima, o trigger dependente (do server) será reavaliado somente quando uma nova métrica for recebida para ele. Isso significa que o trigger dependente pode não ser atualizado imediatamente.

Além disso:

- Uma dependência de trigger pode ser adicionada de um trigger de qualquer host para qualquer outro trigger de host, desde que isso não resulte em uma dependência circular.
- Uma dependência de trigger pode ser adicionada de um template para outro. Se algum trigger do template A depender de algum trigger do template B, o template A só poderá ser vinculado a um host (ou a outro template) junto com o template B, mas o template B poderá ser vinculado sozinho a um host (ou a outro template).
- Uma 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 dependerá do mesmo trigger de template do qual o trigger original dependia. Isso permite, por exemplo, ter um template no qual alguns triggers dependem dos triggers do roteador (host). Todos os hosts vinculados a esse template dependerão desse roteador específico.
- Uma dependência de trigger não pode ser adicionada de um trigger de host para um trigger de template.
- Uma 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 de LLD diferente nem de um trigger criado a partir de um protótipo de trigger. Um protótipo de trigger de host não pode depender de um trigger de um template.

Configuração

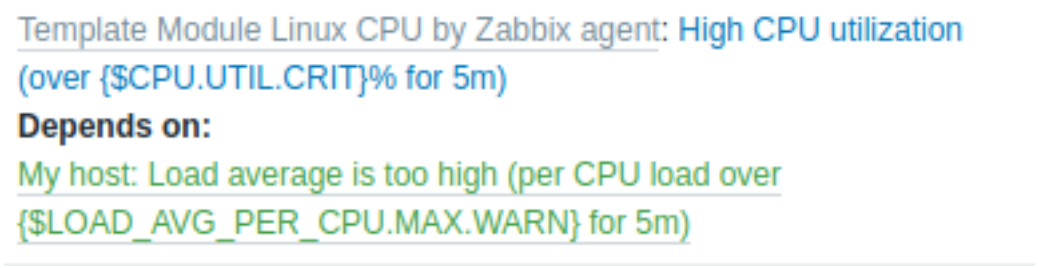
Para definir uma dependência:

1. Abra a aba *Dependências* no formulário de configuração do trigger.
2. Clique em *Adicionar* na seção *Dependências* e selecione um ou mais triggers dos quais o trigger dependerá.



3. Clique em *Atualizar*.

Agora, o trigger apresenta 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 indisponível, então, obviamente, o host e o Router2 também estarão inacessíveis, mas receber três notificações sobre o host, o Router1 e o Router2 estarem todos indisponíveis é excessivo.

Portanto, neste caso, definimos duas dependências:

```
o trigger 'Host is down' depende do trigger 'Router2 is down'
o trigger 'Router2 is down' depende do trigger 'Router1 is down'
```

Antes de alterar o status do trigger 'Host is down', o Zabbix verificará as dependências correspondentes do trigger. Se forem encontradas e um desses triggers estiver no estado **Problem**, então o estado do trigger não será alterado, as ações não serão executadas e nenhuma notificação será enviada.

O Zabbix executa essa verificação recursivamente. Se o Router1 ou o Router2 estiver inacessível, o trigger do host não será atualizado.

Gravidade do trigger

A gravidade do trigger pode ser usada para evitar ruídos de alerta em problemas menos graves.

Veja todas as melhores práticas.

Visão geral

A gravidade representa a importância relativa de um trigger.



A atribuição de gravidade não é útil apenas para a representação visual, mas também para evitar notificações em problemas triviais.

As gravidades de trigger são usadas para:

- Representação visual (cor distinta para cada problema criado, com base na gravidade original do trigger)
- Limitar notificações com base na gravidade (veja condições de ação)
- Mídia do usuário - diferentes mídias (canal de notificação) para diferentes gravidades (por exemplo, SMS para triggers de gravidade *Alta* e *Desastre*, e E-mail para triggers de outras gravidades)
- Áudio em alarmes globais - diferentes áudios para diferentes gravidades

As seguintes gravidades de trigger são suportadas por padrão:

Gravidade	Cor	Descrição
Não classificado	Cinza	Pode ser usada quando o nível de gravidade de um evento é desconhecido, não foi determinado, não faz parte do escopo regular de monitoramento, etc., por exemplo, durante a configuração inicial, como um espaço reservado para avaliação futura ou como parte de um processo de integração.
Informação	Azul claro	Pode ser usada para eventos informativos que não requerem atenção imediata, mas que ainda podem fornecer informações valiosas.
Aviso	Amarelo	Pode ser usada para indicar um possível problema que pode exigir investigação ou ação, mas que não é crítico.
Média	Laranja	Pode ser usada para indicar um problema significativo que deve ser resolvido relativamente em breve para evitar problemas adicionais.
Alta	Vermelho claro	Pode ser usada para indicar problemas críticos que precisam de atenção imediata para evitar interrupções significativas.
Desastre	Vermelho	Pode ser usada para indicar um incidente grave que requer ação imediata para evitar, por exemplo, paralisações do sistema ou perda de dados.

Os nomes e cores das gravidades dos triggers podem ser **personalizados** em *Administração* > *Geral* > *Opções de exibição de triggers*. As cores são compartilhadas entre todos os temas do frontend.

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 os arquivos .mo conforme descrito em <frontend_dir>/locale/README

Aqui **msgid** deve corresponder ao novo nome personalizado de severidade e **msgstr** deve ser a tradução para ele no idioma específico.

Este procedimento deve ser realizado após cada alteração no nome da severidade.

Visualização

Visão geral

A visualização oferece diferentes maneiras de exibir os dados de monitoramento coletados pelo Zabbix. Os gráficos mostram tendências e métricas, os mapas de rede exibem a topologia da infraestrutura, e os painéis combinam widgets personalizáveis para fornecer visões abrangentes de monitoramento.

Ferramentas de visualização

- Gráficos - visualize dados de item como gráficos simples gerados automaticamente, crie gráficos personalizados ou compare vários itens em gráficos ad hoc.
- Mapas de rede - crie representações visuais da infraestrutura de rede com elementos e indicadores personalizáveis.
- Dashboards - personalize layouts combinando uma variedade de widgets de dashboard que visualizam dados em diferentes formatos.

Gráficos

Visão geral

Com muitos dados fluindo para o Zabbix, fica muito mais fácil para os usuários visualizar uma representação do que está acontecendo, em vez de ver apenas números.

É nesse contexto que entram os gráficos. Os gráficos permitem compreender o fluxo de dados rapidamente, correlacionar problemas, descobrir quando algo começou ou ilustrar quando algo pode se transformar em um problema.

O Zabbix oferece aos usuários:

- Gráficos simples integrados de dados de um item.
- A possibilidade de criar gráficos personalizados mais complexos.
- Acesso rápido à comparação de vários itens em gráficos ad hoc.
- Gráficos vetoriais modernos e personalizáveis e gráficos de pizza.

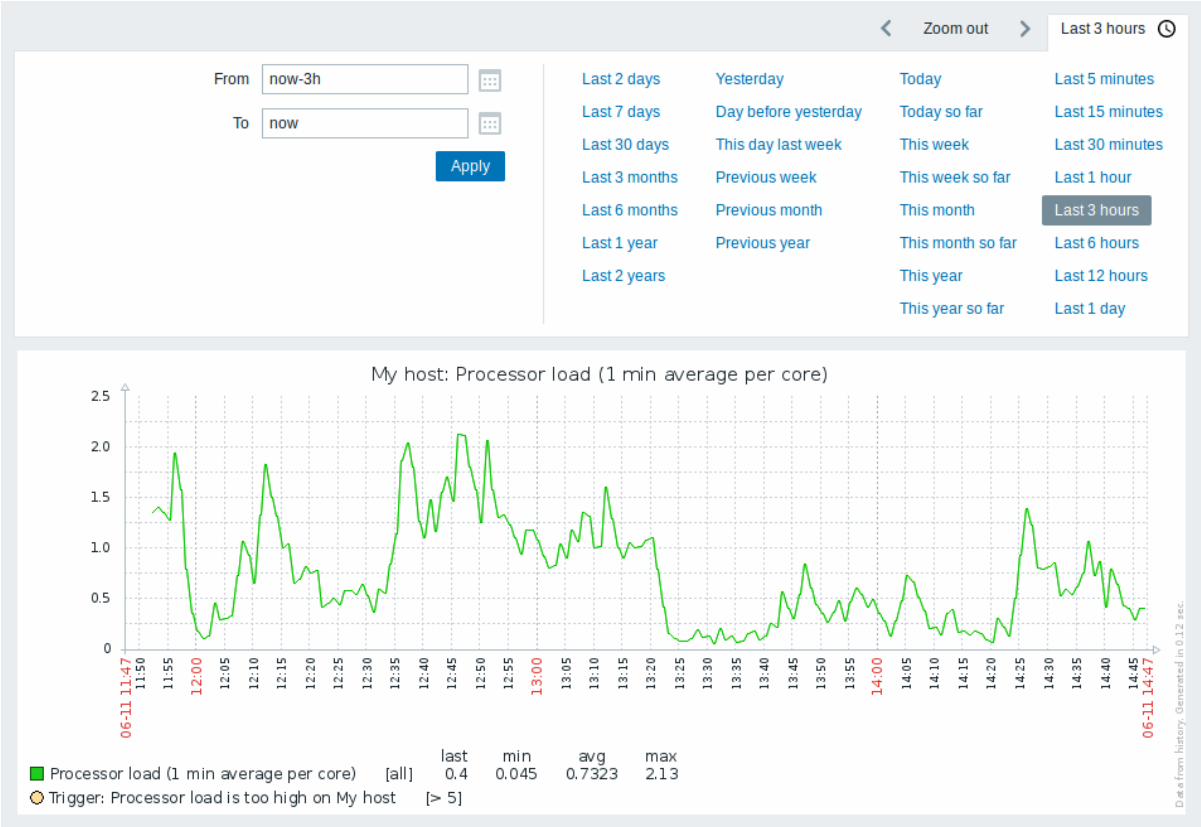
Gráficos simples

Visão geral

Gráficos simples são fornecidos para a visualização dos dados coletados pelos itens.

Não é necessário nenhum esforço de configuração por parte do usuário para visualizar gráficos simples. Eles são disponibilizados gratuitamente pelo Zabbix.

Basta acessar *Monitoring* > *Latest data* e clicar no link *Graph* do item correspondente para que um gráfico seja exibido.



Note:
Gráficos simples são fornecidos para todos os itens numéricos. Para itens de texto, um link para *History* está disponível em *Monitoring* > *Latest data*.

Seletor de período de tempo

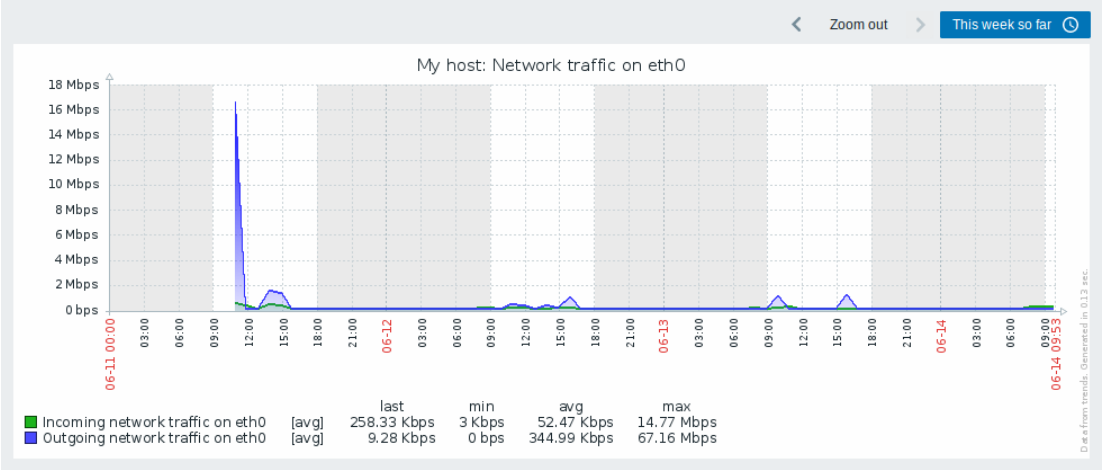
O seletor de período de tempo acima do gráfico permite selecionar períodos frequentemente necessários com um único clique do mouse. Para obter mais informações, consulte Seletores de período de tempo e host.

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 máximo e mínimo naquele momento. O espaço entre os valores mais altos e mais baixos é preenchido com um fundo amarelo.

O horário de trabalho (dias úteis) é exibido nos gráficos com fundo branco, enquanto o horário não útil é exibido em cinza (com o tema padrão do frontend *Original blue*).

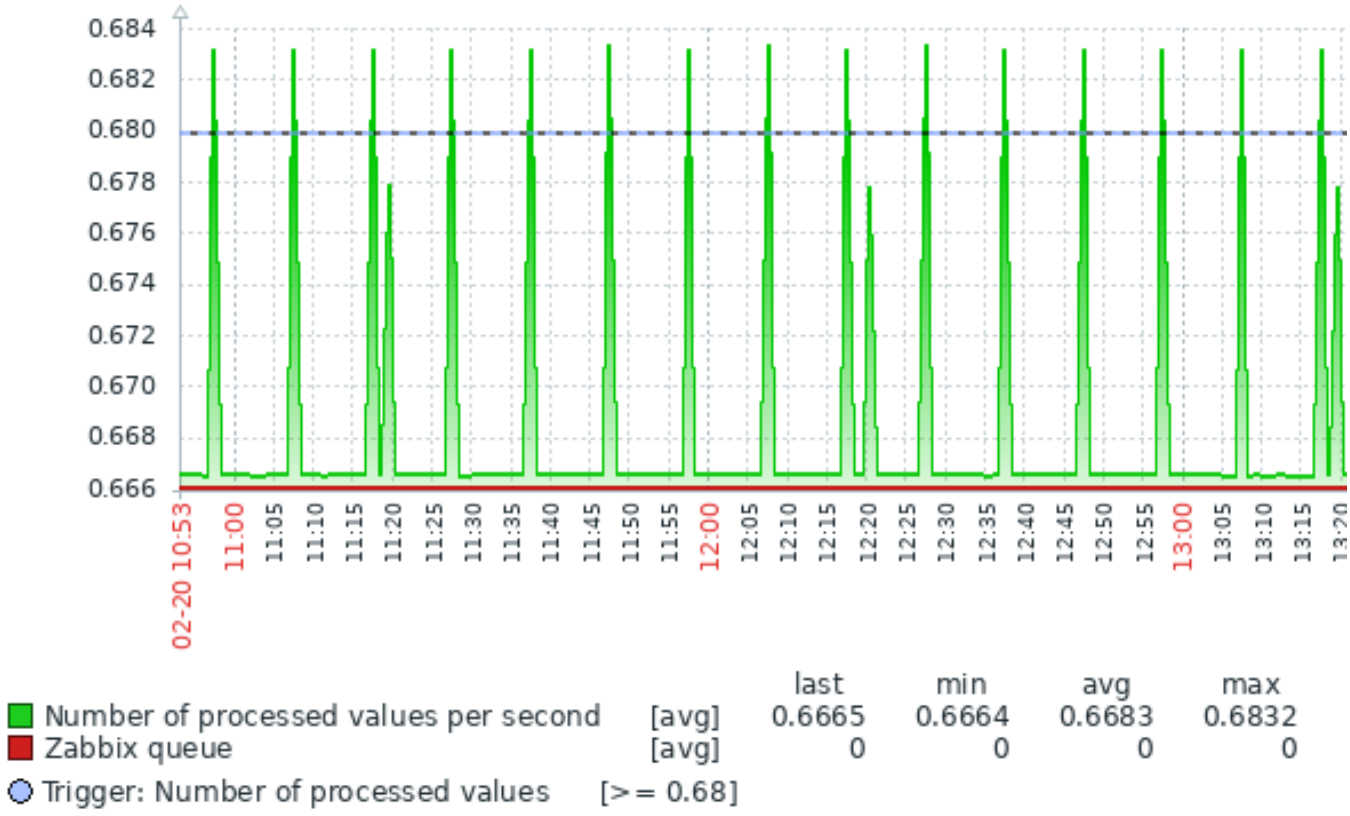


O horário de trabalho é sempre exibido em gráficos simples, enquanto sua exibição em gráficos personalizados é uma preferência do usuário.

O horário de trabalho não é exibido se o gráfico mostrar mais de 3 meses.

Linhas de trigger

Triggers simples são exibidos 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, então os triggers com severidade mais baixa são priorizados. Os triggers são sempre exibidos em gráficos simples, enquanto a exibição deles em gráficos personalizados é uma preferência do usuário.



Geração a partir de histórico/tendências

Os gráficos podem ser desenhados com base no histórico ou nas tendências de um item.

Para 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 serão usados:

- Retenção do histórico do item. Por exemplo, o histórico do item pode ser mantido por 14 dias. Nesse caso, todos os dados anteriores a quatorze dias serão provenientes das tendências.
- Congestionamento de dados no gráfico. Se a quantidade de segundos a ser exibida em um pixel horizontal do gráfico exceder 3600/16, os dados de tendências serão exibidos (mesmo que o histórico do item ainda esteja disponível para o mesmo período).
- Se as tendências estiverem desativadas, o histórico do item será usado para criar o gráfico, se estiver 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 do tipo trapper e items com um intervalo de atualização agendado (e intervalo de atualização regular definido como 0), uma linha reta é desenhada 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

Uma lista suspensa no canto superior direito permite alternar do gráfico simples para as listagens *Valores/500 valores mais recentes*. Isso pode ser útil para visualizar os valores numéricos que compõem o gráfico.

Os valores representados aqui são brutos, ou seja, nenhuma unidade ou pós-processamento dos valores é utilizado. No entanto, o mapeamento de valores é aplicado.

Problemas conhecidos

Consulte problemas conhecidos para gráficos.

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.

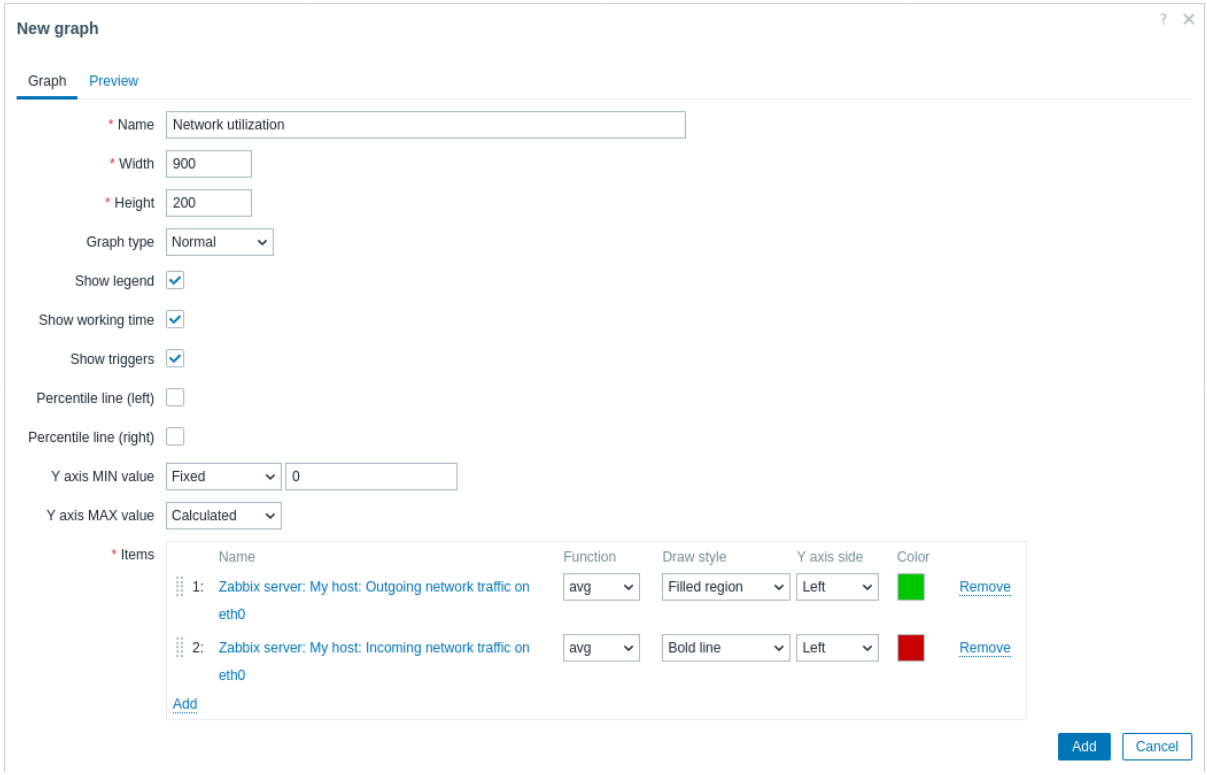
Gráficos personalizados são configurados manualmente.

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:

1. Vá para *Coleta de dados* > *Hosts* (ou *Templates*).
2. Clique em *Gráficos* na linha ao lado do host ou template desejado.
3. Na tela de Gráficos, clique em *Criar gráfico*.
4. Edite os atributos do gráfico.



Todos os campos obrigatórios são marcados com um asterisco vermelho.

Atributos do gráfico:

Parâmetro	Descrição
Nome	Nome exclusivo do gráfico. Macros de expressão macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`). 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 (somente 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”, com as partes exibidas como ”recortadas” da pizza
Mostrar legenda	Marcar esta caixa fará com que a legenda do gráfico seja exibida.
Mostrar horário de trabalho	Se selecionado, as horas não úteis serão mostradas com um fundo cinza. Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Mostrar triggers	Se selecionado, simple triggers serão exibidos como linhas com traços pretos sobre a cor da severidade do trigger. Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Linha de percentil (esquerda)	Exibe o percentil para o eixo Y esquerdo. Se, por exemplo, o percentil de 95% estiver definido, a linha de percentil ficará no nível em que 95% dos valores ficam abaixo. Exibida como uma linha verde brilhante. Disponível apenas para gráficos normais.

Parâmetro	Descrição
Linha de percentil (direita)	Exibe o percentil para o eixo Y direito. Se, por exemplo, o percentil de 95% estiver definido, a linha de percentil ficará no nível em que 95% dos valores ficam abaixo. Exibida 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	Ativa o estilo 3D. Somente para gráficos de pizza e pizza explodida.
Items	Items cujos dados serão exibidos neste gráfico. Clique em *Adicionar* para selecionar items. Você também pode selecionar várias opções de exibição (função, estilo de desenho, exibição no 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 items 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 items disponíveis.
Tipo	Tipo (disponível apenas para gráficos de pizza e pizza explodida): **Simple** - o valor do item é representado proporcionalmente na pizza **Graph sum** - o valor do item representa a pizza inteira Observe que a coloração do item ”graph sum” só ficará visível na medida em que não for ocupada por items ”proporcionais”.
Função	Selecione quais valores serão exibidos quando existir mais de um valor 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, essa configuração não tem efeito; somente para períodos mais longos, quando a concentração de dados em um pixel vertical do gráfico aumenta, ’all’ começa a exibir os valores mínimo, máximo e médio. Esta função está disponível apenas para o tipo de gráfico *Normal*. Veja também: Gerando gráficos a partir de history/trends. **avg** - exibe os valores médios **last** - exibe os valores mais recentes. Esta função está disponível apenas se *Pizza/Pizza explodida* estiver 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 exibir 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.

Gráficos ad hoc

Visão geral

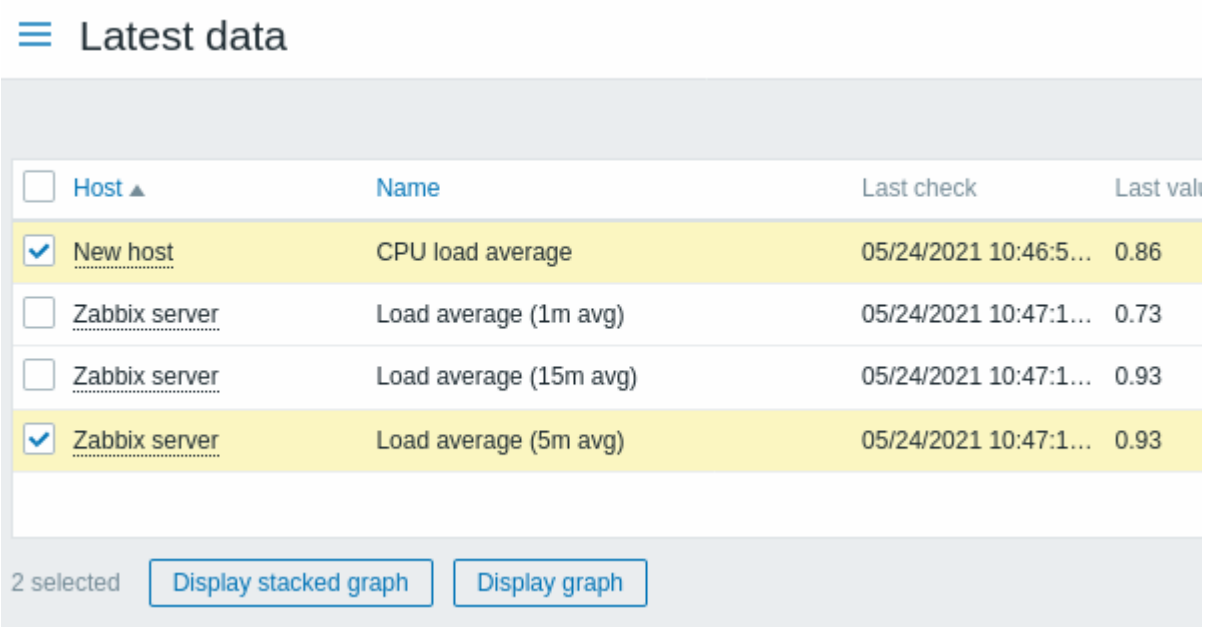
Embora um gráfico simples seja útil para visualizar dados de um único item e os gráficos personalizados ofereçam opções de personalização, nenhum dos dois permite criar rapidamente um gráfico de comparação para vários itens com o mínimo de esforço e manutenção.

Para resolver esse problema, é possível criar gráficos ad hoc para vários itens de forma muito rápida.

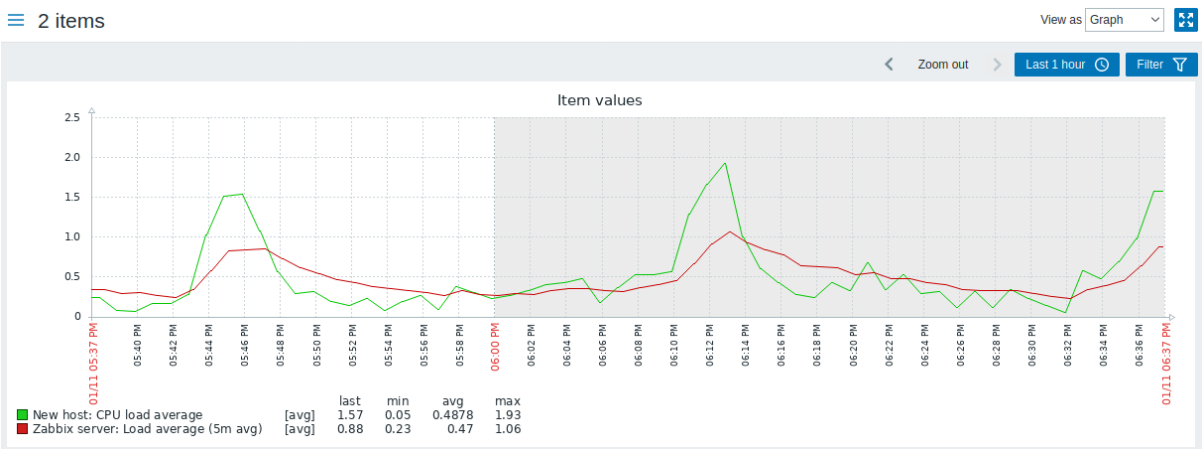
Configuração

Para criar um gráfico ad hoc, faça o seguinte:

1. Vá para *Monitoring* > *Latest data*.
2. Use o filtro para exibir os items que você deseja.
3. Marque as caixas de seleção dos items que você deseja incluir no gráfico.
4. Clique nos botões *Display stacked graph* ou *Display graph*.

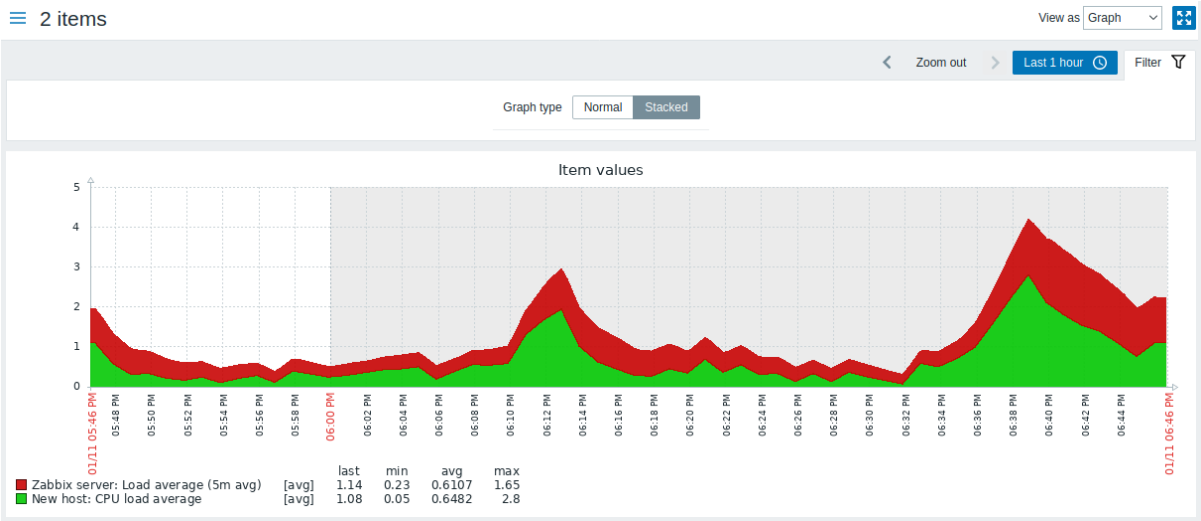


Seu gráfico é criado instantaneamente:



Observe que, para evitar a exibição de muitas linhas no gráfico, apenas o valor médio de cada item é exibido (as 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 um empilhado e vice-versa.



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.

Configurando um mapa de rede

Visão geral

Configurar um mapa no Zabbix exige que você primeiro crie um mapa definindo seus parâmetros gerais e, em seguida, comece a preencher o mapa propriamente dito com elementos e seus links.

Você pode preencher o mapa com elementos que sejam um host, um grupo de hosts, um trigger, uma imagem ou outro mapa.

Os ícones são usados para representar os elementos do mapa. Você pode definir as informações que serão exibidas com os ícones e configurar para que os problemas recentes sejam exibidos de uma forma especial. Você pode vincular os ícones e definir informações a serem exibidas nos links.

Você pode adicionar URLs personalizadas para que fiquem 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 causados.

Os mapas são gerenciados em *Monitoring* > *Maps*, 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 públicos ou privados.

- *Mapas públicos* são visíveis para todos os usuários, embora, para visualizá-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 respectivo mapa, 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 comuns (não Super administradores) só podem compartilhar com os grupos aos quais pertencem e com usuários que pertencem a esses grupos. 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 respectivo mapa, 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.

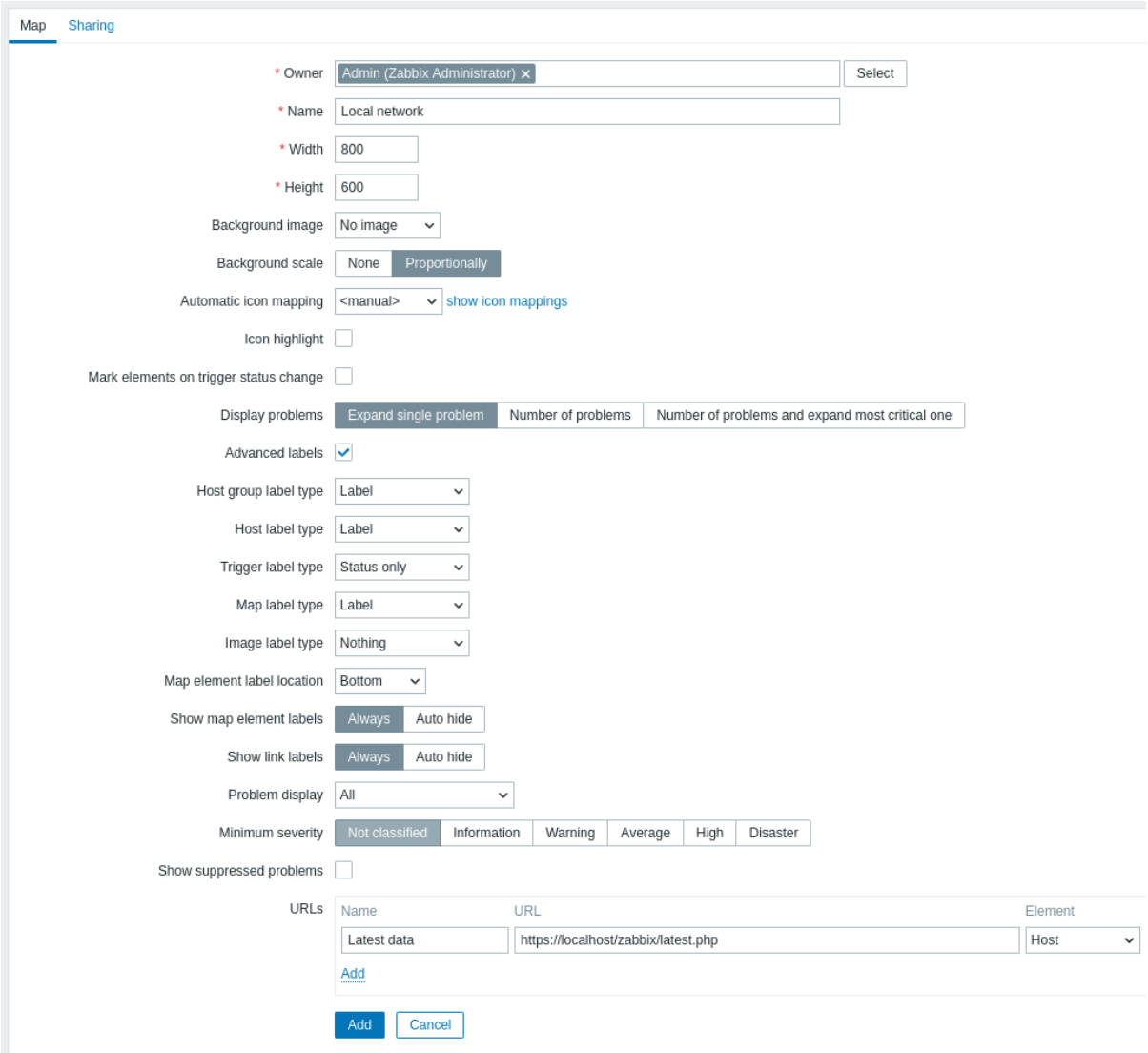
Criando um mapa

Para criar um mapa, faça o seguinte:

1. Acesse *Monitoramento* > *Mapas*.
2. Acesse a visualização com todos os mapas.
3. Clique em *Criar mapa*.
4. Configure os detalhes do mapa.

Você também pode usar o botão *Clonar* no formulário de configuração de um mapa existente para criar um novo mapa. Esse mapa terá todas as propriedades do mapa existente, incluindo os atributos gerais de layout, bem como os elementos do mapa existente.

A aba **Mapa** contém os atributos gerais do mapa:



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Atributos gerais do mapa:

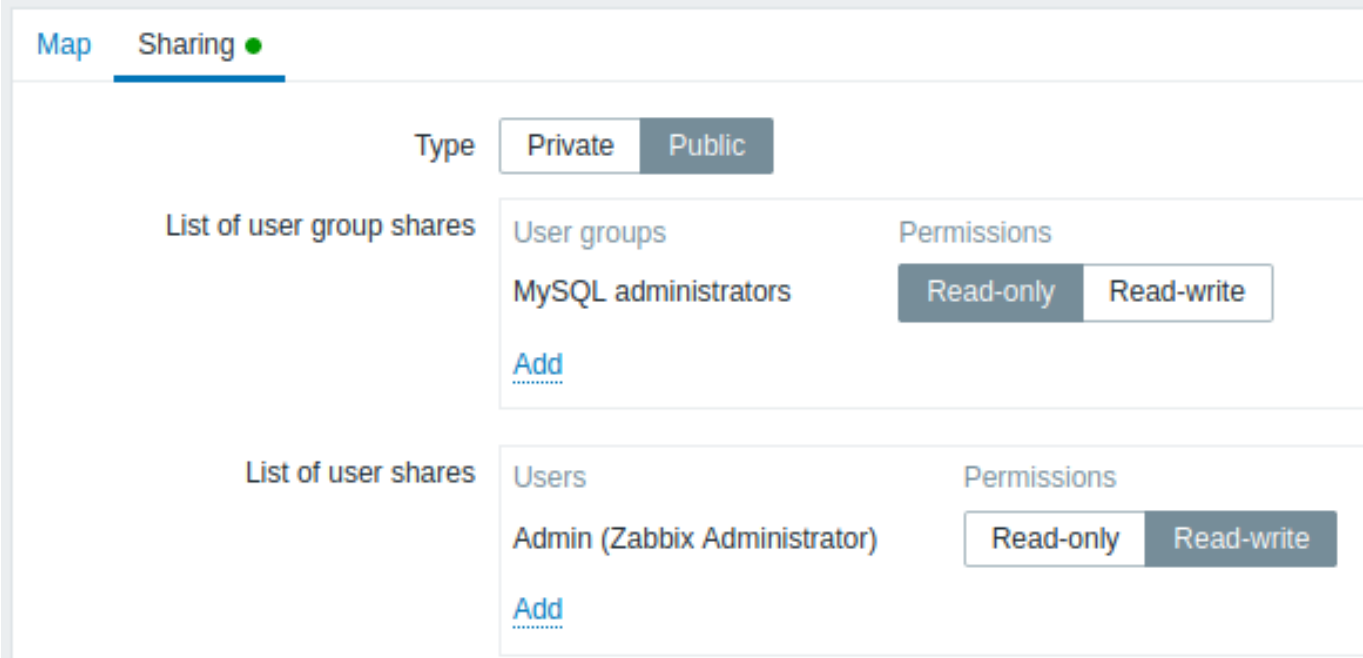
Parâmetro	Descrição
Proprietário	Nome do proprietário do mapa.
Nome	Nome exclusivo do mapa.
Largura	Largura do mapa em pixels.
Altura	Altura do mapa em pixels.

Parâmetro	Descrição
Imagem de fundo	Usar imagem de fundo: **Nenhuma imagem** - nenhuma imagem de fundo (fundo branco) **Imagem** - imagem selecionada a ser usada como imagem de fundo. Você pode usar um mapa geográfico ou qualquer outra imagem para aprimorar o mapa.
Escala do fundo	Usar escala do fundo: **Nenhuma** - sem escala; **Proporcionalmente** - cobrir todo o fundo do mapa, mantendo as proporções da imagem (padrão).
Mapeamento automático de ícones	Você pode definir o uso de um mapeamento automático de ícones, configurado em *Administração > Geral > Mapeamento de ícones*. O mapeamento de ícones permite associar determinados ícones a determinados campos do inventário do host.
Destaque do ícone	Se você marcar esta caixa de seleção, os elementos do mapa serão destacados. Elementos com um trigger ativo receberão um fundo circular, na mesma cor do trigger com a maior severidade. Além disso, uma linha verde espessa será exibida ao redor do círculo se todos os problemas tiverem sido reconhecidos. Elementos com status ”desabilitado” ou ”em manutenção” receberão um fundo quadrado, cinza e laranja, respectivamente. Veja também: Visualizando mapas
Marcar elementos na alteração do status do trigger	Uma alteração recente no estado do trigger (problema recente ou resolução) será destacada com marcadores (triângulos vermelhos apontando para dentro) nos três lados do ícone do elemento que estiverem livres do rótulo. Os marcadores são exibidos por 30 minutos.
Exibir problemas	Selecione como os problemas serão exibidos com um elemento do mapa: **Expandir problema único** - se houver apenas um problema, o nome do problema será exibido. Caso contrário, será exibido o número total de problemas. **Número de problemas** - o número total de problemas será exibido **Número de problemas e expandir o mais crítico** - o nome do problema mais crítico e o número total de problemas serão exibidos. O ”mais crítico” é determinado com base na severidade do problema e, em caso de empate, no ID do evento do problema (o ID mais alto ou o problema mais recente é exibido primeiro). Para um *elemento de mapa de trigger*, isso é baseado na severidade do problema e, em caso de empate, na posição do trigger na lista de triggers. No caso de vários problemas do mesmo trigger, o mais recente será exibido.
Rótulos avançados	Se você marcar esta caixa de seleção, poderá definir tipos de rótulo separados para tipos de elementos separados.
Tipo de rótulo do grupo de hosts *Tipo de rótulo do host* *Tipo de rótulo do trigger* *Tipo de rótulo do mapa* *Tipo de rótulo da imagem*	Tipo de rótulo usado para o elemento do mapa: **Rótulo** - rótulo do elemento do mapa **Endereço IP** - endereço IP **Nome do elemento** - nome do elemento (por exemplo, nome do host) **Somente status** - somente o 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 exibidos (padrão); **Ocultar automaticamente** - ocultar o rótulo se ele não estiver sob o cursor ou selecionado.
Exibir rótulos dos links	Selecione como exibir os rótulos dos links: **Sempre** - sempre exibidos (padrão); **Ocultar automaticamente** - ocultar o rótulo se ele não estiver sob o cursor ou selecionado.
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 parte da contagem total de problemas **Somente não reconhecidos** - somente a contagem de problemas não reconhecidos será exibida

Parâmetro	Descrição
Severidade mínima	Os problemas abaixo do nível de severidade mínima selecionado não serão exibidos no mapa. Por exemplo, com *Aviso* selecionado, as alterações com triggers de nível *Informação* e *Não classificado* não serão refletidas no mapa.
Exibir problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
URLs	As URLs para cada tipo de elemento podem ser definidas aqui, onde: **Nome** - um rótulo para a URL; **URL** - uma URL (até 2048 caracteres); **Elemento** - o tipo de elemento (por exemplo, *Host*, *Grupo de hosts* etc.). Elas serão exibidas como links quando um usuário clicar no elemento no modo de visualização do mapa. Macros podem ser usadas nos nomes e valores das URLs do mapa. Para obter a lista completa, consulte macros compatíveis e pesquise por 'nomes e valores de URLs do mapa'.

Compartilhamento

A aba **Sharing** contém o tipo de mapa, bem como as opções de compartilhamento (grupos de usuários, usuários) para mapas privados:



Parameter	Description
Type	Selecione o tipo de mapa: **Private** - o mapa fica visível apenas para os grupos de usuários e usuários selecionados **Public** - o mapa fica visível para todos
List of user group shares	Selecione os grupos de usuários aos quais o mapa terá acesso. Você pode permitir acesso somente leitura ou leitura e gravação.
List of user shares	Selecione os usuários aos quais o mapa terá acesso. Você pode permitir acesso somente leitura ou leitura e gravação.

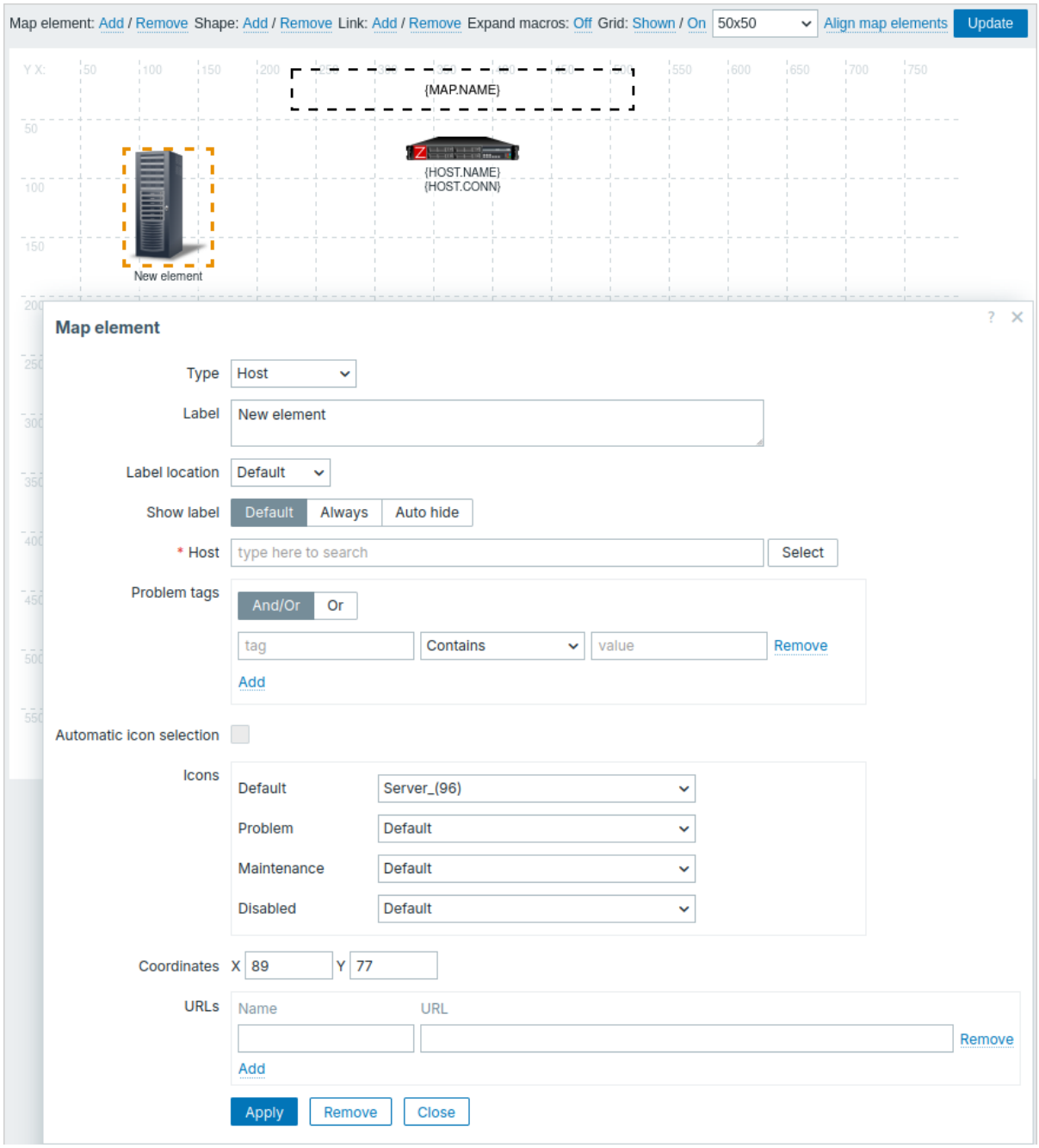
Quando você clica em *Add* para salvar este mapa, você criou um mapa vazio com um nome, dimensões e determinadas preferências. Agora você precisa adicionar alguns elementos. Para isso, clique em *Edit* na lista de mapas para abrir a área editável.

Adicionando elementos

Para adicionar um elemento, clique em *Add* ao lado de `Map element`. O novo elemento aparecerá no canto superior esquerdo do mapa. Arraste e solte-o onde quiser.

Observe que, com a opção Grid "On", os elementos sempre se alinharão à grade (você pode escolher vários tamanhos de grade no menu suspenso, além de ocultar/mostrar a grade). Se quiser posicionar elementos em qualquer lugar sem alinhamento, defina a opção como "Off". (Você pode alinhar elementos aleatórios à grade mais tarde, clicando em *Align map elements*.)

Agora que você já tem alguns elementos posicionados, talvez queira começar a diferenciá-los atribuindo nomes etc. Ao clicar no elemento, um formulário é exibido e você pode definir o tipo do elemento, atribuir um nome, escolher um ícone diferente etc.



Atributos do elemento do mapa:

Parameter	Description
Type	Tipo do elemento: **Host** - ícone que representa o status de todos os triggers do host selecionado **Map** - ícone que representa o status de todos os elementos de um mapa **Trigger** - ícone que representa o status de um ou mais triggers **Host group** - ícone que representa o status de todos os triggers de todos os hosts pertencentes ao grupo selecionado **Image** - um ícone, não vinculado a nenhum recurso
Label	Rótulo do ícone, qualquer string. Macros e strings multilinha podem ser usados. Macros de expressão macros são suportados neste campo, mas apenas com funções `avg`, `last`, `min` e `max`, com tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`). Para uma lista completa de macros suportados, consulte supported macros e procure por 'map element labels'.

Parameter	Description
Label location	Localização do rótulo em relação ao ícone: **Default** - localização padrão do rótulo no mapa; **Bottom** - abaixo do ícone; **Left** - à esquerda; **Right** - à direita; **Top** - acima do ícone.
Show label	Selecione como exibir o rótulo do elemento: **Default** - usa as configurações de *Show map element labels* da configuração do mapa (padrão); **Always** - sempre exibido (padrão); **Auto hide** - oculta o rótulo se ele não estiver com o mouse sobre ele ou selecionado.
Host	Selecione um host; alternativamente, informe o nome do host. Este campo oferece suporte a autocompletar, então, ao começar a digitar o nome de um host, será exibida uma lista suspensa com os hosts correspondentes. Role para baixo para selecionar um. Clique no ícone de remover (x) para remover a seleção. Este campo está disponível apenas para elementos do tipo *Host*.
Map	Selecione um mapa; alternativamente, informe o nome do mapa. Este campo oferece suporte a autocompletar, então, ao começar a digitar o nome de um mapa, será exibida uma lista suspensa com os mapas correspondentes. Role para baixo para selecionar um. Clique no ícone de remover (x) para remover a seleção. Este campo está disponível apenas para elementos do tipo *Map*.
Triggers	Selecione um ou mais triggers no campo *New triggers* abaixo e clique em *Add*.Alternativamente, comece a digitar o nome de um trigger e selecione um na lista suspensa de triggers correspondentes. Clique no ícone de remover (x) para remover a seleção. Este campo está disponível apenas para elementos do tipo *Trigger*. A ordem dos triggers selecionados pode ser alterada, mas apenas dentro da mesma severidade de triggers. A seleção múltipla de triggers também afeta a resolução da macro {HOST.*} tanto no modo de edição quanto no modo de visualização. No modo de edição, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo do primeiro trigger da lista (com base na severidade do trigger). O modo de visualização depende do parâmetro Display problems nos atributos gerais do mapa: - Se o modo *Expand single problem* for escolhido, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo do último trigger de problema detectado (independentemente da severidade) ou do primeiro trigger da lista (caso nenhum problema seja detectado). - Se o modo *Number of problems and expand most critical one* for escolhido, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo da severidade do trigger.
Host group	Selecione um grupo de hosts; alternativamente, informe o grupo de hosts. Este campo oferece suporte a autocompletar, então, ao começar a digitar o nome de um grupo de hosts, será exibida uma lista suspensa com os grupos de hosts correspondentes. Role para baixo para selecionar um. Clique no ícone de remover (x) para remover a seleção. Este campo está disponível apenas para elementos do tipo *Host group*.
Problem tags	Especifique tags para limitar o número de problemas exibidos no widget. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tag especificados **Equals** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas) **Contains** - inclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) **Does not exist** - exclui os nomes de tag especificados **Does not equal** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas) **Does not contain** - exclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida Este campo está disponível apenas para elementos do tipo *Host* e *Host group*.
Automatic icon selection	Quando a caixa de seleção está marcada, o mapeamento de ícones será usado para determinar qual ícone exibir.

Parameter	Description
Icons	Você pode escolher exibir ícones diferentes para o elemento nestes casos: *Default*, *Problem*, *Maintenance* e *Disabled*. Este campo está disponível apenas para elementos do tipo *Host*.
Coordinates	Coordenadas **X** e **Y** do elemento do mapa.
URLs	URLs específicas do elemento podem ser definidas aqui, onde: **Name** - um rótulo para a URL; **URL** - uma URL (até 2048 caracteres). 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 no nível do mapa definidas para seu tipo, elas serão combinadas no mesmo menu. Macros podem ser usadas nos nomes e valores dos elementos do mapa. Para uma lista completa, consulte supported macros e procure por 'map URL names and values'.

Attention:
Os elementos adicionados não são salvos automaticamente. Se você sair da página, todas as alterações poderão ser perdidas.
Portanto, é uma boa ideia clicar no botão **Update** no canto superior direito. Depois de clicado, as alterações são salvas independentemente do que você escolher no pop-up a seguir.
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.

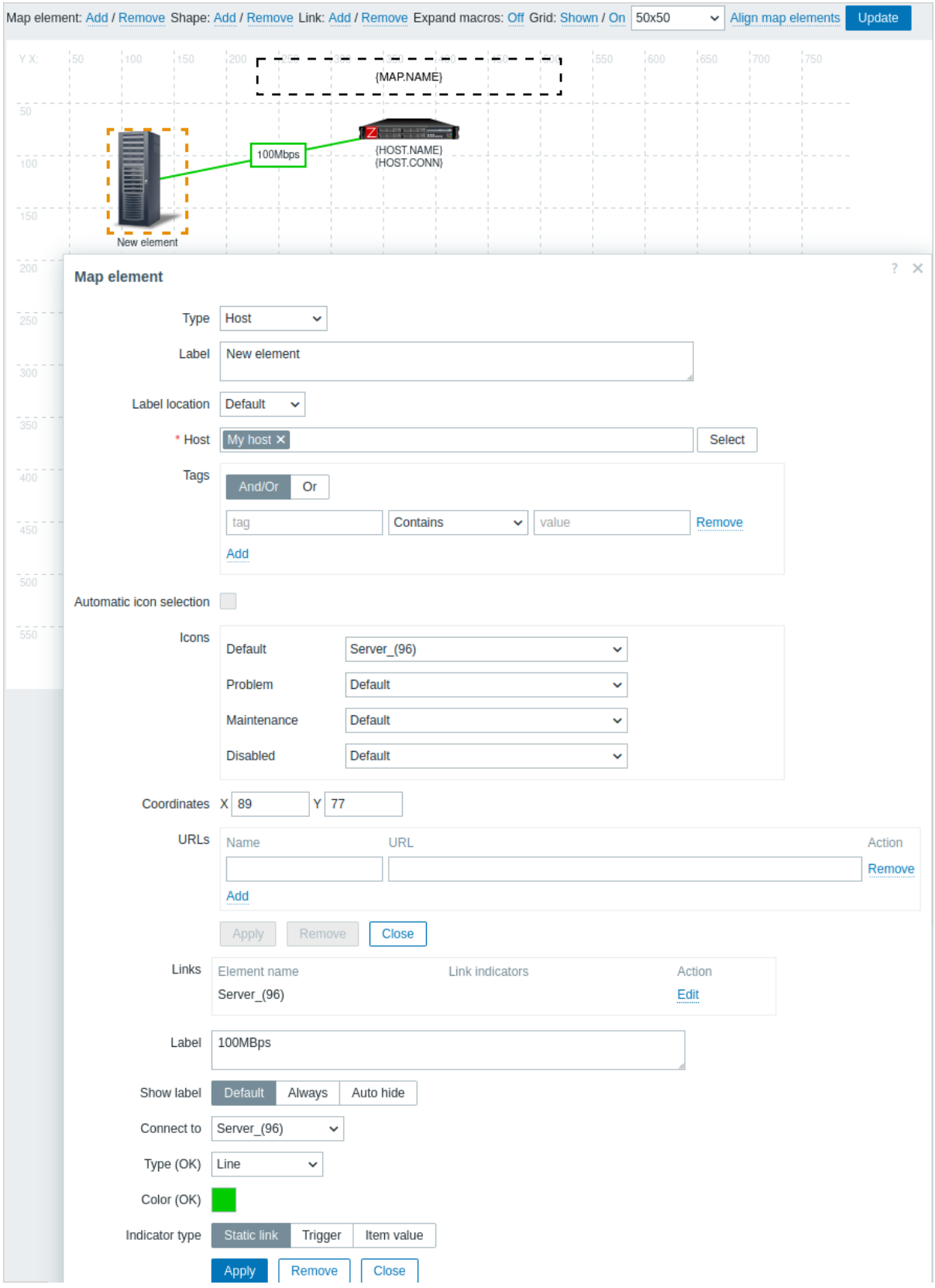
Você também pode selecionar vários elementos arrastando um retângulo na área editável e selecionando todos os elementos dentro dele.

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).

Vinculando elementos

Depois de colocar alguns elementos no mapa, é hora de começar a vinculá-los. Para vincular dois elementos, primeiro você deve selecioná-los. Com os elementos selecionados, clique em *Add* ao lado de Link.

Com um link criado, o formulário de elemento único agora contém uma seção adicional *Links*. Clique em *Edit* para editar os atributos do link.



Atributos do link:

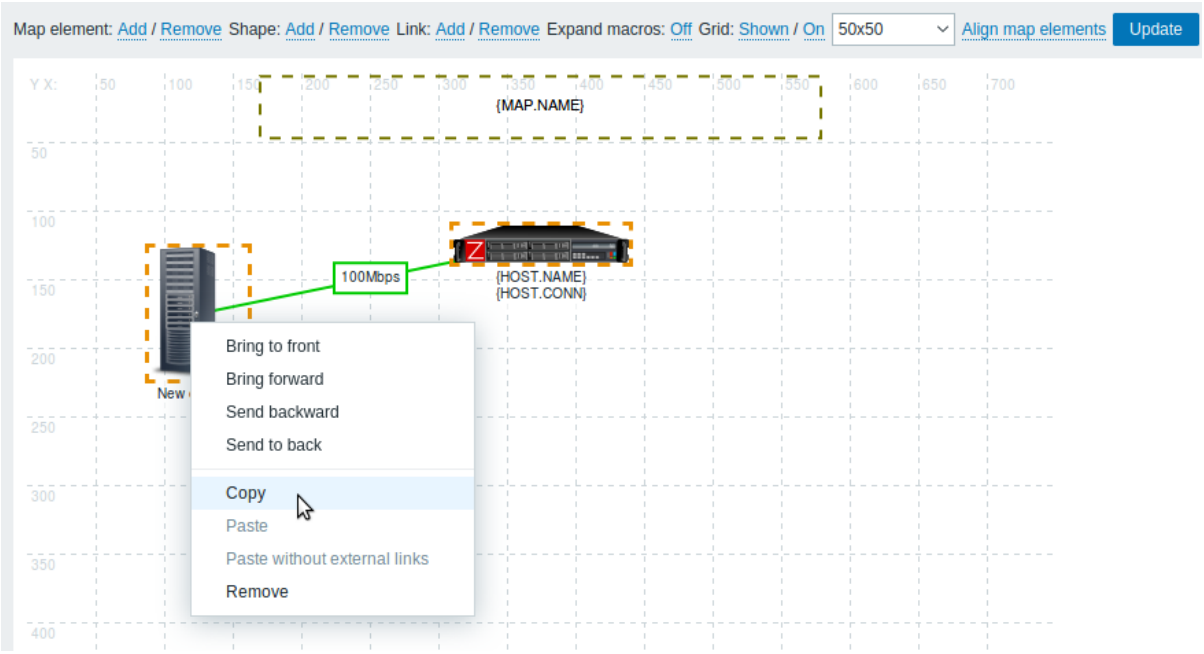
Parameter	Description
Label	Rótulo que será exibido na parte superior do link. Expressões macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`).
Show label	Selecione como exibir o rótulo do link: **Default** - usa as configurações de *Show map element labels* da configuração do mapa (padrão); **Always** - sempre exibido (padrão); **Auto hide** - oculta o rótulo se ele não estiver em foco ou selecionado.

Parameter	Description
Connect to	O elemento ao qual o link se conecta.
Type (OK)	Estilo padrão do link. Selecione o estilo: *Line*, *Bold line*, *Dot* ou *Dashed line*.
Color (OK)	Cor padrão do link. Clique na amostra de cor para selecionar outra cor.
Indicator type	Selecione o tipo de indicador do link: **Static link** - sem indicadores; **Trigger** - permite indicadores de link baseados em trigger; **Item value** - permite indicadores de link baseados no valor do item.
Item	Selecione o item. O valor deste item afetará o estilo do link. Este campo está disponível quando *Item value* é selecionado em *Indicator type*.
Indicators	A lista de link indicators. Este campo está disponível quando *Indicator type* está definido como *Trigger* ou *Item value*. Se definido como *Trigger*, triggers podem ser atribuídos como indicadores do link. Quando um trigger estiver no estado *Problem*, seu estilo (cor e tipo de linha selecionados) será aplicado ao link. Se definido como *Item value* e um item for selecionado no campo *Item*, valores de item podem ser atribuídos como indicadores do link. Quando o valor do item atingir o limite especificado (para um item numérico) ou corresponder ao padrão (para um item de texto), seu estilo (cor e tipo de linha selecionados) será aplicado ao link.

Movendo e copiando e colando elementos

Vários elementos selecionados podem ser **movidos** para outro local no mapa clicando em um dos elementos selecionados, mantendo pressionado o botão do mouse e movendo o cursor para o local desejado.

Um ou mais elementos podem ser **copiados** selecionando os elementos e, em seguida, clicando em um elemento selecionado com o botão direito do mouse e selecionando *Copy* no menu.

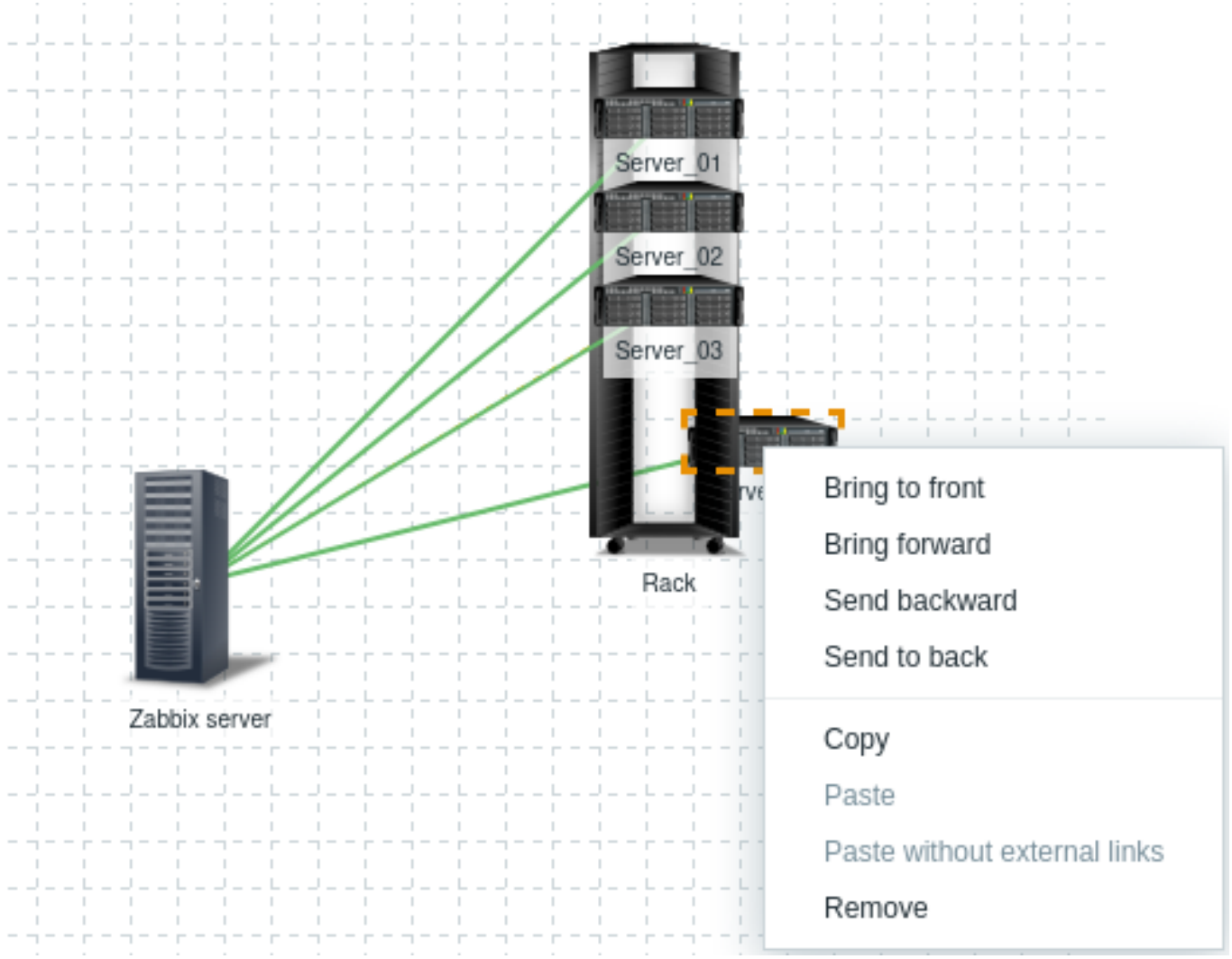


Para colar os elementos, clique em uma área do mapa com o botão direito do mouse e selecione *Paste* no menu. A opção *Paste without external links* colará os elementos mantendo apenas os links que existem entre os elementos selecionados.

A cópia e colagem funcionam dentro da mesma janela do navegador. Atalhos de teclado não são suportados.

Ordenando elementos

Para trazer um elemento para a frente do outro (ou vice-versa), clique no elemento com o botão direito do mouse e selecione *Bring forward*/*Bring to front* ou *Send backward*/*Send to back*.

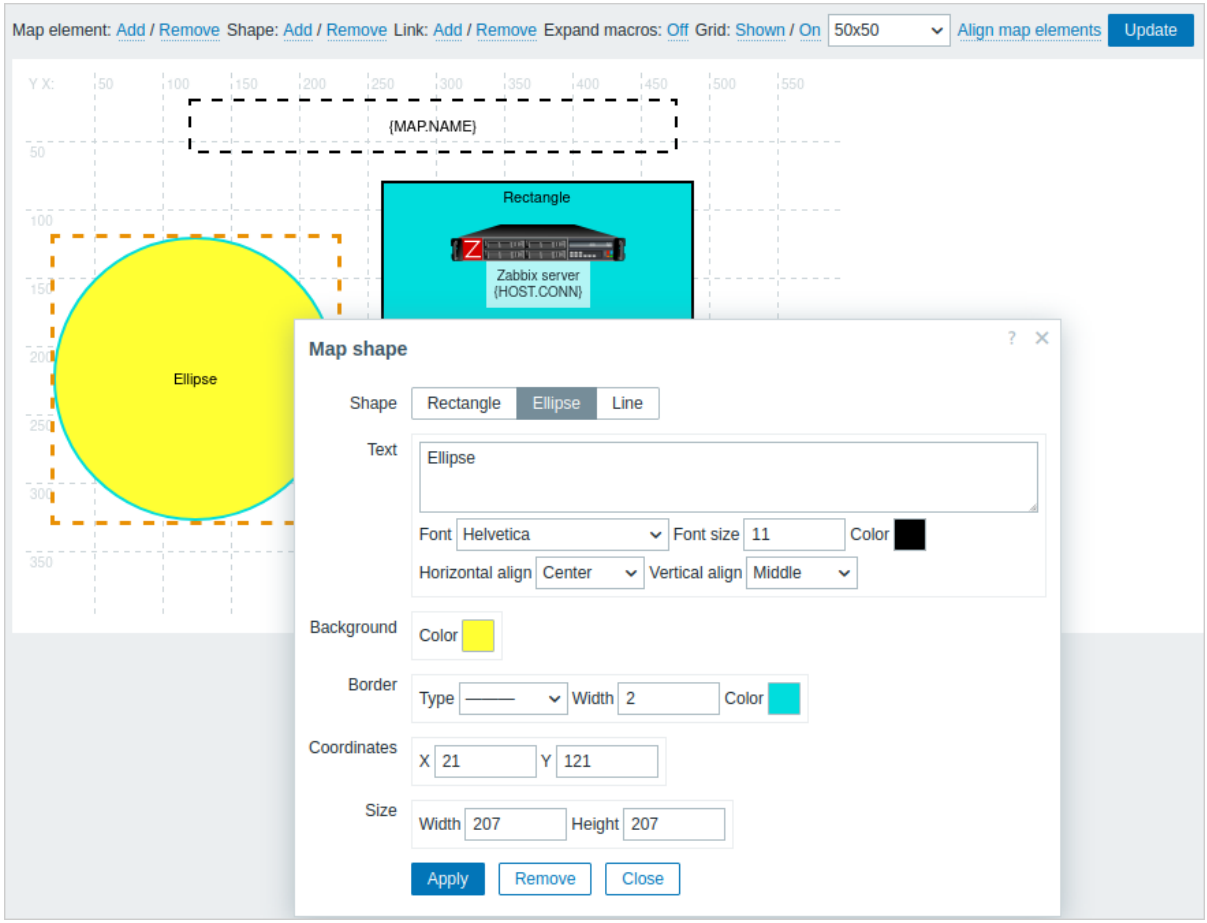


Adicionando formas

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 plano de fundo para agrupar alguns hosts. Formas de retângulo e elipse podem ser adicionadas.

Para adicionar uma forma, clique em *Add* ao lado de Shape. A nova forma aparecerá no canto superior esquerdo do mapa. Arraste e solte-a onde quiser.

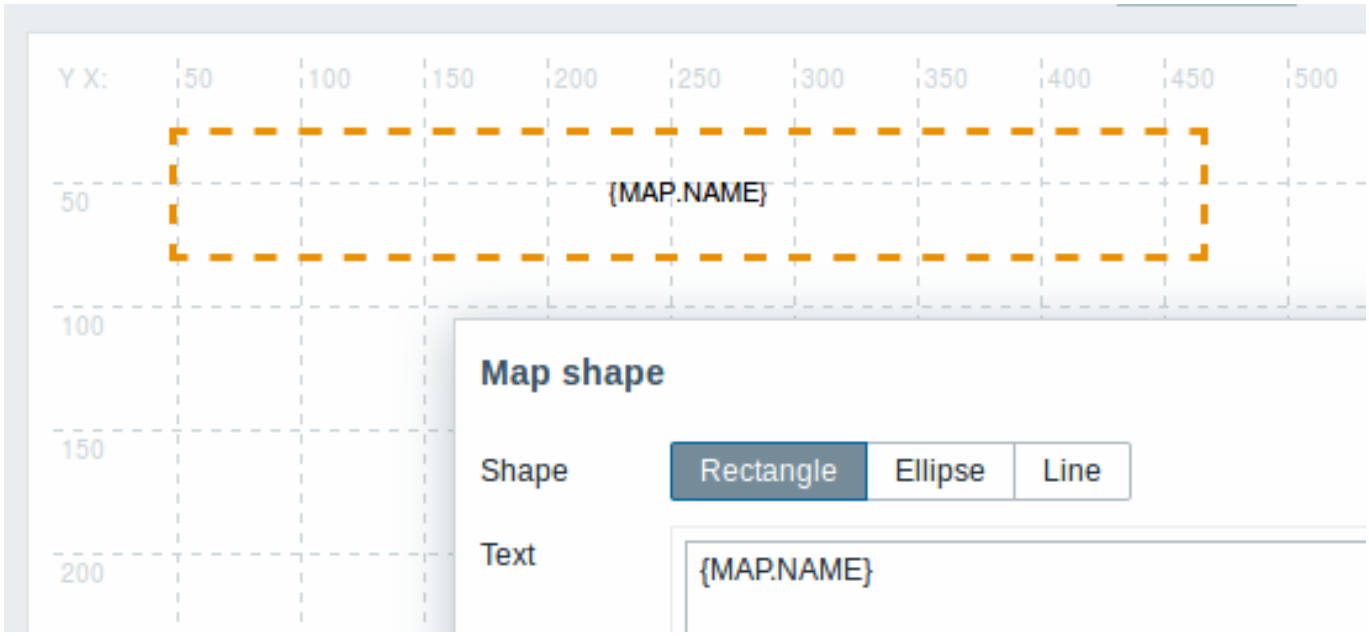
Uma nova forma é adicionada com as cores padrão. Ao clicar na forma, um formulário é exibido e você pode personalizar a aparência da forma, adicionar texto, etc.



Para selecionar formas, selecione uma e depois mantenha pressionado *Ctrl* para selecionar as outras. Com várias formas selecionadas, as propriedades comuns podem ser atualizadas em massa, da mesma forma que com os elementos.

É possível adicionar texto nas formas. Macros de expressão macros são suportadas no texto, mas apenas com as funções `avg`, `last`, `min` e `max`, com tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`).

Para exibir apenas o texto, a forma pode ser tornada invisível removendo a borda da forma (selecione 'None' no campo *Border*). 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 hyperlinks forem usados no texto, eles se tornam clicáveis ao visualizar o mapa.

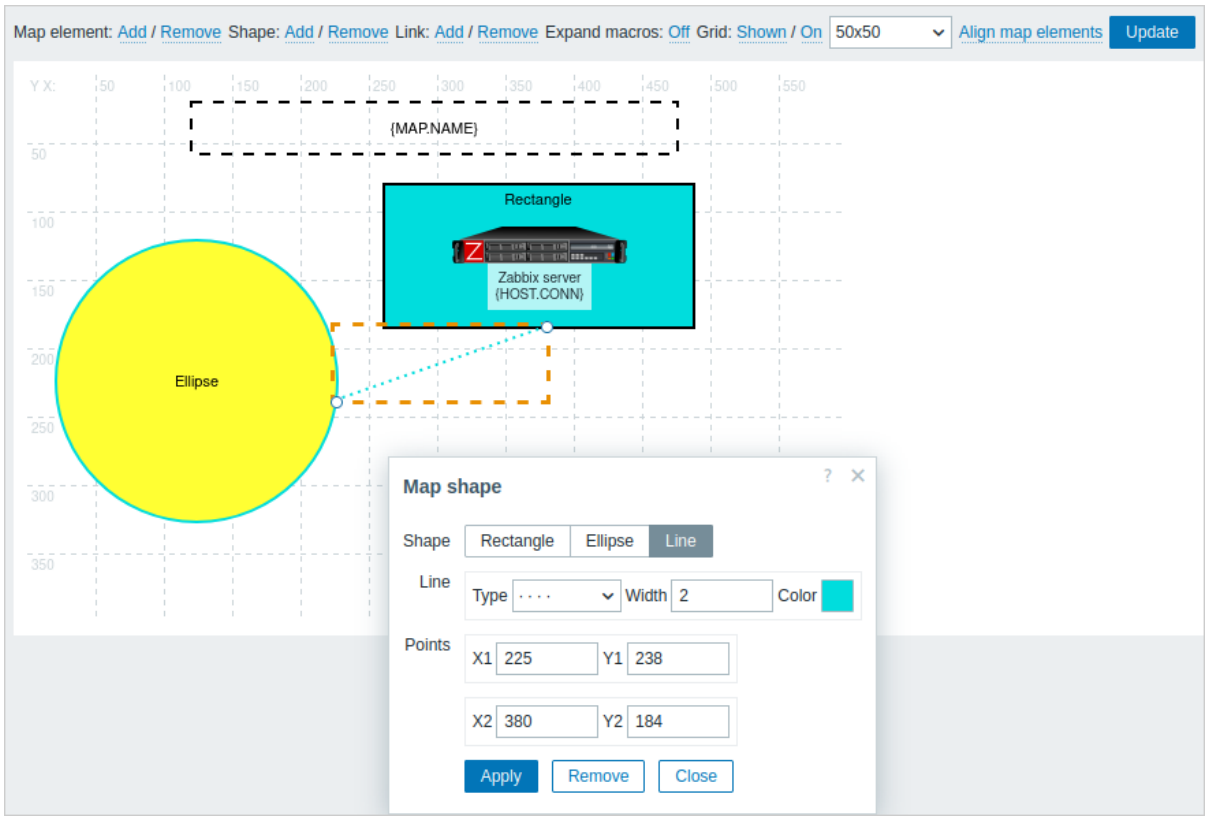
A quebra de linha do texto está sempre "ativada" dentro das formas. No entanto, dentro de uma elipse, as linhas são quebradas como se a elipse fosse um retângulo. A quebra de palavras não é implementada, então palavras longas (palavras que não cabem

na forma) não são quebradas, mas são mascaradas (na página de edição do mapa) ou cortadas (em outras páginas com mapas).

Adicionando linhas

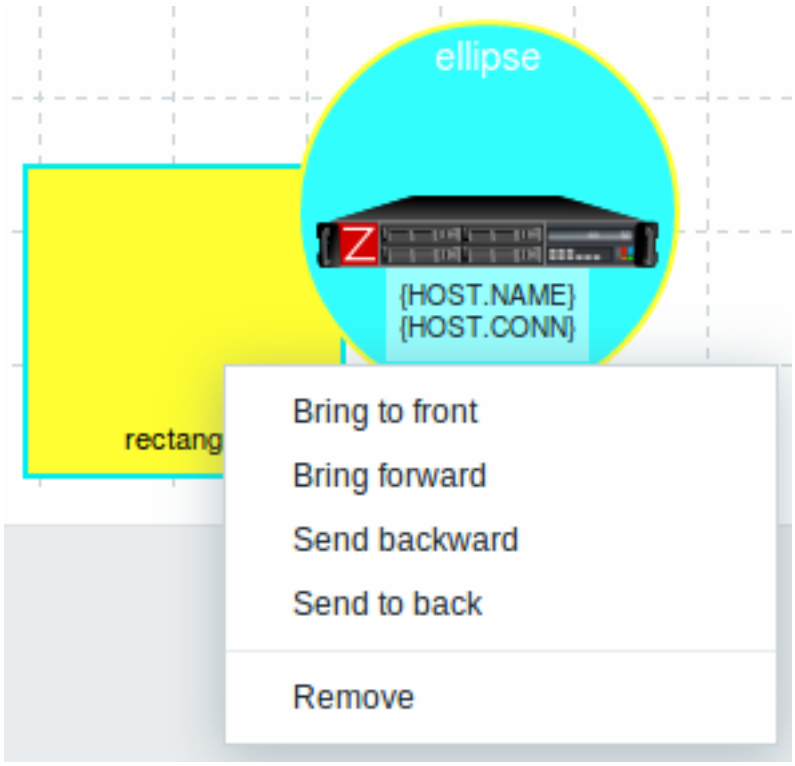
Além de shapes, também é possível adicionar algumas linhas. As linhas podem ser usadas para vincular elementos ou shapes em um mapa.

Para adicionar uma linha, clique em *Add* ao lado de Shape. Um novo shape aparecerá no canto superior esquerdo do mapa. Selecione-o e clique em *Line* no formulário de edição para transformar o shape em uma linha. Em seguida, ajuste as propriedades da linha, como tipo de linha, largura, cor etc.



Ordenando formas e linhas

Para trazer uma forma para frente da outra (ou vice-versa), clique na forma com o botão direito do mouse para abrir o menu de formas do mapa.

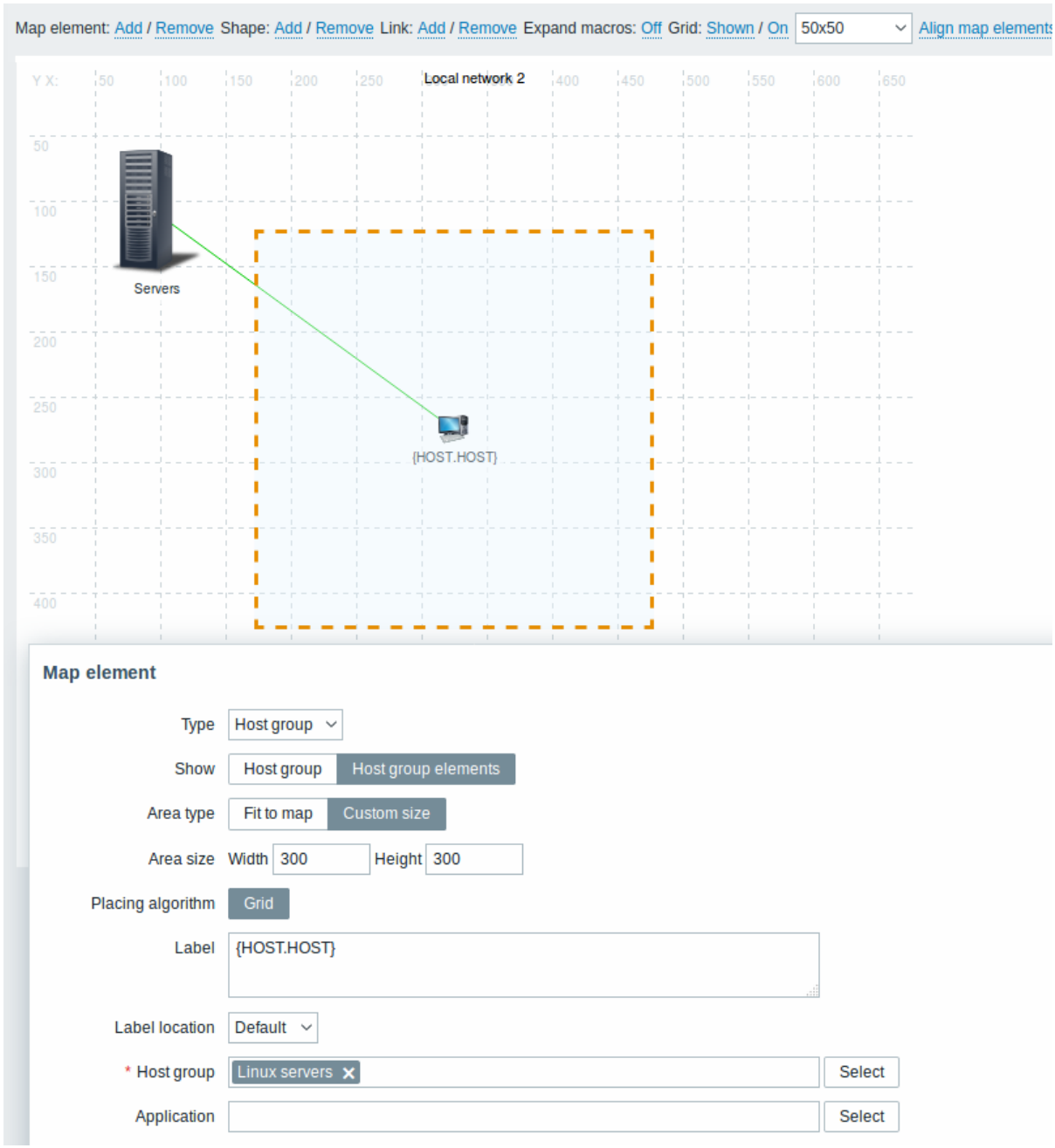


Elementos do grupo de hosts

Visão geral

Esta seção explica como adicionar um elemento do tipo *Grupo de hosts* ao configurar um mapa de rede.

Configuração



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

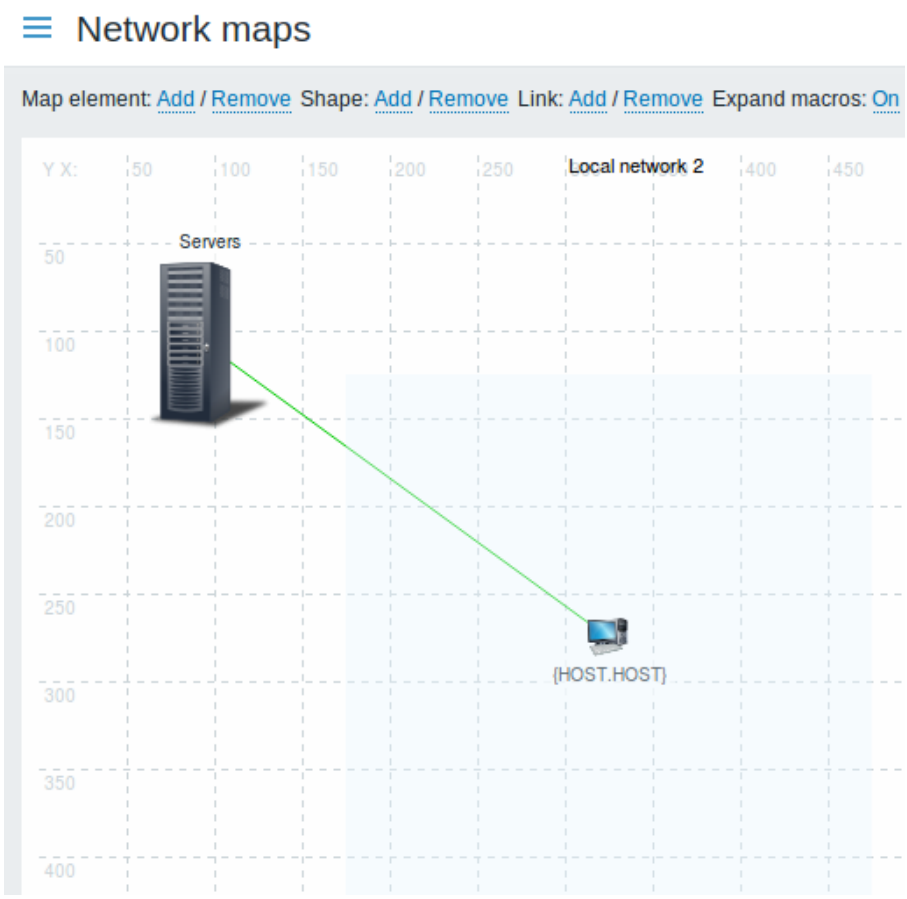
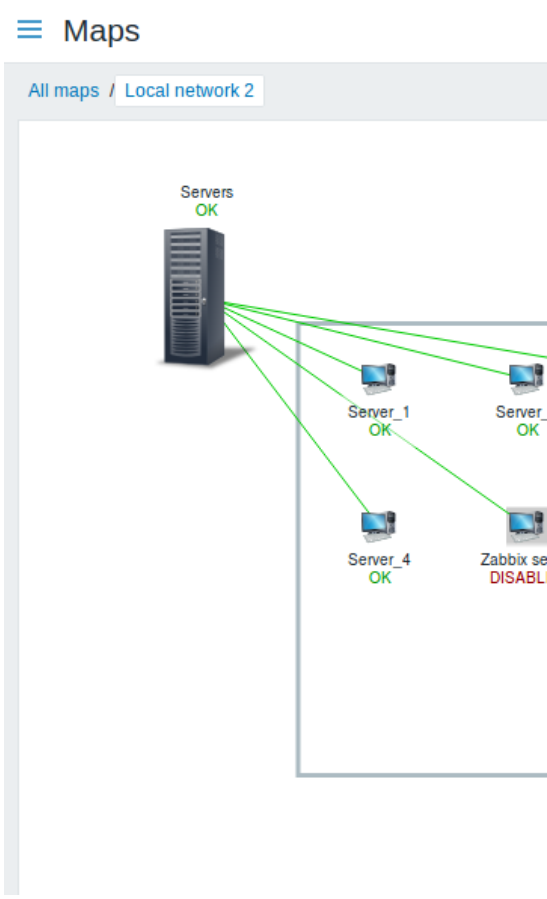
Esta tabela contém parâmetros típicos do tipo de elemento *Host group*:

Parâmetro	Descrição
Type	Selecione o tipo do elemento: **Host group** - Ícone que representa o status de todos os triggers de todos os hosts pertencentes ao grupo selecionado.
Show	Opções de exibição: **Host group** - A seleção desta opção resultará em um único ícone exibindo as informações correspondentes sobre o determinado grupo de hosts. **Host group elements** - A seleção desta opção resultará em vários ícones exibindo as informações correspondentes sobre cada elemento (host) do determinado grupo de hosts.

Parâmetro	Descrição
Area type	Esta configuração está disponível se *Show* estiver definido como *Host group elements*: **Fit to map** - Todos os elementos do grupo de hosts são posicionados igualmente dentro do mapa. **Custom size** - Configuração manual da área do mapa para todos os elementos do grupo de hosts a serem exibidos.
Area size	Esta configuração está disponível se *Area type* estiver definido como *Custom size*: **Width** - Valor em pixels que especifica a largura da área do mapa. **Height** - Valor em pixels que especifica a altura da área do mapa.
Placing algorithm	**Grid** - Única opção disponível para exibir todos os elementos do grupo de hosts.
Label	Rótulo do ícone, qualquer string. Macros e strings multilinha podem ser usadas nos rótulos. Se o tipo do elemento do mapa for *Host group*, a especificação de determinadas macros afetará a visualização do mapa, exibindo as informações correspondentes sobre cada host. Por exemplo, se a macro `{HOST.IP}` for usada, a visualização de edição do mapa exibirá apenas a própria macro {HOST.IP}, enquanto a visualização do mapa incluirá e exibirá o endereço IP exclusivo de cada host.

Visualização dos elementos do grupo de hosts

Esta opção está disponível se a opção de exibição *Elementos do grupo de hosts* estiver selecionada.
Ao selecionar *Elementos do grupo de hosts* como opção de *exibição*, inicialmente você verá apenas um ícone para o grupo de hosts.
No entanto, ao salvar o mapa e acessar a visualização do mapa, você verá que o mapa inclui todos os elementos (hosts) do grupo de hosts específico:

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 são exibidos os nomes exclusivos de todos os hosts.

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 em valores de item. É possível exibir diferentes estilos e cores de link:

- quando triggers entram em um 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 estado do trigger ou do valor do item.

Por exemplo, se qualquer um dos triggers atribuídos entrar em um estado de problema, o estilo e a cor do link mudarão para refletir isso. Assim, talvez o seu link padrão fosse uma linha verde. Agora, com o trigger no estado de problema, seu link pode se tornar vermelho em negrito (se você o tiver definido dessa forma).

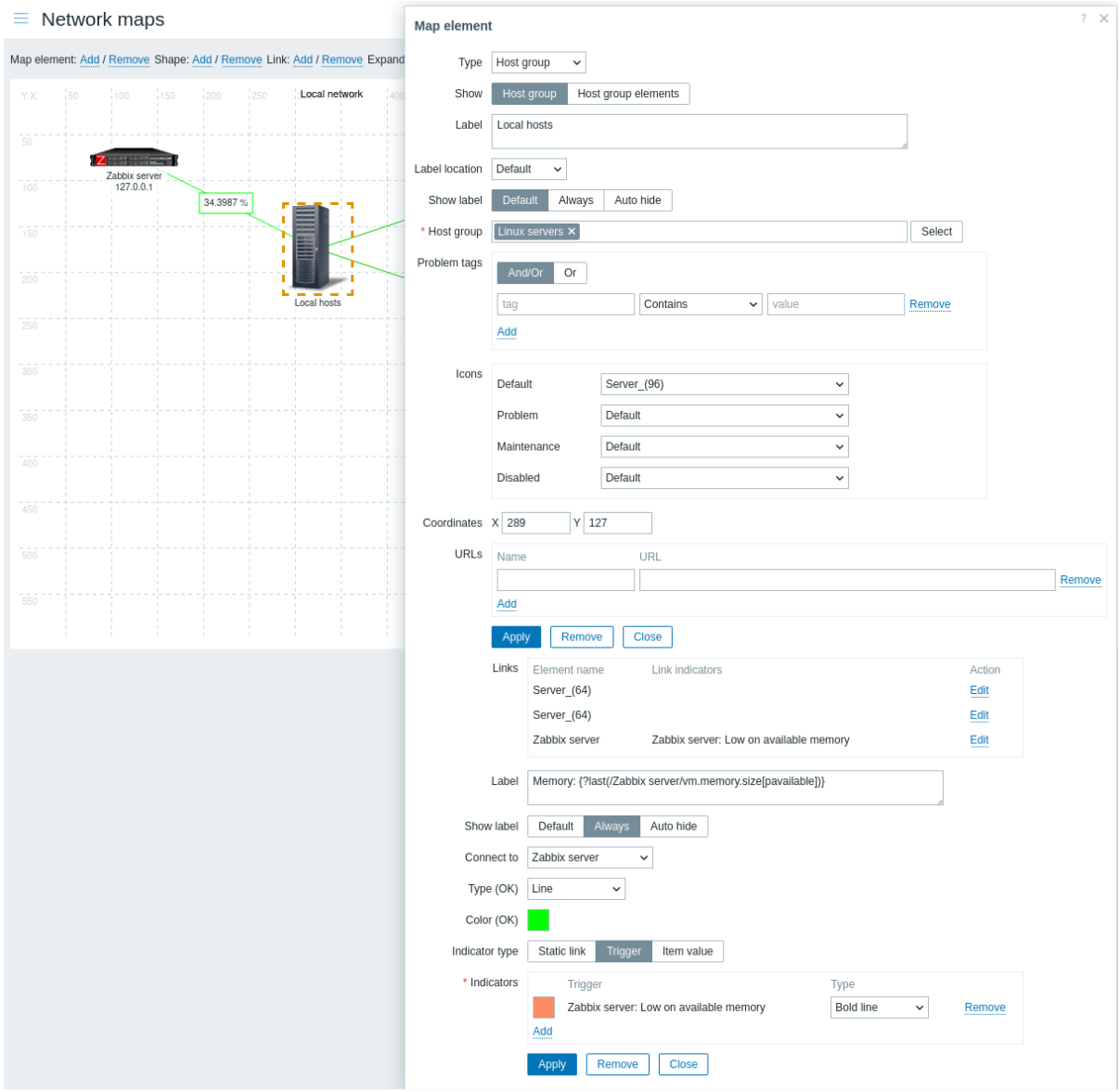
Da mesma forma, se um valor de item atingir um limite especificado ou corresponder a uma expressão regular especificada, o estilo do link poderá refletir isso.

Configuração

Triggers

Para atribuir triggers como indicadores de link, faça o seguinte:

- selecione um elemento do mapa
- clique em *Edit* na seção *Links* para o link apropriado
- selecione *Trigger* como o tipo de indicador
- clique em *Add* no bloco *Indicators* e selecione um ou mais triggers

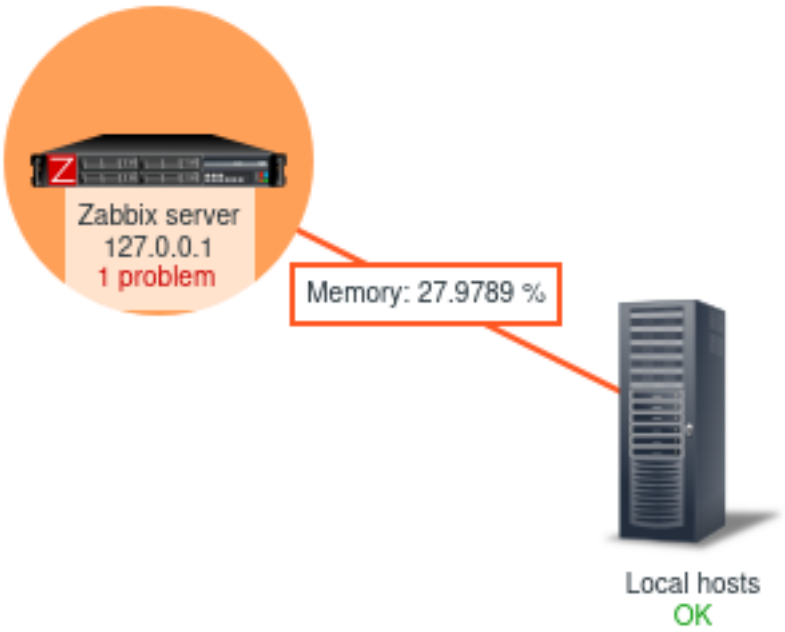


Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Os triggers adicionados podem ser vistos na lista *Indicators*.

Você pode definir o tipo de link e a cor de cada indicador diretamente na lista. Quando terminar, clique em *Apply*, feche o formulário e clique em *Update* para salvar as alterações do mapa.

Em *Monitoring* > *Maps*, o respectivo tipo de link e a cor serão exibidos se o **trigger** entrar em um estado de problema.



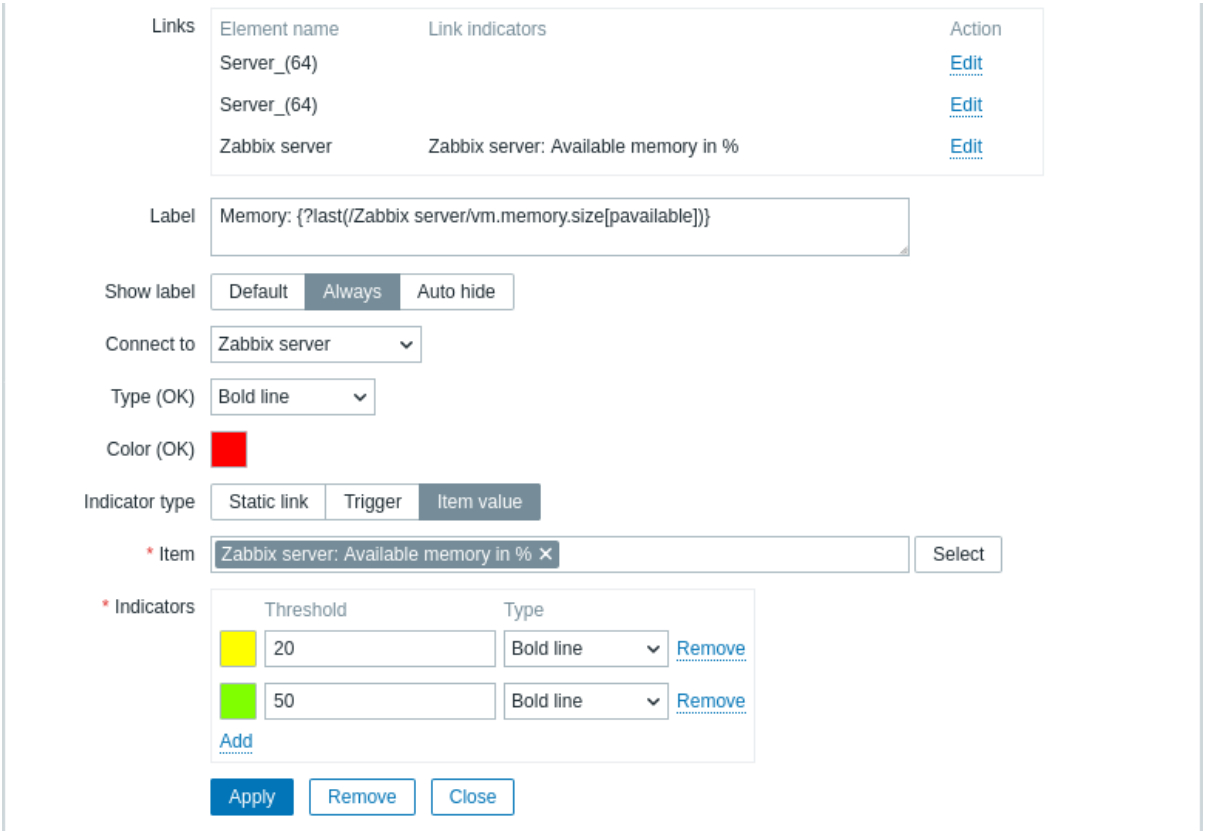
Note:
Se vários triggers entrarem em um estado de problema, o problema com a maior severidade determinará o estilo e a cor do link. Se vários triggers com a mesma severidade forem atribuídos ao mesmo link do mapa, aquele com o menor ID terá precedência. Observe também que:

1. As configurações *Minimum trigger severity* e *Show suppressed problem* da configuração do mapa afetam quais problemas são considerados.
2. No caso de triggers com múltiplos problemas (geração de múltiplos problemas), cada problema pode ter uma severidade diferente da severidade do trigger (alterada manualmente), pode ter tags diferentes (devido a macros) e pode estar suprimido.

Valores de item

Para atribuir valores de item como indicadores de link, faça o seguinte:

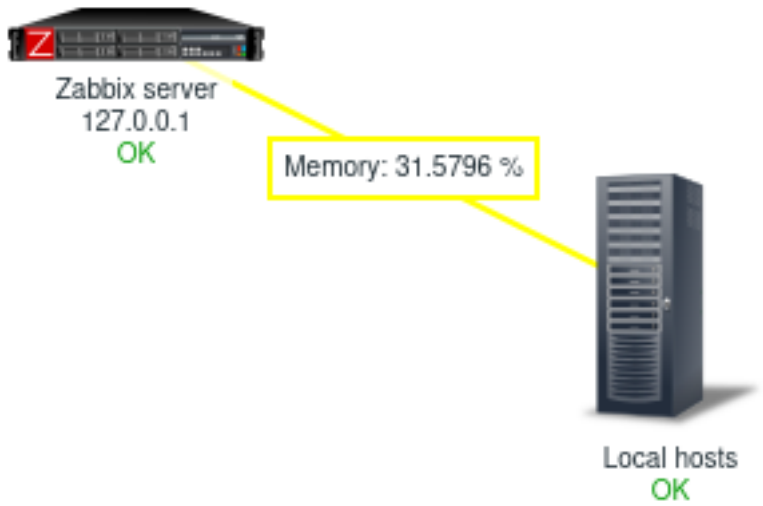
- selecione um elemento do mapa
- clique em *Edit* na seção *Links* para o link apropriado
- selecione *Item value* como o tipo de indicador
- selecione o item
- adicione um ou mais limites ou padrões de valor de item no bloco *Indicators*



Os limites/padrões de item adicionados podem ser vistos na lista *Indicators*.

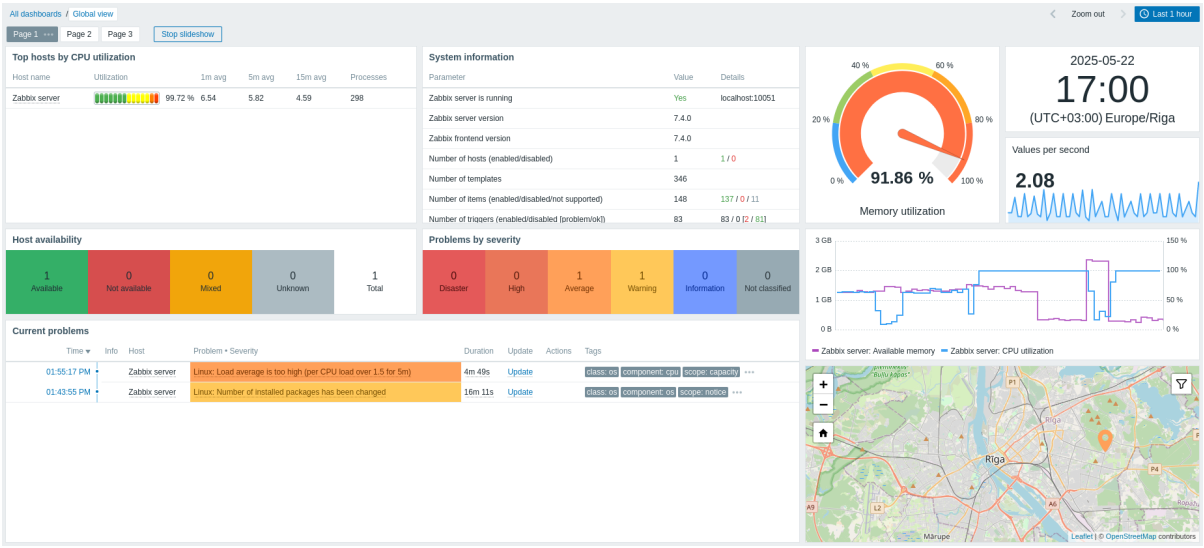
Você pode definir o tipo de link e a cor de cada indicador diretamente na lista. Quando terminar, clique em *Apply*, feche o formulário e clique em *Update* para salvar as alterações do mapa.

Em *Monitoring* > *Maps*, o respectivo tipo de link e a cor 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).



Painéis

Os painéis - tanto painéis globais quanto painéis de host - fornecem uma plataforma de visualização robusta com widgets e ferramentas como gráficos modernos, mapas, apresentações de slides e muito mais.



Templates e grupos de template

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.

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*.

Quando você atualiza o Zabbix, os templates não são atualizados automaticamente. Isso evita substituir quaisquer alterações personalizadas que você tenha feito. Para atualizar ou adicionar templates manualmente, consulte a seção *Atualização de templates* abaixo.

Use a barra lateral para acessar informações sobre tipos específicos de templates e requisitos de operação.

Os dashboards globais projetados para funcionar com templates existentes do Zabbix podem ser baixados do repositório de dashboards e importados para o seu ambiente Zabbix. Uma descrição detalhada de cada dashboard, incluindo os pré-requisitos, está disponível no arquivo `README` do dashboard.

Veja também:

- Vinculando um template
- Problemas conhecidos relacionados a templates

Atualização de templates

Para atualizar ou adicionar um **único template**:

1. Abra o diretório `templates` no repositório Git do Zabbix.

2. Escolha a branch (sob o rótulo *Source*) que corresponde à sua versão do Zabbix (por exemplo, *master* para a versão de desenvolvimento).

3. Localize 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-o em um arquivo local.

5. Na interface de usuário do Zabbix, acesse *Coleta de dados* > *Templates* e importe o arquivo local.

> **Note:**
> Os tipos de mídia Webhook podem ser atualizados de forma semelhante: obtenha-os do diretório `media` e importe em *Alertas* > *Tipos de mídia*.

Para atualizar ou adicionar **vários templates**:

1. Baixe o arquivo dos pacotes de templates de acordo com o seu web server:

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. Extraia o arquivo. Ele contém arquivos YAML, cada um com vários templates, e um arquivo de índice que lista quais templates estão em cada arquivo YAML.

3. Na interface de usuário do Zabbix, acesse *Coleta de dados* > *Templates* e importe os arquivos YAML um por um.

> **Note:**
> Os tipos de mídia Webhook podem ser atualizados de forma semelhante: obtenha o arquivo `media_bundle-8.0-000.yaml` do arquivo dos pacotes de templates e importe em *Alertas* > *Tipos de mídia*.

Para atualizar ou adicionar **todos os templates** de uma só vez, você pode usar o script `import_templates.sh`, incluído no arquivo dos pacotes de templates. O script requer os utilitários `curl` e `jq` e usa o método da API do Zabbix `configuration.import`. Ao executar o script, forneça o diretório que contém os arquivos YAML, a URL da API do Zabbix e o token de autenticação.

> **Warning:**
> O script `import_templates.sh` substitui todos os templates existentes, sem oferecer a opção de revisar ou desfazer as alterações nos templates.

Exemplo:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Compatibilidade do template com o Host Wizard

O Host Wizard é compatível com todos os templates prontos para uso.

Você também pode tornar seus templates personalizados compatíveis com o Host Wizard:

1. No frontend do Zabbix, acesse *Coleta de dados* > *Templates* e exporte o template personalizado.

2. Edite o arquivo exportado:

- Adicione o elemento de template `wizard_ready` para indicar a compatibilidade com o Host Wizard.
- Opcionalmente, adicione o elemento de template `readme` ou `config`, conforme necessário.

3. No frontend do Zabbix, acesse *Coleta de dados* > *Templates* e importe o arquivo de template atualizado.

Operação do template do Zabbix agent

Etapas para garantir a operação correta dos templates que coletam métricas com o Zabbix agent:

1. Certifique-se de que o Zabbix agent esteja instalado no host. Para verificações ativas, certifique-se também de que o endereço do Zabbix server/proxy esteja adicionado ao parâmetro 'ServerActive' do arquivo de configuração do agent.
2. Vincule o template a um host de destino (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros do template.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada de um template, incluindo a lista completa de macros, itens e triggers, está disponível no arquivo README do template (acessível clicando no nome de um template).

Os seguintes templates estão disponíveis:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Operação do template do Zabbix agent 2

Etapas para garantir o funcionamento correto dos templates que coletam métricas com Zabbix agent 2:

1. Certifique-se de que o agent 2 esteja instalado no host e de que a versão instalada contenha o plugin necessário. Em alguns casos, talvez seja preciso atualizar o agent 2 primeiro.
2. Vincule o template a um host de destino (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros do template. Observe que macros de usuário podem ser usadas para substituir parâmetros de configuração.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Attention:
Os templates do Zabbix agent 2 funcionam em conjunto com os plugins. Embora a configuração básica possa ser feita simplesmente ajustando macros de usuário, a personalização mais profunda pode ser obtida ao configurar o plugin em si. Por exemplo, se um plugin oferecer suporte a sessões nomeadas, é possível monitorar várias entidades do mesmo tipo (por exemplo, MySQL1 e MySQL2) especificando uma sessão nomeada com URI, nome de usuário e senha próprios 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 ao clicar no nome de um template).

Os seguintes templates estão disponíveis:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

Operação do template HTTP

Etapas para garantir o funcionamento correto dos templates que coletam métricas com o HTTP agent:

1. Crie um host no Zabbix que corresponda ao destino do monitoramento. Especifique o endereço IP/nome DNS do destino do monitoramento como a interface principal do host. Isso é necessário para que a macro `{HOST.CONN}` seja resolvida corretamente nos itens do template.
2. Vincule o template ao host criado na Etapa 1 (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros no nível do template.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada de um template, incluindo a lista completa de macros, itens e triggers, está disponível no arquivo `README` do template (acessível clicando no nome de um 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
- Claude API por HTTP
- ClickHouse por HTTP
- Cloudflare por HTTP
- CockroachDB por HTTP
- Control-M enterprise manager por HTTP
- Control-M server por HTTP
- Cradlepoint NCM v2 por HTTP
- DELL PowerEdge R720 por HTTP
- DELL PowerEdge R740 por HTTP
- DELL PowerEdge R820 por HTTP

- DELL PowerEdge R840 por HTTP
- Domain RDAP por HTTP
- Elasticsearch Cluster por HTTP
- Envoy Proxy por HTTP
- Etcd por HTTP
- FortiGate por HTTP
- GitHub repository por HTTP
- GitLab por HTTP
- GLPI 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
- Hikvision camera 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 estão obsoletos e foram substituídos pelo template *Kubernetes Cluster por HTTP*.
- Kubernetes Cluster 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
- Podman por HTTP
- Proxmox VE por HTTP
- Pure Storage FlashArray v1 and v2 por HTTP
- RabbitMQ cluster por HTTP
- Ribbon SBC Edge por HTTP
- Ribbon SBC SWe CE por HTTP
- Ribbon SBC SWe Core por HTTP
- TiDB por HTTP
- TiDB PD por HTTP
- TiDB TiKV por HTTP
- Travis CI por HTTP
- Veeam Backup Enterprise Manager por HTTP
- Veeam Backup and Replication por HTTP
- VeloCloud SD-WAN por HTTP
- VeloCloud SD-WAN Edge por HTTP
- YugabyteDB por HTTP
- ZooKeeper por HTTP

Operação do template IPMI

Os templates IPMI não exigem nenhuma configuração específica. Para iniciar o monitoramento, vincule o template a um host de destino (se o template não estiver disponível na 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 ao clicar no nome de um template).

Template disponível:

- Chassis by IPMI

Operação de template JMX

Etapas para garantir o funcionamento correto dos templates que coletam métricas por JMX:

1. Certifique-se de que o Java gateway do Zabbix esteja 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 na sua instalação do Zabbix, talvez seja necessário importar o template primeiro.
3. Se necessário, ajuste os valores das macros do template.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada de um template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível ao clicar no nome de um template).

Os seguintes templates estão disponíveis:

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX
- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX
- WildFly Domain by JMX
- WildFly Server by JMX

Operação do template ODBC

Etapas para garantir o funcionamento correto dos templates que coletam métricas via monitoramento ODBC:

1. Certifique-se de que o driver ODBC necessário esteja instalado no server ou proxy do Zabbix.
2. Vincule o template a um host de destino (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importá-lo 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 de um template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível ao clicar no nome do template).

Os seguintes templates estão disponíveis:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Templates padronizados para dispositivos de rede

Visão geral

Para fornecer monitoramento para dispositivos de rede, como switches e routers, criamos dois chamados models: para o próprio dispositivo de rede (sua chassis, basicamente) e para a interface de rede.

Templates para muitas famílias de dispositivos de rede são fornecidos. Todos os templates abrangem (quando possível obter estes itens do dispositivo):

- Monitoramento de falhas da chassis (fontes de alimentação, fans e temperatura, status geral)

- Monitoramento de desempenho da chassis (itens de CPU e memória)
- Coleta de inventário da chassis (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 na sua instalação do Zabbix, talvez seja necessário importar o template primeiro.

Se você estiver importando os novos templates out-of-the-box, talvez também queira atualizar a expressão regular global `@Network interfaces for discovery` para:

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

para filtrar interfaces loopback e null na maioria dos sistemas.

Dispositivos

Lista de famílias de dispositivos para as quais há templates disponíveis:

Nome do template	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certificado
Aruba CX 8300s by SNMP	HPE Aruba Networking	Série de switches Aruba CX	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certificado
Brocade FC SNMP	Brocade	Switches FC Brocade	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 by SNMP	Check Point	Firewall de próxima geração	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certificado
Ciena 3906 by SNMP	Ciena	CPE Ciena	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certificado

Nome do template	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	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 versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T e 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certificado
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certificado
D-Link DES_DGS Switch SNMP	D-Link	Switches DES/DGX	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
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certificado
FortiGate by 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

Nome do template	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
Huawei AR600 by SNMP	Huawei	Roteadores Huawei NetEngine série AR600	Huawei AR600/AR6100/AR6200/AR6300 series routers with VRP software (V300R019/V300R021/V300R023 or later)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificado
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificado
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certificado
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certificado
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Dispositivos Infiniband Intel/QLogic	Infiniband 12300		ICS-CHASSIS-MIB	Falha Inventário
Juniper SNMP	Juniper	Modelos MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certificado
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 Infiniband Mellanox	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certificado

Nome do template	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
MikroTik CCR<device model> SNMP	MikroTik	Roteadores MikroTik Cloud Core (série CCR)	Há templates dedicados separados 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	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (série CRS)	Há templates dedicados separados 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<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (série CSS)	Há templates dedicados separados 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 <device model> SNMP	MikroTik	MikroTik hEX	Há templates dedicados separados 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	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**Tags**
MikroTik netPower <device model> SNMP	MikroTik	MikroTik netPower	Há templates dedicados separados 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 <device model> SNMP	MikroTik	MikroTik PowerBox	Há templates dedicados separados disponíveis para MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certificado
MikroTik RB<device model> SNMP	MikroTik	Roteadores MikroTik série RB	Há templates dedicados separados 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	Fornecedor	Família de dispositivos	Modelos conhecidos	SO	MIBs usados	**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 by SNMP	Stormshield	Firewalls Stormshield Network Security (SNS)	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	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 sem fio Ubiquiti AirOS	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Desempenho
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Desempenho Inventário

Design do template

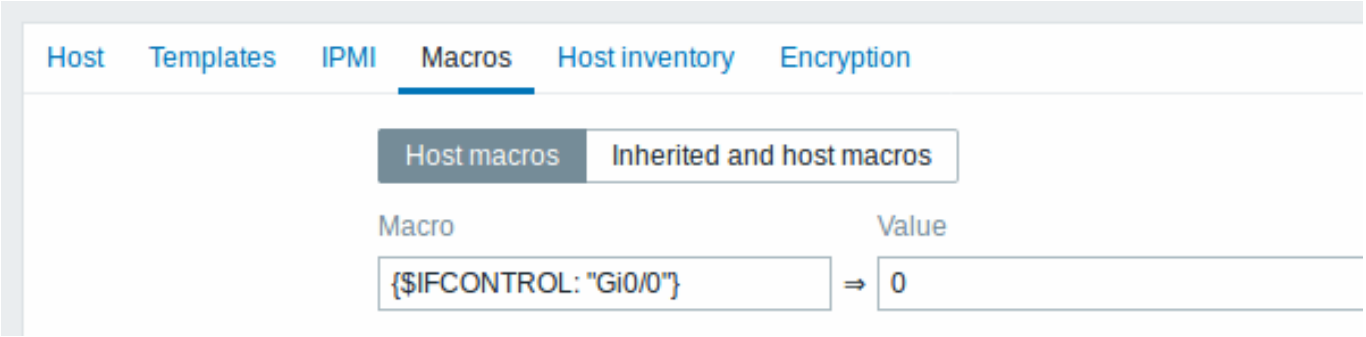
Os templates foram projetados com o seguinte em mente:

- 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:

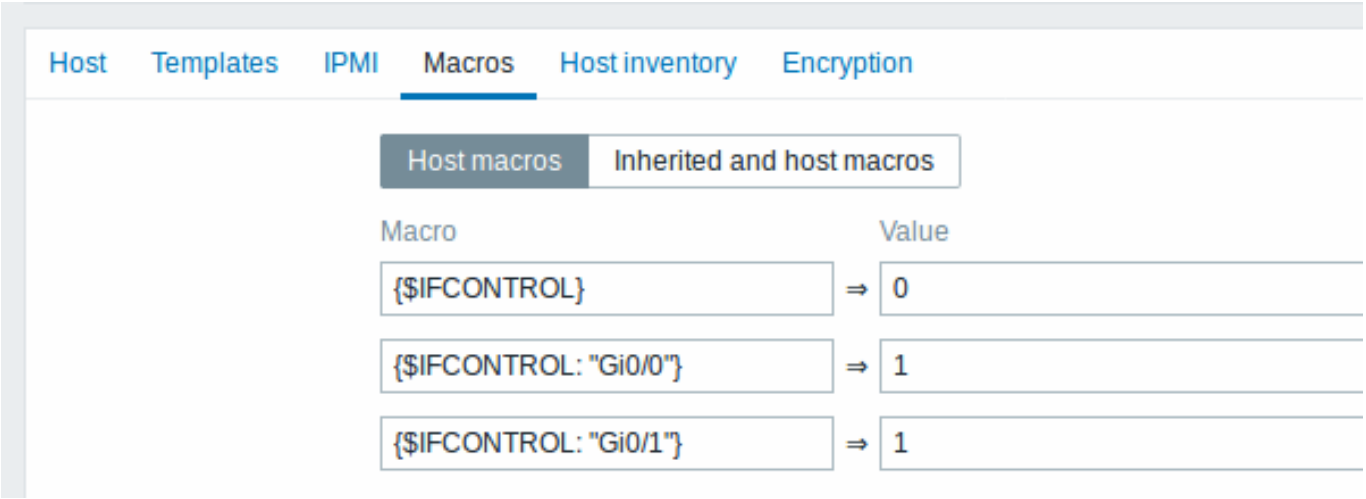
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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

- Desempenho - as MIBs da família de dispositivos fornecem uma forma de monitorar itens de CPU e memória;
- Falha - as MIBs da família de dispositivos fornecem uma forma de monitorar pelo menos um sensor de temperatura;
- Inventário - as MIBs da família de dispositivos fornecem uma forma de coletar pelo menos o número de série e o nome do modelo do dispositivo;
- Certificado - todas as três categorias principais acima são contempladas.

Operação do 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

Operação do template OTLP

O Zabbix fornece uma variedade de templates prontos para uso para coletar e monitorar traces, métricas e logs do OpenTelemetry usando itens *Telemetry query*.

Etapas para garantir a operação correta dos templates:

1. Verifique o arquivo `README` do template para consultar os requisitos de configuração específicos do template.
2. Crie um host no Zabbix que corresponda ao alvo do monitoramento.
3. Certifique-se de que o parâmetro `TelemetryProvider` esteja configurado no arquivo de configuração do server ou do proxy do Zabbix.
4. Vincule o template ao host criado (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importá-lo primeiro).
5. Se necessário, ajuste os valores das macros no nível do template.

Uma descrição detalhada dos templates, incluindo a lista completa de macros, itens e triggers, está disponível no arquivo `README` do template, que pode ser acessado clicando no nome do template.

Os seguintes templates estão disponíveis:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

Notificações após 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.

Tipos de mídia

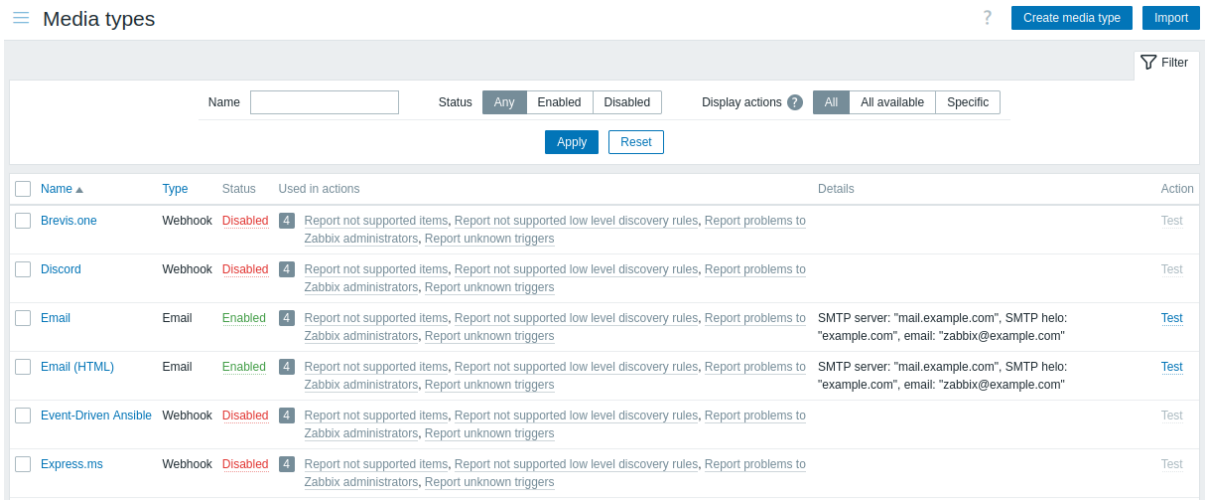
Visão geral

Os tipos de mídia são os canais de entrega usados para enviar notificações e alertas do Zabbix.

Os tipos de mídia são compatíveis com os seguintes métodos de entrega:

- E-mail
- Notificação push
- SMS
- Script personalizado
- Webhook

Os tipos de mídia são gerenciados em *Alertas* > *Tipos de mídia*. Alguns tipos de mídia vêm predefinidos no conjunto de dados padrão. Basta ajustar seus parâmetros para que funcionem.



Para entender como os tipos de mídia se encaixam no processo de alertas, vejamos os três requisitos para entregar notificações do Zabbix aos usuários finais:

1. Deve ser definida uma operação de ação que envie notificações.
2. Deve ser definido um tipo de mídia funcional, como *E-mail*, que envia alertas usando SMTP.
3. Os detalhes de entrega no nível do usuário, como endereços de e-mail, números de telefone etc., devem ser definidos em mídia do usuário.

Teste do tipo de mídia

Para testar se um tipo de mídia configurado funciona, clique em *Testar* na lista de tipos de mídia.

A solicitação de teste será enviada ao Zabbix server. O Zabbix server tentará enviar um alerta usando o tipo de mídia especificado e retornará o resultado ao frontend. O frontend aguardará o retorno dos resultados pelo server. O teste do tipo de mídia tem um tempo limite padrão de 65 segundos (configurável em *Administração* > *Geral* > *Tempos limite*).

Consulte também os detalhes de teste para:

- E-mail
- Notificação push
- Webhook
- Script

Configuração

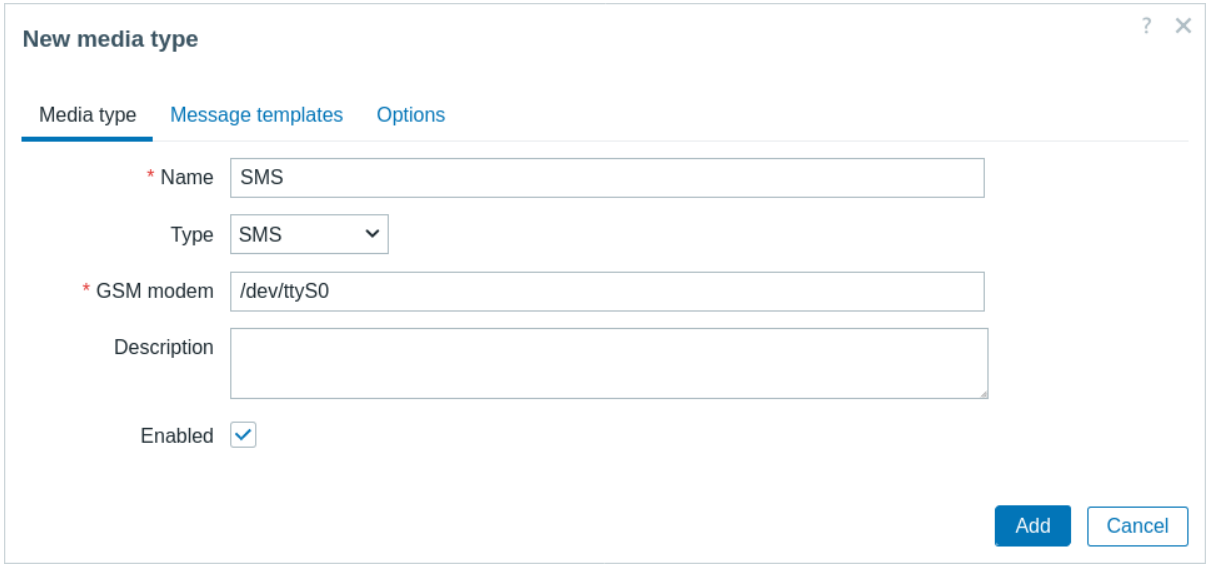
Para criar um tipo de mídia no frontend do Zabbix:

1. Acesse *Alertas* > *Tipos de mídia*.

2. Clique em *Criar tipo de mídia*.
3. Insira os parâmetros do tipo de mídia no formulário.

Alguns parâmetros são **comuns** a todos os métodos de entrega.

Note:
Se um ou mais métodos de entrega forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend (`zabbix.conf.php`), esses métodos de entrega não estarão disponíveis na lista *Tipo*.

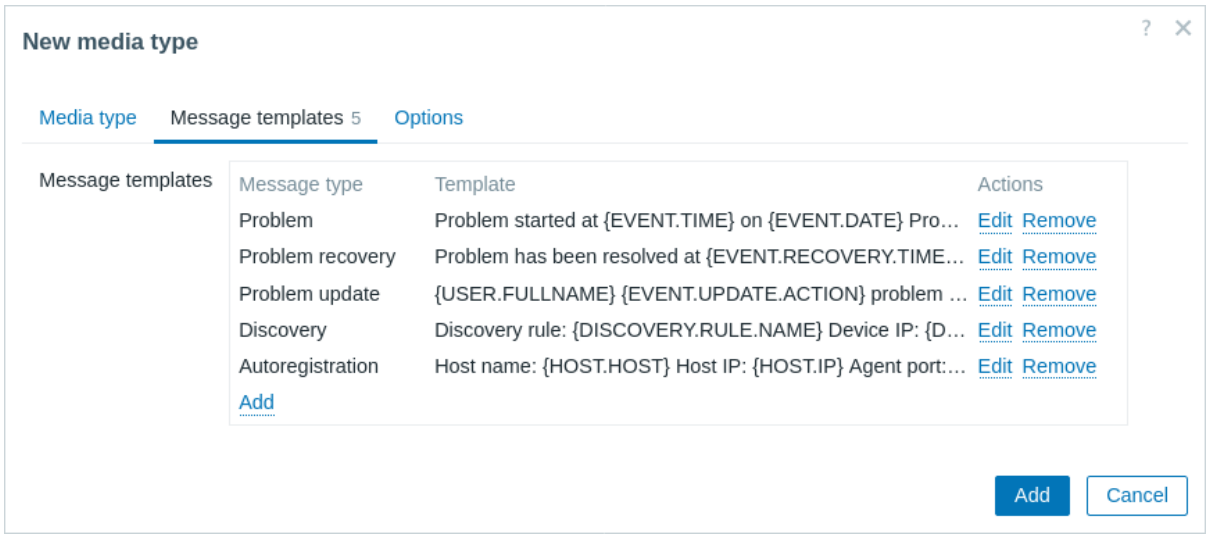


Parâmetro	Descrição
Nome	Nome do tipo de mídia.
Tipo	Selecione o método de entrega para o tipo de mídia.
Descrição	Insira uma descrição para o tipo de mídia.
Habilitado	Marque a caixa de seleção para habilitar o tipo de mídia.

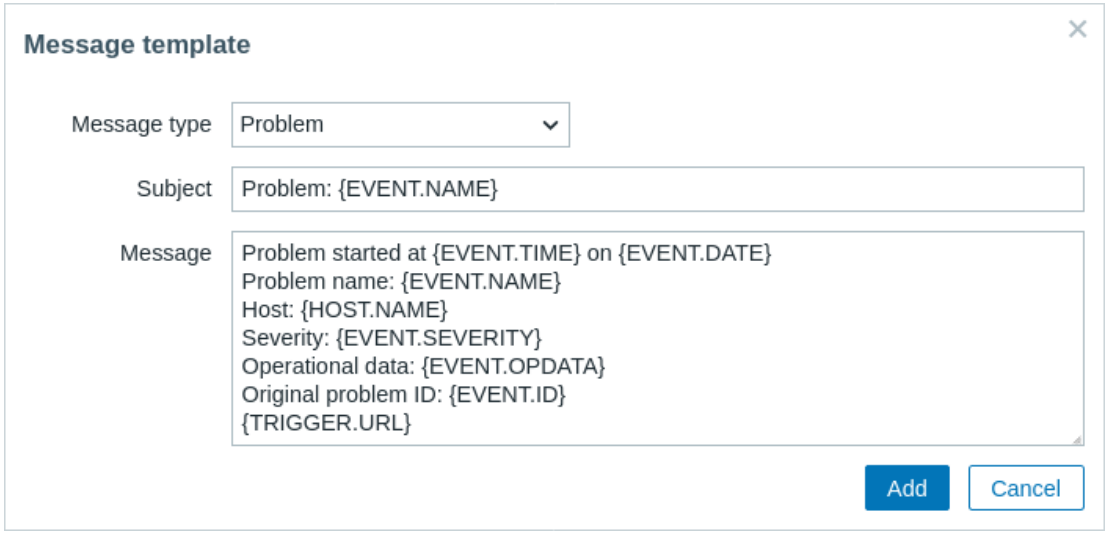
Para obter informações sobre os parâmetros específicos de cada método, consulte as páginas de e-mail, notificação push, SMS, alertscript personalizado ou webhook.

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 em **Editar** para atualizar uma mensagem existente):



Parameter	Description
Message type	Tipo de evento para o qual a mensagem padrão deve ser usada. Apenas uma mensagem padrão pode ser definida para cada tipo de evento.
Subject	Assunto da mensagem padrão. O assunto pode conter macros. Ele é limitado a 255 caracteres. O assunto não está disponível para o tipo de mídia SMS.
Message	A mensagem padrão. Ela é limitada a uma certa quantidade de caracteres, dependendo do tipo de banco de dados (consulte Sending messages 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)}`).

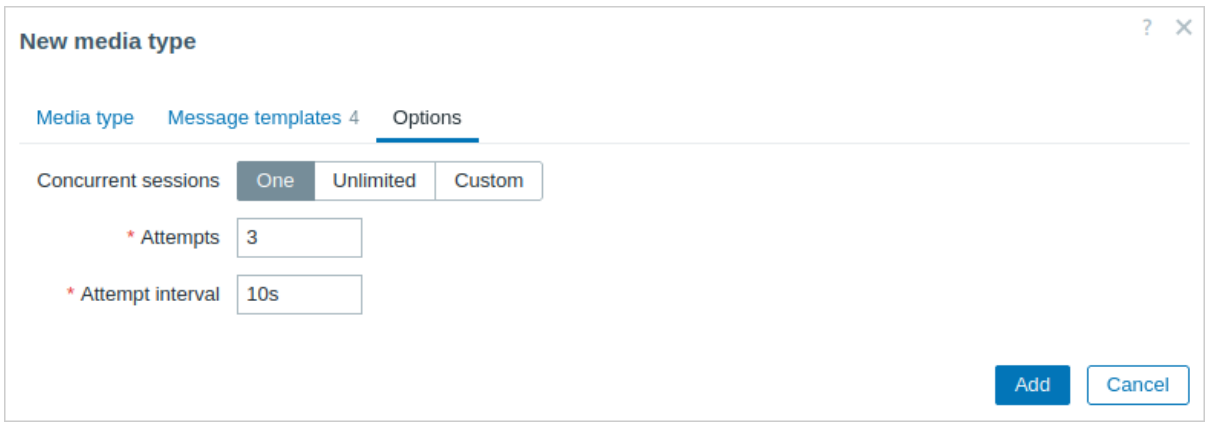
Observe que as mensagens padrão são substituídas por mensagens personalizadas, se definidas em action operations.

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 action "Send message to Pushover webhook" falhará ao enviar notificações de problema, se a mensagem de problema para o Pushover webhook não estiver definida.

Opções

A aba **Opções** contém configurações de processamento de alertas.
O mesmo conjunto de opções pode ser configurado para cada tipo de mídia.

Todos os tipos de mídia são processados em paralelo.
Embora o número máximo de sessões simultâneas possa ser configurado por tipo de mídia,
o número total de processos alerter no server só pode ser limitado pelo parâmetro StartAlerters.
Os alertas gerados por um trigger são processados sequencialmente.
Portanto, várias notificações só poderão ser processadas simultaneamente se forem geradas por vários triggers.



Parâmetro	Descrição
Sessões simultâneas	Selecione o número de sessões alerter paralelas para o tipo de mídia: **Uma** - uma sessão **Ilimitado** - número ilimitado de sessões **Personalizado** - selecione um número personalizado de sessões Valores ilimitados/altos significam mais sessões paralelas e maior capacidade para enviar notificações. Valores ilimitados/altos devem ser usados em ambientes grandes, nos quais muitas notificações podem precisar ser enviadas simultaneamente. Se for necessário enviar mais notificações do que o número de sessões simultâneas, as notificações restantes serão colocadas em fila; elas não serão perdidas.
Tentativas	Número de tentativas para enviar uma notificação. É possível especificar até 100 tentativas; 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 entre tentativas	Frequência das tentativas de reenviar uma notificação caso o envio falhe, em segundos (0-3600). Se '0' for especificado, o Zabbix tentará novamente imediatamente. Sufixos de tempo são compatíveis (por exemplo, 30s, 1m, 1h).

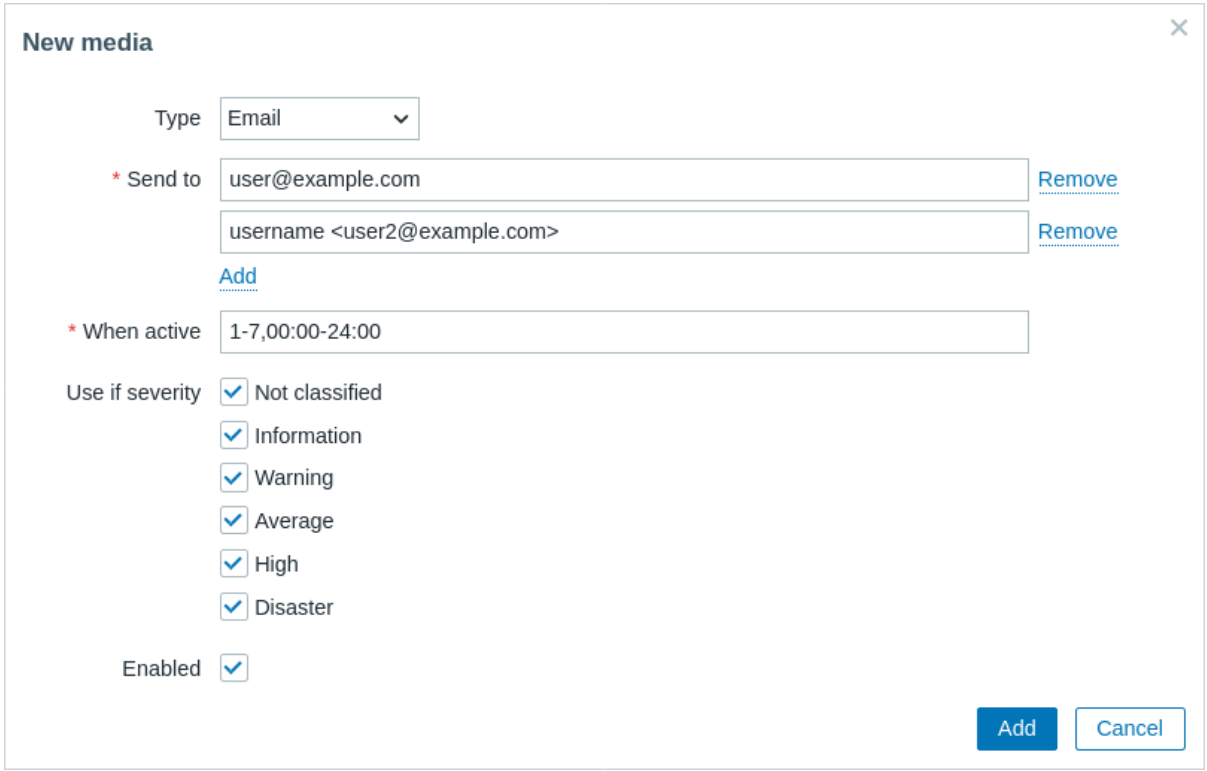
Mídia do usuário

Enquanto os tipos de mídia definem **como** uma notificação será enviada, a mídia do usuário define **para onde** a notificação deve ser enviada.

A mídia do usuário (por exemplo, endereço de e-mail, ID de usuário do webhook etc.) deve ser definida no perfil do usuário, independentemente do método de entrega. Uma ação que envia mensagens ao usuário *Admin* usando o webhook *X* não conseguirá entregá-las se os detalhes de entrega do webhook *X* não estiverem definidos no perfil do usuário Admin.

Para definir a mídia do usuário:

1. Acesse *Usuários* > *Usuários* e abra o formulário de propriedades do usuário. Como alternativa, acesse *Configurações do usuário* > *Notificações* no seu próprio perfil de usuário.
2. Clique em *Adicionar* na aba *Mídia*.
3. Insira os parâmetros da mídia do usuário no formulário.



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.

Parâmetro	Descrição
Enviar para	Insira as informações de contato para onde as mensagens devem ser enviadas. Para o tipo de mídia de e-mail, vários endereços podem ser adicionados clicando no botão Add abaixo do campo de endereço. Nesse caso, as notificações serão enviadas para todos os endereços listados. Para obter exemplos de endereços, consulte a descrição do parâmetro *E-mail* para o tipo de mídia e-mail. Para o tipo de mídia de notificação push, você pode selecionar entre os dispositivos móveis ativos do usuário ou adicionar um ou mais dispositivos específicos.
Quando ativo	Você pode limitar o horário em que as mensagens são enviadas, por exemplo, definindo 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, talvez seja necessário ajustar esse limite para não perder mensagens importantes. Consulte a página Especificação de períodos de tempo para obter uma descrição do formato. Macros de usuário são compatíveis.
Usar se a severidade for	Marque as caixas de seleção das severidades de trigger para as quais deseja receber notificações. *Observação*: a severidade padrão ('Não classificado') **deve** estar marcada se você quiser receber notificações para eventos que não sejam de trigger. Após salvar, as severidades de trigger selecionadas serão exibidas nas cores correspondentes, enquanto as não selecionadas ficarão esmaecidas.
Status	Status da mídia do usuário. **Habilitado** - está em uso. **Desabilitado** - não está sendo usado.

Email

Visão geral

Para configurar o e-mail como canal de entrega de mensagens, é necessário configurar o e-mail como tipo de mídia e atribuir endereços específicos aos usuários.

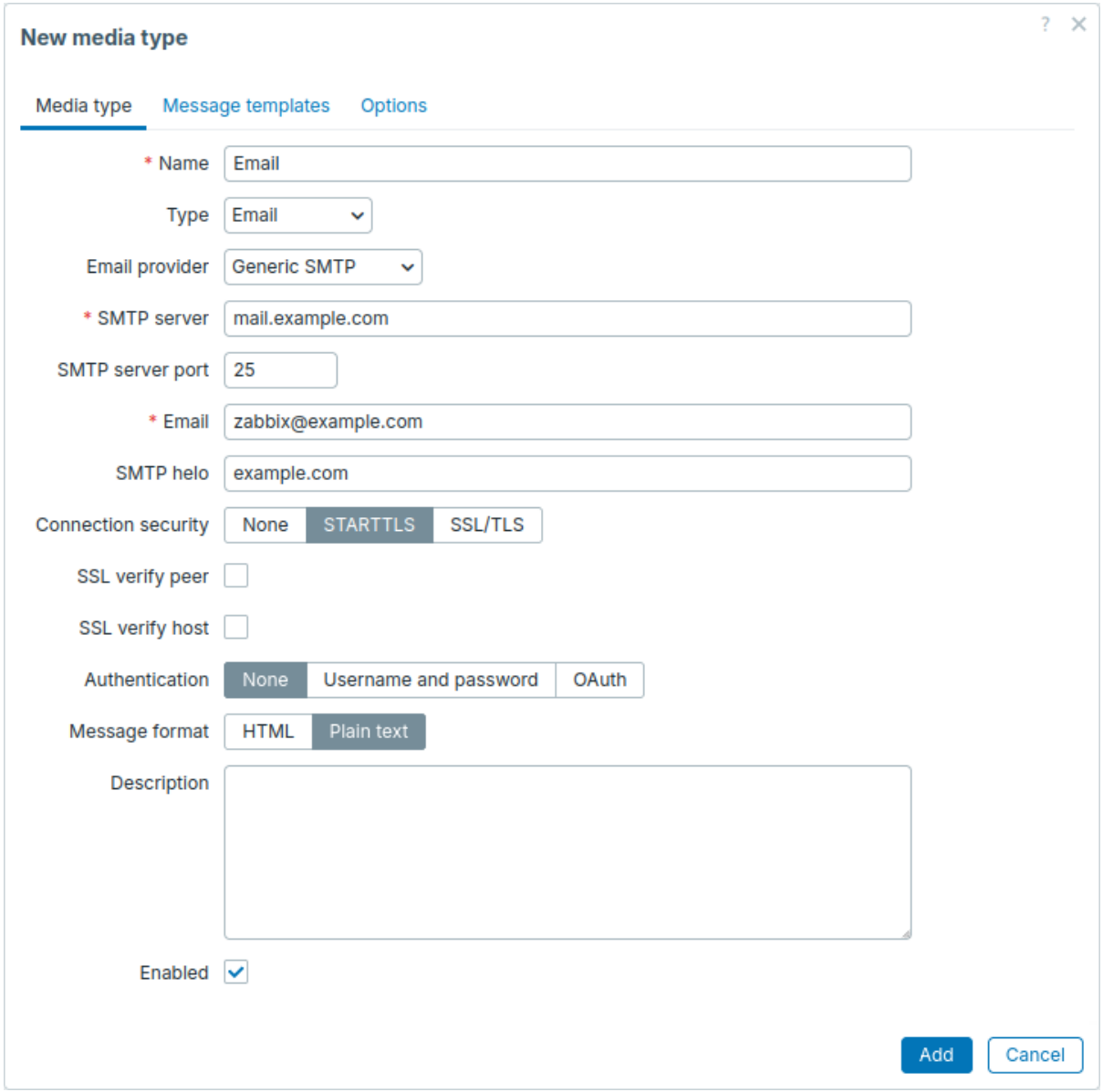
> **Note:**
> Várias notificações para um único evento serão agrupadas no mesmo tópico de e-mail.

Configuração

Para configurar o email como tipo de mídia:

1. Acesse *Alerts* > *Media types*.
2. Clique em *Create media type* (ou clique em *Email* na lista de tipos de mídia predefinidos).
3. Insira os parâmetros do tipo de mídia de email no formulário.

A aba **Media type** contém os atributos gerais do tipo de mídia:



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Warning:
A senha é exportada em texto simples ao exportar a configuração do tipo de mídia de email.

Os parâmetros a seguir são específicos do tipo de mídia de email:

Parâmetro	Descrição
Email provider	Selecione o provedor de email: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* ou *Office365 relay*. Se você selecionar as opções relacionadas ao Gmail/Office365, será necessário fornecer apenas o endereço de email do remetente e a senha; opções como *SMTP server*, *SMTP server port*, *SMTP helo* e *Connection security* serão preenchidas automaticamente pelo Zabbix. Consulte também: Automated Gmail/Office365 media types.
SMTP server	Defina um servidor SMTP para processar as mensagens de saída. Este campo estará disponível se *Generic SMTP* for selecionado como provedor de email.
SMTP server port	Defina a porta do servidor SMTP para processar as mensagens de saída. Este campo estará disponível se *Generic SMTP* for selecionado como provedor de email.

Parâmetro	Descrição
Email	O endereço inserido aqui será usado como endereço **From** nas mensagens enviadas. É possível adicionar um nome de exibição do remetente junto com o endereço de email real (por exemplo, `Zabbix_info <zabbix@company.com>`). Há algumas restrições para nomes de exibição nos emails do Zabbix em comparação com os permitidos pela RFC 5322, conforme ilustrado nos exemplos a seguir. Exemplos válidos: `zabbix@company.com` (somente um endereço de email; os sinais de menor e maior não são obrigatórios) `Zabbix_info <zabbix@company.com>` (um nome de exibição seguido pelo endereço de email entre sinais de menor e maior) `Ω-monitoring <zabbix@company.com>` (caracteres UTF-8 no nome de exibição) Exemplos inválidos: `Zabbix HQ zabbix@company.com` (há um nome de exibição, mas não há sinais de menor e maior ao redor do endereço de email) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (embora seja válido pela RFC 5322, pares entre aspas e comentários não são compatíveis com emails do Zabbix).
SMTP helo	Defina um valor correto para SMTP helo, normalmente um nome de domínio. Se estiver vazio, o nome de domínio do email será enviado (ou seja, o que vem depois de `@` no campo *Email*). Se não for possível obter o nome de domínio, um aviso de nível de depuração será registrado e o nome do host do server será enviado como domínio para o comando HELO. Este campo estará disponível se *Generic SMTP* for selecionado como provedor de email.
Connection security	Selecione o nível de segurança da conexão: **None** - não usar a opção CURLOPT_USE_SSL **STARTTLS** - usar a opção CURLOPT_USE_SSL com o valor CURLUSESSL_ALL **SSL/TLS** - o uso de CURLOPT_USE_SSL é opcional.
SSL verify peer	Marque a caixa de seleção para verificar o certificado SSL do servidor SMTP. O valor da diretiva de configuração do server "SSLCALocation" deve ser inserido em CURLOPT_CAPATH para a validação do certificado. Isso define a opção do cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	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 do cURL CURLOPT_SSL_VERIFYHOST.
Authentication	Selecione o nível de autenticação: **None** - nenhuma opção do cURL é definida **Username and password** - implica "AUTH=*", deixando a escolha do mecanismo de autenticação para o cURL **OAuth** - autenticação OAuth A autenticação OAuth não é compatível com o provedor de email *Office365 relay*.
Username	Nome de usuário a ser usado na autenticação. Isso define o valor de CURLOPT_USERNAME. Macros de usuário compatíveis.
Password	Senha a ser usada na autenticação. Isso define o valor de CURLOPT_PASSWORD. Macros de usuário compatíveis. O valor inserido aqui é exportado como está quando a configuração do tipo de mídia é exportada. Para evitar expor credenciais confidenciais nos arquivos exportados, use uma macro de usuário (por exemplo, `{$EMAIL_SMTP_PASSWORD}`) em vez de uma senha literal — a referência à macro será exportada, enquanto o valor secreto deverá ser (re)configurado no sistema de destino.
OAuth tokens	Clique em *Configure* para configurar os parâmetros de obtenção de OAuth tokens em uma nova janela. Este campo estará disponível somente se "OAuth" for selecionado no campo *Authentication*.
Message format	Selecione o formato da mensagem: **HTML** - enviar como HTML; **Plain text** - enviar como texto simples.

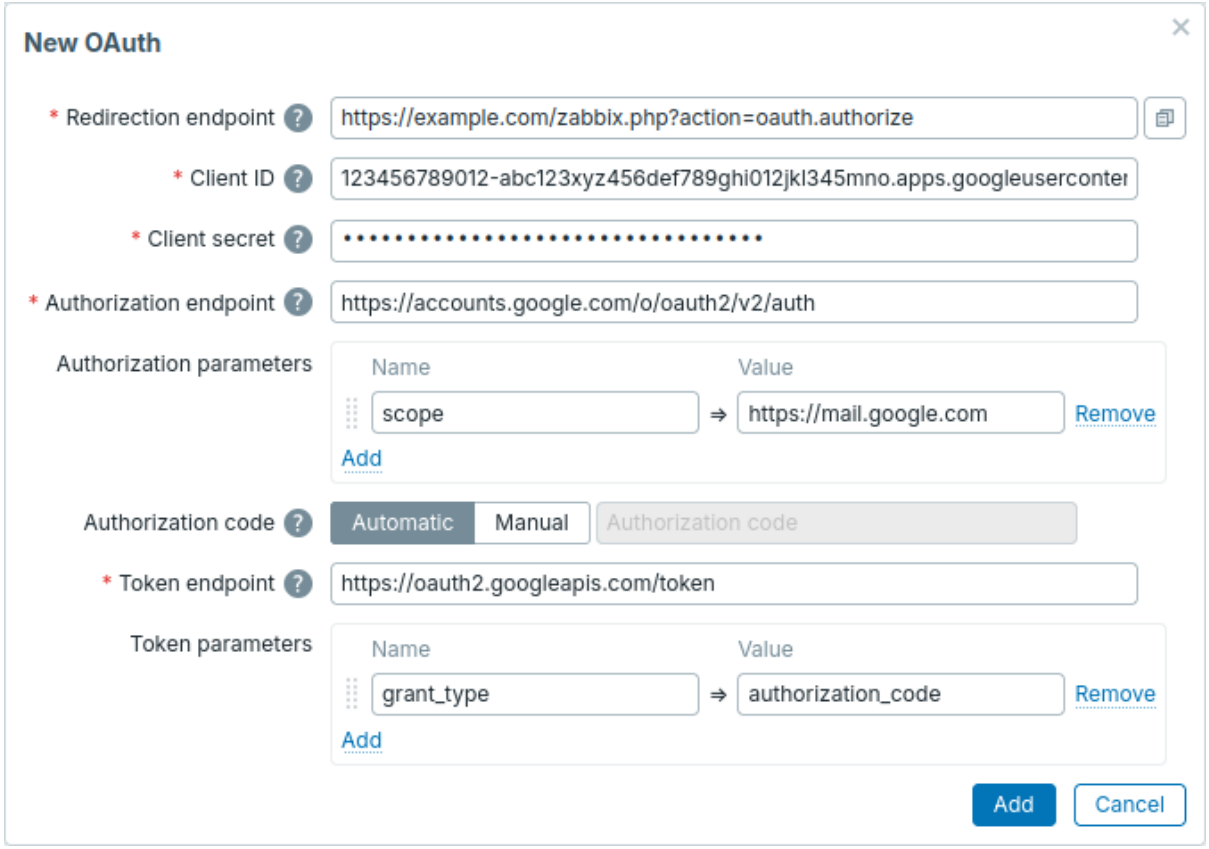
Attention:
Para habilitar a autenticação SMTP, o server do Zabbix deve ser compilado com a opção de compilação `--with-libcurl`, que requer libcurl (consulte os requisitos do server para obter detalhes sobre a versão).

Consulte também os parâmetros comuns de tipos de mídia para obter detalhes sobre como configurar mensagens padrão e opções

de processamento de alertas.

Tokens OAuth

Os seguintes parâmetros são necessários para recuperar tokens OAuth:



Warning:
O segredo do cliente é exportado em texto claro ao exportar a configuração do tipo de mídia de e-mail.

Parameter	Description
Redirection endpoint	Informe a URL do frontend do Zabbix para onde o serviço OAuth redirecionará de volta a autorização OAuth (usando o formato `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Ela é definida automaticamente para um novo tipo de mídia de e-mail quando a URL do frontend é definida.
Client ID	Informe o identificador exclusivo do aplicativo cliente registrado no servidor de autorização OAuth.
Client secret	Informe o segredo privado do aplicativo cliente registrado no servidor de autorização OAuth.
Authorization endpoint	Informe a URL do servidor de autorização OAuth para solicitar a autorização do usuário.
Authorization parameters	Informe os parâmetros para o endpoint de autorização.
Authorization code	Informe o código de autorização: **Automatic** - o código será recuperado automaticamente por meio de uma página de redirecionamento; **Manual** - informe o código manualmente se a recuperação automática falhar.
Token endpoint	Informe a URL do servidor de autorização OAuth para trocar o código de autorização por tokens de acesso e atualização.
Token parameters	Informe 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**. Nesse caso, é necessário informar apenas os valores dos parâmetros *Redirection endpoint*, *Client ID* e *Client secret*. O Zabbix preencherá automaticamente os demais valores necessários (consulte Padrões de URL OAuth por provedor).

Recuperação do token de acesso e atualização do OAuth

Após o envio dos parâmetros OAuth:

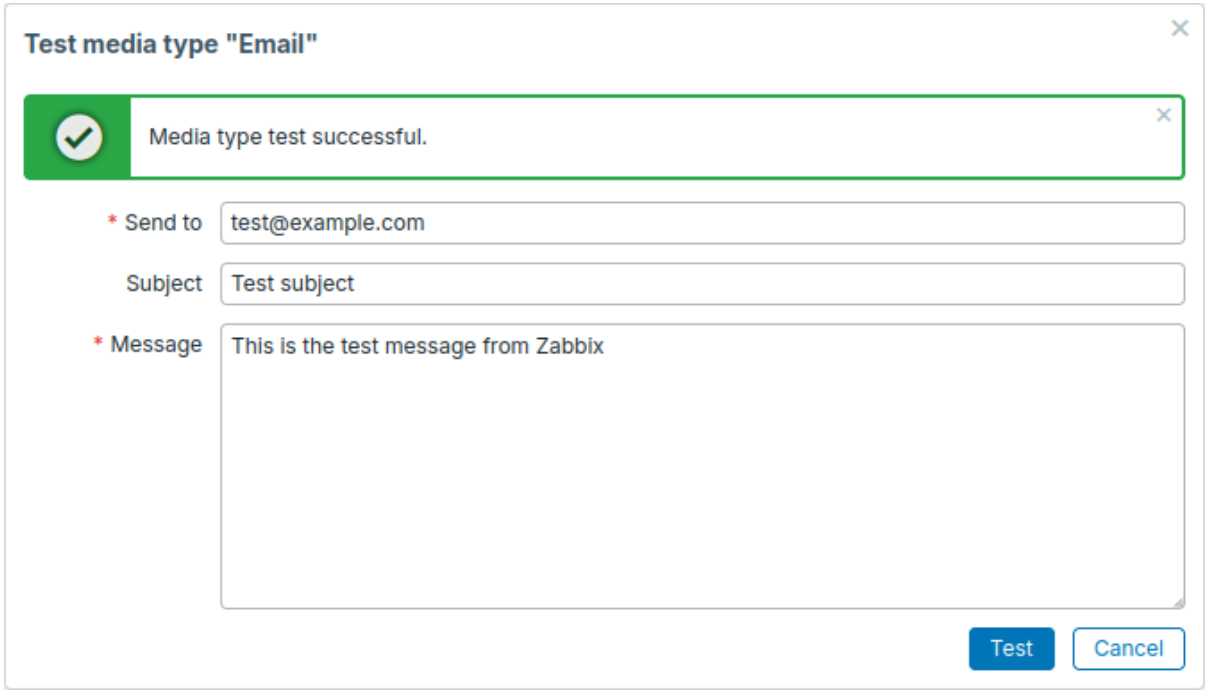
1. Uma janela pop-up do navegador é aberta, direcionando o usuário para o *ponto de extremidade de autorização*.
2. O usuário autoriza o Zabbix no serviço OAuth.
3. O serviço OAuth redireciona o usuário para a ação do frontend do Zabbix `oauth.authorize` com o código de autorização e o valor do escopo.
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.

Teste

Para testar se um tipo de mídia de email configurado funciona corretamente:

1. Localize o email relevante na lista de tipos de mídia.
2. Clique em *Test* na última coluna da lista (uma janela de teste será aberta).
3. Insira um endereço de destinatário em *Send to*, o corpo da mensagem e, opcionalmente, o assunto.
4. Clique em *Test* para enviar uma mensagem de teste.

A mensagem de sucesso ou falha do teste será exibida na mesma janela:



Mídia do usuário

Depois que o tipo de mídia de e-mail estiver configurado, vá para a seção *Users* > *Users* e edite o perfil do usuário para atribuir a mídia de e-mail ao usuário. As etapas para configurar a mídia do usuário, sendo comuns a todos os tipos de mídia, estão descritas na página Media types.

Tipos de mídia automatizados do Gmail/Office365

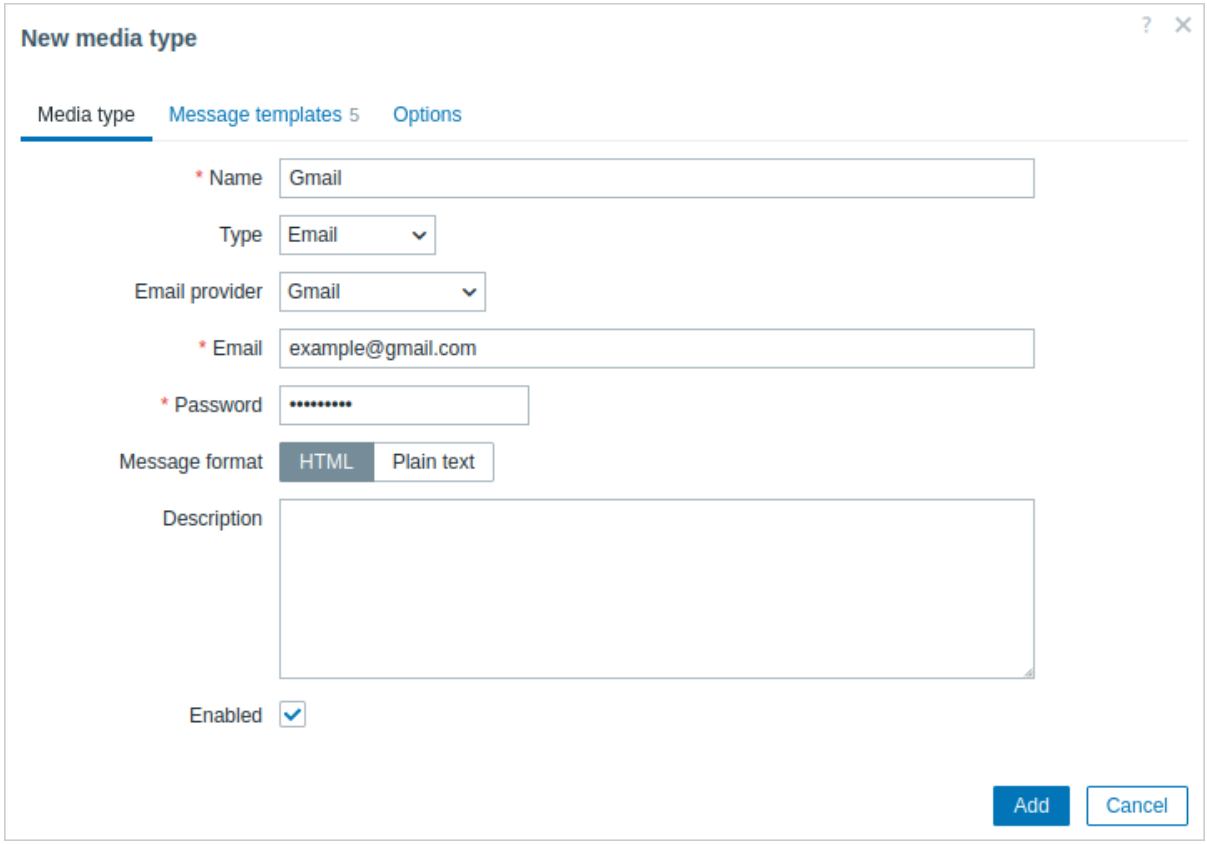
Visão geral

Usuários do Gmail ou Office365 podem se beneficiar de recursos automatizados na configuração do tipo de mídia.

Configuração

O campo *Provedor de email* na configuração do tipo de mídia de email permite selecionar opções pré-configuradas para Gmail e Office 365.

Ao selecionar as opções relacionadas ao Gmail/Office365, é necessário fornecer apenas o endereço de email e a senha do remetente para criar um tipo de mídia funcional.



Assim que o endereço de email e a senha forem fornecidos, o Zabbix poderá preencher automaticamente todas as configurações necessárias para os tipos de mídia Gmail/Office365 com os valores reais/recomendados, ou seja, *server SMTP*, *porta do server SMTP*, *helo SMTP* e *segurança da conexão*. Devido a essa automação, esses campos nem sequer são exibidos. No entanto, é possível ver o server SMTP e os detalhes do email na lista de tipos de mídia (consulte 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 email fornecido será usado para preencher dinamicamente o server 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 end-point	`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
```

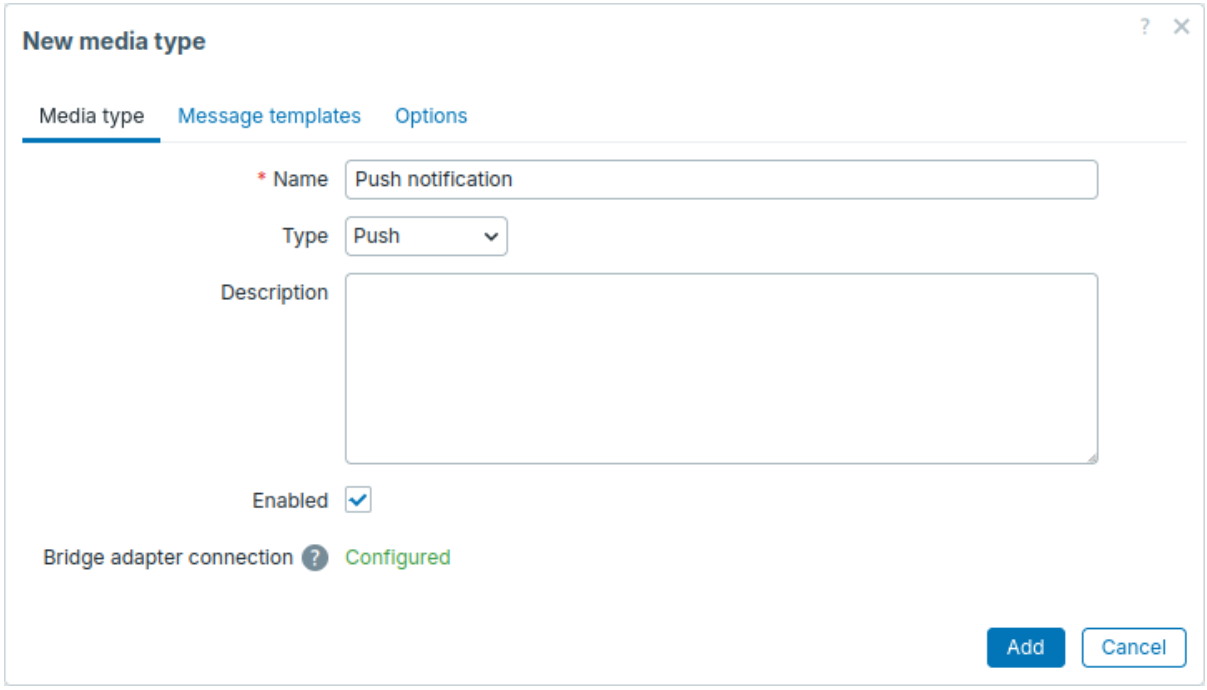
em vez de uma URL com parâmetros de consulta, garantindo a compatibilidade com contas pessoais do Office365.

Notificação push

Este tipo de mídia permite que o Zabbix server envie notificações push para o aplicativo móvel do Zabbix em seu dispositivo móvel.

Configuração

Ao criar um tipo de mídia, selecione *Push* como *Type*:



O formulário de configuração contém parâmetros, modelos de mensagem e opções comuns do tipo de mídia.

O formulário também mostra se o adaptador do Zabbix bridge está configurado.
 Se o adaptador não estiver configurado, ainda será possível criar o tipo de mídia, mas ele não funcionará.

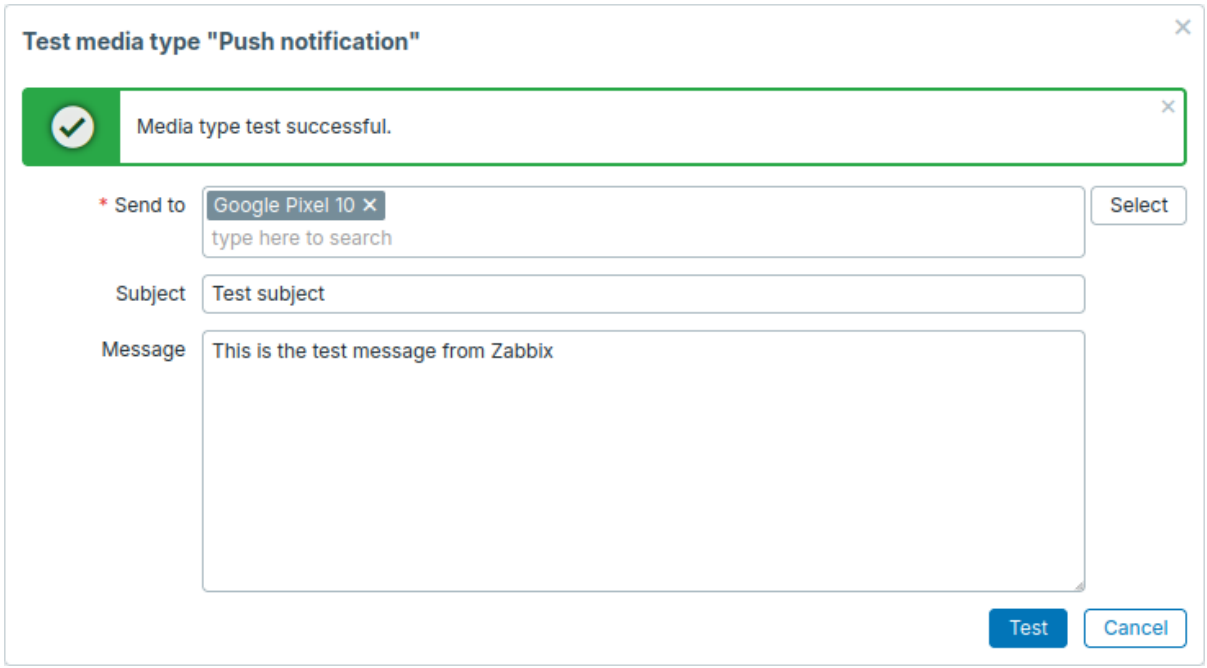
Depois de criar e habilitar o tipo de mídia, adicione um dispositivo para um usuário para adicionar automaticamente o tipo de mídia a esse usuário (ou adicione mídia do usuário manualmente). Isso permite que ele receba notificações push em seu dispositivo móvel.

Teste

Para testar se este tipo de mídia funciona:

1. Encontre o tipo de mídia na lista de tipos de mídia.
2. Clique em *Testar* na última coluna da lista para abrir o formulário de teste.
3. No campo *Enviar para*, selecione um dispositivo.
4. Clique em *Testar* para enviar uma notificação push de teste ao dispositivo selecionado.

Uma mensagem de sucesso ou falha será exibida na mesma janela:



SMS

Visão geral

O Zabbix oferece suporte ao envio de mensagens SMS usando um modem GSM serial conectado à porta serial do server Zabbix.

Certifique-se de que:

- A velocidade do dispositivo serial (normalmente /dev/ttyS0 no Linux) corresponda à do modem GSM. O Zabbix não define a velocidade do link serial. Ele usa as configurações padrão.
- O usuário 'zabbix' tenha 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 tenha o PIN inserido e o preserve após a reinicialização de energia. Como alternativa, você pode desabilitar o PIN no cartão SIM. O PIN pode ser inserido emitindo o comando AT+CPIN="NNNN" (NNNN é o número do seu PIN, as aspas devem estar presentes) em um software de terminal, como o Unix minicom ou o Windows HyperTerminal.

O Zabbix foi testado com estes modems GSM:

- Siemens MC35
- Teltonika ModemCOM/G10

Para configurar SMS como o canal de entrega de mensagens, você também precisa configurar SMS como o tipo de mídia e inserir os respectivos números de telefone dos usuários.

Configuração

Para configurar SMS como o tipo de mídia:

1. Acesse *Alertas* > *Tipos de mídia*.
2. Clique em *Criar tipo de mídia* (ou clique em *SMS* na lista de tipos de mídia predefinidos).
3. Insira os parâmetros do tipo de mídia SMS no formulário.

Os parâmetros a seguir são específicos do 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 server SMSDevices (se especificado).

Consulte os parâmetros comuns dos tipos de mídia para obter detalhes sobre como configurar mensagens padrão e opções de processamento de alertas. Observe que não é possível processar em paralelo o envio de notificações por SMS.

Mídia do usuário

Depois de configurar o tipo de mídia SMS, acesse a seção *Usuários* > *Usuários* e edite o perfil do usuário para atribuir a mídia SMS a ele. As etapas para configurar a mídia do usuário, que são comuns a todos os tipos de mídia, estão descritas na página Tipos de mídia.

Scripts de alerta personalizados

Visão geral

Se você não estiver satisfeito com os tipos de mídia existentes para enviar alertas, há uma forma alternativa de fazer isso. Você pode criar um script que tratará a notificação do seu jeito.

Scripts de alerta personalizados são executados no server do Zabbix. Esses scripts devem estar localizados no diretório especificado no parâmetro AlertScriptsPath do arquivo de configuração do server.

Aqui está um exemplo de um 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 com falha.

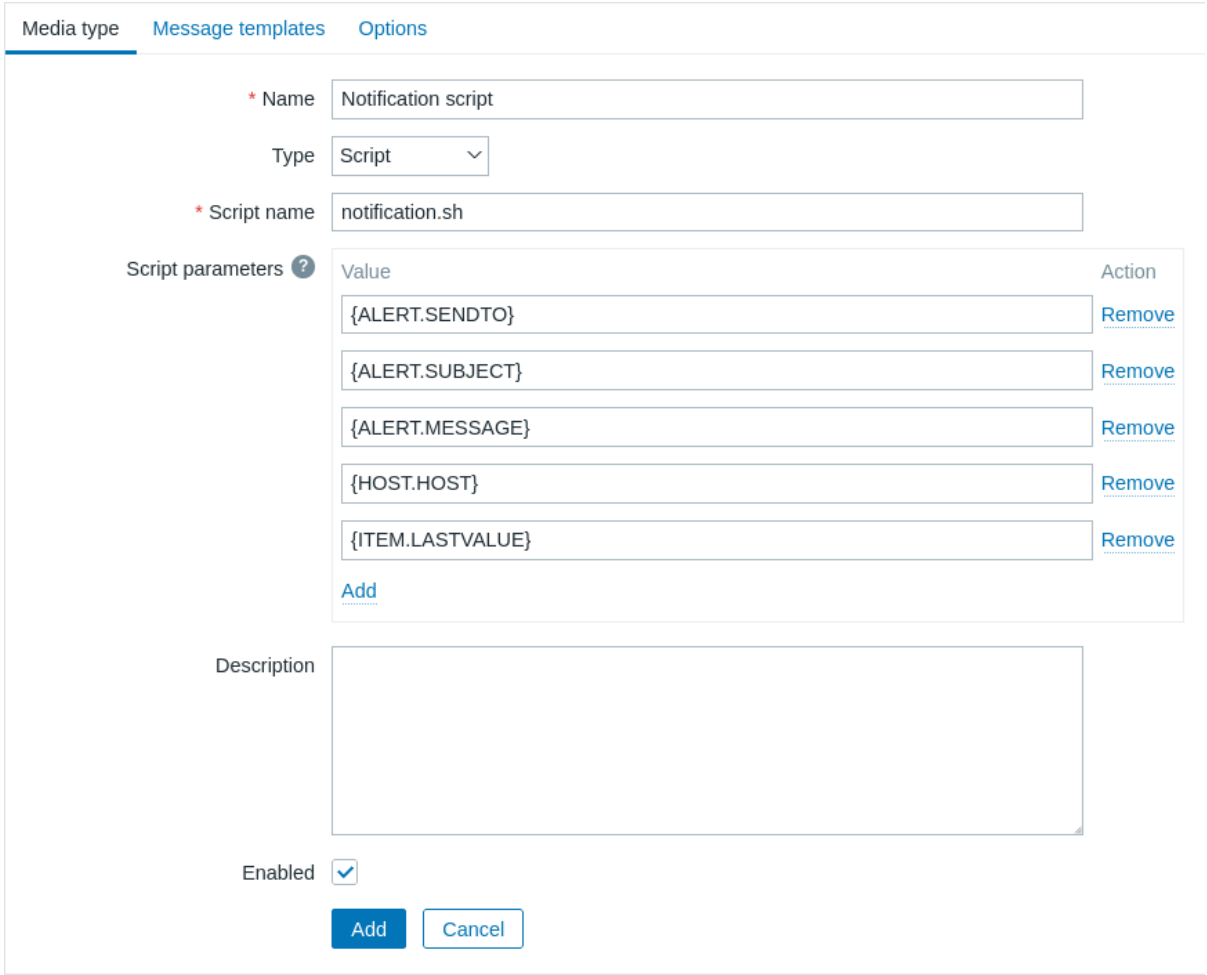
Variáveis de ambiente não são preservadas nem criadas para o script, portanto devem ser tratadas explicitamente.

Configuração

Para configurar scripts de alerta personalizados como um tipo de mídia:

1. Acesse *Alertas* > *Tipos de mídia*.
2. Clique em *Criar tipo de mídia*.
3. Insira os parâmetros do tipo de mídia de scripts no formulário.

A aba **Tipo de mídia** contém os atributos gerais do tipo de mídia:



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Os parâmetros a seguir são específicos do tipo de mídia de script:

Parâmetro	Descrição
Nome do script	Insira o nome do arquivo de script (por exemplo, notification.sh) localizado no diretório especificado no parâmetro de configuração do server AlertScriptsPath.
Parâmetros do script	Adicione parâmetros opcionais do script, que serão passados para o script como argumentos de linha de comando na ordem em que forem definidos. Os parâmetros do script são compatíveis com as macros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e todas as macros compatíveis com notificações, bem como com as macros de usuário.

Consulte parâmetros comuns de tipo de mídia para obter detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

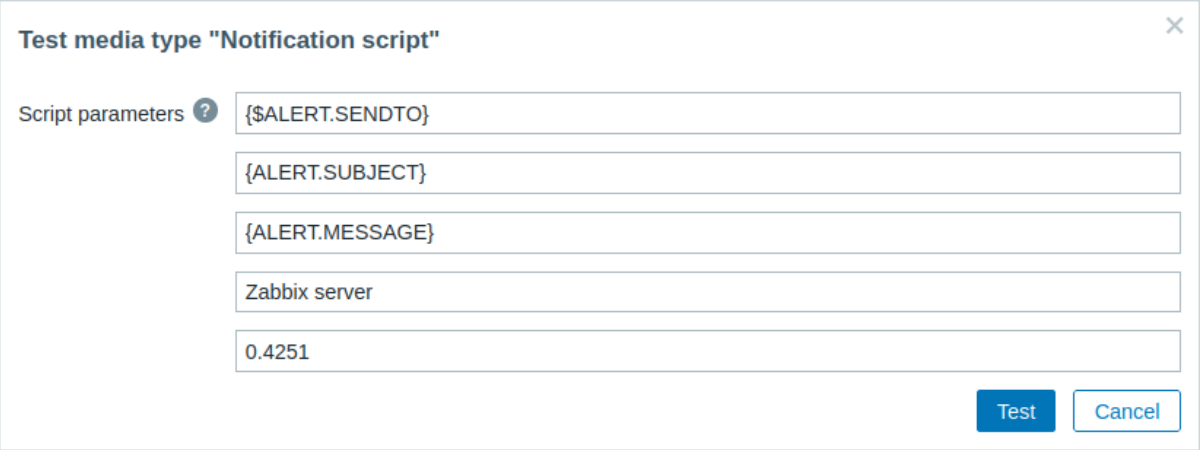
Warning:
Mesmo que um script de alerta não use mensagens padrão, os templates de mensagem para os tipos de operação usados por esse tipo de mídia ainda deverão ser definidos. Caso contrário, uma notificação não será enviada.

Attention:
Se mais de um tipo de mídia de script estiver configurado, esses scripts poderão ser processados em paralelo pelos processos alerter. O número total de processos alerter é limitado pelo parâmetro `StartAlerters` do arquivo de configuração do server.

Teste

Para testar um tipo de mídia de script configurado:

1. Localize o script relevante na lista de tipos de mídia.
2. Clique em *Test* 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 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 *Test*.



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 um evento relacionado do qual obter os detalhes. Observe que macros dentro de 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 estiver configurado, vá para a seção *Users* > *Users* e edite um perfil de usuário atribuindo esse tipo de mídia ao usuário. As etapas para configurar a mídia do usuário, que são comuns a todos os tipos de mídia, estão descritas na página Media types.

Observe que, ao definir a mídia do usuário, o campo *Send to* não pode ficar vazio. Se esse campo não for usado no script de alerta, insira qualquer combinação de caracteres suportados para contornar os requisitos de validação.

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 que tipos de mídia webhook predefinidos sejam usados para enviar notificações do Zabbix para:

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams 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 as notificações do Zabbix em tickets do Spiceworks, crie um tipo de mídia de e-mail e insira o endereço de e-mail do help desk do Spiceworks (por exemplo, help@zabbix.on.spiceworks.com) nas configurações do perfil de um usuário designado do Zabbix.

Configuração

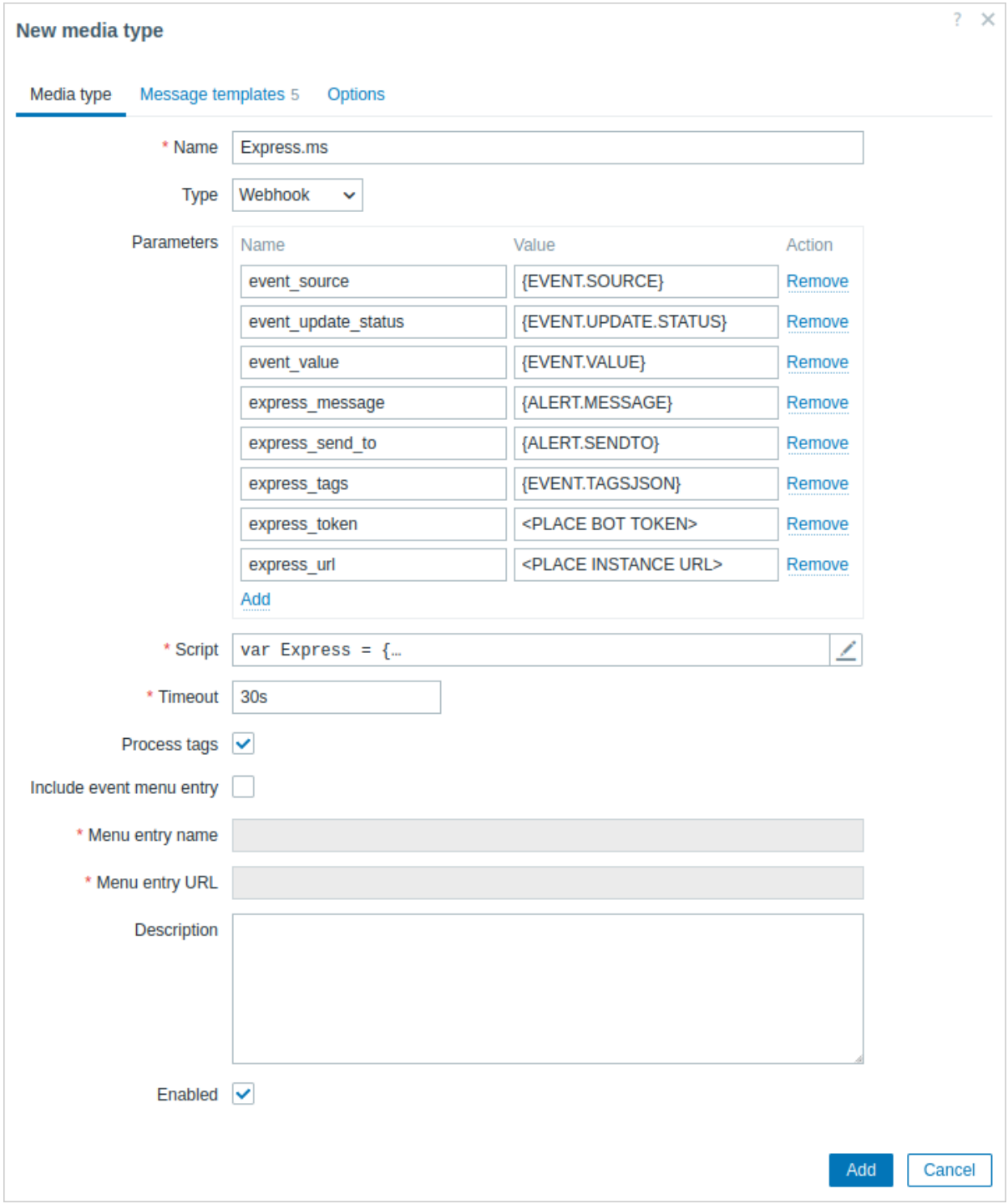
Para começar a usar uma integração de webhook:

1. Localize o arquivo `.yaml` necessário no diretório `templates/media` da versão baixada do Zabbix ou baixe-o do repositório git do Zabbix.
2. Importe o arquivo para a 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 de um webhook acima para acessar rapidamente o arquivo `Readme.md`).

Para criar um webhook personalizado do zero:

1. Acesse *Alertas* > *Tipos de mídia*.
2. Clique em *Criar tipo de mídia*.
3. Insira os parâmetros do tipo de mídia webhook no formulário.

A aba **Tipo de mídia** contém vários atributos específicos para este tipo de mídia:



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Os parâmetros a seguir 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 de acordo com o serviço. Consulte o arquivo *Readme.md* do webhook para obter a descrição dos parâmetros. Para novos webhooks, várias variáveis comuns são incluídas por padrão (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); você pode mantê-las ou removê-las. Os parâmetros do webhook são compatíveis com macros de usuário, todas as macros compatíveis com notificações de problemas e, adicionalmente, com as macros {ALERT.SENDTO}, {ALERT.SUBJECT} e {ALERT.MESSAGE}. Se você especificar um proxy HTTP, o campo será compatível com a mesma funcionalidade do campo proxy HTTP na configuração do item. A string do proxy pode ser prefixada com `[scheme]://` para especificar o tipo de proxy usado (por exemplo, https, socks4, socks5; consulte a documentação).
Script	Insira o código JavaScript no editor modal que é aberto ao clicar no campo do parâmetro ou no ícone de lápis ao lado dele. Esse código executará a operação do webhook. O script é um código de função que aceita pares de parâmetro e 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 solicitações HTTP GET, POST, PUT e DELETE, oferecer suporte a métodos adicionais, como CONNECT, PATCH, HEAD, OPTIONS e TRACE, e controlar os cabeçalhos HTTP e o corpo da solicitação. O script deve conter um operador return; caso contrário, não será válido. Ele pode retornar um status OK junto com uma lista opcional de tags e valores de tags (consulte a opção *Processar tags*) ou uma string de erro. Os eventos de recuperação (gerados automaticamente ou como resultado de um fechamento manual) são criados pelo server e incluem tags de evento resolvidas (incluindo tags herdadas de templates, hosts e triggers). Os scripts de webhook são executados depois que o alerta é criado; portanto, as tags retornadas por um script de webhook são adicionadas somente após a criação inicial do alerta e não estarão presentes nas macros {EVENT.TAGS} e `{EVENT.RECOVERY.TAGS}` da mensagem inicial do problema ou da mensagem de recuperação imediata. *Observação*: recomenda-se usar variáveis locais (por exemplo, `var local = 1`) em vez de variáveis globais (por exemplo, `global = 1`) para garantir que cada script opere sobre seus próprios dados e evitar colisões entre chamadas simultâneas (consulte os problemas conhecidos). Consulte também: Diretrizes de desenvolvimento de webhook, Exemplos de scripts de webhook, Objetos JavaScript adicionais.
Tempo limite	Tempo limite de execução do JavaScript (1-60s, padrão: 30s). Sufixos de tempo são compatíveis (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 de tags vazio: `var result = {tags: {}};` Exemplos de tags que podem ser retornadas: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Incluir entrada no menu de eventos	Marque a caixa de seleção para incluir uma entrada no menu de eventos com um link para um ticket externo criado. Uma entrada será incluída para cada webhook habilitado que tenha esta caixa de seleção marcada. Observe que, se os parâmetros *Nome da entrada do menu* e *URL da entrada do menu* contiverem macros {EVENT.TAGS.<tag name>}, uma entrada será incluída somente se essas macros puderem ser resolvidas (ou seja, se o evento tiver essas tags definidas). Se marcada, o webhook não deve ser usado para enviar notificações a usuários diferentes (considere criar um usuário dedicado) e não deve ser usado em várias ações de alerta para um único evento de problema.

Parâmetro	Descrição
Nome da entrada do menu	Especifique o nome da entrada do menu. A macro {EVENT.TAGS.<tag name>} é compatível. 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 compatível. Este campo só é obrigatório se *Incluir entrada no menu de eventos* estiver marcado.

Consulte parâmetros comuns de tipo de mídia para obter detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

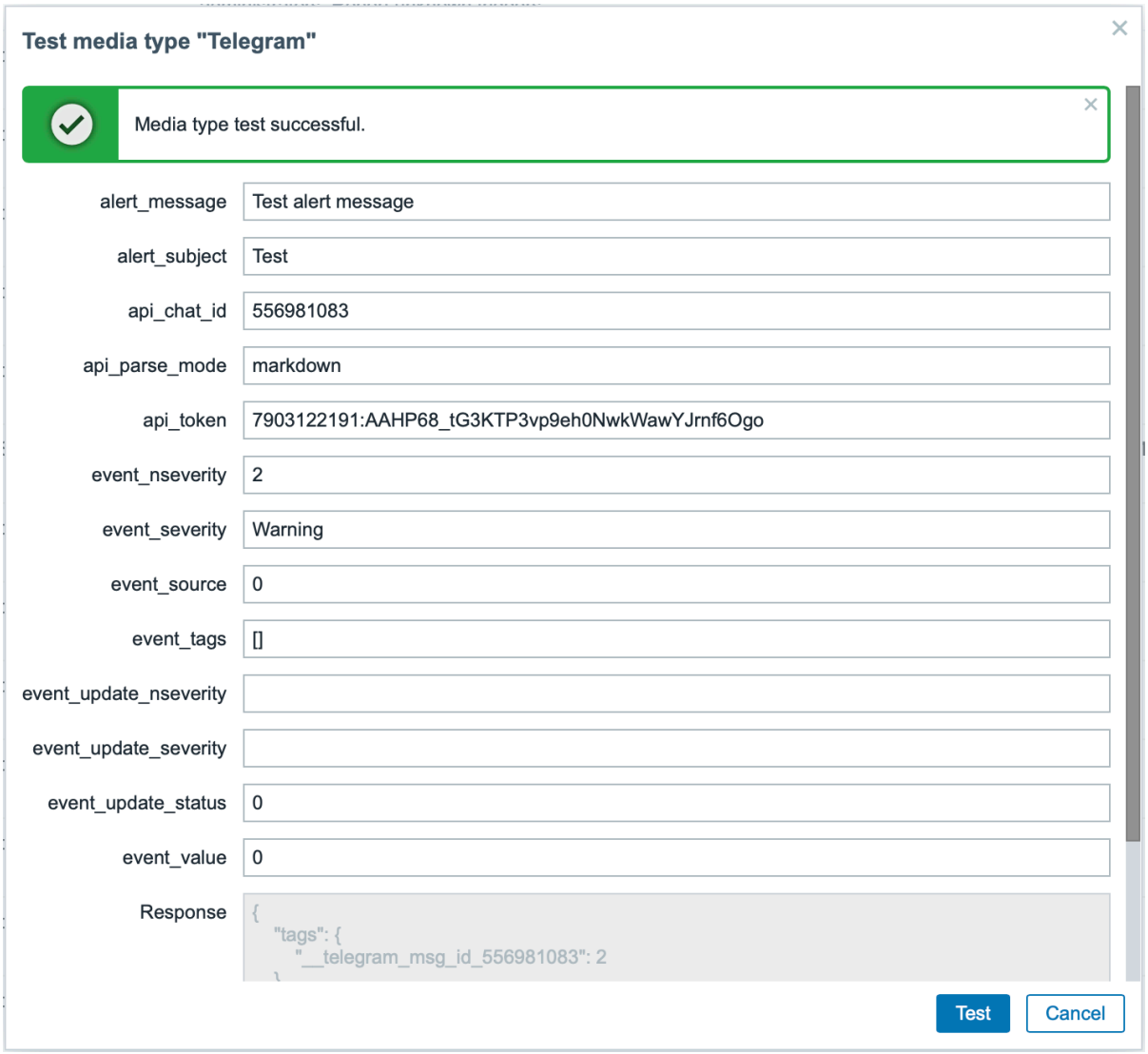
Warning:
Mesmo que um webhook não use mensagens padrão, os templates de mensagem para os tipos de operação usados por esse webhook ainda deverão ser definidos.

Teste

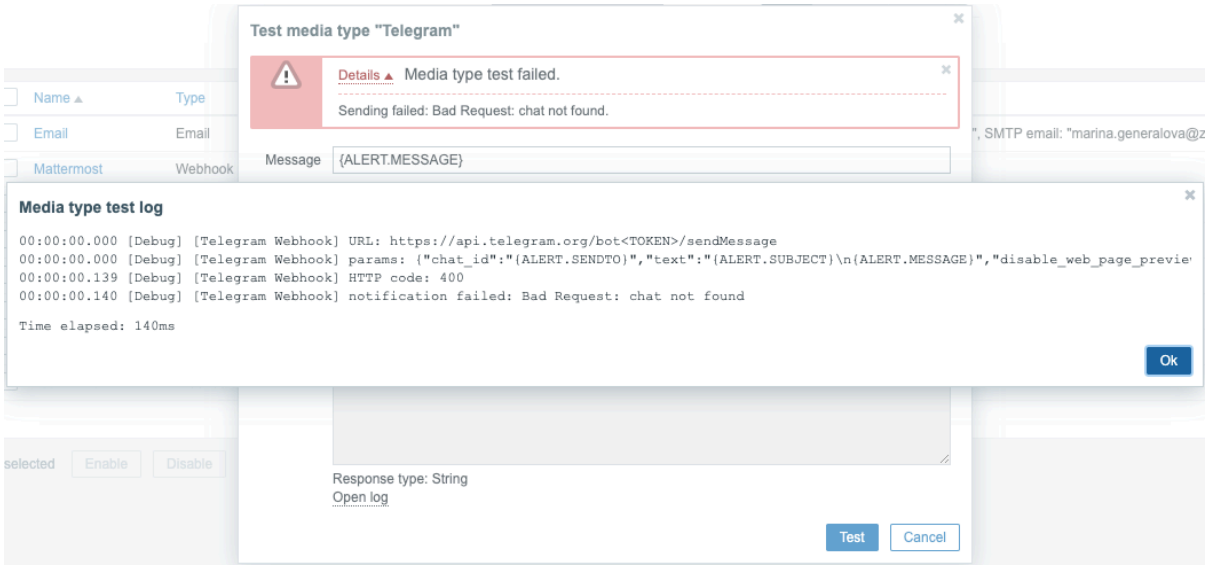
Para testar um tipo de mídia webhook configurado:

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 do log de 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 de configurar o tipo de mídia, acesse 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 existente, que são comuns a 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 a usuários diferentes, pois isso pode causar erros no webhook (aplica-se à maioria dos webhooks que utilizam a opção *Incluir entrada no menu de eventos*). Nesse caso, a prática recomendada é criar um usuário dedicado para representar o webhook:

1. Depois de configurar o tipo de mídia webhook, acesse 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 permanecer com os valores padrão, pois esse usuário não fará login no Zabbix.
2. No perfil do usuário, acesse a aba *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 compatíveis para atender aos requisitos de validação.
3. Conceda a esse usuário pelo menos permissões de leitura para todos os hosts para os quais ele deverá enviar os alertas.

Ao configurar a ação de alerta, adicione esse usuário ao campo *Enviar para usuários* em Detalhes da operação - isso informará ao Zabbix que deve usar o webhook para as notificações dessa ação.

Configurando ações de alerta

As ações determinam quais notificações devem ser enviadas via webhook. As etapas para configurar ações envolvendo webhooks são as mesmas que para todos os outros tipos de mídia, com estas exceções:

- Se um webhook usar webhook tags para armazenar o ID da ticket\mensagem e tratar 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 atualizado 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 de 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 *Include event menu entry*. Observe, no entanto, que um único webhook pode ser usado em várias operações ou etapas de escalonamento da mesma ação, bem como em ações diferentes 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 *Custom message* e definir uma mensagem personalizada na configuração da operação da ação. Caso contrário, uma notificação não será enviada.

Exemplos de script de webhook

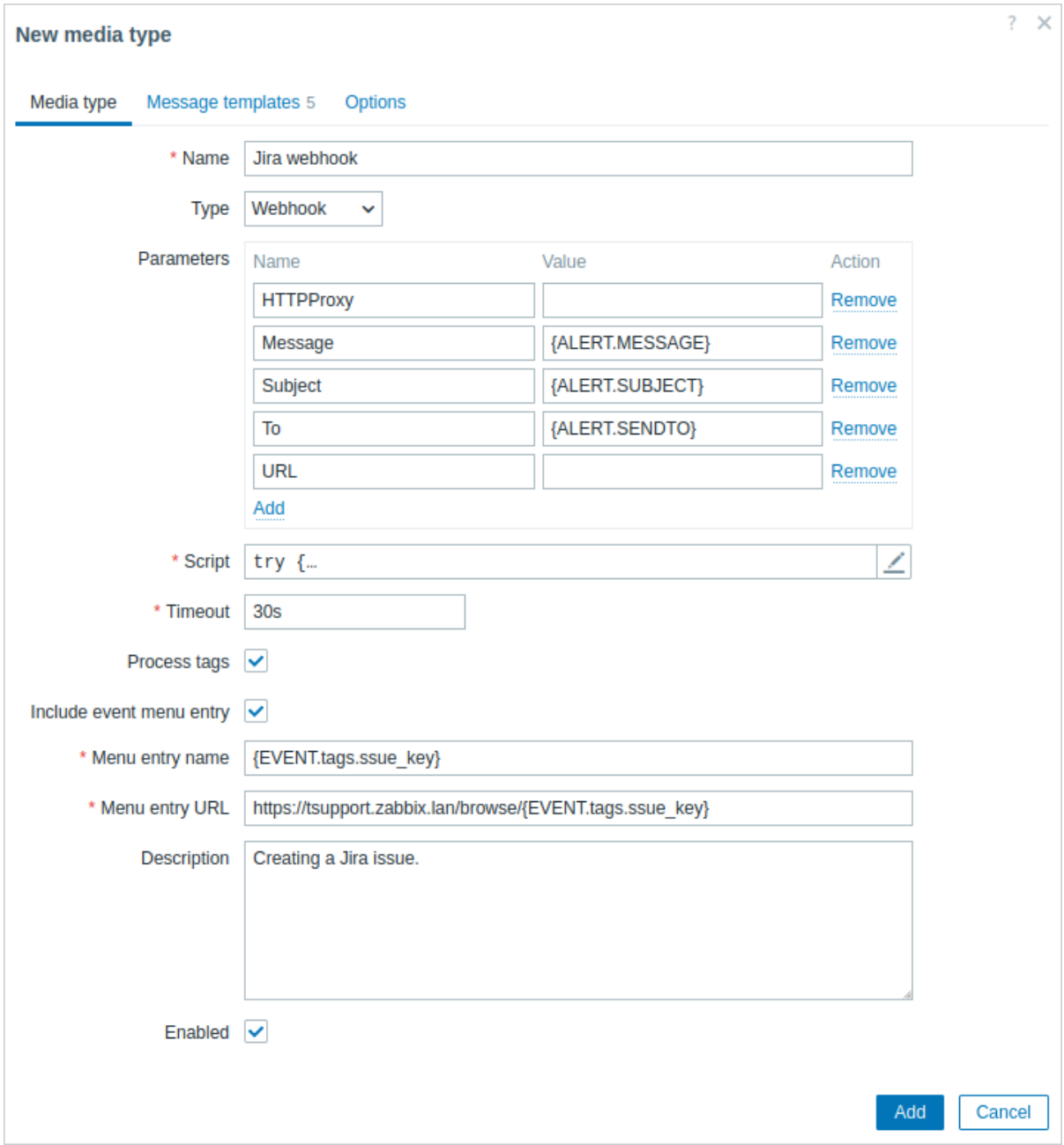
Visão geral

Embora o Zabbix ofereça um grande número de integrações de webhook disponíveis prontas para uso, talvez você queira criar seus próprios webhooks. Esta seção fornece exemplos de scripts de webhook personalizados (usados no parâmetro *Script*). Consulte a seção webhook para a descrição de outros parâmetros de webhook.

Attention:
Não use atribuições não declaradas em JavaScript de pré-processamento. Use `var` para declarar variáveis locais.

Eventos de recuperação (sejam gerados automaticamente ou após um fechamento manual) incluem tags de evento resolvidas (incluindo tags herdadas de templates, hosts e triggers). Os scripts de webhook são executados após a criação do alerta; portanto, as tags retornadas por um script de webhook são aplicadas somente depois da criação inicial do alerta e podem não estar presentes no alerta que acionou o webhook. Se uma integração exigir que as tags produzidas por um webhook estejam presentes na notificação inicial, busque as tags do evento no server (por exemplo, usando a API Event) ou armazene as tags produzidas pelo webhook em um armazenamento persistente externo e faça a correlação lá.

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();
    }

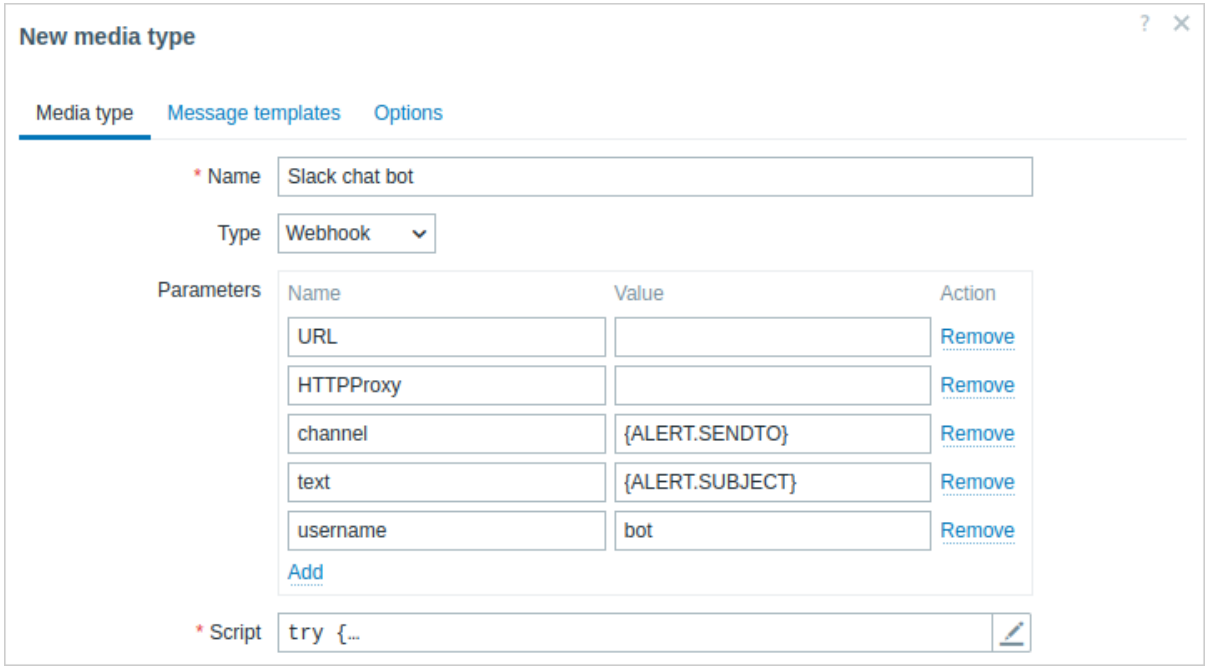
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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;
}
```

webhook do Slack (customizado)

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.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;
}
```

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 estado do trigger muda entre `OK` e `Problema`
- Ações de serviço - para eventos quando o estado do serviço muda entre `OK` e `Problema`
- Ações de descoberta - para eventos quando ocorre a descoberta de rede
- Ações de autorregistro - para eventos quando novos agents ativos se autorregistram (ou quando a metadata do host muda para os já registrados)
- Ações internas - para eventos quando os items se tornam não suportados ou triggers entram em estado desconhecido

As principais diferenças das ações de serviço são:

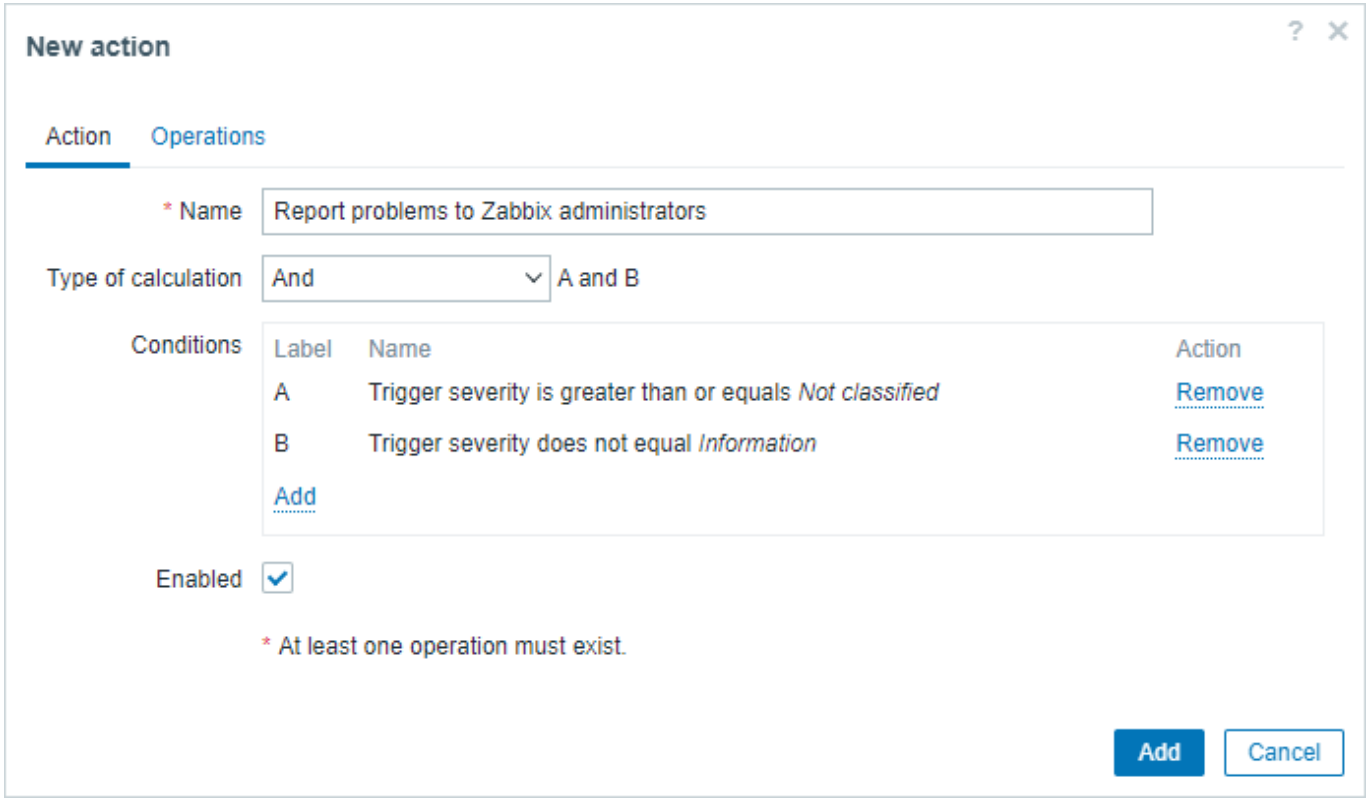
- O acesso do usuário às ações de serviço depende dos direitos de acesso aos serviços concedidos pela função do usuário
- As ações de serviço suportam um conjunto diferente de condições

Configurando uma action

Para configurar uma action, faça o seguinte:

1. Acesse *Alerts* > *Actions* e selecione o tipo de action desejado no submenu (você pode alternar para outro tipo mais tarde, usando o menu suspenso do título).
2. Clique em *Create action*.
3. Dê um nome à action.
4. Escolha as conditions com base nas quais as operações serão executadas.
5. Escolha as operations a serem executadas.

Atributos gerais da action:



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Parameter	Description
Name	Nome exclusivo da action.

Parameter	Description
Type of calculation	Selecione a option de avaliação para as conditions da action (com mais de uma condition): **And** - todas as conditions devem ser atendidas. **Or** - basta que uma condition seja atendida. **And/Or** - combinação dos dois: AND com diferentes tipos de condition e OR com o mesmo tipo de condition. **Custom expression** - uma fórmula de cálculo definida pelo usuário para avaliar as conditions da action.
Conditions	Lista de conditions da action. Clique em *Add* para adicionar uma nova condition. Se nenhuma condition estiver configurada, a action será executada para todo evento que corresponda ao action type configurado.
Enabled	Marque a caixa de seleção para habilitar a action. Caso contrário, ela ficará desabilitada.

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 action.

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.

Tipo de condição	Operadores suportados	Descrição
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 service

As seguintes condições podem ser usadas em ações de service:

Condition type	Supported operators	Description
Service	equals does not equal	Especifique um service ou um service a ser excluído. **equals** - o evento pertence a este service. **does not equal** - o evento não pertence a este service. Especificar um service pai seleciona implicitamente todos os services filhos. Para especificar apenas o service pai, todos os services aninhados também devem ser definidos com o operador **does not equal**.
Service name	contains does not contain	Especifique uma string no nome do service ou uma string a ser excluída. **contains** - o evento é gerado por um service que contém esta string no nome. **does not contain** - esta string não pode ser encontrada no nome do service.
Service tag name	equals does not equal contains does not contain	Especifique uma tag de evento ou uma tag de evento a ser excluída. As tags de evento de service podem ser definidas na seção de configuração do service *Tags*. **equals** - o evento tem esta tag. **does not equal** - o evento não tem esta tag. **contains** - o evento tem uma tag contendo esta string. **does not contain** - o evento não tem uma tag contendo esta string.
Service tag value	equals does not equal contains does not contain	Especifique uma combinação de tag e valor de evento ou uma combinação de tag e valor a ser excluída. As tags de evento de service podem ser definidas na seção de configuração do service *Tags*. **equals** - o evento tem esta tag e este valor. **does not equal** - o evento não tem esta tag e este valor. **contains** - o evento tem uma tag e um valor contendo estas strings. **does not contain** - o evento não tem uma tag e um valor contendo estas strings.

Attention:
Certifique-se de definir message templates para Service actions no menu *Alerts* > *Media types*. 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 compatíveis	Descrição
IP do host	é igual a não é igual a	Especifique um intervalo de endereços IP ou um intervalo a ser excluído para um host descoberto. **é igual a** - o IP do host está no intervalo. **não é igual a** - 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 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 compatíveis.
Tipo de serviço	é igual a não é igual a	Especifique um tipo de serviço de um serviço descoberto ou um tipo de serviço a ser excluído. **é igual a** - corresponde ao serviço descoberto. **não é igual a** - não corresponde ao serviço descoberto. Tipos de serviço disponíveis: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix agent, agent SNMPv1, agent SNMPv2, agent SNMPv3, ping ICMP, telnet.
Porta do serviço	é igual a não é igual a	Especifique um intervalo de portas TCP de um serviço descoberto ou um intervalo a ser excluído. **é igual a** - a porta do serviço está no intervalo. **não é igual a** - a porta do serviço não está no intervalo.
Regra de descoberta	é igual a não é igual a	Especifique uma regra de descoberta ou uma regra de descoberta a ser excluída. **é igual a** - usando esta regra de descoberta. **não é igual a** - usando qualquer outra regra de descoberta, exceto esta.
Verificação de descoberta	é igual a não é igual a	Especifique uma verificação de descoberta ou uma verificação de descoberta a ser excluída. **é igual a** - usando esta verificação de descoberta. **não é igual a** - usando qualquer outra verificação de descoberta, exceto esta.
Objeto de descoberta	é igual a	Especifique o objeto descoberto. **é igual a** - igual ao objeto descoberto (um dispositivo ou um serviço).
Status da descoberta	é igual a	**Up** - corresponde a eventos 'Host Up' e 'Service Up'. **Down** - corresponde a eventos 'Host Down' e 'Service Down'. **Discovered** - corresponde a eventos 'Host Discovered' e 'Service Discovered'. **Lost** - corresponde a eventos 'Host Lost' e 'Service Lost'.
Tempo de atividade/inatividade	é maior ou igual a é menor ou igual a	Tempo de atividade para eventos 'Host Up' e 'Service Up'. Tempo de inatividade para eventos 'Host Down' e 'Service Down'. **é maior ou igual a** - é maior ou igual a. O parâmetro é fornecido em segundos. **é menor ou igual a** - é menor ou igual a. O parâmetro é fornecido em segundos.

Tipo de condição	Operadores compatíveis	Descrição
Valor recebido	é igual a não é igual a é maior ou igual a é menor ou igual a contém não contém	Especifique o valor recebido de uma verificação de agent (Zabbix, SNMP) em uma regra de descoberta. Comparação de strings. Se várias verificações de agent Zabbix ou SNMP estiverem configuradas para uma regra, os valores recebidos de cada uma delas serão verificados (cada verificação gera um novo evento que é comparado com todas as condições). **é igual a** - igual ao valor. **não é igual a** - diferente do valor. **é maior ou igual a** - maior ou igual ao valor. **é menor ou igual a** - menor ou igual ao valor. **contém** - contém a substring. O parâmetro é fornecido como uma string. **não contém** - não contém a substring. O parâmetro é fornecido como uma string.
Proxy	é igual a não é igual a	Especifique um proxy ou um proxy a ser excluído. **é igual a** - usando este proxy. **não é igual a** - usando qualquer outro proxy, exceto este.
Grupo de proxy	é igual a não é igual a	Especifique um grupo de proxy ou um grupo de proxy a ser excluído. **é igual a** - usando este grupo de proxy. **não é igual a** - usando qualquer outro grupo de proxy, exceto este.

Note:
As verificações de serviço em uma regra de descoberta, que resultam em eventos de descoberta, não ocorrem simultaneamente. Portanto, se **vários** valores forem configurados para as condições `Tipo de serviço`, `Porta do serviço` ou `Valor recebido` na ação, eles serão comparados com um evento de descoberta por vez, mas **não** com vários eventos simultaneamente. Como resultado, as ações com vários valores para os mesmos tipos de verificação podem não ser executadas corretamente.

Ações de registro automático

As seguintes condições podem ser usadas em ações baseadas no registro automático de agent ativo:

Tipo de condição	Operadores compatíveis	Descrição
Metadados do host	contém não contém corresponde não corresponde	Especifique os metadados do host ou os metadados do host a serem excluídos. **contém** - os metadados do host contêm a string. **não contém** - os metadados do host não contêm a string. Os metadados do host podem ser especificados em um arquivo de configuração do agent. **corresponde** - os metadados do host correspondem à expressão regular. **não corresponde** - os metadados do host não correspondem à expressão regular.
Nome do host	contém não contém corresponde não corresponde	Especifique um nome de host ou um nome de host a ser excluído. **contém** - o nome do host contém a string. **não contém** - o nome do host não contém a string. **corresponde** - o nome do host corresponde à expressão regular. **não corresponde** - o nome do host não corresponde à expressão regular.
Proxy	é igual a não é igual a	Especifique um proxy ou um proxy a ser excluído. **é igual a** - usando este proxy. **não é igual a** - usando qualquer outro proxy (ou server), exceto este proxy.
Grupo de proxy	é igual a não é igual a	Especifique um grupo de proxy ou um grupo de proxy a ser excluído. **é igual a** - usando este grupo de proxy. **não é igual a** - usando qualquer outro grupo de 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.

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 nenhuma permissão definida para o host.

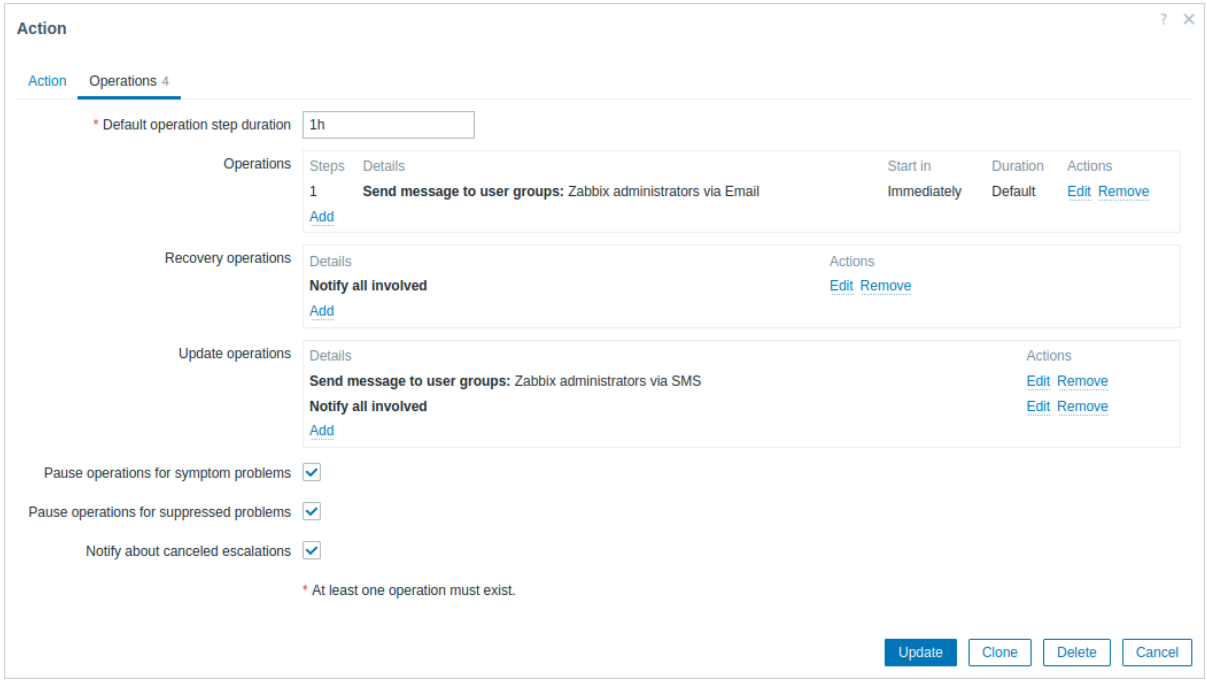
Para eventos de descoberta e autorregistro, operações adicionais estão disponíveis:

- Adicionar host
- Remover host
- Habilitar host
- Desabilitar host
- Adicionar ao grupo de hosts
- Remover do grupo de hosts
- Adicionar tags de host
- Remover tags de host
- Vincular template
- Desvincular template
- Definir modo de inventário do host

Configurando uma operação

Para configurar uma operação:

1. Acesse a aba *Operations* na configuração da action.
2. Configure os detalhes da operação.



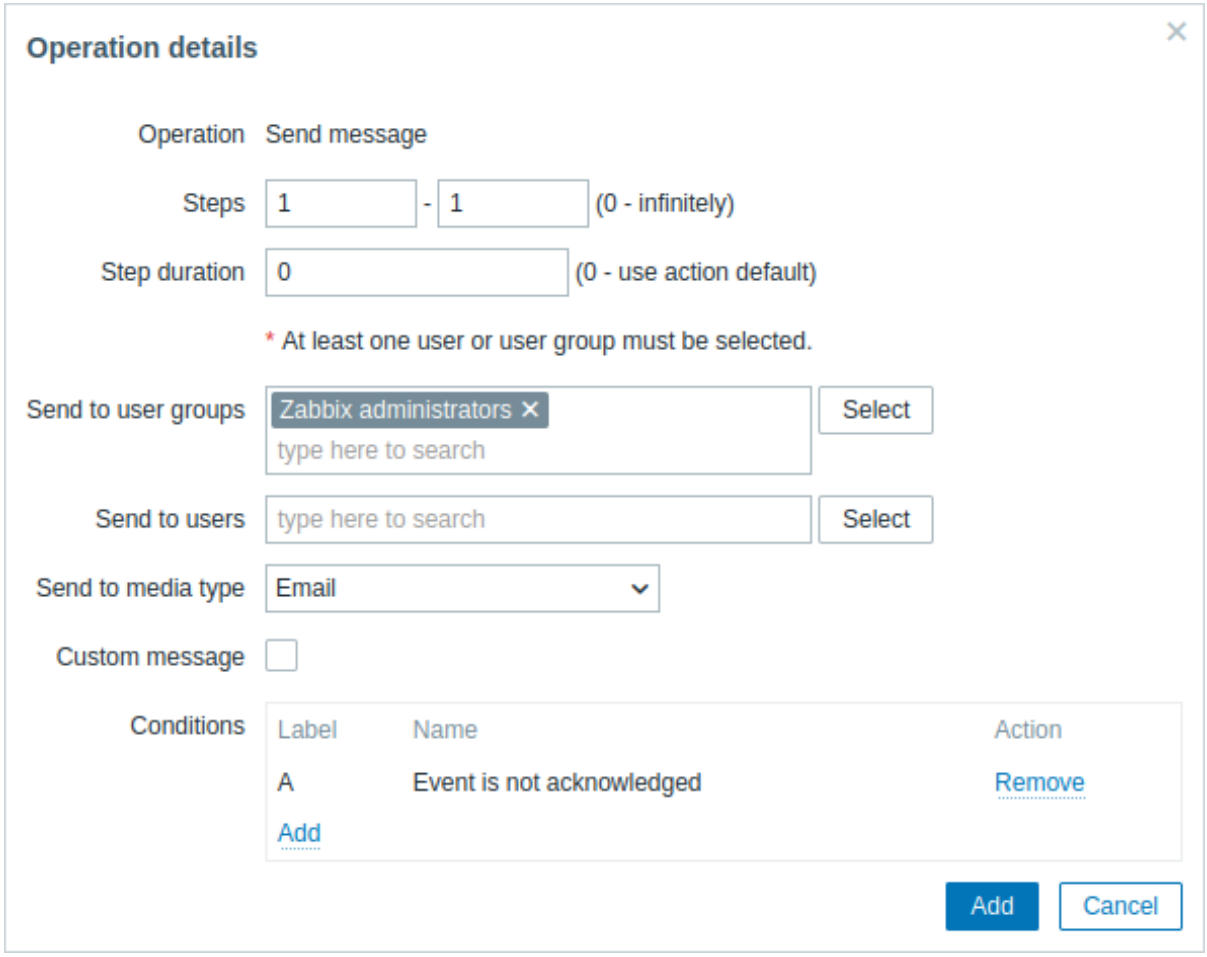
Atributos gerais da operação:

Parameter	Description
Default operation step duration	Duração padrão de uma etapa da operação (de 60 segundos a 1 semana). Por exemplo, uma duração de etapa de uma hora significa que, se uma operação for executada, uma hora passará antes da próxima etapa. Sufixos de tempo são compatíveis, por exemplo, 60s, 1m, 2h, 1d. Macros de usuário são compatíveis.
Operations	As operações da action (se houver) são exibidas com os seguintes detalhes: **Steps** - etapa(s) de escalonamento às quais a operação está atribuída. **Details** - 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 o sobrenome (entre parênteses após o nome de usuário) do destinatário da notificação. **Start in** - quanto tempo após um evento a operação será executada. **Duration (sec)** - a duração da etapa é exibida. *Default* é exibido se a etapa usar a duração padrão, e um período de tempo é exibido se for usada uma duração personalizada. **Actions** - são exibidos links para editar e remover uma operação.
Recovery operations	As operações da action (se houver) são exibidas com os seguintes detalhes: **Details** - 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 o sobrenome (entre parênteses após o nome de usuário) do destinatário da notificação. **Actions** - são exibidos links para editar e remover uma operação.
Update operations	As operações da action (se houver) são exibidas com os seguintes detalhes: **Details** - 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 o sobrenome (entre parênteses após o nome de usuário) do destinatário da notificação. **Actions** - são exibidos links para editar e remover uma operação.
Pause operations for symptom problems	Marque esta caixa de seleção para pausar as operações (após a primeira operação) para problemas sintomáticos. Observe que esta configuração afeta apenas os escalonamentos de problemas; as operações de recuperação e atualização não serão afetadas. Esta opção está disponível somente para *Trigger actions*.
Pause operations for suppressed problems	Marque esta caixa de seleção para atrasar o início das operações durante um período de manutenção. Quando as operações forem iniciadas, após a manutenção, todas as operações serão executadas, incluindo aquelas referentes aos eventos ocorridos durante a manutenção. Observe que esta configuração afeta apenas os escalonamentos de problemas; as operações de recuperação e atualização não serão afetadas. Se você desmarcar esta caixa de seleção, as operações serão executadas sem atraso, mesmo durante um período de manutenção. Esta opção não está disponível para *Service actions*.
Notify about canceled escalations	Desmarque esta caixa de seleção para desabilitar as notificações sobre escalonamentos cancelados (quando host, item, trigger ou action está desabilitado).

Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Para configurar os detalhes de uma nova operação, clique em Add no bloco *Operations*. Para editar uma operação existente, clique em Edit ao lado da operação. Uma janela pop-up será aberta, na qual você poderá editar os detalhes da etapa da operação.

Detalhes da operação



Parâmetro	Descrição
Operação	Selecione a operação: **Enviar mensagem** - envia uma mensagem ao usuário. **<nome do comando remoto>** - executa um comando remoto. Os comandos estão disponíveis para execução se tiverem sido previamente definidos em scripts globais com *Operação de ação* selecionada como escopo. Há mais operações disponíveis para eventos baseados em descoberta e autoregistro (veja acima).
Etapas	Selecione a(s) etapa(s) às quais atribuir a operação em um cronograma de escalonamento: **De** - executa começando por esta etapa. **Até** - executa até esta etapa (0=infinito, a execução não será limitada).
Duração da etapa	Duração personalizada para estas etapas (0=usar a duração padrão da etapa). Sufixos de tempo são compatíveis, por exemplo, 60s, 1m, 2h, 1d. Macros de usuário são compatíveis. Várias operações podem ser atribuídas à mesma etapa. Se essas operações tiverem durações de etapa diferentes definidas, a menor duração será considerada e aplicada à etapa.
Tipo de operação: enviar mensagem	

Parâmetro	Descrição
Enviar para grupos de usuários	Selecione os grupos de usuários para os quais enviar a mensagem. O grupo de usuários deve ter pelo menos permissões de ”leitura” no host para ser notificado.
Enviar para usuários	Selecione os usuários para os quais enviar a mensagem. O usuário deve ter pelo menos permissões de ”leitura” no host para ser notificado.
Enviar para tipo de mídia	Envie a mensagem para todos os tipos de mídia disponíveis (configurados e habilitados) ou somente para um tipo específico.
Mensagem personalizada	Se selecionada, a mensagem personalizada poderá ser configurada. Para notificações sobre eventos internos via webhooks, a mensagem personalizada é obrigatória.
Assunto	Assunto da mensagem personalizada. O assunto pode conter macros. Ele é limitado a 255 caracteres.
Mensagem	A mensagem personalizada. A mensagem pode conter macros. Ela é limitada a uma determinada quantidade de caracteres, dependendo do tipo de banco de dados (consulte Enviar mensagem para obter mais informações).
Tipo de operação: comando remoto	
Lista de destinos	Selecione os destinos nos quais executar o comando: **Host atual** - o comando é executado no host do trigger que causou o evento de problema. Essa opção não funcionará se houver vários hosts no trigger. **Host** - selecione o(s) host(s) nos quais executar o comando. **Grupo de hosts** - selecione o(s) grupo(s) de hosts nos quais executar o comando. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Assim, o comando remoto também será executado nos hosts dos grupos aninhados. Um comando em um host é executado apenas uma vez, mesmo que o host corresponda mais de uma vez (por exemplo, em vários grupos de hosts; individualmente e a partir de um grupo de hosts). A lista de destinos não tem efeito se um script personalizado for executado no Zabbix server. Selecionar mais destinos nesse caso fará apenas com que o script seja executado mais vezes no server. Observe que, para scripts globais, a seleção de destino também depende da configuração *Grupo de hosts* na configuração do script global. A opção *Lista de destinos* não está disponível para *ações de serviço*, pois, nesse caso, os comandos remotos são sempre executados no Zabbix server.
Condições	Condição para executar a operação: **O evento não foi reconhecido** - somente quando o evento não foi reconhecido. **O evento foi reconhecido** - somente quando o evento foi reconhecido. A opção *Condições* está disponível somente para *ações de trigger*.

Quando terminar, clique em *Adicionar* para adicionar a operação à lista de *Operações*.

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 a mensagem será enviada.

Se a operação ocorrer fora do período Quando ativo definido para a mídia selecionada na configuração do usuário, a mensagem não será enviada.

A severidade padrão do trigger (’Não classificado’) **deve estar** selecionada na configuração de mídia do usuário se você quiser receber notificações para eventos que não sejam de trigger, como descoberta, registro automático 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 somente para usuários que tenham pelo menos permissões de ’leitura’ no host que gerou o evento. Pelo menos um host da expressão do trigger deve estar acessível.

Você pode configurar cenários personalizados para o envio de mensagens usando escalonamentos.

Para receber e ler emails do Zabbix corretamente, os servidores/clientes de email devem oferecer suporte ao formato padrão de ’email SMTP/MIME’, pois o Zabbix envia dados em UTF-8 (se o assunto contiver somente caracteres ASCII, ele não será codificado em UTF-8). O assunto e o corpo da mensagem são codificados em base64 para seguir o padrão de formato de ’email SMTP/MIME’.

O limite de mensagens após a expansão de todas as macros é o mesmo que o limite de mensagens para comandos remotos.

Rastreamento de mensagens

Você pode visualizar o status das mensagens enviadas em *Monitoring* > *Problems*.

Na coluna *Actions*, você pode ver informações resumidas sobre as ações executadas. Nela, os números verdes representam mensagens enviadas, e os vermelhos - mensagens com falha. *In progress* indica que uma ação foi iniciada. *Failed* informa que nenhuma ação foi executada com sucesso.

Se você clicar no horário do evento para ver os detalhes do evento, poderá ver os detalhes das mensagens enviadas (ou não enviadas) devido ao evento no bloco *Actions*.

Em *Reports* > *Action log*, você verá os detalhes de todas as ações executadas para aqueles eventos que têm uma ação configurada.

Comandos remotos

Visão geral

Com os comandos remotos, você pode definir que um determinado comando predefinido seja executado automaticamente no host monitorado quando uma condição for atendida.

Assim, os comandos remotos são um mecanismo avançado para o monitoramento proativo inteligente.

Nos usos mais óbvios desse recurso, você pode:

- Reiniciar automaticamente algum aplicativo (web server, middleware, CRM) se ele não responder
- Usar o comando `reboot` do IPMI para reiniciar algum server remoto se ele não responder às solicitações
- Liberar espaço em disco automaticamente (removendo arquivos antigos, limpando `/tmp`) quando o espaço em disco estiver acabando
- Migrar uma VM de um equipamento físico para outro dependendo da carga da CPU
- Adicionar novos nós a um ambiente de nuvem quando os recursos de CPU (disco, memória ou qualquer outro) forem insuficientes

Configurar uma ação para comandos remotos é semelhante a configurar uma ação para enviar uma mensagem; a única diferença é que o Zabbix executa um comando em vez de enviar uma mensagem.

Os comandos remotos podem ser executados pelo Zabbix server, pelo Zabbix proxy ou pelo Zabbix agent. Observe o seguinte:

- Os comandos remotos no Zabbix agent podem ser executados diretamente pelo Zabbix server ou por meio do Zabbix proxy.

- Os comandos remotos executados pelo Zabbix server são executados conforme descrito em Execução de comandos, incluindo a verificação do código de saída.
- Os comandos remotos são executados mesmo quando o host de destino está em modo de manutenção.

Os comandos remotos são desabilitados por padrão tanto no Zabbix agent quanto no Zabbix proxy. Eles podem ser habilitados da seguinte forma:

- Para comandos remotos executados pelo Zabbix agent, adicione um parâmetro `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` para cada comando permitido na configuração do agent, em que `*` corresponde a qualquer um dos modos de execução (`wait` ou `nowait`).
- Para comandos remotos executados pelo Zabbix proxy, defina o parâmetro `EnableRemoteCommands` como 1 na configuração do proxy. Observe que não é necessário habilitar os comandos remotos no Zabbix proxy quando eles são executados pelo Zabbix agent em um host monitorado pelo proxy.

Attention:
Não use `AllowKey=system.run[*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` a menos que seja necessário, pois isso permite a execução de qualquer comando no host. Por motivos de segurança, é recomendável permitir apenas comandos exatos, enquanto `system.run[*]` é negado por padrão (`DenyKey=system.run[*]`).

Consulte também Restringindo verificações do agent.

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 exigem 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 do comando remoto IPMI é baseado na biblioteca IPMI instalada. Observe que os limites do banco de dados se aplicam a todos os comandos remotos.

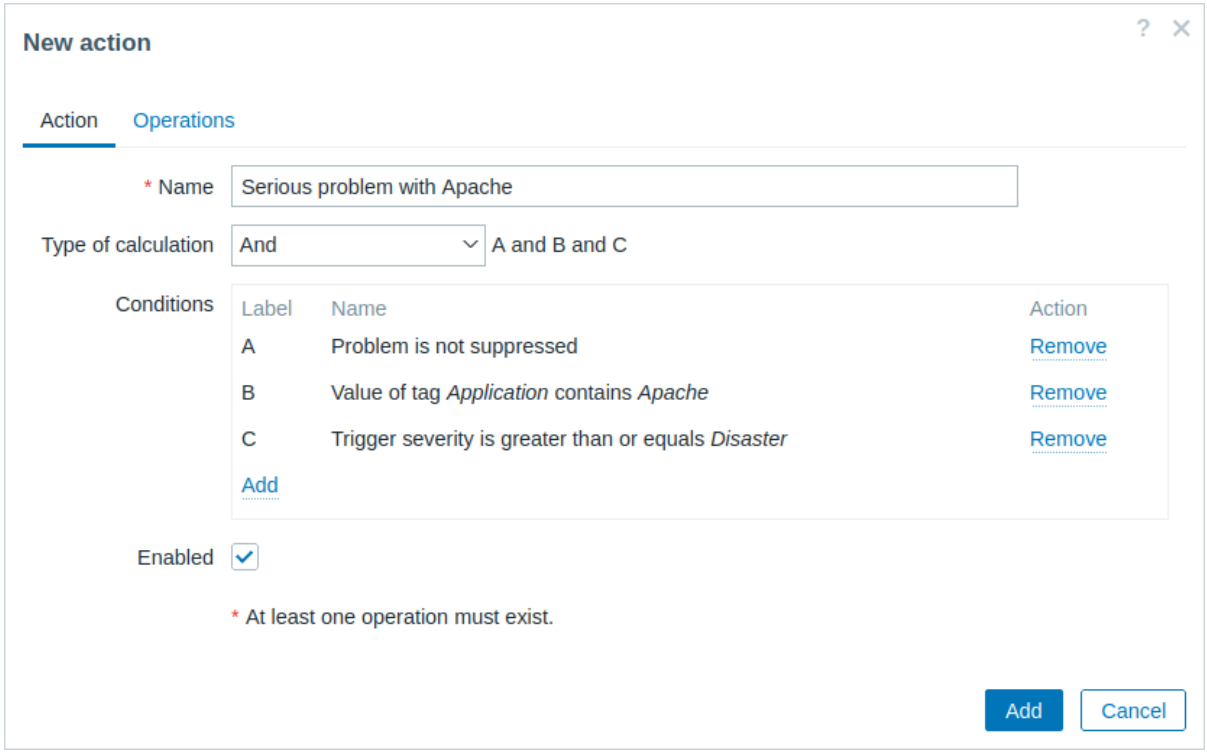
Configuração

Os comandos remotos executados pelo Zabbix agent devem primeiro ser permitidos na configuração do agent usando o parâmetro `AllowKey` ou `AllowKeyRegexp`.

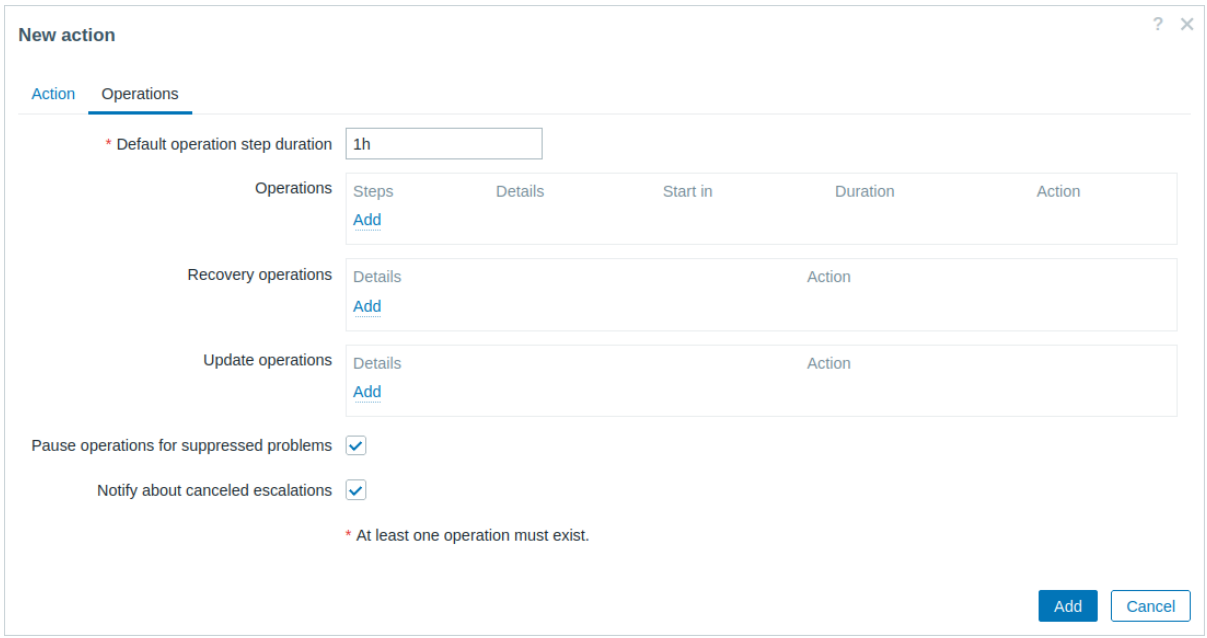
Certifique-se de que o parâmetro `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` seja adicionado à configuração do agent para cada comando permitido, a fim de permitir sua execução no modo `nowait`. Reinicie o daemon do agent após alterar o parâmetro.

Em seguida, ao configurar uma nova ação em *Alertas* > *Ações* > *Ações de trigger*:

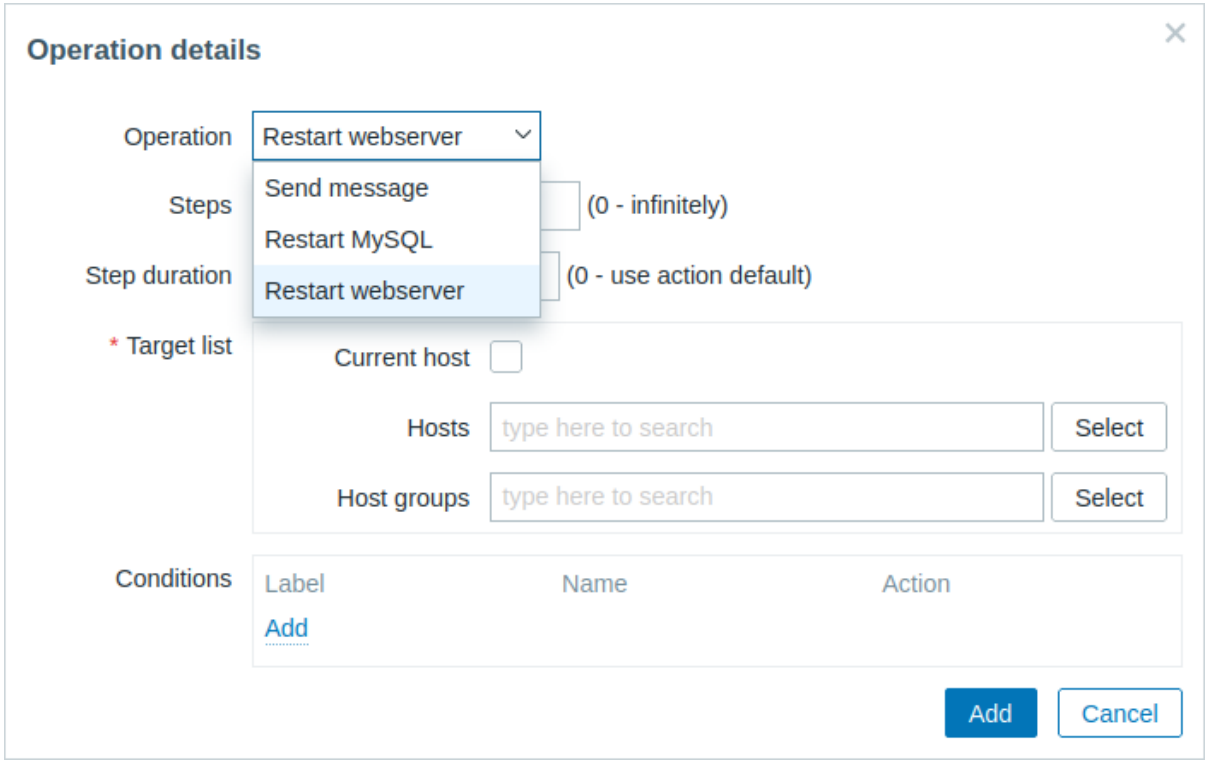
1. Defina as condições apropriadas; por exemplo, configure a ação para ser ativada quando ocorrer qualquer problema de desastre em uma das aplicações Apache.



2. Na aba *Operações*, clique em *Adicionar* na seção *Operações*, *Operações de recuperação* ou *Operações de atualização*.



3. Selecione um dos scripts predefinidos na lista suspensa *Operação* e defina a *Lista de destinos* para o script.



Scripts predefinidos

Scripts que estão disponíveis para operações de ação (webhook, script, SSH, Telnet, IPMI) são definidos em global scripts com *Action operation* selecionado como seu escopo.

Por exemplo:

```
sudo /etc/init.d/apache restart
```

Nesse caso, o Zabbix tentará reiniciar um processo do Apache. Com este comando, certifique-se de que o comando seja executado no Zabbix agent (clique no botão *Zabbix agent* em *Execute on*).

Attention:
Observe o uso de sudo - o usuário do Zabbix não tem permissões para reiniciar serviços do sistema por padrão. Veja abaixo dicas sobre como configurar sudo.

Note:
A partir do Zabbix agent 7.0, comandos remotos também podem ser executados em um agent que esteja operando em modo ativo. O Zabbix agent - seja ativo ou passivo - deve ser executado no host remoto e executa os comandos em segundo plano.

Comandos remotos no Zabbix agent são executados sem timeout pela chave `system.run[,nowait]` e não são verificados quanto aos resultados de execução. No Zabbix server e no Zabbix proxy, os comandos remotos são executados com timeout conforme definido no parâmetro `TrapperTimeout` do arquivo `zabbix_server.conf` ou `zabbix_proxy.conf` e são verificados quanto aos resultados de execução. Para informações adicionais, consulte *Script timeout*.

Permissões de acesso

Certifique-se de que o usuário 'zabbix' tenha permissões de execução para os comandos configurados. Pode ser interessante usar sudo para conceder 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

Reinicialização do Windows em determinada condição.

Para reiniciar automaticamente o Windows quando um problema for detectado pelo Zabbix, defina o seguinte script:

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`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

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.

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
```

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

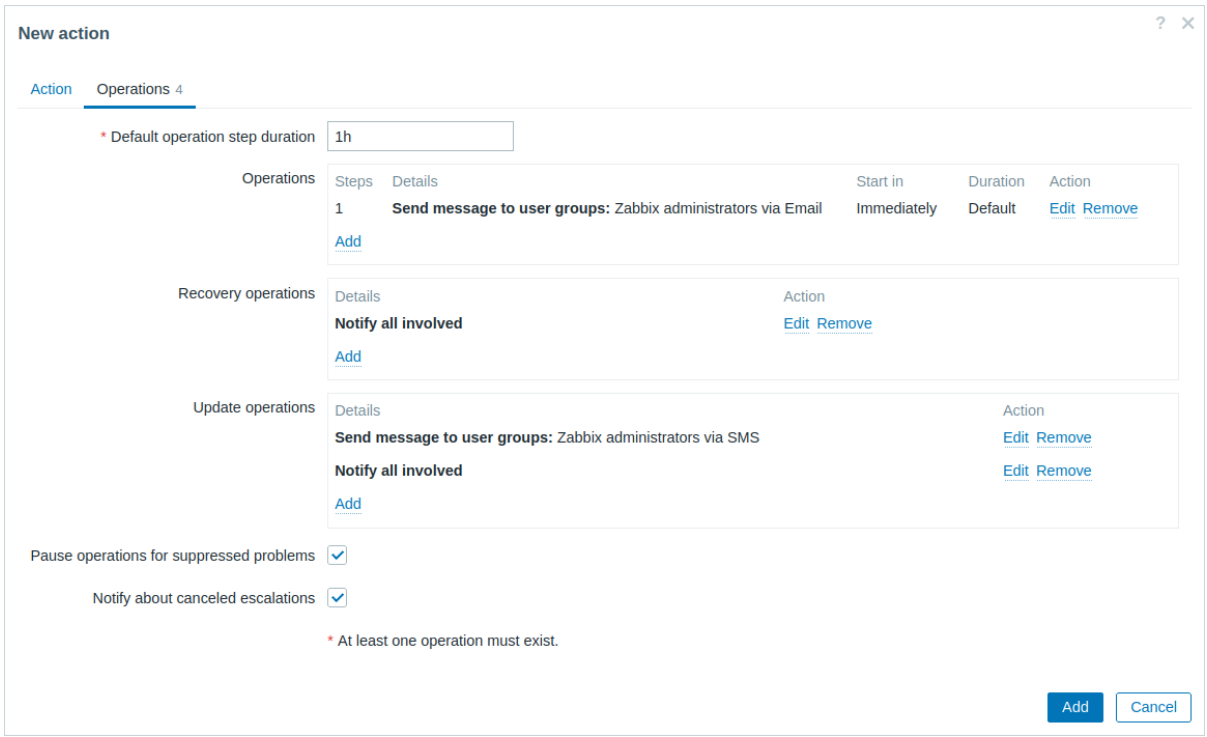
Alguns casos de uso para operações de recuperação são os seguintes:

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.

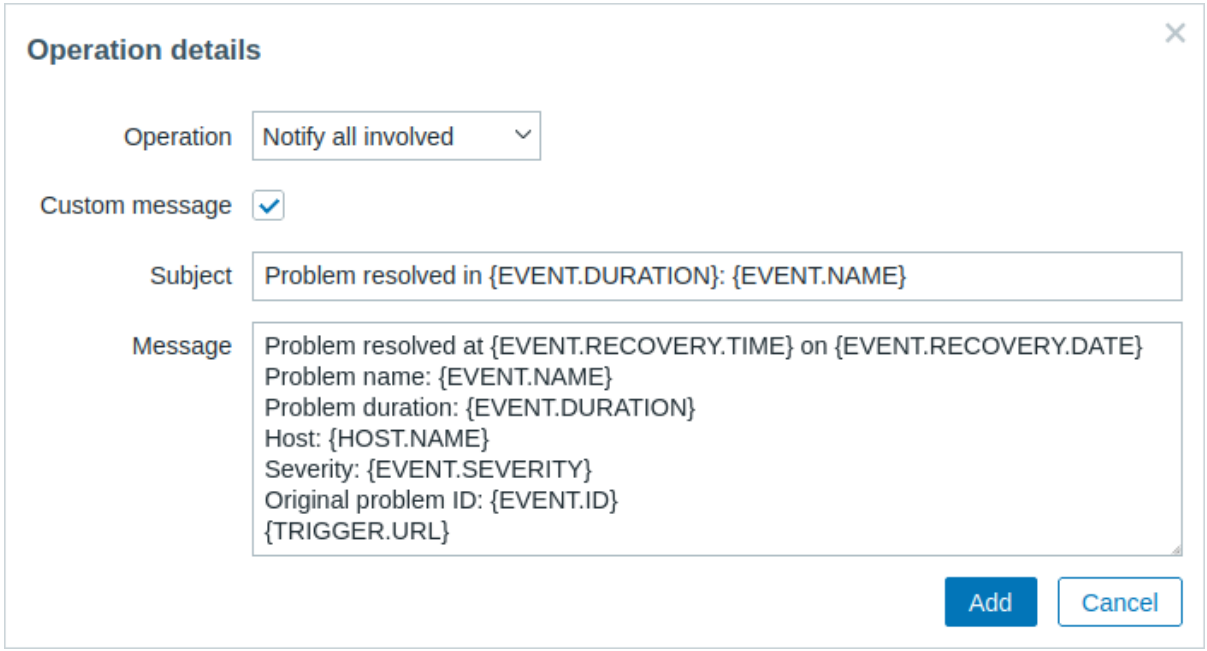
Configurando uma operação de recuperação

Para configurar uma operação de recuperação, siga estas etapas:

1. Vá para a guia *Operações* no formulário de configuração de ação.
2. Para configurar uma nova operação de recuperação, clique em *Adicionar* na seção *Operações de recuperação*. Para editar uma operação existente, clique em *Editar* ao lado da operação.
3. Configure os detalhes da operação.



Detalhes da operação de recuperação



Três tipos de operação estão disponíveis para eventos de recuperação:

- **Enviar mensagem** - envia uma mensagem de recuperação para o usuário especificado;
- **Notificar todos os envolvidos** - envia uma mensagem de recuperação para todos os usuários que foram notificados no evento de problema;
- **<nome do comando remoto>** - executa um comando remoto. Os comandos ficam disponíveis para execução se tiverem sido previamente definidos em scripts globais com *Action operation* selecionado como seu escopo.

Os parâmetros de cada tipo de operação são descritos abaixo. Todos os campos de entrada obrigatórios são 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 estiver definido em vários tipos de operação sem uma *Mensagem personalizada* especificada, notificações duplicadas não serão 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.
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 nos quais o comando será executado: **Host atual** - o comando é executado no host do trigger que causou o evento de problema. Essa opção não funcionará se houver vários hosts no trigger. **Host** - selecione o(s) host(s) nos quais o comando será executado. **Grupo de hosts** - selecione o(s) grupo(s) de hosts nos quais o comando será executado. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Assim, o comando remoto também será executado nos hosts dos grupos aninhados. Um comando em um host é executado apenas uma vez, mesmo que o host corresponda mais de uma vez (por exemplo, em vários grupos de hosts; individualmente e a partir de um grupo de hosts). A lista de destinos não tem efeito se o comando for executado no Zabbix server. Selecionar mais destinos nesse caso fará com que o comando seja executado no server mais vezes. Observe que, para scripts globais, a seleção de destinos 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.

Operações de atualização

Visão geral

As operações de atualização estão disponíveis em actions com as seguintes fontes de evento:

- *Triggers* - quando problemas são atualizados por outros usuários, ou seja, recebem comentário, são reconhecidos, a severidade foi alterada, são fechados (manualmente);
- *Services* - quando a severidade de um service foi alterada, mas o service ainda não foi recuperado.

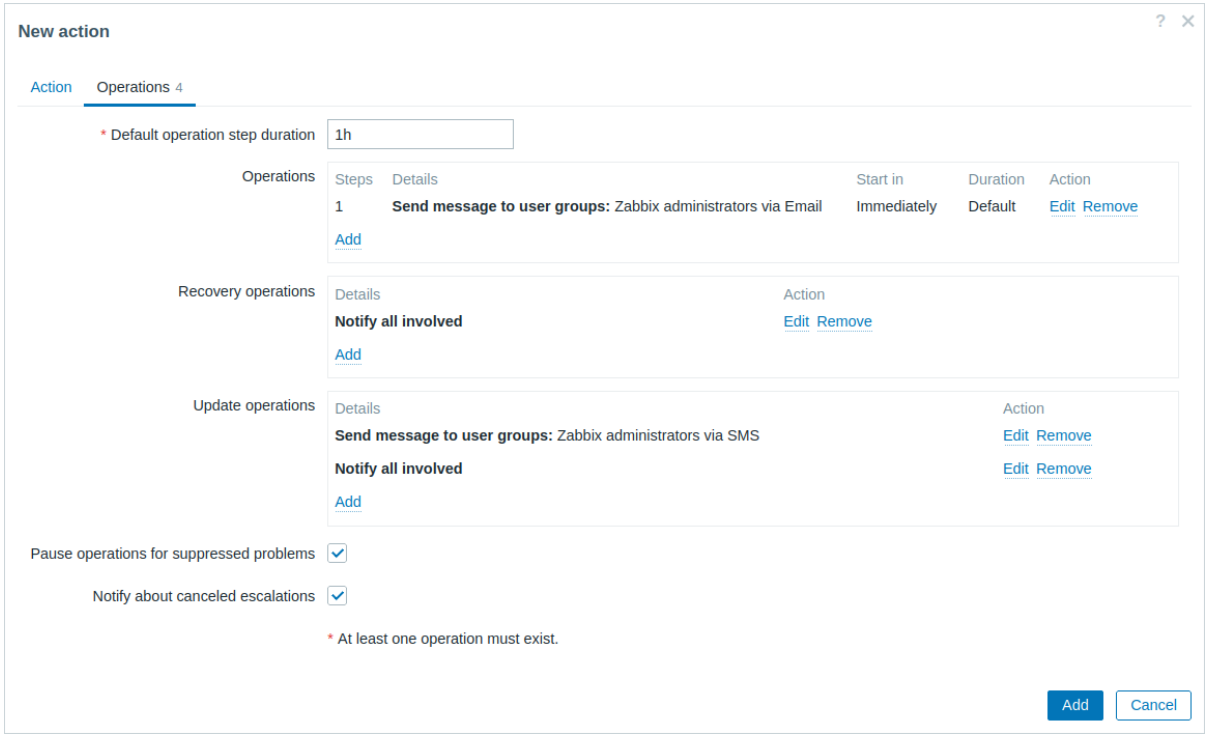
Observe que os usuários não recebem notificações sobre suas próprias atualizações.

Tanto mensagens quanto comandos remotos são suportados em operações de atualização. Embora várias operações possam ser adicionadas, a escalada não é suportada - todas as operações são atribuídas a uma única etapa e, portanto, serão executadas simultaneamente.

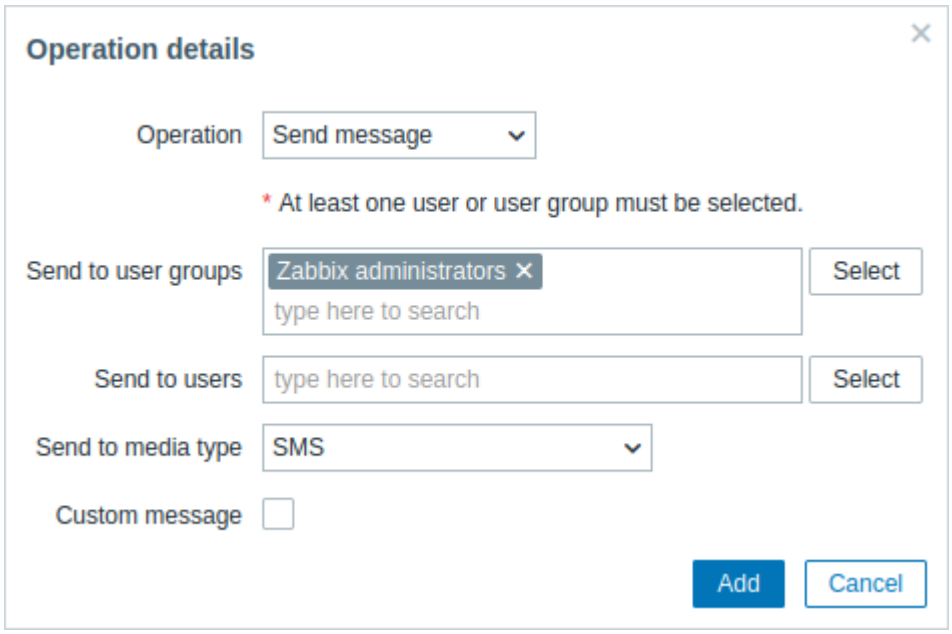
Configurando uma operação de atualização

Para configurar uma operação de atualização, siga estas etapas:

1. Vá para a guia *Operações* no formulário de configuração de ação.
2. Para configurar uma nova operação de atualização, clique em *Adicionar* na seção *Operações de atualização*. Para editar uma operação existente, clique em *Editar* ao lado da operação.
3. Configure os detalhes da operação.



Detalhes da operação de atualização



As operações de atualização oferecem o mesmo conjunto de parâmetros das operações de recuperação.

Escalações

Visão geral

Com escalonamentos, você pode criar cenários personalizados para enviar notificações ou executar comandos remotos.

Em termos práticos, 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 na **etapa de escalonamento**. Cada etapa tem uma duração.

Você pode definir tanto a duração padrão quanto uma duração personalizada de uma etapa individual. A duração mínima de uma etapa de escalonamento é de 60 segundos.

Você pode iniciar ações, como enviar notificações ou executar comandos, a partir de qualquer etapa. A etapa um é para ações imediatas. Se você quiser adiar uma ação, poderá atribuí-la a uma etapa posterior. Para cada etapa, várias ações podem ser definidas.

O número de etapas de escalonamento não é limitado.

Os escalonamentos são definidos ao configurar uma operação. Os escalonamentos são compatíveis apenas com operações de problema, não de recuperação.

Aspectos diversos do comportamento de escalonamento

Vamos considerar o que acontece em diferentes circunstâncias quando uma ação contém várias etapas de escalonamento.

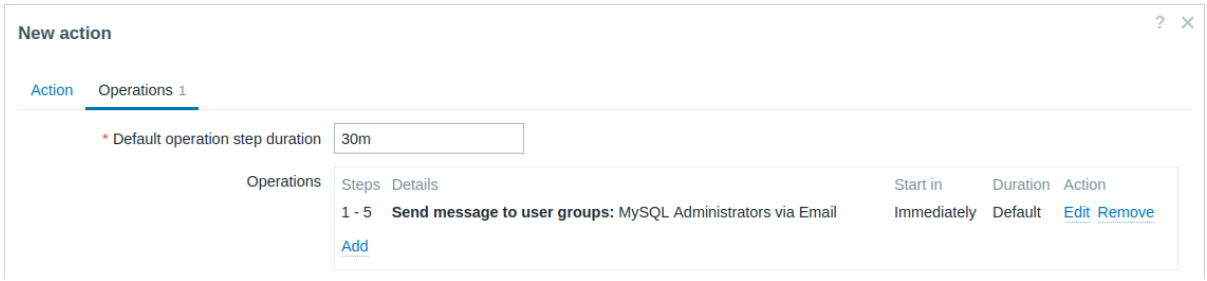
Situação	Comportamento
O host em questão entra em manutenção depois que a notificação inicial do problema é enviada	Dependendo da configuração *Pausar operações para problemas suprimidos* na configuração da ação, todas as etapas de escalonamento restantes são executadas com um atraso causado pelo período de manutenção ou sem atraso. Um período de manutenção não cancela as operações.
*O período definido na condição de ação **Período de tempo** termina depois que a notificação inicial é enviada*	Todas as etapas de escalonamento restantes são executadas. A condição *Período de tempo* não pode interromper as operações; ela tem efeito sobre quando as ações são iniciadas ou não, e não sobre as operações.
Um problema começa durante a manutenção e continua (não é resolvido) após o término da manutenção	Dependendo da configuração *Pausar operações para problemas suprimidos* na configuração da ação, todas as etapas de escalonamento são executadas a partir do momento em que a manutenção termina ou imediatamente.
Um problema começa durante uma manutenção sem dados e continua (não é resolvido) após o término da manutenção	É necessário aguardar o disparo do trigger antes que todas as etapas de escalonamento sejam executadas.
Diferentes escalonamentos ocorrem 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 relacionadas a eventos criados a cada avaliação de problema do trigger.
Durante um escalonamento em andamento (como durante o envio de uma mensagem), com base em qualquer tipo de evento: - a ação é desabilitada Com base em um evento de trigger: - o trigger é desabilitado - o host ou item é desabilitado Com base em um evento interno sobre triggers: - o trigger é desabilitado Com base em um 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 do escalonamento é enviada. A mensagem subsequente terá o texto de cancelamento no início do corpo da mensagem (*OBSERVAÇÃO: Escalonamento cancelado*), indicando o motivo (por exemplo, *OBSERVAÇÃO: Escalonamento cancelado: ação '<Action name>' desabilitada*). Dessa forma, o destinatário é informado de que o escalonamento foi cancelado e que nenhuma outra etapa será executada. Essa mensagem é enviada a todos que receberam as notificações anteriormente. O motivo do cancelamento também é registrado no arquivo de log do server (a partir do Nível de depuração 3=Warning). Observe que a mensagem *Escalonamento cancelado* também é enviada quando as operações são concluídas, mas as operações de recuperação estão configuradas e ainda não foram executadas.
Durante um escalonamento em andamento (como durante o envio de uma mensagem), a ação é excluída	Nenhuma outra mensagem é enviada. As informações são registradas no arquivo de log do server (a partir do Nível de depuração 3=Warning), por exemplo: `escalation canceled: action id:334 deleted`

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 que possuem intervalos de etapas sobrepostos e intervalos personalizados. A duração padrão da etapa da operação é de 30 minutos.

As notificações serão enviadas da seguinte forma:

- Para os administradores do MySQL às 0:00, 0:30, 1:00 e 1:30 após o início do problema.
- Para o gerente do banco de dados às 2:00 e 2:10 (a duração da etapa personalizada mais curta de 10 minutos definida na operação subsequente substitui a duração da etapa mais longa de 1 hora configurada aqui, conforme descrito em Detalhes da operação para *Duração da etapa* quando as etapas se sobrepõem).
- Para os administradores do Zabbix às 2:00, 2:10 e 2:20 após o início do problema (a duração da etapa personalizada de 10 minutos é aplicada).

- Para os usuários convidados às 4:00 após o início do problema (a duração padrão da etapa de 30 minutos entra em vigor entre as etapas 8 e 11).

Recebendo notificações 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 ”unsupported” (e vice-versa);
- quando triggers passam de ”normal” para ”unknown” (e vice-versa);
- quando regras de descoberta de baixo nível passam de ”normal” para ”unsupported” (e vice-versa).

Esta seção apresenta um passo a passo para **receber notificação** quando um item se torna unsupported.

Configuração

No geral, o processo de configuração da notificação deve ser familiar para aqueles que já configuraram alertas no Zabbix antes.

Etapa 1

Configure algum meio de comunicação, como e-mail, SMS ou script, para usar nas notificações. Consulte as seções correspondentes do manual para realizar esta tarefa.

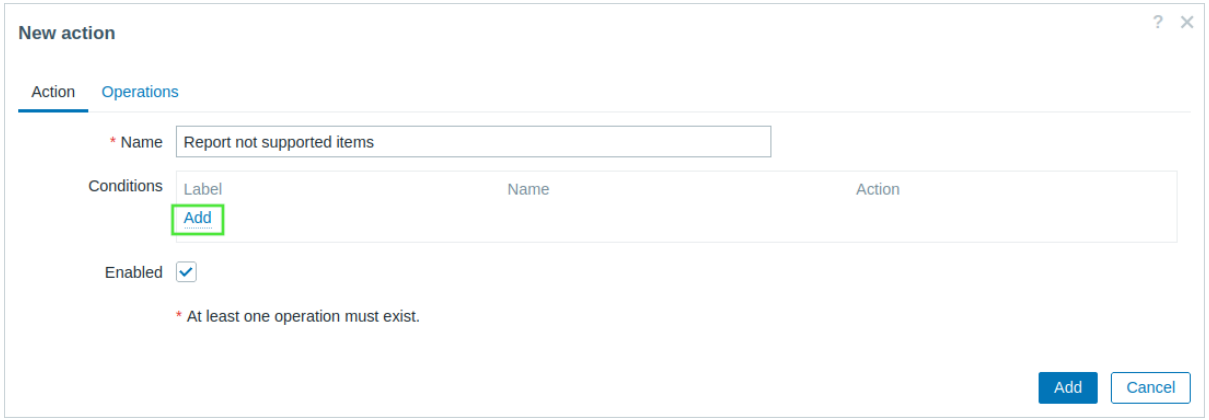
Attention:
Para notificações sobre eventos internos, a severidade padrão (’Not classified’) é usada, portanto, deixe-a marcada ao configurar a mídia do usuário se quiser receber notificações de eventos internos.

Etapa 2

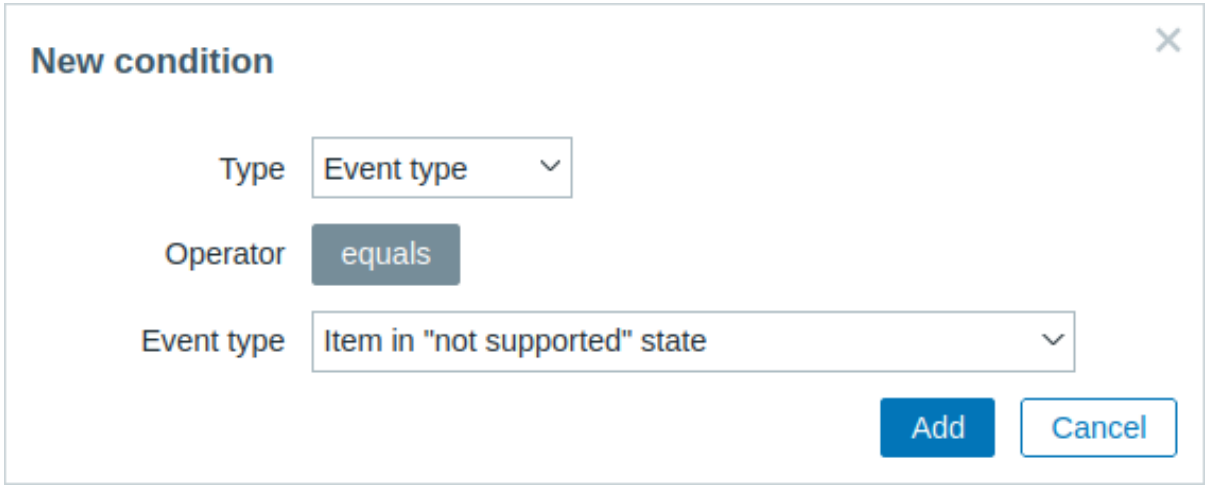
1. Vá para *Alerts* > *Actions* > *Internal actions*.
2. Clique em *Create action* no canto superior direito da página para abrir um formulário de configuração de action.

Etapa 3

1. Na aba *Action*, insira um nome para a ação.
2. Clique em *Add* no bloco *Conditions* para adicionar uma nova condição.



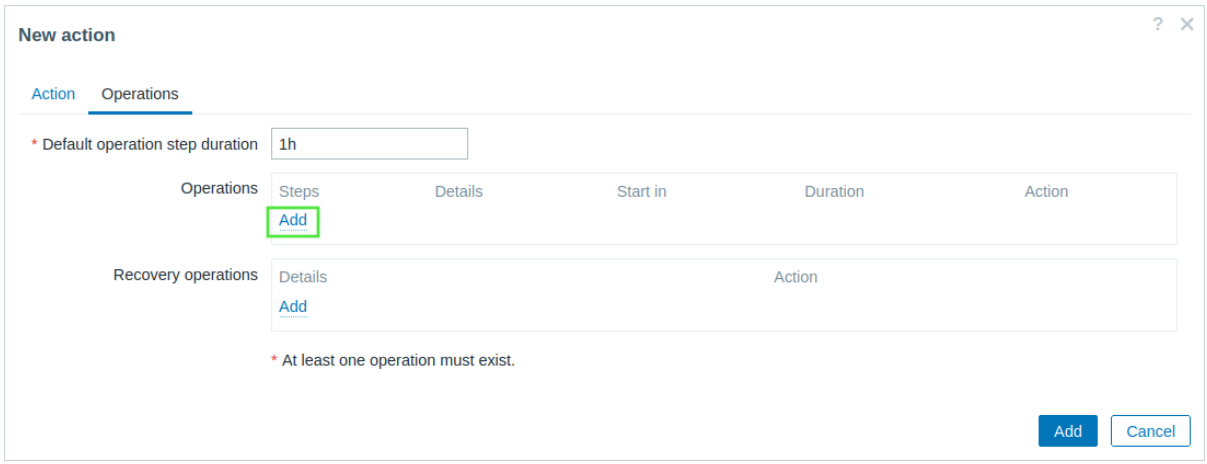
3. Na janela pop-up *New condition*, selecione ”Event type” como o tipo de condição e, em seguida, selecione ”Item in ’not supported’ state” como o tipo de evento.



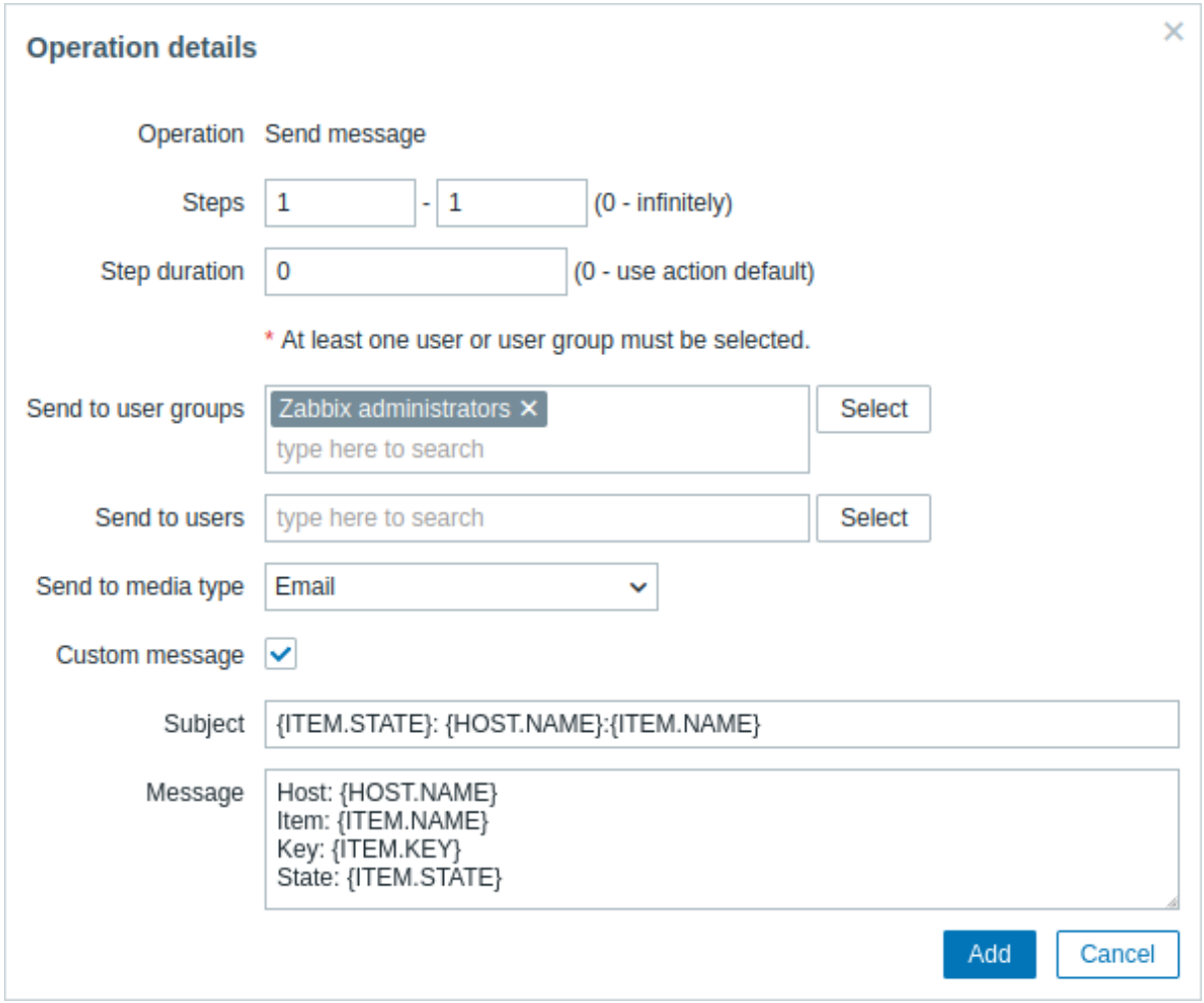
4. Clique em *Add* para listar efetivamente a condição no bloco *Conditions*.

Etapa 4

1. Na aba *Operations*, clique em *Add* no bloco *Operations* para adicionar uma nova operação.



2. Selecione alguns destinatários da mensagem (grupos de usuários/usuários) e o tipo de mídia (ou ”All”) a ser usado para a entrega.
3. Marque a caixa de seleção *Custom message* se desejar inserir o assunto/conteúdo personalizado da mensagem do problema.



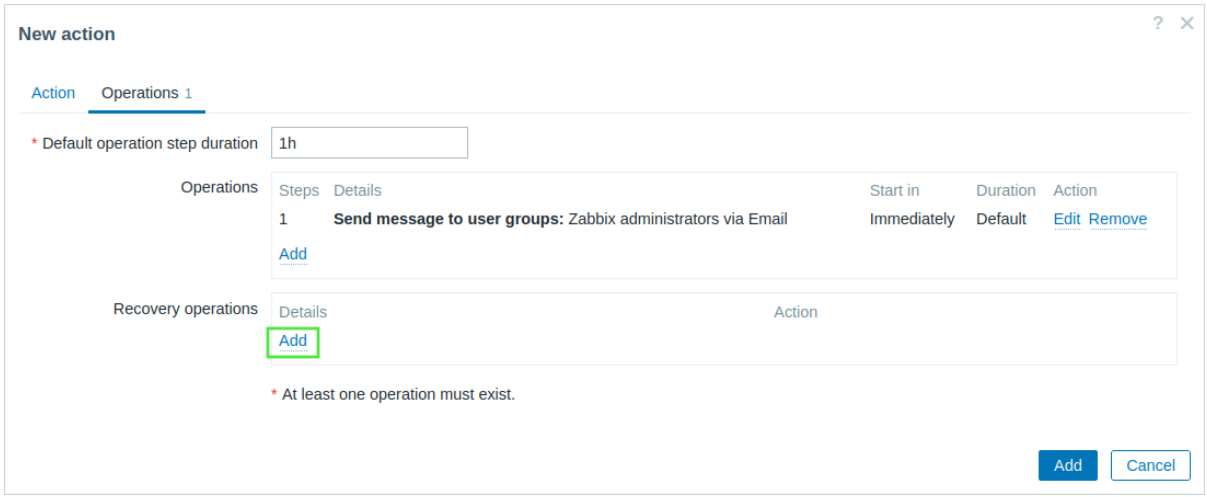
4. Clique em *Add* para listar efetivamente a operação no bloco *Operations*.

Se desejar receber mais de uma notificação, defina a duração da etapa da operação (intervalo entre as mensagens enviadas) e adicione outra etapa.

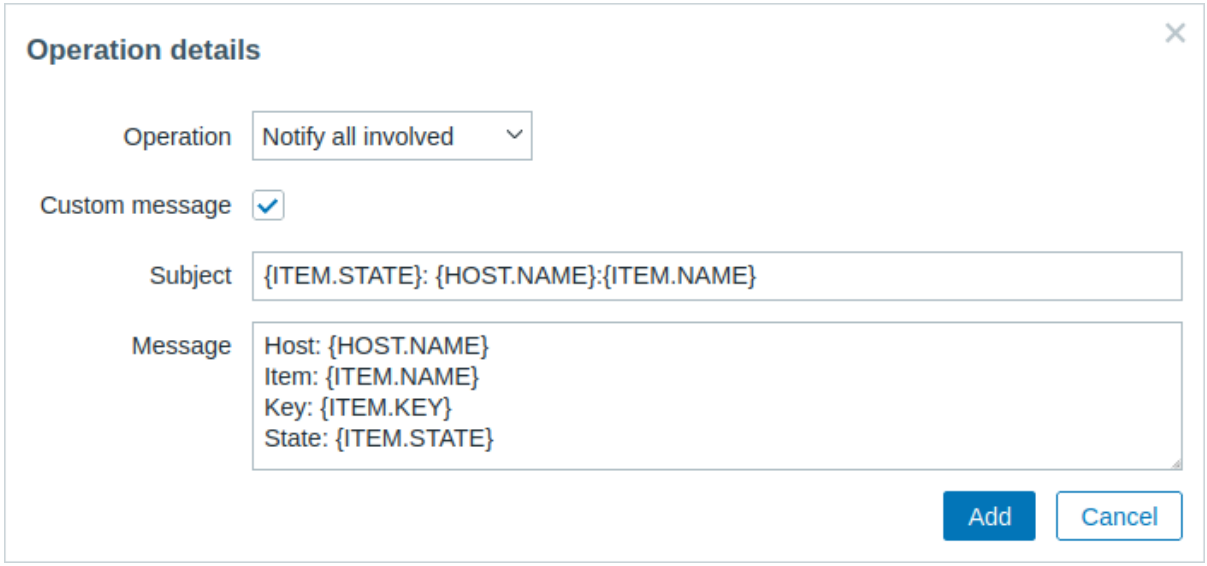
Etapa 5

O bloco *Operações de recuperação* permite configurar uma notificação de recuperação quando um item retorna ao estado normal.

1. Clique em *Adicionar* no bloco *Operações de recuperação* para adicionar uma nova operação de recuperação.



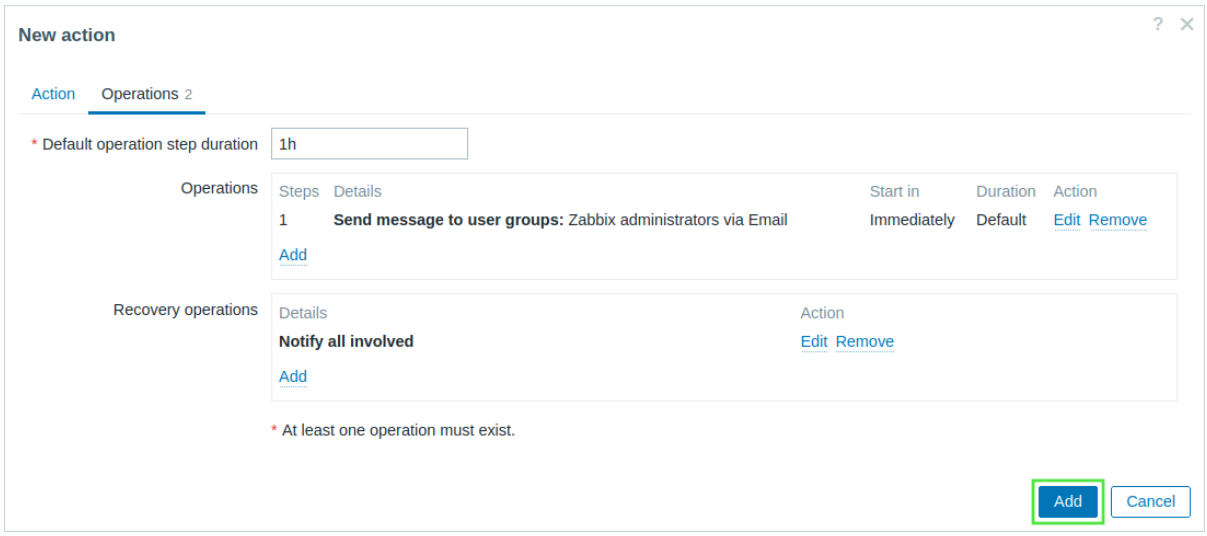
2. Selecione o tipo de operação ”Notificar todos os envolvidos”.
3. Selecione a caixa de seleção *Mensagem personalizada* se desejar inserir o assunto/conteúdo personalizado da mensagem de recuperação.



4. Clique em *Adicionar* na janela pop-up *Detalhes da operação* para listar efetivamente a operação no bloco *Operações de recuperação*.

Etapa 6

Quando terminar, clique no botão *Add* na parte inferior do formulário.



E pronto, você concluiu! Agora você pode aguardar o recebimento da sua primeira notificação do Zabbix se algum item ficar sem suporte.

Macros

Visão geral

O Zabbix oferece suporte a vários macros integrados, que podem ser usados em diversas situações. Esses macros são variáveis identificadas por uma sintaxe específica:

`{MACRO}`

Os macros são resolvidos para um valor específico, dependendo do contexto.

O uso eficaz de macros permite economizar tempo e tornar a configuração do Zabbix mais transparente.

Em um dos usos típicos, um macro pode ser usado em um template. Assim, um trigger em um template pode ser nomeado como "Processor load is too high on {HOST.NAME}". Quando o template é aplicado ao host, como o Zabbix server, o nome será resolvido para "Processor load is too high on Zabbix server" quando o trigger for exibido na seção Monitoring.

Os macros podem ser usados nos parâmetros de chave de um item. Um macro pode ser usado em apenas uma parte do parâmetro, por exemplo, `item.key[server_{HOST.HOST}_local]`. Não é necessário colocar o parâmetro entre aspas duplas, pois o Zabbix cuidará de quaisquer símbolos especiais ambíguos, caso estejam presentes no macro resolvido.

Existem outros tipos de macros no Zabbix.

O Zabbix oferece suporte aos seguintes macros:

- {MACRO} - macro integrado (consulte a lista completa)
- {<macro>.<func>(<params>)} - funções de macro
- {$MACRO} - macro definido pelo usuário, opcionalmente com contexto
- {#MACRO} - macro para descoberta de baixo nível
- {?EXPRESSION} - macro de expressão

Funções de macro

Visão geral

As funções de macro oferecem a capacidade de personalizar valores de macro (por exemplo, encurtar ou extrair substrings específicas), tornando-os mais fáceis de usar.

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 separada por vírgulas, que deve estar entre **aspas duplas** se:
  - começar com um espaço ou aspas duplas;
  - contiver 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 indicado que apenas uma macro é esperada (por exemplo, ao configurar macros de host ou filtros de regra de descoberta de baixo nível).

É suportada uma única função por macro; várias funções de macro em cadeia não são suportadas.

> **Note:**
> Consulte os exemplos de escape para casos em que funções de macro são usadas dentro de outros contextos (função, chave de item, outra macro etc).

Funções compatíveis

As funções são listadas sem informações adicionais.
Clique na função para ver todos os detalhes.

Função	Descrição
btoa	Codificação do valor da macro em Base64.
fmtnum	Formatação de números para controlar a quantidade de dígitos exibidos após o ponto decimal.
fmttime	Formatação de hora.
htmldecode	Decodificação do valor da macro a partir da codificação HTML.
htmlencode	Codificação do valor da macro em HTML.
iregsub	Extração de substring por correspondência de expressão regular (sem distinção entre maiúsculas e minúsculas).
jsonpath	Retorna o resultado do JSONPath.
lowercase	Transformação dos caracteres do valor da macro para minúsculas.

Função	Descrição
regrepl	Substituição de caractere ou substring no valor da macro.
regsub	Extração de substring por correspondência de expressão regular (com distinção entre maiúsculas e minúsculas).
tr	Transliteração dos caracteres do valor da macro.
uppercase	Transformação dos caracteres do valor da macro para maiúsculas.
urldecode	Decodificação do valor da macro a partir da codificação de URL.
urlencode	Codificação do valor da macro em URL.
xmlxpath	Retorna o resultado do XPath XML.

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 tempo.
 Observe que esta função pode ser usada com macros que resolvem para um valor em um dos seguintes formatos de tempo:

- `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) - deslocamento de tempo aplicado ao tempo 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 múltiplas etapas 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 назад; `-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:

```
{{TIMESTAMP}.fmttime(%B)} - retornará "October" a partir de um valor recebido de "1633098961"
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - retornará "1 September" a partir de um valor recebido de "1633098961"
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - retornará a representação local da data e hora; por exemplo, "07/15
{{EVENT.TIME}.fmttime("%x %X")} - retornará a representação local da data e hora; por exemplo, "07/15/26 1
```

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
&	&
<	<
>	>
"	"
'	'

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
```

jsonpath(pattern,<default>)

Retorna o resultado do JSONPath.
 Tipos de valor compatíveis: *String*, *Text*, *Log*.

Parâmetros:

- **pattern** - o padrão obrigatório;
- **default** - o valor de fallback opcional caso a consulta JSONPath não retorne dados. Observe que um erro é retornado em caso de outros erros.

Comentários:

- A função falha com um erro se:
  - O JSON for inválido.
  - A consulta JSONPath falhar.
  - O resultado do JSONPath não contiver dados e o parâmetro `default` não for fornecido.

Exemplo:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

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. **\1 - \9** placeholders são suportados em strings de substituição para capturar grupos.

Comentários:

- Todas as ocorrências que correspondem ao pattern são substituídas.
- Os patterns e replacements 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")} - substituirá "oldParam" por "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - todos os caracteres não alfabéticos serão escapados com uma ba
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - removerá uma barra invertida final (por exemplo, para s
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - substituirá 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"
```

xmlxpath(pattern,<default>)

Retorna o resultado do XPath XML.
 Tipos de valor compatíveis: *String*, *Text*, *Log*.

Parâmetros:

- **pattern** - o padrão obrigatório;
- **default** - o valor de fallback opcional caso a consulta XPath não retorne dados. Observe que uma falha será retornada em caso de outros erros.

Comentários:

- A função falha com um erro se:
  - O XML for inválido.
  - A consulta XPath falhar.
  - O resultado do XPath não contiver dados e o parâmetro `default` não for fornecido.
  - O Zabbix tiver sido compilado sem suporte a XML.

Exemplo:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

Exemplos adicionais

A tabela abaixo mostra mais exemplos de uso de funções de macro.

`{#IFALIAS}` é uma macro LLD e só é definida em contextos de descoberta de baixo nível (regras de descoberta, protótipos e os items/triggers criados a partir deles). Usá-la fora do LLD deixará o token não expandido.

Função de macro	Valor recebido	Saída
{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}	123Log line	Problem
{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}	123 Log line	Problem
{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	Log line	Problem ID:
{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}	MySQL crashed errno 123	Problem ID: MySQL_123
{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}	123 Log line	*UNKNOWN* (expressão regular inválida)
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}	customername_1	customername
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}	customername_1	1
{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}	customername_1	{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)} (expressão regular inválida)
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"customername"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}	customername_1	{$MACRO:"1"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"} (expressão regular inválida)
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+\", \1)}\"}"	customername_1	"{$MACRO:"\customername\"}"
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+\", \2)}\"}"	customername_1	"{$MACRO:"\1\"}"
"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"	customername_1	"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}" (expressão regular inválida)

Visualizando valores completos de items

Valores longos das macros {ITEM.VALUE} e {ITEM.LASTVALUE} resolvidas para items de texto/log são truncados para 20 caracteres em alguns locais do frontend. Para ver os valores completos dessas macros, você pode usar funções de macro, por exemplo:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Veja também: detalhes das macros {ITEM.VALUE} e {ITEM.LASTVALUE} macro details.

Macros de usuário

Visão geral

As macros de usuário são compatíveis com o Zabbix para oferecer maior flexibilidade, além das macros compatíveis por padrão.

As macros de usuário podem ser definidas no 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 no nível do host (verificadas primeiro).
2. Macros definidas para os templates de primeiro nível do host (ou seja, templates vinculados diretamente ao host), ordenadas pelo ID do template.
3. Macros definidas para os templates de segundo nível do host, ordenadas pelo ID do template.
4. Macros definidas para os templates de terceiro nível do host, ordenadas pelo ID do template, e assim por diante.
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 ordem crescente de profundidade. Se ainda assim não for encontrada, uma macro global será usada, se existir.

Warning:
Se uma macro com o **mesmo nome** existir em vários templates vinculados no mesmo nível, será usada a macro do template com o menor ID. Portanto, ter macros com o mesmo nome em vários templates representa um risco de configuração.

Se o Zabbix não conseguir encontrar uma macro, ela não será resolvida.

Attention:
As macros (incluindo macros de usuário) permanecem não resolvidas na seção Configuration (por exemplo, na lista de triggers) por design, para tornar configurações complexas mais transparentes.

As 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 da expressão do trigger (consulte os exemplos)
- muitos outros locais - consulte 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.

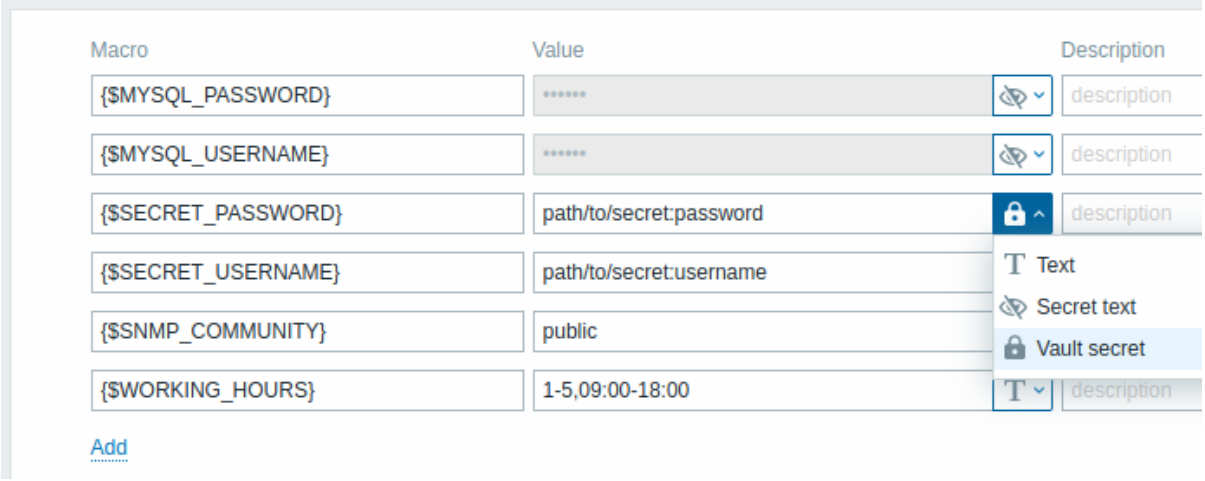
Note:
Recomenda-se usar macros de host em vez de macros globais, porque adicionar, atualizar ou excluir macros globais força uma atualização incremental da configuração para todos os hosts. Para mais informações, consulte Passive and active agent checks.

Configuração

Para definir macros de usuário, vá para o local correspondente no frontend:

- Para macros globais, acesse *Administration* > *Macros*.
- Para macros no nível de host e template, abra as propriedades do host ou template e procure a aba *Macros*.

Uma macro de usuário tem os seguintes atributos:



Parameter	Description
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 de macro: **A-Z** (somente maiúsculas) , **0-9** , _ , **.**
Value	Valor da macro. Há suporte para três tipos de valor: **Text** (padrão) - valor em texto simples **Secret text** - o valor é mascarado com asteriscos **Vault secret** - o valor contém um caminho/consulta para um vault secret. Para alterar o tipo de valor, clique no botão no final do campo de entrada do valor. O comprimento máximo de um valor de macro de usuário é de 2048 caracteres.
Description	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 itens ou triggers de template, considere adicionar essas macros ao template também (mesmo que elas estejam definidas globalmente); dessa forma, macros do tipo *Text* funcionarão como esperado após exportar o template para XML e importá-lo em outro sistema (os valores de macros secretas não são exportados)
- se macros de usuário forem usadas em expressões de trigger, essas macros só serão resolvidas 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 tenha uma regra de descoberta em baixo nível com protótipos de host, os hosts descobertos herdarão todas as macros de usuário definidas nesse host

Exemplos

Exemplo 1

Uso de macro no nível do host na chave do item "Status do daemon SSH":

`net.tcp.service[ssh,,{$SSH_PORT}]`

Este item pode ser atribuído a vários hosts, desde que o valor de `{$SSH_PORT}` esteja definido nesses hosts.

Exemplo 2

Uso de macro no nível do host no trigger "A carga da CPU está muito alta":

`last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}`

Esse trigger seria criado no template, não editado em hosts individuais.

> **Note:**
> Se 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 desta forma: `SOME_PERIOD => #3`.

Exemplo 3

Uso de duas macros no trigger "A carga da CPU está muito alta":

`min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}`

Observe que uma macro pode ser usada como parâmetro de uma função de trigger; neste exemplo, a função `min()`.

Exemplo 4

Sincronize a condição de indisponibilidade do agent com o intervalo de atualização do item:

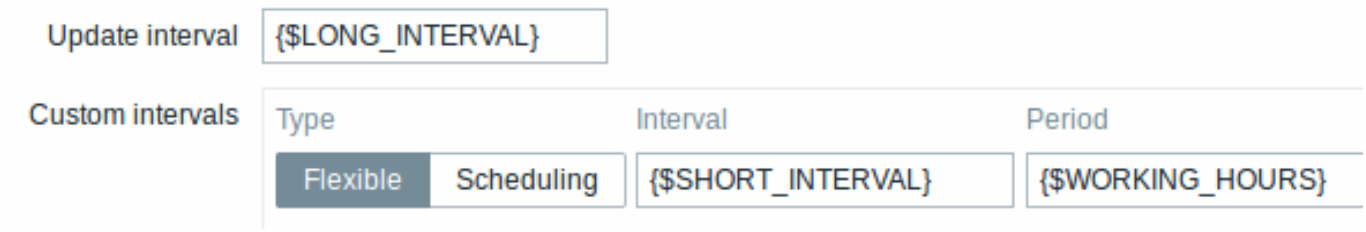
- defina a macro `{$INTERVAL}` e use-a no intervalo de atualização do item;
- use `{$INTERVAL}` como parâmetro do trigger de indisponibilidade do agent:

`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* > *Interface gráfica*;
- use-a no campo *Quando ativo* em *Usuários* > *Usuários*, na aba *Mídia* de um usuário;
- use-a para configurar uma coleta mais frequente de itens durante o horário de trabalho:

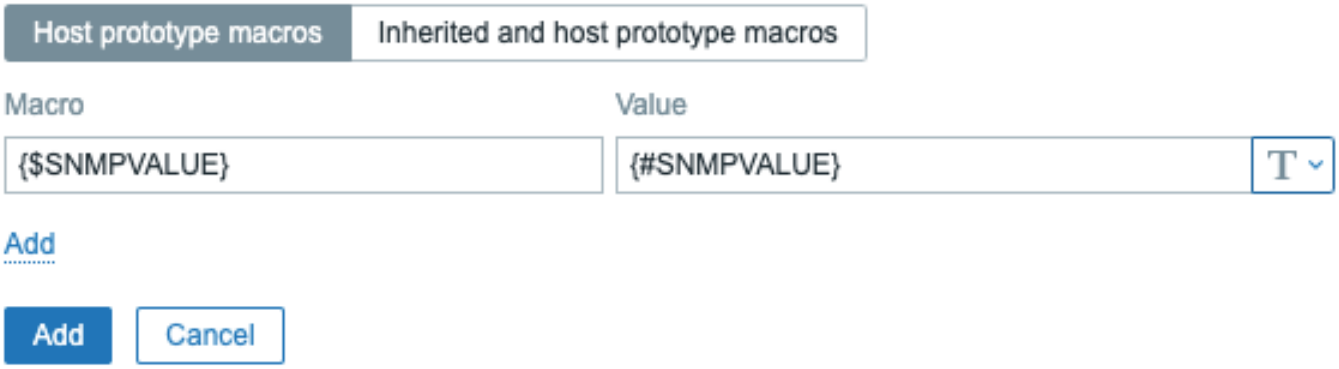


- use-a na condição de ação *Período de tempo*;
- ajuste o horário de trabalho em *Administração* > *Macros*, se necessário.

Exemplo 6

Use uma macro de protótipo de host para configurar itens para hosts descobertos:

1. Em um protótipo de host, defina a macro de usuário `{$SNMPVALUE}` com a macro de descoberta de baixo nível `{#SNMPVALUE}` como valor:



2. Atribua o template *Generic SNMPv2* ao protótipo de host.
3. Use `{$SNMPVALUE}` no campo *SNMP OID* dos itens do template *Generic SNMPv2*.

Contexto de macro de usuário

Consulte macros de usuário com contexto.

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 para o contexto.

O contexto é anexado ao nome da macro; a sintaxe depende de o contexto ser 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 macros de usuário. Se o prefixo `regex:` for usado em outro local como contexto de macro de usuário, por exemplo, em uma expressão de trigger, ele será tratado como contexto estático.

As aspas do contexto são opcionais (consulte também as observações 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). Equivalente 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 obter limites mais flexíveis em expressões de trigger (com base nos valores recuperados pela descoberta em 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 em baixo nível pode ser usada como contexto de macro em um protótipo de trigger para a descoberta de sistemas de arquivos montados:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Após a descoberta, diferentes limites de pouco espaço em disco serão aplicados em triggers, dependendo dos pontos de montagem ou dos 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.

Macros de usuário secretos

Visão geral

O Zabbix fornece duas opções para proteger informações confidenciais nos valores das macros de usuário:

- Texto secreto
- Segredo do Vault

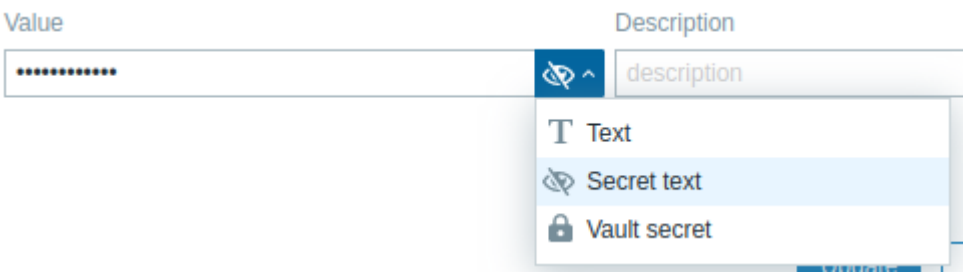
Note:
Embora o valor de uma macro secreta fique oculto, ele pode ser revelado por meio do uso em items. Por exemplo, em um script externo, uma instrução `echo` referenciando uma macro secreta pode ser usada para revelar o valor da macro para o frontend, porque o Zabbix server tem acesso ao valor real 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 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 *Value* e selecione a opção *Secret text*:



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 *Value* e clique no botão *Set new value* (aparece ao passar o mouse):



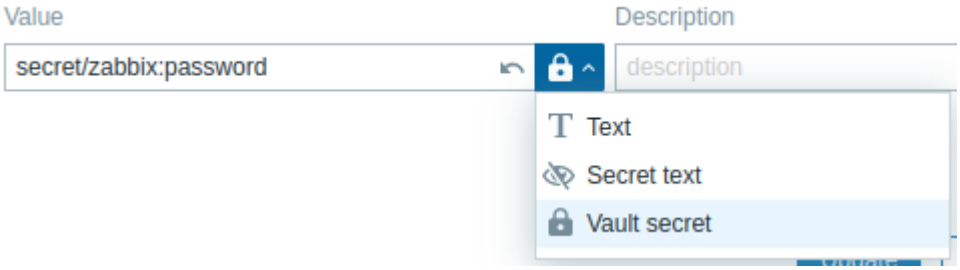
Quando você clica no botão *Set new value* (ou altera o tipo de valor da macro), o valor atual será apagado. Você pode restaurar o valor original clicando na seta no final do campo *Value* (disponível apenas antes de salvar a nova configuração). Observe que restaurar o valor original não o revelará.

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 *Value* e selecione a opção *Vault secret*:



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 de macros secretas do Vault são recuperados do vault pelo Zabbix server (e pelo Zabbix proxy, se *Resolve secret vault macros by* estiver definido como *Zabbix server and 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 *Resolve secret vault macros by* estiver definido como *Zabbix server*, então os segredos do vault serão recuperados somente pelo server, e o Zabbix proxy receberá os valores das macros secretas do Vault do Zabbix server em cada sincronização de configuração e os armazenará em seu próprio cache de configuração. Isso significa que um Zabbix proxy não pode iniciar a coleta de dados após uma reinicialização até receber a atualização de configuração do Zabbix server.

Para atualizar manualmente os valores secretos do vault, use a opção de controle em tempo de execução `secrets_reload` (somente 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 o valor de uma macro não puder ser recuperado com sucesso, o item correspondente que usa esse valor ficará não suportado.

Locais sem mascaramento

Esta lista fornece os locais dos parâmetros em que os valores de macro secretos não são mascarados.

Note:
Os valores de macro secretos permanecerão mascarados nos locais abaixo se forem referenciados indiretamente. Por exemplo, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} macros integradas usadas em tipos de mídia (parâmetros Script ou Webhook) serão resolvidas para chaves de item que contêm macros secretos mascarados, como `net.tcp.port[******,******]` em vez de `net.tcp.port[192.0.2.1,80]`.

Contexto	Parâmetro
Items, 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*
agent SNMP	*Comunidade SNMP*
	Nome do contexto (SNMPv3)
	Nome de segurança (SNMPv3)
	Frase secreta de autenticação (SNMPv3)
	Frase secreta de privacidade (SNMPv3)
agent HTTP	*URL*
	Campos de consulta
	Corpo da requisição
	Cabeçalhos
	Nome de usuário
	Senha
	Senha da chave SSL
Script	*Parâmetros*
	Script
Browser	*Parâmetros*
	Script
Monitor de banco de dados	*Consulta SQL*
agent TELNET	*Script*
	Nome de usuário
	Senha
agent SSH	*Script*
	Nome de usuário
	Senha
Verificação simples	*Nome de usuário*
	Senha
agent JMX	*Nome de usuário*
	Senha
Pré-processamento do valor do 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 de postagem
	Postagem bruta
Autenticação do cenário web	*Usuário*
	Senha
	Senha da chave SSL

Contexto	Parâmetro
Conectores	
Conector	*URL*
	Nome de usuário
	Senha
	Token
	Proxy HTTP
	Arquivo de certificado SSL
	Arquivo de chave SSL
	Senha da chave SSL
Descoberta de rede	
SNMP	*Comunidade SNMP*
	Nome do contexto (SNMPv3)
	Nome de segurança (SNMPv3)
	Frase secreta de autenticação (SNMPv3)
	Frase secreta de privacidade (SNMPv3)
Scripts globais	
Webhook	*Script JavaScript*
	Valor do parâmetro do script JavaScript
Telnet	*Nome de usuário*
	Senha
SSH	*Nome de usuário*
	Senha
Script	*Script*
Tipos de mídia	
Script	*Parâmetros do script*
Webhook	*Parâmetros*
Gerenciamento IPMI	
Host	*Nome de usuário*
	Senha

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 usada em uma regra LLD e retorna valores reais do nome do sistema de arquivos, interface de rede, OID SNMP, etc.

Essas macros podem ser usadas para criar protótipos de entidades (protótipos de item, trigger, gráfico, regra LLD, host e grupo de host). Durante a descoberta de sistemas de arquivos, interfaces de rede, máquinas virtuais, etc., essas macros são substituídas por valores reais e são a base para a criação de entidades reais.

Algumas macros de descoberta de baixo nível vêm "pré-empacotadas" com a função LLD no Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. No entanto, não é obrigatório aderir a esses nomes ao criar uma regra de descoberta de baixo nível personalizada. Então você pode usar qualquer outro nome de macro LLD e referenciar esse nome.

Tipos de dados suportados

Ao definir regras de descoberta personalizadas, os valores das propriedades retornados em objetos JSON para macros LLD devem ser um dos seguintes tipos primitivos:

- string
- number
- boolean

Arrays, objetos e valores nulos não são suportados. Qualquer macro LLD que se refira a esse valor permanecerá não expandida e aparecerá literalmente (por exemplo, `'{#MY_MACRO}'`) durante o pré-processamento e criação do item.

Locais suportados

As macros LLD podem ser usadas:

- no filtro da regra de descoberta de baixo nível
- em regras de descoberta de baixo nível aninhadas, em
  - parâmetros de pré-processamento JSONPath
  - campo JSONPath para macros LLD personalizadas
- 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 solicitação 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 gráficos 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 estática.

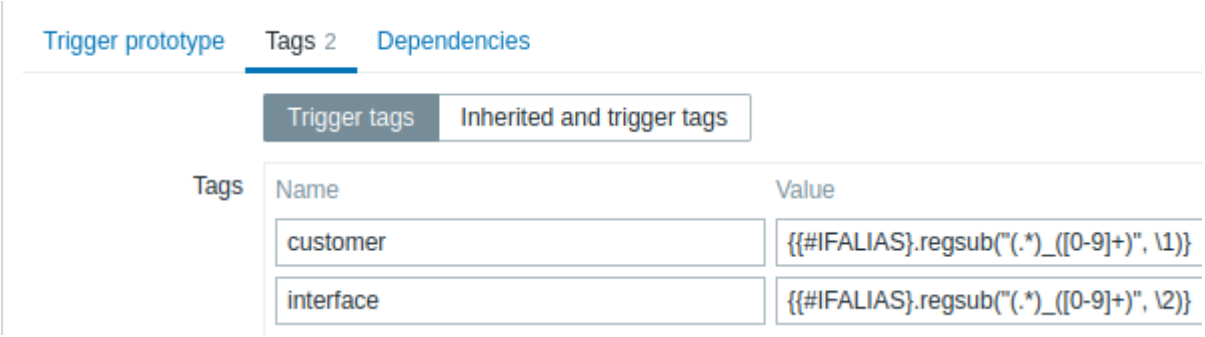
Uso de funções de macro

As funções de macro são compatíveis com macros de descoberta de baixo nível (exceto na filtragem da regra de descoberta de baixo nível), permitindo extrair uma determinada parte do valor da macro usando uma expressão regular.

Por exemplo, talvez você queira extrair o nome do cliente e o número da interface da macro LLD a seguir 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 delimitados por aspas. Portanto, o parâmetro que contém uma função de macro deve estar entre aspas. O caractere de barra invertida (\) deve ser usado para escapar aspas duplas dentro do parâmetro. Exemplo:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Para obter mais informações sobre a sintaxe das funções de macro, consulte: Funções de macro

As funções de macro são compatíveis com macros de descoberta de baixo nível desde o Zabbix 4.0.

Protótipos de item sem parâmetros de chave

Quando os parâmetros de chave do item não são usados, coloque a macro LLD dentro dos colchetes do parâmetro item key `[...]` para que seja tratada como um parâmetro e substituída durante a descoberta:

```
v_[{#MACRO}]
```

Notas de rodapé

[1] Nos campos marcados com [1], uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas.

Macros de expressão

Visão geral

Macros de expressão permitem que você realize cálculos em campos.

O valor delas é calculado primeiro resolvendo quaisquer macros internas e, em seguida, avaliando a expressão resultante.

Sintaxe:

```
{?EXPRESSION}
```

`EXPRESSION` usa a mesma sintaxe e oferece suporte às mesmas functions que as trigger expressions.

Exemplo:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Para locais que oferecem suporte a macros de expressão, procure por "{?EXPRESSION}" na tabela de supported macro.

Notas de uso:

- Use macros {FUNCTION.*} para referenciar valores de função de trigger expressions/recovery expressions.
- 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 completamente o host 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 template linking.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Veja também Trigger expression examples para um exemplo de uso de uma macro de expressão em um nome de evento.

Usuários e grupos de usuários

Visão geral

Todos os usuários no Zabbix acessam a aplicação Zabbix por meio 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.

Configurando um usuário

Visão geral

A instalação inicial do Zabbix tem dois usuários predefinidos:

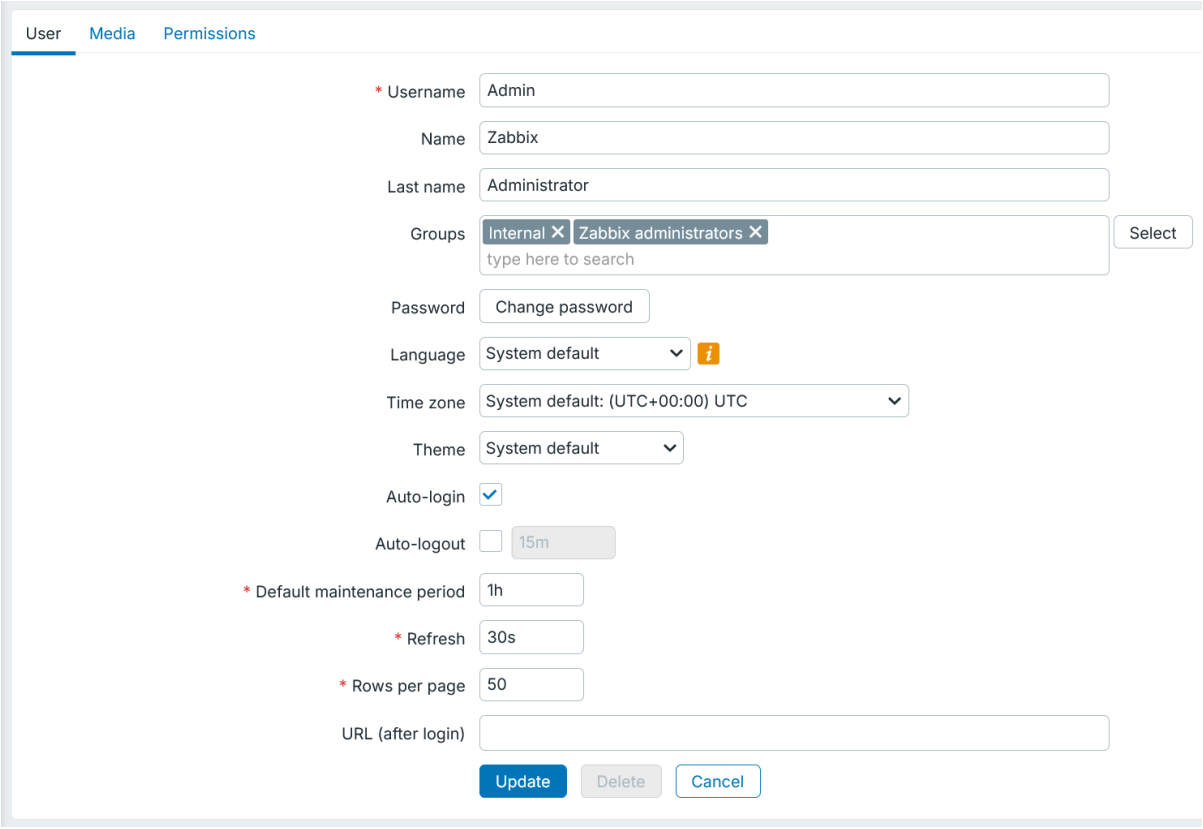
- *Admin* - um superusuário do Zabbix com permissões completas.
- *guest* - um usuário especial do Zabbix. O usuário 'guest' é desabilitado por padrão. Para habilitar o usuário, remova o grupo predefinido *Disabled* dos grupos de usuários. Você pode fazer login com esse usuário e acessar as páginas de monitoramento no Zabbix. Observe que, por padrão, 'guest' não tem permissões sobre os objetos do Zabbix.

Para configurar um usuário:

1. Acesse *Users*> *Users*.
2. Clique em *Create user* (ou no nome de um usuário para editar um usuário existente).
3. Edite os atributos do usuário no formulário.

Atributos gerais

A aba *Usuário* contém os atributos gerais do usuário:



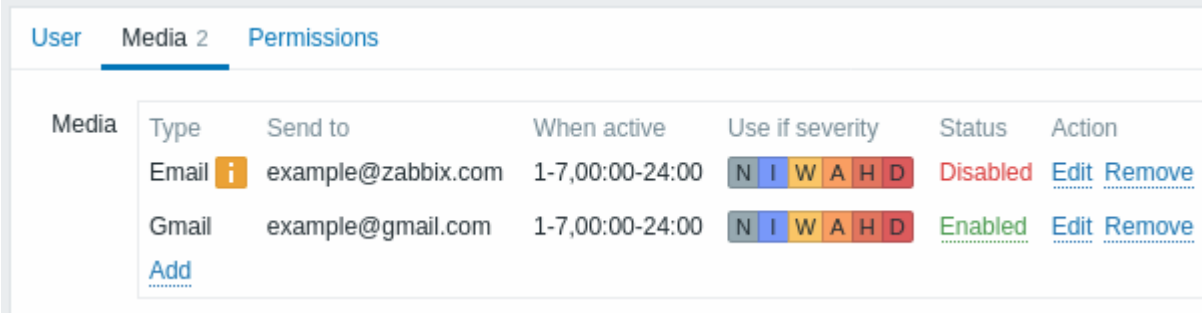
Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome de usuário	Nome de usuário exclusivo, usado como nome de login.
Nome	Primeiro nome do usuário (opcional). Se não estiver vazio, ficará visível nas informações de reconhecimento e nas informações do destinatário da notificação.
Sobrenome	Sobrenome do usuário (opcional). Se não estiver vazio, ficará 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 possui preenchimento automático; 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. Como alternativa, clique em *Selecionar* para adicionar grupos. Clique em 'x' para remover o grupo selecionado. A associação a grupos de usuários determina a quais grupos de hosts e hosts o usuário terá acesso.
Senha	Dois campos para inserir a senha do usuário ou um botão *Alterar senha* se o usuário já existir. Clicar no botão *Alterar senha* abre dois campos para inserir uma nova senha. Para o usuário com a *função de superadministrador* que estiver alterando a própria senha, clicar no botão *Alterar senha* abrirá um campo adicional para inserir a senha atual (antiga). Após uma alteração de senha bem-sucedida, o usuário cuja senha foi alterada será desconectado de todas as sessões ativas. Observe que a senha só pode ser alterada para usuários que utilizam a autenticação interna do Zabbix.
Idioma	Idioma do frontend do Zabbix. A extensão php gettext é necessária para que as traduções funcionem.
Fuso horário	Selecione o fuso horário para substituir o fuso horário global no nível do usuário ou selecione **Padrão do sistema** para usar as configurações globais de fuso horário.

Parâmetro	Descrição
Tema	Define a aparência do frontend: **Padrão do sistema** - usa as configurações padrão do sistema **Azul** - tema azul padrão **Azul (clássico)** - tema azul padrão usando fontes legadas **Escuro** - tema escuro alternativo **Azul escuro** - tema azul escuro **Escuro (clássico)** - tema escuro alternativo usando fontes legadas **Claro de alto contraste** - tema claro com alto contraste **Escuro de alto contraste** - tema escuro com alto contraste
Login automático	Marque esta caixa de seleção para que o Zabbix memorize o usuário e faça 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 habilitado para uma reautenticação contínua. Ao fazer login sem *Lembrar por 30 dias*: - O login automático é habilitado para uma reautenticação contínua. - O logout automático continua sendo controlado pela configuração de tempo limite padrão. Cookies do navegador são usados para isso.
Logout automático	Com esta caixa de seleção marcada, o usuário será desconectado automaticamente após o número de segundos definido (mínimo de 90 segundos, máximo de 1 dia). Observe que esta configuração será substituída se *Lembrar por 30 dias* estiver habilitado, pois a sessão será estendida por todo o período. Sufixos de tempo são compatíveis, por exemplo, 90s, 5m, 2h, 1d. Observe que esta opção não funcionará: * Se a opção de configuração global ”Mostrar aviso se o Zabbix server estiver inativo” estiver habilitada e o frontend do Zabbix permanecer aberto. * Quando as páginas do menu Monitoramento realizarem atualizações de informações em segundo plano. * Se o login for feito com a opção *Lembrar de mim por 30 dias* marcada.
Período de manutenção padrão	Define o período de manutenção padrão para manutenções ad hoc.
Atualização	Define a taxa de atualização usada para gráficos, dados em texto simples etc. Pode ser definida como 0 para desabilitar. Sufixos de tempo são compatíveis, por exemplo, 90s, 5m, 1h.
Linhas por página	Você pode determinar quantas linhas por página serão exibidas nas listas.
URL (após o login)	Você pode fazer com que o Zabbix transfira o usuário para uma URL específica após um 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 enviar notificações.



Clique em *Adicionar* para atribuir uma mídia ao usuário.

Se o tipo de mídia tiver sido desabilitado:

- Um ícone de informações amarelo será exibido após o nome.
- *Desabilitado* será exibido na coluna Status.

Note:
As permissões do usuário para alterar os detalhes de suas próprias mídias podem ser concedidas/revogadas com base em sua função de usuário (consulte a opção *Criar e editar a própria mídia*). As permissões do superadministrador para alterar os detalhes das mídias de outros usuários também podem ser concedidas/revogadas com base em sua função de usuário (consulte a opção *Criar e editar mídias de usuários*).

Consulte a seção Tipos de mídia para obter detalhes sobre a configuração das mídias do usuário.

Permissões

A aba *Permissões* contém informações sobre os seguintes elementos:

- Função do usuário (obrigatória para qualquer usuário recém-criado) que só pode ser alterada por um usuário do tipo *Super admin*.

Attention:
Usuários não podem ser criados sem uma função de usuário (exceto com a User API do Zabbix). Usuários criados anteriormente que não possuem uma função ainda podem ser editados sem atribuir uma função a eles. No entanto, uma vez que uma função é atribuída, ela só pode ser alterada, não removida.

 Observe que usuários sem função podem fazer login no Zabbix apenas usando autenticação LDAP ou SAML, desde que suas informações LDAP/SAML correspondam aos mapeamentos de grupos de usuários configurados no Zabbix.

- Tipo de usuário (*User*, *Admin*, *Super admin*) que é definido na configuração da função do usuário.
- Grupos de hosts e templates aos quais o usuário tem acesso.
  - Usuários do tipo *User* e *Admin*, por padrão, não têm acesso a nenhum grupo, template ou host. Para conceder 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 que são negadas - em cinza claro.

Consulte a página Permissões para mais detalhes.

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, por padrão, não tem acesso a nenhum recurso. Quaisquer permissões para grupos de hosts ou templates devem ser atribuídas explicitamente.
- *Administrador* - tem direitos de acesso incompletos às seções do menu (veja abaixo). Por padrão, o usuário não tem acesso a nenhum grupo de hosts. Quaisquer permissões para grupos de hosts ou templates devem ser concedidas explicitamente.
- *Super administrador* - tem acesso a todas as seções do menu. O usuário tem acesso de leitura e gravação a todos os grupos de hosts e templates. As permissões não podem ser revogadas negando o acesso a grupos específicos.

Acesso ao menu

Note:
O acesso restrito a alguns elementos da interface do usuário apenas impede a abertura dessa página — isso não remove a possibilidade 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	Administrador	Super administrador
Dashboards	+	+	+
Monitoramento	+	+	+
Problemas	+	+	+

Seção do menu		Usuário	Administrador	Super administrador
	Hosts	+	+	+
	Dados mais recentes	+	+	+
	Mapas	+	+	+
	Descoberta		+	+
APM		+	+	+
	Traces	+	+	+
	Métricas	+	+	+
	Logs	+	+	+
Serviços		+	+	+
	Serviços	+	+	+
	SLA		+	+
	Relatório de SLA	+	+	+
Inventário		+	+	+
	Visão geral	+	+	+
	Hosts	+	+	+
Relatórios		+	+	+
	Informações do sistema		+	+
	Relatórios agendados		+	+
	Relatório de disponibilidade	+	+	+
	100 principais 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 de trigger		+	+
	Ações de serviço		+	+
	Ações de descoberta		+	+
	Ações de registro automático		+	+
	Ações internas		+	+
	Tipos de mídia			+
	Scripts			+
Usuários				+
	Grupos de usuários			+
	Funções de usuário			+
	Usuários			+
	Tokens de API			+
	Autenticação			+
	Dispositivos			+
Administração				+
	Fonte de dados			+
	Geral			+
	Log de auditoria			+
	Limpeza de dados			+
	Grupos de proxies			+
	Proxies			+
	Macros			+
	Fila			+

Funções de usuário

As funções de usuário permitem fazer ajustes personalizados nas permissões definidas pelo tipo de usuário.
Embora não seja possível adicionar permissões (isso excederia as permissões 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 da 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 na formulário de configuração de usuário, na aba *Permissões*, por usuários Super admin.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

Acesso a hosts

O acesso a quaisquer dados de host e template no Zabbix é concedido a grupos de usuários somente no nível de grupo de host/template.

Isso significa que um usuário individual não pode receber diretamente acesso a um host (ou grupo de hosts). O acesso a um host só pode ser concedido a um usuário por meio de sua participação em um grupo de usuários que tenha acesso ao grupo de hosts que contém o host.

Da mesma forma, o acesso a um template só pode ser concedido a um usuário por meio de sua participação em um grupo de usuários que tenha acesso ao grupo de templates que contém o template.

Grupos de usuários

Visão geral

Os grupos de usuários permitem agrupar usuários tanto para fins organizacionais quanto para atribuir 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 estão disponíveis para outro. Isso pode ser feito agrupando os usuários e atribuindo diferentes permissões aos grupos de hosts e templates.

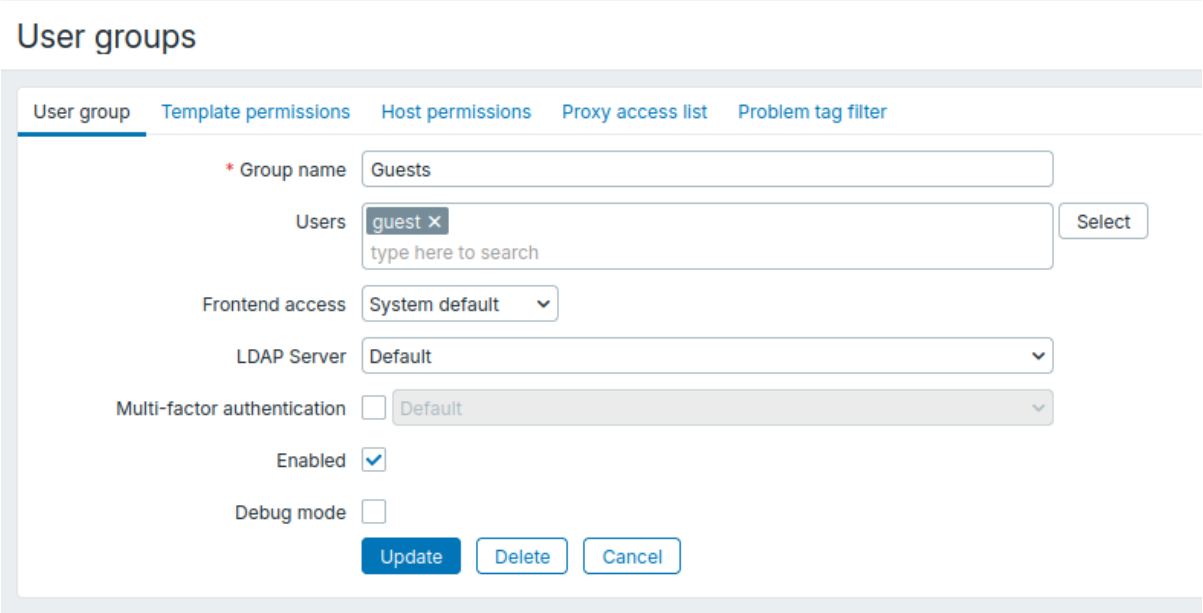
Um usuário pode pertencer a qualquer número de grupos.

Configuração

Para configurar um grupo de usuários:

1. Acesse *Usuários* > *Grupos de usuários*.
2. Clique em *Criar grupo de usuários* (ou no nome do grupo para editar um grupo existente).
3. Edite os atributos do grupo no formulário.

A aba **Grupo de usuários** contém os atributos gerais do grupo:

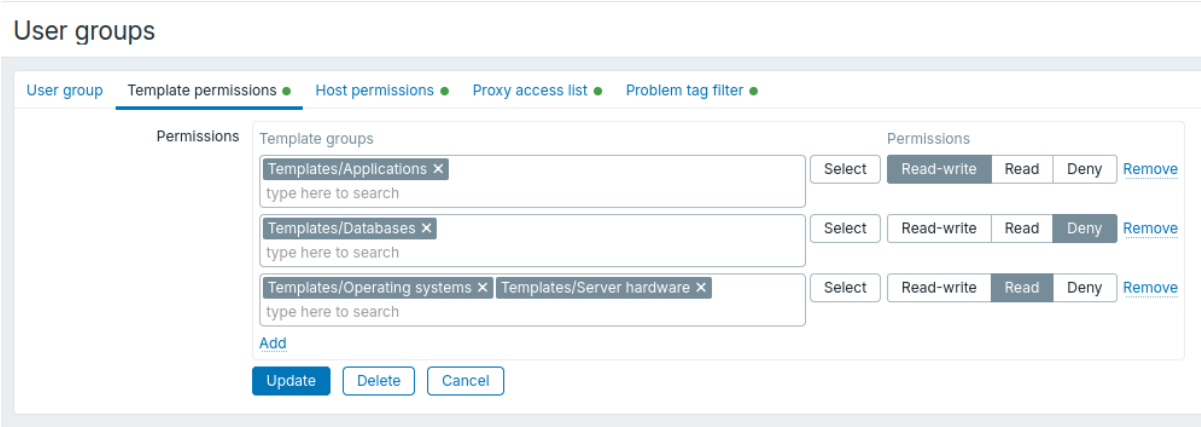


Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

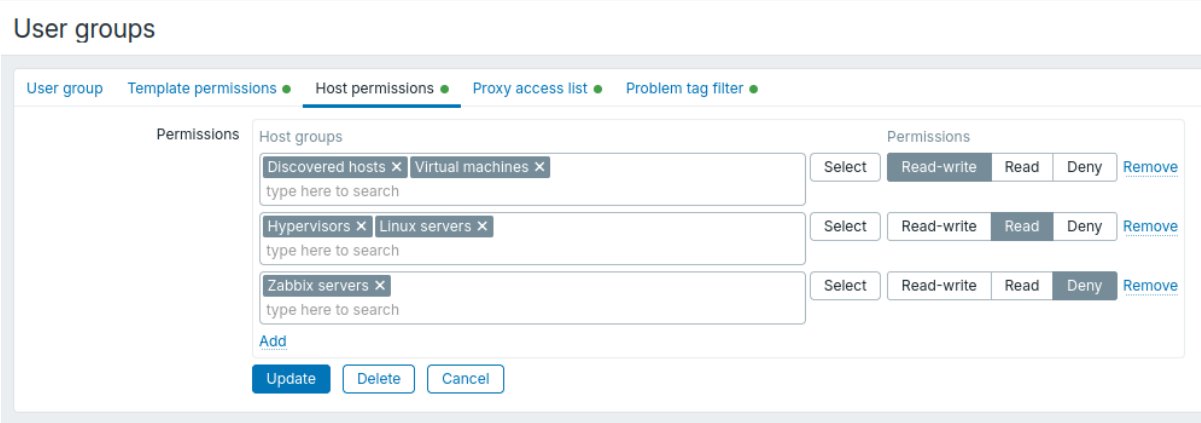
Parâmetro	Descrição
Nome do grupo	Nome exclusivo do grupo.
Usuários	Para adicionar usuários ao grupo, comece a digitar o nome de um usuário existente. Quando a lista suspensa com os nomes de usuários correspondentes aparecer, role para baixo para selecionar. Como alternativa, clique no botão *Selecionar* para selecionar usuários em uma janela pop-up.

Parâmetro	Descrição
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 a 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 fica habilitado somente quando *Acesso ao frontend* está 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 de MFA; esta opção é selecionada por padrão para novos grupos de usuários se a MFA estiver habilitada; **<Nome do método>** - usa o método selecionado (por exemplo, ”Zabbix TOTP”); **Desabilitado** - a MFA está desabilitada para este grupo; esta opção é selecionada por padrão para novos grupos de usuários se a MFA estiver desabilitada. Observe que, se um usuário pertencer a vários grupos de usuários com MFA habilitada (ou se pelo menos um grupo tiver MFA habilitada), as seguintes regras de autenticação serão aplicadas: se algum grupo usar o método de MFA ”Padrão”, ele autenticará o usuário; caso contrário, o primeiro método (em ordem alfabética) será usado para autenticação.
Habilitado	Status do grupo de usuários e de seus membros. *Marcado* - o grupo de usuários e os usuários estão habilitados *Desmarcado* - o grupo de usuários e os usuários estão desabilitados
Modo de depuração	Marque esta caixa de seleção para ativar o modo de depuração para os usuários.

A aba **Permissões de template** permite especificar o acesso do grupo de usuários aos grupos de templates e, por meio deles, aos dados dos templates:



A aba **Permissões de host** permite especificar o acesso do grupo de usuários aos grupos de hosts e, por meio deles, aos dados dos hosts:



Clique em Add para escolher os grupos de templates/hosts (sejam grupos pai ou aninhados) e atribuir permissões a eles. Comece a digitar o nome do grupo (uma lista suspensa com os grupos correspondentes será exibida) ou clique em *Selecionar* para abrir

uma janela pop-up com todos os grupos.

Em seguida, use os botões de opção para atribuir permissões aos grupos escolhidos. As permissões possíveis são:

- **Leitura e gravação** - acesso de leitura e gravação a um grupo
- **Leitura** - acesso somente leitura a um grupo
- **Negar** - acesso negado a um grupo

Se o mesmo grupo de template/host for adicionado em várias linhas com diferentes permissões definidas, a permissão mais restritiva será aplicada.

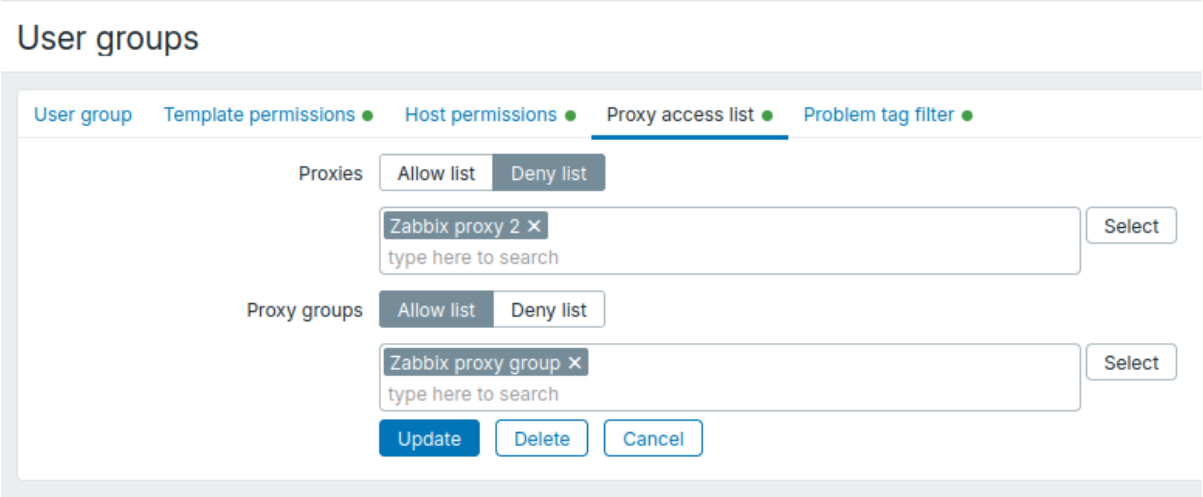
Observe que um usuário *Super administrador* pode impor que os grupos aninhados tenham o mesmo nível de permissões do grupo pai; isso pode ser feito no formulário de configuração do grupo de hosts/templates.

As abas **Permissões de template** e **Permissões de host** oferecem o mesmo conjunto de parâmetros.

As permissões atuais para os grupos são exibidas no bloco *Permissões* e podem ser modificadas ou removidas.

Note:
Se um grupo de usuários tiver permissões de **Leitura e gravação** para um host e permissões **Negar** ou nenhuma permissão para um template vinculado a esse host, os usuários desse grupo não poderão editar itens baseados em templates no host, e o nome do template será exibido como *Template inacessível*.

A aba **Lista de acesso a proxies** permite especificar o acesso do grupo de usuários aos proxies, individualmente ou por meio de grupos de proxies:



Use o campo *Grupos de proxies* para aplicar a permissão aos grupos selecionados. Um proxy que pertence a um grupo só pode ser permitido ou negado por meio desse grupo. Use o campo *Proxies* para adicionar proxies individuais que não pertencem a nenhum grupo de proxies.

Proxies e *Grupos de proxies* oferecem o mesmo conjunto de parâmetros.

Clique no botão *Selecionar* ao lado do campo *Proxies* ou *Grupos de proxies* para selecionar proxies ou grupos de proxies, respectivamente. Como alternativa, comece a digitar o nome de um proxy ou grupo de proxies no campo correspondente. Uma lista suspensa de proxies ou grupos de proxies correspondentes será exibida.

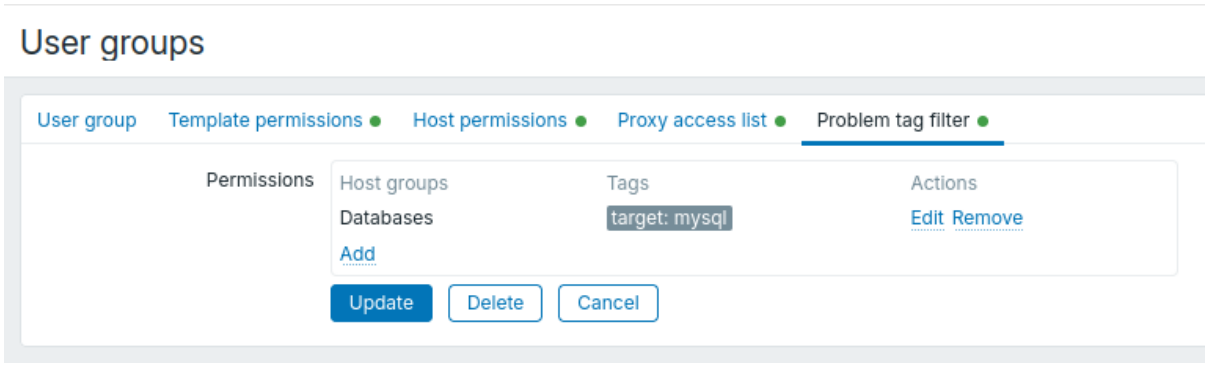
Selecione *Lista de permissões* para permitir que o grupo de usuários acesse somente os proxies e grupos de proxies selecionados.
 Se a seleção de *Proxies* estiver vazia para a *Lista de permissões*, o grupo de usuários não terá acesso a nenhum proxy que não pertença a um grupo de proxies. Se a seleção de *Grupos de proxies* estiver vazia para a *Lista de permissões*, o grupo de usuários não terá acesso a nenhum grupo de proxies.

Selecione *Lista de bloqueios* para negar ao grupo de usuários o acesso aos proxies e grupos de proxies selecionados. O acesso a todos os outros proxies e grupos de proxies será permitido.
 Se a seleção de *Proxies* estiver vazia para a *Lista de bloqueios*, o grupo de usuários terá acesso total a todos os proxies que não pertençam a um grupo de proxies. Se a seleção de *Grupos de proxies* estiver vazia para a *Lista de bloqueios*, o grupo de usuários terá acesso total a todos os grupos de proxies.

Regras de precedência de permissões para usuários que pertencem a vários grupos de usuários:

- A *Lista de bloqueios* substitui a *Lista de permissões*. Se qualquer grupo negar o acesso a um proxy/grupo de proxies, o usuário não poderá acessá-lo.
- As permissões da *Lista de permissões* são cumulativas. Se vários grupos permitirem o acesso, o usuário poderá acessar todos os proxies permitidos por esses grupos.

A aba **Filtro de tags de problemas** permite definir permissões baseadas em tags para que os grupos de usuários visualizem problemas filtrados por nome e valor da tag:

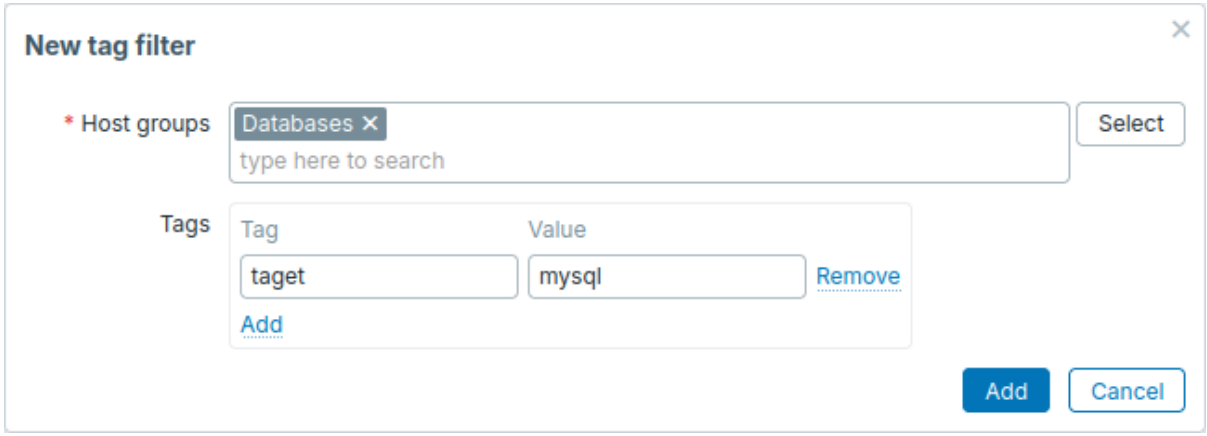


Clique em Add para escolher os grupos de hosts. Para selecionar um grupo de hosts ao qual aplicar um filtro de tags, clique em *Selecionar* para obter a lista completa dos grupos de hosts existentes ou comece a digitar o nome de um grupo de hosts para obter uma lista suspensa com os grupos correspondentes. Somente grupos de hosts serão exibidos, pois o filtro de tags de problemas não pode ser aplicado a grupos de templates.

É possível adicionar nomes de tags sem valores, mas não valores sem nomes. Somente as três primeiras tags (com valores, se houver) são exibidas no bloco *Permissões*; se houver mais, elas poderão ser visualizadas clicando ou passando o cursor sobre o ícone •••.

O filtro de tags permite separar o acesso ao grupo de hosts da possibilidade de visualizar problemas.

Por exemplo, se um administrador de banco de dados precisar visualizar somente problemas de bancos de dados "MySQL", primeiro será necessário criar 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 de tag "target" para o grupo de hosts selecionado.

Certifique-se de que o nome e o valor da tag estejam especificados corretamente; 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. Nesse caso, a filtragem 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 tags						
Grupo de hosts	*Nome da tag*	*Valor da tag*	*Grupo de hosts*	*Nome da tag*	*Valor da tag*	
Bancos de dados	target	mysql	Bancos de dados	target	oracle	problemas target:mysql ou target:oracle visíveis

Não configurado no **Filtro de tags de problemas**	Bancos de dados	target	oracle	problemas target:oracle visíveis

Acesso a partir de vários grupos de usuários

Um usuário pode pertencer a qualquer número de grupos de usuários. Esses grupos podem ter diferentes permissões de acesso a hosts ou templates.

Portanto, é importante saber a quais entidades um usuário sem privilégios poderá acessar como resultado. No exemplo a seguir, 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** ao '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 ao '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** ao '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 ao 'X' será **negado** ao usuário.

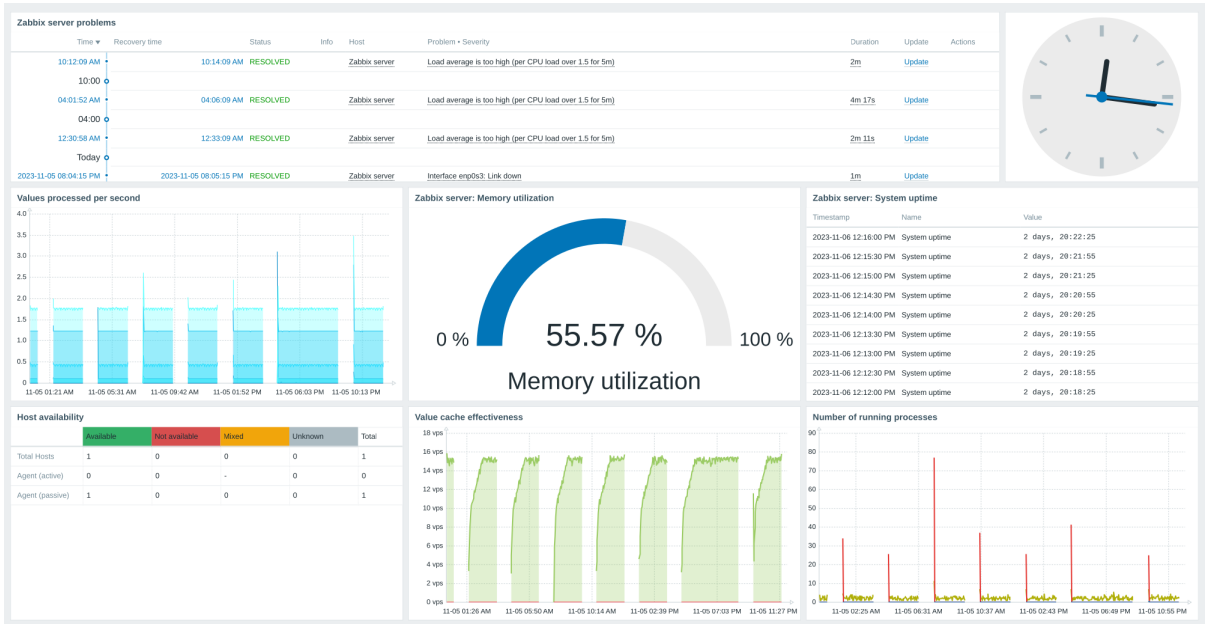
Outros detalhes

- Um usuário de nível Admin com acesso *Leitura-escrita* a um host não poderá vincular/desvincular templates, se ele não tiver acesso ao grupo de templates ao qual eles pertencem. Com acesso *Leitura* ao grupo de templates, ele poderá vincular/desvincular templates ao host, no entanto, não verá nenhum template na lista de templates e não poderá operar com templates em outros lugares.
- Um usuário de nível Admin com acesso *Leitura* a um host não verá o host na lista de hosts da seção de configuração; no entanto, os triggers do host estarão acessíveis na configuração do serviço 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".

Relatórios agendados

Visão geral

Com o recurso *Relatórios agendados*, você pode configurar uma versão em PDF de um determinado dashboard para ser enviada aos destinatários especificados em intervalos recorrentes.



Pré-requisitos:

- O Zabbix web service deve estar instalado e configurado corretamente para permitir a geração de relatórios agendados - consulte Configuração de relatórios agendados para obter 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 dos relatórios)
  - *Gerenciar relatórios agendados* no bloco *Acesso às ações* (para criar/editar relatórios)

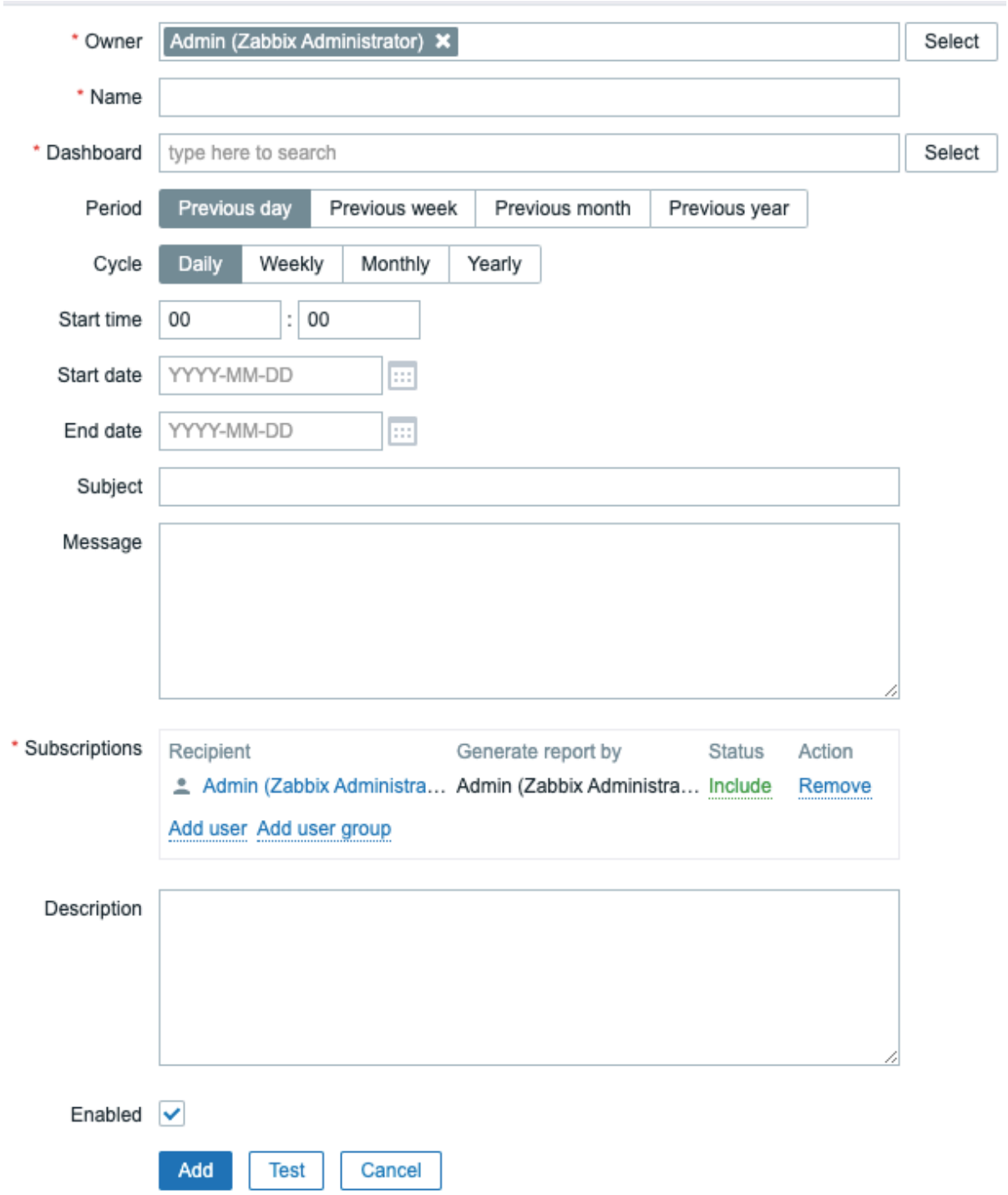
Para criar um relatório agendado no frontend do Zabbix, faça o seguinte:

1. Acesse *Relatórios* > *Relatórios agendados*.
2. Clique em *Criar relatório* no canto superior direito da tela.
3. Insira os parâmetros do relatório no formulário.

Como alternativa, você pode criar um relatório abrindo um relatório existente, clicando no botão *Clonar* e salvando-o com um nome diferente.

Configuração

A aba *Relatórios agendados* contém os atributos gerais do relatório.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Proprietário	Usuário que cria um 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 exclusivo.
Dashboard	Dashboard no qual o relatório se baseia; somente um dashboard pode ser selecionado por vez. Para selecionar um dashboard, comece a digitar o nome - uma lista de dashboards correspondentes será exibida; role para baixo para selecionar. Como alternativa, clique em *Selecionar* ao lado do campo e selecione um dashboard na lista exibida.
Período	Período para o qual o relatório será preparado. Selecione o dia, a semana, o mês ou o 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 o fuso horário do Zabbix server será usado.

Parâmetro	Descrição
Repetir em	Dias da semana em que o relatório será enviado. Este campo está disponível somente se *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 compatíveis: {TIME}, {TIMESTAMP}.
Mensagem	Corpo do e-mail do relatório. Macros compatíveis: {TIME}, {TIMESTAMP}.
Assinaturas	Lista de destinatários do relatório. Por padrão, inclui somente o proprietário do relatório. Qualquer usuário do Zabbix com uma 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* - define se os dados do relatório devem ser gerados com base nas permissões do dashboard do usuário atual ou do destinatário. *Status* - selecione ”Incluir” para enviar o relatório ao usuário ou ”Excluir” para impedir o envio do relatório a esse usuário. Pelo menos um usuário deve ter o status ”Incluir”. O status ”Excluir” pode ser usado para excluir usuários específicos de um grupo de usuários 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 o 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 de seleção 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 ao usuário atual.
Delete	Excluir o relatório.
Cancel	Cancelar a edição das propriedades do relatório.

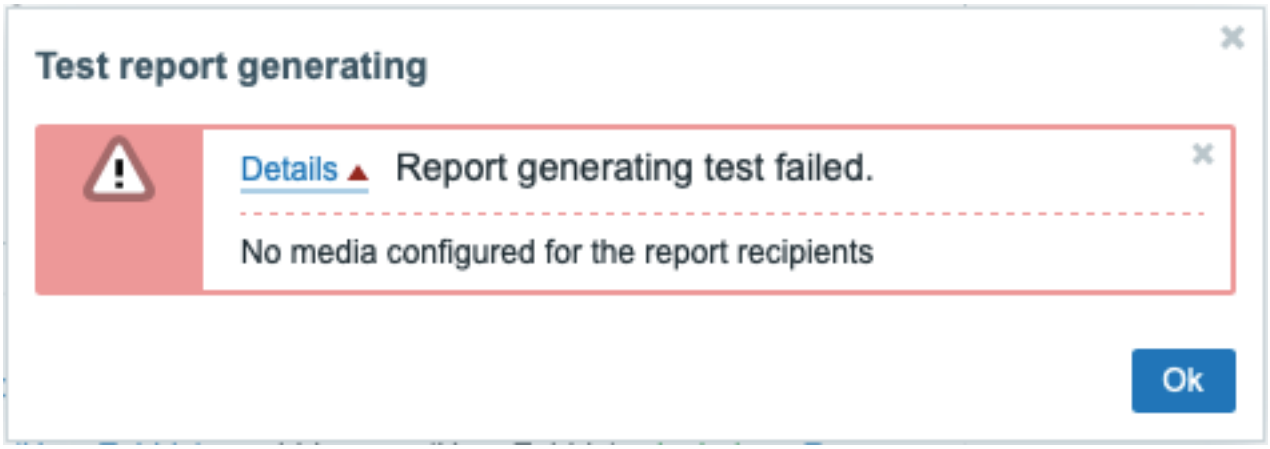
Teste

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 tiver sido aberto no menu de ações do dashboard.

Se a configuração estiver correta, o relatório de teste será enviado imediatamente ao usuário atual. Para relatórios de teste, as configurações de assinantes e do usuário *Gerar relatório por* são ignoradas.

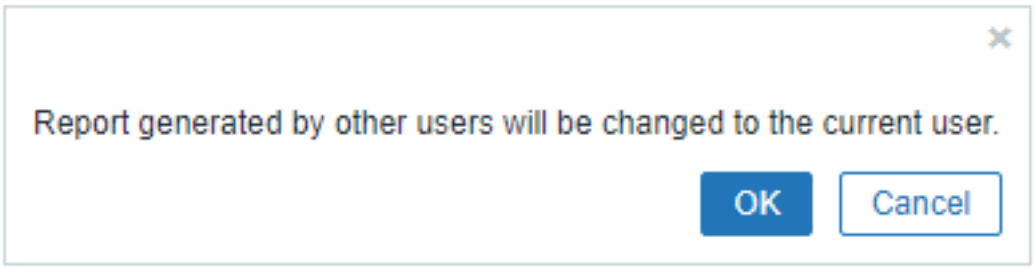
Se a configuração estiver incorreta, será exibida uma mensagem de erro descrevendo a possível causa.



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*.

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)

Exportar para arquivos

Visão geral

É possível configurar a exportação em tempo real de eventos de trigger, valores de item e tendências em um formato JSON delimitado por nova linha.

A exportação é feita em arquivos, em que cada linha do arquivo de exportação é um objeto JSON. Os 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 a renomeação), o item de dados é descartado e nunca é gravado no arquivo de exportação. Ele é gravado apenas no banco de dados do Zabbix. A gravação de dados no arquivo de exportação é retomada quando o problema de gravação for resolvido.

O arquivo de exportação é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo pode ser lido pelo grupo do proprietário. Todas as outras permissões são negadas.

Observe que host/item pode não ter metadados (grupos de host, nome do host, nome do item) se o host/item tiver sido removido após os dados terem sido recebidos, mas antes de o 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 - consulte o parâmetro ExportDir na configuração do server.

Há outros dois parâmetros 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 o tamanho exceder o limite configurado, o arquivo será renomeado adicionando `.old` ao nome, e um novo arquivo com o nome original será criado.

Attention:
Um arquivo será criado para cada processo que gravará dados (ou seja, aproximadamente de 4 a 30 arquivos). Como o tamanho padrão de cada arquivo de exportação é 1G, manter arquivos de exportação grandes pode consumir rapidamente o espaço em disco.

- `ExportType` permite especificar quais tipos de entidade (eventos, histórico, tendências) serão exportados.

Protocolo de exportação JSON delimitado por nova linha

Esta seção apresenta detalhes do protocolo de exportação em um formato JSON delimitado por nova linha, usado em:

- exportação de dados para arquivos
- streaming para sistemas externos

Todos os arquivos têm a extensão `.ndjson`. Cada linha do arquivo de exportação é um objeto JSON.

Exportação de evento de trigger

As seguintes informações são exportadas para um evento de problema:

Field	Type	Description
clock	number	Número de segundos desde a Epoch até o momento em que o problema foi detectado (parte inteira).
ns	number	Número de nanossegundos a serem adicionados a `clock` para obter o momento exato da 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.

Field		Type	Description
-		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 do 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:

Field	Type	Description
clock	number	Número de segundos desde a Epoch até o momento em que o problema foi resolvido (parte inteira).
ns	number	Número de nanossegundos a serem adicionados a `clock` para obter o momento exato da resolução do problema.
value	number	0 (sempre).
eventid	number	ID do evento de recuperação.
p_eventid	number	ID do evento de problema.

Exemplos de exportação de eventos de trigger

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}
```

Exportação de valor de item

As seguintes informações são exportadas para um valor de item coletado:

Field		Type	Description
host		object	Nome do host do item.
	host	string	Nome do host.
	name	string	Nome visível do host.
groups		array	Lista de grupos de host do host do item; deve haver pelo menos um elemento no array.
	-	string	Nome do grupo de host.
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.

Field	Type	Description
clock	number	Número de segundos desde a Epoch até o momento em que o valor foi coletado (parte inteira).
ns	number	Número de nanossegundos a serem adicionados a `clock` para obter um horário preciso de coleta do valor.
timestamp (*Log* only)	number	0 se não estiver disponível.
source (*Log* only)	string	String vazia se não estiver disponível.
severity (*Log* only)	number	0 se não estiver disponível.
eventid (*Log* only)	number	0 se não estiver disponível.
value	number (for numeric items) or string (for text items)	Valor do item coletado.
type	number	Tipo do valor coletado: 0 - float numérico, 1 - caractere, 2 - log, 3 - inteiro sem sinal, 4 - texto, 5 - binário, 6 - JSON

Exemplo de exportação de valor de item

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":[{"
```

Exportação de tendências

As seguintes informações são exportadas para um valor de tendência calculado:

Field	Type	Description
host	object	Nome do host do host do item.
host	string	Nome do host.
name	string	Nome visível do host.
groups	array	Lista de grupos de host do host do item; deve haver pelo menos um elemento no array.
-	string	Nome do grupo de host.
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 Epoch até o momento em que o valor foi coletado (parte inteira).
count	number	Número de valores coletados para uma determinada hora.
min	number	Valor mínimo do item para uma determinada hora.
avg	number	Valor médio do item para uma determinada hora.
max	number	Valor máximo do item para uma determinada hora.

Field	Type	Description
type	number	Tipo de valor: 0 - float numérico, 3 - inteiro sem sinal

Exemplos de exportação de trends

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":[{"
```


Streaming para sistemas externos

Visão geral

É possível transmitir valores de item e eventos do Zabbix para sistemas externos por HTTP (consulte os detalhes do protocolo).

O filtro de tags pode ser usado para transmitir subconjuntos de valores de item ou eventos.

Dois tipos de processos do Zabbix server são responsáveis pela transmissão de dados: `connector manager` e `connector worker`. Um item interno do Zabbix, `zabbix[connector_queue]`, permite monitorar a quantidade de valores enfileirados na fila do conector.

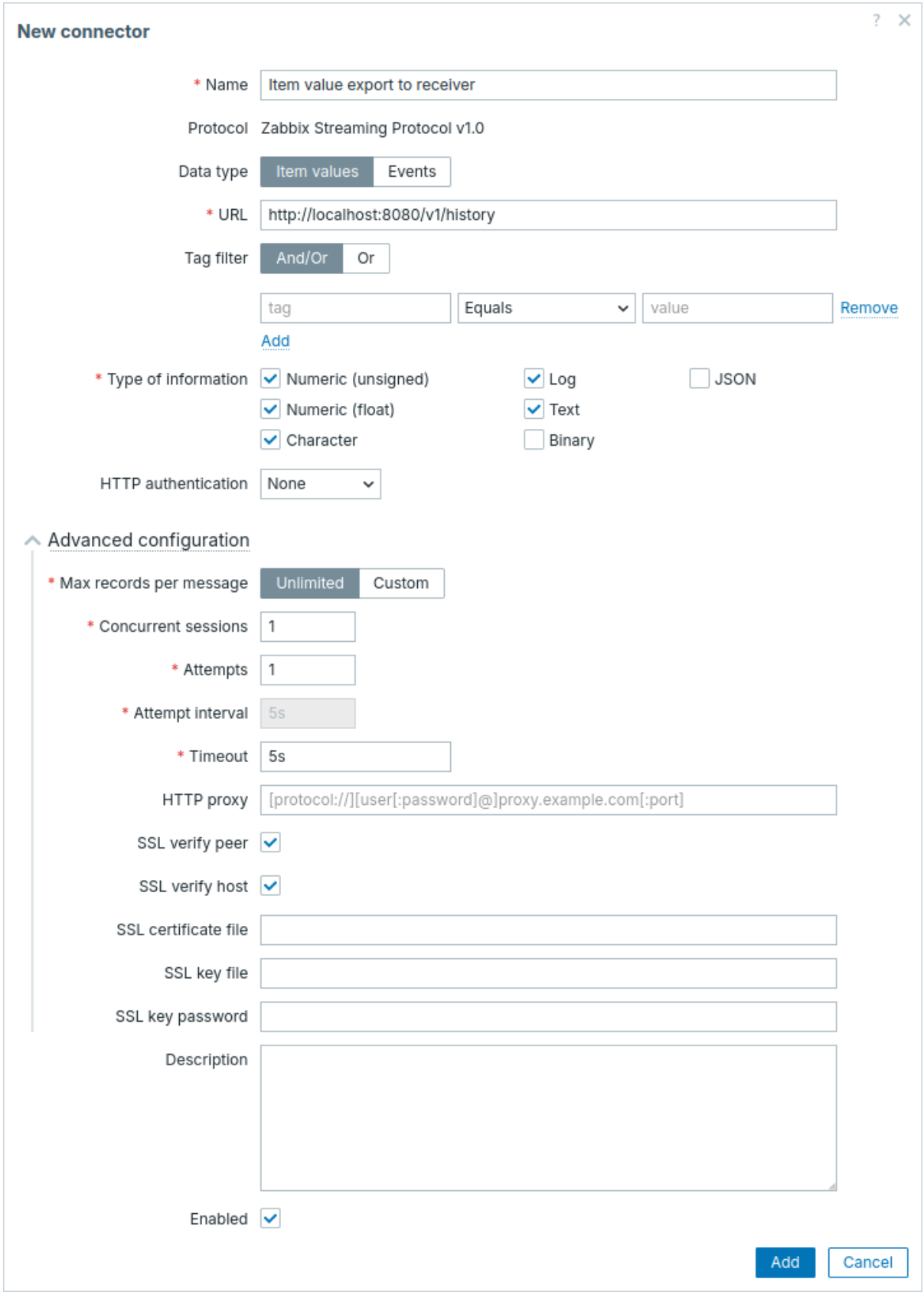
Configuração

As etapas a seguir são necessárias para configurar o streaming de dados para um sistema externo:

1. Tenha um sistema remoto configurado para receber dados do Zabbix. Para isso, estão disponíveis as seguintes ferramentas:
   - Um exemplo de um receiver simples que registra as informações recebidas nos arquivos `events.ndjson` e `history.ndjson`.
   - Kafka connector for Zabbix server - um server leve escrito em Go, projetado para encaminhar valores de item e eventos de um server Zabbix para um broker Kafka.

2. Defina o número necessário de workers de connector no Zabbix ajustando o parâmetro `StartConnectors` em `zabbix_server.conf`. O número de workers de connector deve corresponder ao número de connectors configurado no frontend do Zabbix (ou ser maior, se o número de sessões simultâneas for superior a 1). Em seguida, reinicie o server Zabbix.

3. Configure um novo connector no frontend do Zabbix (*Administração* > *Geral* > *Connectors*) e recarregue o cache do server com o comando `zabbix_server -R config_cache_reload`.



Os campos obrigatórios são marcados com um asterisco.

Parâmetro	Descrição
Nome	Insira o nome do connector.
Tipo de dados	Selecione o tipo de dados a transmitir: **Valores de item** - transmite valores de item do Zabbix para sistemas externos; **Eventos** - transmite eventos do Zabbix para sistemas externos.
URL	Insira a URL do receiver. Macros de usuário são compatíveis.

Parâmetro	Descrição
Filtro de tags	Exporte somente valores de item ou eventos que correspondam ao filtro de tags. Se não for definido, tudo será exportado. É possível incluir e excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência de nomes de tags sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **É igual a** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados nos quais os valores de tags contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual a** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados nos quais os valores de tags contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as 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 (ponto flutuante), caractere etc.) pelo qual filtrar os valores de item que o connector deve transmitir. Este campo está disponível quando *Tipo de dados* está definido como ”Valores de item”.
Autenticação HTTP	Selecione a opção de autenticação: **Nenhuma** - nenhuma autenticação é usada; **Básica** - é usada autenticação básica; **NTLM** - é usada autenticação NTLM (Windows NT LAN Manager); **Kerberos** - é usada autenticação Kerberos (consulte também: Configurando o Kerberos com o Zabbix); **Digest** - é usada autenticação Digest; **Bearer** - é usada autenticação Bearer.
Nome de usuário	Insira o nome de usuário (até 255 caracteres). Macros de usuário são compatíveis. Este campo está disponível quando *Autenticação HTTP* está definida como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Senha	Insira a senha do usuário (até 255 caracteres). Macros de usuário são compatíveis. Este campo está disponível quando *Autenticação HTTP* está definida como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Token Bearer	Insira o token Bearer. Macros de usuário são compatíveis. Este campo está disponível e é obrigatório quando *Autenticação HTTP* está definida como ”Bearer”.
Configuração avançada	Clique no cabeçalho *Configuração avançada* para exibir as opções de configuração avançada (veja abaixo).
Máximo de registros por mensagem	Especifique o número máximo de valores ou eventos que podem ser transmitidos em uma mensagem.
Sessões simultâneas	Selecione o número de processos de envio a serem executados para este connector. É possível especificar até 100 sessões; o valor padrão é ”1”.
Tentativas	Número de tentativas para transmitir dados. É possível especificar até 5 tentativas; o valor padrão é ”1”.
Intervalo entre tentativas	Especifique quanto tempo o connector deve aguardar após uma tentativa malsucedida de transmitir dados. É possível especificar até 10s; o valor padrão é ”5s”. Este campo está disponível quando *Tentativas* está definido como ”2” ou mais. Tentativas malsucedidas são aquelas em que não foi possível estabelecer uma conexão ou em que o código de resposta HTTP não é 200, 201, 202, 203 ou 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 ou 422. Os redirecionamentos são seguidos; portanto, 302 -> 200 é uma resposta positiva, enquanto 302 -> 503 acionará uma nova tentativa.
Tempo limite	Especifique o tempo limite da mensagem (1-60 segundos, padrão - 5 segundos). Sufixos de tempo são compatíveis (por exemplo, `30s`, `1m`). Macros de usuário são compatíveis.

Parâmetro	Descrição
Proxy HTTP	Você pode especificar um proxy HTTP a ser usado no seguinte formato: `[protocol://][username[:password]@]proxy.example.com[:port]` Macros de usuário são compatíveis. 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). Quando nenhum protocolo é especificado, o proxy será tratado como um proxy HTTP. Por padrão, a porta 1080 será usada. Se *Proxy HTTP* for especificado, o proxy substituirá variáveis de ambiente relacionadas a proxy, como `http_proxy` e `HTTPS_PROXY`. Se não for especificado, o proxy não substituirá as variáveis de ambiente relacionadas a proxy. O valor inserido é transmitido como está, sem qualquer verificação de validade. Você também pode inserir um endereço de proxy SOCKS. Se especificar o protocolo incorreto, o connector não conseguirá transmitir valores de item ou eventos do Zabbix. Observe que somente a autenticação simples é compatível com proxy HTTP.
Verificar peer SSL	Marque a caixa de seleção para verificar o certificado SSL do web server. O certificado do server é verificado em relação aos certificados da autoridade de certificação (CA) confiável no local de CA de todo o sistema. Você pode substituir o local dos arquivos de CA usando o parâmetro de configuração do server ou proxy Zabbix `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 web server 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 compatíveis. 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 server ou proxy Zabbix `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 compatíveis. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do server ou proxy Zabbix `SSLKeyLocation`.
Senha da chave SSL	Senha do arquivo de chave privada SSL. Macros de usuário são compatíveis.
Descrição	Insira a descrição do connector.
Habilitado	Marque a caixa de seleção para habilitar o connector.

Quando o Kafka connector é configurado com uma lista de endereços de brokers de bootstrap separados por vírgulas (por exemplo, `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), o cliente Kafka se conecta ao(s) broker(s) que respondem primeiro e usa os metadados do cluster. Se a lista contiver endereços de diferentes clusters Kafka, somente o cluster que responder mais rapidamente será usado e os outros endereços serão registrados como indisponíveis; como resultado, avisos de inicialização como o seguinte podem aparecer mesmo quando o connector está conectado:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

Em alguns ambientes (redes privadas, redes de containers ou configurações não padrão de DNS/hosts), nomes de host ou IPs podem ser resolvidos para endereços de loopback (por exemplo, `127.0.0.1/localhost`) ou normalizados pelo cliente, o que pode tornar esses avisos enganosos. Para reduzir a confusão, certifique-se de que todos os endereços `Kafka.URL` pertençam ao mesmo cluster Kafka, verifique a resolução DNS a partir do host do connector e os `advertised.listeners` dos brokers, e prefira endereços que sejam resolvidos para o endereço anunciado pelo broker.

Protocolo

A comunicação entre o server e o receiver é feita via HTTP usando a REST API, NDJSON, "Content-Type: application/x-ndjson".

Para mais detalhes, consulte Newline-delimited JSON export protocol.

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 receiver

A resposta consiste no código de status da resposta HTTP e na carga JSON. O código de status da resposta HTTP deve ser ”200”, ”201”, ”202”, ”203” ou ”204” para requisições processadas com sucesso; qualquer outro indica falha na requisição.

Exemplo de sucesso:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Exemplo com erros:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

gateway SNMP

Visão geral

O Zabbix SNMP gateway é uma extensão AgentX para o snmpd que suporta polling e trapping SNMP.

Com o Zabbix SNMP gateway é possível usar o protocolo SNMP para recuperar:

- dados de trigger;
- dados de trigger de problema;
- status do grupo de hosts (contagem de triggers por estado 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 do grupo de hosts.

O **sufixo** do OID é definido na configuração dos triggers do host como uma **tag** (por exemplo, `OIDSuffix:3`) no frontend.

Neste caso, todas as informações para o trigger estarão disponíveis em `OID=1.3.6.1.4.1.3043.7.55.X.3`. "X" aqui será o número 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 obter instruções sobre:

- instalar e configurar o snmpd;
- habilitar o suporte ao AgentX;
- configurar o Zabbix SNMP gateway;
- configurar traps SNMP para alterações de estado de trigger.

Recuperando dados

Com tudo configurado corretamente, você pode usar os comandos `snmpwalk` e `snmpget` para recuperar dados:

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

Opções de filtragem

Você pode limitar as informações de trigger de problema na configuração do SNMP gateway:

- por severidade (por padrão `ProblemMinSeverity=-1`)
- ocultando problemas reconhecidos (por padrão `ProblemHideAck=false`)

Você pode limitar a contagem de problemas por grupo de host na configuração do SNMP gateway:

- por triggers em estado desconhecido (por padrão `CountUnknown=false`)
- por triggers com problemas reconhecidos/não reconhecidos/todos (por padrão `CountAcknowledgeStatus=all`)

Exportação/importação de configuração

Visão geral

A funcionalidade de exportação/importação do Zabbix torna possível trocar 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
- enviar um template para Zabbix Community templates. Então, outros podem baixar o template e importar o arquivo para o Zabbix.
- integrar com ferramentas de terceiros - os formatos universais YAML, XML e JSON tornam possível a integração e a importação/exportação de dados com ferramentas e aplicativos de terceiros

O que pode ser exportado/importado

Os objetos que podem ser exportados/importados são:

- Host groups (*somente por meio da Zabbix API*)
- Template groups (*somente por meio da Zabbix API*)
- Templates

- Hosts
- Network maps
- Media types
- Imagens
- Dashboards
- Regular expressions

Formato de exportação

Os dados podem ser exportados usando o web frontend do Zabbix ou a Zabbix API. Os formatos de exportação suportados são YAML, XML e JSON.

Detalhes sobre a exportação

- Todos os elementos suportados são exportados em um único arquivo.
- As entidades de host e template (items, triggers, gráficos, regras de descoberta) herdadas de templates vinculados não são exportadas. Quaisquer alterações feitas nessas entidades no nível do host (como alteração do intervalo do item, modificação da expressão regular ou adição de protótipos à regra de descoberta de baixo nível) serão perdidas ao exportar; ao importar, todas as entidades dos templates vinculados são recriadas como no template vinculado original.
- As entidades criadas por descoberta de baixo nível e quaisquer entidades que dependam 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 contiver 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 da imagem por meio da importação.
- Ao importar hosts/templates usando a opção ”Delete missing”, macros de host/template que não estiverem presentes no arquivo de importação serão excluídas do host/template após a importação.
- Tags vazias para items, triggers, graphs, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes não têm significado, ou seja, é o mesmo que se estivessem ausentes.
- Se as entidades do host/template importado tiverem seus próprios timeouts configurados, eles serão aplicados; caso contrário, os timeouts do proxy/global serão aplicados.
- A importação suporta YAML, XML e JSON; o arquivo de importação deve ter uma extensão correta: .yaml e .yml para YAML, .xml para XML e .json para JSON. Consulte compatibility information sobre as versões de XML suportadas.
- A importação suporta arquivos de configuração somente em codificação UTF-8 (com ou sem BOM); outras codificações (UTF16LE, UTF16BE, UTF32LE, UTF32BE, etc.) resultarão em um erro de conversão na importação.

Formato base YAML

O formato de exportação YAML contém os seguintes nós:

- Nó raiz para exportação YAML do Zabbix
- Versão de 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.

Grupos de template

Visão geral

No frontend, grupos de template podem ser exportados apenas com a exportação de template. Quando um template é exportado, todos os grupos aos quais ele pertence são exportados automaticamente junto com ele.

A API permite exportar grupos de template independentemente dos templates.

Formato de exportação

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Network devices'
```

Elementos exportados

Element	Type	Description
uuid	string	Identificador exclusivo para este grupo de template.
name	string	Nome do grupo.

Grupos de host

Visão geral

No frontend, os grupos de host podem ser exportados somente com a exportação de host. Quando um host é exportado, todos os grupos aos quais ele pertence são exportados automaticamente junto com ele.

A API permite exportar grupos de host independentemente dos hosts.

Formato de exportação

```
   host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix servers'
```

Elementos exportados

Element	Type	Description
uuid	string	Identificador único para este grupo de hosts.
name	string	Nome do grupo.

Templates

Visão geral

Templates são exportados com muitos objetos relacionados e relações entre objetos.

A exportação de template contém:

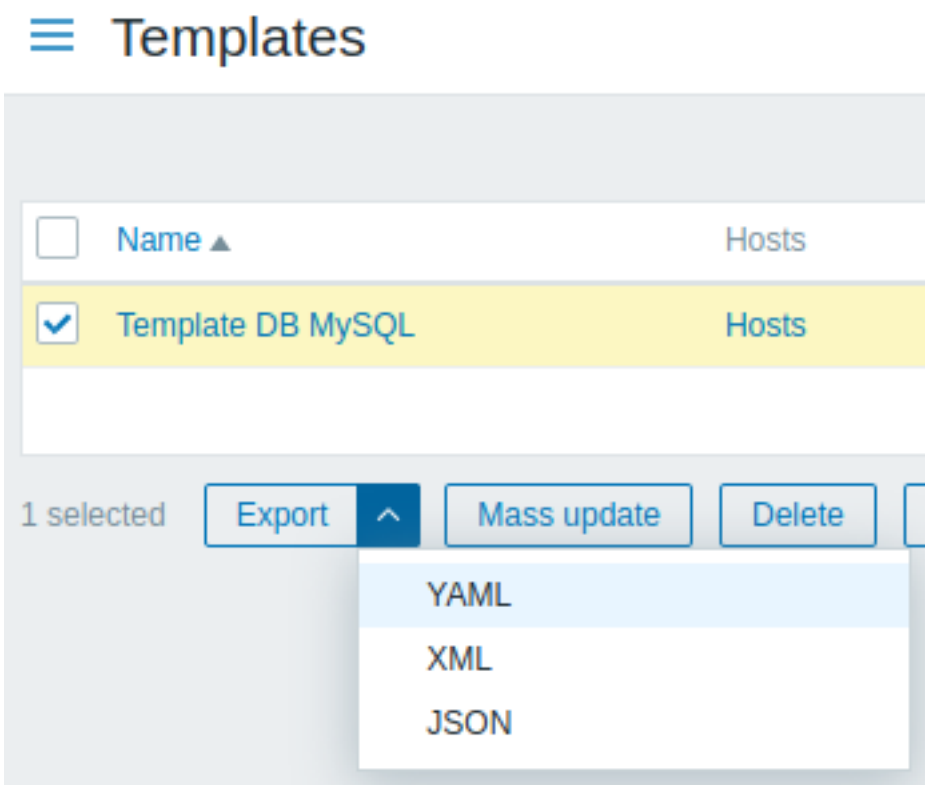
- Grupos de template vinculados
- Grupos de host vinculados (se usados na configuração de host prototype)
- Dados do template
- Vinculação a outros templates
- Vinculação a grupos de template

- items vinculados diretamente
- triggers vinculados diretamente
- gráficos vinculados diretamente
- dashboards vinculados diretamente
- regras de descoberta vinculadas diretamente com todos os prototypes
- cenários web vinculados diretamente
- mapas de valor

Exportando

Para exportar templates, faça o seguinte:

1. Acesse *Data collection* > *Templates*.
2. Marque as caixas de seleção dos templates a serem exportados.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



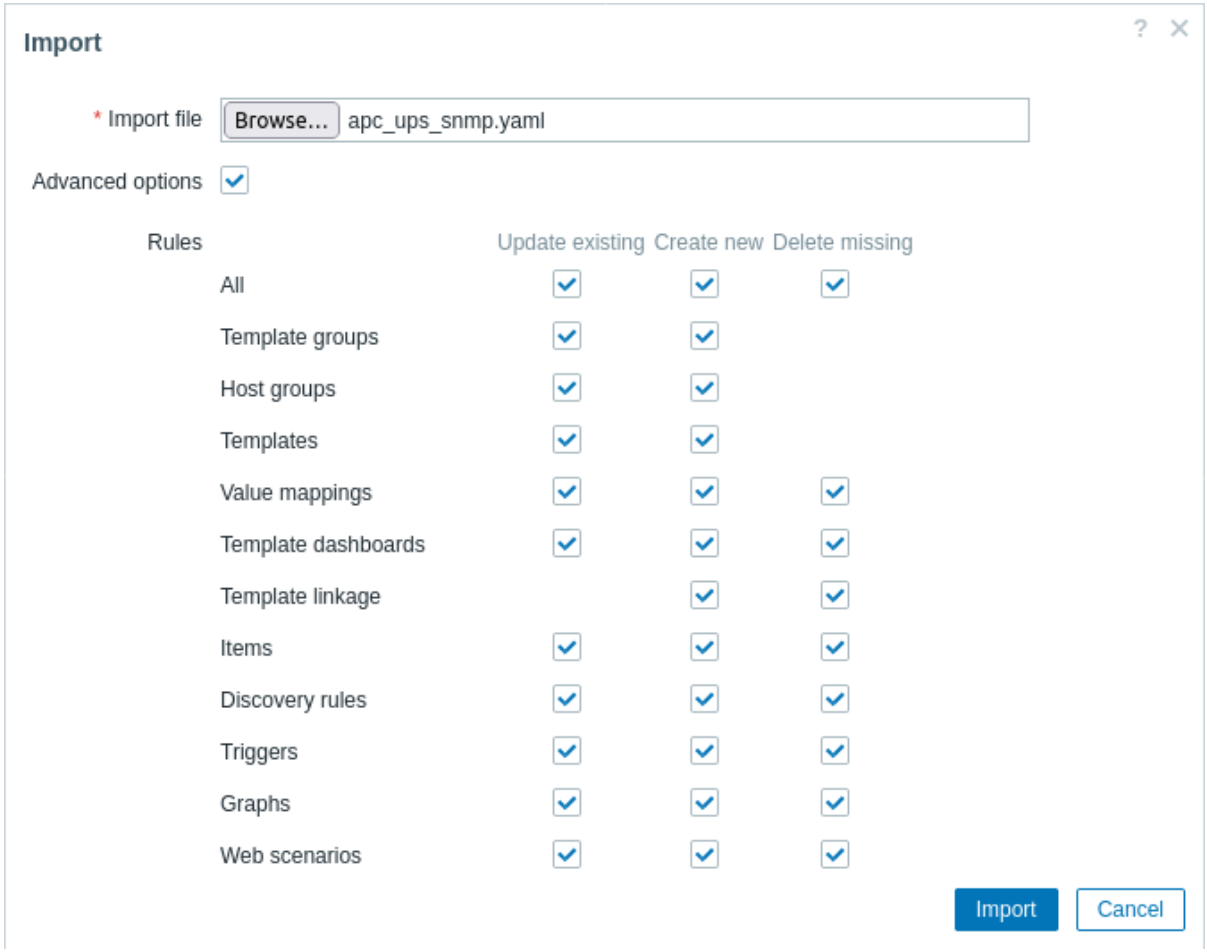
Dependendo do formato selecionado, os templates são exportados para um arquivo local com um nome padrão:

- `zabbix_export_templates.yaml` - na exportação YAML;
- `zabbix_export_templates.xml` - na exportação XML;
- `zabbix_export_templates.json` - na exportação JSON.

Importação

Para importar templates, faça o seguinte:

1. Acesse *Coleta de dados* > *Templates*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (veja abaixo).
5. Clique em *Importar* no canto inferior direito do formulário de configuração.



Se você marcar a caixa de seleção *Opções avançadas*, será exibida uma lista detalhada de todos os elementos que podem ser importados. Marque ou desmarque cada regra de importação conforme necessário.

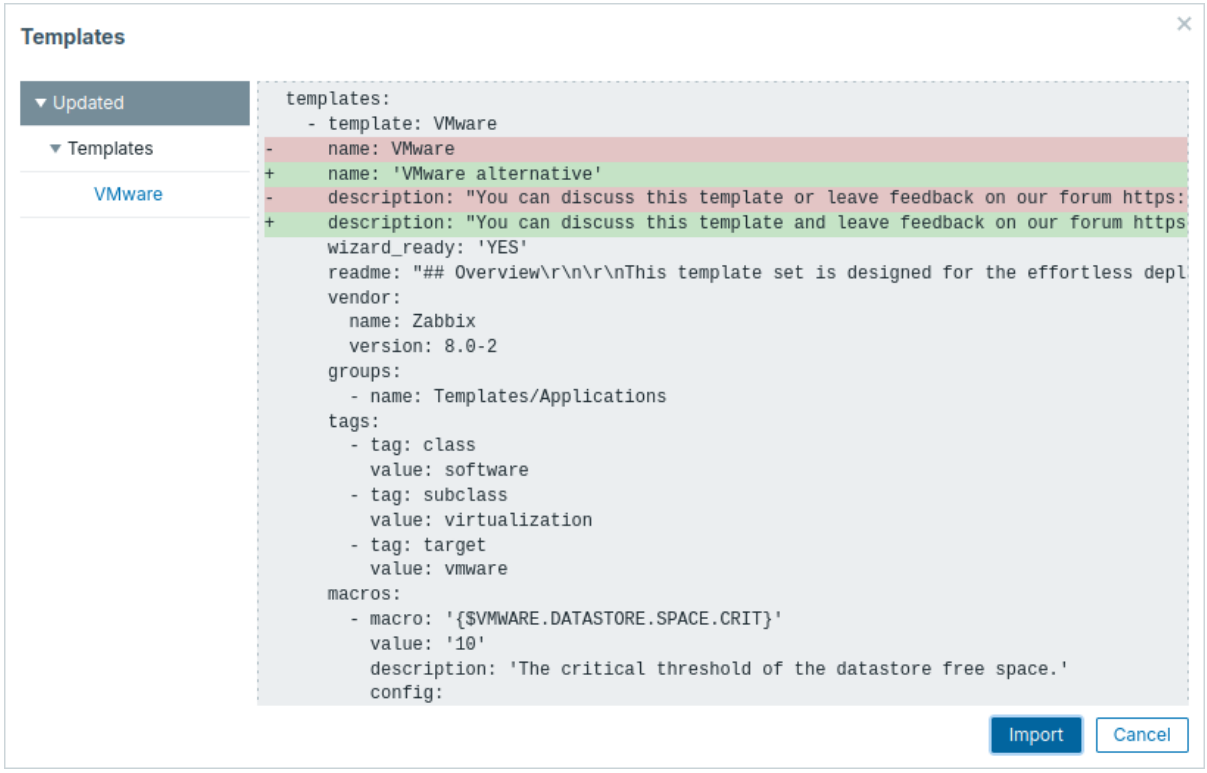
Se você clicar na caixa de seleção da linha *Todos*, todos os elementos abaixo dela serão marcados ou desmarcados.

Regras de importação:

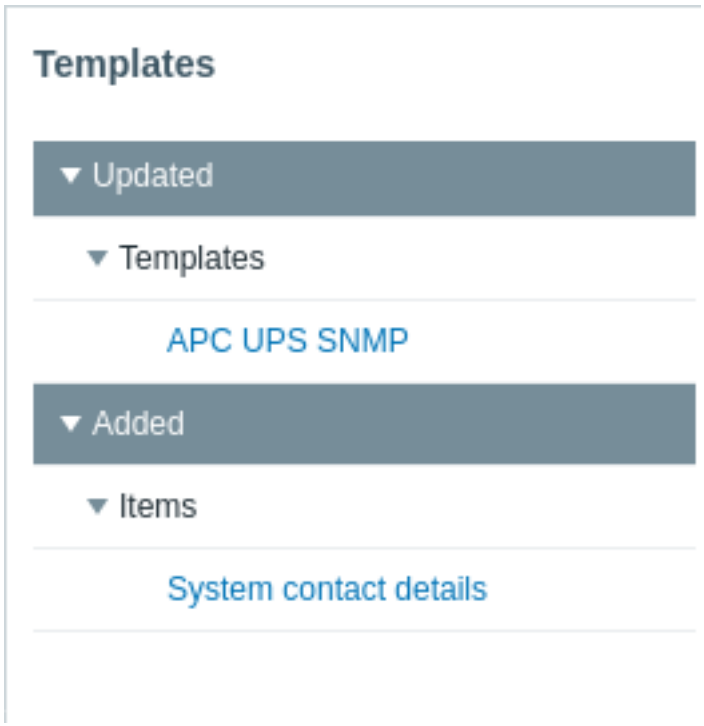
Regra	Descrição
Atualizar existentes	Os elementos existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Criar novos	Novos elementos serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.
Excluir ausentes	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*, as vinculações atuais de templates que não estiverem presentes no arquivo de importação serão desvinculadas. As entidades (itens, triggers, gráficos etc.) herdadas dos templates desvinculados não serão removidas (a menos que a opção *Excluir ausentes* também esteja selecionada para cada entidade).

Attention:
Se já existirem templates com os mesmos nomes, as opções *Excluir ausentes* deverão ser marcadas durante a importação para obter uma importação limpa. Dessa forma, os itens antigos que não estiverem mais no template atualizado serão removidos (observe que isso implicará a perda do histórico desses itens antigos).

Na tela seguinte, você poderá visualizar o conteúdo de um template que está sendo importado. Se for um novo template, todos os elementos serão listados em verde. Se um template existente estiver sendo atualizado, 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 sobre um fundo cinza.



O menu à esquerda pode ser usado para navegar pela lista de alterações. A seção *Atualizados* destaca todas as alterações feitas nos elementos existentes do template. A seção *Adicionados* lista os novos elementos do template. Os elementos de 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 outros.

Cada elemento tem uma finalidade específica 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 reticência (`...`) indica elementos omitidos por brevidade. A observação (`see table below`) é usada quando o elemento é explicado com mais detalhes em seções posteriores.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

Metadados do template

Element	Type	Description
version	string	(obrigatório) Versão da exportação.
template_groups		(obrigatório) Elemento raiz para grupos de template.
uuid	string	(obrigatório) Identificador exclusivo para este grupo de template.
name	string	(obrigatório) Nome do grupo de template.
host_groups		Elemento raiz para grupos de host usados por protótipos de host.
uuid	string	(obrigatório) Identificador exclusivo 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 do template, que são os mesmos que para template item triggers.
graphs		Elemento raiz para elementos de gráfico do template, que são os mesmos que para host graphs.

Templates

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Este é um template oficial do Linux. Requer Zabbix agent 8.0 ou mais recente. (...)
      wizard_ready: 'YES'
      readme: |
        ## Visão geral

        Este é um template oficial do Linux. Requer Zabbix agent 8.0 ou mais recente. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Sistemas operacionais'
      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)
  (...)
```

Element	Type	Description
uuid	string	(obrigatório) Identificador exclusivo deste template.
template	string	(obrigatório) Nome exclusivo do template.
name	string	Nome visível do template.
description	text	Descrição do template.
wizard_ready	text	Indica se o template está disponível para seleção no Host Wizard. Valores possíveis:[1] NO (0, padrão), YES (1). Veja também: Template object (`wizard_ready`).
readme	text	Instruções de configuração específicas do template para exibição no Host Wizard. 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 da 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 template items.
discovery_rules		Elemento raiz para template low-level discovery rules.
httptests		Elemento raiz para template web scenarios.
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: User macro object (`type`).
value	string	Valor da macro de usuário.
description	string	Descrição da macro de usuário.
config		Elemento raiz para template macro configuration responsável por como a macro é exibida no Host Wizard.
dashboards		Elemento raiz para template dashboards.
valuemaps		Elemento raiz para template value maps.

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: 'Status do ping do Zabbix agent'
          tags:
            - tag: component
              value: system
        (...)
```

```yaml
      - uuid: 58818005e76d46dda14d6592f601ab00
        name: 'Número de pacotes instalados'
        key: system.sw.packages.get
        delay: 1h
        preprocessing: (consulte a tabela abaixo)
        tags:
          - tag: component
            value: os
        triggers: (consulte a tabela abaixo)
      - uuid: 403cebed115441369e94d35d070ca7b8
        name: 'Utilização de memória'
        type: DEPENDENT
        key: vm.memory.utilization
        value_type: FLOAT
        units: '%'
        description: 'A porcentagem de memória utilizada é calculada como `100-pavailable`.'
        preprocessing: (consulte a tabela abaixo)
        master_item:
          key: 'vm.memory.size[pavailable]'
        tags:
          - tag: component
            value: memory
        triggers: (consulte a tabela abaixo)
      (...)
```

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo deste item.
name	string	(obrigatório) Nome do item.
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), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), TELEMETRY_QUERY (24).
snmp_oid	string	(obrigatório para itens SNMP_AGENT) ID do objeto SNMP.
key	string	(obrigatório) Chave do item.
delay	string	Intervalo de atualização do item. Padrão: 1m. O valor será sempre 0 para itens TRAP.
history	string	Período (usando sufixo de tempo, macro de usuário ou macro de LLD) durante o qual os dados de histórico devem ser armazenados. Padrão: 31d.
trends	string	Período (usando sufixo de tempo, macro de usuário ou macro de LLD) durante o qual 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), JSON (6).
allowed_hosts	string	Lista de endereços IP, separados por vírgulas, dos hosts autorizados a enviar dados para o item. Compatível com itens TRAP e HTTP_AGENT.
units	string	Unidades do valor recebido (bps, B etc.).
params	text	Parâmetros adicionais dependendo do tipo de 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. Compatível com itens IPMI.

Elemento	Tipo	Descrição
authtype	string	Tipo de autenticação. Compatível com itens SSH e HTTP_AGENT. Valores possíveis para itens SSH:[1] PASSWORD (0, padrão), PUBLIC_KEY (1). Valores possíveis para itens HTTP_AGENT:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
username	string	(obrigatório para itens SSH e TELNET) Nome de usuário para autenticação. Compatível com itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `password` (veja abaixo) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
password	string	(obrigatório para itens SSH e TELNET) Senha para autenticação. Compatível com itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usada 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 de chave pública.
privatekey	string	(obrigatório para itens SSH) Nome do arquivo de chave privada.
description	text	Descrição do item.
inventory_link	string	Campo do inventário do host preenchido pelo item. Valores possíveis:[1] NONE (0), ALIAS (4) etc. (consulte Inventário do host para ver os campos compatíveis).
valuemap		Elemento raiz para mapas de valores do item.
name	string	(obrigatório) Nome do mapa de valores a ser usado para o item.
logtimefmt	string	Formato da hora nas entradas de log. Compatível com itens do tipo de valor LOG.
preprocessing		Elemento raiz para o pré-processamento de valores do item.
step		Elemento raiz para etapas de pré-processamento de valores de itens do template.
jmx_endpoint	string	Endpoint JMX. Compatível com 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	Tempo limite da solicitação de coleta de dados do item. Compatível com 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. Compatível com itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta. Compatível com itens HTTP_AGENT.
parameters		Elemento raiz para parâmetros definidos pelo usuário. Compatível com 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. Compatível com itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dados do corpo da solicitação HTTP(S). Compatível com itens HTTP_AGENT.
status_codes	string	Intervalos de códigos de status HTTP exigidos, separados por vírgulas. Compatível com itens HTTP_AGENT.
follow_redirects	string	Seguir redirecionamentos da resposta durante a coleta de dados. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0), YES (1, padrão).
post_type	string	Tipo do corpo dos dados enviados. Compatível com itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (2), XML (3).
http_proxy	string	String de conexão do proxy HTTP(S). Compatível com itens HTTP_AGENT.
headers		Elemento raiz para cabeçalhos de solicitação HTTP(S). Compatível com itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do cabeçalho.
value	string	(obrigatório para itens HTTP_AGENT) Valor do cabeçalho.

Elemento	Tipo	Descrição
retrieve_mode	string	Parte da resposta que deve ser armazenada. Compatível com itens HTTP_AGENT. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
request_method	string	Tipo de método da solicitação. Compatível com itens HTTP_AGENT. Valores possíveis:[1] GET (0, padrão), POST (1), PUT (2), HEAD (3).
output_format	string	Como processar a resposta. Compatível com itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (1).
allow_traps	string	Permitir que o valor seja preenchido de forma semelhante ao item trapper. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
ssl_cert_file	string	Caminho do arquivo de chave pública SSL. Compatível com itens HTTP_AGENT.
ssl_key_file	string	Caminho do arquivo de chave privada SSL. Compatível com itens HTTP_AGENT.
ssl_key_password	string	Senha do arquivo de chave SSL. Compatível com itens HTTP_AGENT.
verify_peer	string	Indica se o certificado do host deve ser validado quanto à autenticidade. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Indica se o nome do host da conexão deve ser validado para verificar se corresponde ao nome no certificado do host. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
time_shift	string	Quanto para trás, em relação à hora atual, termina o intervalo de tempo consultado pelo item. Compatível com itens TELEMETRY_QUERY.
lookback_limit	string	Quanto para trás, em relação à hora atual, o item consulta dados quando não há uma consulta anterior da qual continuar (por exemplo, na primeira execução do item) ou quando a consulta anterior é mais antiga que `delay` (por exemplo, porque o Zabbix server estava indisponível). Compatível com itens TELEMETRY_QUERY.
granularity	string	Duração de cada parte em que o intervalo de tempo completo da consulta é dividido. Compatível com itens TELEMETRY_QUERY.
query		(obrigatório para itens TELEMETRY_QUERY) Elemento raiz da consulta de telemetria. Define o que o item consulta na fonte de dados, como filtra os dados e como agrega o resultado.
signal_type	string	(obrigatório) Tipo de sinal consultado pelo item. Compatível com itens TELEMETRY_QUERY. Valores possíveis:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obrigatório para o tipo de sinal METRICS) Tipo de ponto de métrica consultado pelo item. Compatível com itens TELEMETRY_QUERY. Valores possíveis:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obrigatório) Elemento raiz das colunas de dados de telemetria consultadas pelo item. Este elemento contém os seguintes elementos: `column (string)` - (obrigatório) Nome da coluna de dados de telemetria consultada pelo item. `attribute_key (string)` - (obrigatório para colunas que contêm pares de chave-valor) Nome da chave consultada pelo item dentro da coluna que contém pares de chave-valor.

Elemento	Tipo	Descrição
aggregated_columns		(obrigatório) Elemento raiz das colunas agregadas de dados de telemetria consultadas pelo item. Este elemento contém os seguintes elementos: `column` (`string`) - (obrigatório para as funções MIN, MAX, AVG, SUM e PERCENTILE) Nome da coluna de dados de telemetria consultada pelo item. `function` (`string`) - (obrigatório) Função aplicada pelo item aos registros da coluna. Valores possíveis:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (obrigatório para funções PERCENTILE) Percentil calculado pelo item a partir dos registros da coluna. `alias` (`string`) - (obrigatório) Nome exclusivo do valor calculado, usado para identificá-lo no valor do item.
filter		(obrigatório) Elemento raiz do filtro da consulta de telemetria para os registros de coluna na consulta. Este elemento contém os seguintes elementos: `evaltype` (`string`) - (obrigatório) Método de avaliação da condição do filtro. Valores possíveis:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obrigatório para métodos de avaliação FORMULA) Expressão definida pelo usuário usada para avaliar condições de filtros com uma expressão personalizada. `conditions` (`string`) - (obrigatório) Elemento raiz das condições de filtro de dados de telemetria que um registro em uma coluna de dados de telemetria deve atender. O elemento `conditions` contém os seguintes elementos: `column` (`string`) - (obrigatório) Nome da coluna de dados de telemetria à qual as condições de filtro se aplicam. `attribute_key` (`string`) - (obrigatório para colunas que contêm pares de chave-valor) Nome da chave à qual a condição se aplica, dentro da coluna que contém pares de chave-valor. `operator` (`string`) - (obrigatório) Operador da condição de filtro. Valores possíveis para colunas de valor único:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valores possíveis para colunas que contêm pares de chave-valor:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obrigatório para métodos de avaliação FORMULA) ID exclusivo arbitrário usado para referenciar a condição em uma expressão personalizada. `value` (`string`) - (obrigatório para operações EQUAL, NOT_EQUAL, LIKE e NOT LIKE) Valor a ser comparado.
tags		Elemento raiz das tags do item.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
triggers		Elemento raiz dos triggers de itens do template.

Note:
Veja também: Objeto item (consulte a propriedade relevante com um nome correspondente).

Etapas de pré-processamento do valor do item do template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
```

```
            parameters:
              - 12h
        (...)
```

Element	Type	Description
type	string	(obrigatório) O tipo da etapa de pré-processamento do valor do item. Valores possíveis:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, calculado: valor recebido - valor anterior), CHANGE_PER_SECOND (10, calculado: (valor recebido - valor anterior)/(hora atual - hora da última verificação)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(obrigatório) Elemento raiz para os parâmetros da etapa de pré-processamento do valor do item.
parameter	string	Parâmetro individual da etapa de pré-processamento do valor do item.
error_handler	string	Tipo de ação usado 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 erro.

Note:
Veja também: Objeto de pré-processamento do item (consulte a propriedade relevante com o nome correspondente).

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
      (...)
```

Element	Type	Description
uuid	string	(obrigatório) Identificador exclusivo para este trigger.
expression	string	(obrigatório) Expressão do trigger.
recovery_mode	string	Base para gerar eventos OK. Valores possíveis:[1] EXPRESSION (0, padrão), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Expressão de recuperação do trigger.
correlation_mode	string	Modo de correlação (sem correlação de eventos ou correlação por tag). Valores possíveis:[1] DISABLED (0, padrão), TAG_VALUE (1).
correlation_tag	string	Nome da tag a ser usada para correlação de eventos.
name	string	(obrigatório) Nome do trigger.
event_name	string	Nome do evento.
opdata	string	Dados operacionais.
url_name	string	Rótulo para a URL associada ao trigger.
url	string	URL associada ao trigger.
status	string	Status do trigger. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
priority	string	Severidade do trigger. Valores possíveis:[1] NOT_CLASSIFIED (0, padrão), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Descrição do trigger.
type	string	Tipo de geração de evento (evento de problema único ou múltiplos eventos de problema). Valores possíveis:[1] SINGLE (0, padrão), MULTIPLE (1).
manual_close	string	Fechamento manual de eventos de problema. Valores possíveis:[1] NO (0, padrão), YES (1).
dependencies		Elemento raiz para dependências.
name	string	(obrigatório) Nome do trigger de dependência.
expression	string	(obrigatório) Expressão do trigger de dependência.
recovery_expression	string	Expressão de recuperação do trigger de dependência.
tags		Elemento raiz para tags do trigger.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto Trigger (consulte a propriedade relevante com o nome correspondente).

Regras de descoberta de baixo nível de template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'A descoberta de interfaces de rede.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
A maioria dos elementos da regra de descoberta de baixo nível de template é a mesma que a dos itens de template. A tabela abaixo descreve os elementos que diferem dos itens de template.

Element	Type	Description
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 descobertos que não existirem mais serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para excluir recursos LLD perdidos. Valores possíveis: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos descobertos que não existirem mais 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) da regra de descoberta de baixo nível de protótipo. Esta propriedade indica que se trata de 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 da 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 o pré-processamento do valor da regra de descoberta de baixo nível.
step		Elemento raiz para elementos de etapa de pré-processamento do valor da regra de descoberta de baixo nível, que são os mesmos que para etapas de pré-processamento do valor do item de template, exceto por haver menos valores possíveis. Veja também: objeto de pré-processamento da regra LLD.
overrides		Elemento raiz para regras de substituição da regra de descoberta de baixo nível.
name	string	(obrigatório) Nome exclusivo da substituição.
step	string	(obrigatório) Número de ordem exclusivo da substituição.
stop	string	Interromper o processamento das próximas substituições se houver correspondência.
filter		Elemento raiz para elementos de filtro da regra de substituição da 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 substituição da regra de descoberta de baixo nível de template.

Note:
Veja também: objeto da regra LLD (consulte a propriedade relevante com um nome correspondente).

Filtros da regra de descoberta de baixo nível do 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
        (...)
```

Element	Type	Description
evaltype	string	Substitui o método de avaliação da condição do filtro. Valores possíveis:[1] AND_OR (0, padrão), AND (1), OR (2), FORMULA (3).
formula	string	Fórmula de cálculo personalizada para as condições do filtro.
conditions		Elemento raiz para as condições do filtro.
macro	string	(obrigatório) O nome da macro de descoberta de baixo nível a ser verificada.
value	string	Valor com o qual 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 na expressão personalizada. Pode conter apenas letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las depois.

Note:
Veja também: Objeto de filtro da regra LLD (consulte a propriedade relevante com o nome correspondente).

Operações de override da regra de descoberta de baixo nível do template

Element	Type	Description
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 override. 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 da condição de override.
status	string	Status do objeto após a operação de override.
discover	string	Se o objeto é adicionado como resultado da descoberta.
delay	string	Intervalo de atualização definido para o item prototype após a operação de override.
history	string	Período de armazenamento do histórico definido para o item prototype após a operação de override.

Element	Type	Description
trends	string	Período de armazenamento de tendências definido para o item prototype após a operação de override.
severity	string	Severidade do trigger prototype definida após a operação de override.
tags		Elemento raiz para o conjunto de tags definido para o objeto após a operação de override.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
templates		Elemento raiz para os templates vinculados ao host prototype após a operação de override.
name	string	(obrigatório) Nome do template.
inventory_mode	string	Modo de inventário do host prototype definido após a operação de override.

Note:
Veja também: Objeto da operação de override da regra LLD (consulte a propriedade relevante com o nome correspondente).

Cenários web do template

Element	Type	Description
uuid	string	(required) Identificador exclusivo para este cenário web.
name	string	(required) Nome do cenário web.
delay	string	Frequência (usando segundos, time suffix, ou user macro) de execução do cenário web. Default: 1m.
attempts	integer	O número de tentativas para executar as etapas do cenário web. Valores possíveis:[1] 1-10 (default: 1).
agent	string	Agent do cliente. O Zabbix irá se passar pelo navegador selecionado. Isso é útil quando um site retorna conteúdo diferente para navegadores diferentes. Default: Zabbix.
http_proxy	string	Proxy que será usado pelo cenário web, informado como: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento raiz para variáveis do cenário web que podem ser usadas nas etapas do cenário.
name	string	(required) Nome da variável.
value	text	(required) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma requisição. Os cabeçalhos devem ser listados usando a mesma sintaxe que teriam no protocolo HTTP.
name	string	(required) Nome do cabeçalho.
value	text	(required) Valor do cabeçalho.
status	string	Status do cenário web. Valores possíveis:[1] ENABLED (0, default), DISABLED (1).
authentication	string	Método de autenticação. Valores possíveis:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3), ou Digest (4).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.
http_password	string	Senha usada para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.
verify_peer	string	Verifica o certificado SSL do servidor web. Valores possíveis:[1] NO (0, default), 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, default), 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		(required) Elemento raiz para etapas de cenário web do template.
tags		Elemento raiz para tags do cenário web.
tag	string	(required) 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

Element	Type	Description
name	string	(required) Nome da etapa do cenário web.
url	string	(required) URL para monitoramento.
query_fields		Elemento raiz para parâmetros de consulta (uma matriz de campos HTTP a serem adicionados à URL ao executar uma requisição).
name	string	(required) 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 uma matriz de campos HTTP (dados de campos de formulário)).
name	string	(required) Nome do campo de post.
value	string	(required) Valor do campo de post.
variables		Elemento raiz de variáveis no nível da etapa (macros) que devem ser aplicadas após esta etapa. Se o valor da variável tiver o prefixo 'regex:', então seu valor é extraído dos dados retornados por esta etapa de acordo com o padrão de expressão regular que segue o prefixo 'regex:'
name	string	(required) Nome da variável.
value	text	(required) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma requisição.
name	string	(required) Nome do cabeçalho.
value	text	(required) Valor do cabeçalho.
follow_redirects	string	Seguir redirecionamentos HTTP. Valores possíveis:[1] NO (0), YES (1, padrão).
retrieve_mode	string	Modo de recuperação da resposta HTTP. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
timeout	string	Tempo limite (em segundos, time suffix ou user macro) 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: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Segundos desde a última vez que o agent foi visto'
            description: 'Tempo limite após o qual o agent é considerado indisponível.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'Limite de utilização de CPU expresso'
```

```
            description: 'Limite crítico de utilização de CPU expresso em %. No intervalo de 0 a 100, incl
            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'
        (...)
```

Element	Type	Description
type	string	(obrigatório) Tipo do 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 na qual 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 um estado marcado. Valores possíveis:[1] 0 (false), 1 (true). Suportado para macros CHECKBOX.
unchecked	string	(obrigatório para macros CHECKBOX) Valor que representa um estado desmarcado. Valores possíveis:[1] 0 (false), 1 (true). 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: (see table below)
        (...)
```

Element	Type	Description
uuid	string	(required) Identificador exclusivo para este dashboard.
name	string	(required) Nome do dashboard de template.
display pe-riod	integer	Período de exibição das páginas do dashboard.

Element		Type	Description
auto_start		string	Início automático da apresentação de slides. Valores possíveis:[1] NO (0), YES (1, padrão).
pages			Elemento raiz para as páginas do dashboard de template.
	name	string	Nome da página.
	display period	integer	Intervalo da apresentação de slides.
	sortorder	integer	Ordem de classificação da página.
widgets			Elemento raiz para widgets do dashboard de template.

Note:
Veja também: Objeto de dashboard de template (consulte a propriedade relevante com o nome correspondente).

Widgets do dashboard do template

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Gráfico de utilização de espaço (relativo ao tot
                    - 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})]: Gráfico de uso de espaço, em %, (relativo ao máx
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Description
type	string	(obrigatório) Tipo do widget.
name	string	Nome do widget.

Element	Type	Description
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 do campo do widget. Valores possíveis:[1] INTEGER (0), STRING (1), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(obrigatório) Nome do campo do widget.
value	mixed	(obrigatório) Valor do campo do widget, dependendo do tipo de campo.

Note:
Veja também: Objeto do 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 os 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 value map (consulte a propriedade relevante com um nome correspondente).

Notas de rodapé [1] Valores inteiros da API entre parênteses, por exemplo, ENABLED (0), são mencionados apenas para referência. Para mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

Hosts

Visão geral

Hosts são exportados com muitos objetos relacionados e relações entre objetos.

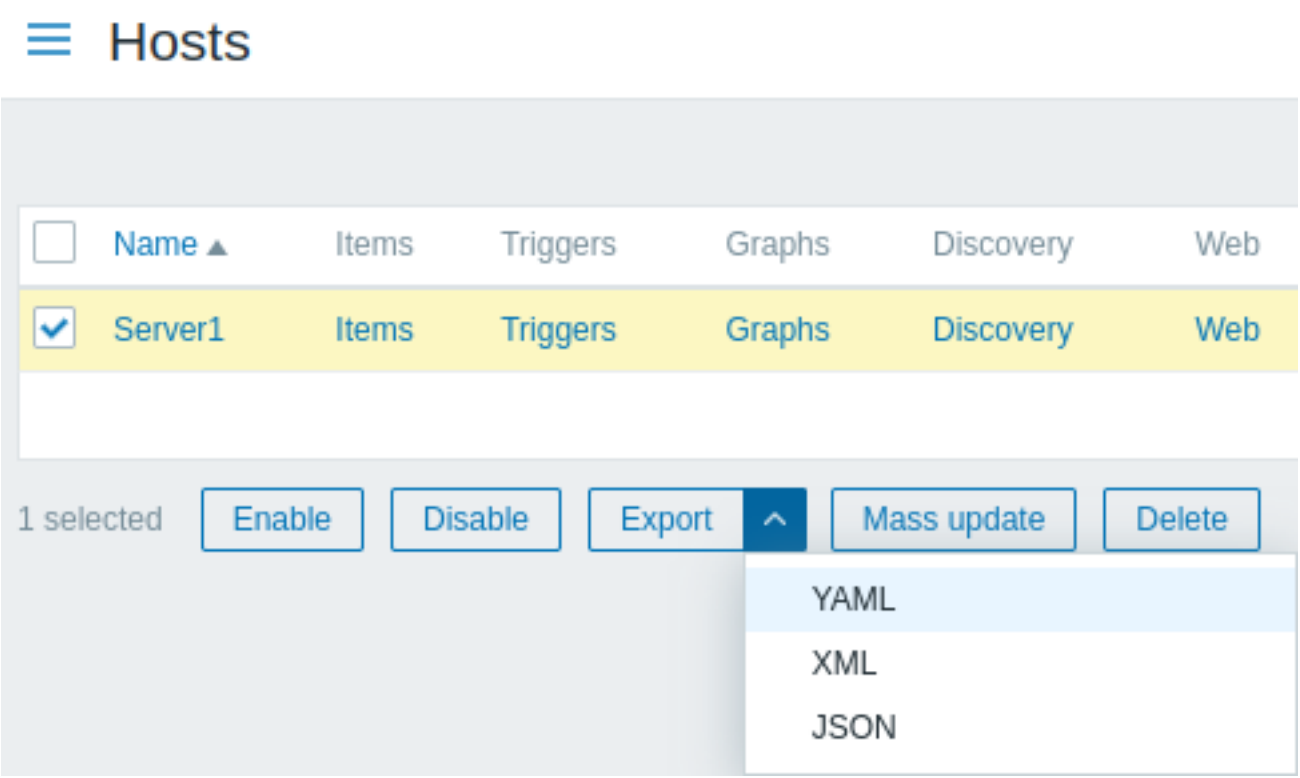
A exportação de host contém:

- Grupos de host vinculados
- Dados do host
- Vinculação de template
- Vinculação de grupo de host
- Interfaces do host
- items 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 valor
- Gráficos vinculados

Exportando

Para exportar hosts, faça o seguinte:

1. Acesse *Data collection* > *Hosts*.
2. Marque as caixas de seleção dos hosts a serem exportados.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



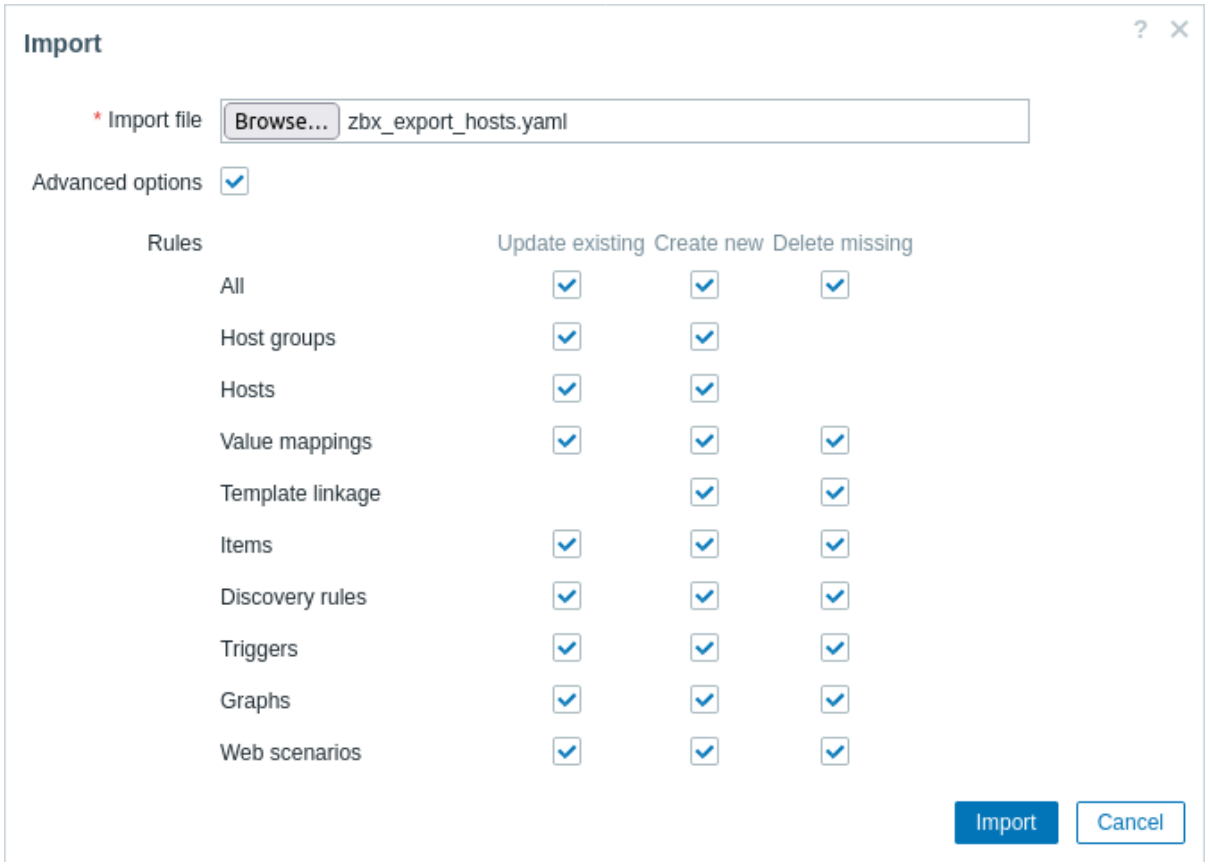
Dependendo do formato selecionado, os hosts são exportados para um arquivo local com um nome padrão:

- `zabbix_export_hosts.yaml` - na exportação YAML;
- `zabbix_export_hosts.xml` - na exportação XML;
- `zabbix_export_hosts.json` - na exportação JSON.

Importação

Para importar hosts, faça o seguinte:

1. Acesse *Coleta de dados* > *Hosts*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (veja abaixo).
5. Clique em *Importar* no canto inferior direito do formulário de configuração.



Se você marcar a caixa de seleção *Opções avançadas*, será exibida uma lista detalhada de todos os elementos que podem ser importados. Marque ou desmarque cada regra de importação conforme necessário.

Se você clicar na caixa de seleção da linha *Todos*, todos os elementos abaixo dela serão marcados ou 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*, as vinculações atuais de template que não estiverem presentes no arquivo de importação serão desvinculadas. As entidades (items, triggers, gráficos etc.) herdadas dos templates desvinculados não serão removidas (a menos que a opção *Excluir ausentes* também esteja selecionada para cada entidade).

Não é possível revisar as alterações nos hosts durante a importação. 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 outros.

Cada elemento tem uma finalidade específica 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 depois que o host é criado. Desvincular mantém todos os elementos na configuração (ao contrário de desvincular e limpar, que os remove).

Uma reticência (`...`) indica elementos omitidos por brevidade. A observação (`veja a tabela abaixo`) é usada quando o elemento é explicado com mais detalhes em seções posteriores.

```
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		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 do host.

Hosts

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(obrigatório) Nome único 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.

Element	Type	Description
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).
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 items 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 do 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: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Description
default	string	Se esta é a interface principal do host. Observe que pode haver apenas uma interface principal de um tipo em um host. Valores possíveis:[1] NO (0), YES (1, default).
type	string	Tipo de interface. Valores possíveis:[1] ZABBIX (1, default), SNMP (2), IPMI (3), JMX (4).
useip	string	Se deve usar IP como a interface para conexão com o host (caso contrário, será usado DNS). Valores possíveis:[1] NO (0), YES (1, default).
ip	string	(obrigatório para conexões IP) Endereço IP (IPv4 ou IPv6).
dns	string	(obrigatório para conexões DNS) Nome DNS.

Element	Type	Description
port	string	Número da porta.
details		Elemento raiz para detalhes da interface.
version	string	Use esta versão do SNMP. Valores possíveis:[1] SNMPV1 (1), SNMP_V2C (2, default), SNMP_V3 (3).
community	string	(obrigatório para itens SNMPv1 e SNMPv2) Comunidade SNMP.
max_repetitions	string	Valor máximo de repetição para solicitações bulk nativas de SNMP (PDUs GetBulkRequest). Compatível com itens SNMPv2 e SNMPv3 (`discovery[]` e `walk[]`). Padrão: 10.
contextname	string	Nome do contexto SNMPv3. Compatível com itens SNMPv3.
securityname	string	Nome de segurança SNMPv3. Compatível com itens SNMPv3.
securitylevel	string	Nível de segurança SNMPv3. Compatível com itens SNMPv3. Valores possíveis:[1] NOAUTHNOPRIV (0, default), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Protocolo de autenticação SNMPv3. Compatível com itens SNMPv3. Valores possíveis:[1] MD5 (0, default), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Frase secreta de autenticação SNMPv3. Compatível com itens SNMPv3.
privprotocol	string	Protocolo de privacidade SNMPv3. Compatível com itens SNMPv3. Valores possíveis:[1] DES (0, default), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Frase secreta de privacidade SNMPv3. Compatível com itens SNMPv3.
bulk	string	Usar solicitações bulk para SNMP. Valores possíveis:[1] NO (0), YES (1, default).
interface_ref	string	Nome de referência da interface a ser usado em itens (formato: `if<N>`).

Note:
Veja também: Objeto da interface do host (consulte a propriedade relevante com o nome correspondente).

Itens do host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Uso do disco do cache do Binlog'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Número de transações que usaram um cache de disco temporário porque não couberam n
          preprocessing: (consulte a tabela abaixo)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Utilização do buffer pool'
          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 em relação ao total de páginas no buffer pool.'
        tags:
          - tag: component
            value: memory
        triggers: (consulte a tabela abaixo)
      (...)
      - name: Uptime
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: 'Número de segundos durante os quais o server esteve em execução.'
        preprocessing: (consulte a tabela abaixo)
        master_item:
          key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
        tags:
          - tag: component
            value: application
        triggers: (consulte a tabela abaixo)
      (...)
```

Elemento	Tipo	Descrição
name	string	(obrigatório) Nome do item.
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), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(obrigatório para itens SNMP_AGENT) ID do objeto SNMP.
key	string	(obrigatório) Chave do item.
delay	string	Intervalo de atualização do item. Padrão: 1m. O valor será sempre 0 para itens TRAP.
history	string	Período (usando sufixo de tempo, macro de usuário ou macro LLD) durante o qual os dados de histórico devem ser armazenados. Padrão: 31d.
trends	string	Período (usando sufixo de tempo, macro de usuário ou macro LLD) durante o qual 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 do valor recebido. Valores possíveis:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, padrão), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Lista de endereços IP, separados por vírgulas, dos hosts autorizados a enviar dados para o item. Compatível com itens TRAP e HTTP_AGENT.
units	string	Unidades do valor recebido (bps, B etc.).
params	text	Parâmetros adicionais dependendo do tipo de 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. Compatível com itens IPMI.

Elemento	Tipo	Descrição
authtype	string	Tipo de autenticação. Compatível com itens SSH e HTTP_AGENT. Valores possíveis para itens SSH:[1] PASSWORD (0, padrão), PUBLIC_KEY (1). Valores possíveis para itens HTTP_AGENT:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
username	string	(obrigatório para itens SSH e TELNET) Nome de usuário para autenticação. Compatível com itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `password` (consulte abaixo) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
password	string	(obrigatório para itens SSH e TELNET) Senha para autenticação. Compatível com itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `username` (consulte acima) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
publickey	string	(obrigatório para itens SSH) Nome do arquivo de chave pública.
privatekey	string	(obrigatório para itens SSH) Nome do arquivo de chave privada.
description	text	Descrição do item.
inventory_link	string	Campo do inventário do host preenchido pelo item. Valores possíveis:[1] NONE (0), ALIAS (4) etc. (consulte Inventário do host para ver os campos compatíveis).
valuemap		Elemento raiz para mapas de valores do item.
name	string	(obrigatório) Nome do mapa de valores a ser usado para o item.
logtimefmt	string	Formato da hora nas entradas de log. Compatível com itens do tipo de valor LOG.
preprocessing		Elemento raiz para o pré-processamento dos valores do item.
step		Elemento raiz para etapas de pré-processamento dos valores dos itens do host.
interface_ref	string	Referência à interface do host (formato: `if<N>`).
jmx_endpoint	string	Endpoint JMX. Compatível com 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	Tempo limite da solicitação de consulta dos dados do item. Compatível com a lista Timeouts de tipos de item.
url	string	(obrigatório para itens HTTP_AGENT) String de URL.
query_fields		Elemento raiz para parâmetros de consulta. Compatível com itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta. Compatível com itens HTTP_AGENT.
parameters		Elemento raiz para parâmetros definidos pelo usuário. Compatível com 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. Compatível com itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dados do corpo da solicitação HTTP(S). Compatível com itens HTTP_AGENT.
status_codes	string	Intervalos dos códigos de status HTTP obrigatórios, separados por vírgulas. Compatível com itens HTTP_AGENT.
follow_redirects	string	Seguir redirecionamentos da resposta durante a consulta dos dados. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0), YES (1, padrão).
post_type	string	Tipo do corpo dos dados enviados. Compatível com itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (2), XML (3).
http_proxy	string	String de conexão do proxy HTTP(S). Compatível com itens HTTP_AGENT.
headers		Elemento raiz para cabeçalhos de solicitação HTTP(S). Compatível com itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do cabeçalho.
value	string	(obrigatório para itens HTTP_AGENT) Valor do cabeçalho.

Elemento	Tipo	Descrição
retrieve_mode	string	Parte da resposta que deve ser armazenada. Compatível com itens HTTP_AGENT. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
request_method	string	Tipo de método da solicitação. Compatível com itens HTTP_AGENT. Valores possíveis:[1] GET (0, padrão), POST (1), PUT (2), HEAD (3).
output_format	string	Como processar a resposta. Compatível com itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (1).
allow_traps	string	Permitir que o valor seja preenchido da mesma forma que para um item trapper. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
ssl_cert_file	string	Caminho do arquivo de chave pública SSL. Compatível com itens HTTP_AGENT.
ssl_key_file	string	Caminho do arquivo de chave privada SSL. Compatível com itens HTTP_AGENT.
ssl_key_password	string	Senha do arquivo de chave SSL. Compatível com itens HTTP_AGENT.
verify_peer	string	Indica se o certificado do host deve ser validado como autêntico. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Indica se o nome do host da conexão deve ser validado para corresponder ao nome no certificado do host. Compatível com itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
time_shift	string	Quanto para trás, antes da hora atual, termina o intervalo de tempo consultado pelo item. Compatível com itens TELEMETRY_QUERY.
lookback_limit	string	Quanto para trás, antes da hora atual, o item consulta dados quando não há uma consulta anterior a partir da qual continuar (por exemplo, na primeira execução do item) ou quando a consulta anterior é mais antiga que `delay` (por exemplo, porque o Zabbix server estava indisponível). Compatível com itens TELEMETRY_QUERY.
granularity	string	Duração de cada parte em que o intervalo de tempo completo da consulta é dividido. Compatível com itens TELEMETRY_QUERY.
query		(obrigatório para itens TELEMETRY_QUERY) Elemento raiz para a consulta de telemetria. Define o que o item consulta na fonte de dados, como filtra os dados e como agrega o resultado.
signal_type	string	(obrigatório) Tipo de sinal consultado pelo item. Compatível com itens TELEMETRY_QUERY. Valores possíveis:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obrigatório para o tipo de sinal METRICS) Tipo de ponto de métrica consultado pelo item. Compatível com itens TELEMETRY_QUERY. Valores possíveis:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obrigatório) Elemento raiz para as colunas de dados de telemetria consultadas pelo item. Este elemento contém os seguintes elementos: `column (string)` - (obrigatório) Nome da coluna de dados de telemetria consultada pelo item. `attribute_key (string)` - (obrigatório para colunas que contêm pares chave-valor) Nome da chave consultada pelo item dentro da coluna que contém pares chave-valor.

Elemento	Tipo	Descrição
aggregated_columns		(obrigatório) Elemento raiz para as colunas agregadas de dados de telemetria consultadas pelo item. Este elemento contém os seguintes elementos: `column` (`string`) - (obrigatório para as funções MIN, MAX, AVG, SUM e PERCENTILE) Nome da coluna de dados de telemetria consultada pelo item. `function` (`string`) - (obrigatório) Função aplicada pelo item aos registros da coluna. Valores possíveis:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (obrigatório para funções PERCENTILE) Percentil calculado pelo item a partir dos registros da coluna. `alias` (`string`) - (obrigatório) Nome exclusivo do valor calculado, usado para identificá-lo no valor do item.
filter		(obrigatório) Elemento raiz para o filtro da consulta de telemetria aplicado aos registros de coluna na consulta. Este elemento contém os seguintes elementos: `evaltype` (`string`) - (obrigatório) Método de avaliação da condição do filtro. Valores possíveis:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obrigatório para métodos de avaliação FORMULA) Expressão definida pelo usuário usada para avaliar as condições dos filtros com uma expressão personalizada. `conditions` (`string`) - (obrigatório) Elemento raiz para as condições de filtro de dados de telemetria que um registro em uma coluna de dados de telemetria deve atender. O elemento `conditions` contém os seguintes elementos: `column` (`string`) - (obrigatório) Nome da coluna de dados de telemetria à qual as condições do filtro se aplicam. `attribute_key` (`string`) - (obrigatório para colunas que contêm pares chave-valor) Nome da chave à qual a condição se aplica, dentro da coluna que contém pares chave-valor. `operator` (`string`) - (obrigatório) Operador da condição do filtro. Valores possíveis para colunas de valor único:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valores possíveis para colunas que contêm pares chave-valor:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obrigatório para métodos de avaliação FORMULA) ID exclusivo arbitrário usado para referenciar a condição em uma expressão personalizada. `value` (`string`) - (obrigatório para operações EQUAL, NOT_EQUAL, LIKE e NOT LIKE) Valor a ser comparado.
tags		Elemento raiz para as tags do item.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
triggers		Elemento raiz para os triggers dos itens do host.

Note:
Consulte também: Objeto item (consulte a propriedade relevante com um nome correspondente).

Etapas de pré-processamento do valor do item do host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Uso de disco do cache de binlog'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
```

```
              - $.Binlog_cache_disk_use
          - type: DISCARD_UNCHANGED_HEARTBEAT
            parameters:
              - 6h
        (...)
```

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 do último check)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(obrigatório) Elemento raiz para os parâmetros da etapa de pré-processamento do valor do item.
parameter	string	Parâmetro individual da etapa de pré-processamento do valor do item.
error_handler	string	Tipo de ação usado 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 erro.

Note:
Veja também: Objeto de pré-processamento de item (consulte a propriedade relevante com o nome correspondente).

Triggers de item do 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: 'O Zabbix não recebeu nenhum dado para os items nos últimos 30 minutos.'
              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: 'O uptime do MySQL é inferior a 10 minutos.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Element	Type	Description
uuid	string	(required) Identificador exclusivo para este trigger.
expression	string	(required) Expressão do trigger.
recovery_mode	string	Base para gerar eventos OK. Valores possíveis:[1] EXPRESSION (0, default), 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, default), TAG_VALUE (1).
correlation_tag	string	Nome da tag a ser usada para correlação de eventos.
name	string	(required) 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, default), DISABLED (1).
priority	string	Severidade do trigger. Valores possíveis:[1] NOT_CLASSIFIED (0, default), 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 vários eventos de problema). Valores possíveis:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	Fechamento manual de eventos de problema. Valores possíveis:[1] NO (0, default), YES (1).
dependencies		Elemento raiz para dependências.
name	string	(required) Nome do trigger de dependência.
expression	string	(required) Expressão do trigger de dependência.
recovery_expression	string	Expressão de recuperação do trigger de dependência.
tags		Elemento raiz para tags do trigger.
tag	string	(required) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto Trigger (consulte a propriedade relevante com o nome correspondente).

Regras de descoberta em baixo nível do host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$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 em baixo nível do host é a mesma que a dos itens do host. A tabela abaixo descreve os elementos que diferem dos itens do host.

Element	Type	Description
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 em baixo nível.
filter		Elemento raiz para filtros de regras de descoberta em baixo nível do host.
lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos descobertos que não existirem mais serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para excluir recursos LLD perdidos. Valores possíveis: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos descobertos que não existirem mais 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 os elementos de protótipo de item do host, que são os mesmos que os de itens do host.
trigger_prototypes		Elemento raiz para os elementos de protótipo de trigger do host, que são os mesmos que os de triggers de item do host.
graph_prototypes		Elemento raiz para os protótipos de gráfico do host, que são os mesmos que os de gráficos do host.
host_prototypes		Elemento raiz para os protótipos de host, que são os mesmos que os de hosts.
parent_discovery_rule		Elemento raiz para a regra de descoberta em baixo nível pai (ou protótipo de regra) do protótipo de regra de descoberta em baixo nível. Esta propriedade indica que se trata de um protótipo de regra LLD, filho direto da regra/protótipo de regra referenciado.
key	string	(obrigatório) A chave da regra de descoberta em 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 os caminhos de macro da regra de descoberta em baixo nível.
lld_macro	string	(obrigatório) Nome da macro da regra de descoberta em baixo nível.
path	string	(obrigatório) Seletor do valor que será atribuído à macro correspondente.
preprocessing		Elemento raiz para o pré-processamento do valor da regra de descoberta em baixo nível.
step		Elemento raiz para os elementos de etapa de pré-processamento do valor da regra de descoberta em baixo nível, que são os mesmos que os de etapas de pré-processamento do valor do item do host, exceto por haver menos valores possíveis. Veja também: objeto de pré-processamento da regra LLD.
overrides		Elemento raiz para as regras de substituição da regra de descoberta em baixo nível.
name	string	(obrigatório) Nome exclusivo da substituição.
step	string	(obrigatório) Número de ordem exclusivo da substituição.
stop	string	Interrompe o processamento das próximas substituições se houver correspondência.

Element	Type	Description
filter		Elemento raiz para os elementos de filtro da regra de substituição da regra de descoberta em baixo nível, que são os mesmos que os de filtros de regras de descoberta em baixo nível do host.
operations		Elemento raiz para operações de substituição de regras de descoberta em baixo nível do host.

Note:
Veja também: objeto da regra LLD (consulte a propriedade relevante com o nome correspondente).

Filtros da regra de descoberta de baixo nível do 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
          (...)
```

Element	Type	Description
evaltype	string	Substitui o método de avaliação das condições do filtro. Valores possíveis:[1] AND_OR (0, padrão), AND (1), OR (2), FORMULA (3).
formula	string	Fórmula de cálculo personalizada para as condições do filtro.
conditions		Elemento raiz para as condições do filtro.
macro	string	(obrigatório) O nome da macro de descoberta de baixo nível a ser verificada.
value	string	Valor com o qual comparar.
operator	string	Operador da condição. Valores possíveis:[1] MATCHES_REGEX (8, padrão), NOT_MATCHES_REGEX (9).
formulaid	string	(obrigatório) ID único arbitrário usado para referenciar uma condição na expressão personalizada. Pode conter apenas letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las depois.

Note:
Veja também: Objeto de filtro da regra LLD (consulte a propriedade relevante com o nome correspondente).

Operações de substituição da regra de descoberta de baixo nível do host

Element	Type	Description
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 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 da condição de substituição.
status	string	Status do objeto após a operação de substituição.
discover	string	Se o objeto é adicionado como resultado da descoberta.

Element	Type	Description
delay	string	Intervalo de atualização definido para o protótipo de item após a operação de substituição.
history	string	Período de armazenamento do histórico definido para o protótipo de item após a operação de substituição.
trends	string	Período de armazenamento de tendências definido para o protótipo de item após a operação de substituição.
severity	string	Severidade do protótipo de trigger definida após a operação de substituição.
tags		Elemento raiz para o conjunto de tags definido 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 da operação de substituição da regra LLD (consulte a propriedade relevante com o nome correspondente).

Cenários web do host

Element	Type	Description
uuid	string	(required) Identificador único para este cenário web.
name	string	(required) Nome do cenário web.
delay	string	Frequência (usando segundos, time suffix, ou user macro) 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 se passar pelo navegador selecionado. Isso é útil quando um site retorna conteúdo diferente para navegadores diferentes. Padrão: Zabbix.
http_proxy	string	Proxy que será usado pelo cenário web, fornecido como: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento raiz para variáveis do cenário web que podem ser usadas nas etapas do cenário.
name	string	(required) Nome da variável.
value	text	(required) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma requisição. Os cabeçalhos devem ser listados usando a mesma sintaxe que teriam no protocolo HTTP.
name	string	(required) Nome do cabeçalho.
value	text	(required) Valor do cabeçalho.
status	string	Status do cenário web. Valores possíveis:[1] ENABLED (0, default), DISABLED (1).
authentication	string	Método de autenticação. Valores possíveis:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3), ou Digest (4).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP), NTLM, Kerberos, ou Digest.
http_password	string	Senha usada para autenticação BASIC (HTTP), NTLM, Kerberos, ou Digest.
verify_peer	string	Verificar o certificado SSL do servidor web. Valores possíveis:[1] NO (0, default), 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, default), 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.

Element	Type	Description
steps		(required) Elemento raiz para etapas de cenário web do host.
tags		Elemento raiz para tags do cenário web.
tag	string	(required) 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

Element	Type	Description
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 campos de formulário)).
name	string	(obrigatório) Nome do campo de post.
value	string	(obrigatório) Valor do campo de post.
variables		Elemento raiz de variáveis no nível da etapa (macros) que devem ser aplicadas após esta etapa. Se o valor da variável tiver o prefixo 'regex:', então seu valor é extraído dos dados retornados por esta etapa de acordo com o padrão de expressão regular que segue o prefixo 'regex:'
name	string	(obrigatório) Nome da variável.
value	text	(obrigatório) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma requisição.
name	string	(obrigatório) Nome do cabeçalho.
value	text	(obrigatório) Valor do cabeçalho.
follow_redirects	string	Seguir redirecionamentos HTTP. Valores possíveis:[1] NO (0), YES (1, padrão).
retrieve_mode	string	Modo de recuperação da resposta HTTP. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
timeout	string	Tempo limite (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	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).

Gráficos do host

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (see table below)
    (...)
```

Element	Type	Description
uuid	string	Identificador exclusivo do gráfico.
name	string	(obrigatório) Nome do gráfico.

Element	Type	Description
width	integer	Largura do gráfico, em pixels. Usado para pré-visualização e para gráficos de pizza/explodidos. Valores possíveis:[1] 20-65535 (padrão: 900).
height	integer	Altura do gráfico, em pixels. Usado para pré-visualização e para gráficos de pizza/explodidos. Valores possíveis:[1] 20-65535 (padrão: 900).
yaxismin	double	Valor mínimo do eixo Y. Compatível com o valor mínimo FIXED do eixo Y. Padrão: 0.
yaxismax	double	Valor máximo do eixo Y. Compatível com o valor mínimo FIXED do eixo X. Padrão: 0.
show_work_period	string	Destacar horas não úteis. Compatível com gráficos NORMAL e STACKED. Valores possíveis:[1] NO (0), YES (1, padrão).
show_triggers	string	Exibir valores simples de trigger como uma linha. Compatível com gráficos NORMAL e STACKED. Valores possíveis:[1] NO (0), YES (1, padrão).
type	string	Tipo de gráfico. Valores possíveis:[1] NORMAL (0, padrão), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Exibir a legenda do gráfico. Valores possíveis:[1] NO (0), YES (1, padrão).
show_3d	string	Ativar estilo 3D. Compatível com gráficos NORMAL e STACKED. Valores possíveis:[1] NO (0, padrão), YES (1).
percent_left	double	Exibir a linha de percentil para o eixo esquerdo. Compatível com gráficos NORMAL. Padrão: 0.
percent_right	double	Exibir a linha de percentil para o eixo direito. Compatível com gráficos NORMAL. Padrão: 0.
ymin_type_1	string	Valor mínimo do eixo Y. Compatível com gráficos NORMAL e STACKED. 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 de item individuais.
host	string	(obrigatório) host do item.
key	string	(obrigatório) chave do item.
ymax_type_1	string	Valor máximo do eixo Y. Compatível com gráficos NORMAL e STACKED. 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 de item individuais.
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 Graph (consulte a propriedade relevante com o nome correspondente).

Itens do gráfico do 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 NORMAL. 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 NORMAL e STACKED. Valores possíveis:[1] LEFT (0, padrão), RIGHT (1).
calc_fnc	string	Dados a serem desenhados se existir mais de um valor para um item. Valores possíveis:[1] MIN (1), AVG (2, padrão), MAX (4), ALL (7; mínimo, média e máximo; suportado para gráficos simples), LAST (9, suportado para gráficos de pizza/explodidos).
type	string	Tipo do item do gráfico. Valores possíveis:[1] SIMPLE (0, padrão), GRAPH_SUM (2; o valor do item representa a pizza inteira; 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 Graph item (consulte a propriedade relevante com um 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 value map (consulte a propriedade relevante com um nome correspondente).

Notas de rodapé [1] Os valores inteiros da API entre parênteses, por exemplo, ENABLED (0), são mencionados apenas para referência. Para mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

Mapas de rede

Visão geral

O mapa de rede exportação contém:

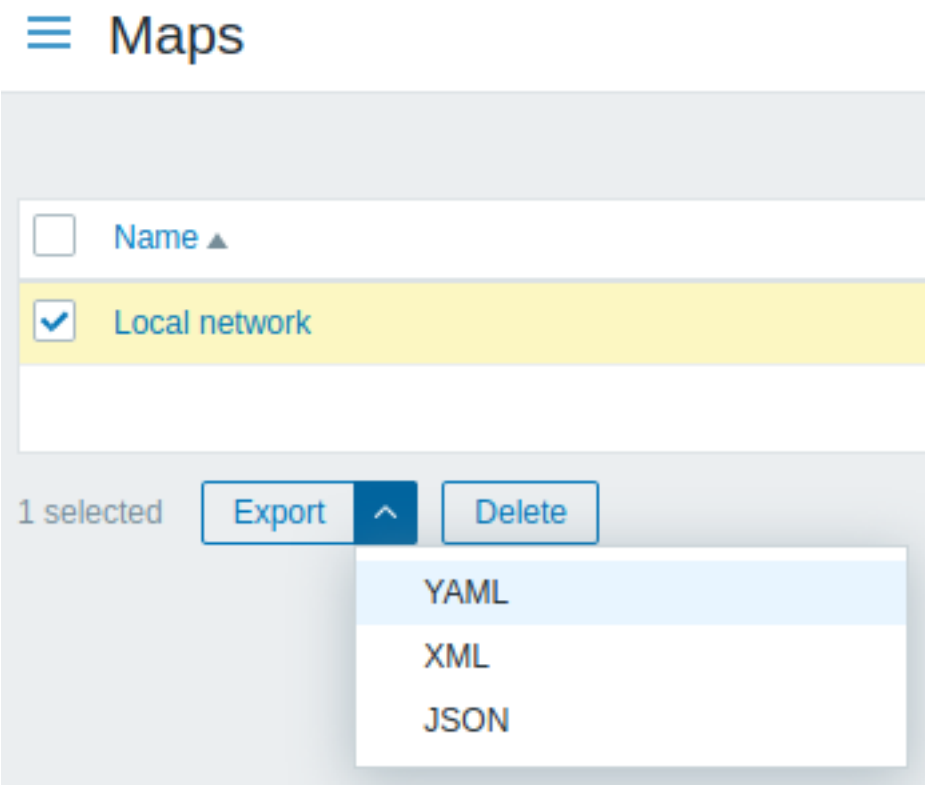
- Todas as imagens relacionadas
- Estrutura do mapa (todas as configurações do mapa, todos os elementos contidos com suas configurações, links do mapa e indicadores de status dos links do mapa)

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. Acesse *Monitoring* > *Maps*.
2. Marque as caixas de seleção dos mapas de rede a serem exportados.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



Dependendo do formato selecionado, os mapas são exportados para um arquivo local com um nome padrão:

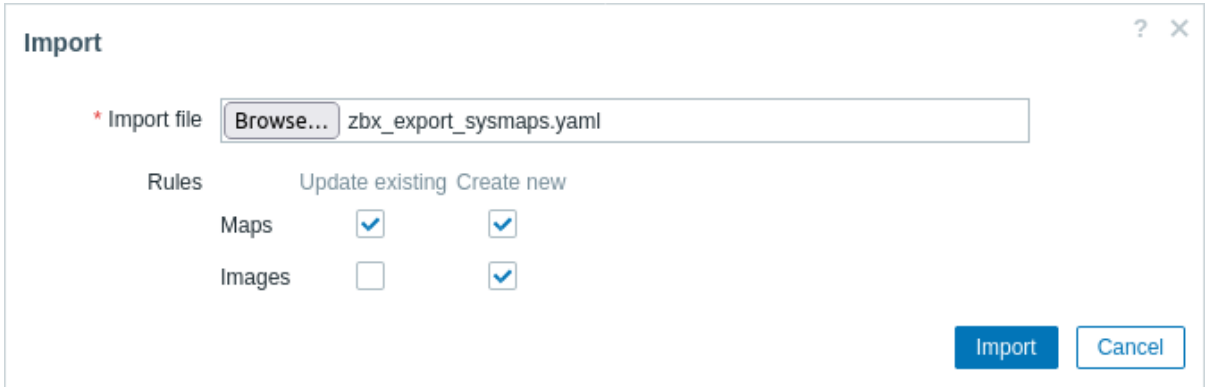
- `zabbix_export_maps.yaml` - na exportação YAML;
- `zabbix_export_maps.xml` - na exportação XML;
- `zabbix_export_maps.json` - na exportação JSON.

Importação

Para importar mapas de rede, faça o seguinte:

1. Acesse *Monitoring* > *Maps*.
2. Clique em *Import* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (veja abaixo).

5. Clique em *Import* no canto inferior direito do formulário de configuração.



Regras de importação:

Regra	Descrição
Update existing	Os mapas existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Create new	Novos mapas serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.

Se você desmarcar ambas as opções de mapas e marcar as respectivas opções para imagens, somente as imagens serão importadas. A importação de imagens está disponível apenas para usuários *Super admin*.

Não é possível revisar as alterações nos mapas de rede durante a importação. Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Warning:
Se uma imagem existente for substituída, isso afetará todos os mapas que usam essa imagem.

Formato de exportação

Exportar para 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.

Element	Type	Description
images		Elemento raiz para images.
name	string	Nome exclusivo da image.
imagetype	integer	Tipo de image. Valores possíveis: 1 - Image; 2 - Background.

Element	Type	Description
encodedImage	string	Image codificada em Base64.
maps		Elemento raiz para maps.

Maps

Element	Type	Description
name	string	Nome exclusivo do mapa.
width	integer	Largura do mapa, em pixels.
height	integer	Altura do mapa, em pixels.
label_type	integer	Tipo de rótulo do elemento do mapa. Valores possíveis: 0 - Rótulo; 1 - Endereço IP do host; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada.
label_location	integer	Localização padrão do rótulo do elemento do mapa. Valores possíveis: 0 - Inferior; 1 - Esquerda; 2 - Direita; 3 - Superior.
highlight	integer	Ativar realce de ícones 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 tiveram alteração recente de 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 separada de problemas reconhecidos e não reconhecidos.
severity_min	integer	Severidade mínima de trigger a ser exibida 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 mostrados) devido à manutenção do host. Valores possíveis: 0 - Não; 1 - Sim.
grid_size	integer	Tamanho da célula de uma grade do mapa em pixels. Compatível 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.

Element	Type	Description
grid_align	integer	Alinhar automaticamente os ícones na configuração do mapa. Valores possíveis: 0 - Sim; 1 - Não.
label_format	integer	Usar configuração avançada de rótulos. Valores possíveis: 0 - Não; 1 - Sim.
label_type_host	integer	Exibir rótulo como rótulo de host. Compatível se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 1 - Endereço IP do host; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada; 5 - Rótulo personalizado.
label_type_hostgroup	integer	Exibir rótulo como rótulo de grupo de hosts. Compatível 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. Compatível 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. Compatível 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. Compatível 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. Compatível se `label_type_host` estiver definido como 5.
label_string_hostgroup	string	Rótulo personalizado para elementos de grupo de hosts. Compatível se `label_type_hostgroup` estiver definido como 5.
label_string_trigger	string	Rótulo personalizado para elementos de trigger. Compatível se `label_type_trigger` estiver definido como 5.
label_string_map	string	Rótulo personalizado para elementos de mapa. Compatível se `label_type_map` estiver definido como 5.
label_string_image	string	Rótulo personalizado para elementos de imagem. Compatível se `label_type_image` estiver definido como 5.
expand_macros	integer	Expandir macros nos rótulos na configuração do mapa. Valores possíveis: 0 - Não; 1 - Sim.

Element	Type	Description
background		Elemento raiz para imagem de fundo (se houver). Compatível 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 pelos mapas ou por 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 map selements.
shapes		Elemento raiz para formas do mapa.
type	integer	Tipo de forma. Valores possíveis: 0 - Retângulo; 1 - Elipse.
x	integer	Coordenada X da forma em pixels.
y	integer	Coordenada 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 - Centro; 1 - Esquerda; 2 - Direita.
text_valign	integer	Alinhamento vertical do texto. Valores possíveis: 0 - Meio; 1 - Superior; 2 - Inferior.
border_type	integer	Tipo da borda da 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.

Element		Type	Description
	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	Coordenada X do ponto 1 da linha em pixels.
	y1	integer	Coordenada Y do ponto 1 da linha em pixels.
	x2	integer	Coordenada X do ponto 2 da linha em pixels.
	y2	integer	Coordenada 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).
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 caracteres, hexadecimal).
	label	string	Rótulo do link.
	selementid1	id	ID de um dos elementos a conectar.
	selementid2	id	ID do outro elemento a conectar.
linktriggers			Elemento raiz para indicadores de status de link.

Note:
Veja também: Objeto Map (consulte a propriedade relevante com o mesmo nome).

Elementos do mapa

Element	Type	Description
elementtype	integer	Tipo de elemento do mapa. Valores possíveis: 0 - Host; 1 - Map; 2 - Trigger; 3 - Grupo de hosts; 4 - Imagem.
elements		Elemento raiz para entidades do Zabbix (host, grupo de hosts, map, 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 o padrão do mapa; 0 - Inferior; 1 - Esquerda; 2 - Direita; 3 - Superior.
x	integer	Localização no eixo X.
y	integer	Localização no eixo Y.

Element	Type	Description
elementsubtype	integer	Subtipo do elemento. Compatível se `elementtype` estiver definido como 3. Valores possíveis: 0 - Grupo de hosts único; 1 - Todos os grupos de hosts.
areatype	integer	Tamanho da área. Compatível se `elementtype` estiver definido como 1. Valores possíveis: 0 - Igual ao mapa inteiro; 1 - Tamanho personalizado.
width	integer	Largura da área. Compatível se `areatype` estiver definido como 1.
height	integer	Altura da área. Compatível se `areatype` estiver definido como 1.
viewtype	integer	Algoritmo de posicionamento da área. Compatível se `elementsubtype` estiver definido como 1. Valores possíveis: 0 - Posicionar uniformemente na área.
use_iconmap	integer	Usar mapeamento de ícones para este elemento. Relevante somente 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 estado `OK`.
name	string	Nome exclusivo da imagem.
icon_on		Elemento raiz para a imagem a ser usada quando o elemento estiver no estado `Problem`.
name	string	Nome exclusivo da imagem.
icon_disabled		Elemento raiz para a imagem a ser usada quando o elemento estiver desativado.
name	string	Nome exclusivo da imagem.
icon_maintenance		Elemento raiz para a imagem a ser usada quando o elemento estiver em manutenção.
name	string	Nome exclusivo da imagem.
urls		Elemento raiz para URLs usadas por mapas ou por cada elemento do mapa.
name	string	Nome do link.
url	string	URL do link.
evaltype	integer	Tipo de avaliação para tags.
tags		Elemento raiz para tags de problema (para elementos de host e grupo de hosts). Se as tags forem fornecidas, somente problemas com essas tags serão exibidos no mapa.
tag	string	Nome da tag.
value	string	Valor da tag.
operator	integer	Operador.

Note:
Veja também: Objeto de elemento do mapa (consulte a propriedade relevante com o nome correspondente).

Indicadores de status do link do mapa

Element	Type	Description
drawtype	integer	Estilo do link quando o trigger está no estado 'Problem'. Valores possíveis: 0 - Linha; 2 - Linha em negrito; 3 - Linha pontilhada; 4 - Linha tracejada.
color	string	Cor do link (6 símbolos, hexadecimal) quando o trigger está no estado 'Problem'.
trigger		Elemento raiz do trigger usado para indicar o status do link.
description	string	Nome do trigger.

Element	Type	Description
expression	string	Expressão do trigger.
recovery_expression	string	Expressão de recuperação do trigger.

Note:
Veja também: Objeto de trigger do link do mapa (consulte a propriedade relevante com o nome correspondente).

Tipos de mídia

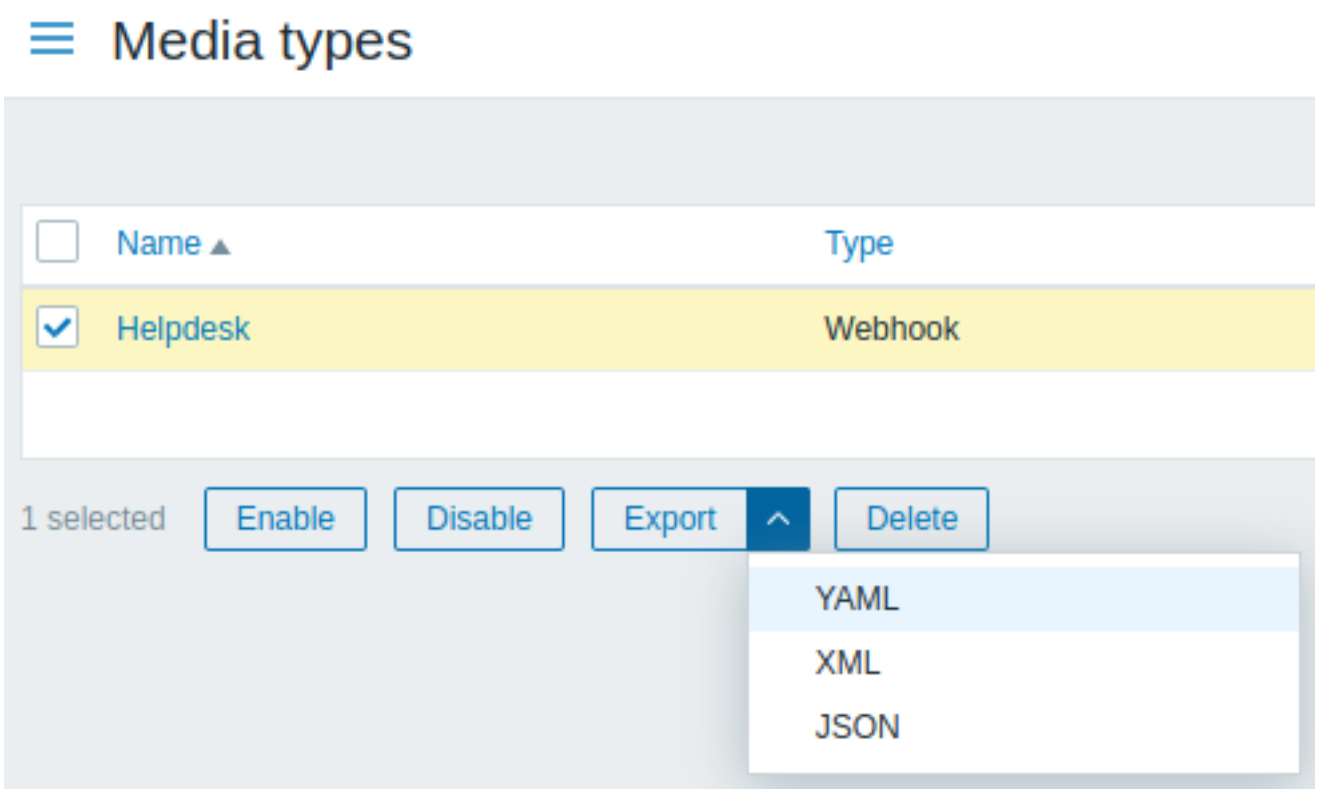
Visão geral

Os tipos de mídia são exportados com todos os objetos relacionados e relações entre objetos.

Exportando

Para exportar tipos de mídia, faça o seguinte:

1. Acesse *Alerts* > *Media types*.
2. Marque as caixas de seleção dos tipos de mídia a serem exportados.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



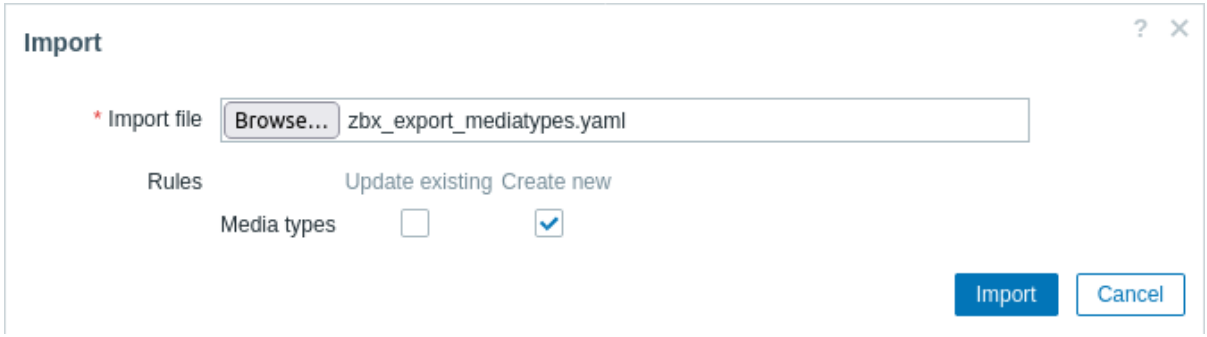
Dependendo do formato selecionado, os tipos de mídia são exportados para um arquivo local com um nome padrão:

- `zabbix_export_mediatypes.yaml` - na exportação YAML;
- `zabbix_export_mediatypes.xml` - na exportação XML;
- `zabbix_export_mediatypes.json` - na exportação JSON.

Importação

Para importar tipos de mídia, faça o seguinte:

1. Acesse *Alertas* > *Tipos de mídia*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (veja abaixo).
5. Clique em *Importar* no canto inferior direito do formulário de configuração.



Regras de importação:

Regra	Descrição
Atualizar existentes	Marque esta caixa de seleção para atualizar os tipos de mídia existentes usando os dados do arquivo de importação.
Criar novos	Marque esta caixa de seleção para criar novos tipos de mídia usando os dados do arquivo de importação. Se o arquivo de importação contiver um tipo de mídia que já exista (por exemplo, *E-mail*), um novo tipo não será criado, e o tipo de mídia existente também não será atualizado.

Não é possível revisar as alterações nos tipos de mídia durante a importação. 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 'Parâmetro "event_source" incorreto fornecido: "' + params.event_source + '".\nDeve
        }

        if (params.event_value !== '0' && params.event_value !== '1'
            && (params.event_source === '0' || params.event_source === '3')) {
            throw 'Parâmetro "event_value" incorreto fornecido: ' + params.event_value + '\nDeve ser 0
        }

        if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
            params.event_nseverity = '7';
        }

        if (params.event_value === '0') {
            params.event_nseverity = '6';
        }

        priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

        if (isNaN(priority) || priority < -2 || priority > 2) {
            throw '"priority" deve ser -2..2';
        }

        if (params.event_source === '0' && isNaN(params.triggerid)) {
            throw 'o campo "triggerid" não é um número';
```

```
}

if (isNaN(params.eventid)) {
    throw 'o campo "eventid" não é um número';
}

if (typeof params.message !== 'string' || params.message.trim() === '') {
    throw 'o campo "message" não pode estar vazio';
}

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 'o campo "retry" deve ser um número com valor mínimo de 30 se "priority" estiver
    }

    if (isNaN(params.expire) || params.expire > 10800) {
        throw 'o campo "expire" deve ser um número com valor máximo de 10800 se "priority" est
    }

    data.retry = params.retry;
    data.expire = params.expire;
}

data = JSON.stringify(data);
Zabbix.log(4, '[ Pushover Webhook ] Enviando solicitação: ' + params.endpoint + '\n' + data);

request.addHeader('Content-Type: application/json');
response = request.post(params.endpoint, data);

Zabbix.log(4, '[ Pushover Webhook ] Resposta recebida com código de status ' + request.getStat

if (response !== null) {
    try {
        response = JSON.parse(response);
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Falha ao analisar a resposta recebida do 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 'Erro desconhecido. Verifique o log de depuração para mais informações.';
    }
}
```

```
        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Falha na notificação do Pushover: ' + error);
        throw 'Falha na notificação do Pushover: ' + 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 a mídia Pushover ao usuário do Zabbix, adicione a chave do usuário no campo send to.
message_templates:
  - event_source: TRIGGERS
    operation_mode: PROBLEM
    subject: 'Problema: {EVENT.NAME}'
    message: |
      O problema começou em {EVENT.TIME} em {EVENT.DATE}
      Nome do problema: {EVENT.NAME}
      host: {HOST.NAME}
      Severidade: {EVENT.SEVERITY}
      Dados operacionais: {EVENT.OPDATA}
      ID original do problema: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Resolvido em {EVENT.DURATION}: {EVENT.NAME}'
    message: |
      O problema foi resolvido em {EVENT.RECOVERY.TIME} em {EVENT.RECOVERY.DATE}
      Nome do problema: {EVENT.NAME}
      Duração do problema: {EVENT.DURATION}
      host: {HOST.NAME}
      Severidade: {EVENT.SEVERITY}
      ID original do problema: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Problema atualizado em {EVENT.AGE}: {EVENT.NAME}'
    message: |
      {USER.FULLNAME} {EVENT.UPDATE.ACTION} o problema em {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
      {EVENT.UPDATE.MESSAGE}

      O estado atual do problema é {EVENT.STATUS}, a idade é {EVENT.AGE}, reconhecido: {EVENT.ACK.ST
  - 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.

Element	Type	Description
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 time suffix) entre as tentativas de repetição. Valores possíveis:[1] 0-60s (padrão: 10s).
description	string	Descrição do tipo de mídia.
message_templates		Elemento raiz para os modelos de mensagem do tipo de mídia.
event_source	string	(obrigatório) Origem do evento. Valores possíveis:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Modo de operação. Valores possíveis:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Assunto da mensagem.
message	string	Corpo da mensagem.

Note:
Veja também: Media type object (consulte a propriedade relevante com o nome correspondente).

Email

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Email*.

Element	Type	Description
provider	string	Provedor de email.
smtp_server	string	Servidor SMTP.
smtp_port	integer	Porta do servidor SMTP para conexão. Default: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	Endereço de email a partir do qual as notificações serão enviadas.
smtp_security	string	Nível de segurança da conexão SMTP a ser usado. Valores possíveis:[1] NONE (0, default), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Verificação de host SSL para SMTP. Valores possíveis:[1] NO (0, default), YES (1).
smtp_verify_peer	string	Verificação de peer SSL para SMTP. Valores possíveis:[1] NO (0, default), YES (1).
smtp_authentication	string	Método de autenticação SMTP a ser usado. Valores possíveis:[1] NONE (0, default), PASSWORD (1), OAUTH (2).
username	string	Nome de usuário.
password	string	Senha de autenticação.
redirection_url	string	URL do frontend do Zabbix para redirecionar de volta a autorização OAuth.
client_id	string	O 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, default).

Note:
Veja também: Media type object (consulte a propriedade relevante com o nome correspondente).

SMS

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *SMS*.

Element	Type	Description
gsm_modem	string	(required) Nome do dispositivo serial do modem GSM.

Note:
Veja também: Media type object (consulte a propriedade relevante com o nome correspondente).

Script

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Script*.

Element	Type	Description
script name	string	(required) Nome do script.
parameters		Elemento raiz para os parâmetros do script.
sortorder	string	(required) Ordem dos parâmetros do script passados ao script como argumentos de linha de comando.
value	string	Valor do parâmetro do script.

Note:
Veja também: Objeto de tipo de mídia (consulte a propriedade relevante com um nome correspondente).

Webhook

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Webhook*.

Element	Type	Description
script	string	Script.
timeout	string	Intervalo de timeout da requisição HTTP do script JavaScript. Valores possíveis:[1] 1-60s (padrão: 30s).
process_tags	string	Indica 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.*} tiver sido 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 os parâmetros do tipo de mídia webhook.
name	string	(obrigatório) Nome do parâmetro do webhook.
value	string	Valor do parâmetro do webhook.

Note:
Veja também: Media type object (consulte a propriedade relevante com o nome correspondente).

Notas de rodapé

[1] Valores inteiros da API entre parênteses, por exemplo, ENABLED (0), são mencionados apenas para referência. Para mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

Painéis

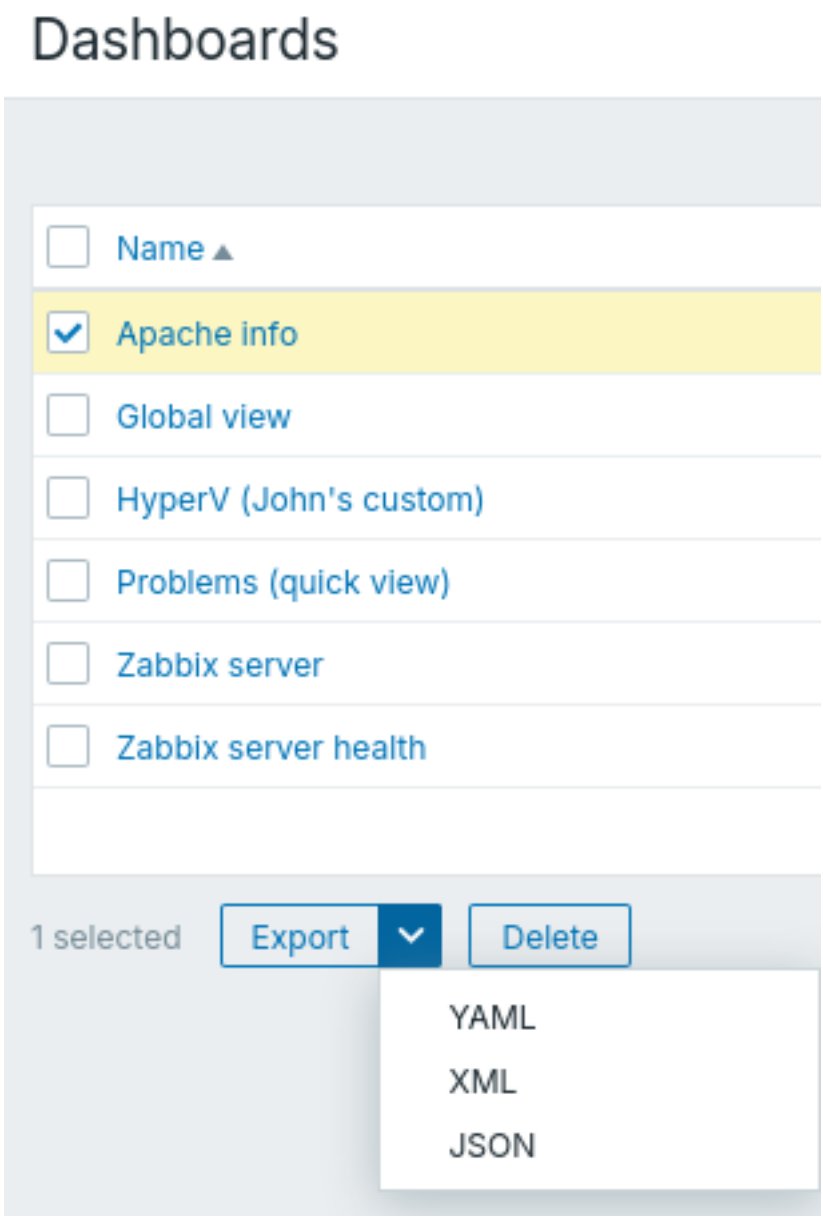
Visão geral

Os dashboards são exportados com todos os objetos relacionados e as relações entre os objetos.

Exportando

Para exportar dashboards, faça o seguinte:

1. Acesse *Dashboards* > *All dashboards*.
2. Marque as caixas de seleção dos dashboards que deseja exportar.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



Dependendo do formato selecionado, os dashboards são exportados para um arquivo local com um nome padrão:

- `zbx_export_dashboards.yaml` - na exportação YAML;
- `zbx_export_dashboards.xml` - na exportação XML;
- `zbx_export_dashboards.json` - na exportação JSON.

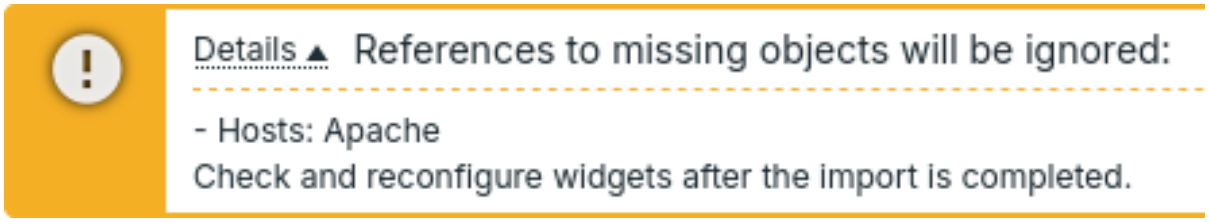
Importação

Para importar dashboards, faça o seguinte:

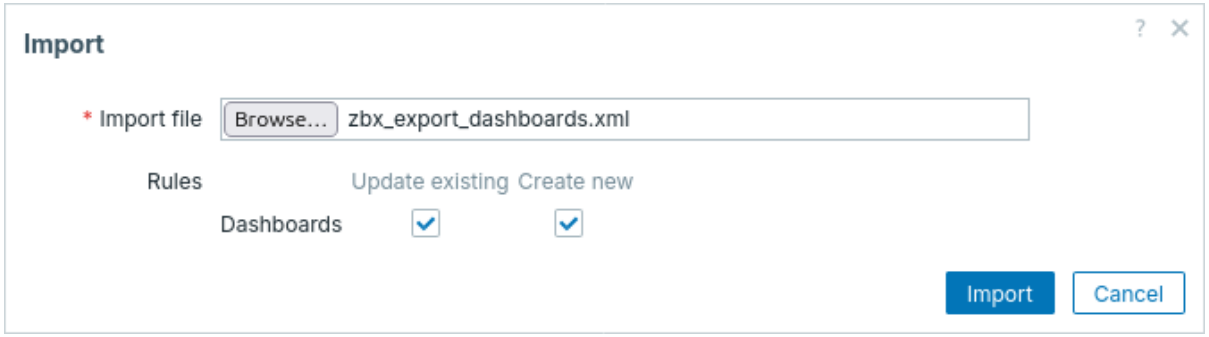
1. Acesse *Dashboards* > *Todos os dashboards*.
2. Clique em *Importar* no canto superior direito. Essa opção fica desabilitada se o usuário não tiver permissão para *Criar e editar dashboards*.

3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (consulte abaixo).
5. Clique em *Importar* no canto inferior direito do formulário de configuração.

Se o arquivo de importação fizer referência a objetos que não existem no sistema de destino, uma mensagem de alerta será exibida. Isso é esperado para widgets que fazem referência a hosts, itens ou outros objetos específicos. Nesses casos, o Zabbix permite importar o dashboard, limpa as referências ausentes e marca o widget como não totalmente configurado quando os dados são obrigatórios ou remove os critérios de filtragem quando os campos são usados apenas para filtragem.



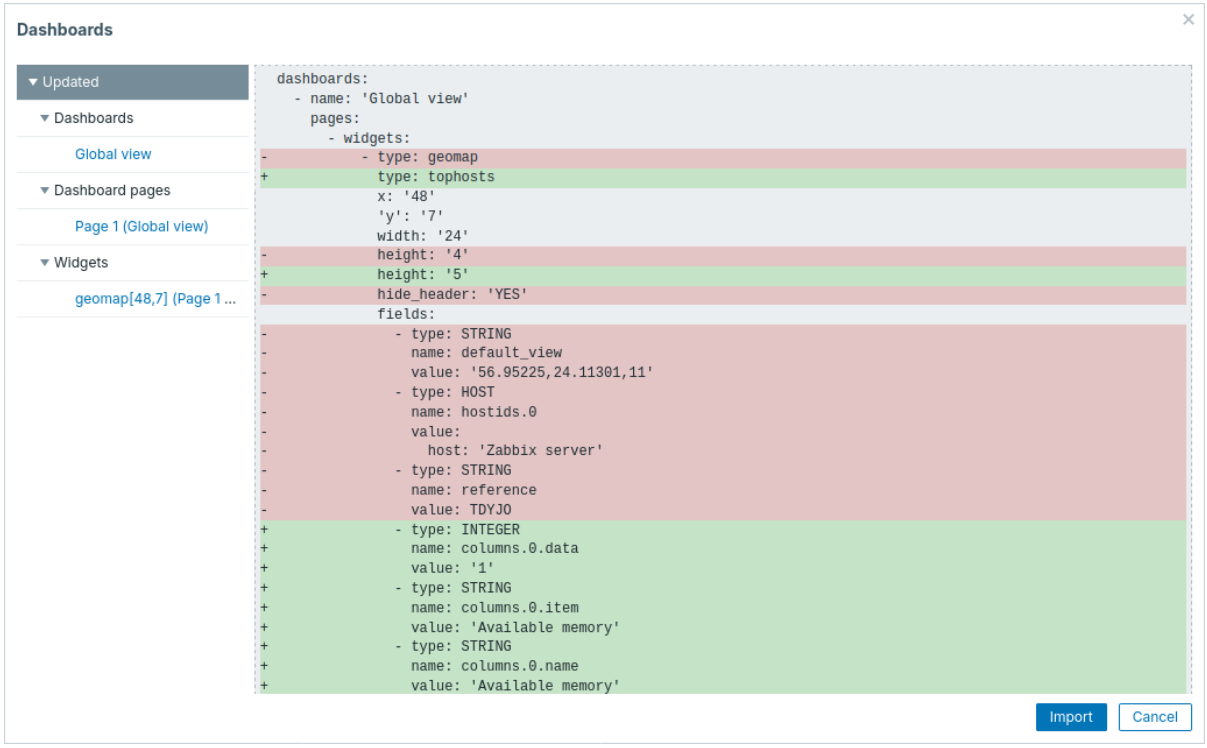
Clicar novamente no botão *Importar* adicionará e/ou atualizará o dashboard.



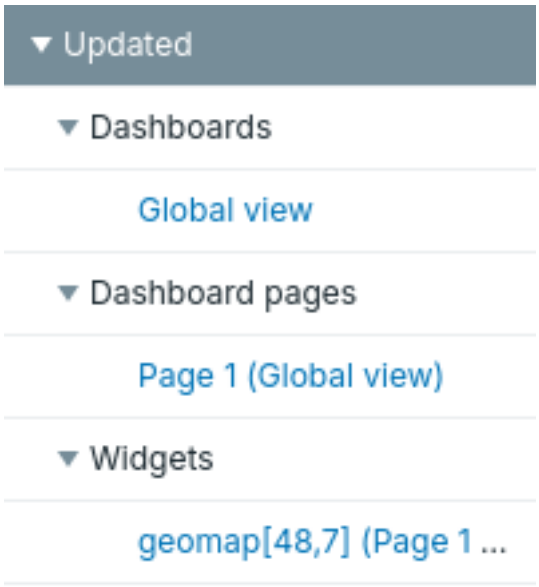
Regras de importação:

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.

Na tela seguinte, você poderá visualizar o conteúdo dos dashboards que estão sendo importados. Se este for um novo dashboard, todos os elementos serão listados em verde. Se um dashboard existente estiver sendo atualizado, os novos elementos do dashboard serão destacados em verde; os elementos removidos do dashboard 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 *Atualizados* destaca todas as alterações feitas nos elementos existentes do dashboard. A seção *Adicionados* lista os novos elementos do dashboard. Os elementos de cada seção são agrupados por tipo de elemento; clique na seta cinza para expandir ou recolher o grupo de elementos.



Revise as alterações do dashboard e clique em *Importar* para realizar a importação do dashboard. Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Formato de exportação

Exportar para YAML:

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: 'Top hosts by CPU utilization'
```

```
            width: '25'
            height: '4'
            fields:
              - type: INTEGER
                name: column
                value: '1'
              - type: INTEGER
                name: columns.0.data
                value: '2'
              - type: STRING
                name: columns.0.item
                value: 'System name'
              - type: STRING
                name: columns.0.name
                value: 'Host name'
              - type: STRING
                name: columns.1.base_color
                value: 4CAF50
              - type: INTEGER
                name: columns.1.data
                value: '1'
              - type: INTEGER
                name: columns.1.display
                value: '3'
              - type: STRING
                name: columns.1.item
                value: 'CPU utilization'
              - type: STRING
                name: columns.1.max
                value: '100'
              - type: STRING
                name: columns.1.min
                value: '0'
              - type: STRING
                name: columns.1.name
                value: Utilization
              - type: INTEGER
                name: columns.2.data
                value: '1'
              - type: STRING
                name: columns.2.item
                value: 'Load average (1m avg)'
              - type: STRING
                name: columns.2.name
                value: '1m avg'
              - type: INTEGER
                name: columns.3.data
                value: '1'
              - type: STRING
                name: columns.3.item
                value: 'Load average (5m avg)'
              - type: STRING
                name: columns.3.name
                value: '5m avg'
              - type: INTEGER
                name: columns.4.data
                value: '1'
              - type: STRING
                name: columns.4.item
                value: 'Load average (15m avg)'
              - type: STRING
                name: columns.4.name
```

```
            value: '15m avg'
          - type: INTEGER
            name: columns.5.data
            value: '1'
          - type: INTEGER
            name: columns.5.decimal_places
            value: '0'
          - type: STRING
            name: columns.5.item
            value: 'Number of processes'
          - type: STRING
            name: columns.5.name
            value: Processes
          - type: STRING
            name: columnsthresholds.1.color.0
            value: FFFF00
          - type: STRING
            name: columnsthresholds.1.color.1
            value: FF8000
          - type: STRING
            name: columnsthresholds.1.color.2
            value: FF4000
          - type: STRING
            name: columnsthresholds.1.threshold.0
            value: '50'
          - type: STRING
            name: columnsthresholds.1.threshold.1
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
      - type: hostavail
        'y': '4'
        width: '25'
        fields:
          - type: INTEGER
            name: interface_type.0
            value: '1'
      - type: problems
        name: 'Problemas atuais'
        'y': '6'
        width: '48'
        height: '5'
        fields:
          - type: STRING
            name: reference
            value: BRUQE
          - type: INTEGER
            name: show
            value: '3'
          - type: INTEGER
            name: show_tags
            value: '3'
      - type: systeminfo
        x: '25'
        width: '23'
        height: '4'
      - type: problemsbysv
        name: 'Problemas por severidade'
```

```
            x: '25'
            'y': '4'
            width: '23'
            fields:
              - type: STRING
                name: reference
                value: LBSVB
              - type: INTEGER
                name: show_type
                value: '1'
          - type: gauge
            name: 'Zabbix server'
            x: '48'
            width: '12'
            height: '4'
            hide_header: 'YES'
            fields:
              - type: INTEGER
                name: angle
                value: '270'
              - type: INTEGER
                name: desc_size
                value: '8'
              - type: ITEM
                name: itemid.0
                value:
                  host: 'Zabbix server'
                  key: vm.memory.utilization
              - type: STRING
                name: max
                value: '100'
              - type: STRING
                name: min
                value: '0'
              - type: INTEGER
                name: scale_size
                value: '11'
              - type: INTEGER
                name: show.0
                value: '1'
              - type: INTEGER
                name: show.1
                value: '2'
              - type: INTEGER
                name: show.2
                value: '3'
              - type: INTEGER
                name: show.3
                value: '4'
              - type: INTEGER
                name: show.4
                value: '5'
              - type: STRING
                name: thresholds.0.color
                value: 42A5F5
              - type: STRING
                name: thresholds.0.threshold
                value: '0'
              - type: STRING
                name: thresholds.1.color
                value: 9CCC65
              - type: STRING
```

```
            name: thresholds.1.threshold
            value: '20'
          - type: STRING
            name: thresholds.2.color
            value: FFEE58
          - type: STRING
            name: thresholds.2.threshold
            value: '40'
          - type: STRING
            name: thresholds.3.color
            value: FFA726
          - type: STRING
            name: thresholds.3.threshold
            value: '60'
          - type: STRING
            name: thresholds.4.color
            value: FF7043
          - type: STRING
            name: thresholds.4.threshold
            value: '80'
          - type: INTEGER
            name: th_arc_size
            value: '9'
          - type: INTEGER
            name: th_show_arc
            value: '1'
          - type: INTEGER
            name: th_show_labels
            value: '1'
          - type: INTEGER
            name: units_bold
            value: '1'
          - type: INTEGER
            name: units_size
            value: '24'
          - type: INTEGER
            name: value_arc_size
            value: '30'
          - type: INTEGER
            name: value_bold
            value: '1'
          - type: INTEGER
            name: value_size
            value: '24'
      - type: svggraph
        x: '48'
        'y': '4'
        width: '24'
        height: '3'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: axisx
            value: '0'
          - type: STRING
            name: ds.0.color
            value: AB47BC
          - type: INTEGER
            name: ds.0.fill
            value: '0'
          - type: STRING
            name: ds.0.hosts.0
```

```
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
      - type: geomap
        x: '48'
        'y': '7'
        width: '24'
        height: '4'
        hide_header: 'YES'
        fields:
          - type: STRING
            name: default_view
            value: '56.95225,24.11301,11'
          - type: HOST
            name: hostids.0
            value:
              host: 'Zabbix server'
          - type: STRING
            name: reference
            value: TDYJO
      - type: clock
        name: 'Hora local'
        x: '60'
        width: '12'
        hide_header: 'YES'
```

```
      fields:
        - type: INTEGER
          name: clock_type
          value: '1'
        - type: INTEGER
          name: show.0
          value: '1'
        - type: INTEGER
          name: show.1
          value: '2'
        - type: INTEGER
          name: show.2
          value: '3'
        - type: INTEGER
          name: time_sec
          value: '0'
        - type: INTEGER
          name: tzone_format
          value: '1'
    - type: item
      name: 'Desempenho do server'
      x: '60'
      'y': '2'
      width: '12'
      hide_header: 'YES'
      fields:
        - type: INTEGER
          name: aggregate_function
          value: '2'
        - type: INTEGER
          name: decimal_size
          value: '32'
        - type: STRING
          name: description
          value: 'Values per second'
        - type: INTEGER
          name: desc_h_pos
          value: '0'
        - type: INTEGER
          name: desc_v_pos
          value: '0'
        - type: ITEM
          name: itemid.0
          value:
            host: 'Zabbix server'
            key: 'zabbix[wcache,values]'
        - type: INTEGER
          name: show.0
          value: '1'
        - type: INTEGER
          name: show.1
          value: '2'
        - type: INTEGER
          name: show.2
          value: '5'
        - type: INTEGER
          name: sparkline.fill
          value: '2'
        - type: INTEGER
          name: sparkline.width
          value: '2'
        - type: STRING
```

```
      name: thresholds.0.color
      value: DCEDC8
    - type: STRING
      name: thresholds.0.threshold
      value: '20'
    - type: STRING
      name: thresholds.1.color
      value: FFF9C4
    - type: STRING
      name: thresholds.1.threshold
      value: '40'
    - type: STRING
      name: thresholds.2.color
      value: FFCCBC
      value: '60'
    - type: INTEGER
      name: units_bold
      value: '0'
    - type: INTEGER
      name: units_show
      value: '0'
    - type: INTEGER
      name: value_h_pos
      value: '0'
    - type: INTEGER
      name: value_size
      value: '32'
```

Elementos exportados

Os elementos exportados são explicados na tabela abaixo.

Element		Type	Description
name		string	(required) Nome do dashboard.
pages			Elemento raiz para as páginas do dashboard
	widgets		Elemento raiz para as páginas do dashboard

Note:
Veja também: Dashboard object (consulte a propriedade relevante com o nome correspondente).

Widget do dashboard

Element	Type	Description
type	string	(obrigatório) Tipo do widget do dashboard. Valores possíveis: actionlog - Log de ações; clock - Relógio; discovery - Status de descoberta; favgraphs - Gráficos favoritos; favmaps - Mapas favoritos; gauge - Medidor; geomap - Geomap; graph - Gráfico (clássico); graphprototype - Protótipo de gráfico; honeycomb - Honeycomb; 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; scatterplot - Gráfico de dispersão; 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.
x	integer	Uma posição horizontal a partir do lado esquerdo do dashboard. Valores possíveis:[1] 0-71.
y	integer	Uma posição vertical a partir do topo do dashboard. Valores possíveis:[1] 0-63.
name	string	Nome personalizado do widget.
width	integer	A largura do widget. Valores possíveis:[1] 1-72.
height	integer	A 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.

Note:
Veja também: Dashboard widget (consulte a propriedade relevante com o nome correspondente).

Campo de widget do dashboard

Element	Type	Description
type	string	(required) Tipo de campo do widget. Valores possíveis:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(required) Nome do campo do widget.
value	mixed	(required) Valor do campo do widget, dependendo do tipo de campo.
host	string	Host do item.
key	string	Chave do item.

Note:
Veja também: Campo de widget do dashboard (consulte a propriedade relevante com o nome correspondente).

Notas de rodapé

[1] Valores inteiros da API entre parênteses, por exemplo, ENABLED (0), são mencionados apenas para referência. Para mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

Expressões regulares

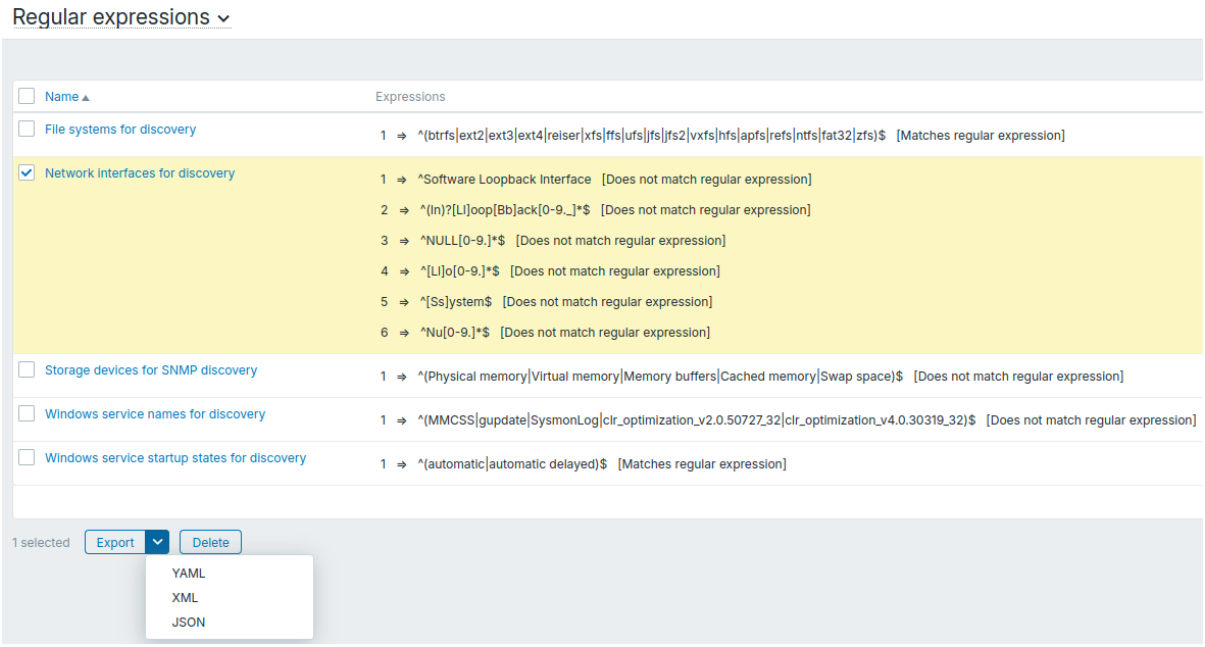
Visão geral

As expressões regulares são exportadas com todos os objetos relacionados e as relações entre objetos.

Exportando

Para exportar expressões regulares, faça o seguinte:

1. Vá para *Administration* > *General* > *Regular expressions*.
2. Marque as caixas de seleção das expressões regulares a serem exportadas.
3. Clique em *Export* abaixo da lista para exportar no formato YAML padrão, ou clique na seta para selecionar um formato de exportação diferente.



Dependendo do formato selecionado, as expressões regulares são exportadas para um arquivo local com um nome padrão:

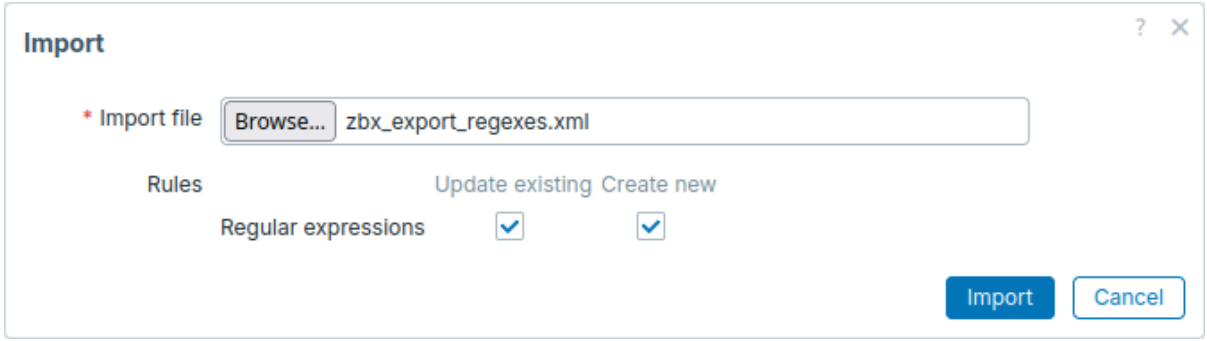
- `zbx_export_regexes.yaml` - na exportação YAML;
- `zbx_export_regexes.xml` - na exportação XML;
- `zbx_export_regexes.json` - na exportação JSON.

Importação

Para importar expressões regulares, faça o seguinte:

1. Acesse *Administration* > *General* > *Regular expressions*.

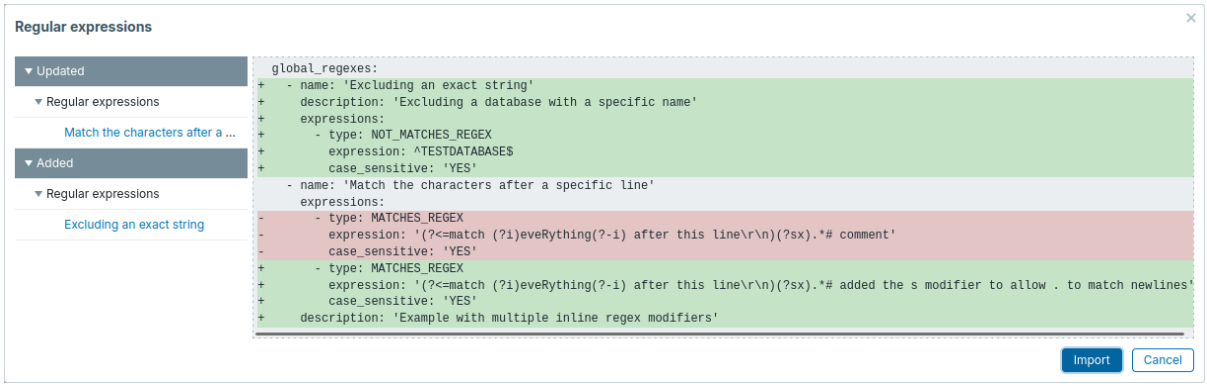
2. Clique em *Import* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as caixas de seleção das regras de importação que deseja aplicar (veja abaixo).
5. Clique em *Import* no canto inferior direito do formulário de configuração.



Regras de importação:

Regra	Descrição
Update existing	As expressões regulares existentes serão atualizadas usando os dados do arquivo de importação, substituindo completamente as expressões existentes com o mesmo nome. Caso contrário, elas não serão atualizadas.
Create new	Novas expressões regulares serão criadas usando os dados do arquivo de importação. Caso contrário, elas não serão criadas.

Na tela seguinte, você poderá visualizar o conteúdo das expressões regulares que estão sendo importadas. Se esta for uma nova expressão regular, todos os dados serão listados em verde. Ao atualizar uma expressão regular existente, os novos dados da expressão regular serão destacados em verde; os dados removidos serão destacados em vermelho; os dados inalterados serão exibidos sobre um fundo cinza.



O menu à esquerda pode ser usado para navegar pela lista de alterações. A seção *Updated* destaca todas as alterações feitas nas expressões regulares existentes. A seção *Added* lista as novas expressões regulares. Clique na seta para expandir ou recolher um grupo de expressões regulares.

Revise as alterações nas expressões regulares e clique em *Import* para realizar a importação das expressões regulares. Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Formato de exportação

Exportar para YAML:

```yaml
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'File systems for discovery'
      description: 'Expressão regular global usada para descobrir e identificar sistemas de arquivos no ho
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Storage devices for SNMP discovery'
      expressions:
        - type: NOT_MATCHES_REGEX
```

```
        expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
        case_sensitive: 'YES'
```

Elementos exportados

Elementos exportados de expressões regulares:

Element	Type	Description
global_regexes	object	Elemento raiz para expressões regulares.
name	string	(obrigatório) Nome da expressão regular.
description	text	Descrição da expressão regular.
expressions	object	Elemento raiz para expressões regulares.
type	string	Tipo da expressão regular.
expression	string	Expressão regular.
exp_delimiter	string	Delimitador da expressão regular; presente quando o tipo é *Contém qualquer substring da lista*.
case_sensitive	string	Opção de diferenciação entre maiúsculas e minúsculas para a expressão regular.

Note:
Veja também: Objeto de expressão regular (consulte a propriedade relevante com o nome correspondente).

Expressões regulares

Visão geral Perl Compatible Regular Expressions (PCRE, PCRE2) são suportadas no Zabbix.

Há duas maneiras de usar expressões regulares no Zabbix:

- Inserir manualmente uma expressão regular
- Usar uma expressão regular global criada no Zabbix

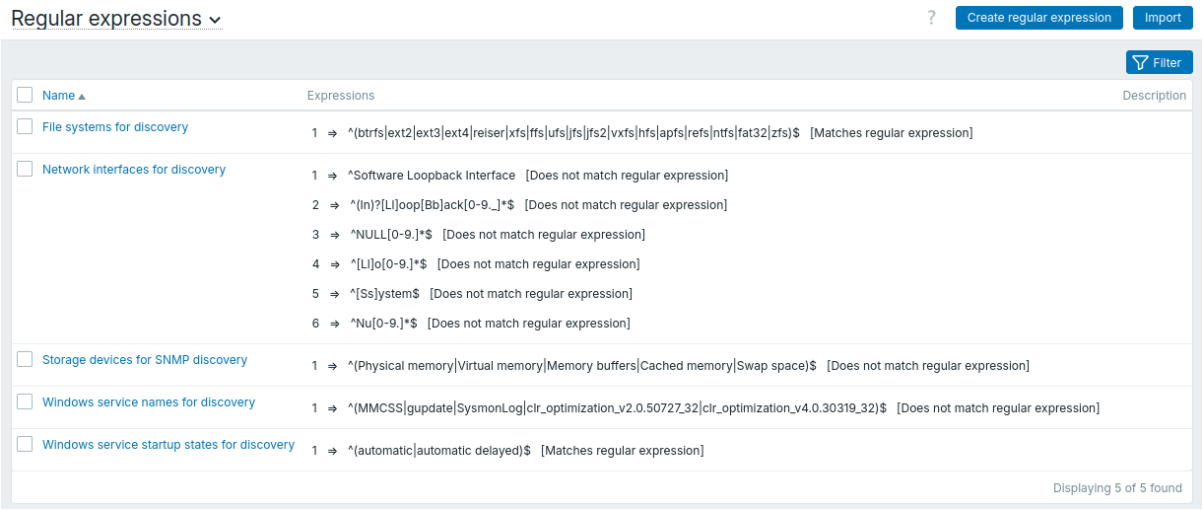
Expressões regulares Você pode inserir manualmente uma expressão regular em locais suportados.
 Observe que a expressão não deve começar com @, pois esse símbolo é usado no Zabbix para referenciar expressões regulares globais.

Attention:
É possível ficar sem memória de pilha ao usar expressões regulares. Consulte o PCRE PERFORMANCE para obter mais informações.

Observe que, na correspondência em várias linhas, os âncoras ^ e $ correspondem ao início e ao fim de cada linha, respectivamente, em vez do início e do fim de toda a string.

Veja também exemplos de escape correto em vários contextos.

Expressões regulares globais O Zabbix fornece um editor avançado no frontend para criar e testar expressões regulares complexas.
 Para acessá-lo, vá para *Administration* > *General* > *Regular expressions*.
 O Zabbix inclui várias expressões regulares globais predefinidas em seu conjunto de dados padrão.

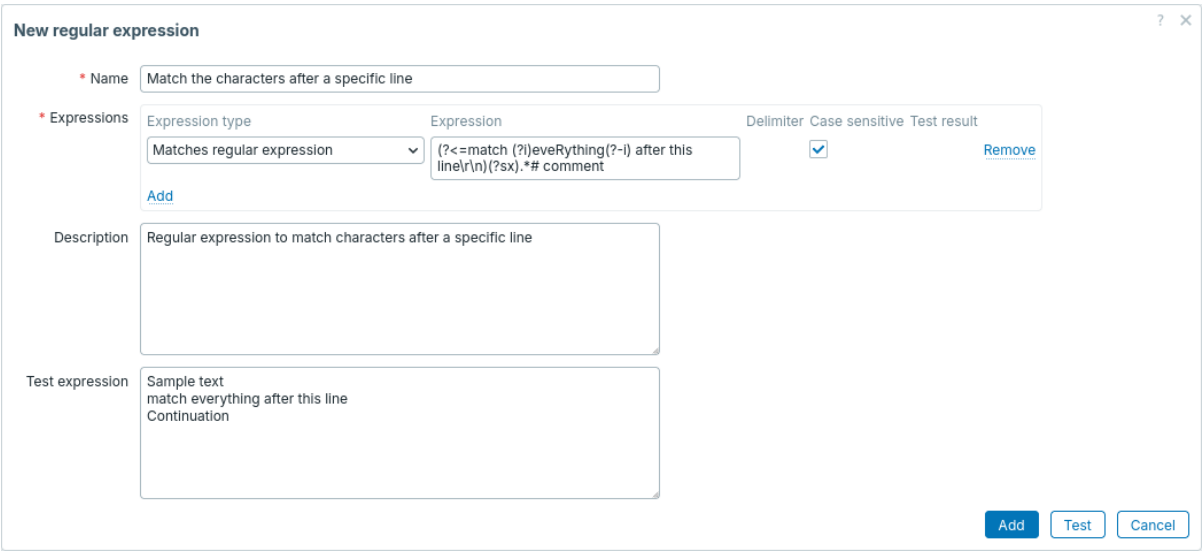


Você pode filtrar a lista de expressões regulares por nome e descrição para encontrar as expressões de que precisa.

Configuração

Para criar uma expressão regular global:

1. Clique em *Create regular expression*.
2. Insira os parâmetros da expressão regular no formulário.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parameter	Description
Name	Defina o nome da expressão regular. Quaisquer caracteres Unicode são permitidos.
Expressions	Clique em *Add* na seção *Expressions* para adicionar uma nova subexpressão.
Expression type	Selecione o tipo de expressão: **Contains string** - corresponde à substring. **Contains any substring from list** - corresponde a qualquer substring de uma lista delimitada. A lista delimitada inclui uma vírgula `,`, um ponto `.` ou uma barra `/`. **Does not contain string** - corresponde a qualquer string, exceto a substring. **Matches regular expression** - corresponde à expressão regular. **Does not match regular expression** - não corresponde à expressão regular.
Expression	Insira a substring/expressão regular. Comprimento máximo: 2048 caracteres.
Delimiter	Uma vírgula `,`, um ponto `.` ou uma barra `/` para separar strings de texto em uma expressão regular. Este parâmetro fica ativo somente quando o tipo de expressão *Contains any substring from list* está selecionado.
Case sensitive	Uma caixa de seleção para especificar se a expressão regular diferencia maiúsculas de minúsculas.
Description	Insira uma descrição para a expressão regular.

Parameter	Description
Test expression	Um valor a ser usado para testar a expressão regular.

Uma barra / na expressão é tratada literalmente, e não como um delimitador. Isso permite salvar expressões que contêm uma barra sem erros.

Attention:
Um nome de expressão regular personalizada no Zabbix pode conter vírgulas, espaços etc. Nos casos em que isso possa causar uma interpretação incorreta ao fazer referência a ela (por exemplo, uma vírgula no parâmetro de uma chave de item), toda a referência pode ser colocada entre aspas, desta forma: *"@My custom regexp for purpose1, purpose2"*.
Os nomes de expressões regulares não devem ser colocados entre aspas em outros locais (por exemplo, nas propriedades de regras de LLD).

Depois que uma expressão regular é criada, ela pode ser usada em vários locais do frontend fazendo referência ao seu nome, precedido por `@`, por exemplo, `@mycustomregexp`.

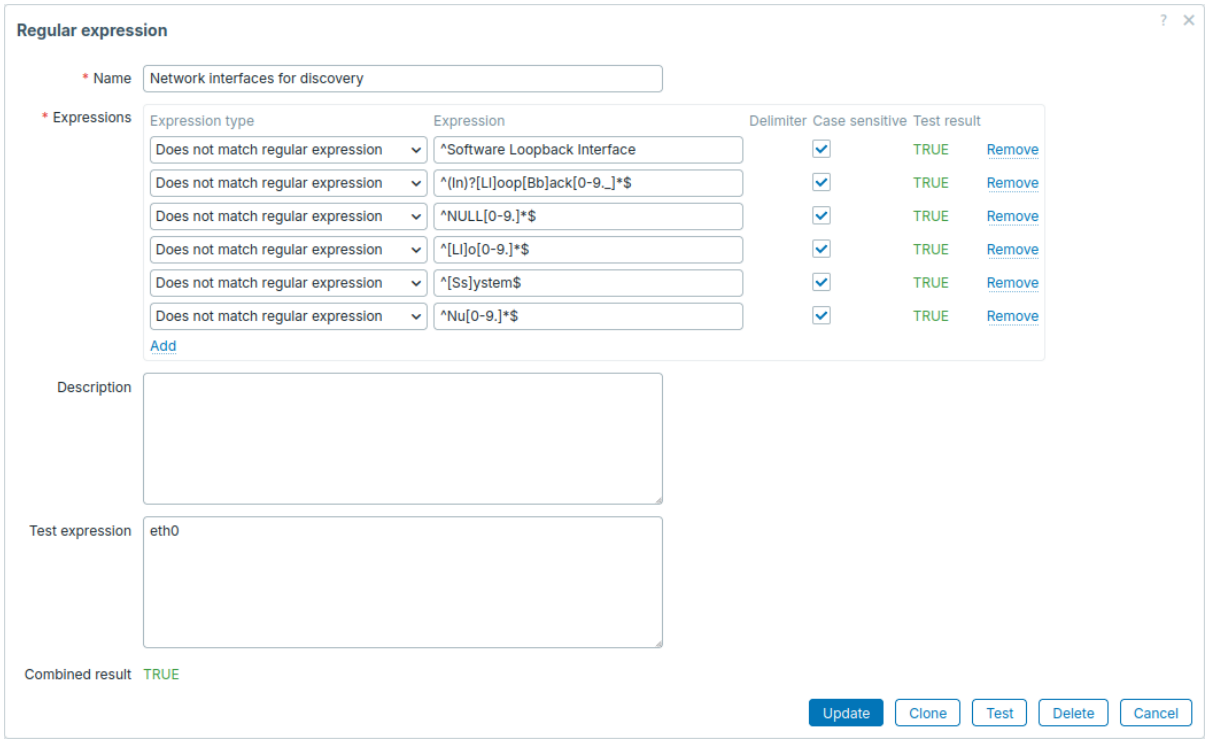
Botões do formulário

Os botões na parte inferior do formulário permitem que você execute várias operações:

Button	Description
Add	Adicionar uma expressão regular. Este botão está disponível apenas para novas expressões regulares.
Update	Atualizar as propriedades de uma expressão regular.
Clone	Criar outra expressão regular com base nas propriedades da expressão regular atual.
Test	Testar uma expressão regular e suas subexpressões em relação a um valor fornecido.
Delete	Excluir a expressão regular.
Cancel	Cancelar a edição das propriedades da expressão regular.

Testando uma expressão regular

Você pode testar a expressão regular e suas subexpressões inserindo um valor no campo *Test expression* e clicando no botão *Test*.



Na coluna *Test result*, o status de cada subexpressão é exibido. O campo *Combined result* exibe o status geral da expressão regular. Se várias subexpressões estiverem definidas, o Zabbix usa o operador lógico AND para calcular o *Combined result*; isto é, se pelo menos um resultado for `False`, o *Combined result* também será `False`.

Expressões regulares globais padrão

O conjunto de dados padrão inclui as seguintes expressões regulares globais predefinidas:

Name	Expression	Matches
File systems for discovery	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	”btrfs” ou ”ext2” ou ”ext3” ou ”ext4” ou ”reiser” ou ”xfs” ou ”ffs” ou ”ufs” ou ”jfs” ou ”jfs2” ou ”vxfs” ou ”hfs” ou ”apfs” ou ”refs” ou ”ntfs” ou ”fat32” ou ”zfs”.
Network interfaces for discovery	`^Software Loopback Interface`	Strings que começam com ”Software Loopback Interface”.
	`^lo$`	”lo”.
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Strings que começam opcionalmente com ”In”, seguidas por ”L” ou ”l”, seguidas por ”oop”, seguidas por ”B” ou ”b”, seguidas por ”ack” e, opcionalmente, seguidas por qualquer número de dígitos, pontos ou sublinhados.
	`^NULL[0-9.]*$`	Strings que começam com ”NULL” e são opcionalmente seguidas por qualquer número de dígitos ou pontos.
	`^[Ll]o[0-9.]*$`	Strings que começam com ”Lo” ou ”lo” e são opcionalmente seguidas por qualquer número de dígitos ou pontos.
	`^[Ss]ystem$`	”System” ou ”system”.
	`^Nu[0-9.]*$`	Strings que começam com ”Nu” e são opcionalmente seguidas por qualquer número de dígitos ou pontos.
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” ou ”Virtual memory” ou ”Memory buffers” ou ”Cached memory” ou ”Swap space”.
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS” ou ”gupdate” ou ”SysmonLog” ou strings que correspondem a ”clr_optimization_v2.0.50727_32” ou ”clr_optimization_v4.0.30319_32”, em que cada ponto pode ser substituído por qualquer caractere, exceto uma nova linha.
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	”automatic” ou ”automatic delayed”.

Importando/exportando uma expressão regular

Você pode importar ou exportar expressões regulares globais conforme descrito na página de configuração de exportação/importação de Regular expressions.

Exemplos Exemplo de exclusão de uma string exata

Use a seguinte expressão na descoberta de baixo nível para descobrir bancos de dados, exceto um banco de dados com um nome específico:

```
^TESTDATABASE$
```

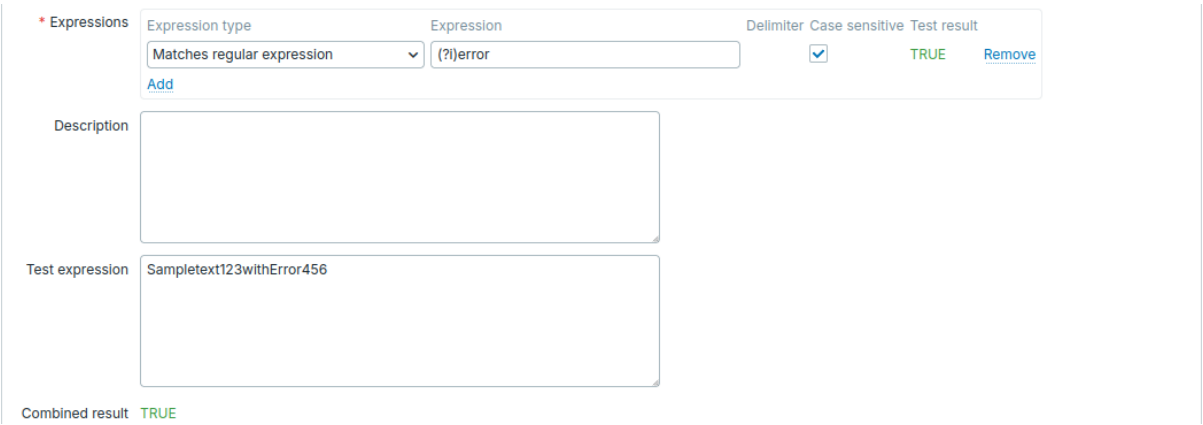


Tanto *Test result* quanto *Combined result* exibem `FALSE`: o nome corresponde a "TESTDATABASE", portanto a condição *Does not match regular expression* não é satisfeita.

Exemplo com um modificador de regex inline para uma correspondência sem distinção entre maiúsculas e minúsculas

Use a seguinte expressão regular, que inclui o modificador inline `(?i)` para realizar uma correspondência sem distinção entre maiúsculas e minúsculas dos caracteres "error":

```
(?i)error
```



Tanto *Test result* quanto *Combined result* exibem `TRUE`: os caracteres "error" são correspondidos mesmo com *Case sensitive* selecionado, porque o modificador inline `(?i)` torna a correspondência sem distinção entre maiúsculas e minúsculas.

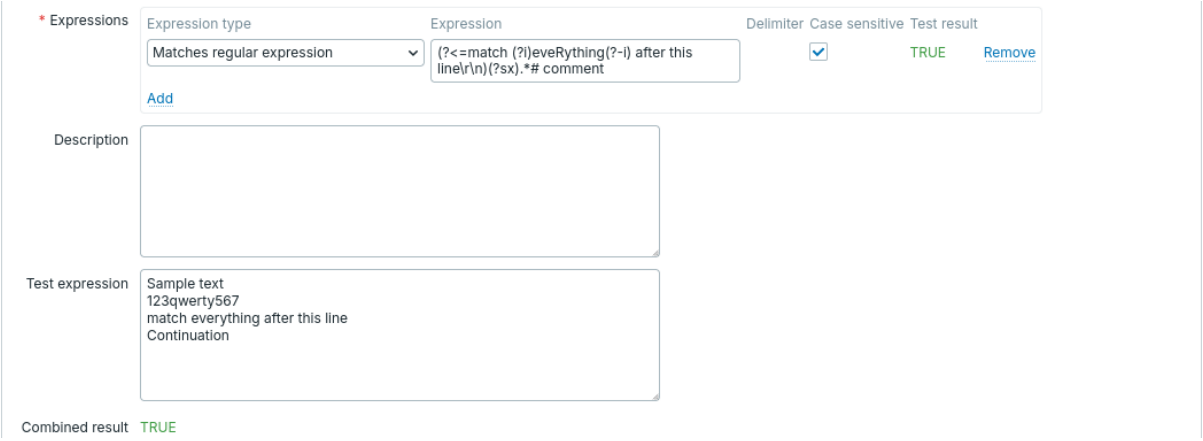
Exemplo com vários modificadores inline de regex

Use a seguinte expressão regular, que inclui vários modificadores inline, para corresponder aos caracteres após uma linha específica:

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

onde:

- `(?i)` - ativa a correspondência sem distinção entre maiúsculas e minúsculas;
- `(?-i)` - desativa a correspondência sem distinção entre maiúsculas e minúsculas;
- `(?s)` - ativa o modo dotall, permitindo que `.` corresponda a caracteres de nova linha;
- `(?x)` - ativa o modo estendido, permitindo que espaços em branco e comentários na expressão regular sejam ignorados.



Tanto *Test result* quanto *Combined result* exibem TRUE: os caracteres que seguem a linha específica são correspondidos.

Attention:
O modificador g não pode ser especificado inline. A lista de modificadores disponíveis pode ser encontrada na página man do pcresyntax. Para mais informações sobre a sintaxe PCRE, consulte a documentação HTML do PCRE.

Suporte a expressão regular por local

Location		Regular expression	Global regular expression	Multiline matching	Comments
Itens do Agent					
	eventlog[]	Yes	Yes	Yes	Parâmetros `regexp`, `severity`, `source`, `eventid`
	eventlog.count[]				Parâmetros `regexp`, `severity`, `source`, `eventid`
	log[]				Parâmetro `regexp`
	log.count[]				
	logrt[]		Yes/No		O parâmetro `regexp` suporta ambos; o parâmetro `file_regexp` suporta apenas expressões não globais
	logrt.count[]				
	proc.cpu.util[]		No	No	Parâmetro `cmdline`
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	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[]			Yes	Parâmetro `regexp`
	vfs.file.regmatch[]				
	web.page.regexp[]				

Location		Regular expression	Global regular expression	Multiline matching	Comments
Traps SNMP					
	snmptrap[]	Yes	Yes	No	Parâmetro `regexp`
Pré-processamento do valor do item		Yes	No	No	Parâmetro `pattern`
Funções para trig-gers/itens calcula-dos					
	count()	Yes	Yes	Yes	Parâmetro `pattern` se o parâmetro `operator` for *regexp* ou *iregexp*
	countunique()	Yes	Yes		
	find()	Yes	Yes		
	logeventid()	Yes	Yes	No	Parâmetro `pattern`
	logsource()				
Descoberta em baixo nível					
	Filters	Yes	Yes	No	Campo *Regular expression*
	Overrides	Yes	No		Nas opções *matches*, *does not match* para condições de *Operation*
Condições de ação		Yes	No	No	Nas opções *matches*, *does not match* para condições de autorregistro de *Host name* e *Host metadata*
Scripts		Yes	Yes	No	Campo *Input validation rule*
Monitoramento web		Yes	No	Yes	*Variables* com prefixo **regex:** Campo *Required string*
Contexto de macro do usuário Funções de macro		Yes	No	No	No contexto da macro com prefixo **regex:**
	regsub()	Yes	No	No	Parâmetro `pattern`
	iregsub()				
Indicadores de link em mapas		Yes	No	No	Campo *Pattern* (para itens de texto)
Mapeamento de ícones		Yes	Yes	No	Campo *Expression*
Mapeamento de valores		Yes	No	No	Campo *Value* se o tipo de mapeamento for `regexp`
Parâmetros de configu-ração do Agent					
	AllowKeyRegexp	Yes	No	No	Pattern

Location		Regular expression	Global regular expression	Multiline matching	Comments
	DenyKeyRegexp				

Eventos e resolução

Visão geral Há vários tipos de eventos gerados no Zabbix:

- eventos de trigger - sempre que um trigger altera seu status (*OK→PROBLEM→OK*)
- eventos de serviço - sempre que um serviço altera seu status (*OK→PROBLEM→OK*)
- eventos de descoberta - quando hosts ou serviços são detectados
- eventos de autorregistro - quando agents ativos são registrados automaticamente pelo server
- eventos internos - quando um item/regra de descoberta de baixo nível se torna não suportado ou um trigger entra em um estado desconhecido

Os eventos têm marcação de tempo e podem servir de base para ações como envio de e-mail de notificação, criação de tickets ou execução de comandos remotos.

Para ver os detalhes do evento no frontend, vá para *Monitoring* > *Problems*. Lá, você pode clicar na data e hora do evento para abrir a página de detalhes do evento, onde é possível ver informações adicionais sobre o evento, incluindo se ele foi suprimido por causa de manutenção ou suprimido manualmente por um usuário.

Resolução de problemas Eventos de problema criados por triggers são resolvidos automaticamente se a condição que causou o problema não existir mais.

Para adicionar flexibilidade e conveniência ao gerenciamento de eventos de problema, existem as seguintes opções:

- Correlação de eventos
- Reconhecimento e supressão de eventos
- Fechamento manual de problemas
- Períodos de manutenção - definem períodos em que os problemas são suprimidos
- Marcação - marque problemas por característica comum

Eventos de trigger A mudança de estado do trigger é a fonte mais frequente e mais importante de eventos. Cada vez que o trigger altera seu estado, um evento é gerado. O evento contém detalhes da mudança de estado do trigger - quando ela aconteceu e qual é o novo estado.

Dois tipos de eventos são criados por triggers - `Problem` e `OK`.

Eventos de problema

Um evento de problema é criado:

- quando uma expressão de trigger é avaliada como TRUE se o trigger estiver no estado OK;
- a cada vez que uma expressão de trigger é avaliada como TRUE 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 ”OK event generation” e ”OK event closes”;
- correlação de eventos
- gerenciador de tarefas - quando um evento é fechado manualmente

Triggers têm uma configuração de ”OK event generation” que controla como os eventos OK são gerados:

- *Expression* - um evento OK é gerado para um trigger em estado de problema quando sua expressão é avaliada como FALSE. Esta é a configuração mais simples, habilitada por padrão.
- *Recovery expression* - um evento OK é gerado para um trigger em estado de problema quando sua expressão é avaliada como FALSE e a expressão de recuperação é avaliada como TRUE. Isso pode ser usado se o critério de recuperação do trigger for diferente do critério de problema.
- *None* - 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 acontece.

Além disso, triggers têm uma configuração de ”OK event closes” que controla quais eventos de problema são fechados:

- *All problems* - um evento OK fechará todos os problemas abertos criados pelo trigger
- *All problems if tag values match* - 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 de trigger ”Tag for matching”. Se não houver eventos de problema para fechar, então o evento OK não é gerado. Isso é frequentemente chamado de correlação de eventos no nível do trigger.

Correlação de eventos

A correlação de eventos (também chamada de correlação global de eventos) é uma forma de configurar regras personalizadas para o fechamento de eventos (resultando na geração de eventos OK).

As regras definem como os novos eventos de problema são associados aos eventos de problema existentes, permitindo que o novo evento ou os eventos correspondentes sejam fechados por meio da geração dos 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 configurada incorretamente, fechar mais eventos do que o pretendido (no pior caso, até mesmo todos os eventos de problema poderiam ser fechados). Algumas dicas de configuração:

1. sempre reduza o escopo da correlação definindo uma tag exclusiva para o evento de controle (o evento associado aos eventos antigos) e use a condição de correlação ’new event tag’
2. não se esqueça de adicionar uma condição baseada no evento antigo ao usar a operação ’close old event’, caso contrário, todos os problemas existentes poderão ser fechados
3. evite usar nomes de tag comuns utilizados por diferentes configurações de correlação

Gerenciador de tarefas

Se a configuração ’Allow manual close’ 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 ’close event’ é criada, que é tratada pelo gerenciador de tarefas em seguida. O gerenciador de tarefas gerará um evento OK correspondente e o evento de problema será fechado.

Eventos de service Os eventos de service são gerados somente se as ações de service para esses eventos estiverem habilitadas. Nesse caso, cada alteração do estado do service cria um novo evento:

- Evento de problema - quando o estado do service é alterado de `OK` para `Problem`
- Evento OK - quando o estado do service é alterado de `Problem` para `OK`

O evento contém detalhes da alteração do estado do service - quando ela aconteceu e qual é o novo estado.

Eventos de descoberta O Zabbix verifica periodicamente os intervalos de IP definidos nas regras de descoberta de rede. A frequência da verificação é configurável individualmente para cada regra. Assim que um host ou um serviço é descoberto, um evento de descoberta (ou vários eventos) é gerado.

O Zabbix gera os seguintes eventos:

Evento	Quando gerado
Service Up	Sempre que o Zabbix detecta um serviço ativo.
Service Down	Sempre que o Zabbix não consegue detectar o serviço.
Host Up	Se pelo menos um dos serviços estiver UP para o IP.
Host Down	Se todos os serviços não estiverem respondendo.
Service Discovered	Se o serviço voltar após um período de indisponibilidade ou for descoberto pela primeira vez.
Service Lost	Se o serviço for perdido após estar UP.
Host Discovered	Se o host voltar após um período de indisponibilidade ou for descoberto pela primeira vez.
Host Lost	Se o host for perdido após estar UP.

Eventos de autorregistro do agent ativo O autorregistro do agent ativo cria eventos no Zabbix.

Se configurado, um evento de autorregistro do agent ativo é criado quando um agent ativo anteriormente desconhecido solicita verificações ou se os metadados do host foram alterados. O server adiciona um novo host autorregistrado, usando o endereço IP e a porta recebidos do agent.

Para mais informações, consulte a página de autorregistro do agent ativo.

Eventos internos Eventos internos acontecem quando:

- um item muda de estado de 'normal' para 'unsupported'
- um item muda de estado de 'unsupported' para 'normal'
- uma regra de descoberta de baixo nível muda de estado de 'normal' para 'unsupported'
- uma regra de descoberta de baixo nível muda de estado de 'unsupported' para 'normal'
- um trigger muda de estado de 'normal' para 'unknown'
- um trigger muda de estado de 'unknown' para 'normal'

O objetivo da introdução de eventos internos é permitir que os usuários sejam notificados quando qualquer evento interno ocorrer, por exemplo, quando um item se torna unsupported e para de coletar dados.

Eventos internos só são criados quando as ações internas para esses eventos estão habilitadas. Para interromper a geração de eventos internos (por exemplo, para itens que se tornam unsupported), desabilite todas as ações para eventos internos em *Alerts > Actions > Internal actions*.

> **Note:**
> Se as ações internas estiverem desabilitadas, enquanto um objeto estiver no estado 'unsupported', o evento de recuperação para esse objeto ainda será criado.
> Se as ações internas estiverem habilitadas, enquanto um objeto estiver no estado 'unsupported', o evento de recuperação para esse objeto será criado, mesmo que o 'problem event' não tenha sido criado para o objeto.

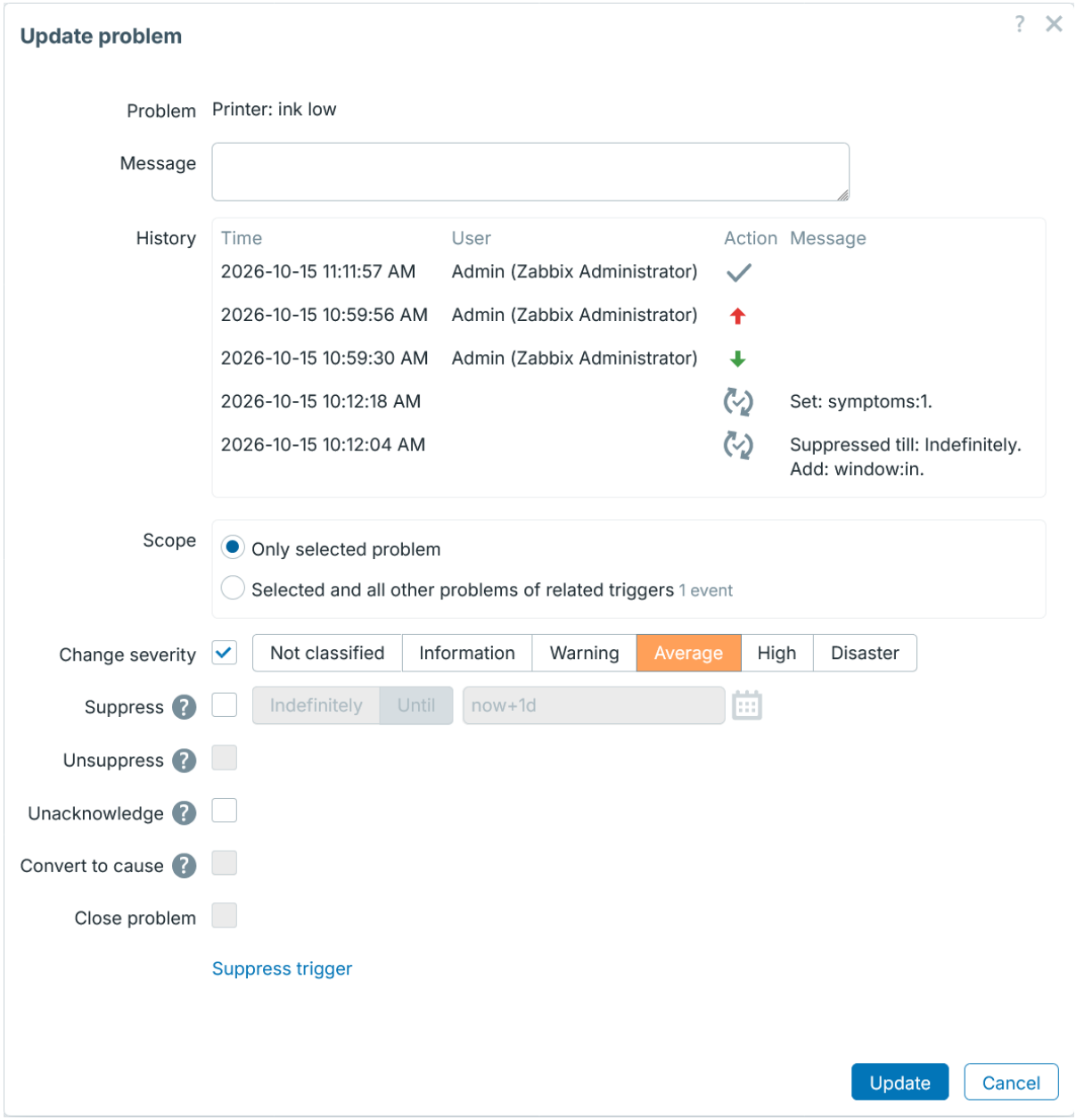
Veja também: Recebendo notificação sobre itens unsupported

Reconhecimento e supressão de problemas

Eventos de problema no Zabbix podem ser reconhecidos ou suprimidos por usuários.

Atualizando problemas O pop-up de atualização do problema permite:

- Comentar sobre o problema.
- Visualizar os comentários e as ações realizadas até o momento.
- Alterar a severidade do problema.
- Suprimir/remover a supressão do problema.
- Confirmar/cancelar a confirmação do problema.
- Alterar o problema sintoma para problema causa.
- Fechar manualmente o problema.
- Criar uma manutenção ad hoc para o trigger.



Todos os campos de entrada 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, *N problemas selecionados* será exibido.
Mensagem	Insira o texto do comentário sobre o problema (máximo de 2048 caracteres).
Histórico	As atividades e os comentários anteriores sobre o problema são listados, juntamente com o horário e os detalhes do usuário. Para obter o significado dos ícones usados para indicar as ações dos usuários, consulte a página de detalhes do evento. Observe que o histórico é exibido somente quando um problema é selecionado para atualização.
Escopo	Defina o escopo de ações como alterar a severidade, confirmar ou fechar manualmente problemas: **Somente o problema selecionado** - afetará somente este evento **O problema selecionado e todos os outros problemas dos triggers relacionados** - no caso de confirmação/fechamento do problema, afetará este evento e todos os outros problemas que ainda não foram confirmados/fechados. Se o escopo contiver problemas já confirmados ou fechados, eles não serão confirmados/fechados novamente. Por outro lado, o número de operações de alteração de mensagem e severidade não é limitado.

Parâmetro	Descrição
Alterar severidade	Marque a caixa de seleção e clique no botão de severidade para atualizar a severidade do problema. A caixa de seleção para alterar a severidade estará disponível se houver permissões de leitura e gravação para pelo menos um dos problemas selecionados. Somente os problemas com permissão de leitura e gravação serão atualizados ao clicar em *Atualizar*. Se não houver permissões de leitura e gravação para nenhum dos triggers selecionados, a caixa de seleção ficará desabilitada.
Suprimir	Marque a caixa de seleção para suprimir o problema: **Indefinidamente** - suprimir indefinidamente **Até** - suprimir até um horário especificado. Os formatos de horário absoluto e relativo são compatíveis, 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 compatível. A disponibilidade dessa opção depende das configurações da função de usuário "Suprimir problemas". Veja também: Supressão de problemas
Remover supressão	Marque a caixa de seleção para remover a supressão do problema. Essa caixa de seleção estará ativa somente se pelo menos um dos problemas selecionados estiver suprimido. A disponibilidade dessa opção depende das configurações da função de usuário "Suprimir problemas".
Confirmar	Marque a caixa de seleção para confirmar o problema. Essa caixa de seleção estará disponível se houver pelo menos um problema não confirmado entre os selecionados. Não é possível adicionar outra confirmação a um problema já confirmado (mas é possível adicionar outro comentário).
Cancelar confirmação	Marque a caixa de seleção para cancelar a confirmação do problema. Essa caixa de seleção estará disponível se houver pelo menos um problema confirmado entre os selecionados.
Converter em 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 estará disponível se a opção *Permitir fechamento manual* estiver marcada na configuração do trigger para pelo menos um dos problemas selecionados. Somente os problemas que podem ser fechados serão fechados ao clicar em *Atualizar*. Se nenhum problema puder ser fechado manualmente, a caixa de seleção ficará desabilitada. Problemas já fechados não serão fechados novamente.
Suprimir trigger	Clique para iniciar uma manutenção ad hoc para o trigger. Um formulário de configuração do período de manutenção será aberto. Nele, um período de manutenção único de 1h (configurável nas configurações e no perfil do usuário), com início a partir de agora, será pré-selecionado. O trigger correspondente também será pré-selecionado. O nome da manutenção será preenchido previamente no formato `Ad-hoc: <event name>`.

Reconhecimento de problema 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 sobre ele, dizendo que está trabalhando nisso ou qualquer outra coisa que queira mencionar.

Dessa forma, se outro usuário do sistema identificar o mesmo problema, ele verá imediatamente se ele já foi reconhecido e quais comentários foram feitos até o momento.

Assim, o fluxo de trabalho de resolução de problemas com 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 a um gerente de nível superior somente 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 fechar o problema, um usuário deve ter permissões de leitura e gravação para os triggers correspondentes.

Há **várias** maneiras de acessar a janela pop-up de atualização do problema, que permite reconhecer um problema.

- Você pode selecionar problemas em *Monitoring* > *Problems* e então clicar em *Mass update* abaixo da lista

- Você pode clicar em *Update* na coluna *Update* de um problema em:
  - *Dashboards* (widgets *Problems* e *Problems by severity*)
  - *Monitoring* > *Problems*
  - *Monitoring* > *Problems* > *Event details*
- Você pode clicar em uma célula de problema não resolvido em:
  - *Dashboards* (widget *Trigger overview*)

O menu pop-up contém uma opção *Update* que levará você à janela de atualização do problema.

Com base nas informações de reconhecimento, é possível configurar como a contagem de problemas é exibida no dashboard ou nos mapas. Para isso, você deve fazer seleções na opção *Exibição de problemas*, disponível tanto na configuração do mapa quanto no widget do dashboard *Problemas por severidade*.

É possível exibir a contagem total de problemas, a contagem de problemas não reconhecidos separada da contagem total ou somente 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.

Supressão de problemas 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 e dar prioridade máxima às questões mais urgentes. Por exemplo, às vezes pode surgir um problema no fim de semana que não seja urgente o suficiente para ser tratado imediatamente; nesse caso, ele pode ser "adiado" até segunda-feira de manhã.

A supressão de problemas permite ocultar um *único* problema, ao contrário da supressão 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 ocorre com a manutenção do host.

Um problema pode ser suprimido por meio da janela **atualização de problema**, na qual a supressão é uma das opções de atualização do problema, juntamente com comentários, alteração de severidade, reconhecimento etc.

Um problema também pode ser dessuprimido por meio da mesma janela de atualização de problema.

Exibição

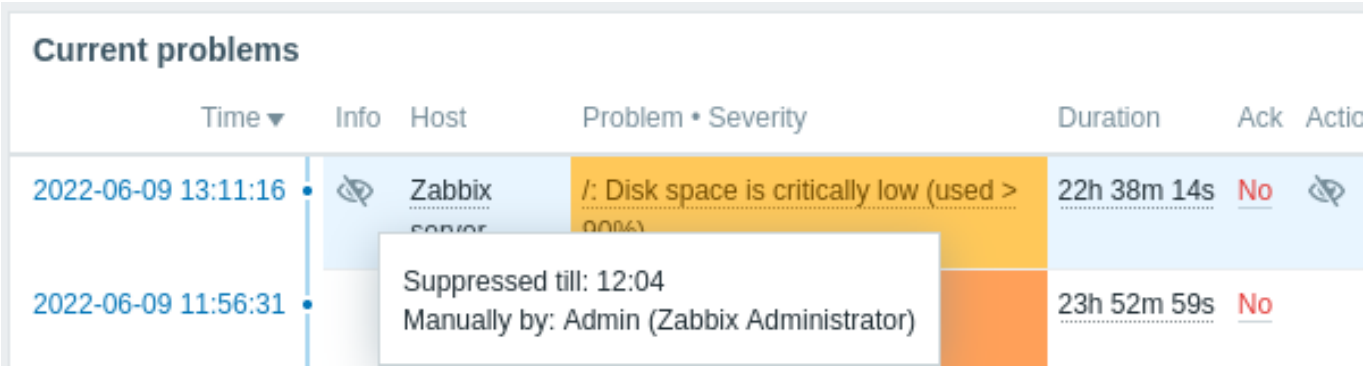
Depois de suprimido, o problema é marcado por um ícone de supressão piscante na coluna *Info*, antes de ser ocultado.

Se a supressão ocorrer devido à manutenção do host, o ícone de supressão na coluna *Info* é exibido em laranja. Se tanto a manutenção quanto a supressão manual se aplicarem, a supressão manual tem prioridade na cor do ícone.

O ícone de supressão pisca enquanto a tarefa de supressão estiver na lista de espera. Assim que o gerenciador de tarefas suprimir o problema, o ícone deixará 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 indisponível e o gerenciador de tarefas não conseguir concluir a tarefa. A mesma lógica se aplica à dessupressão. No curto período após o envio da tarefa e antes de o server concluí-la, o ícone de dessupressão pisca.

Um problema suprimido pode estar oculto ou visível, dependendo das configurações do filtro/widget de problemas.

Quando exibido na lista de problemas, um problema suprimido é marcado pelo ícone de supressão e os detalhes da supressão são mostrados 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 *Actions*.

Fechamento manual de problemas

Visão geral Embora, em geral, os eventos de problema sejam resolvidos automaticamente quando o estado do trigger muda de `Problem` para `OK`, pode haver casos em que seja 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 informar que alguns parâmetros do kernel precisam ser ajustados para um desempenho ideal. Nesse caso, o problema é comunicado aos administradores do Linux, eles o corrigem e então fecham o problema manualmente.

Os problemas podem ser fechados manualmente apenas para triggers com a opção *Allow manual close* habilitada.

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.

O evento *OK* gerado inclui o conjunto completo de tags de evento resolvidas para esse evento (incluindo tags herdadas de templates, hosts e triggers). Essas tags estão disponíveis em notificações e em macros como `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

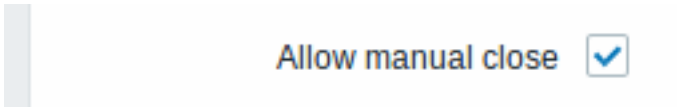
Um problema fechado manualmente não significa que o trigger subjacente nunca mais entrará no estado *Problem*. A expressão do trigger é reavaliada e pode resultar em um problema:

- Quando novos dados chegam para qualquer item incluído na expressão do trigger (observe que os valores descartados por uma etapa de pré-processamento de limitação não são considerados como recebidos e não farão com que a expressão do trigger seja reavaliada);
- Quando as funções date and time e/ou nodata() são usadas na expressão.

Configuração São necessárias duas etapas para fechar um problema manualmente.

Configuração de trigger

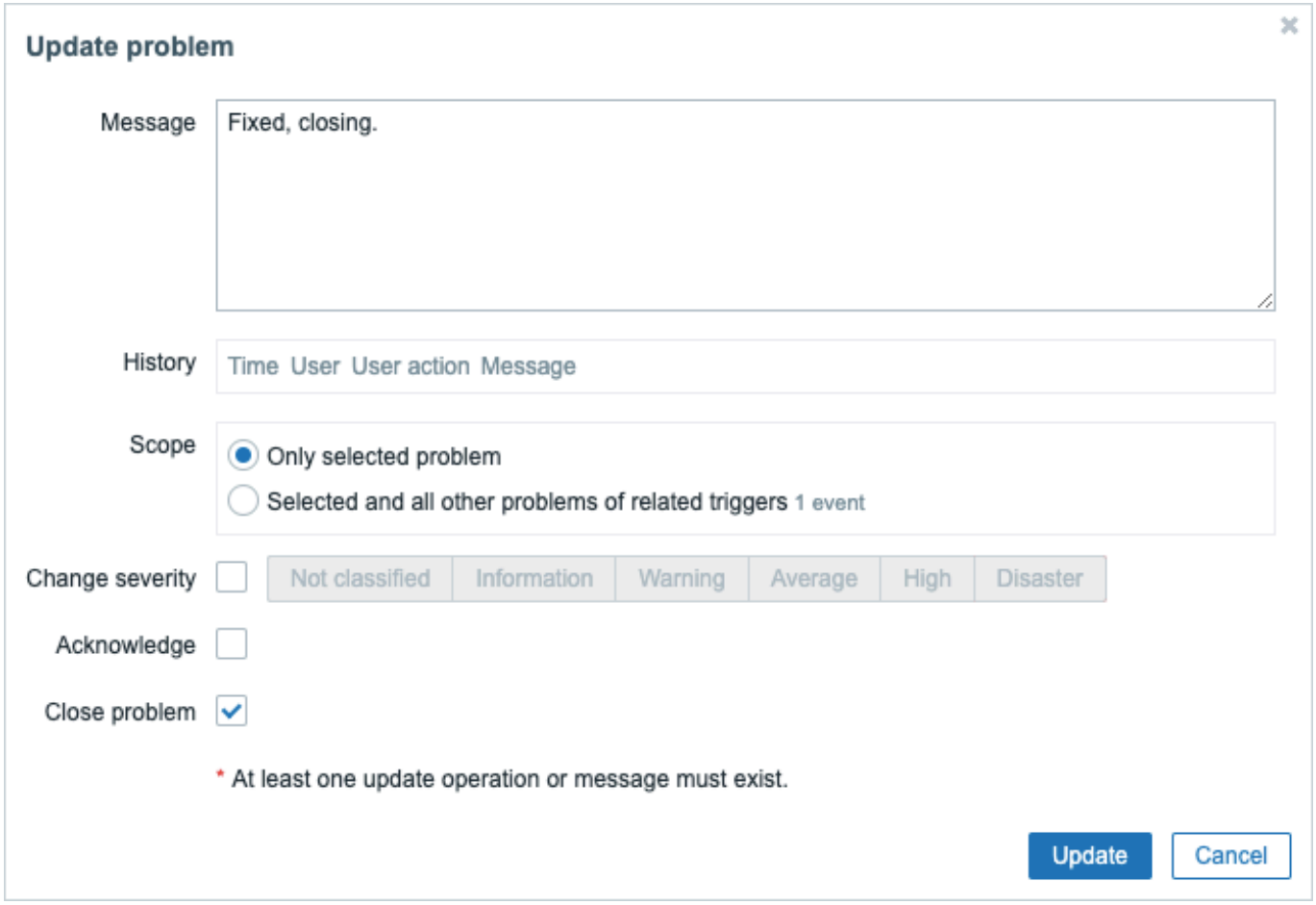
Na configuração de trigger, habilite a opção *Allow manual close*.



Janela de atualização de problema

Se surgir um problema para um trigger com a flag *Manual close*, você pode abrir a janela pop-up de atualização de problema desse problema e fechá-lo manualmente.

Para fechar o problema, marque a opção *Close problem* no formulário e clique em *Update*.



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

A solicitação é processada pelo Zabbix server. Normalmente, levará alguns segundos para fechar o problema. Durante esse processo, *CLOSING* é exibido em *Monitoring* > *Problems* como o status do problema.

Verificação É possível verificar que um problema foi encerrado manualmente:

- nos detalhes do evento, disponíveis em *Monitoring* > *Problems*;
- usando a macro `{EVENT.UPDATE.HISTORY}` nas mensagens de notificação, que fornecerá essa informação.

Períodos de manutenção

Visão geral

A manutenção é usada para suprimir problemas durante períodos predefinidos.

Você pode definir períodos de manutenção para estas entidades:

- Grupos de hosts
- Hosts
- Triggers
- Nomes de eventos
- Tags

Se a manutenção for definida para tags específicas ou nomes de eventos, ela será ativada somente para os triggers correspondentes; todos os outros triggers do host ou grupo de hosts não estarão em manutenção.

Há dois tipos de manutenção: *Com coleta de dados* e *Sem coleta de dados*.

Durante a manutenção *Com coleta de dados*, os triggers são processados normalmente e os eventos são criados quando necessário. No entanto, se a opção *Pausar operações para problemas suprimidos* estiver marcada na configuração da ação, as escalações de problemas serão pausadas para as entidades em manutenção. Nesse caso, as etapas de escalação, 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 não são suprimidas durante a manutenção, apenas as escalações. As notificações de recuperação não são enviadas se o problema tiver começado durante a manutenção.

Por exemplo, se as etapas de escalação estiverem programadas para 0, 30 e 60 minutos após o início de um problema, e houver uma manutenção de meia hora com duração entre 10 e 40 minutos após o surgimento de um problema real, as etapas dois e três serão executadas meia hora mais tarde, ou seja, aos 60 e 90 minutos (desde que o problema ainda exista). Da mesma forma, se um problema surgir durante a manutenção, a escalação começará após a manutenção.

Para receber normalmente as notificações de problemas durante a manutenção (sem atraso), desmarque a opção *Pausar operações para problemas suprimidos* na configuração da ação.

Note:
Se pelo menos um host (usado na expressão do trigger) não estiver no modo de manutenção, o Zabbix enviará uma notificação de problema.

O Zabbix server deve estar em execução durante a manutenção. As manutenções são recalculadas a cada minuto ou assim que o cache de configuração é recarregado, caso haja alterações no período de manutenção.

Os processos de timer verificam se o status do host deve ser alterado para/de manutenção aos 0 segundos de cada minuto. Além disso, a cada segundo, o processo de timer verifica se alguma manutenção deve ser iniciada/interrompida com base nas alterações nos períodos de manutenção após a atualização da configuração. Assim, a velocidade de início/interrupção 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 desde/Ativo até*. Além disso, se uma entidade (grupo de hosts/host/trigger/etc.) for adicionada a um período de manutenção ativo existente, as alterações só serão ativadas pelo processo de timer no início do próximo minuto.

Observe que, quando um host entra em manutenção, os processos de timer do Zabbix server lerão todos os problemas abertos para verificar se é necessário suprimi-los. Isso pode afetar o desempenho se houver muitos problemas abertos. O Zabbix server também lerá todos os problemas abertos durante a inicialização, mesmo que não haja manutenções configuradas naquele momento.

Observe que o Zabbix server (ou proxy) sempre coleta dados, independentemente do tipo de manutenção (incluindo a manutenção *Sem coleta de dados*). Os dados serão posteriormente ignorados pelo server se *Sem coleta de dados* estiver definido.

Quando a manutenção *Sem coleta de dados* termina, os triggers que usam a função `nodata()` não serão acionados antes da próxima verificação durante o período que estão verificando.

Se um item de log for adicionado enquanto um host estiver em manutenção e a manutenção terminar, somente as novas entradas do arquivo de log desde o fim da manutenção serão coletadas.

Se um valor com timestamp for enviado para um host que esteja em manutenção *Sem coleta de dados* (por exemplo, usando o Zabbix sender), esse valor será descartado; no entanto, é possível enviar um valor com timestamp para um período de manutenção expirado, e ele será aceito.

Se o período de manutenção ou a opção de manutenção (grupo de hosts, host, trigger etc.) forem alterados pelo usuário, as alterações só entrarão em vigor após a sincronização do cache de configuração.

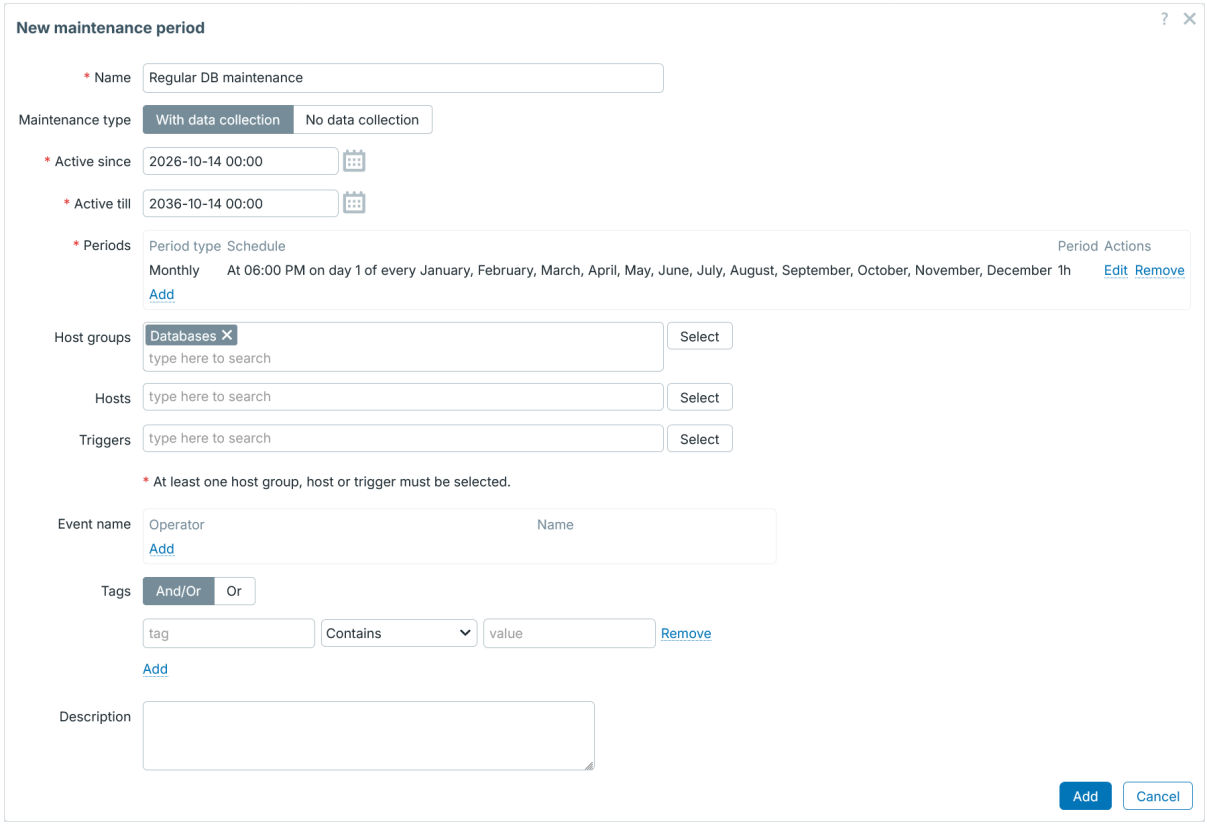
Configuração

Para iniciar um período de manutenção imediatamente:

1. Acesse *Monitoring* > *Problems*.
2. Clique no nome do problema.
3. Selecione a opção de manutenção no menu do evento (host, trigger, nome do evento, tags).
4. Modifique os parâmetros de manutenção no formulário.

Para configurar um período de manutenção planejado:

1. Acesse *Data collection* > *Maintenance*.
2. Clique em *Create maintenance period* (ou no nome de um período de manutenção existente).
3. Insira os parâmetros de manutenção no formulário.



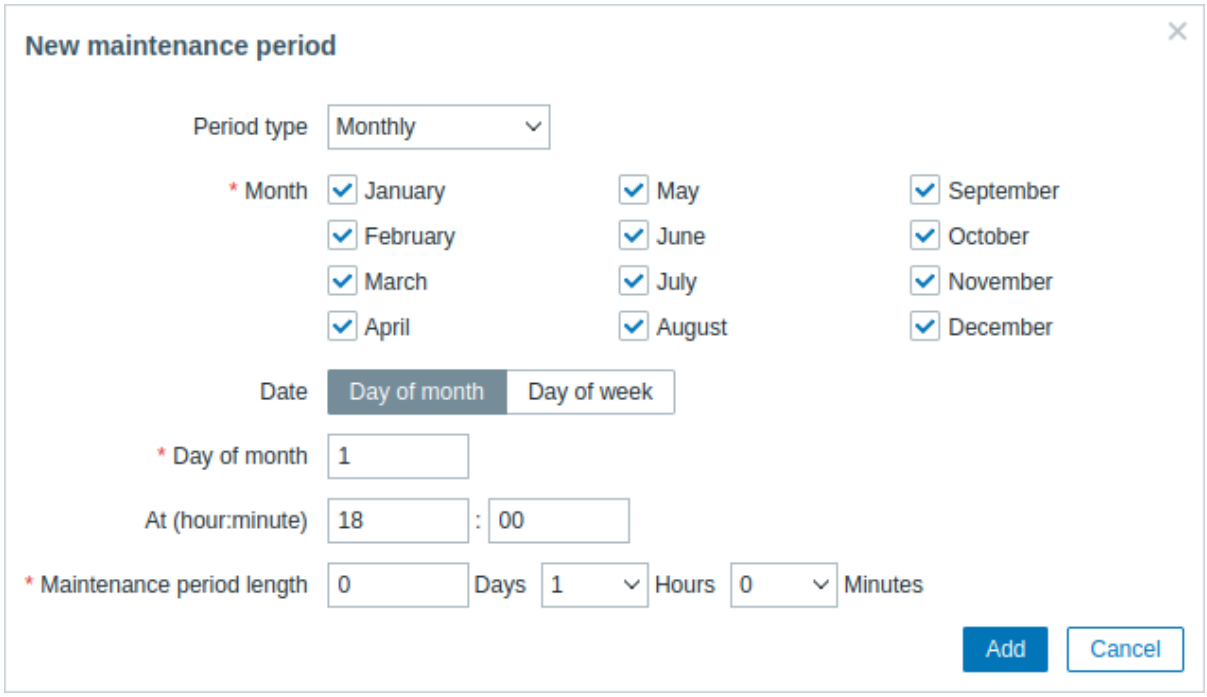
Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Name	Nome do período de manutenção.
Maintenance type	Dois tipos de manutenção podem ser definidos: **With data collection** - os dados são coletados pelo server durante a manutenção, e os triggers são processados; **No data collection** - os dados ainda podem ser coletados, mas não são salvos no banco de dados durante a manutenção, e os triggers (incluindo a função `nodata()`) não são acionados. Consulte Efeito dos períodos de manutenção para saber como cada tipo afeta o relatório de disponibilidade.
Active since	A data e a hora em que a execução dos períodos de manutenção se torna ativa. *Observação:* Definir apenas esse horário não ativa um período de manutenção; os períodos de manutenção devem ser configurados em *Periods* (veja abaixo).
Active till	A data e a hora em que a execução dos períodos de manutenção deixa de estar ativa.
Periods	Este bloco permite definir os dias e horários exatos em que a manutenção ocorrerá. Clicar em Add abre uma janela pop-up com um formulário flexível de *Maintenance period*, no qual é possível definir a programação da manutenção. Consulte Períodos de manutenção para obter uma descrição detalhada.
Host groups	Selecione os grupos de hosts para os quais a manutenção será ativada. A manutenção será ativada para todos os hosts dos grupos de hosts especificados. Este campo oferece preenchimento automático; ao começar a digitar, uma lista suspensa com todos os grupos de hosts disponíveis será exibida. 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 oferece preenchimento automático; ao começar a digitar, uma lista suspensa com todos os hosts disponíveis será exibida.
Triggers	Selecione os triggers para os quais a manutenção será ativada. Este campo oferece preenchimento automático; ao começar a digitar, uma lista suspensa com todos os triggers disponíveis será exibida.

Parâmetro	Descrição
Event name	Especifique os nomes de eventos para os quais a manutenção será ativada. Várias condições podem ser definidas. Há dois operadores disponíveis para cada condição: **Contains** - inclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, sem distinção entre maiúsculas e minúsculas) **Does not contain** - exclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, sem distinção entre maiúsculas e minúsculas)
Tags	Especifique as tags para as quais a manutenção será ativada. Várias condições podem ser definidas. A correspondência dos nomes das tags sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Contains** - inclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, com distinção entre maiúsculas e minúsculas); **Equals** - inclui os nomes e valores de tags especificados (com distinção entre maiúsculas e minúsculas). **Does not equal** - exclui os nomes e valores de tags especificados (com distinção entre maiúsculas e minúsculas) **Does not contain** - exclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, sem distinção entre maiúsculas e minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - basta que uma condição seja atendida. As tags só podem ser especificadas se o tipo de manutenção *With data collection* estiver selecionado.
Description	Descrição do período de manutenção.

Períodos de manutenção

A janela do período de manutenção é usada para agendar um horário para uma manutenção recorrente ou única. O formulário é dinâmico, com campos disponíveis que mudam com base no *Tipo de período* selecionado.



Tipo de período	Descrição
Apenas uma vez	Configure um período de manutenção de apenas uma vez: *Data* - data e hora do período de manutenção; *Duração do período de manutenção* - por quanto tempo a manutenção ficará 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)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção ficará ativa. Quando o parâmetro *A cada dia(s)* for maior que ”1”, o dia de início é o dia em que o horário de *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 1 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)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção ficará ativa. Quando o parâmetro *A cada semana(s)* for maior que ”1”, a semana de início é a semana em que o horário de *Ativo desde* cai. Para exemplos, consulte a descrição do parâmetro *Diário* acima.
Mensal	Configure um período de manutenção mensal: *Mês* - selecione todos os meses durante os quais a manutenção regular é realizada; *Data:* ***Dia do mês*** - selecione esta opção se a manutenção deve ocorrer na mesma data todos os meses (por exemplo, todo dia 1 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); em seguida, selecione (na lista suspensa) a semana do mês necessária (primeira, segunda, terceira, quarta ou última) e depois marque as caixas de seleção para o(s) dia(s) de manutenção; *Às (hora:minuto)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção ficará ativa.

Attention:
Ao criar um período de manutenção, é usado o fuso horário do usuário que o cria. No entanto, quando períodos de manutenção recorrentes (*Daily*, *Weekly*, *Monthly*) são agendados, é usado o fuso horário do server Zabbix. Para garantir um comportamento previsível dos períodos de manutenção recorrentes, é necessário usar um fuso horário comum para todas as partes do Zabbix.

Quando terminar, clique em *Add* para adicionar o período de manutenção à seção *Periods*.

Observe que as alterações de Horário de Verão (DST) não afetam a duração da manutenção. Por exemplo, suponha que tenhamos uma manutenção configurada para duas horas, que normalmente começa às 01:00 e termina às 03:00:

- se, após uma hora de manutenção (às 02:00), ocorrer uma alteração de DST 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 alteração de DST e o horário atual mudar de 03:00 para 02:00, a manutenção será encerrada, porque duas horas já terão passado;
- se um período de manutenção começar durante a hora que é ignorada por uma alteração de DST, então a manutenção não será iniciada.

Se um período de manutenção estiver definido como ”1 day” (o período real da manutenção é de 24 horas, já que o Zabbix calcula dias em horas), começar às 00:00 e terminar às 00:00 do dia seguinte:

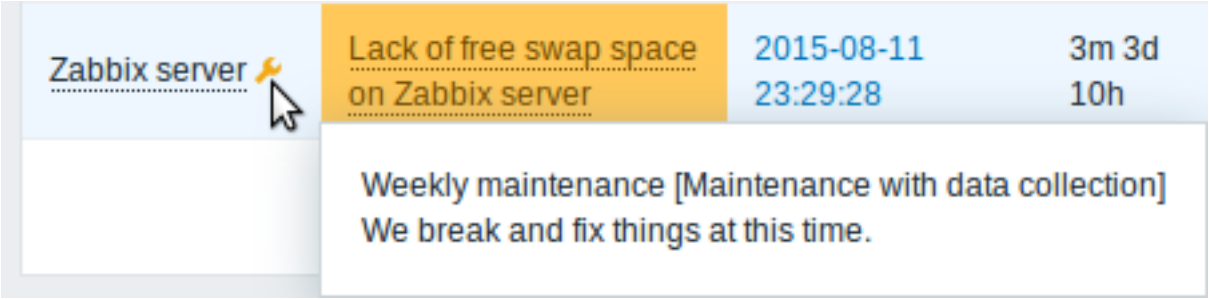
- a manutenção será encerrada às 01:00 do dia seguinte se o horário atual avançar uma hora;
- a manutenção será encerrada às 23:00 daquele dia se o horário atual retroceder uma hora.

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*
- *Monitoring* > *Problems*
- *Inventory* > *Hosts* > *Host inventory details*
- *Data collection* > *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 *Monitoring* > *Maps*.

Exibindo problemas suprimidos

Normalmente, os problemas de hosts em manutenção são suprimidos, ou seja, não são exibidos no frontend.

Problemas suprimidos também podem ficar ocultos porque foram suprimidos manualmente.

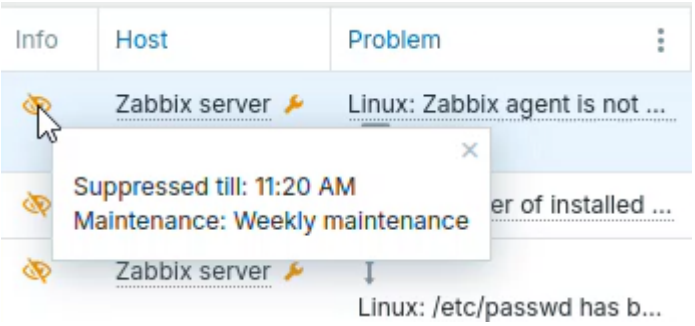
É possível exibir problemas suprimidos selecionando a opção *Mostrar problemas suprimidos* nestes locais:

- *Dashboards* (na configuração dos widgets *Problem hosts*, *Problems*, *Problems by severity*, *Trigger overview*)
- *Monitoring* > *Problems* (no filtro)
- *Monitoring* > *Maps* (na configuração do mapa)
- notificações globais (na configuração do perfil do usuário)

Quando problemas suprimidos são exibidos, o seguinte ícone é mostrado: . Se a supressão for devido à manutenção do host, os seguintes ícones são mostrados:

- - o problema foi suprimido devido à manutenção do host.
- - o problema deixou de ser suprimido após a manutenção do host.

Ao passar o mouse sobre o ícone, são exibidos mais detalhes. Nos detalhes do evento, você pode verificar se o evento foi suprimido por manutenção ou suprimido manualmente. Se tanto a manutenção quanto a supressão manual se aplicarem, a supressão manual tem prioridade na cor do ícone.



Cálculo das filas durante a manutenção

As filas exibidas no frontend do Zabbix (Administration > Queue) são calculadas pelo server do Zabbix. Elas não incluem items em manutenção *No data collection*, e o tamanho da fila é sempre zero para esses items, mesmo quando seus valores estão atrasados. Items atrasados em manutenção *With data collection* 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 do Zabbix e o proxy. As 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 tamanhos de fila podem ser relatados para os mesmos items em manutenção *No data collection* pelo frontend do Zabbix e pelas verificações internas em proxies do Zabbix.

Marcação

Tags têm múltiplas finalidades, principalmente para marcar events. Quando entidades são marcadas com tags, qualquer novo evento relacionado a uma entidade com tag herdará suas tags. Por exemplo:

- com templates com tags - qualquer problema de host (criado por triggers do template) herdará as tags do template.
- com hosts com tags - qualquer problema de host herdará as tags do host.
- com items/scenarios web com tags - qualquer problema de item/scenario web herdará as tags do item/scenario web.
- com triggers com tags - 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/scenarios web, triggers. Combinações idênticas de `tag:value` (após a resolução de macros) são mescladas em uma só, evitando assim duplicação.

Eventos de recuperação gerados por um manual close também incluem as tags de evento resolvidas herdadas de templates, hosts, items/scenarios web e triggers. Essas tags estão disponíveis em notificações e por meio de macros como `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

Tags de evento personalizadas oferecem mais flexibilidade. Por exemplo:

- event correlation pode ser configurada com base em tags de evento.
- action conditions podem ser configuradas com base em tags de evento.
- problemas de item podem ser agrupados com base em tags de evento.
- tags de problema podem ser usadas para mapear problemas para services.

As entidades podem ser marcadas com o mesmo nome de tag, mas com valores de tag diferentes (por exemplo, `component:memory` e `component:storage`). Da mesma forma, uma entidade pode ter uma tag sem valor e a mesma tag com um valor (por exemplo, `database` e `database:postgresql`). Essas tags não são consideradas duplicadas.

Tags consistem em um nome de tag e um valor de tag. Ao marcar entidades, você pode usar apenas o nome ou combiná-lo com um valor (por exemplo, `mysql`, `jira`, `target:mysql`, `service:jira`, etc.).

Tags podem ser definidas para várias entidades:

- Templates
- Hosts
- Items
- Web scenarios
- Triggers
- Services
- Template items and triggers
- Host, item, and trigger prototypes

Nas listas *Templates*, *Hosts*, *Items*, *Triggers*, *Web scenarios* e em seus protótipos, a coluna *Tags* exibe tanto as tags regulares quanto as herdadas. Ao passar o mouse sobre uma tag *herdada* ou clicar nela, aparece uma dica de ferramenta JavaScript com o texto "Inherited tag". Se uma tag herdada também existir como tag regular, um texto diferente de dica de ferramenta é exibido dependendo da lista em que ela aparece (por exemplo, "Inherited and template tag" na lista de templates ou "Inherited and host tag" na lista de hosts). Em outros locais do frontend, as tags regulares e herdadas são exibidas juntas (as duplicatas são removidas), sem ícones e sem texto adicional de dica de ferramenta.

> **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 em baixo nível.

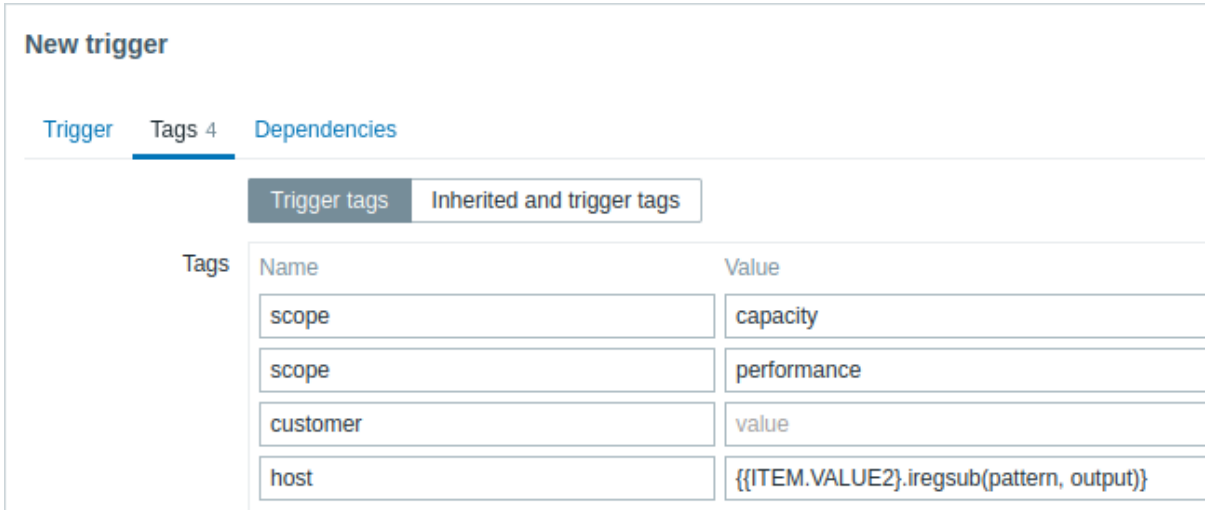
Casos de uso Alguns casos de uso comuns para marcação são os seguintes:

1. Marcar eventos de trigger:
   - Defina uma tag de trigger (por exemplo, `scope:performance`).
   - Os problemas criados por esse 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 *Monitoring* > *Latest data*, 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 uma macro como valor da tag (por exemplo, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Em *Monitoring* > *Problems*, 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 item de monitoramento de log que extrairá valores do valor do item usando uma macro (por exemplo, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Na configuração do trigger, configure a correlação de eventos:
     - defina o modo de geração de eventos *PROBLEM event generation mode* como "Multiple";
     - defina *OK event closes* como "All problems if tag values match";
     - 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 filtragem por tag nas action conditions para receber notificações apenas nos eventos que correspondem aos dados da tag.
8. Identificar problemas nas notificações:
   - Defina tags de trigger.
   - Use a macro {EVENT.TAGS} na notificação de problema.
   - A notificação de problema conterá as tags de trigger, facilitando a identificação de qual aplicação/serviço a notificação pertence.
9. Simplificar tarefas de configuração usando tags de template:
   - Defina uma tag de trigger de template.
   - Os triggers criados a partir 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 uma macro LLD no nome ou valor da tag (por exemplo, `scope:{#FSNAME}`).
    - Os triggers criados a partir do protótipo de trigger terão sua tag.
11. Correspondência de serviços usando tags de serviço:
    - Defina service tags.
    - Configure service actions 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 problem tag na configuração do serviço (por exemplo, `target:mysql`).
    - Problemas com uma tag correspondente serão correlacionados automaticamente ao serviço, e o status do serviço mudará com base nas regras configuradas de cálculo do status do serviço.
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 de grupo de usuários.
    - Os usuários no grupo de usuários poderão visualizar apenas os problemas com as tags definidas.

Configuração As tags podem ser definidas em uma aba dedicada, por exemplo, em configuração de trigger:



Suporte a macros Built-in e user macros em tags são resolvidas no momento do evento. Até que o evento ocorra, essas macros serão exibidas no frontend do Zabbix como não resolvidas.

Low-level discovery macros são resolvidas durante o processo de descoberta.

As seguintes macros podem ser usadas em nomes e valores de tags de trigger:

- macros built-in {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 built-in {INVENTORY.*} (para referenciar valores de inventário do host de um ou vários hosts em uma expressão de trigger)
- user macros e user macros com contexto (o contexto pode incluir low-level discovery macros)
- low-level discovery macros (somente em tags de prototype de trigger)

As seguintes macros podem ser usadas em nomes e valores de tags de template, host e item/web scenario:

- macros built-in {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID}
- macros built-in {INVENTORY.*}
- user macros
- low-level discovery macros (somente em tags de prototype de host e item)

As seguintes macros podem ser usadas em notificações baseadas em trigger:

- macros built-in {EVENT.TAGS} e {EVENT.RECOVERY.TAGS} (essas macros serão resolvidas para uma lista separada por vírgulas de tags de evento ou tags de evento de recuperação)
- macros built-in {EVENT.TAGSJSON} e {EVENT.RECOVERY.TAGSJSON} (essas macros serão resolvidas para uma matriz JSON contendo objetos de tag de evento objects ou objetos de tag de evento de recuperação)

Extração de substring em tags de trigger

A extração de substring é suportada para preencher o nome da tag ou o valor da tag, usando uma função de macro. A função aplica uma expressão regular ao valor obtido pela macro suportada. Por exemplo:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

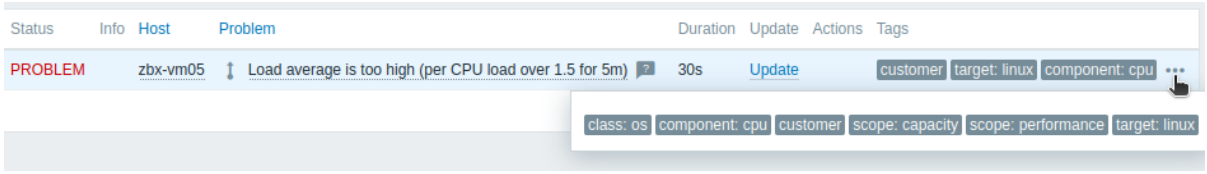
Se o nome ou o valor da tag exceder 255 caracteres após a resolução da macro, ele será truncado para 255 caracteres.

Veja também: Uso de funções de macro em macros de descoberta em baixo nível para marcação de eventos.

Visualizando tags de eventos As tags, se definidas, podem ser visualizadas com novos eventos em:

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

A ordem e o número de tags exibidas são determinados pelas opções de filtragem *Tag display priority* e *Show tags* em *Monitoring > Problems* ou no widget *Problems* do dashboard. 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.



Correlação de eventos

Visão geral

A correlação de eventos permite correlacionar eventos de problema com sua resolução de 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 - os problemas podem ser correlacionados à sua solução a partir de um trigger/método de coleta diferente, usando regras de correlação globais.

Correlação de eventos baseada em trigger

Visão geral

A correlação de eventos baseada em trigger permite correlacionar problemas separados relatados por um trigger.

Embora, em geral, um evento OK possa fechar todos os eventos de problema criados por um trigger, há casos em que é necessária uma abordagem mais detalhada. Por exemplo, ao monitorar arquivos de log, talvez você queira detectar determinados problemas em um arquivo de log e fechá-los individualmente, em vez de fechá-los todos juntos.

Esse é o caso dos triggers que têm o parâmetro *PROBLEM event generation mode* definido como *Multiple*. Esses triggers normalmente são usados para monitoramento de logs, processamento de traps etc.

No Zabbix, é possível relacionar eventos de problema com base em tags. As tags são usadas para extrair valores e criar uma identificação para os eventos de problema. Aproveitando esse recurso, os problemas também podem ser fechados individualmente com base na correspondência de uma tag.

Em outras palavras, o mesmo trigger pode criar eventos separados identificados pela tag do evento. Dessa forma, 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 conseguir associar o evento de problema da Linha1 à resolução da Linha3 e o evento de problema da Linha2 à resolução da Linha4, e encerrar esses problemas um a um:

```
Linha1: Serviço 1 parado
Linha3: Serviço 1 foi reiniciado #problema da Linha 1 encerrado

Linha2: Serviço 2 parado
Linha4: Serviço 2 foi reiniciado #problema da Linha 2 encerrado
```

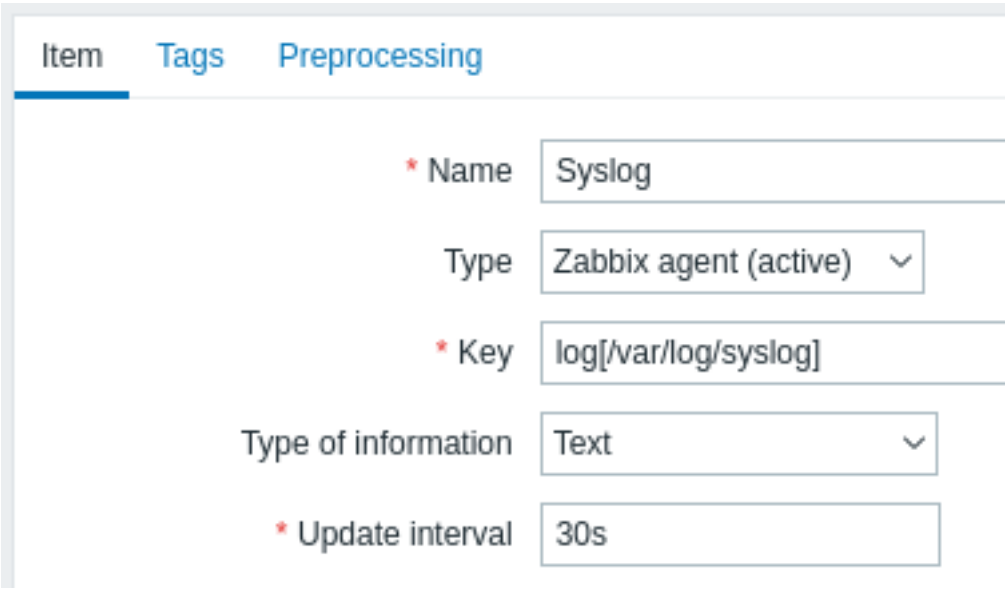
Para fazer isso, você precisa marcar esses eventos relacionados, por exemplo, como ”Serviço 1” e ”Serviço 2”. Isso pode ser feito aplicando uma expressão regular à linha do log para extrair o valor da tag. Então, quando os eventos são criados, eles recebem as tags ”Serviço 1” e ”Serviço 2” respectivamente, e o problema pode ser associado à resolução.

Configuração

Item

Para começar, talvez você queira 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 Latest data para verificar se o item começou a coletar dados.

Trigger

Com o item em funcionamento, você precisa configurar o trigger. É importante decidir quais entradas no arquivo de log merecem atenção. Por exemplo, a seguinte expressão de trigger procurará 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, defina também o *Problem event generation mode* na configuração do trigger como 'Multiple'.

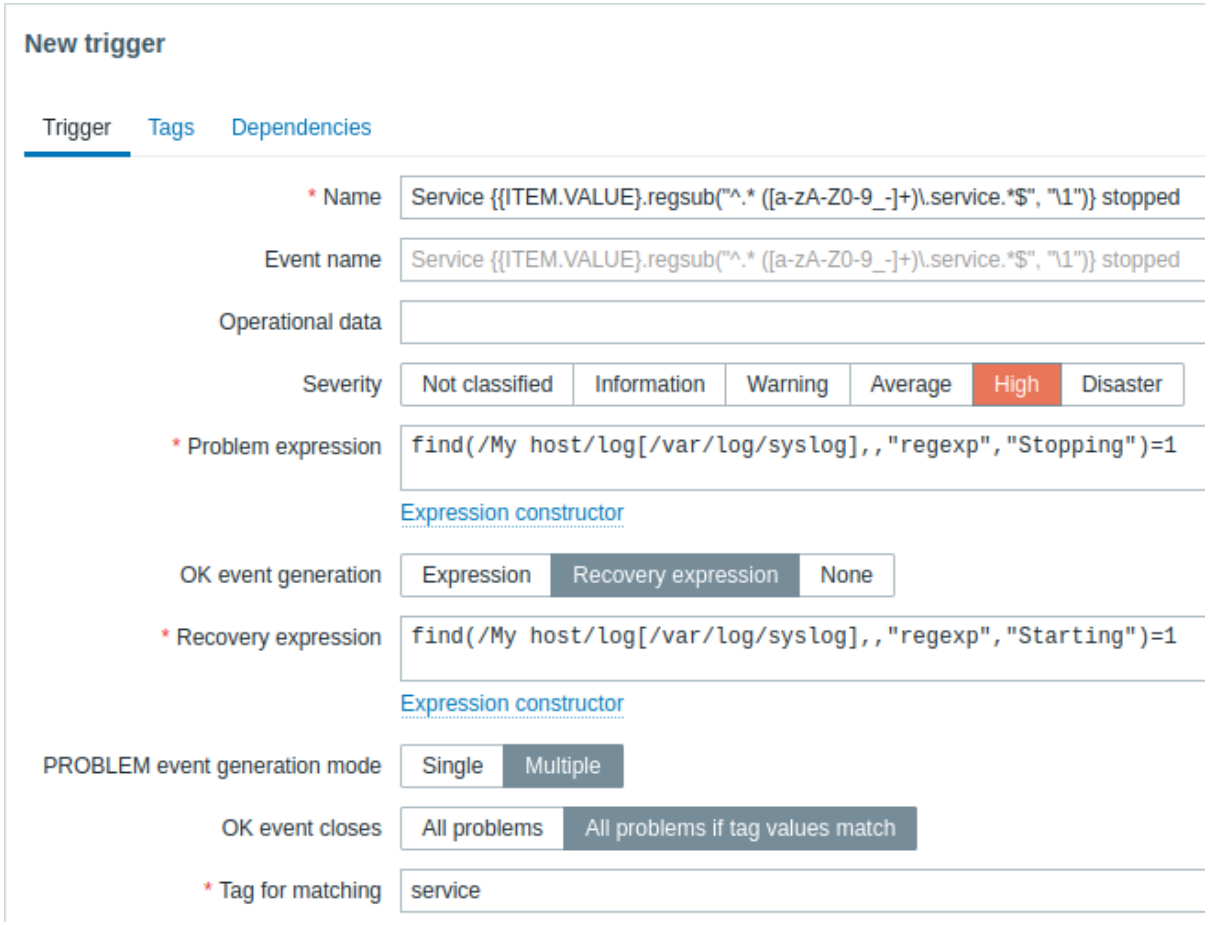
Em seguida, defina uma expressão de recuperação. A seguinte expressão de recuperação resolverá todos os problemas se for encontrada uma linha de log contendo a string "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

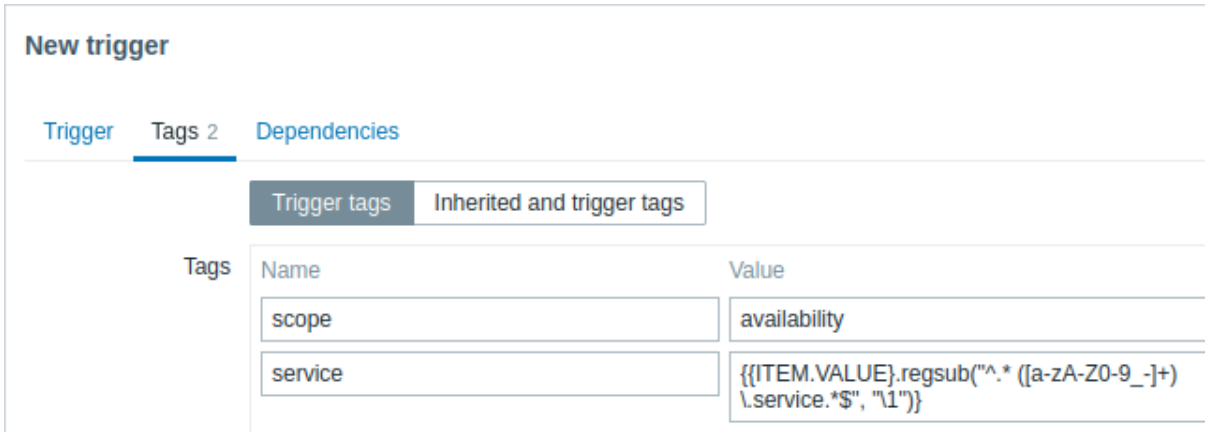
Como não queremos isso, é importante garantir de alguma forma que os problemas raiz correspondentes sejam fechados, e não apenas todos os problemas. É aí que a marcação pode ajudar.

Problemas e resoluções podem ser associados especificando uma tag na configuração do trigger. As seguintes configurações precisam ser feitas:

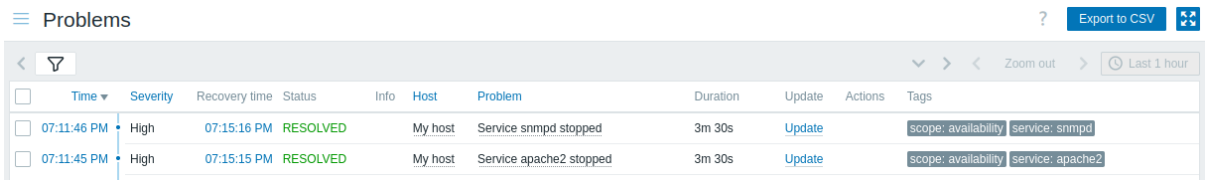
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- Informe o nome da tag para correspondência de eventos



- configure as tags para extrair valores de tag das linhas de log



Se configurado corretamente, você poderá ver eventos de problema marcados por aplicação e associados à sua resolução em *Monitoring* > *Problems*.



Warning:
Como uma configuração incorreta é possível, quando tags de eventos semelhantes podem ser criadas para problemas **não relacionados**, revise os casos descritos abaixo!

- Macros indexadas sempre se referem ao campo *Expression* da configuração do trigger, e não à *Recovery expression*. Por exemplo, em um evento de recuperação, **{ITEM.VALUE1}** será resolvido para o valor mais recente do primeiro item **na expressão do problema** no momento da recuperação. Se a expressão de recuperação for baseada em um item diferente e o valor do item da expressão do problema mudar até o momento da recuperação, os eventos terão tags diferentes e não

serão correlacionados.

- Com duas aplicações gravando mensagens de erro e de recuperação no mesmo arquivo de log, um usuário pode decidir usar duas tags de *service* no mesmo trigger com valores de tag diferentes, usando expressões regulares separadas nos valores das tags para extrair os nomes, por exemplo, do service A e do service 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 gerarão valores de tag vazios, e um único valor de tag vazio tanto em eventos de problema quanto em eventos OK é suficiente para correlacioná-los. Assim, uma mensagem de recuperação do service A pode encerrar acidentalmente uma mensagem de erro do service B.
- As tags e os valores de tag reais só se tornam visíveis quando um trigger dispara. Se a expressão regular usada for inválida, ela será silenciosamente substituída pela string *UNKNOWN*. Se o evento de problema inicial com o valor de tag *UNKNOWN* for perdido, podem aparecer eventos OK subsequentes com o mesmo valor de tag *UNKNOWN* que podem encerrar eventos de problema que não deveriam ter sido encerrados.
- Se um usuário usar a macro {ITEM.VALUE} sem funções de macro como o 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.

Correlação global de eventos

Visão geral

A correlação global de eventos permite abranger 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 evitar milhares de notificações repetitivas e concentrar-se nas causas-raiz de um problema!

A correlação global de eventos é um mecanismo avançado que permite desvincular-se da lógica de problemas e resoluções baseada em um único trigger. Até agora, um único evento de problema era criado por um trigger, e dependíamos desse mesmo trigger para resolver o problema. Não era possível resolver um problema criado por um trigger com outro trigger. Mas isso é possível com a correlação de eventos baseada em tags de eventos.

Por exemplo, um trigger de log pode relatar problemas na aplicação, enquanto um trigger de polling pode relatar que a aplicação está funcionando. Aproveitando as tags de eventos, você pode marcar o trigger de log como *status:down* e o trigger de polling como *status:up*. Em seguida, 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 fosse possível obter apenas um relatório de problema por problema de porta de rede? Não haveria 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 associados aos eventos de problema existentes e o que fazer em caso de correspondência (fechar o novo evento ou fechar os eventos antigos correspondentes gerando os respectivos eventos OK). Se um problema for fechado pela correlação global, isso será informado na coluna *Info* de *Monitoring* > *Problems*.

A configuração de regras de correlação global está disponível somente para usuários com nível de 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 configurada incorretamente, fechar mais eventos do que o pretendido (no pior caso, até mesmo todos os eventos de problema poderiam ser fechados).

Para configurar regras de correlação global com **segurança**, observe as seguintes recomendações importantes:

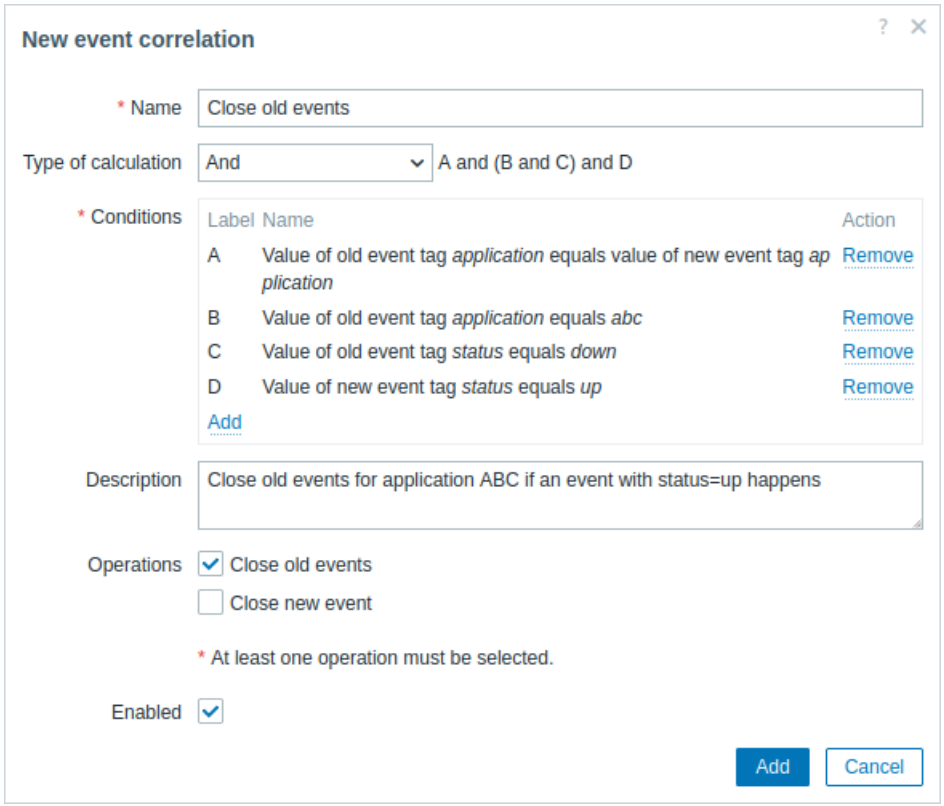
- Reduza o escopo da correlação. Sempre defina uma tag exclusiva para o novo evento associado a eventos antigos e use a condição de correlação *New event tag name*.
- Adicione uma condição explícita para eventos antigos ao usar **Close old events**. Sempre adicione pelo menos uma condição de *Old event* (por exemplo, *Old event tag name*, *Old event tag value* ou *Event tag pair*) ao selecionar **Close old events**; caso contrário, a regra poderá corresponder e fechar eventos de problema existentes não relacionados (no pior caso, todos os problemas). Prefira *Event tag pair* para fazer a correspondência de valores em tempo de execução (host:port, ID de sessão etc.) e restrinja ainda mais a correspondência por host ou grupo de hosts sempre que possível.
- Evite usar nomes de tags comuns que possam acabar sendo usados por diferentes configurações de correlação.
- Mantenha o número de regras de correlação limitado àquelas de que você realmente precisa.

Veja também: known issues.

Configuração

Para configurar regras de correlação de eventos globalmente:

1. Vá para *Data collection* > *Event correlation*.
2. Clique em *Create event correlation* à direita (ou no nome da correlação para editar uma regra existente).
3. Insira os parâmetros da regra de correlação no formulário.

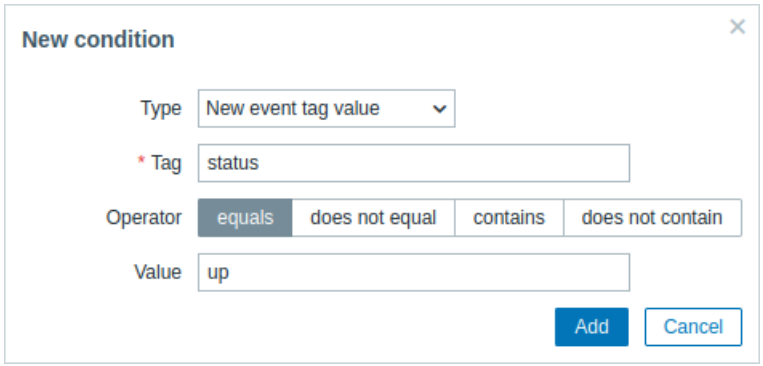


Todos os campos obrigatórios são marcados com um asterisco vermelho.

Parameter	Description
Name	Nome exclusivo da regra de correlação.
Type of calculation	As seguintes opções de cálculo de condições estão disponíveis: **And** - todas as condições devem ser atendidas **Or** - basta que uma condição seja atendida **And/Or** - AND com diferentes tipos de condição e OR com o mesmo tipo de condição **Custom expression** - uma fórmula de cálculo definida pelo usuário para avaliar as condições da action. Ela deve incluir todas as condições (representadas por letras maiúsculas A, B, C, ...) e pode incluir espaços, tabulações, parênteses ( ), **and** (case sensitive), **or** (case sensitive), **not** (case sensitive).
Conditions	Lista de condições. Veja abaixo os detalhes sobre como configurar uma condição.
Description	Descrição da regra de correlação.
Operations	Marque a caixa de seleção da operação a ser executada quando o evento for correlacionado. As seguintes operações estão disponíveis: **Close old events** - fecha eventos antigos quando um novo evento acontece. Sempre adicione uma condição baseada no evento antigo ao usar a operação *Close old events*, caso contrário todos os problemas existentes poderão ser fechados. **Close new event** - fecha o novo evento quando ele acontece. **Aviso!** Não deixe as condições de evento antigo/novo vazias ao usar *Close old events*/*Close new event*. Se você selecionar a operação *Close old events* sem adicionar uma condição que corresponda ao evento antigo, o Zabbix poderá corresponder a todos os eventos antigos existentes e fechá-los. Sempre adicione uma condição explícita de evento antigo (por exemplo, *Old event tag name* ou *Event tag pair*) ao usar *Close old events*. Por exemplo, uma regra que use apenas uma *New event condition* e a operação *Close old events* corresponderá a todos os eventos antigos que atendam aos critérios de evento antigo (ausentes) - fechando, na prática, problemas antigos.

Parameter	Description
Enabled	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 Conditions. Uma janela pop-up será aberta, onde você poderá editar os detalhes da condição.



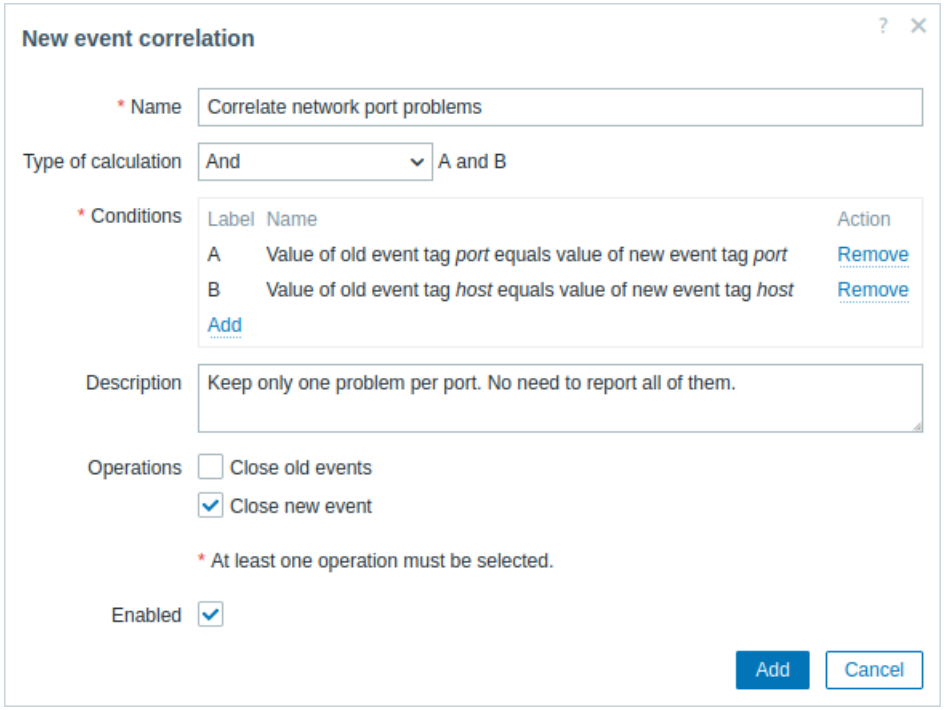
Parameter	Description
New condition	Selecione uma condição para correlacionar eventos. *Observação* que, se nenhuma condição de evento antigo for especificada, todos os eventos antigos poderão ser correspondidos e fechados. Da mesma forma, se nenhuma condição de evento novo for especificada, todos os eventos novos poderão ser correspondidos e fechados. As seguintes condições estão disponíveis: **Old event tag name** - especifique o nome da tag do evento antigo para correspondência. **New event tag name** - especifique o nome da tag do evento novo para correspondência. **New event host group** - especifique o grupo de hosts do evento novo para correspondência. **Event tag pair** - especifique o nome da tag do evento novo e o nome da tag do evento antigo para correspondência. Nesse caso, haverá 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 Example). **Old event tag value** - especifique o nome e o valor da tag do evento antigo para correspondência, usando os seguintes operadores: *equals* - tem o valor da tag do evento antigo *does not equal* - não tem o valor da tag do evento antigo *contains* - contém a string no valor da tag do evento antigo *does not contain* - não contém a string no valor da tag do evento antigo **New event tag value** - especifique o nome e o valor da tag do evento novo para correspondência, usando os seguintes operadores: *equals* - tem o valor da tag do evento novo *does not equal* - não tem o valor da tag do evento novo *contains* - contém a string no valor da tag do evento novo *does not contain* - não contém a string no valor da tag do evento novo

Warning:
Como é possível haver configuração incorreta, quando tags de evento semelhantes puderem ser criadas para problemas **não relacionados**, revise os casos descritos abaixo!

- As tags e os valores de tag reais só se tornam visíveis quando um trigger dispara. Se a expressão regular usada for inválida, ela é substituída silenciosamente 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.

Exemplo

Evite eventos de problema repetitivos da mesma porta de rede.



Esta regra global de correlação correlacionará problemas se os valores das tags *host* e *port* existirem no trigger e forem iguais no evento original e no novo.

A operação fechará novos eventos de problema na mesma porta de rede, mantendo apenas o problema original aberto.

Descoberta

Visão geral A descoberta fornece maneiras automáticas de adicionar hosts e configurar o monitoramento sem configuração manual. O Zabbix oferece suporte a vários métodos de descoberta para se adaptar a diferentes cenários.

Métodos de descoberta

- Descoberta de rede - detecta automaticamente hosts e serviços disponíveis na sua rede.
- Autorregistro de agent ativo - permite que agents ativos se registrem automaticamente no server.
- Descoberta de baixo nível - cria automaticamente itens, triggers, gráficos e hosts para entidades descobertas, como sistemas de arquivos ou interfaces de rede, usando protótipos e regras de descoberta aninhadas.

Descoberta de rede

Visão geral

O Zabbix oferece funcionalidade de descoberta automática de rede que é eficaz e muito flexível.

Com a descoberta de rede configurada corretamente, 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 (somente o modo não criptografado é suportado)
- Informações recebidas do agent SNMP

Ela NÃO fornece descoberta da topologia de rede.

A descoberta de rede basicamente consiste em duas fases: descoberta e ações.

Descoberta

O Zabbix verifica periodicamente os intervalos de IP definidos nas regras de descoberta de rede. A frequência da verificação é configurável individualmente para cada regra.

Cada regra tem um conjunto de verificações de serviço definido para ser executado no intervalo de IP.

As regras de descoberta são processadas pelo discovery manager. O discovery manager cria um job para cada regra com uma lista de tarefas (verificações de rede). As verificações de rede são executadas em paralelo pelos discovery workers disponíveis (o número é configurável no frontend para cada regra). A exceção são as verificações SNMPv3, que são processadas por um único worker. Somente verificações com o mesmo IP e porta são agendadas sequencialmente, porque alguns dispositivos não aceitam conexões paralelas na mesma porta.

O tamanho da fila das 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 `zabbix[discovery_queue]` item interno 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 de serviço
Service Discovered	O serviço está 'up' após estar 'down' ou quando é descoberto pela primeira vez.
Service Up	O serviço está 'up', após já estar 'up'.
Service Lost	O serviço está 'down' após estar 'up'.
Service Down	O serviço está 'down', após já estar 'down'.
Host Discovered	Pelo menos um serviço de um host está 'up' após todos os serviços desse host estarem 'down' ou um serviço é descoberto e pertence a um host não registrado.
Host Up	Pelo menos um serviço de um host está 'up', após pelo menos um serviço já estar 'up'.
Host Lost	Todos os serviços de um host estão 'down' após pelo menos um estar 'up'.
Host Down	Todos os serviços de um host estão 'down', após já estarem 'down'.

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 operação 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 *Status da descoberta: up* para evitar que ações como *Adicionar host* sejam acionadas em eventos *Serviço perdido*/*Serviço inativo*. Caso contrário, se um host descoberto for removido manualmente, ele ainda gerará eventos *Serviço perdido*/*Serviço inativo* 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 *Add host* estiver selecionada. Um host também é adicionado, mesmo que a operação *Add host* esteja ausente, se você selecionar operações que resultem em ações em um host. Essas operações são:

- Enable host.
- Disable host.
- Add host to a host group.
- Link template to a host.

Os hosts criados são adicionados ao grupo *Discovered hosts* (por padrão, configurável em *Administration* > *General* > *Other*). Se você quiser que os hosts sejam adicionados a outro grupo, adicione uma operação *Remove from host groups* (especificando ”Discovered hosts”) e também adicione uma operação *Add to host groups* (especificando outro host group), porque um host deve pertencer a um host group.

O endereço IP do dispositivo descoberto, juntamente com a origem da descoberta (Zabbix server, Zabbix proxy ou grupo de proxy) e o tipo de interface, é usado como critério para localizar um host no sistema. Se já existir um host com o mesmo endereço IP, tipo de interface e origem da descoberta, esse host será o alvo para a execução das operações. Se a origem da descoberta for diferente, a entidade descoberta será tratada como um host diferente e um novo host poderá ser criado.

Se o endereço IP do host descoberto for alterado ou a interface for excluída, um novo host será criado na próxima descoberta.

Nomeação de hosts

Ao adicionar hosts, o nome do host é o resultado de uma consulta reversa de DNS ou o endereço IP se a consulta reversa falhar. A consulta é realizada a partir do Zabbix server ou Zabbix proxy, dependendo de qual está realizando a descoberta. Se a consulta 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 consulta 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 agente SNMP 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 e a fonte de descoberta (Zabbix server, proxy ou grupo de proxy) não tiver mudado, um novo host não é criado. Se a fonte de descoberta for diferente, a entidade descoberta é tratada como distinta e um novo host pode ser criado. No entanto, se a ação de descoberta contiver operações (vincular template, adicionar ao grupo de hosts, etc), elas serão realizadas no host existente que corresponde pelo endereço IP, tipo de interface e fonte de descoberta.

Remoção de host

Hosts descobertos por uma regra de descoberta de rede são removidos automaticamente de *Monitoramento* > *Descoberta* se uma entidade descoberta não estiver mais no intervalo de IPs da regra. Os hosts são removidos imediatamente.

Criação de interfaces ao adicionar hosts

Quando hosts são adicionados como resultado da descoberta de rede, as interfaces são criadas de acordo com estas regras:

- Os serviços detectados - por exemplo, se uma verificação SNMP foi bem-sucedida, uma interface SNMP será criada.
- Se um host respondeu tanto a solicitações do Zabbix agent quanto do SNMP, ambos os tipos de interfaces serão criados.
- Se os critérios de unicidade forem dados retornados pelo Zabbix agent ou pelo SNMP, a primeira interface encontrada para um host será criada como a padrão. Outros endereços IP serão ignorados, a menos que a interface padrão fique indisponível. No caso de a interface padrão ficar indisponível, o menor dos outros endereços IP será usado como substituto da interface padrão. As condições da ação (como Host IP) 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 de agent, ele será criado somente com uma interface agent. Se ele começar a responder a SNMP posteriormente, interfaces SNMP adicionais serão adicionadas.
- Se 3 hosts separados foram inicialmente criados, tendo sido descobertos pelos critérios de unicidade ”IP”, e depois a regra de descoberta for modificada de modo que os hosts A, B e C tenham o mesmo resultado de critérios de unicidade, B e C serão criados como interfaces adicionais para A, o primeiro host. Os hosts individuais B e C permanecem. Em *Monitoring* > *Discovery*, as interfaces adicionadas serão exibidas na coluna ”Discovered device”, em fonte preta e com recuo, mas a coluna ”Monitored host” exibirá apenas A, o primeiro host criado. ”Uptime/Downtime” não é medido para IPs que são considerados interfaces adicionais.

Alterando a configuração de proxy

Os hosts descobertos por diferentes proxies nem sempre são tratados como hosts diferentes. As verificações de descoberta e de unicidade dependem da estrutura do grupo de proxy: quando um proxy executa uma regra de descoberta e cria um host, esse host é adicionado ao grupo de proxy pai do proxy, e não atribuído ao próprio proxy. Quando o Zabbix avalia a unicidade de IP durante a descoberta, ele verifica os hosts monitorados pelo grupo de proxy pai. Hosts monitorados por proxies individuais dentro desse grupo (incluindo o proxy que executou a descoberta) são ignorados na verificação de unicidade, o que pode resultar em hosts duplicados se vários proxies monitorarem sub-redes sobrepostas.

Embora esse comportamento permita que a descoberta opere em intervalos de IP sobrepostos usados por diferentes sub-redes, alterar o proxy atribuído a uma sub-rede já monitorada é mais complexo, porque as alterações de proxy devem ser aplicadas de forma consistente aos hosts descobertos e à associação ao grupo de proxy pai para evitar duplicidades.

Por exemplo, os passos para substituir o proxy em uma regra de descoberta:

1. Desative a regra de descoberta.
2. Sincronize a configuração do proxy.
3. Substitua o proxy na regra de descoberta.
4. Substitua o proxy para todos os hosts descobertos por esta regra (garanta que os hosts no grupo de proxy pai e quaisquer hosts monitorados por proxies individuais nesse grupo sejam atualizados para evitar duplicidades).
5. Ative a regra de descoberta.

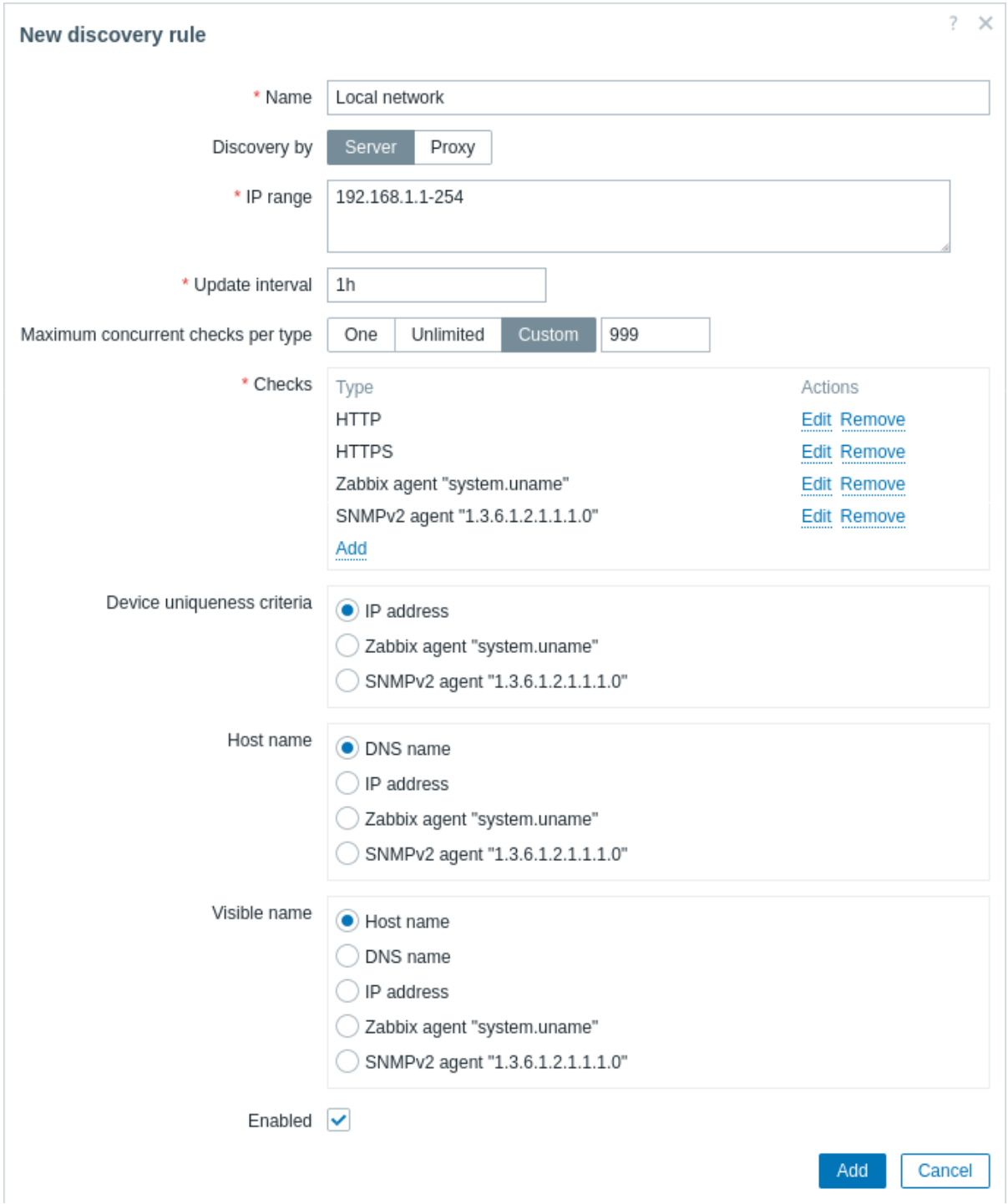
Configurando uma regra de descoberta de rede

Visão geral

Para configurar uma regra de descoberta de rede usada pelo Zabbix para descobrir hosts e serviços:

1. Vá para *Data collection* > *Discovery*.
2. Clique em *Create discovery rule* (ou no nome da regra para editar uma existente).
3. Edite os atributos da regra de descoberta.

Atributos da regra



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Parameter	Description
Name	Nome exclusivo da regra. Por exemplo, ”Local network”.
Discovery by	A descoberta é realizada por: **Server** - pelo Zabbix server **Proxy** - pelo Zabbix proxy (selecionado no campo de nome do proxy)
IP range	O intervalo de endereços IP para descoberta. Ele 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 IP: 192.168.4.0/24 máscaras 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 várias linhas.

Parameter	Description
Update interval	Este parâmetro define com que frequência o Zabbix executará a regra. O intervalo é medido após o término da execução da instância anterior de descoberta, portanto não há sobreposição. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são suportadas. *Observação* 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 do exemplo).
Maximum concurrent checks per type	Defina o número máximo de threads de descoberta (workers) por verificação de serviço para processamento paralelo das verificações de descoberta: **One** - uma thread **Unlimited** - número ilimitado de threads (mas não mais do que no parâmetro StartDiscoverers) **Custom** - defina um número personalizado de threads (0-999). As verificações de serviço assíncronas SNMPv1 e SNMPv2 são processadas simultaneamente pelos workers disponíveis. Todas as regras de descoberta com verificações de serviço assíncronas SNMPv3 são processadas por um worker devido às peculiaridades da implementação da libsnmp; portanto, aumentar o número de workers não aumentará a velocidade da descoberta.
Checks	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 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 'Ports' 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 executadas 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, então as verificações HTTP funcionarão como nas versões anteriores (ou seja, como verificações TCP), mas as verificações HTTPS não funcionarão.
Device uniqueness criteria	Os critérios de unicidade podem ser: **IP address** - não processe vários dispositivos de IP único. Se já existir um dispositivo com o mesmo IP, ele será considerado já descoberto e um novo host não será adicionado. **<discovery check>** - verificação do Zabbix agent ou do SNMP agent. Observe que os critérios de unicidade usados durante a descoberta não são os mesmos da identificação de host no sistema ao executar ações. Os critérios de unicidade durante a descoberta definem se dois ou mais dispositivos descobertos são iguais (ou diferentes), enquanto apenas o endereço IP é o critério para identificação de host no Zabbix (consulte Criação de host).
Host name	Defina o nome técnico do host criado usando: **DNS name** - nome DNS (padrão) **IP address** - endereço IP **<discovery check>** - valor de string recebido da verificação de descoberta (por exemplo, Zabbix agent, verificação do SNMP agent) Veja também: Nomeação de host.
Visible name	Defina o nome visível do host criado usando: **Host name** - nome técnico do host (padrão) **DNS name** - nome DNS **IP address** - endereço IP **<discovery check>** - valor de string recebido da verificação de descoberta (por exemplo, Zabbix agent, verificação do SNMP agent) Veja também: Nomeação de host.
Enabled	Com a caixa de seleção marcada, a regra fica ativa e será executada pelo Zabbix server. Se desmarcada, a regra não fica ativa. Ela não será executada.

Excedendo o limite de descritores de arquivo

No caso de um grande número de verificações simultâneas, é possível esgotar o limite de descritores de arquivo para o discovery

manager.

O número de descritores de arquivo necessários para a detecção equivale ao número de discovery workers * 1000. Por padrão, há 5 discovery workers, enquanto o limite flexível do sistema é de aproximadamente 1024.

Se esse limite estiver próximo de ser atingido, o Zabbix reduzirá o número padrão de verificações simultâneas por tipo para cada worker e gravará um aviso no arquivo de log. Mas, se o usuário tiver definido um valor maior para *Maximum concurrent checks per type* 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.

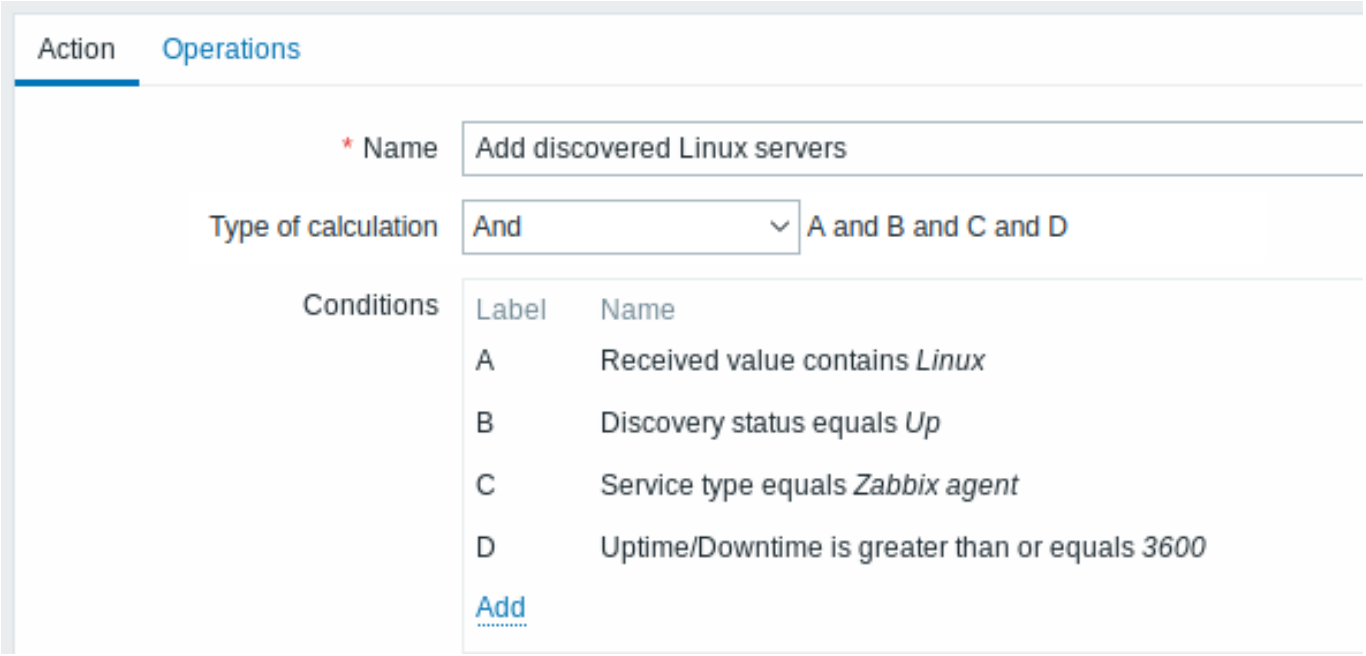
O Zabbix tentará descobrir hosts no intervalo de IPs 192.168.1.1-192.168.1.254 conectando-se aos agents do Zabbix e obtendo o valor da chave **system.uname**. O valor recebido do agent pode ser usado para nomear os hosts e também para aplicar diferentes ações para diferentes sistemas operacionais. Por 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.

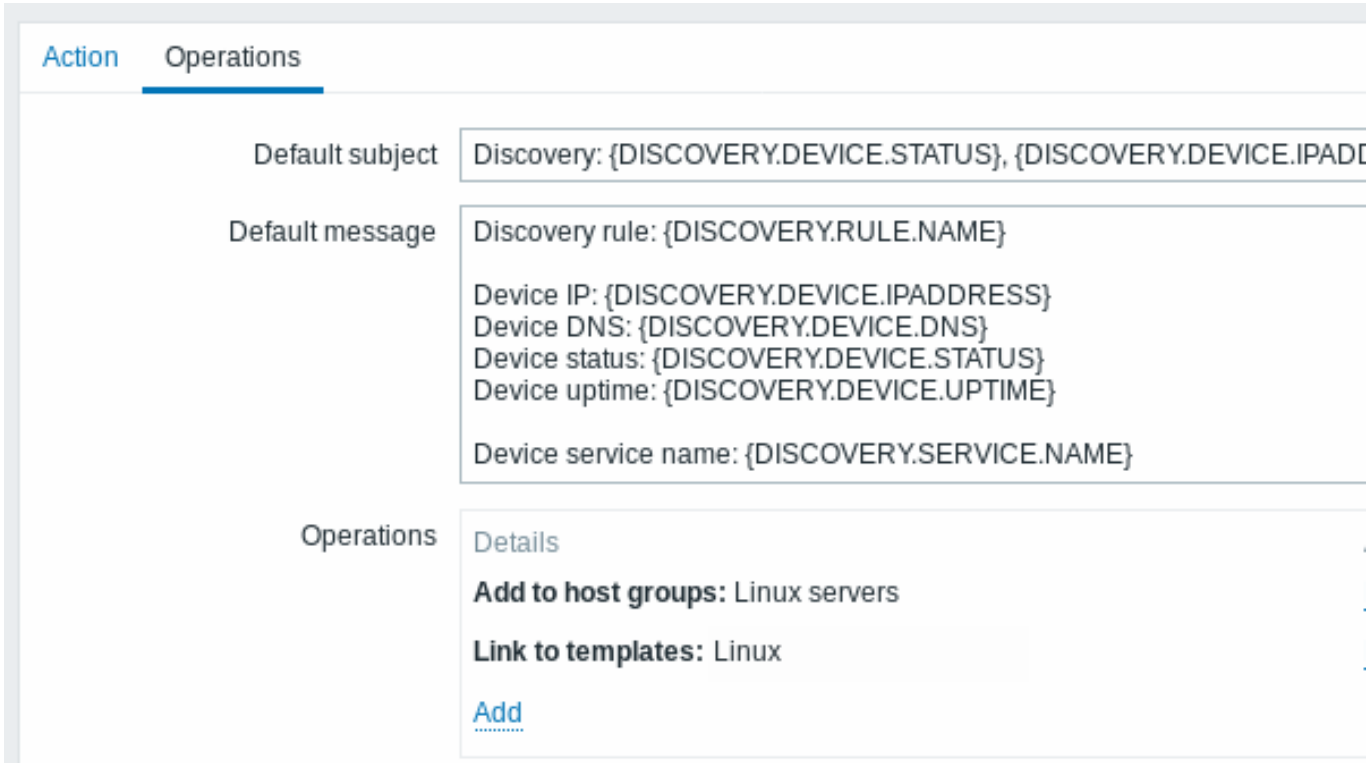
Passo 2

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:

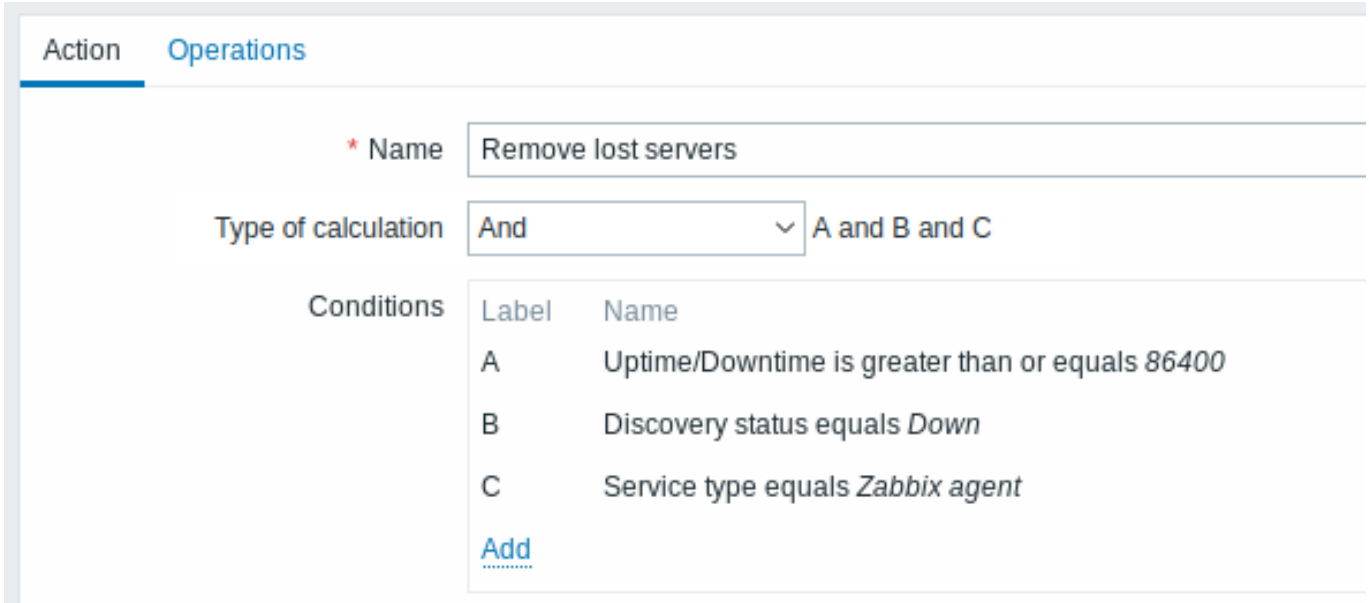
- adicionar o host descoberto ao grupo ”Linux servers” (e também adicionar o host se ele não tiver sido adicionado anteriormente)
- vincular o host ao *Linux* template. O Zabbix começará automaticamente a monitorar o host usando os items e triggers do template ”Linux”.

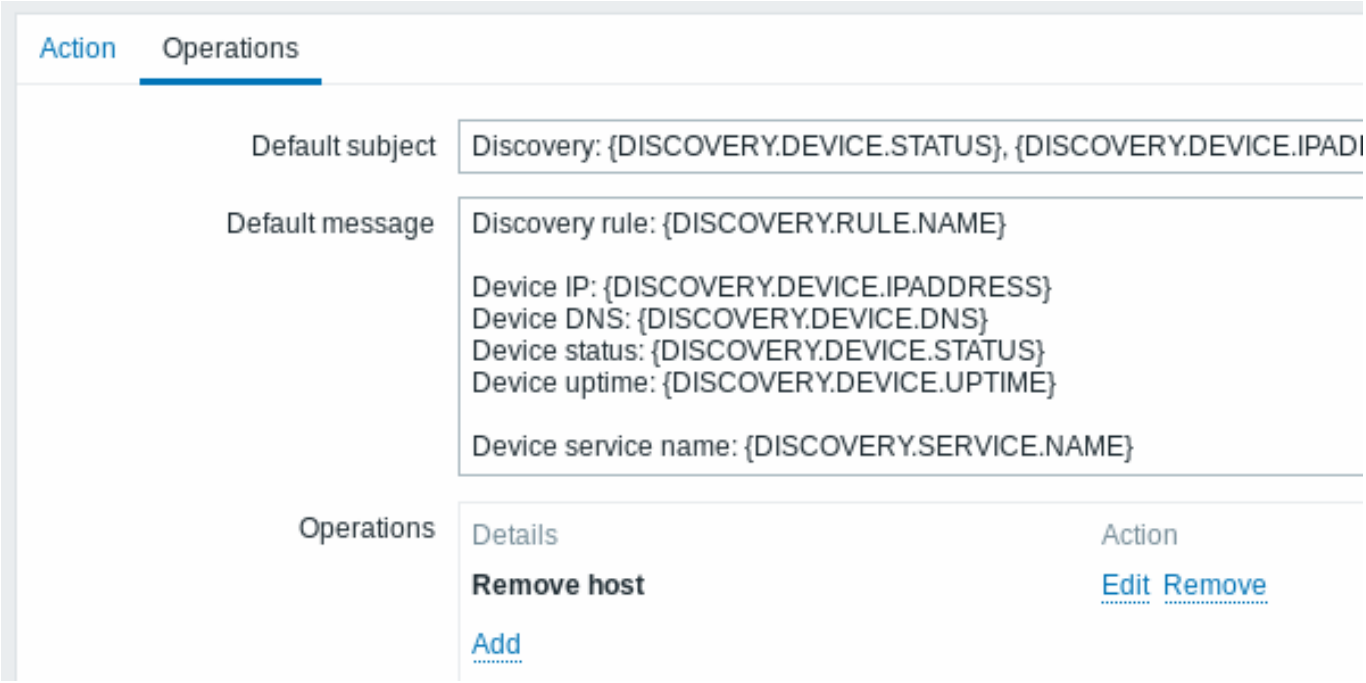
Passo 3

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).

Autoregistro do agent ativo

Visão geral

É possível permitir a autoregistração do agent ativo do Zabbix, após o que o server pode começar a monitorá-los. Dessa forma, novos hosts podem ser adicionados para monitoramento sem configurá-los manualmente no server.

A autoregistração pode ocorrer quando um agent ativo anteriormente desconhecido solicita verificações.

O recurso pode ser muito útil para o monitoramento automático de novos nós na nuvem. Assim que você tiver um novo nó no Zabbix Cloud, a coleta de dados de desempenho e disponibilidade do host será iniciada automaticamente.

A autoregistração do agent ativo também oferece suporte ao monitoramento de hosts adicionados com verificações passivas. Quando o agent ativo solicita verificações, desde que os parâmetros de configuração `ListenIP` ou `ListenPort` estejam definidos no arquivo de configuração, eles são enviados ao server. Se vários endereços IP forem especificados, o primeiro é enviado ao server.

O server, ao adicionar o novo host autoregistrado, usa o endereço IP e a porta recebidos para configurar o agent. Se nenhum valor de endereço IP for recebido, será usado o endereço IP da conexão de entrada. Se nenhum valor de porta for recebido, 10050 será usado.

É possível especificar que o host deve ser autoregistrado com um nome DNS como interface padrão do agent.

A autoregistração é executada novamente:

- se as informações de metadados do host forem alteradas:
  - devido à alteração de HostMetadata e reinicialização do agent
  - devido à alteração do valor retornado por HostMetadataItem
- para hosts criados manualmente com metadados ausentes
- se um host for alterado manualmente para ser monitorado por outro Zabbix proxy
- se a autoregistração do mesmo host vier de um novo Zabbix proxy

O heartbeat da autoregistração do agent ativo para o Zabbix server e o Zabbix proxy é de 120 segundos. Portanto, caso um host descoberto seja excluído, a autoregistração será executada novamente em 120 segundos.

Configuração

Especificar o server

Certifique-se de que o server do Zabbix esteja identificado no arquivo de configuração do agent (`zabbix_agentd.conf`):

```
ServerActive=10.0.0.1
```

A menos que você defina especificamente um *Hostname* em `zabbix_agentd.conf`, o nome do host do sistema onde o agent está localizado será usado pelo server para nomear o host. No Linux, o nome do host do sistema pode ser obtido executando o comando `hostname`.

Se *Hostname* for definido na configuração do agent do Zabbix como uma lista de hosts separada por vírgulas, serão criados hosts para todos os nomes de host listados.

Reinicie o agent após fazer qualquer alteração no arquivo de configuração.

Ação para o registro automático de agent ativo

Quando o server recebe uma solicitação de registro automático de um agent, ele chama uma ação. Uma ação com a origem do evento "Registro automático" deve ser configurada para o registro automático de agent.

> **Note:**
> Não é necessário configurar a descoberta de rede para que os agents ativos sejam registrados automaticamente.

No frontend do Zabbix:

1. Acesse *Alertas* > *Ações*.
2. Selecione *Ações de registro automático*.
3. Clique em *Criar ação*.
4. Na aba *Ação*, dê um nome à sua ação.
5. Opcionalmente, especifique condições. É possível usar correspondência de substring ou de expressão regular nas condições para nome do host/metadados do host. Se você pretende usar o tipo de condição *Metadados do host*, consulte Usando metadados do host.
6. Na aba *Operações*, adicione as operações relevantes, como *Adicionar host*, *Adicionar ao grupo de hosts* (por exemplo, *Hosts descobertos*), *Vincular templates* etc.

> **Note:**
> Se for provável que os hosts que serão registrados automaticamente sejam compatíveis apenas com o monitoramento ativo (como hosts protegidos por firewall em relação ao seu Zabbix server), talvez seja interessante criar um template específico, como *Template_Linux-active*, para vinculá-lo.

Os hosts criados são adicionados ao grupo *Hosts descobertos* (por padrão, configurável em *Administração* > *Geral* > *Outros*). Se quiser que os hosts sejam adicionados a outro grupo, adicione uma operação *Remover do grupo de hosts* (especificando "Hosts descobertos") e também uma operação *Adicionar ao grupo de hosts* (especificando outro grupo de hosts), pois um host deve pertencer a um grupo de hosts.

Autoregistro seguro

É possível realizar o autoregistro de forma segura configurando a autenticação baseada em PSK com conexões criptografadas.

O nível de criptografia é configurado globalmente em *Administração* > *Geral* > *Autoregistro*. É possível selecionar nenhuma criptografia, criptografia TLS com autenticação PSK ou ambas (para que alguns hosts possam se registrar sem criptografia, enquanto outros utilizem criptografia).

A autenticação por PSK é verificada pelo Zabbix server antes de adicionar um host. Se for bem-sucedida, o host será adicionado e *Conexões de/para o host* serão definidas como 'PSK', usando a mesma identidade/chave pré-compartilhada configurada na definição global de autoregistro.

> **Attention:**
> Para garantir a segurança do autoregistro em instalações que utilizam proxies, a criptografia entre o Zabbix server e o proxy deve ser habilitada.

Usando DNS como interface padrão

HostInterface e HostInterfaceItem parâmetros de configuração permitem especificar um valor personalizado para a interface do host durante o registro automático.

Mais especificamente, eles são úteis se o host precisar ser registrado automaticamente com um nome DNS como interface padrão do agent, em vez de seu endereço IP. Nesse caso, o nome DNS deve ser especificado ou retornado como o valor de um dos parâmetros HostInterface ou HostInterfaceItem. Se o valor de um desses parâmetros for alterado — por exemplo, de um endereço IP para um nome DNS ou vice-versa —, a interface padrão do host registrado automaticamente será atualizada de acordo. Essa atualização é aplicada ao host existente, e não por meio da criação de um novo host. Para enviar o novo valor, o agent deve ser reiniciado para que o processo de registro automático seja iniciado novamente.

Note:
Se os parâmetros HostInterface ou HostInterfaceItem não estiverem configurados, o parâmetro listen_dns será usado. Esse valor é determinado por meio de uma consulta DNS reversa do endereço IP do agent. Se o DNS reverso não estiver configurado corretamente ou retornar um nome inválido, isso poderá resultar em um registro automático incorreto ou malsucedido devido a um valor de interface inválido.

Usando metadados do host

Quando o agent envia uma solicitação de registro automático ao server, ele envia o nome do host. Em alguns casos (por exemplo, nós de nuvem da Amazon), um nome de host não é suficiente para que o server do Zabbix diferencie os hosts descobertos. Os metadados do host podem ser usados opcionalmente para enviar outras informações de um agent para o server.

Os metadados do host são configurados no arquivo de configuração do agent (`zabbix_agentd.conf`). Há 2 maneiras de especificar os metadados do host no arquivo de configuração:

```
HostMetadata
HostMetadataItem
```

Consulte 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 registro automático ocorre sempre que um agent ativo envia uma solicitação ao server para atualizar as verificações ativas. O intervalo entre as solicitações é especificado no parâmetro RefreshActiveChecks do agent. A primeira solicitação é enviada imediatamente após a reinicialização do agent.

Exemplos

Autoregistro por SO usando HostMetadata

Suponha que você queira que os hosts sejam 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 tem os templates ”Linux by Zabbix agent” e ”Windows by Zabbix agent” disponíveis em seu frontend Zabbix. Portanto, no registro do host, você gostaria que o template Linux/Windows apropriado fosse aplicado ao host que está sendo registrado. Por padrão, apenas o nome do host é enviado ao server no autoregistro, o que pode não ser suficiente. Para garantir que o template correto seja aplicado ao host, você deve usar metadados do host.

Configuração do frontend

A primeira coisa a fazer é configurar o frontend. Crie 2 ações. A primeira ação:

- Nome: Autoregistro de host Linux
- Condições: Metadados do host contém *Linux*
- Operações: Vincular templates: Linux pelo agent Zabbix

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 pelo agent Zabbix

Configuração do agent

Agora você precisa configurar os agents. Adicione a seguinte linha aos arquivos de configuração do agent:

```
HostMetadataItem=system.uname
```

Dessa forma, você garante que os metadados do host conterão ”Linux” ou ”Windows”, dependendo do host em que o agent está sendo executado. Um exemplo de metadados do host neste caso:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-0PXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

Usando HostMetadata para controlar o registro automático e os templates

Etapa 1 - Proteger o registro com HostMetadata

Usando metadados do host para permitir alguma proteção básica contra o registro de hosts indesejados.

Configuração do frontend

Crie uma ação no frontend, usando algum código secreto difícil de adivinhar para impedir hosts indesejados:

- Nome: Ação de autorregistro Linux
- Condições:
  - Tipo de cálculo: AND
  - Condição (A): Metadados do host contém //Linux//
  - Condição (B): Metadados do host contém //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Operações:
  - Enviar mensagem para usuários: Admin via todos os meios
  - Adicionar aos grupos de hosts: Servidores Linux
  - Associar templates: Linux pelo agent Zabbix

Observe que este método sozinho não fornece proteção forte porque os dados são transmitidos em texto simples. É necessário recarregar o cache de configuração para que as alterações tenham efeito imediato.

Configuração do agent

Adicione a seguinte linha ao arquivo de configuração do agent:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

onde ”Linux” é uma plataforma e o restante da string é o texto secreto difícil de adivinhar.

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

Etapa 2 - Adicionar template ao host registrado

É possível adicionar templates adicionais para um host já registrado. Neste caso, o template MySQL by Zabbix agent será vinculado apenas aos hosts cujo HostMetadata contenha o token `MySQL`.

Configuração do frontend

Atualize a ação no frontend:

- Nome: Autoregistration action Linux
- Condições:
  - Tipo de cálculo: AND
  - Condição (A): Host metadata contém Linux
  - Condição (B): Host metadata contém 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Condição (C): Host metadata contém MySQL
- Operações:
  - Enviar mensagem para usuários: Admin via todos os meios
  - Adicionar aos grupos de hosts: Linux servers
  - Associar templates: Linux by Zabbix agent
  - Associar templates: MySQL by Zabbix Agent

Configuração do agent

Atualize a seguinte linha no arquivo de configuração do agent:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

Descoberta de baixo nível

Visão geral A descoberta de baixo nível (LLD) descobre automaticamente entidades (por exemplo, sistemas de arquivos, interfaces de rede) em um host e cria os itens, triggers e gráficos correspondentes sem configuração manual para cada uma.

Para usar a LLD, você cria uma **regra de descoberta** que coleta dados JSON descrevendo entidades e **prototypes** (item, trigger e graph) para cada entidade.

O JSON retornado pela regra deve ser um array de objetos, em que cada objeto representa uma entidade descoberta usando pares chave-valor. Por exemplo, uma regra de descoberta com `net.if.discovery` pode retornar:

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Um conjunto correspondente de itens, triggers e gráficos é então criado a partir dos prototypes — um por interface (`lo` e `eth0`).

Se os objetos contiverem chaves no formato `{#MACRO}`, essas macros são usadas diretamente nos prototypes. Se não, ou se macros adicionais/personalizadas forem necessárias, as macros podem ser definidas manualmente e mapeadas para campos JSON usando JSONPath.

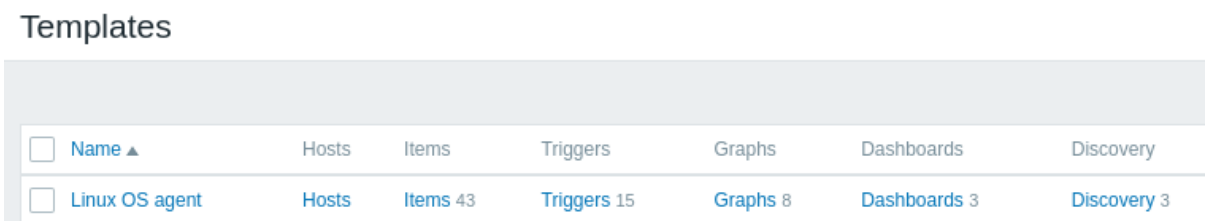
A LLD também pode criar hosts (por exemplo, para máquinas virtuais descobertas em um hypervisor) e oferece suporte a regras de descoberta aninhadas para descoberta em vários níveis.

Entidades que não forem mais descobertas podem ser desabilitadas ou excluídas automaticamente.

Configurando 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:

1. Vá para *Data collection* > *Templates* ou *Hosts*.
2. Clique em *Discovery* na linha do template/host apropriado.



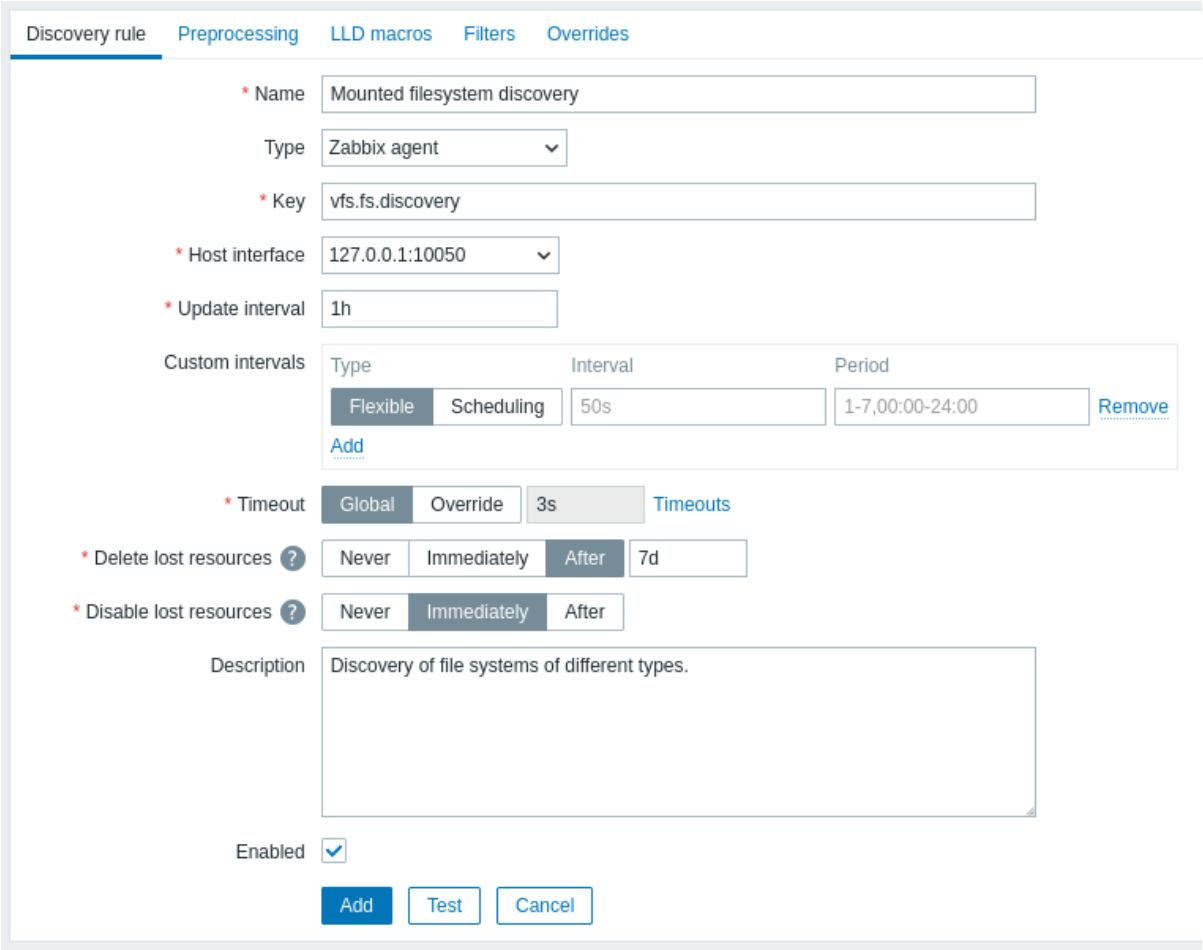
3. Clique em *Create discovery rule* no canto superior direito da tela.
4. Preencha o formulário da regra de descoberta com os detalhes necessários.

Regra de descoberta

O formulário da regra de descoberta contém cinco abas que representam, da esquerda para a direita, o fluxo de dados durante a descoberta:

- *Regra de descoberta* - especifique, principalmente, o item integrado ou o script personalizado para recuperar os dados de descoberta.
- *Pré-processamento* - aplique algum pré-processamento aos dados descobertos.
- *Macros de LLD* - permita a extração de valores de macros para uso em itens descobertos, triggers etc.
- *Filtros* - permita a filtragem dos valores descobertos.
- *Substituições* - permita a modificação de itens, triggers, gráficos ou protótipos de host ao aplicar alterações a objetos descobertos específicos.

A aba **Regra de descoberta** contém a chave do item a ser usada para a descoberta (bem como alguns atributos gerais da regra de descoberta):



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome da regra de descoberta.
Tipo	O tipo de verificação a ser executada para 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. Ela não pode depender de outra regra de descoberta. Para um item dependente, selecione o tipo correspondente (*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 integrada "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 a descoberta de sistemas de arquivos é usar os resultados da descoberta pela chave de agent "vfs.fs.get" (consulte o exemplo).
Intervalo de atualização	Este campo especifica com que frequência o Zabbix executa a descoberta. No início, quando você estiver configurando a descoberta do sistema de arquivos, talvez queira definir um intervalo pequeno, mas, depois de confirmar que ela funciona, poderá defini-lo como 30 minutos ou mais, pois os sistemas de arquivos geralmente não mudam com muita frequência. Sufixos de tempo são compatíveis, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são compatíveis. *Observação*: o intervalo de atualização só pode ser definido como '0' se existirem intervalos personalizados com um valor diferente de zero. Se definido como '0' e existir um intervalo personalizado (flexível ou agendado) com um valor diferente de zero, o item será consultado durante a duração do intervalo personalizado. Novas regras de descoberta serão verificadas em até 60 segundos após sua criação, a menos que tenham um intervalo de atualização de Agendamento ou Flexível e o *Intervalo de atualização* esteja definido como 0. *Observação*: para uma regra de descoberta existente, a descoberta pode ser executada 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 personalizado de consultas. Para obter informações detalhadas, consulte Intervalos personalizados.
Tempo limite	Defina o tempo limite da verificação de descoberta. Selecione a opção de tempo limite: **Global** - o tempo limite do proxy/global é usado (exibido no campo *Tempo limite* esmaecido); **Substituição** - um tempo limite personalizado é usado (definido no campo *Tempo limite*; intervalo permitido: 1 - 600s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são compatíveis. Clicar no link *Tempos limite* permite configurar os tempos limite do proxy ou os tempos limite globais (se um proxy não for usado). Observe que o link *Tempos limite* fica 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 depois que seu status de descoberta se tornar "Não descoberta": *Nunca* - ela não será excluída; *Imediatamente* - ela será excluída imediatamente; *Após* - ela será excluída depois do período especificado. O valor deve ser maior que o valor de *Desabilitar recursos perdidos*. Sufixos de tempo são compatíveis, por exemplo, 2h, 1d. Macros de usuário são compatíveis. *Observação*: não é recomendável usar "Imediatamente", pois editar o filtro incorretamente pode resultar na exclusão da entidade com todos os dados históricos. Observe que os recursos desabilitados manualmente não serão excluídos pela descoberta de baixo nível.
Desabilitar recursos perdidos	Especifique em quanto tempo a entidade descoberta será desabilitada depois que seu status de descoberta se tornar "Não descoberta": *Nunca* - ela não será desabilitada; *Imediatamente* - ela será desabilitada imediatamente; *Após* - ela será desabilitada depois do período especificado. O valor deve ser maior que o intervalo de atualização da regra de descoberta. Observe que os recursos desabilitados automaticamente serão habilitados novamente se forem redescobertos pela descoberta de baixo nível. Os recursos desabilitados manualmente não serão habilitados novamente se forem redescobertos. Este campo não será exibido se *Excluir recursos perdidos* estiver definido como "Imediatamente". Sufixos de tempo são compatíveis, por exemplo, 2h, 1d. Macros de usuário são compatíveis.
Descrição	Insira uma descrição.
Habilitado	Se selecionado, a regra será processada.

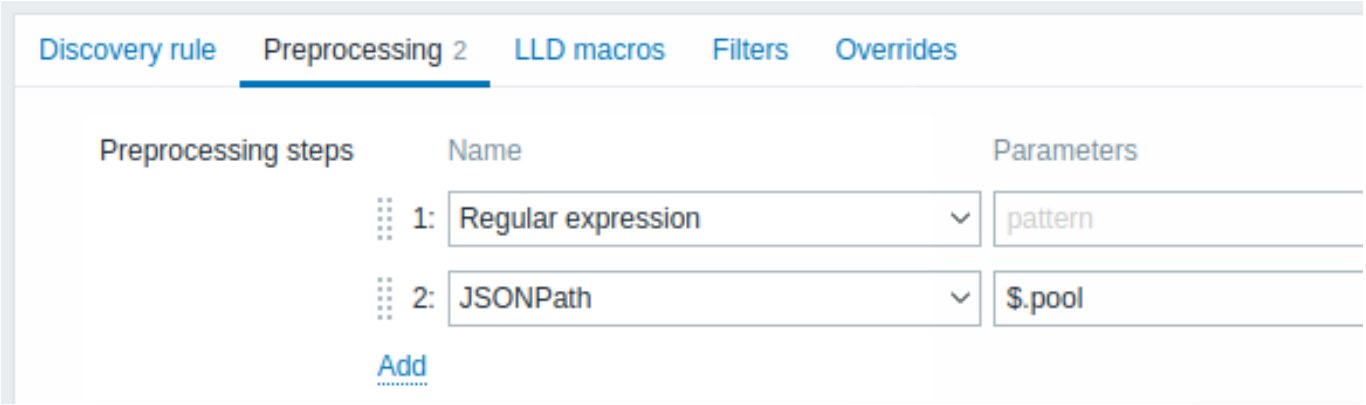
Note:
O histórico da regra de descoberta não é preservado.

Pré-processamento

A aba **Pré-processamento** permite definir regras de transformação a serem aplicadas ao resultado da descoberta. Uma ou várias transformações podem ser usadas nesta etapa. As transformações são executadas na ordem em que são definidas. Todo o pré-processamento é realizado 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 permite a extração de no máximo 10 grupos capturados com a sequência \N. Parâmetros: **pattern** - expressão regular **output** - template de formatação da 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 de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: 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 string (ou por nada). Todas as ocorrências da string de pesquisa serão substituídas. Parâmetros: **search string** - string a ser localizada e substituída, diferenciando maiúsculas de minúsculas (obrigatório) **replacement** - string pela qual a string de pesquisa será substituída. A string de substituição também pode estar vazia, permitindo efetivamente que a string de pesquisa seja excluída quando encontrada. É possível usar sequências de escape para pesquisar ou substituir quebras de linha, retornos 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, retornos de carro e tabulações é feito automaticamente durante a descoberta de baixo nível.
Dados estruturados		
	JSONPath	Extrai um valor ou fragmento de dados JSON usando a funcionalidade JSONPath. Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

	XML XPath	Extrai um 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 compatíveis.
		Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

Tipo	
CSV para JSON	Converte dados de um arquivo CSV para o formato JSON. Para obter mais informações, consulte: Pré-processamento de CSV para JSON.
XML para JSON	Converte dados no formato XML para JSON. Para obter mais informações, consulte: Regras de serialização. Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
SNMP	
Valor da varredura SNMP	Extrai o valor pelo OID/nome MIB especificado e aplica opções de formatação: **Inalterado** - retorna Hex-STRING como uma string hexadecimal sem escape (*observe* que as dicas de exibição ainda são aplicadas); **UTF-8 de Hex-STRING** - converte Hex-STRING em uma string UTF-8; **MAC de Hex-STRING** - converte Hex-STRING em uma string de endereço MAC (na qual `' '` será 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 de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Varredura SNMP 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 com os valores no caminho OID SNMP especificado. Você pode usar esta etapa de pré-processamento para a descoberta de OID SNMP. Estão disponíveis opções de formatação de valores semelhantes às da etapa *Valor da varredura SNMP*. Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Valor obtido por SNMP	Aplica opções de formatação ao valor obtido por SNMP: **UTF-8 de Hex-STRING** - converte Hex-STRING em uma string UTF-8; **MAC de Hex-STRING** - converte Hex-STRING em uma string de endereço MAC (na qual `' '` será 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 de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Scripts personalizados	
JavaScript	Insira 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 tamanho disponível do JavaScript depende do banco de dados usado. Para obter mais informações, consulte: Pré-processamento Javascript
Validação	
Não corresponde à expressão regular	Especifique uma expressão regular à qual um valor não deve corresponder. Por exemplo, `Error:(.*?)\.` Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

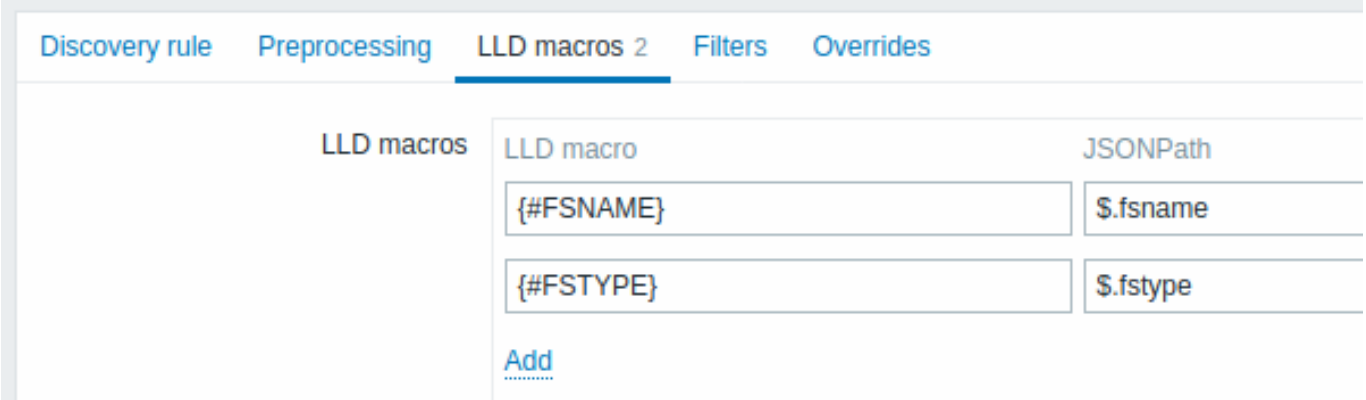
Tipo	
Verificar erro em JSON	Verifica uma mensagem de erro no nível da aplicação localizada no JSONPath. Interrompe o processamento se for encontrada uma mensagem não vazia; caso contrário, continua o processamento com o valor existente antes desta etapa de pré-processamento. Observe que esses erros de serviços externos são relatados ao usuário como estão, sem adicionar informações sobre a etapa de pré-processamento. Por exemplo, `$.errors`. Se um JSON como `{"errors":"e1"}` for recebido, a próxima etapa de pré-processamento não será executada. Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Verificar erro em XML	Verifica uma mensagem de erro no nível da aplicação localizada no Xpath. Interrompe o processamento se for encontrada uma mensagem não vazia; caso contrário, continua o processamento com o valor existente antes desta etapa de pré-processamento. Observe que esses erros de serviços externos são relatados ao usuário como estão, sem adicionar informações sobre a etapa de pré-processamento. Nenhum erro será relatado caso não seja possível analisar XML inválido. Se você marcar a caixa de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: 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 de seleção *Personalizado em caso de falha*, será possível especificar opções personalizadas de tratamento de erros: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Limitação	
Descartar valores inalterados com heartbeat	Descarta um valor se ele não tiver sido alterado durante o período definido (em segundos). Valores inteiros positivos são compatíveis 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. Por exemplo, `1m`. Se um texto idêntico for passado duas vezes para esta regra em um intervalo de 60 segundos, ele será descartado. *Observação*: alterar protótipos de item não redefine a limitação. A limitação é redefinida somente quando as etapas de pré-processamento são alteradas.
Prometheus	
Prometheus para JSON	Converte as métricas Prometheus necessárias para JSON. Consulte Verificações do Prometheus para obter mais detalhes.

Observe que, se a regra de descoberta tiver sido aplicada ao host por meio de um template, o conteúdo desta aba será somente leitura.

Macros personalizadas

A aba **LLD macros** permite especificar LLD macros personalizadas quando o JSON retornado ainda não contém chaves no formato `{#MACRO}`.

Por exemplo, a chave de item `vfs.fs.get` do agent retorna dados de filesystem em JSON sem nenhuma LLD macro pre-definida.
Aqui, você define as macros manualmente e as mapeia para valores no JSON usando expressões JSONPath:



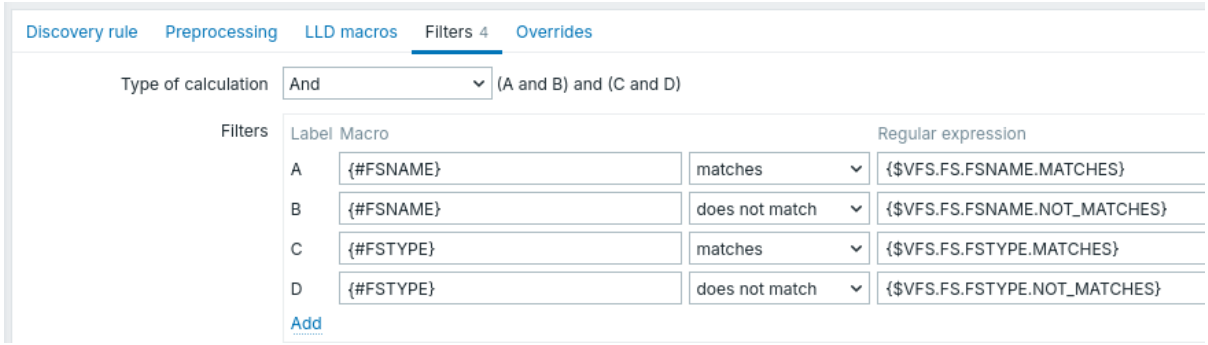
Parameter	Description
LLD macro	Nome da LLD macro no formato `{#MACRO}`.
JSONPath	Expressão que define qual valor extrair de cada entidade descoberta no array JSON. Por exemplo, `$.foo` extrai `"bar"` e `"baz"` de `[{"foo":"bar"}, {"foo":"baz"}]`. JSONPath oferece suporte tanto à notação com ponto quanto à notação com colchetes. A notação com colchetes deve ser usada para chaves que contenham caracteres especiais ou Unicode, por exemplo, `$['unicode + special chars #1']['unicode + special chars #2']`.

A regra de descoberta deve retornar um array JSON na raiz (`$.`).
Por compatibilidade com versões anteriores, também é aceito um objeto JSON contendo um único array `"data"`. Nesse caso, o array `"data"` é automaticamente desembrulhado:

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Filtro

Um filtro pode ser usado para gerar itens, triggers e gráficos reais somente para entidades que correspondam aos critérios.
A aba **Filtros** contém definições de filtros de regras de descoberta que permitem filtrar valores de descoberta:



Parâmetro	Descrição
Tipo de cálculo	As seguintes opções para calcular filtros estão disponíveis: **And** - todos os filtros devem ser aprovados; **Or** - basta que um filtro seja aprovado; **And/Or** - usa *And* com nomes de macros diferentes e *Or* com o mesmo nome de macro; **Expressão personalizada** - oferece a possibilidade de definir um cálculo personalizado dos filtros. A fórmula deve incluir todos os filtros da lista. Limitada a 255 símbolos.
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). Você pode inserir uma expressão regular ou fazer referência a uma expressão regular global no campo *Expressão regular*. 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 - nenhum aviso.

Por exemplo, se você estiver interessado apenas nos sistemas de arquivos C:, D: e E:, poderá inserir `{#FSNAME}` no campo *Macro* e `^C|^D|^E` no campo *Expressão regular*.
 Também é possível filtrar por tipos de sistema de arquivos usando a macro `{#FSTYPE}` (por exemplo, `^ext|^reiserfs`) e por tipos de unidade (compatível somente com o Windows agent) usando a macro `{#FSDRIVETYPE}` (por exemplo, `fixed`).

Para testar uma expressão regular, você pode usar `grep -E`, por exemplo:

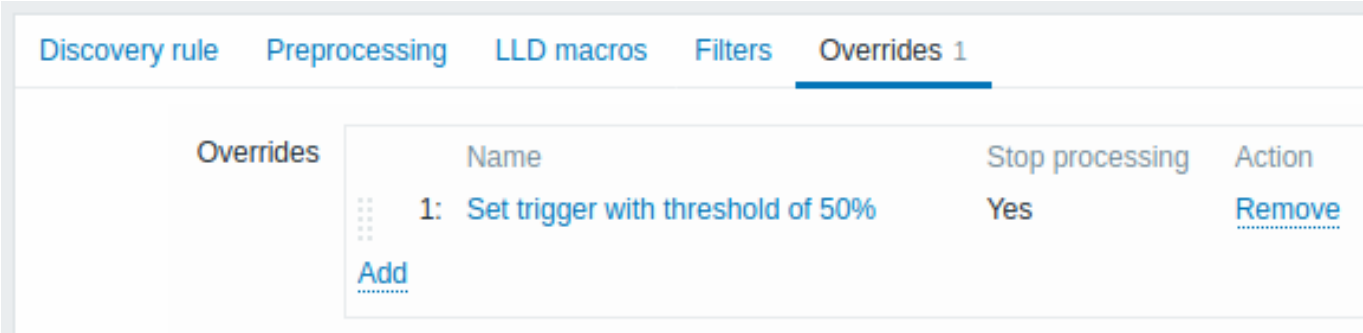
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Um erro ou uma digitação incorreta 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 de muitos hosts.

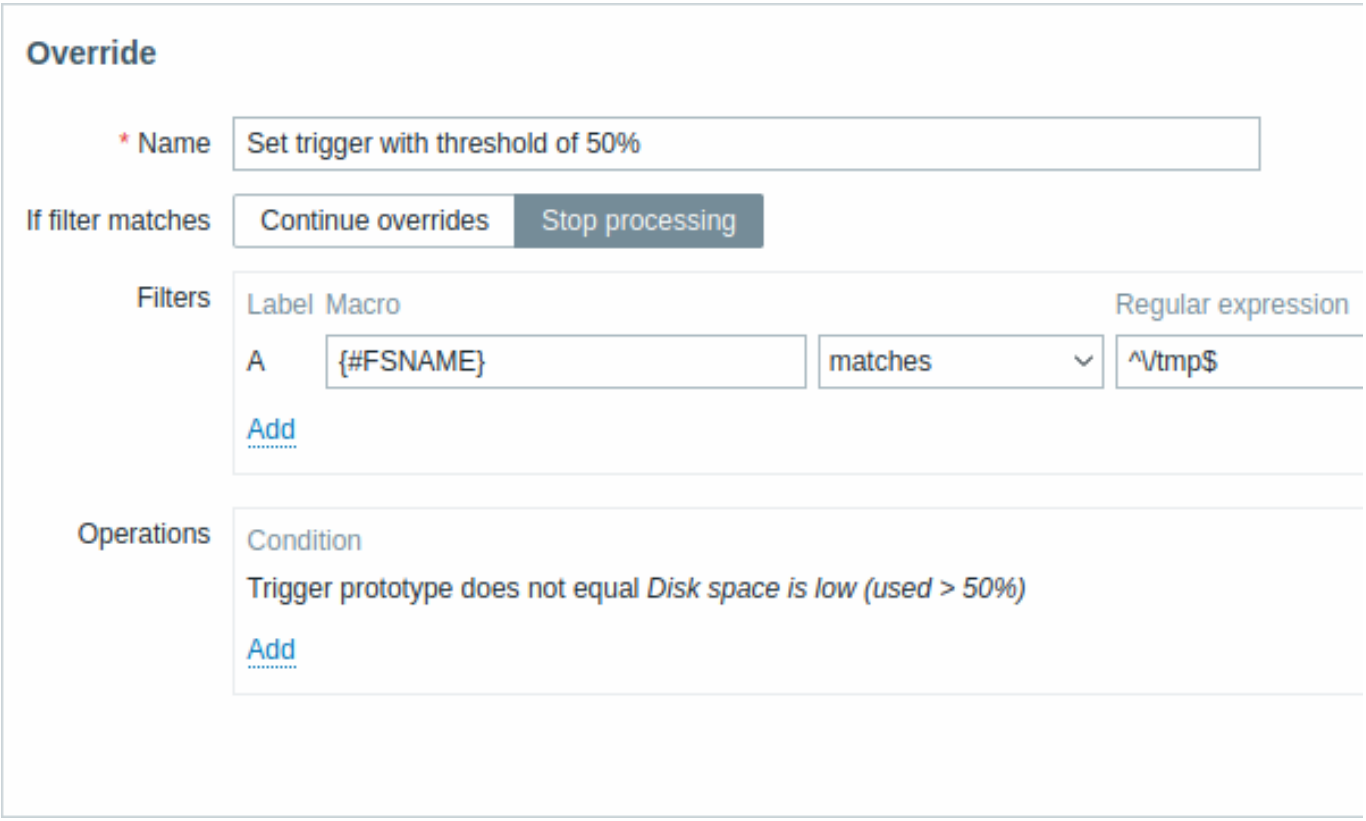
Attention:
O banco de dados do Zabbix no MySQL deve ser criado com diferenciação entre maiúsculas e minúsculas para que os nomes de sistemas de arquivos que diferem apenas pela capitalização sejam descobertos corretamente.

Substituição

A aba **Substituições** permite definir regras para modificar a lista de protótipos de item, trigger, gráfico, host e descoberta, ou seus atributos, para objetos descobertos que atendam aos critérios especificados.



As substituições (se houver) são exibidas em uma lista reordenável por arrastar e soltar e executadas na ordem em que são definidas. Para configurar os detalhes de uma nova substituição, clique em Add no bloco *Substituições*. Para editar uma substituição existente, clique no nome da substituição. Uma janela pop-up será aberta, permitindo editar os detalhes da regra de substituição.

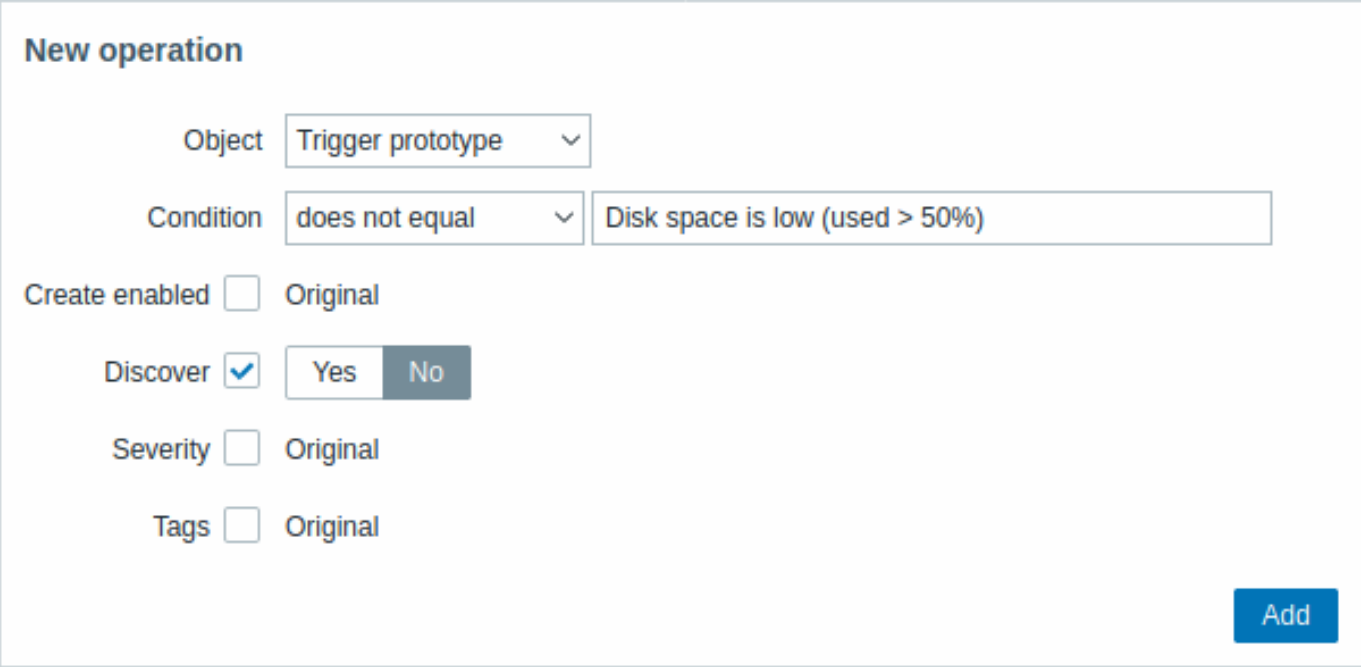


Todos os parâmetros obrigatórios são marcados com asteriscos vermelhos.

Parâmetro	Descrição
Nome	Um nome exclusivo (por regra de LLD) para a substituição.
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. **Interromper processamento** - as operações anteriores (se houver) e desta substituição serão executadas; as substituições subsequentes serão ignoradas para as linhas de LLD correspondentes.
Filtros	Determina a quais entidades descobertas a substituição deve ser aplicada. Os filtros de substituição são processados após os filtros da regra de descoberta e têm a mesma funcionalidade.
Operações	As operações de substituição são exibidas com estes detalhes: **Condição** - um tipo de objeto e uma condição a ser atendida para o nome do objeto; por exemplo: o protótipo de trigger não é igual a *Disk space is low (used > 50%)*. **Ações** - são exibidos links para editar e remover uma operação.

Configurando uma operação

Para configurar os detalhes de uma nova operação, clique em Add no bloco Operações. Para editar uma operação existente, clique em Edit ao lado da operação. Uma janela pop-up será aberta, permitindo editar os detalhes da operação.



Parâmetro	Descrição
Objeto	Cinco 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 compatíveis: **é igual a** - aplicar a este protótipo **não é igual a** - aplicar a todos os protótipos, exceto este **contém** - aplicar se o nome do protótipo contiver esta string **não contém** - aplicar se o nome do protótipo não contiver esta string **corresponde** - aplicar se o nome do protótipo corresponder à expressão regular **não corresponde** - aplicar se o nome do protótipo não corresponder à expressão regular
Padrão	Uma expressão regular ou uma string a ser pesquisada.

Parâmetro	Descrição
Objeto: *Protótipo de item*	
Criar habilitado	Quando a caixa de seleção é marcada, os botões aparecem, 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 é marcada, os botões aparecem, 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 é marcada, duas opções aparecem, permitindo definir um intervalo diferente para o item: *Atraso* - intervalo de atualização do item. Macros de usuário, macros de LLD e sufixos de tempo (por exemplo, 30s, 1m, 2h, 1d) são compatíveis. Deve ser definido como 0 se *Intervalo personalizado* for usado. *Intervalo personalizado* - clique em Add para especificar intervalos flexíveis/agendados. Para obter informações detalhadas, consulte Intervalos personalizados.
Histórico	Quando a caixa de seleção é marcada, os botões aparecem, permitindo definir períodos de armazenamento de histórico diferentes para o item: *Não armazenar* - se selecionado, o histórico não será armazenado. *Armazenar por até* - se selecionado, um campo de entrada para especificar o período de armazenamento aparecerá à direita. Macros de usuário e macros de LLD são compatíveis.
Tendências	Quando a caixa de seleção é marcada, os botões aparecem, permitindo definir períodos de armazenamento de tendências diferentes para o item: *Não armazenar* - se selecionado, as tendências não serão armazenadas. *Armazenar por até* - se selecionado, um campo de entrada para especificar o período de armazenamento aparecerá à direita. Macros de usuário e macros de LLD são compatíveis.
Tags	Quando a caixa de seleção é marcada, um novo bloco aparece, permitindo especificar pares de tag e valor. Macros de usuário e macros de LLD são compatíveis. Essas tags serão anexadas às tags especificadas no protótipo de item, mesmo que os nomes das tags sejam iguais.
Objeto: *Protótipo de trigger*	
Criar habilitado	Quando a caixa de seleção é marcada, os botões aparecem, 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 é marcada, os botões aparecem, 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 é marcada, os botões de severidade do trigger aparecem, permitindo modificar a severidade do trigger.
Tags	Quando a caixa de seleção é marcada, um novo bloco aparece, permitindo especificar pares de tag e valor. Macros de usuário e macros de LLD são compatíveis. Essas tags serão anexadas às tags especificadas no protótipo de trigger, mesmo que os nomes das tags sejam iguais.

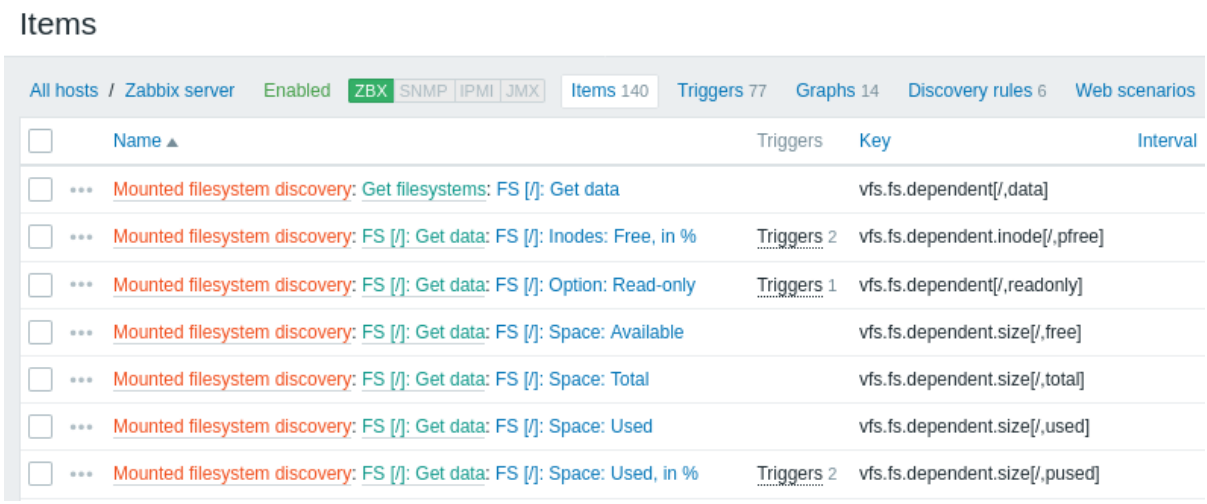
Parâmetro	Descrição
Objeto: *Protótipo de gráfico*	
Descobrir	Quando a caixa de seleção é marcada, os botões aparecem, permitindo substituir as configurações originais do protótipo de gráfico: *Sim* - o gráfico será adicionado. *Não* - o gráfico não será adicionado.
Objeto: *Protótipo de host*	
Criar habilitado	Quando a caixa de seleção é marcada, os botões aparecem, 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 é marcada, os botões aparecem, 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 é marcada, um campo de entrada para especificar templates aparece. Comece a digitar o nome do template ou clique em *Selecionar* ao lado do campo e selecione os templates na lista de 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 é marcada, um novo bloco aparece, permitindo especificar pares de tag e valor. Macros de usuário e macros de LLD são compatíveis. Essas tags serão anexadas às tags especificadas no protótipo de host, mesmo que os nomes das tags sejam iguais.
Inventário do host	Quando a caixa de seleção é marcada, os botões aparecem, permitindo selecionar um modo de inventário diferente para o protótipo de host: *Desabilitado* - não preencher o inventário do host *Manual* - fornecer os detalhes manualmente *Automatizado* - preencher automaticamente os dados do inventário do host com base nas métricas coletadas.
Objeto: *Protótipo de descoberta*	
Criar habilitado	Quando a caixa de seleção é marcada, os botões aparecem, permitindo substituir as configurações originais do protótipo de descoberta: *Sim* - a regra de descoberta será adicionada em estado habilitado. *Não* - a regra de descoberta será adicionada em estado desabilitado.
Descobrir	Quando a caixa de seleção é marcada, os botões aparecem, permitindo substituir as configurações originais do protótipo de descoberta: *Sim* - a regra de descoberta será adicionada. *Não* - a regra de descoberta não será adicionada.
Intervalo de atualização	Quando a caixa de seleção é marcada, duas opções aparecem, permitindo definir um intervalo diferente para a regra de descoberta: *Atraso* - intervalo de atualização da regra. Macros de usuário, macros de LLD e sufixos de tempo (por exemplo, 30s, 1m, 2h, 1d) são compatíveis. Deve ser definido como 0 se *Intervalo personalizado* for usado. *Intervalo personalizado* - clique em  para especificar intervalos flexíveis/agendados. Para obter informações detalhadas, consulte Intervalos personalizados.

Botões do formulário

Os botões na parte inferior do formulário permitem realizar várias operações.

Add	Adicionar uma regra de descoberta. Este botão está disponível apenas para novas regras de descoberta.
Update	Atualizar as propriedades de uma regra de descoberta. Este botão está disponível apenas para regras de descoberta existentes.
Clone	Criar outra regra de descoberta com base nas propriedades da regra de descoberta atual.
Execute now	Realizar a descoberta imediatamente com base na regra de descoberta. A regra de descoberta já deve existir. Consulte mais detalhes. *Observação*: ao realizar 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	Testar a configuração da regra de descoberta. Use este botão para verificar as configurações (como conectividade e correção dos parâmetros) sem aplicar permanentemente nenhuma alteração.
Delete	Excluir a regra de descoberta.
Cancel	Cancelar 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 da qual elas se originam.



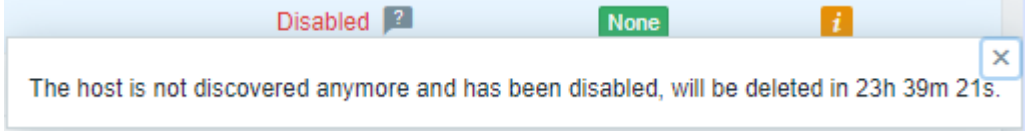
Observe que as entidades descobertas não serão criadas caso já existam entidades com os mesmos critérios de exclusividade, por exemplo, um item com a mesma chave ou um gráfico com o mesmo nome. Uma mensagem de erro é exibida neste caso no frontend informando que a regra de descoberta de baixo nível não pôde criar determinadas entidades. A própria regra de descoberta, no entanto, não ficará como não suportada porque alguma entidade não pôde ser criada e teve que ser ignorada. A regra de descoberta continuará criando/atualizando outras entidades.

Se uma entidade descoberta (host, sistema de arquivos, interface, etc) parar de ser descoberta (ou não passar mais no filtro), as entidades que foram criadas com base nela podem ser automaticamente desativadas e eventualmente excluídas.

Recursos perdidos podem ser automaticamente desativados com base no valor do parâmetro *Desativar recursos perdidos*. Isso afeta hosts, items e triggers perdidos.

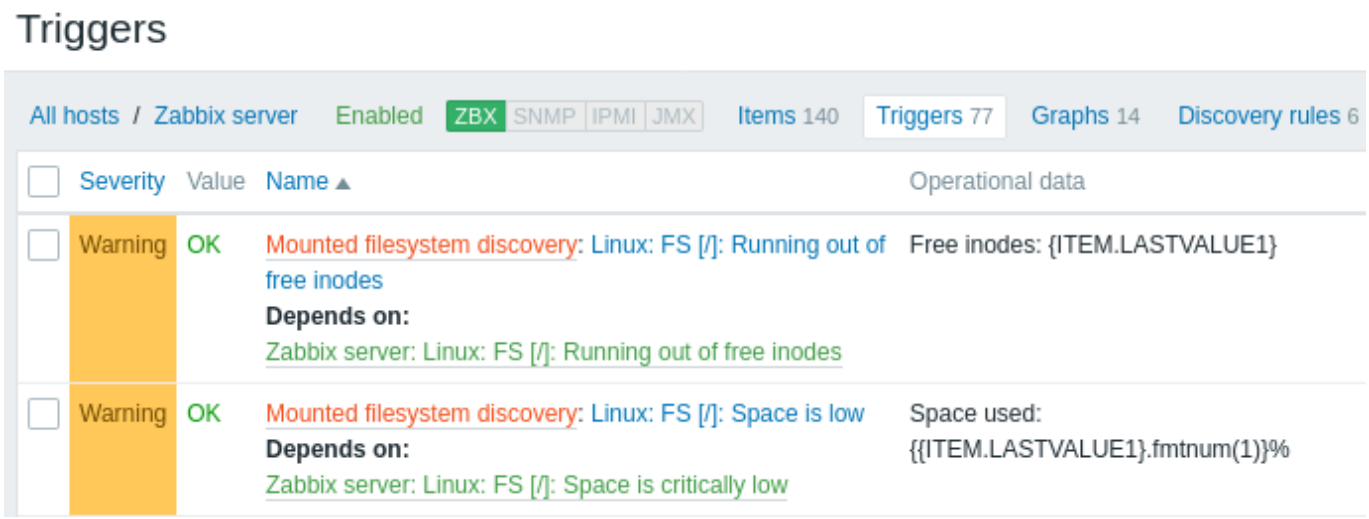
Recursos perdidos podem ser automaticamente excluídos com base no valor do parâmetro *Excluir recursos perdidos*. Isso afeta hosts, grupos de hosts, items, triggers e gráficos perdidos.

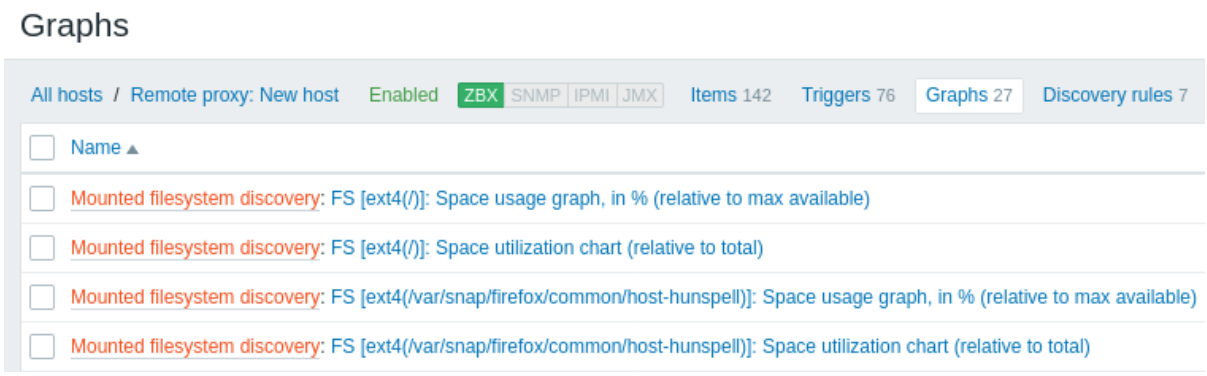
Quando as entidades descobertas se tornam 'Não descobertas mais', um indicador de tempo de vida é exibido na lista de entidades. Mova o ponteiro do mouse sobre ele e uma mensagem será exibida indicando os detalhes do status.



Se as entidades foram marcadas para exclusão, mas não foram excluídas no momento esperado (regra de descoberta ou host do item desativado), elas serão excluídas na próxima vez que a regra de descoberta for processada.

Entidades que contêm outras entidades, que estão marcadas para exclusão, não serão atualizadas se forem alteradas no nível da regra de descoberta. Por exemplo, triggers baseados em LLD não serão atualizados se contiverem items que estão marcados para exclusão.





Outros tipos de descoberta Mais detalhes e tutoriais sobre outros tipos de descoberta prontos para uso estão disponíveis nas seguintes seções:

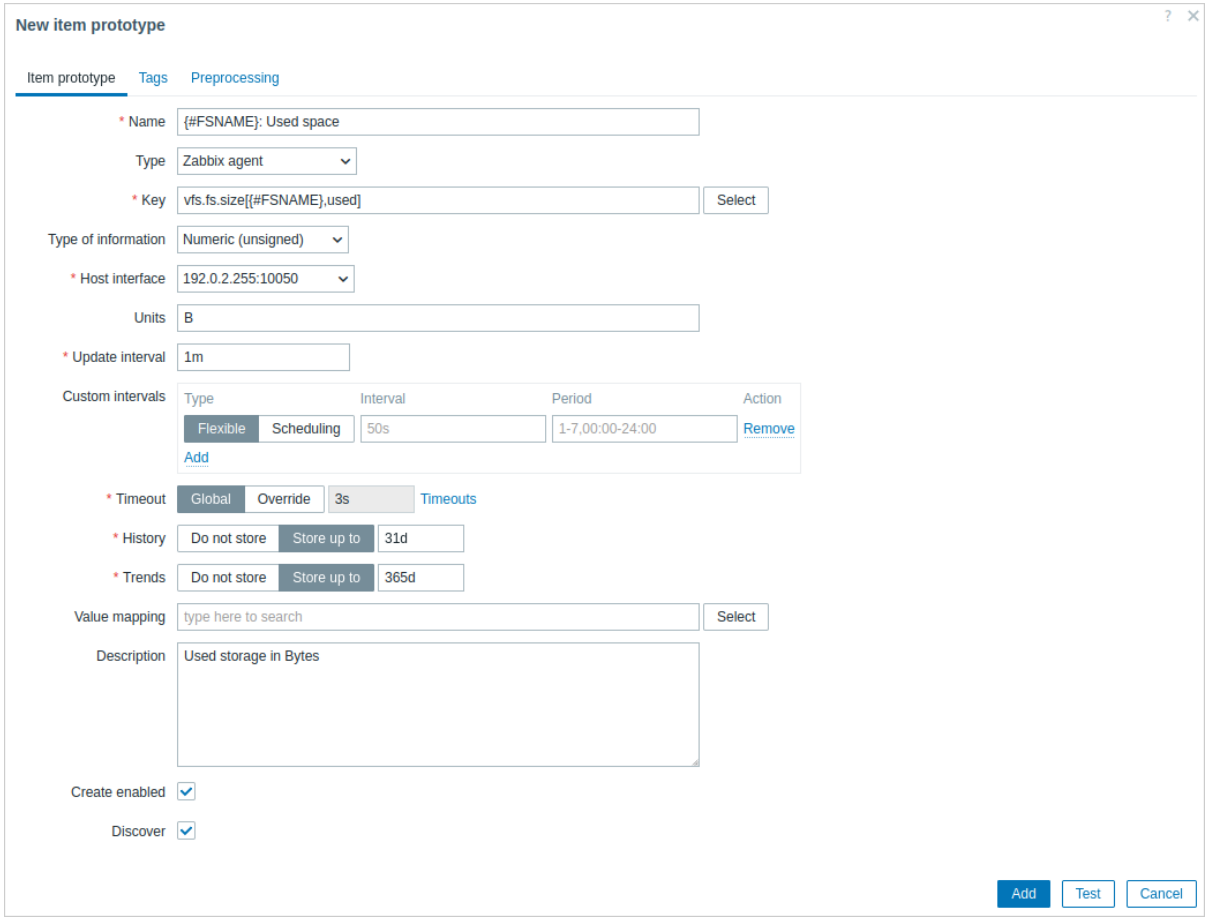
- descoberta de interfaces de rede
- descoberta de CPUs e núcleos de CPU
- descoberta de OIDs SNMP
- descoberta de objetos JMX;
- descoberta usando consultas SQL ODBC
- descoberta de serviços do Windows
- descoberta de interfaces de host no Zabbix

Para mais detalhes sobre o formato JSON para items de descoberta e um exemplo de como implementar seu próprio descobridor de sistema de arquivos como um script Perl, veja criando regras LLD personalizadas.

Protótipos de item

Depois que uma regra for criada, acesse os itens dessa regra e clique em *Criar protótipo de item* para criar um protótipo de item.

Observe como a macro `{#FSNAME}` é usada quando um nome de sistema de arquivos é necessário. O uso de uma macro de descoberta de baixo nível é obrigatório na chave do item para garantir que a descoberta seja processada corretamente. Quando a regra de descoberta for processada, essa macro será substituída pelo sistema de arquivos descoberto.



Note:
Quando os parâmetros da chave do item não são usados, coloque a macro de LLD dentro dos colchetes `[...]` da chave do item (por exemplo, `v_[{#MACRO}]`).

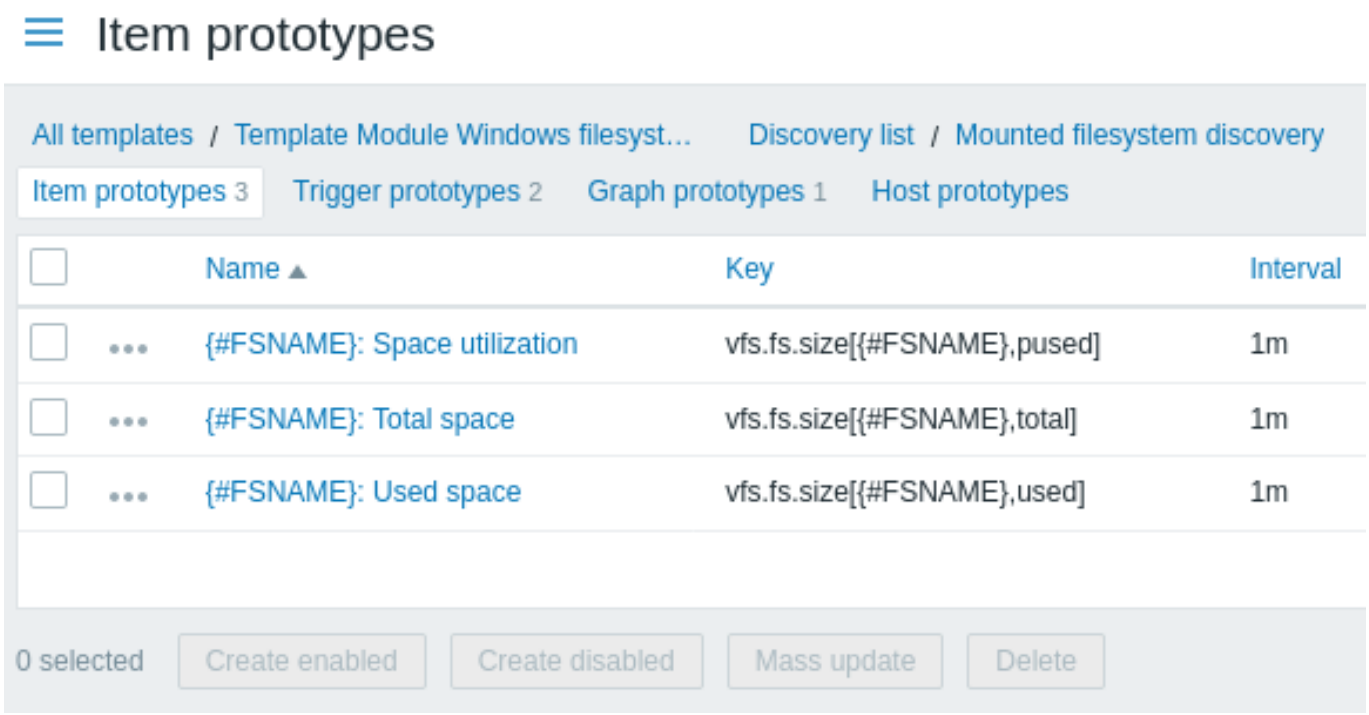
As macros de descoberta de baixo nível e as macros de usuário são compatíveis com a configuração de protótipos de item e com os parâmetros de pré-processamento de valores de item. Observe que, quando usadas em intervalos de atualização, uma única macro deve preencher todo o campo. Não há suporte para várias macros em um único campo nem para macros combinadas com texto.

Note:
O escape específico do 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 dos protótipos de item:

Parâmetro	Descrição
Criar habilitado	Se selecionado, o item será adicionado em um estado habilitado. Se não selecionado, o item será adicionado à entidade descoberta, mas em um estado desabilitado.
Descobrir	Se selecionado (padrão), o item será adicionado à entidade descoberta. Se não selecionado, o item não será adicionado à entidade descoberta, a menos que essa configuração seja substituída na regra de descoberta.

Podemos criar vários protótipos de item para cada métrica de sistema de arquivos de nosso interesse:



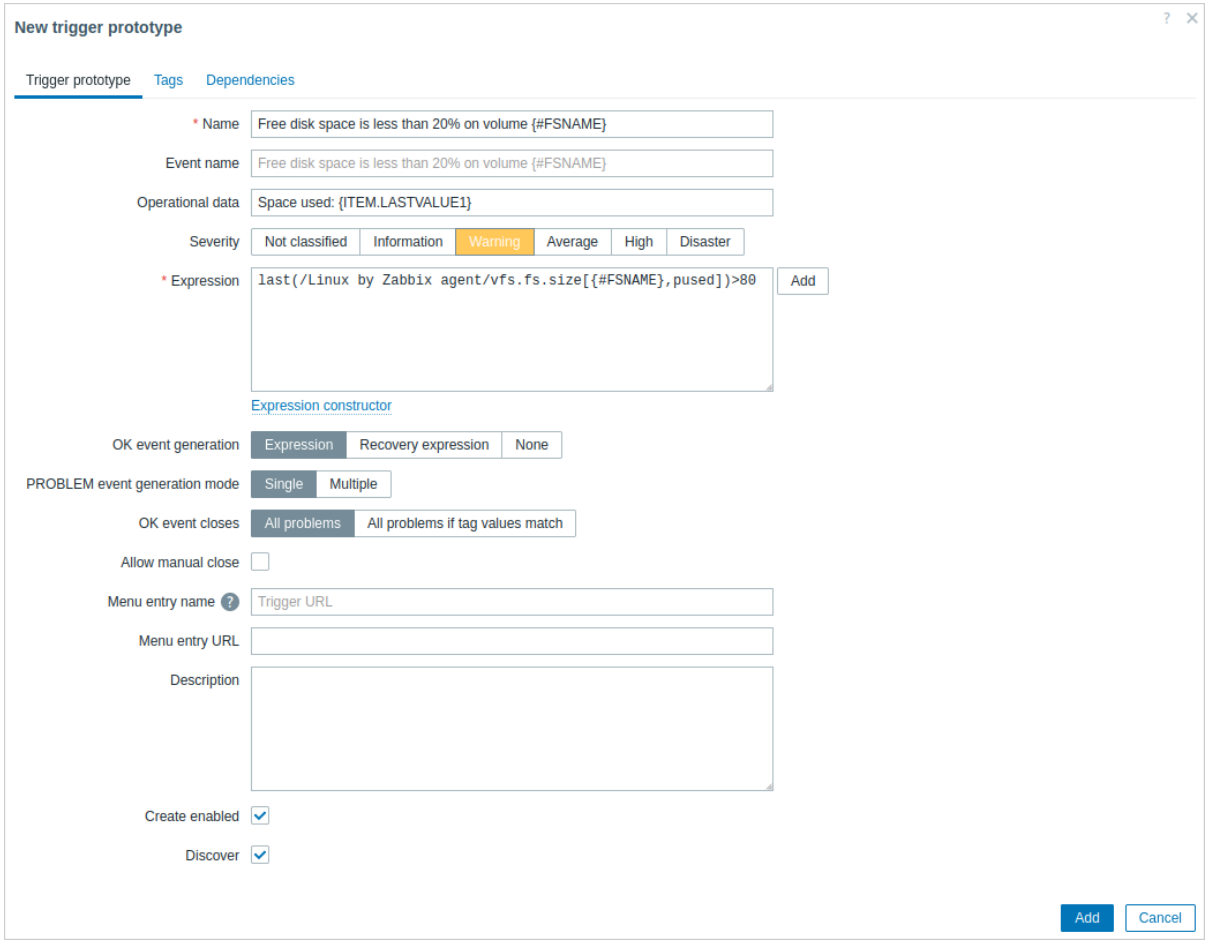
Clique no ícone de três pontos para abrir o menu do protótipo de item específico com estas opções:

- *Criar protótipo de trigger* - cria um protótipo de trigger com base neste protótipo de item
- *Protótipos de trigger* - clique para ver uma lista com links para os protótipos de trigger deste protótipo de item que já foram configurados
- *Criar item dependente* - cria um item dependente para este protótipo de item

A opção *Atualização em massa* está disponível caso você queira atualizar as propriedades de vários protótipos de item de uma só vez.

Protótipos de trigger

Criamos protótipos de trigger de forma semelhante aos protótipos de item:

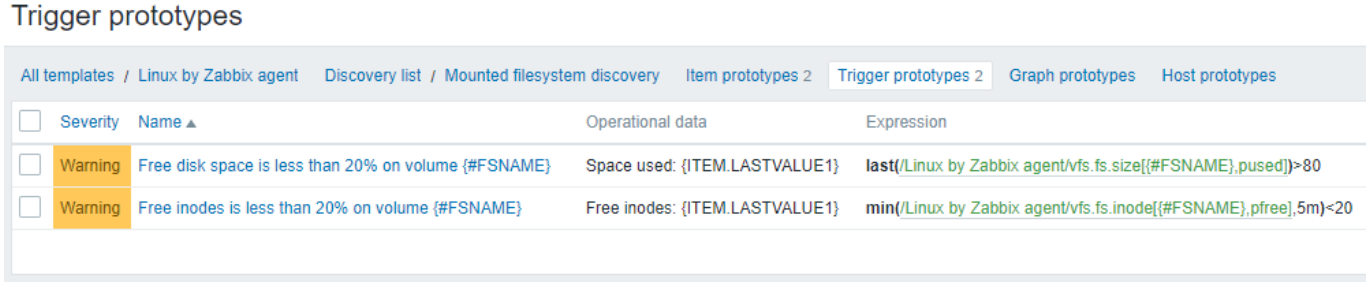


Atributos específicos para protótipos de trigger:

Parameter	Description
Create enabled	Se marcado, o trigger será adicionado em estado habilitado. Se desmarcado, o trigger será adicionado a uma entidade descoberta, mas em estado desabilitado.
Discover	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 essa configuração seja substituída na regra de descoberta.

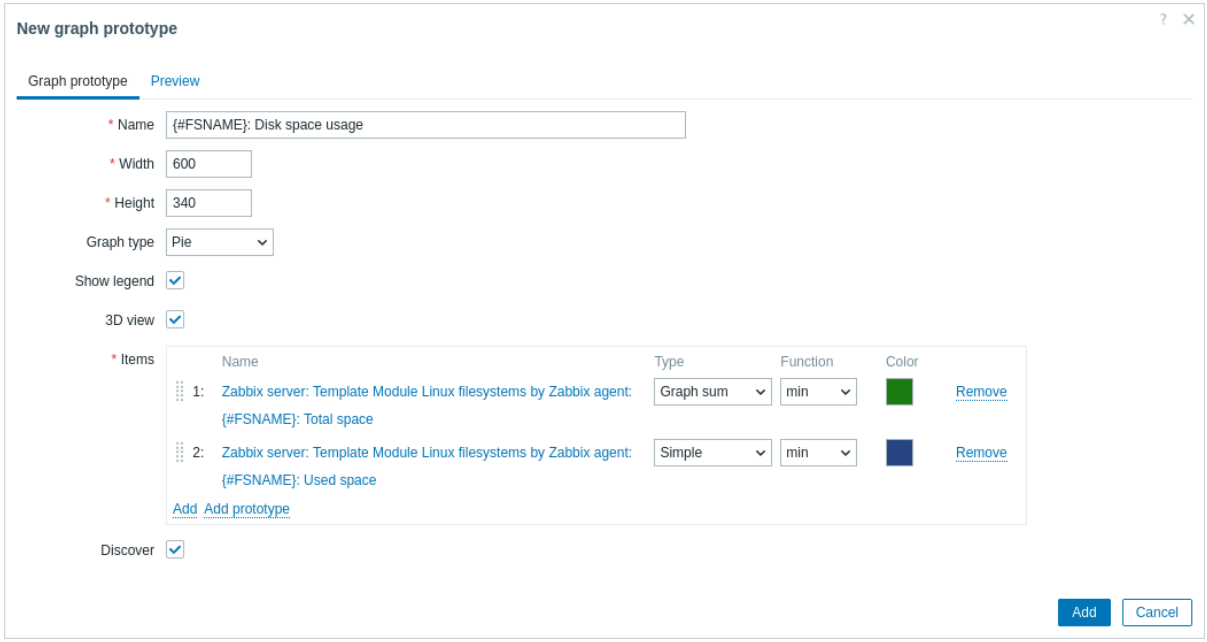
Quando triggers reais são criados a partir dos protótipos, pode ser necessário ter flexibilidade quanto à constante ('20' em nosso exemplo) usada para comparação na expressão. Veja como macros de usuário com contexto podem ser úteis para alcançar essa flexibilidade.

Você pode definir dependências entre protótipos de trigger. Para isso, vá até a aba *Dependencies*. Um protótipo de trigger pode depender de outro protótipo de trigger da mesma regra de descoberta em 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 nem de um trigger criado a partir de um protótipo de trigger. O protótipo de trigger do host não pode depender de um trigger de um template.



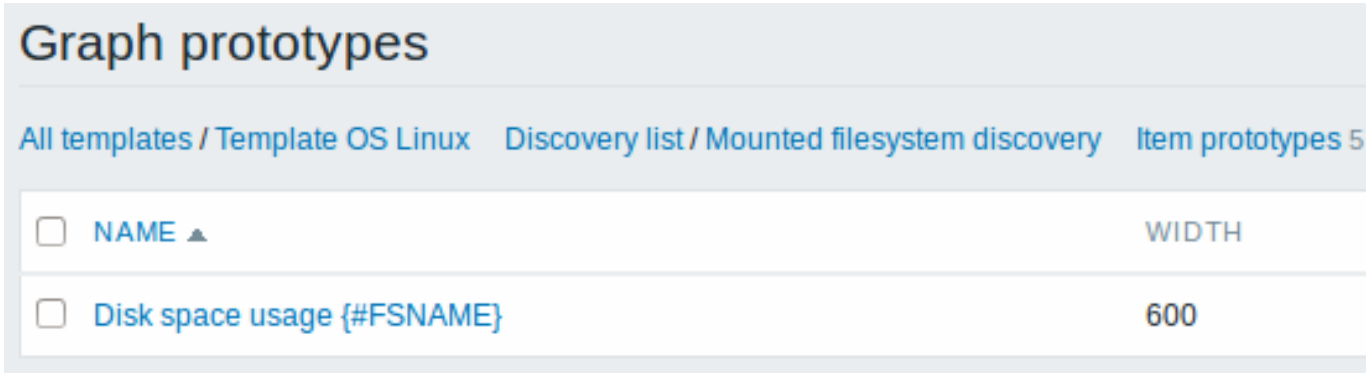
Protótipos de gráfico

Também podemos criar protótipos de gráfico:

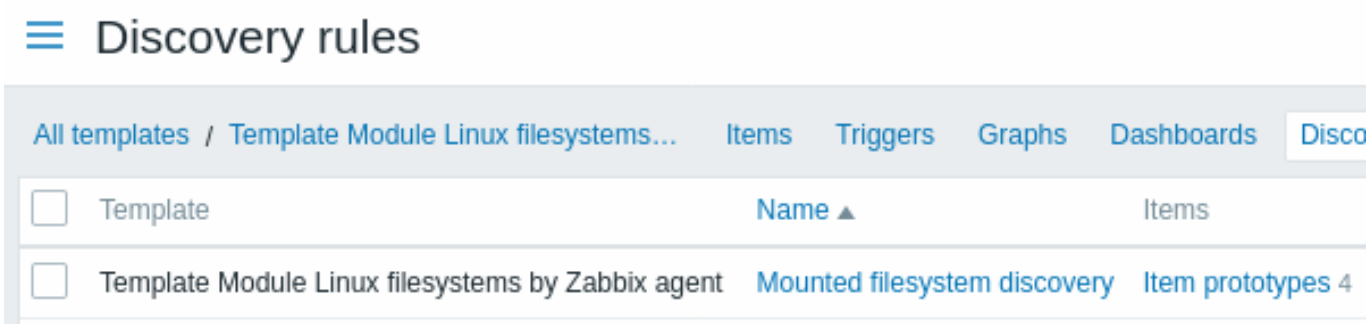


Atributos específicos para protótipos de gráfico:

Parameter	Description
Discover	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 aparece como mostrado abaixo. Ela tem cinco protótipos de item, dois protótipos de trigger e um protótipo de gráfico.



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

Os host prototypes são configurados em regras de descoberta em baixo nível.

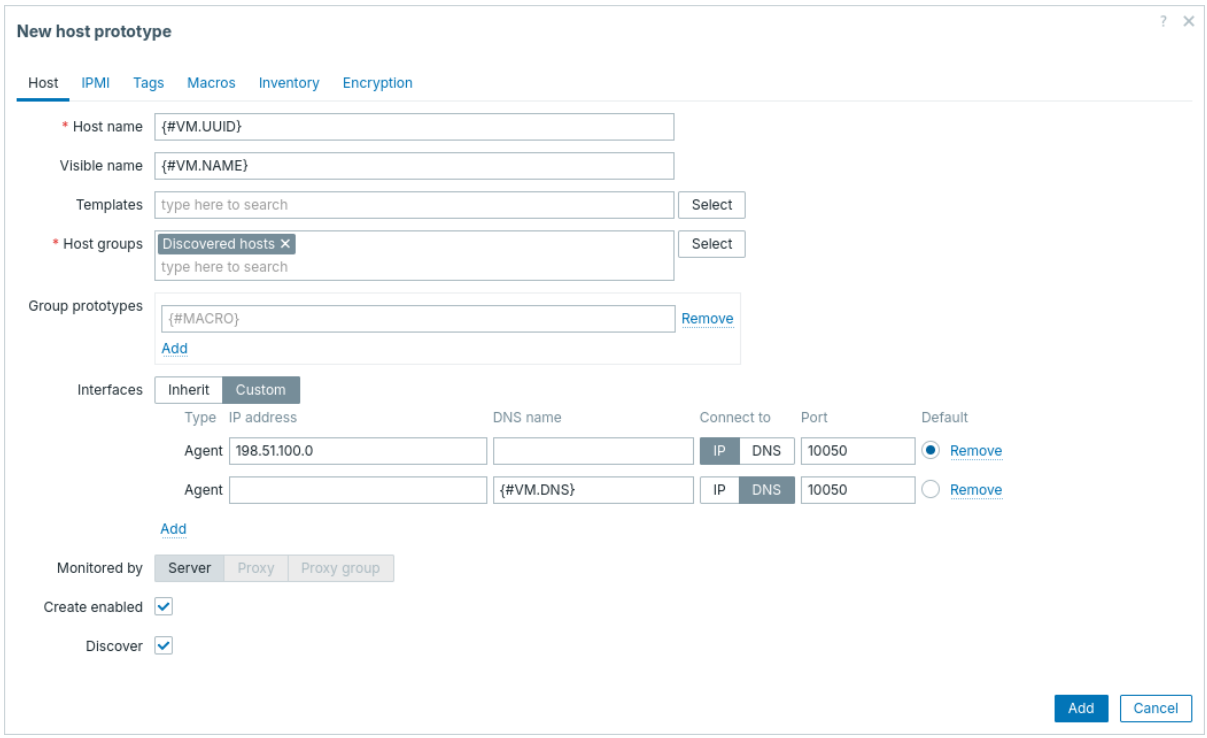
Para criar um host prototype:

1. Acesse *Coleta de dados > Hosts*.

2. Clique em *Discovery* para o host desejado para navegar até a lista de regras de descoberta em baixo nível configuradas para esse host.

3. Clique em *Host prototypes* para a regra de descoberta desejada.

4. Clique no botão *Create host prototype* no canto superior direito e preencha o formulário de configuração.



Todos os campos obrigatórios de entrada são marcados com um asterisco vermelho.

Parameter	Description
Host name	Este parâmetro deve conter pelo menos uma macro de descoberta em baixo nível para garantir nomes de host exclusivos para os hosts criados.
Visible name	Macros de descoberta em baixo nível são suportadas.
Group prototypes	Permite especificar protótipos de grupos de host usando macros de descoberta em baixo nível. Com base nos protótipos de grupo especificados, host groups serão descobertos, criados e vinculados aos hosts criados; grupos descobertos que já tenham sido criados por outras regras de descoberta em baixo nível também serão vinculados aos hosts criados. No entanto, host groups descobertos que correspondam a host groups criados manualmente não serão vinculados aos hosts criados.
Interfaces	Defina se os hosts descobertos herdarão o IP do host ao qual a regra de descoberta pertence (padrão) ou se دنواخ receber interfaces personalizadas. Macros de descoberta em baixo nível e macros de usuário são suportadas.
Monitored by	Este campo é sincronizado automaticamente com o respectivo campo em configuração do host.
Create enabled	Defina o status dos hosts descobertos; se desmarcado, os hosts serão criados como desabilitados.
Discover	Defina se os hosts serão criados a partir do host prototype; se desmarcado, os hosts não serão criados a partir do host prototype (a menos que essa configuração seja substituída na regra de descoberta em baixo nível).

Note:
Macros de descoberta em baixo nível também são suportadas para valores de tags e valores de macros de usuário do host prototype.
 Value maps não são suportados para host prototypes.

Para um exemplo de como configurar um host prototype, consulte *Monitoramento de máquina virtual*.

Interfaces do host

Para adicionar interfaces personalizadas, altere o seletor *Interfaces* de ”Herdar” para ”Personalizado”. Clique em Add e selecione o tipo de interface - 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 principal 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).

Protótipos de descoberta

Visão geral

Os protótipos de descoberta são regras de **descoberta de baixo nível aninhadas** dentro de uma regra de descoberta ”pai”, permitindo a descoberta em vários níveis de objetos com seus próprios itens, triggers etc. Por exemplo, você pode querer descobrir todas as instâncias de banco de dados em um servidor de banco de dados, depois descobrir os tablespaces de cada instância e, em seguida, descobrir as tabelas de cada tablespace.

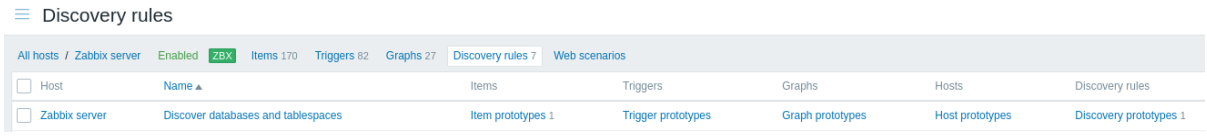
Os protótipos de descoberta têm seus próprios protótipos de item, trigger, gráfico, host e descoberta. Um protótipo de descoberta aninhado usará o mesmo valor JSON que a regra pai se você especificar um tipo *Nested*.

Os níveis de aninhamento dos protótipos de descoberta são ilimitados.

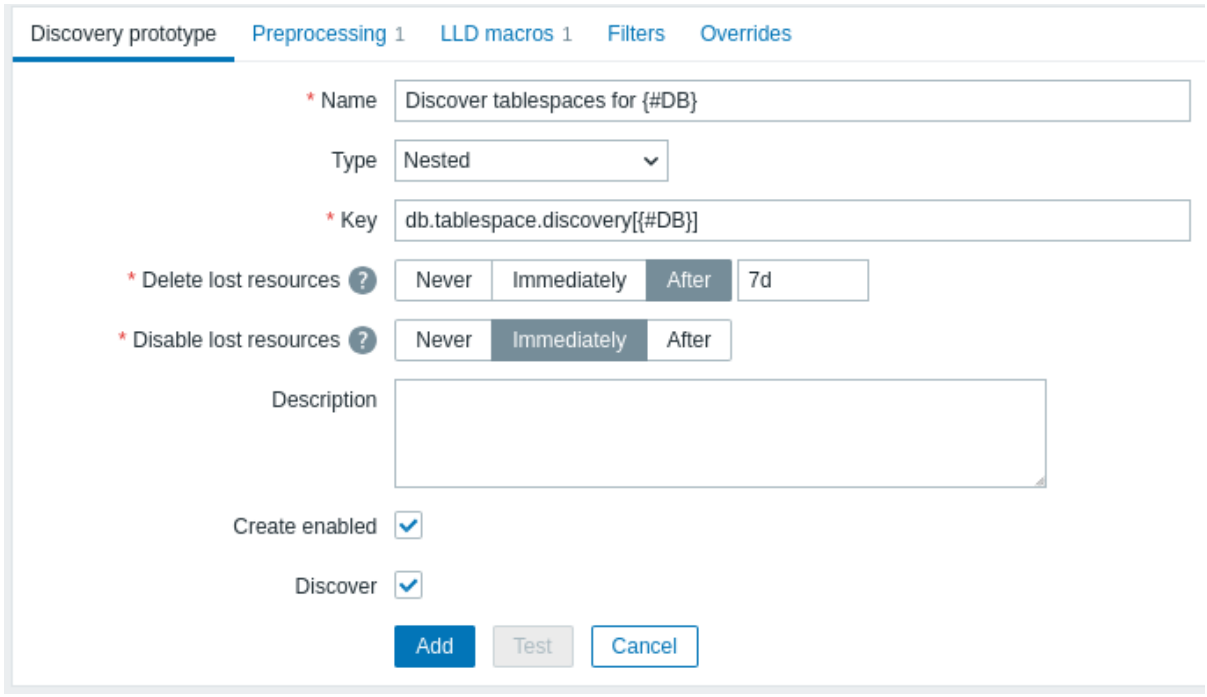
Configuração

Para criar um protótipo de descoberta:

- Clique em *Discovery prototypes* na linha de uma regra de descoberta existente



- Clique em *Create discovery prototype*



Os campos de configuração deste formulário são compartilhados com a descoberta em baixo nível regular.

Se você selecionar ”Nested” como *Type* no formulário de protótipo de descoberta aberto, então as regras de descoberta (a partir do protótipo de descoberta) são geradas com base em um objeto JSON a partir do mesmo valor JSON da regra de descoberta pai. Por exemplo, se o JSON original for [<object A>, <object B>] e houver um protótipo de regra de descoberta aninhada, então duas regras de descoberta serão geradas com base nos dados do objeto A e do objeto B, respectivamente.

Nesse 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 outra ”fatia” dos mesmos dados, já adquiridos pelo 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 uma 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

Descoberta de instâncias de banco de dados em um server de banco de dados e, em seguida, descoberta dos tablespaces de cada instância.

1. Você tem pelo menos um host relacionado à descoberta do server de banco de dados.
2. Crie uma regra de LLD para esse host com o nome *Descobrir bancos de dados e tablespaces*.
3. Vá para a aba *Macros de LLD* nessa regra e adicione a macro `{#DB}=$.database`.
4. Adicione um protótipo de item para essa regra com o nome *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 essa regra com o nome *Descobrir tablespaces para {#DB}* (Tipo: Nested, Chave: `db.tablespace.discovery[{#DB}]`).
7. Vá para a aba *Pré-processamento* desse protótipo de descoberta e adicione a etapa `JSONPath=$.tablespaces`.
8. Vá para a aba *Macros de LLD* desse protótipo de descoberta e adicione a macro `{#TSNAME}=$.name`.
9. Crie um protótipo de item para esse protótipo de descoberta com o nome *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

Descoberta de instâncias de banco de dados no server de banco de dados, representando-as como hosts descobertos, e depois descoberta dos tablespaces de cada instância.

1. Você tem pelo menos um host (host raiz) relacionado à descoberta do server 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 de LLD para este template chamada *Descobrir tablespaces* (Tipo: Nested).

5. Vá para a aba *Preprocessing* desta regra e adicione a etapa `JSONPath=$.tablespaces`.
6. Vá para a aba *LLD Macros* desta regra e adicione a macro `{#TSNAME}=$.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 de LLD para este host chamada *Descobrir bancos de dados e tablespaces*.
9. Vá para a aba *LLD Macros* desta regra e adicione a macro `{#DB}=$.database`.
10. Adicione um protótipo de host para esta regra chamado *Host para banco de dados {#DB}*.
11. Vá para a aba *Macros* deste protótipo de host e adicione a macro `{$DB}={#DB}` (para o nome e a chave do item da Etapa 3).
12. Vincule o template da etapa 2 a este protótipo de host.
13. Os hosts descobertos contêm os itens descobertos relacionados a cada banco de dados e seus tablespaces:

Host	Items
Host para 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 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 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

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, é necessário definir o parâmetro Alias do agent, 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.

Exemplos de regras de descoberta

Esta seção fornece exemplos práticos de configuração de regras de descoberta para diferentes tipos de entidades, como sistemas de arquivos, interfaces de rede, serviços, sensores, objetos SNMP, aplicações Java e bancos de dados.

Cada exemplo ilustra como uma regra de descoberta pode ser estruturada, incluindo a chave do item ou a consulta usada, a saída JSON com macros LLD e como esses resultados são usados para criar protótipos de item, trigger e gráfico.

- Descoberta de dispositivos de bloco
- Descoberta de CPUs e núcleos de CPU

- Descoberta de interfaces de host no Zabbix
- Descoberta de sensores IPMI
- Descoberta de objetos JMX
- Descoberta de sistemas de arquivos montados
- Descoberta de interfaces de rede
- Descoberta de OIDs SNMP
- Descoberta de OIDs SNMP (legado)
- Descoberta de serviços systemd
- Descoberta de instâncias de contador de desempenho do Windows
- Descoberta de serviços do Windows
- Descoberta usando consultas WMI
- Descoberta usando consultas SQL ODBC
- Descoberta usando dados do Prometheus

Regras de LLD personalizadas

Visão geral

Também é possível criar uma regra de LLD totalmente personalizada, descobrindo qualquer tipo de entidade — por exemplo, bancos de dados em um server de banco de dados.

Para isso, deve ser criado um item personalizado que retorne uma string JSON, especificando os objetos encontrados e, opcionalmente, algumas de suas propriedades. A quantidade de macros por entidade não é limitada — enquanto as regras de descoberta integradas retornam uma ou duas macros (por exemplo, duas para a descoberta de sistemas 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 (um que não oferece suporte à chave `vfs.fs.discovery`), mas ainda precise 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 forma 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 da 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 usados nos protótipos; a alternativa é extrair os valores das macros LLD usando JSONPath {#FSNAME} → $.fsname e {#FSTYPE} → $.fstype, tornando possível um script como este:

```
####!/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 da 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 *Filter* 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 pode usar quaisquer nomes que desejar. Caso JSONPath seja usado, então a linha LLD será um elemento de array que pode ser um objeto, mas também pode ser outro array ou um valor.

Observe que, ao usar um parâmetro de usuário, o valor de retorno é limitado a 16 MB. Para mais detalhes, consulte limites de dados para valores de retorno de LLD.

Monitoramento de máquina virtual

Visão geral O Zabbix pode usar regras de descoberta de baixo nível para descobrir automaticamente hipervisores VMware e máquinas virtuais, e criar hosts para monitorá-los com base em prototipos de host predefinidos.

O Zabbix também inclui templates prontos para uso para monitorar 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 de sources, ele deverá 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 algum tempo em instalações grandes).

 Em geral, o valor de `StartVMwareCollectors` não deve ser menor que 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 oferecer suporte às métricas de capacidade do datastore, certifique-se de que o valor da chave VMware `vpxd.stats.maxQueryMetrics` esteja definido como pelo menos 64. Para mais informações, consulte o artigo da *VMware Knowledge Base*.

Descoberta

O Zabbix pode usar regras de descoberta de baixo nível (por exemplo, `vmware.hv.discovery[{$VMWARE.URL}]`) para descobrir automaticamente hipervisores e máquinas virtuais VMware. Além disso, o Zabbix pode usar protótipos de host para gerar automaticamente hosts reais para as entidades descobertas. Para mais informações, consulte *Host prototypes*.

Exemplos de configuração

Para um exemplo básico de como configurar o Zabbix para monitorar VMware usando o template *VMware FQDN*, consulte *Monitor VMware with Zabbix*.

Para um exemplo mais detalhado de como criar um host, uma regra de descoberta em baixo nível e um protótipo de host para monitorar VMware, consulte *Setup example*.

Registro estendido Os dados coletados pelos processos do *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"`, em que `N` é o número do processo.

Por exemplo, para aumentar o nível de depuração de 4 para 5 para todos os processos do *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 do *vmware collector*, execute o seguinte comando:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Quando o registro estendido dos dados do VMware não for mais necessário, é recomendável reduzir 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 foram tornadas indisponíveis ou desativadas por padrão em versões recentes do VMware vSphere, ou se não há alguns limites definidos 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 erro de número máximo de caracteres permitido, adicione um parâmetro `config.vpxd.stats.maxQueryMetrics` às configurações do vCenter Server. O valor desse parâmetro deve ser o mesmo que o valor de `maxQuerysize` no arquivo `web.xml` da VMware. Para mais informações, consulte o artigo *VMware Knowledge Base*.
- Se você suspeitar que sua instalação do Zabbix está usando memória em excesso, consulte Profiling excessive memory usage with tcmalloc.

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.

Itens suportados As chaves de item são listadas sem parâmetros e informações adicionais. Clique na chave de item para ver os detalhes completos.

Chave de item	Descrição	Grupo de item
vmware.eventlog	O log de eventos do VMware.	Serviço geral
vmware.fullname	O nome completo do serviço VMware.	
vmware.version	A versão do serviço VMware.	
vmware.cl.perfcounter	As métricas do contador de desempenho do cluster VMware.	Cluster
vmware.cluster.alarms.get	Os dados de alarmes do cluster VMware.	
vmware.cluster.discovery	A descoberta de clusters VMware.	
vmware.cluster.property	A propriedade do cluster VMware.	
vmware.cluster.status	O status do cluster VMware.	
vmware.cluster.tags.get	A matriz de tags do cluster VMware.	
vmware.datastore.alarms.get	Os dados de alarmes do datastore VMware.	Datastore

Chave de item	Descrição	Grupo de item
vmware.datastore.discovery	A descoberta de datastores VMware.	
vmware.datastore.hv.list	A lista de hypervisors 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	O tempo gasto em 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	A matriz de tags do datastore VMware.	
vmware.datastore.write	O tempo gasto em uma operação de gravação no datastore.	
vmware.dc.alarms.get	Os dados de alarmes do datacenter VMware.	Datacenter
vmware.dc.discovery	A descoberta de datacenters VMware.	
vmware.dc.tags.get	A matriz de tags do datacenter VMware.	
vmware.dvswitch.discovery	A descoberta de VMware vSphere Distributed Switches.	vSphere Distributed Switch
vmware.dvswitch.fetchportsget	Os dados das portas do VMware vSphere Distributed Switch.	
vmware.hv.alarms.get	Os dados de alarmes do hypervisor VMware.	Hypervisor
vmware.hv.cluster.name	O nome do cluster do hypervisor VMware.	
vmware.hv.connectionstate	O estado de conexão do hypervisor VMware.	
vmware.hv.cpu.usage	O uso do processador do hypervisor VMware (Hz).	
vmware.hv.cpu.usage.perf	O uso do processador do hypervisor VMware como porcentagem durante o intervalo.	
vmware.hv.cpu.utilization	O uso do processador do hypervisor VMware como porcentagem durante o intervalo, dependendo do gerenciamento de energia ou HT.	
vmware.hv.datacenter.name	O nome do datacenter do hypervisor VMware.	
vmware.hv.datastore.discovery	A descoberta de datastores do hypervisor VMware.	
vmware.hv.datastore.list	A lista de datastores do hypervisor VMware.	
vmware.hv.datastore.multipath	O número de caminhos de datastore disponíveis.	
vmware.hv.datastore.read	A quantidade média de tempo para uma operação de leitura do datastore.	
vmware.hv.datastore.size	O espaço do datastore VMware em bytes ou em porcentagem do total.	
vmware.hv.datastore.write	A quantidade média de tempo para uma operação de gravação no datastore.	
vmware.hv.discovery	A descoberta de hypervisors VMware.	
vmware.hv.diskinfo.get	Os dados de disco do hypervisor VMware.	
vmware.hv.fullname	O nome completo do produto, incluindo informações de versão.	
vmware.hv.hw.cpu.freq	A frequência do processador do hypervisor VMware.	
vmware.hv.hw.cpu.model	O modelo do processador do hypervisor VMware.	
vmware.hv.hw.cpu.num	O número de núcleos do processador no hypervisor VMware.	
vmware.hv.hw.cpu.threads	O número de threads do processador no hypervisor VMware.	
vmware.hv.hw.memory	O tamanho total de memória do hypervisor VMware.	
vmware.hv.hw.model	O modelo do hypervisor VMware.	
vmware.hv.hw.sensors.get	O valor dos sensores de hardware do hypervisor VMware.	
vmware.hv.hw.serialnumber	O número de série do hypervisor VMware.	
vmware.hv.hw.uuid	O UUID da BIOS do hypervisor VMware.	
vmware.hv.hw.vendor	O nome do fornecedor do hypervisor VMware.	
vmware.hv.maintenance	O status de manutenção do hypervisor VMware.	
vmware.hv.memory.size.ballooned	O tamanho da memória ballooned do hypervisor VMware.	
vmware.hv.memory.used	O tamanho da memória usada do hypervisor VMware.	
vmware.hv.net.if.discovery	A descoberta de interfaces de rede do hypervisor VMware.	
vmware.hv.network.in	As estatísticas de entrada de rede do hypervisor VMware.	
vmware.hv.network.linkspeed	A velocidade da interface de rede do hypervisor VMware.	
vmware.hv.network.out	As estatísticas de saída de rede do hypervisor VMware.	
vmware.hv.perfcounter	O valor do contador de desempenho do hypervisor VMware.	
vmware.hv.property	A propriedade do hypervisor VMware.	
vmware.hv.power	O uso de energia do hypervisor VMware.	
vmware.hv.sensor.health.state	O sensor agregado do estado de integridade do hypervisor VMware.	
vmware.hv.sensors.get	Os sensores de estado do fornecedor de hardware do hypervisor VMware.	
vmware.hv.status	O status do hypervisor VMware.	
vmware.hv.tags.get	A matriz de tags do hypervisor VMware.	
vmware.hv.uptime	O tempo de atividade do hypervisor VMware.	
vmware.hv.version	A versão do hypervisor VMware.	
vmware.hv.vm.num	O número de máquinas virtuais no hypervisor VMware.	

Chave de 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 VMware resource pool.	
vmware.alarms.get	Os dados de alarmes do VMware virtual center.	Virtual center
vmware.vm.alarms.get	Os dados de alarmes da máquina virtual VMware.	Virtual machine
vmware.vm.attribute	O valor do atributo personalizado da máquina virtual VMware.	
vmware.vm.cluster.name	O nome da máquina virtual VMware.	
vmware.vm.consolidationneeded	O disco da máquina virtual VMware requer consolidação.	
vmware.vm.cpu.latency	A porcentagem de tempo em que a máquina virtual não consegue executar porque está competindo pelo acesso à(s) CPU(s) física(s).	
vmware.vm.cpu.num	O número de processadores na máquina virtual VMware.	
vmware.vm.cpu.readiness	A porcentagem de tempo em que a máquina virtual estava pronta, mas não conseguiu ser agendada para execução na CPU física.	
vmware.vm.cpu.ready	O tempo em que a máquina virtual estava pronta, mas não conseguiu ser agendada para execução na CPU física.	
vmware.vm.cpu.swapwait	A porcentagem de tempo de CPU gasto aguardando 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 movida 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 hypervisor da máquina virtual VMware.	
vmware.vm.hv.name	O nome do hypervisor da máquina virtual VMware.	
vmware.vm.memory.size	O tamanho total de 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 compactada da máquina virtual VMware.	
vmware.vm.memory.size.consumed	A quantidade de memória física do host consumida para fazer 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 swap 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	A matriz de tags da máquina virtual VMware.	
vmware.vm.tools	O estado ou a versão das ferramentas do guest da máquina virtual VMware.	
vmware.vm.uptime	O tempo de atividade da máquina virtual VMware.	
vmware.vm.vfs.dev.discovery	A descoberta de dispositivos de disco da máquina virtual VMware.	
vmware.vm.vfs.dev.read	As estatísticas de leitura do dispositivo de disco da máquina virtual VMware.	

Chave de 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 estar entre aspas se mais de uma severidade for especificada em uma lista delimitada 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 o exemplo de filtragem de registros do 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: JSON object.

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: JSON object.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do cluster VMware.

Este item funciona com vSphere 6.5 e versões mais recentes.

vmware.datastore.alarms.get[url,uuid]

 Os dados de alarmes do datastore VMware.
 Valor de retorno: Objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore VMware.

vmware.datastore.discovery[url,<filter_uuid>]

 A descoberta de datastores VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **filter_uuid** - um único identificador exclusivo de HV ou VM (padrão: vazio). Se especificado, apenas os datastores vinculados serão descobertos.

vmware.datastore.hv.list[url,datastore]

 A lista de 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: JSON object.

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 versões mais recentes.

vmware.datastore.write[url,datastore,<mode>]

 A quantidade de tempo para uma operação de gravação no datastore (milissegundos).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (valor médio, padrão) ou *maxlatency* (valor máximo).

vmware.dc.alarms.get[url,id]

 Os dados de alarmes do datacenter VMware.
 Valor de retorno: JSON object.

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]

 A matriz de tags do datacenter VMware.
 Valor de retorno: JSON object.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do datacenter VMware.

Este item funciona com vSphere 6.5 e versões mais recentes.

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 vírgulas 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 hypervisor VMware.
 Valor de retorno: JSON object.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor 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 de disco 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.fullname[url,uuid]

 O nome completo do produto, incluindo informações de versão.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.cpu.freq[url,uuid]

 A frequência do processador do 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 hypervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.status[url,uuid]

 O status do hypervisor VMware.
 Valor de retorno: *Integer*: *0* - cinza; *1* - verde; *2* - amarelo; *3* - vermelho.

Parâmetros:

- **url** - 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: JSON object.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item funciona com o vSphere 6.5 e versões mais recentes.

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 centro virtual VMware.
 Valor de retorno: JSON object.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.vm.alarms.get[url,uuid]

 Os dados de alarmes da máquina virtual VMware.
 Valor de retorno: JSON object.

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.

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: JSON object.

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: JSON object.

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 versões mais recentes.

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 itens é obtido dos contadores de desempenho do VMware e o parâmetro VMwarePerfFrequency é usado para atualizar seus dados no cache do 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.

Campos de chave de descoberta de máquina virtual

A tabela a seguir lista os campos retornados pelas chaves de descoberta relacionadas a máquinas virtuais.

Item key		
Descrição	**Campo**	**Conteúdo recuperado**
vmware.cluster.discovery		
Realiza a descoberta de cluster.	{#CLUSTER.ID}	Identificador do cluster.
	{#CLUSTER.NAME}	Nome do cluster.

Item key		

	”resource_pool”	Uma matriz contendo dados do resource pool, incluindo ID do grupo de recursos, matriz de tags, caminho do resource pool e número de máquinas virtuais.
		
		Estrutura da matriz:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Para a estrutura da matriz `"tags"`, consulte o campo ”tags”.

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Realiza a descoberta de datastore.	{#DATASTORE}	Nome do datastore.

	{#DATASTORE.EXTENT}	Uma matriz contendo o ID da partição de extent do datastore e o nome da instância.
		
		Estrutura da matriz:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Tipo de datastore. Exemplos de valores: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Identificador do datastore.

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dc.discovery		
Realiza a descoberta de datacenter.	{#DATACENTER}	Nome do datacenter.
	{#DATACENTERID}	Identificador do datacenter.

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dvswitch.discovery		
Realiza a descoberta de switches distribuídos do vSphere.	{#DVS.NAME}	Nome do switch.

Item key		
	{#DVS.UUID}	Identificador do switch.
vmware.hv.discovery		
Realiza a descoberta do hypervisor.	{#HV.UUID}	Identificador exclusivo do hypervisor.
	{#HV.ID}	Identificador do hypervisor (nome do objeto gerenciado HostSystem).
	{#HV.NAME}	Nome do hypervisor.
	{#HV.NETNAME}	Nome de host de rede do hypervisor.
	{#HV.IP}	Endereço IP do hypervisor, pode estar vazio. No caso de uma configuração de HA com várias interfaces de rede, é observada a seguinte prioridade de seleção da interface: - prefira o IP que compartilha a sub-rede IP com o IP do vCenter; - prefira o IP da sub-rede IP com gateway padrão; - prefira 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 hypervisor.
	{#PARENT.TYPE}	Tipo de contêiner no qual o hypervisor está armazenado. Os valores podem ser `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, onde ”VMware” representa um tipo de contêiner desconhecido.

	”resource_pool”	Uma matriz contendo dados do resource pool, incluindo ID do grupo de recursos, matriz de tags, caminho do resource pool e número de máquinas virtuais.
		
		Estrutura da matriz:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Para a estrutura da matriz `"tags"`, consulte o campo ”tags”.

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.datastore.discovery		
Realiza a descoberta de datastore do hypervisor. Observe que vários hypervisors 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.

Item key		

	”datastore_extent”	Uma matriz contendo o nome da instância do extent do datastore e o ID da partição.
		
		Estrutura da matriz:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Realiza a descoberta de interfaces de rede do hypervisor.	{#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		
Realiza a descoberta de máquina virtual.	{#VM.UUID}	Identificador exclusivo da máquina virtual.
	{#VM.ID}	Identificador da máquina virtual (nome do objeto gerenciado VirtualMachine).
	{#VM.NAME}	Nome da máquina virtual.
	{#HV.NAME}	Nome do hypervisor.
	{#HV.UUID}	Identificador exclusivo do hypervisor.
	{#HV.ID}	Identificador do hypervisor (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 guest da máquina virtual, pode estar vazio.
	{#VM.GUESTFULLNAME}	Nome completo do sistema operacional guest da máquina virtual, 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 resource pool.
	{#VM.RPOOL.PATH}	Caminho completo do resource pool, excluindo o nome ”root” ”Resources”. Os nomes das pastas são combinados com ”/”.
	{#VM.SNAPSHOT.COUNT}	Número de snapshots da VM.

	”tags”	Uma matriz contendo tags com nome da tag, descrição e categoria.
		
		Estrutura da matriz:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

Item key		

	”vm_customattribute”	Uma matriz de atributos personalizados da máquina virtual (se definidos).
		
		Estrutura da matriz:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	”net_if”	Uma matriz de interfaces de rede da máquina virtual.
		
		Estrutura da matriz:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		Para a descrição dos dados retornados, consulte a chave de item ”vmware.vm.net.if.discovery”.

vmware.vm.net.if.discovery		
Realiza 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}	Estado de conexão da interface (`false` - desconectada; `true` - conectada).
	{#IFTYPE}	Tipo da interface.
	{#IFBACKINGDEVICE}	Nome do dispositivo de suporte.
	{#IFDVSWITCH.UUID}	Identificador exclusivo do vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Port group distribuído.
	{#IFDVSWITCH.PORT}	Porta do vSphere Distributed Switch.
	”ifip”	Uma matriz de endereços da interface.
vmware.vm.vfs.dev.discovery		
Realiza a descoberta de dispositivos de disco da máquina virtual.	{#DISKNAME}	Nome do dispositivo de disco.
vmware.vm.vfs.fs.discovery		
Realiza a descoberta do sistema de arquivos da máquina virtual.	{#FSNAME}	Nome do sistema de arquivos.

Exemplos de JSON para itens VMware

Visão geral Esta seção fornece informações adicionais sobre objetos JSON retornados por vários items da VMware.

vmware.*.alarms.get Os itens **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,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
    ]
}
```


vmware.*.tags.get 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
}
```


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 aborda a configuração do VMware. Presume-se que o VMware já esteja configurado.

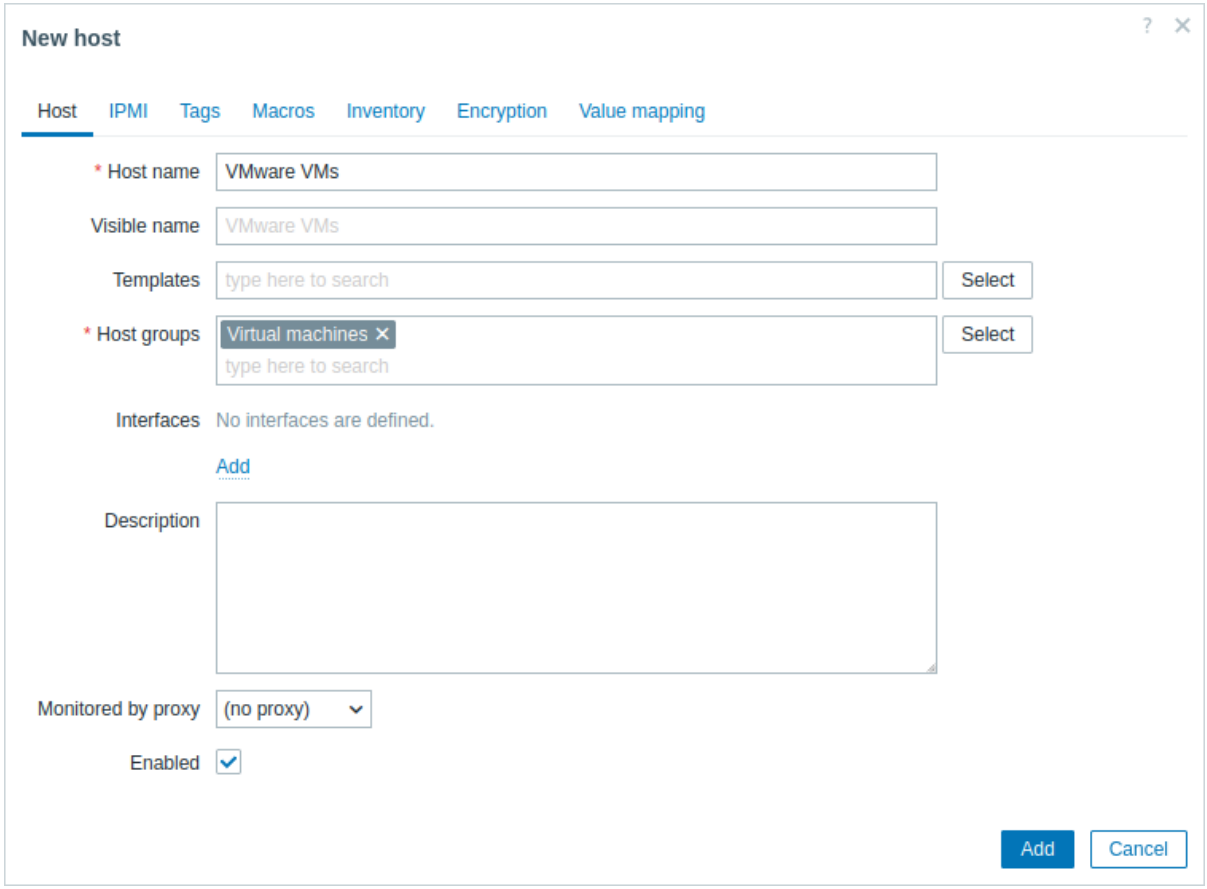
Antes de prosseguir, defina o parâmetro `StartVMwareCollectors` no arquivo de configuração do Zabbix server para 2 ou mais (o valor padrão é 0).

Criar um host

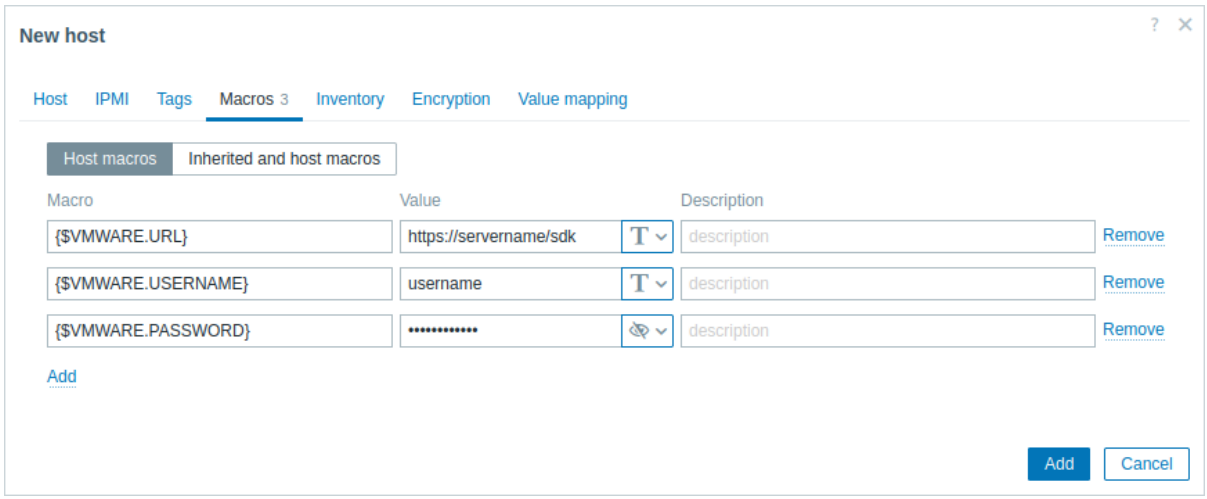
1. Vá para *Data collection* > *Hosts*.

2. Crie um host:

- No campo *Host name*, insira um nome de host (por exemplo, "VMware VMs").
- No campo *Host groups*, digite ou selecione um grupo de hosts (por exemplo, "Virtual machines").



- Na aba *Macros*, defina as seguintes macros do host:
  - {$VMWARE.URL} - URL do SDK do serviço VMware (hipervisor ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nome de usuário do serviço VMware
  - {$VMWARE.PASSWORD} - senha do usuário {$VMWARE.USERNAME} do serviço VMware



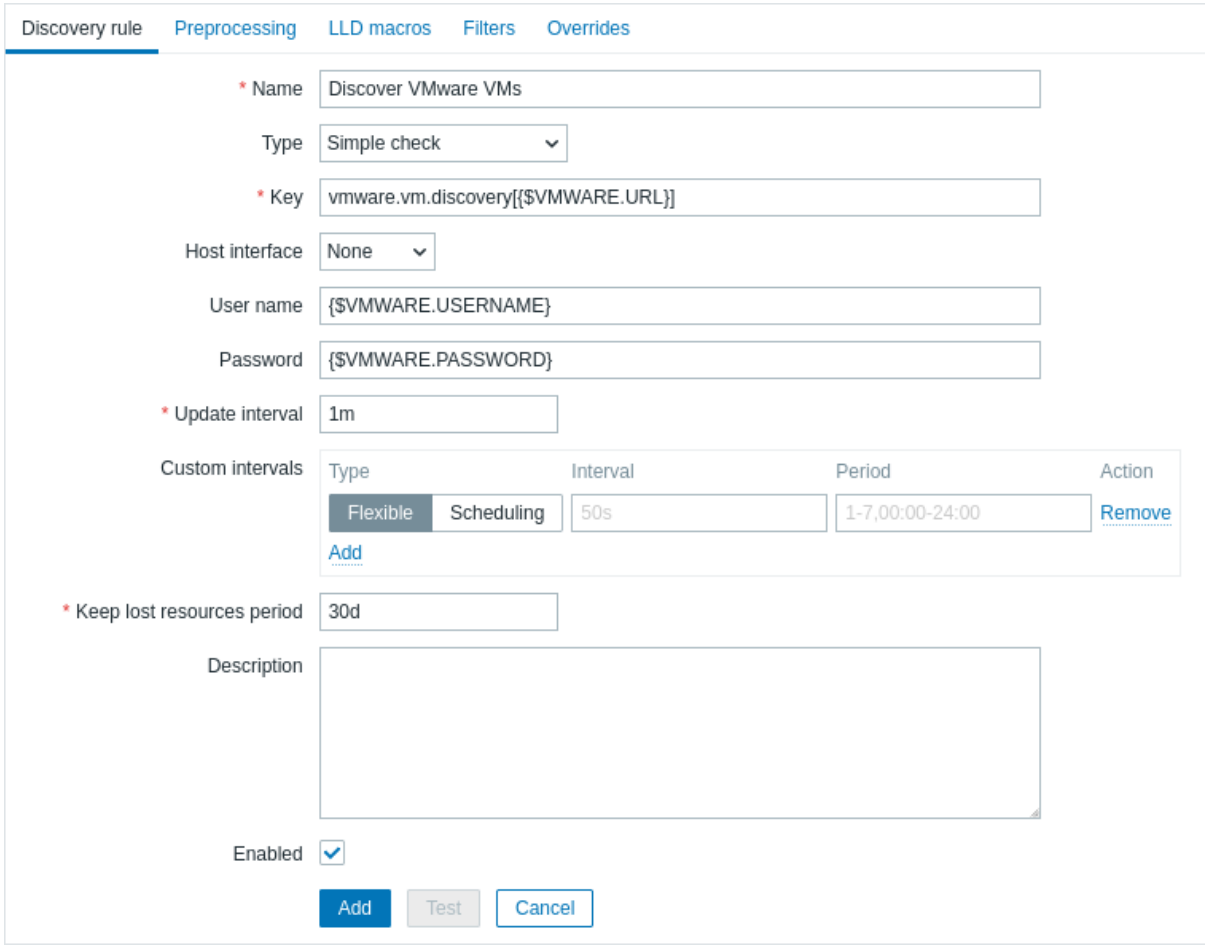
3. Clique no botão *Add* para criar o host. Este host representará seu ambiente VMware.

Criar uma regra de descoberta de baixo nível

1. Clique em *Discovery* no host criado para ir para a lista de regras de descoberta de baixo nível desse host.

2. Crie uma regra de descoberta de baixo nível:

- No campo *Name*, insira um nome para a regra de descoberta de baixo nível (por exemplo, "Discover VMware VMs").
- No campo *Type*, selecione "Simple check".
- No campo *Key*, insira a chave de item integrada para descobrir máquinas virtuais VMware: `vmware.vm.discovery[{$VMWARE.URL}]`
- Nos campos *User name* e *Password*, insira as macros correspondentes configuradas anteriormente no host.



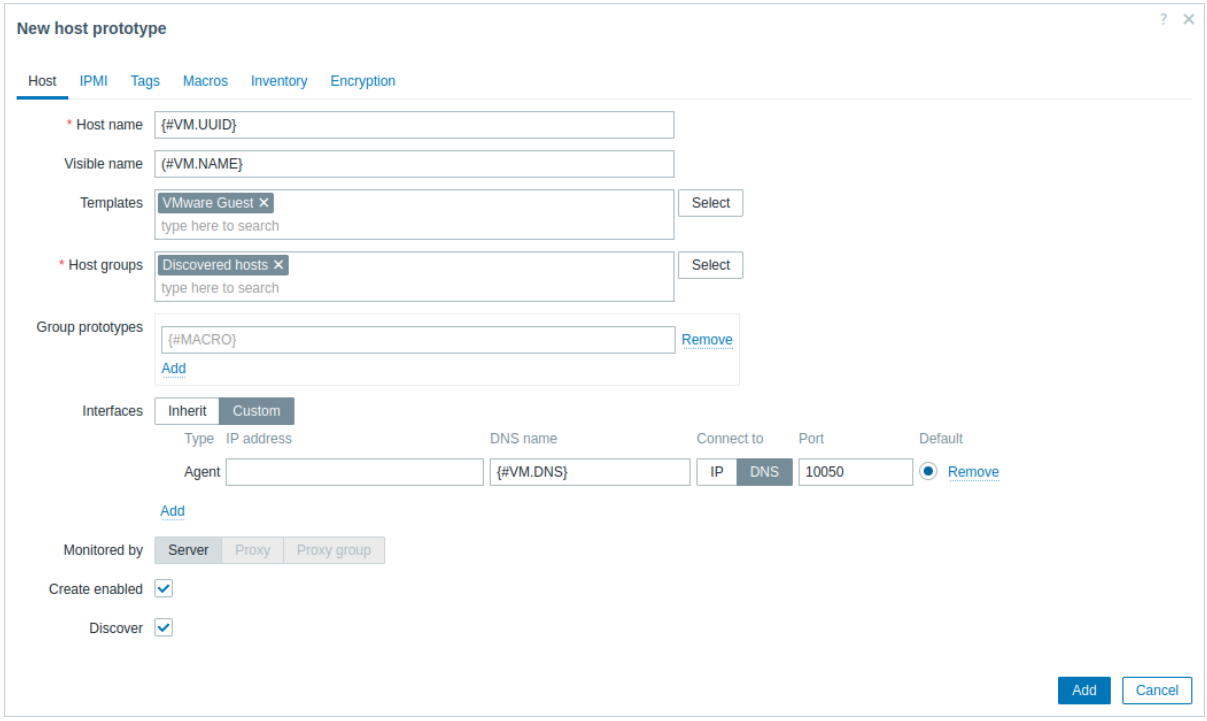
3. Clique no botão *Add* para criar a regra de descoberta de baixo nível. Essa regra de descoberta descobrirá máquinas virtuais no seu ambiente VMware.

Criar um protótipo de host

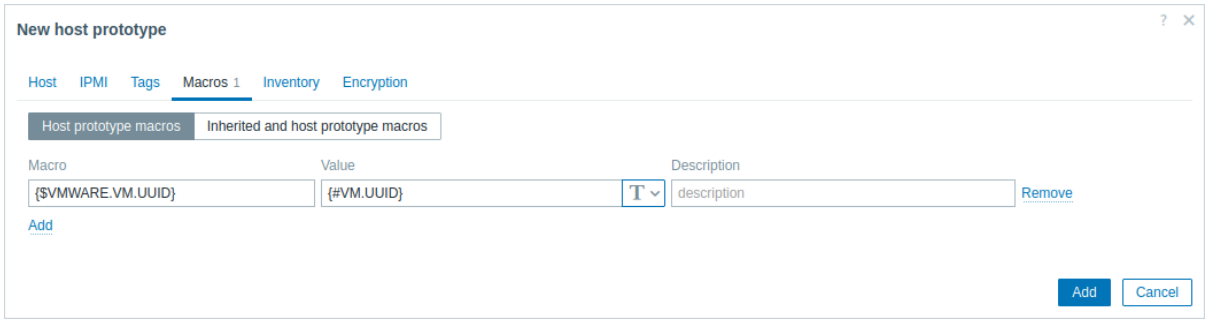
1. Na lista de regras de descoberta de baixo nível, clique em *Protótipos de host* para a regra de descoberta de baixo nível criada anteriormente.

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}]*).

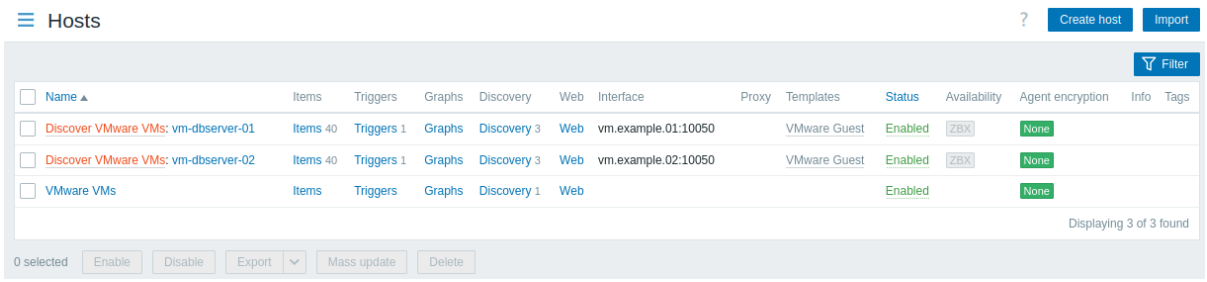


3. Clique no botão *Adicionar* para criar o protótipo de host. Esse protótipo de host será usado para criar hosts para máquinas virtuais descobertas pela regra de descoberta de baixo nível criada anteriormente.

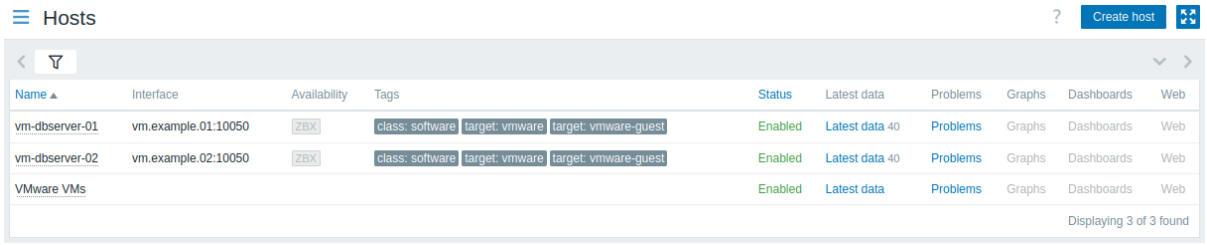
Exibir hosts e métricas

Depois que o host prototype tiver sido criado, 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 *Latest data* 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*.

Monitoramento web

Visão geral Com o Zabbix, você pode verificar vários aspectos de disponibilidade de sites.

> **Attention:**
> Para realizar o monitoramento web, o server do Zabbix 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 ”steps”. Os steps são executados periodicamente pelo server do Zabbix em uma ordem predefinida. Se um host for monitorado

por proxy, os steps são executados pelo proxy.

Os cenários web são associados a hosts/templates da mesma forma que items, triggers etc. Isso significa que os cenários web também podem ser criados no 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 todos os steps de todo o cenário
- número do step que falhou
- última mensagem de erro

As seguintes informações são coletadas em qualquer step de cenário web:

- velocidade de download por segundo
- tempo de resposta
- código de resposta

Para mais detalhes, consulte 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.

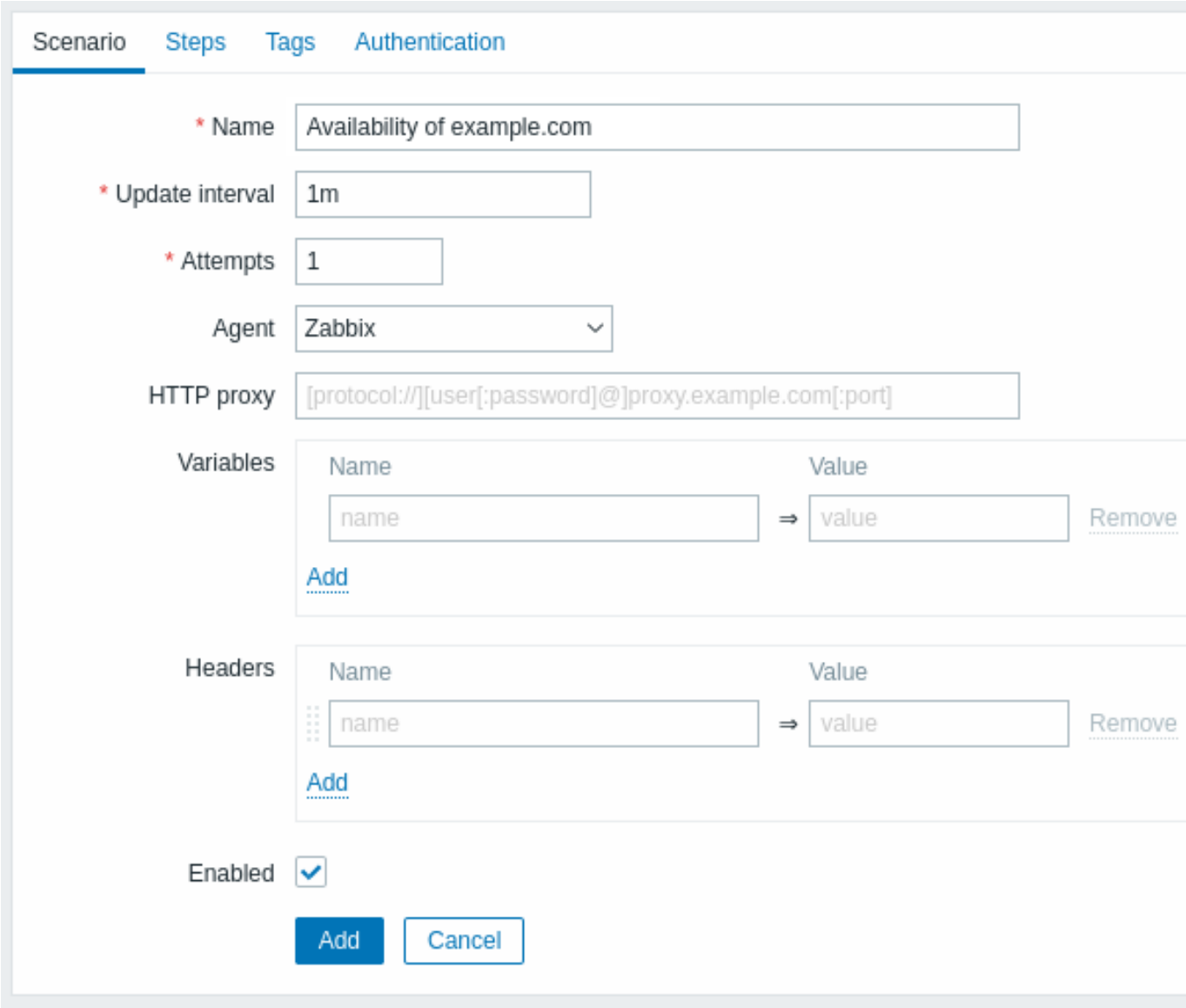
O Zabbix também pode verificar se uma página HTML recuperada contém uma string predefinida. Ele pode executar um login simulado e seguir um caminho de cliques simulados do mouse na página.

O monitoramento web do Zabbix oferece suporte a HTTP e HTTPS. Ao executar um cenário web, o Zabbix seguirá redirecionamentos opcionalmente (consulte a opção *Follow redirects* abaixo). O número máximo de redirecionamentos é fixado 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:

1. Acesse: *Coleta de dados* > *Hosts* (ou *Templates*).
2. Clique em *Web* na linha do host/template.
3. Clique em *Criar cenário web* à direita (ou no nome do cenário para editar um cenário existente).
4. Insira os parâmetros do cenário no formulário.

A aba **Cenário** permite configurar os parâmetros gerais de um cenário web.



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parâmetros do cenário:

Parâmetro	Descrição
Nome	Nome exclusivo do cenário. Macros de usuário são compatíveis. *Observe* que, 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 compatíveis, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são compatíveis. *Observe* que, 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 adiante no futuro com os valores do 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 (tempo limite, falta de 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 simulará o navegador selecionado. Isso é útil quando um site retorna conteúdo diferente para navegadores diferentes. Macros de usuário podem ser usadas neste campo.

Parâmetro	Descrição
Proxy HTTP	Você pode especificar um proxy HTTP a ser usado, utilizando 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). Quando nenhum protocolo é especificado, o proxy será tratado como um proxy HTTP. Por padrão, a porta 1080 será usada. Se especificado, o proxy substituirá variáveis de ambiente relacionadas a proxy, como http_proxy e HTTPS_PROXY. Se não for especificado, o proxy não substituirá as variáveis de ambiente relacionadas a proxy. O valor inserido é passado "como está", sem qualquer verificação de validade. Você também pode inserir um endereço de proxy SOCKS. Se especificar o protocolo incorreto, a conexão falhará e o item ficará sem suporte. *Observe* que somente a autenticação simples é compatível com proxy HTTP. Macros de usuário podem ser usadas neste campo.
Variáveis	Variáveis que podem ser usadas nas etapas do cenário (URL, variáveis de post). Elas têm o seguinte formato: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (consulte 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 as substituirá automaticamente pelos valores reais. Observe que variáveis com `regex:` precisam de uma etapa para obter o valor da expressão regular; portanto, o valor extraído só pode ser aplicado à etapa seguinte. Se a parte do valor começar com `regex:`, a parte posterior será tratada como uma expressão regular que pesquisa a página web e, se encontrada, armazena a correspondência na variável. Deve haver pelo menos um subgrupo para que o valor correspondente possa ser extraído. Macros de usuário e macros `{HOST.*}` são compatíveis. As variáveis são codificadas automaticamente em URL quando usadas em campos de consulta ou em dados de formulário para variáveis de post, mas devem ser codificadas manualmente em URL quando usadas em post bruto ou diretamente na URL.
Cabeçalhos	Os cabeçalhos HTTP são usados ao realizar uma solicitação. Podem ser usados cabeçalhos padrão e personalizados. Os cabeçalhos serão atribuídos usando 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. **Observe que definir o cabeçalho no nível da etapa descarta automaticamente todos os cabeçalhos definidos anteriormente, exceto um cabeçalho padrão atribuído pela seleção de 'User-Agent' em uma lista suspensa no nível do cenário.** No entanto, até mesmo o cabeçalho padrão 'User-Agent' pode ser substituído especificando-o no nível da etapa. Para remover o cabeçalho no nível do cenário, o cabeçalho deve ser nomeado e atribuído sem valor no nível da etapa. Os cabeçalhos devem ser listados usando a mesma sintaxe com que apareceriam no protocolo HTTP, opcionalmente usando alguns recursos adicionais compatíveis com a 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 macros `{HOST.*}` são compatíveis.

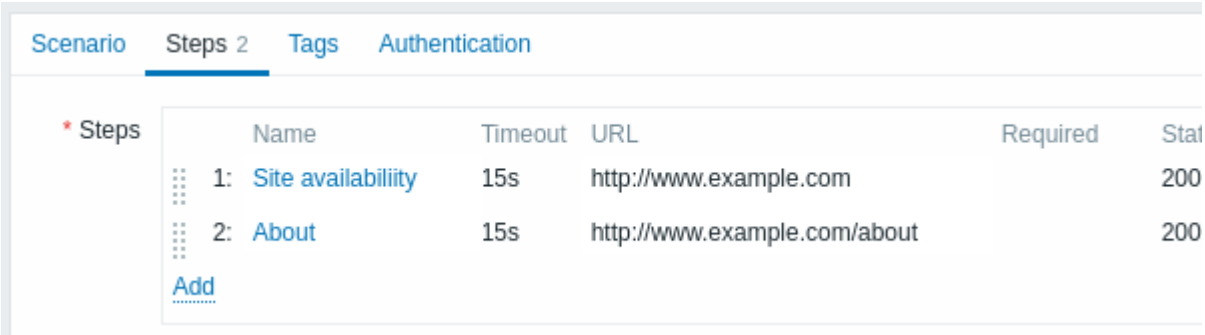
Parâmetro	Descrição
Habilitado	O cenário estará ativo se esta caixa estiver marcada; caso contrário, estará desabilitado.

Observe que, ao editar um cenário existente, dois botões adicionais ficam disponíveis no formulário:

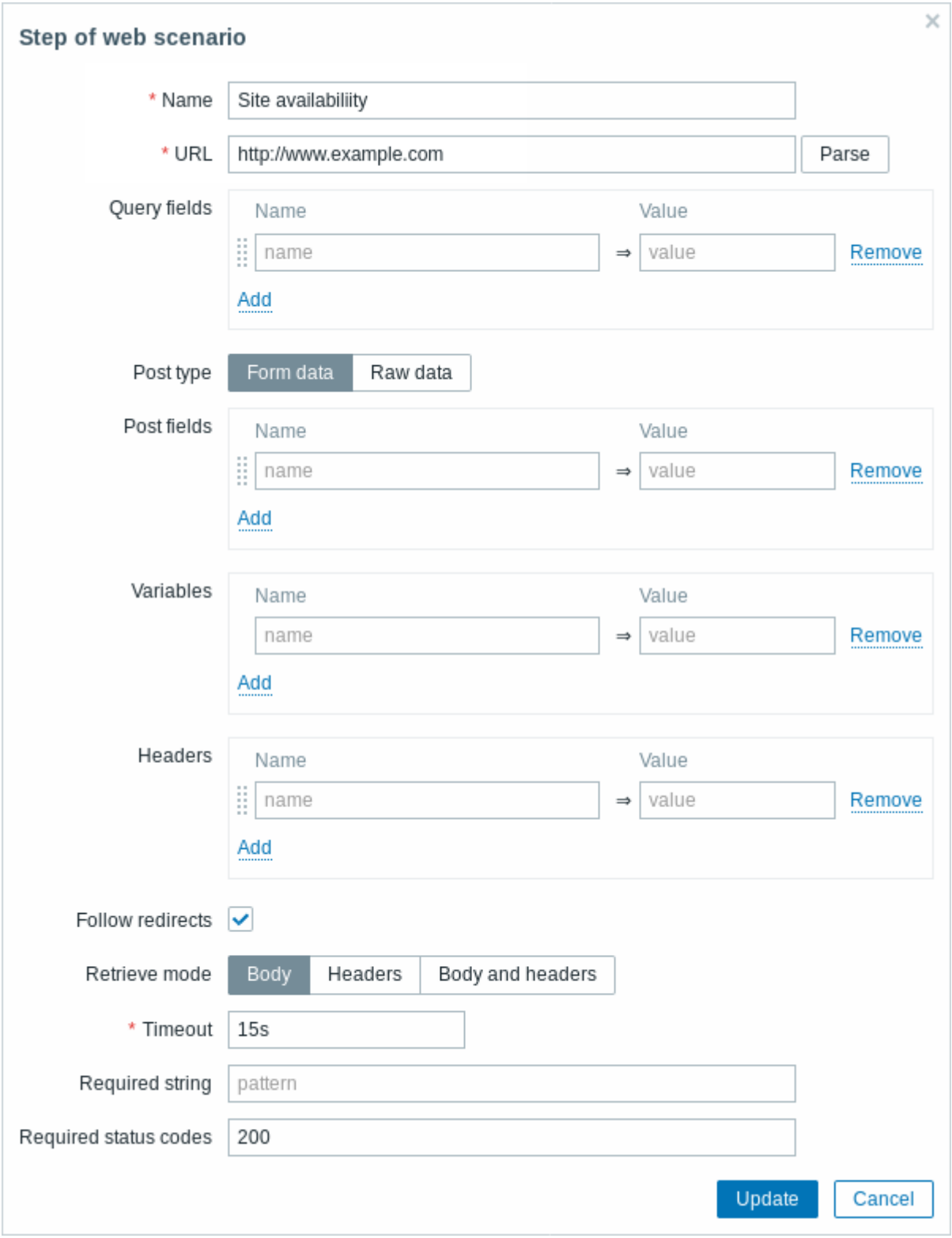
Clone	Cria outro cenário com base nas propriedades do cenário existente.
Clear history and trends	Exclui o histórico e os dados de tendência do cenário. Isso fará com que o server execute o cenário imediatamente após a exclusão dos dados.

Note:
Se o campo *Proxy HTTP* estiver vazio, outra forma de usar um proxy HTTP é definir variáveis de ambiente relacionadas a proxy.
Para verificações HTTP, defina a variável de ambiente `http_proxy` para o usuário do server do Zabbix. Por exemplo, `http_proxy=http://proxy_ip:proxy_port`.
Para verificações HTTPS, defina a variável de ambiente `HTTPS_PROXY`. Por exemplo, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Mais detalhes estão disponíveis executando um comando do 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 compatíveis. *Observe* que, se macros de usuário forem usadas, essas macros permanecerão não resolvidas nos nomes dos itens de monitoramento web.

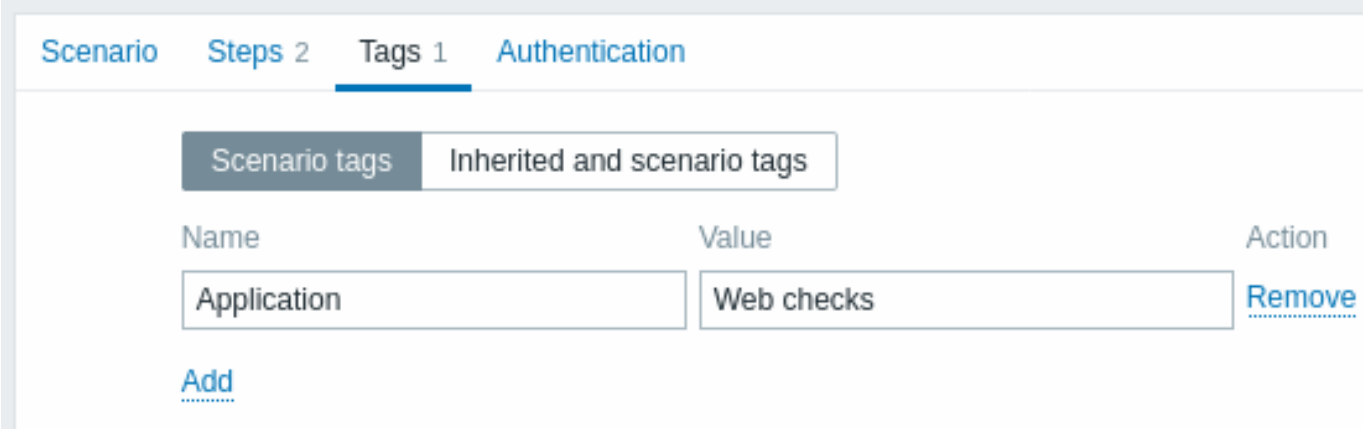
Parâmetro	Descrição
URL	URL à qual se conectar e da qual recuperar dados. Por exemplo: https://www.example.com http://www.example.com/download Nomes de domínio podem ser especificados usando caracteres Unicode. Eles são automaticamente convertidos para punycode ASCII ao executar a etapa do cenário web. O botão *Analisar* pode ser usado para separar campos de consulta opcionais (como `?key=value&limit=10`) da URL, movendo os atributos e valores para *Campos de consulta* para codificação automática da URL. Variáveis podem ser usadas na URL, utilizando a sintaxe {macro}. As variáveis podem ser codificadas manualmente na URL usando a sintaxe {{macro}.urlencode()}. Macros de usuário e macros `{HOST.*}` são compatíveis. Por motivos de segurança, não inclua nomes de usuário ou senhas reais nas URLs. Use macros de usuário para evitar a exposição das credenciais. Limitado a 2048 caracteres.
Campos de consulta	Variáveis HTTP GET para a URL. Especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Os valores de variáveis do cenário, macros de usuário ou macros `{HOST.*}` são resolvidos e, em seguida, codificados automaticamente na URL. Usar a sintaxe {{macro}.urlencode()} fará uma codificação dupla na URL. Macros de usuário e macros `{HOST.*}` são compatíveis.
Post	Variáveis HTTP POST. No modo **Dados do formulário**, especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Os valores de variáveis do cenário, macros de usuário ou macros `{HOST.*}` são resolvidos e, em seguida, codificados automaticamente na URL. No modo **Dados brutos**, os atributos/valores são exibidos em uma única linha e concatenados com o símbolo **&**. Os valores brutos podem ser codificados/decodificados manualmente na URL usando a sintaxe {{macro}.urlencode()} ou {{macro}.urldecode()}. Por exemplo: id=2345&userid={user} Se {user} for definido como uma variável do cenário web, ele será substituído pelo seu valor quando a etapa for executada. Se quiser codificar a variável na URL, substitua {user} por {{user}.urlencode()}. Macros de usuário e macros `{HOST.*}` são compatíveis.
Variáveis	Variáveis no nível da etapa que podem ser usadas nas funções GET e POST. Especificadas como pares de atributo e valor. As variáveis no nível da etapa substituem as variáveis no nível do cenário ou as variáveis da etapa anterior. No entanto, o valor de uma variável no nível da etapa afeta somente a etapa seguinte (e não a etapa atual). Elas têm o seguinte formato: **{macro}**=value **{macro}**=regex:<regular expression> Para obter mais informações, consulte a descrição das variáveis no nível do cenário. As variáveis são codificadas automaticamente na URL quando usadas em campos de consulta ou em dados de formulário para variáveis POST, mas devem ser codificadas manualmente na URL quando usadas em POST bruto ou diretamente na URL.
Cabeçalhos	Cabeçalhos HTTP personalizados que serão enviados ao realizar uma solicitação. Especificados como pares de atributo e valor. Um cabeçalho definido no nível da etapa será usado especificamente nessa etapa. **Observe que definir o cabeçalho no nível da etapa descarta automaticamente todos os cabeçalhos definidos anteriormente, exceto um cabeçalho padrão atribuído ao selecionar 'User-Agent' em uma lista suspensa no nível do cenário.** No entanto, até mesmo o cabeçalho padrão 'User-Agent' pode ser substituído especificando-o no nível da etapa. Por exemplo, atribuir um nome ao cabeçalho, mas não definir um valor, removerá o cabeçalho padrão no nível do cenário. Macros de usuário e macros `{HOST.*}` são compatíveis. Isso define a opção `CURLOPT_HTTPHEADER` do cURL.
Seguir redirecionamentos	Marque a caixa de seleção para seguir redirecionamentos HTTP. Isso define a opção CURLOPT_FOLLOWLOCATION do cURL.

Parâmetro	Descrição
Modo de recuperação	Selecione o modo de recuperação: **Corpo** - recupera somente o corpo da resposta HTTP **Cabeçalhos** - recupera somente os cabeçalhos da resposta HTTP **Corpo e cabeçalhos** - recupera o corpo e os cabeçalhos da resposta HTTP
Tempo limite	O Zabbix não gastará mais do que o tempo definido processando a URL (de um segundo até o máximo de 1 hora). Na prática, esse parâmetro define o tempo máximo para estabelecer a conexão com a URL e o tempo máximo para realizar uma solicitação HTTP. Portanto, o Zabbix não gastará mais do que **2 x Tempo limite** segundos na etapa. Sufixos de tempo são compatíveis, por exemplo, 30s, 1m, 1h. Macros de usuário são compatíveis.
String obrigatória	Padrão de expressão regular obrigatório. A etapa falhará se o conteúdo recuperado (HTML) não corresponder ao padrão obrigatório. Se estiver vazio, nenhuma verificação da string obrigatória será realizada. Por exemplo: Homepage of Zabbix Welcome.*admin *Observação*: não há suporte neste campo para referências a expressões regulares criadas no frontend do Zabbix. Macros de usuário e macros `{HOST.*}` são compatíveis.
Códigos de status obrigatórios	Lista dos códigos de status HTTP esperados. Se o Zabbix receber um código que não esteja na lista, a etapa falhará. Se estiver vazia, nenhuma verificação dos códigos de status será realizada. Por exemplo: 200,201,210-299 Macros de usuário são compatíveis.

Note:
Todas as alterações nas etapas do cenário web somente serão salvas quando o cenário inteiro for salvo.

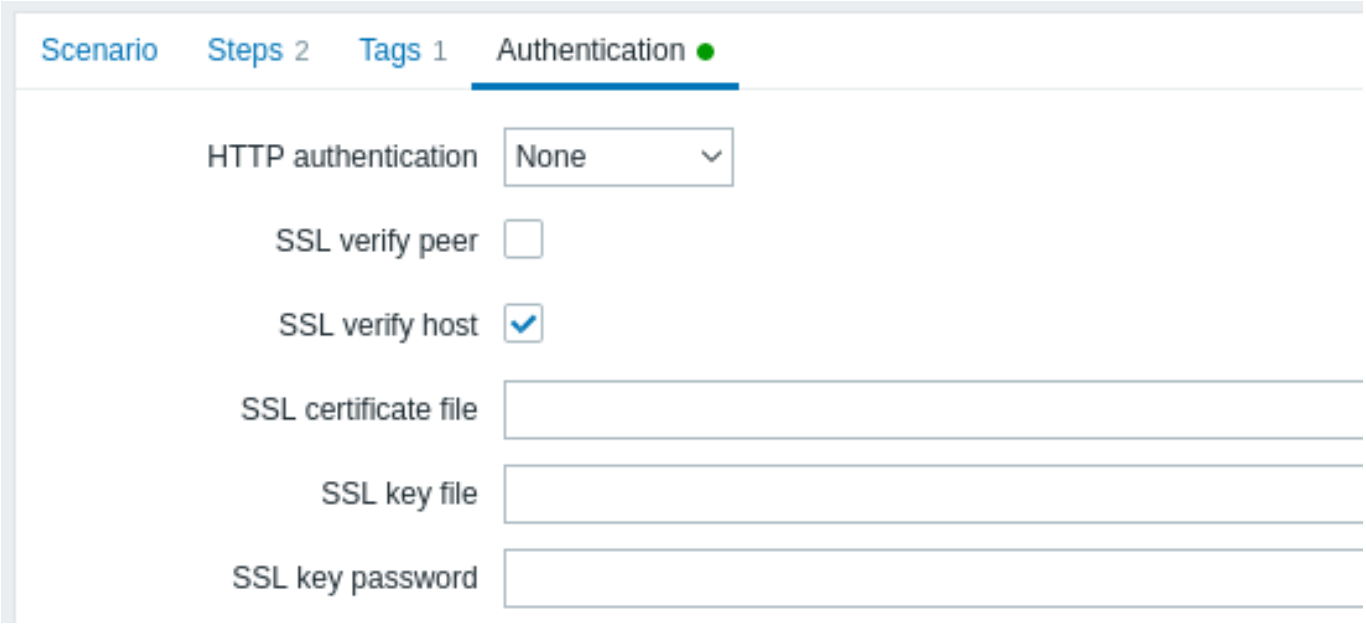
Consulte também um exemplo prático de como as etapas de monitoramento web podem ser configuradas.

Configurando tags A aba **Tags** permite definir tags no nível do cenário.



A utilização de tags permite filtrar cenários web e itens de monitoramento web.

Configurando a 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** - a autenticação básica é usada; **NTLM** - a autenticação NTLM (Windows NT LAN Manager) é usada; **Kerberos** - a autenticação Kerberos é usada (consulte também: Configurando o Kerberos com o Zabbix); **Digest** - a autenticação Digest é usada.
Usuário	Insira o nome de usuário (até 255 caracteres). Este campo está disponível quando *Autenticação HTTP* está definida como Básica, NTLM, Kerberos ou Digest. Macros de usuário são compatíveis.
Senha	Insira a senha do usuário (até 255 caracteres). Este campo está disponível quando *Autenticação HTTP* está definida como Básica, NTLM, Kerberos ou Digest. Macros de usuário são compatíveis.
Verificar peer SSL	Marque a caixa de seleção para verificar o certificado SSL do web server. O certificado do server é verificado em relação aos certificados de autoridade certificadora (CA) confiáveis do local de CA em todo o sistema. É possível substituir o local dos arquivos de CA usando o parâmetro de configuração do Zabbix server ou proxy SSLCALocation. Isso define a opção cURL CURLOPT_SSL_VERIFYPEER.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do web server 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* vazio. Se a chave estiver criptografada, especifique a senha no campo *Senha da chave SSL*. O diretório que contém esse 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 esse 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 oferece suporte apenas a arquivos de certificado e chave privada no formato PEM. Caso os dados do certificado e da chave privada estejam em um arquivo no formato PKCS #12 (geralmente com a extensão *.p12 ou *.pfx), você pode gerar o arquivo PEM usando os seguintes comandos:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

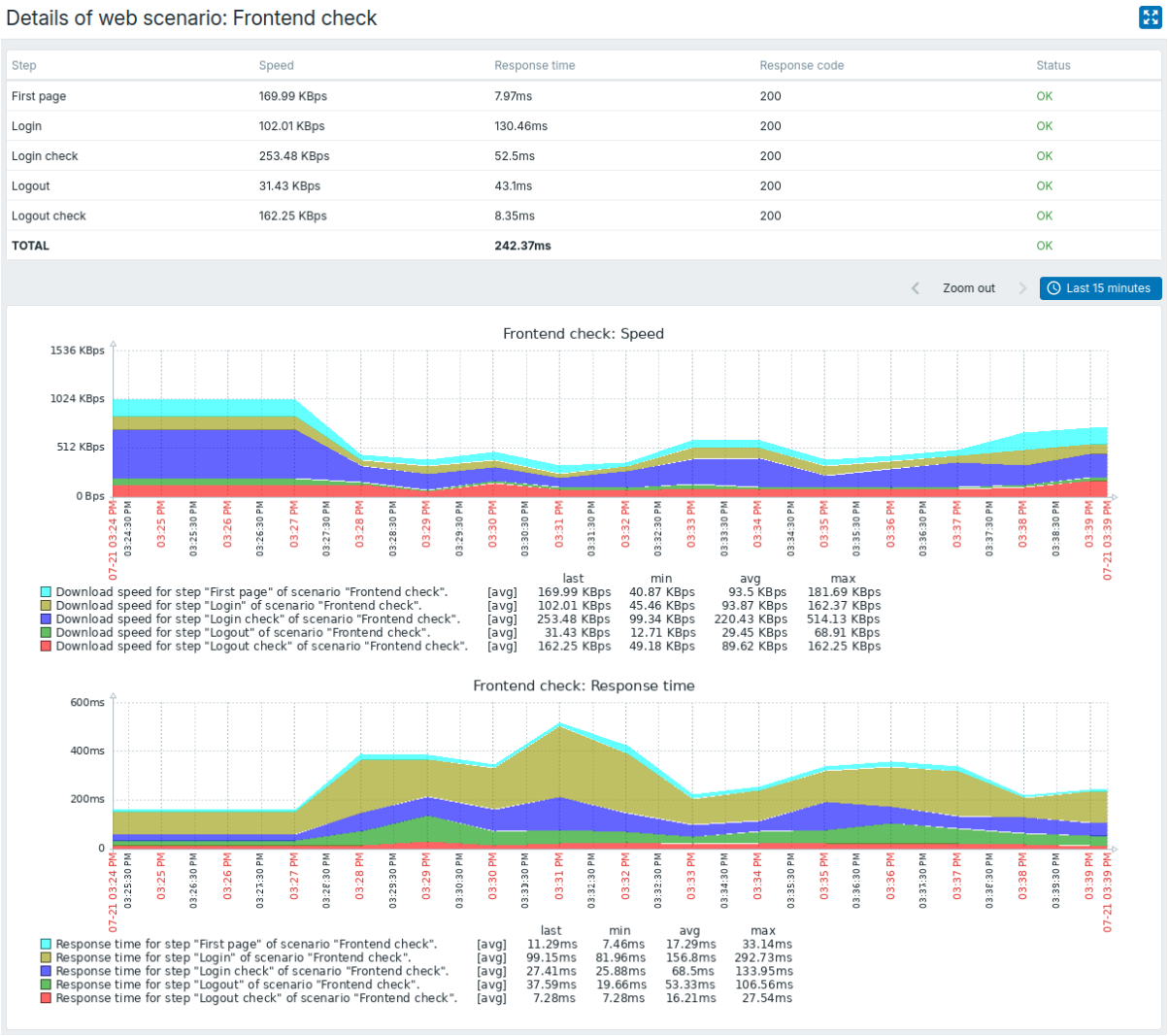
Note:
O Zabbix server detecta alterações nos certificados sem necessidade de reinicialização.

Note:
Se o certificado do cliente e a chave privada estiverem em um único arquivo, basta especificá-lo no campo "Arquivo de certificado SSL" e deixar o campo "Arquivo de chave SSL" vazio. O certificado e a chave ainda devem estar no formato PEM. É fácil combinar o certificado e a chave:

```
cat client.crt client.key > client.pem
```

Exibição Para visualizar os cenários web configurados para um host:

1. Acesse *Monitoring* > *Hosts*.
2. Localize o host na lista e clique no hiperlink *Web* na última coluna.
3. 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 *Monitoring* > *Latest data*.

Monitoramento estendido Às vezes, é necessário registrar o conteúdo HTML da página recebida.
Isso é especialmente útil se alguma etapa de um cenário web falhar.

O nível de depuração 5 (trace) atende a essa finalidade.
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:

```
# Increase log level of all http pollers:
zabbix_server -R log_level_increase="http poller"

# Increase log level of second 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`.

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`

Cenário da vida real

Visão geral

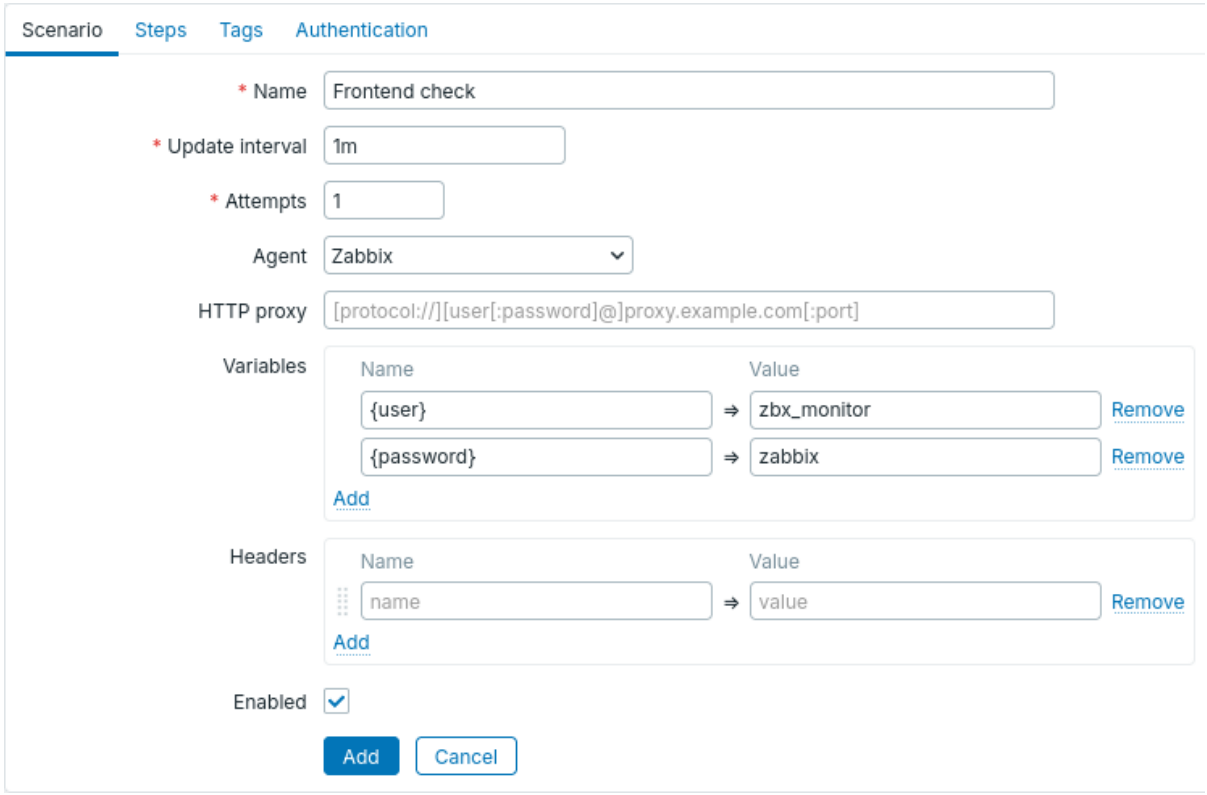
Esta seção apresenta um exemplo prático e real, passo a passo, de como o monitoramento web pode ser usado.

O monitoramento web do Zabbix será usado para monitorar o frontend do Zabbix. O objetivo é determinar se ele está disponível, se fornece o conteúdo correto e com que rapidez ele funciona. Para isso, várias etapas são necessárias, incluindo verificar a disponibilidade da primeira página, fazer login com nome de usuário e senha, verificar se o login foi bem-sucedido, fazer logout e confirmar o logout.

Cenário

Adicionar um novo cenário web

1. Acesse *Data collection* > *Hosts*.
2. Escolha um host e clique em *Web* na linha desse host.
3. Clique em *Create web scenario*.



4. No formulário do novo cenário, na aba *Scenario*, preencha os seguintes campos:

    - *Name*: Frontend check.
    - *Update interval*: 1m.
    - *Attempts*: 1.
    - *Agent*: Zabbix.
 Na seção *Variables*, adicione os seguintes pares nome-valor:
    - `{user}`: suas credenciais existentes de usuário do Zabbix;
    - `{password}`: suas credenciais existentes de usuário do Zabbix.

Note:
Por motivos de segurança, é recomendável criar um usuário separado com permissões mínimas para uso em monitoramento.

Opcionalmente, vá para a aba *Tags* e adicione tags do cenário web. Você pode usar tags de cenário web para identificar rapidamente items e triggers relacionados ou pesquisar nos dados coletados. Por exemplo, tags adequadas para este tutorial são `component: web-scenario` e/ou `target: frontend`.

Configurar etapas do cenário web

Alterne para a aba *Steps* e defina as etapas para o cenário.
 Após configurar cada etapa, clique no botão *Add*.

Attention:
Observe que o frontend do Zabbix usa redirecionamento JavaScript ao fazer login, portanto o login deve ocorrer primeiro, e os recursos disponíveis após o login só podem ser verificados nas etapas seguintes. Além disso, a etapa de login deve usar a URL completa para o arquivo **index.php**.

Campos comuns a todas as etapas

Além dos campos específicos de cada etapa, preencha os seguintes campos comuns para cada etapa descrita abaixo:

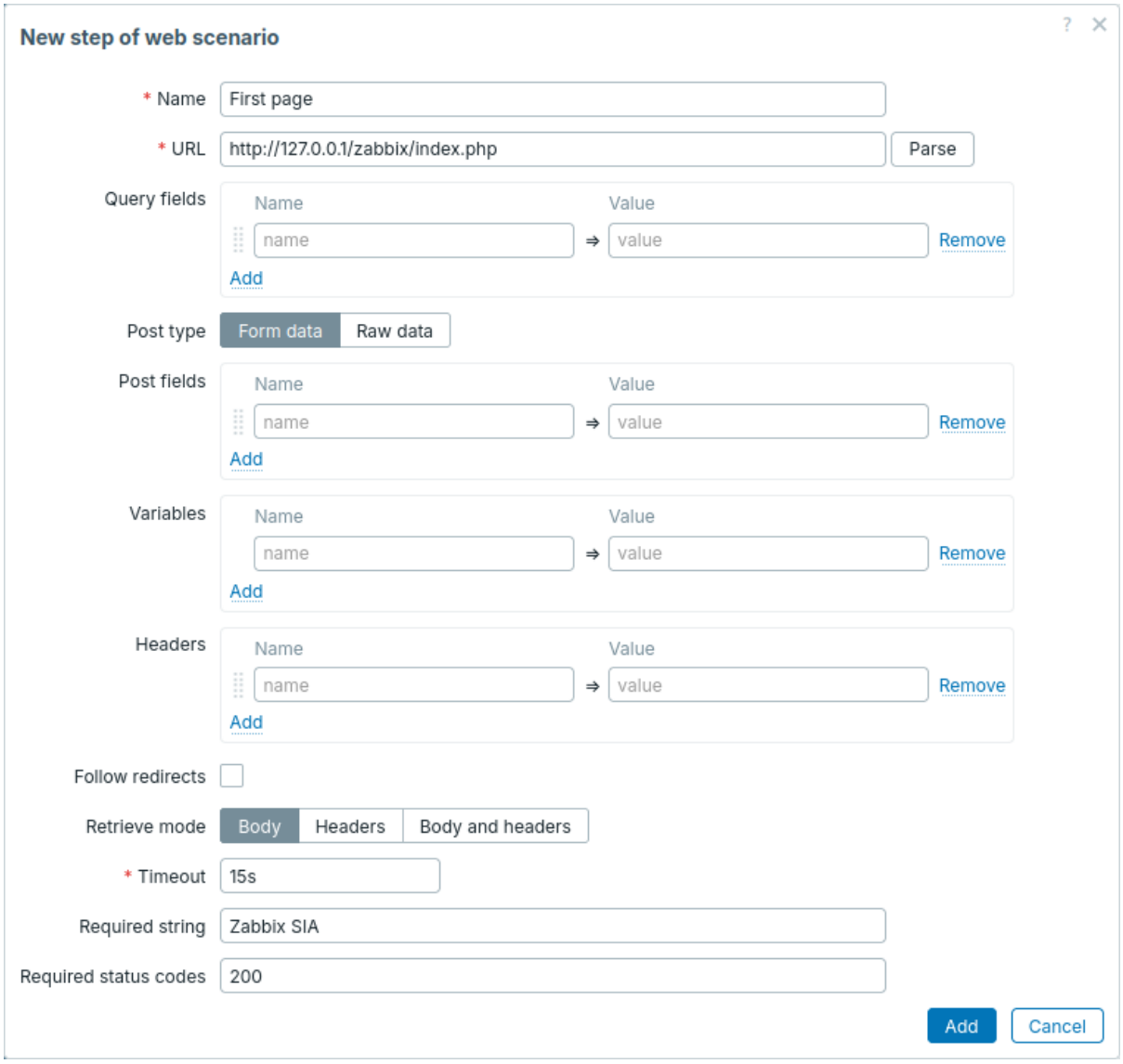
- *URL*: a URL do Zabbix frontend (`http://<server_ip_or_name>/zabbix/index.php` (para Apache) ou `http://<server_ip_or_n` (para NGINX)). Para a etapa 3 do cenário Web, use uma URL diferente, conforme descrito abaixo.
- *Timeout*: 15s.
- *Códigos de status obrigatórios*: 200.

Etapa 1 do cenário web

Verifique se a primeira página responde corretamente, retorna o código de status HTTP 200 e contém o texto *Zabbix SIA*.
 Preencha os seguintes campos:

- *Name*: First page.

- *Timeout*: 15s.
- *Required string*: Zabbix SIA.
- Os campos comuns para todas as etapas.

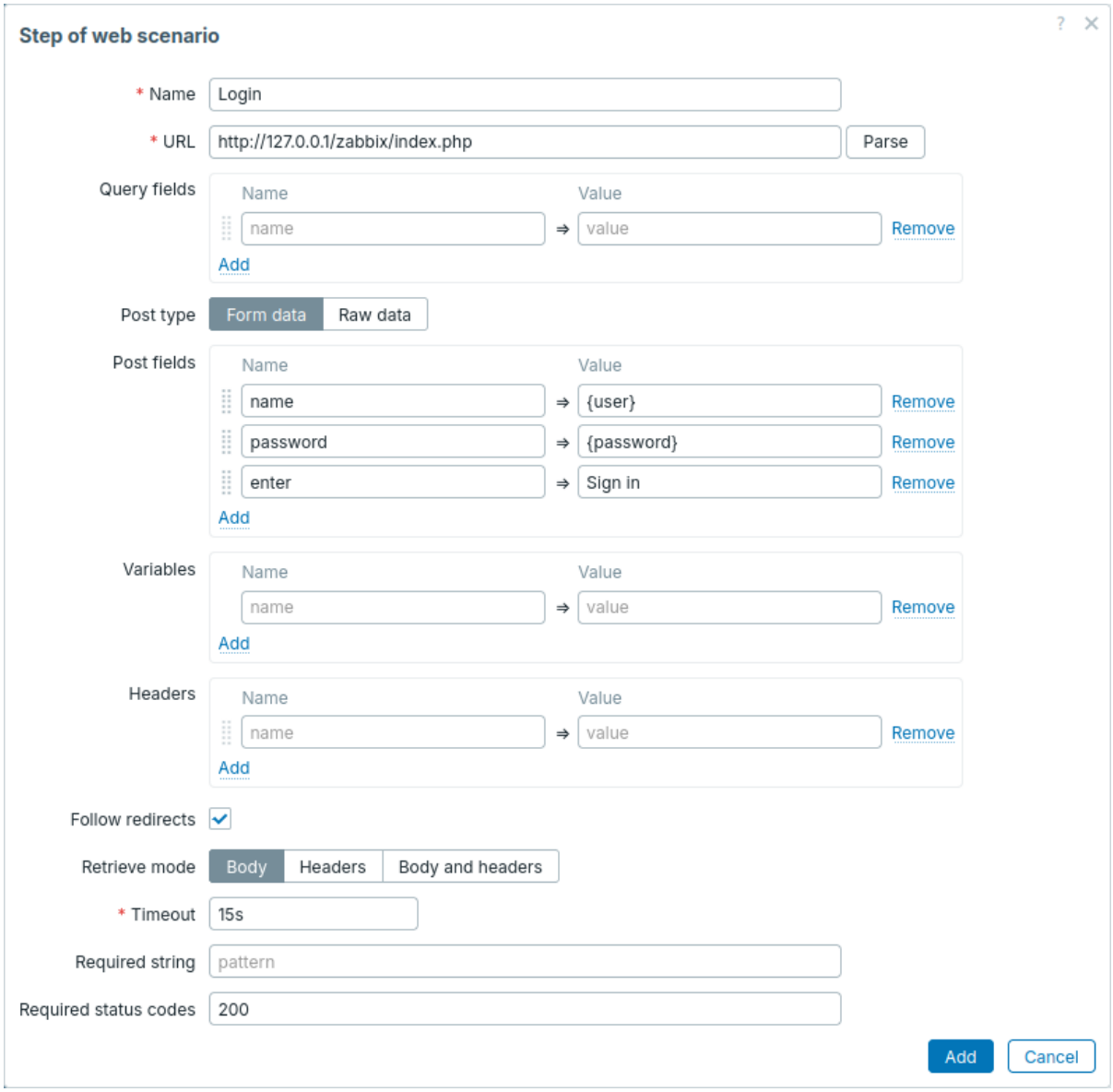


Etapa 2 do cenário web

Faça login no frontend do Zabbix usando as macros (variáveis) definidas na aba *Scenario*: *{user}* e *{password}*.
 Preencha os seguintes campos:

- *Name*: Login.
- Na seção *Post fields*, adicione os seguintes pares nome-valor:
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- Os campos comuns para todas as etapas.

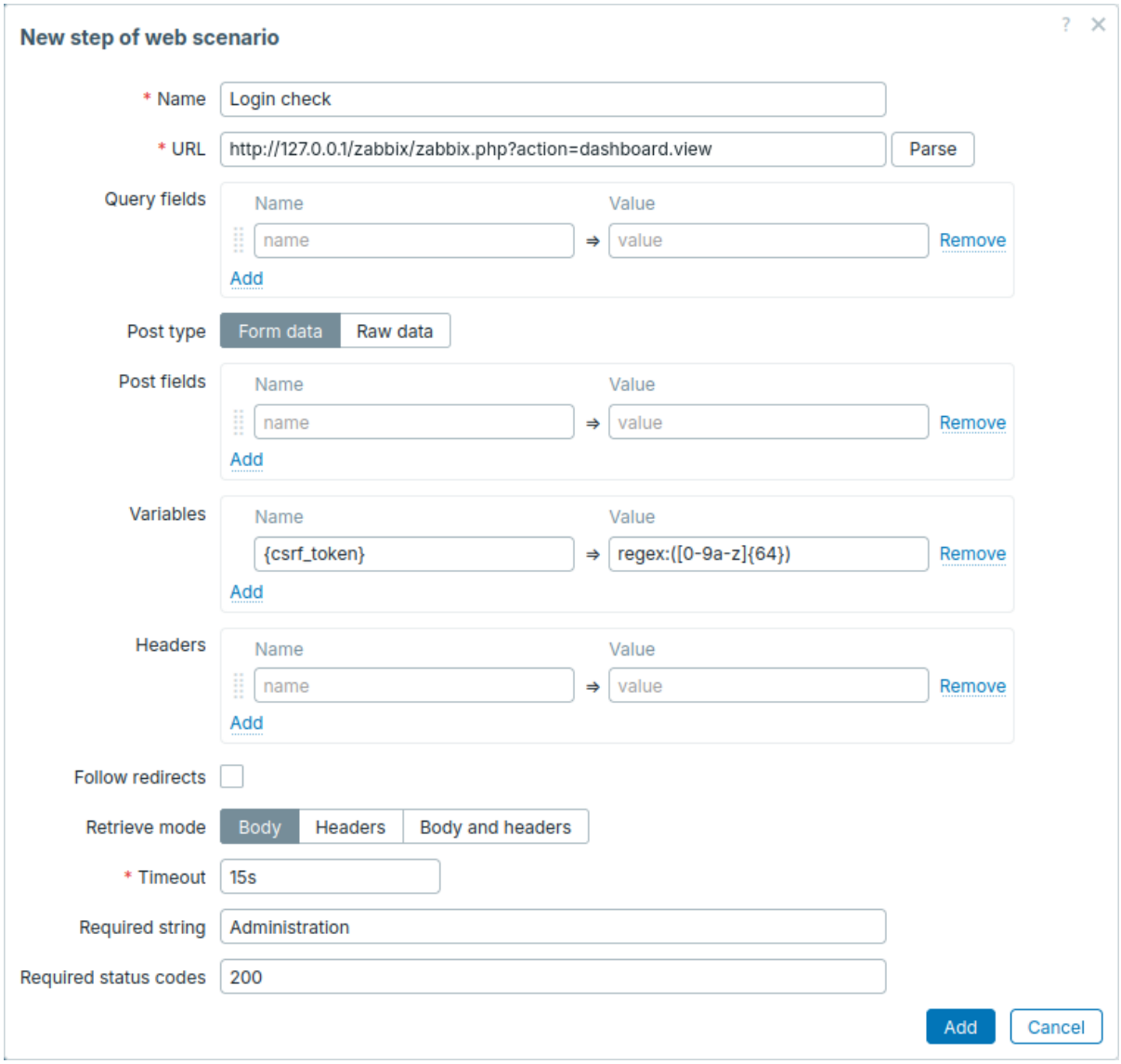
Selecione a caixa de seleção *Follow redirects*.



Etapa 3 do cenário web

Após fazer login, verifique o sucesso procurando uma string visível somente quando o usuário está autenticado, por exemplo, *Administration*.
 Preencha os seguintes campos:

- *Name*: Login check.
- *URL*: a URL para a qual o usuário é redirecionado após o login (por exemplo, http://127.0.0.1/zabbix/zabbix.php?action=dashboard.view).
- Na seção *Variables*, adicione o par nome-valor `{csrf_token}`: `regex:([0-9a-z]{64})`. Essa variável capturará o valor do token CSRF atribuído para reutilização na etapa 4 do cenário web.
- *Required string*: Administration.
- Os campos comuns para todas as etapas.



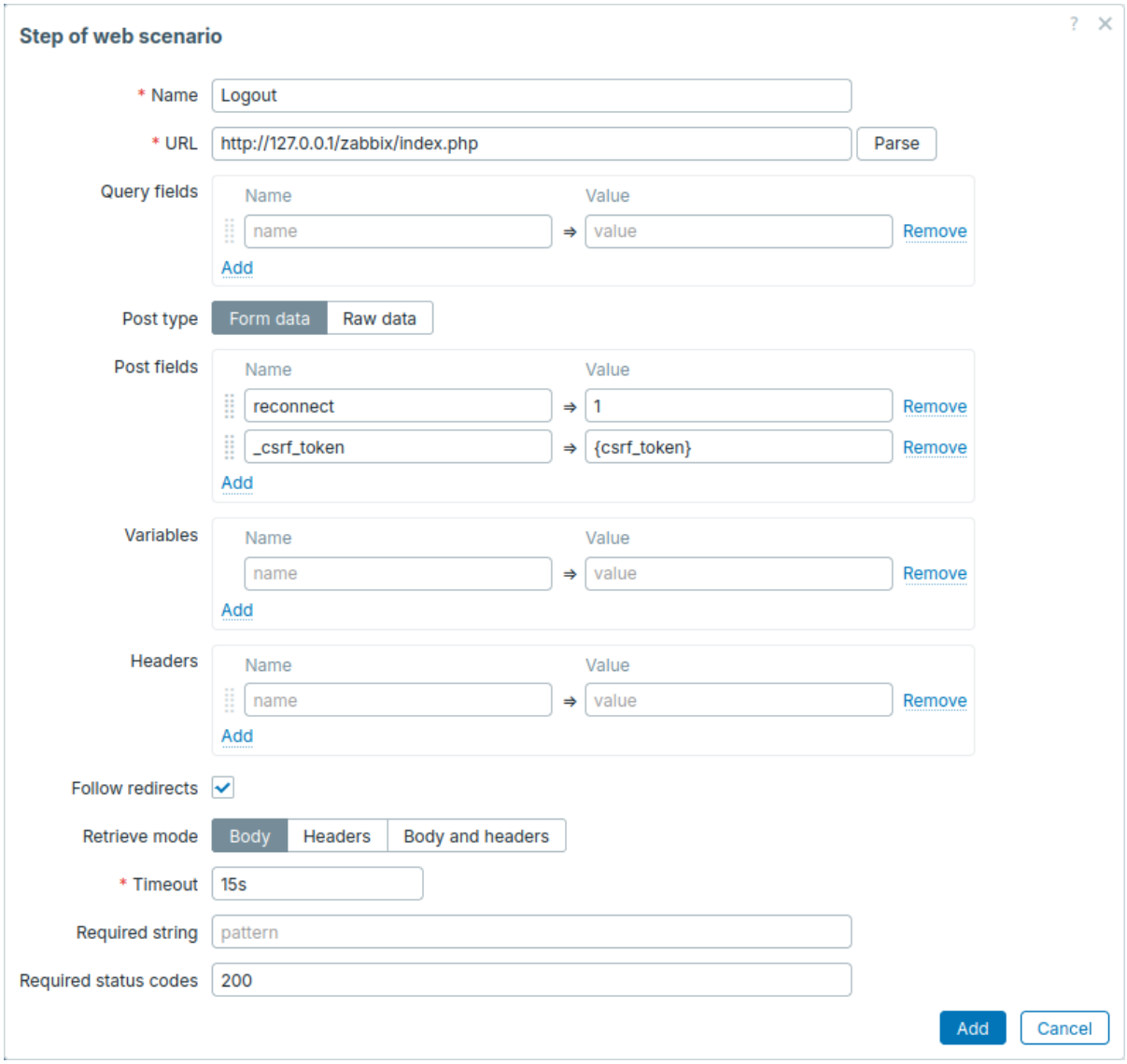
Etapa 4 do cenário web

Depois de verificar a acessibilidade e o login do frontend, adicione uma etapa de logout; caso contrário, o banco de dados do Zabbix ficará cheio de muitos registros de sessão abertos.
 Preencha os seguintes campos:

- *Nome*: Logout.
- Na seção *Campos de postagem*, adicione os seguintes pares nome-valor:
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`.
- Os campos comuns a todas as etapas.

Selecione a caixa de seleção *Seguir redirecionamentos*.

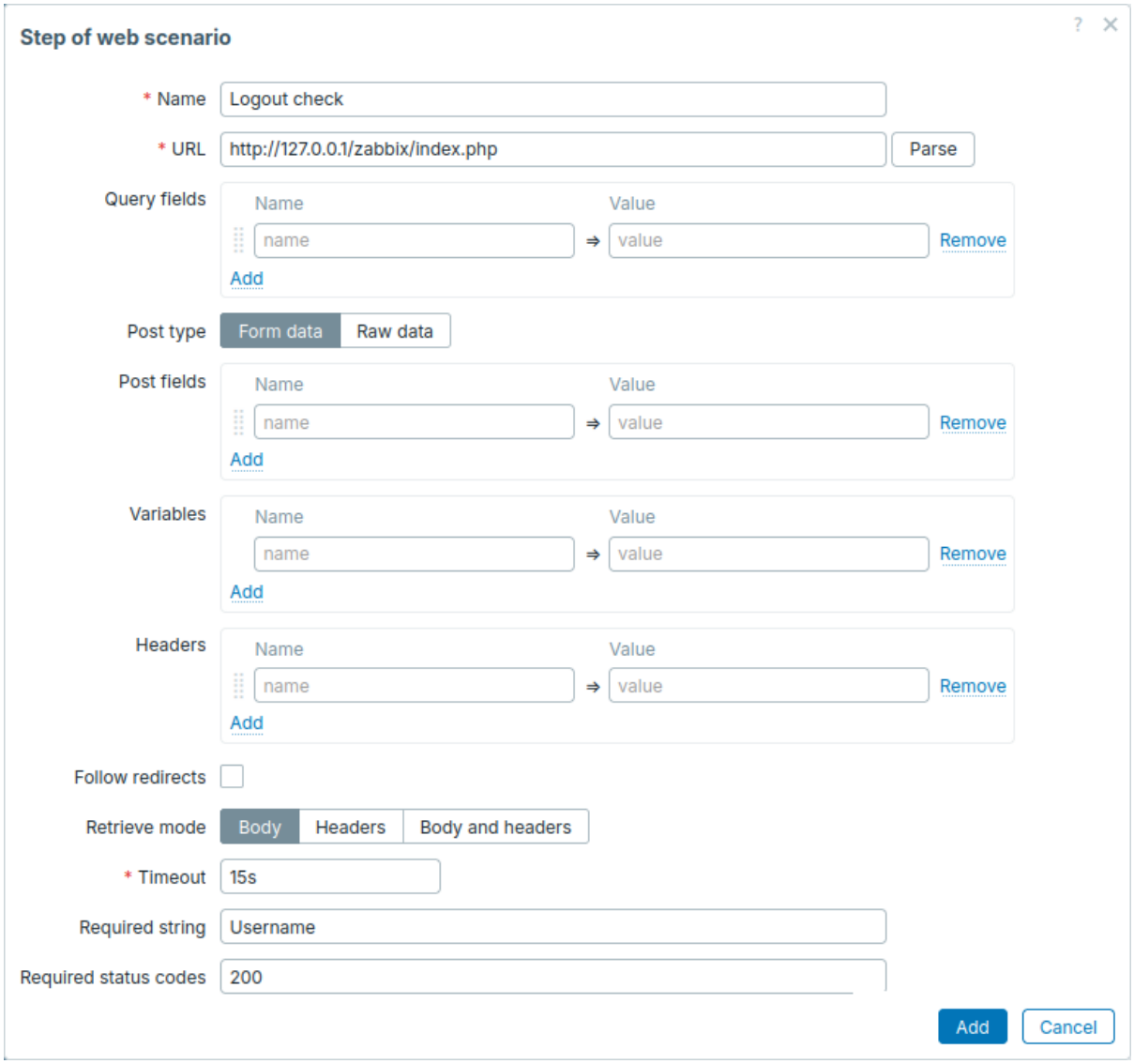
Esta etapa usa a variável `{csrf_token}` obtida na etapa 3 do cenário web.



Etapa 5 do cenário web

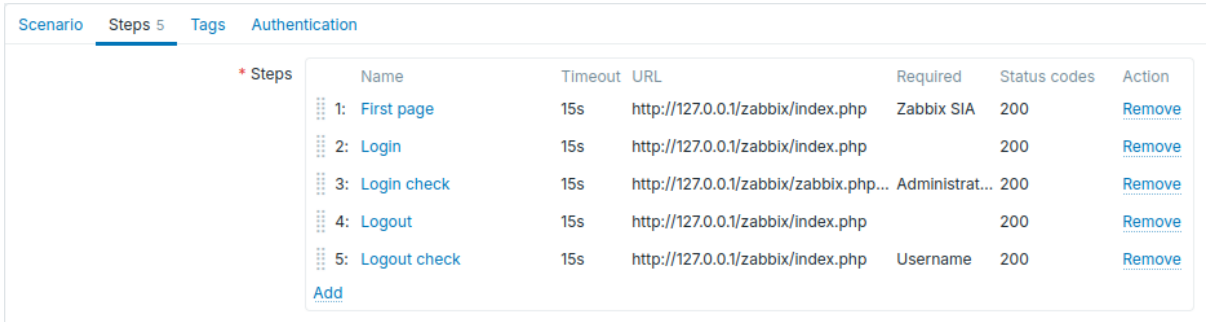
Para confirmar o logout, verifique a string *Username*.
 Preencha os seguintes campos:

- *Name*: Logout check.
- *Required string*: Username.
- Os campos comuns para todas as etapas.



Configuração completa dos passos

Uma configuração completa dos passos de um cenário web deve ser semelhante a esta:

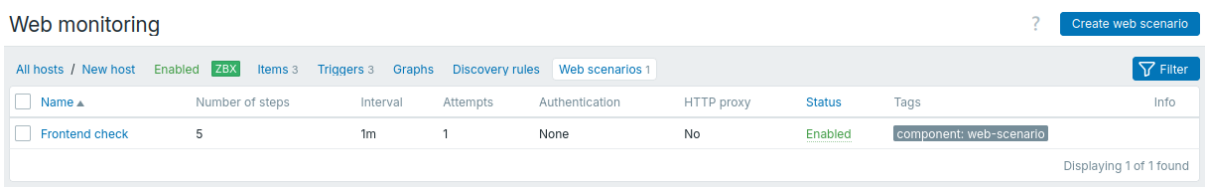


Verifique os resultados

Salve o cenário de monitoramento web concluído.

Quando o cenário web é criado, novos itens de monitoramento web são adicionados automaticamente para monitoramento.

O cenário será adicionado ao host. Para ver as informações do cenário web, vá para *Monitoring* > *Hosts*, localize o host na lista e clique no hiperlink *Web* na última coluna.



Clique no nome do cenário para ver estatísticas detalhadas:



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 consultar outras informações em um nível mais alto. O monitoramento de serviços concentra-se 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 funcionalidades para encontrar a causa raiz de um problema quando um serviço não está funcionando conforme o esperado.

O monitoramento de serviços oferece uma maneira de criar uma representação hierárquica dos dados monitorados.

Uma estrutura de serviço muito simples pode ser semelhante a:

```
Serviço
|
|-Estações de trabalho
| |
| |-Estação de trabalho1
| |
| |-Estação de trabalho2
|
|-Servidores
```

Cada nó da estrutura tem um atributo de status. O status é calculado e propagado para os níveis superiores de acordo com o algoritmo selecionado. O status de nós individuais é afetado pelo status dos problemas mapeados. O mapeamento de problemas é realizado com tags.

O Zabbix pode enviar notificações ou executar automaticamente um script no Zabbix server caso seja detectada uma alteração no estado do serviço. É possível definir regras flexíveis para determinar se um serviço pai deve entrar em um "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 criar uma hierarquia da 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 estado do serviço.

- SLA

Nesta seção, você pode definir acordos de nível de serviço e estabelecer 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:

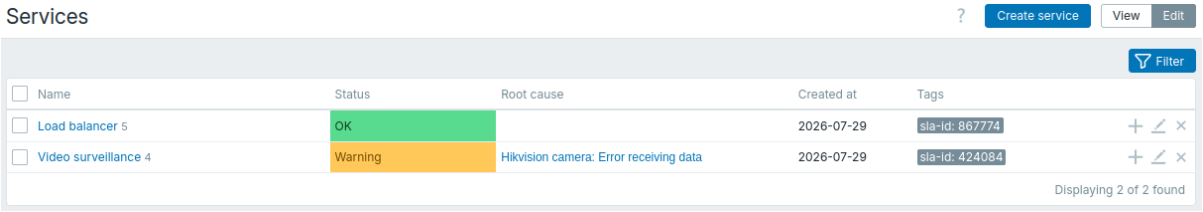
- enviar uma notificação de que um serviço está indisponível
- executar um comando remoto no Zabbix server após uma alteração no estado do serviço
- enviar uma notificação de recuperação quando um serviço estiver disponível novamente.

Veja também:

- Exemplo de configuração do monitoramento de SLA
- Observações sobre a atualização de serviços de versões do Zabbix anteriores à 6.0

Árvore de serviços

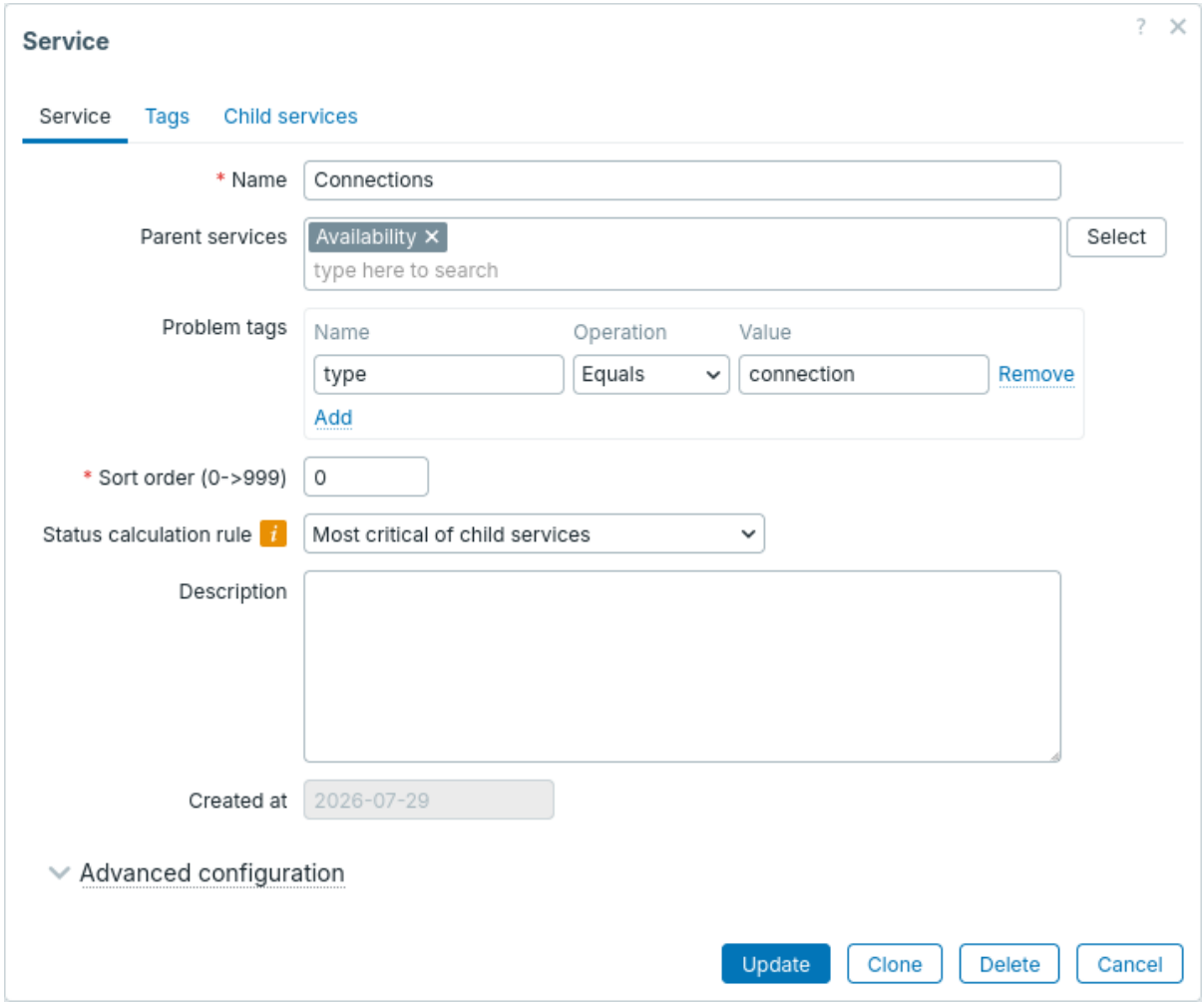
A árvore de serviços é configurada na seção de menu *Services* > *Services*. No canto superior direito, alterne de view para o modo de edição.



Para configurar um novo serviço, clique no botão *Create service* no canto superior direito.

Para adicionar rapidamente um serviço filho, você também pode clicar no ícone de mais ao lado do serviço pai. Isso abrirá o mesmo formulário de configuração de serviço, mas o parâmetro *Parent services* será preenchido automaticamente.

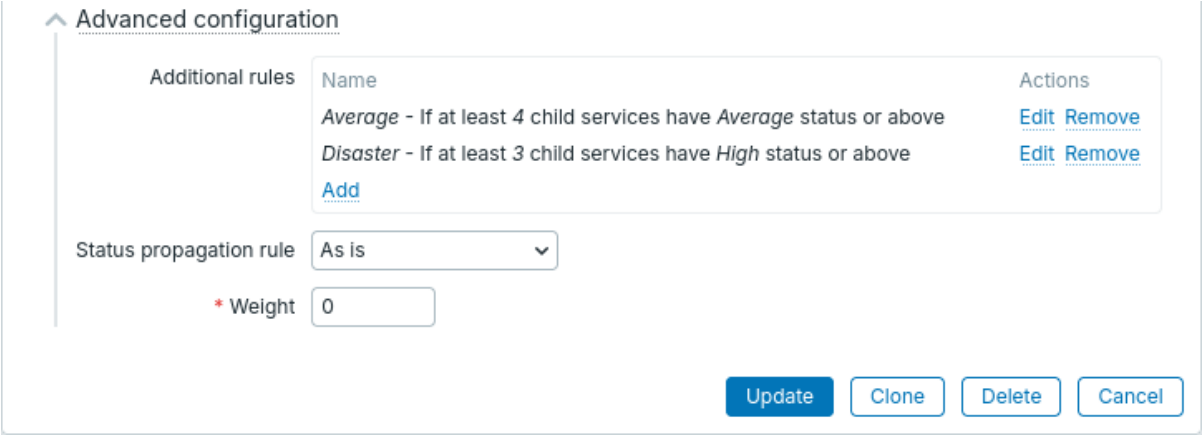
Configuração do service Na aba **Service**, especifique os parâmetros obrigatórios do service:



Todos os campos de entrada obrigatórios são marcados com um asterisco vermelho.

Parameter	Description
Name	Nome do service.
Parent services	Services pai aos quais o service pertence. Deixe este campo em branco se estiver adicionando o service de nível mais alto. Um service pode ter vários services pai. Nesse caso, ele será exibido na árvore de services abaixo de cada um dos services pai.
Problem tags	Especifique tags para mapear os dados de problema para o service. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há dois operadores disponíveis para cada condição: **Equals** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas) **Contains** - inclui os nomes de tag especificados em que os valores da tag contêm a string informada (correspondência de substring, diferencia maiúsculas de minúsculas)
Sort order	Ordem de classificação para exibição; o menor vem primeiro.
Status calculation rule	Regra para calcular o status do service: **Most critical if all children have problems** - o problema mais crítico nos services filhos é usado para colorir o status do service, se todos os filhos tiverem problemas **Most critical of child services** - o problema mais crítico nos services filhos é usado para colorir o status do service **Set status to OK** - não calcula o status do service Regras adicionais de cálculo de status podem ser configuradas nas opções de advanced configuration.
Description	Descrição do service.
Created at	O momento em que o service foi criado; exibido ao editar um service existente.
Advanced configuration	Clique no cabeçalho *Advanced configuration* para exibir as opções de advanced configuration.

Configuração avançada



Parameter	Description
Additional rules	Clique em *Add* para configurar regras adicionais de cálculo de status.
Set status to	Defina o status do service como *OK* (padrão), *Not classified*, *Information*, *Warning*, *Average*, *High* ou *Disaster* em caso de correspondência com a condição.
Condition	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, será definido o nível de severidade mais alto.
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), *Not classified*, *Information*, *Warning*, *Average*, *High* ou *Disaster*.
Status propagation rule	Regra para propagar o status do service para o service pai: **As is** - o status é propagado sem alteração **Increase by** - você pode aumentar o status propagado em 1 a 5 níveis de severidade **Decrease by** - você pode diminuir o status propagado em 1 a 5 níveis de severidade **Ignore this service** - o status não é propagado para o service pai **Fixed status** - o status é propagado estaticamente, ou seja, sempre com o mesmo valor
Weight	Peso do service (inteiro no intervalo de 0 (padrão) a 1000000).

Note:
As regras adicionais de cálculo de status só podem ser usadas para aumentar o nível de severidade acima do nível calculado de acordo com o parâmetro principal *Status calculation rule*. Se, de acordo com as regras adicionais, o status devesse ser Warning, mas, de acordo com a *Status calculation rule*, o status for Disaster - o service terá status Disaster.

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 quiser adicionar um novo serviço filho, salve este serviço primeiro e, em seguida, clique no ícone de adição ao lado do serviço que você acabou de criar.

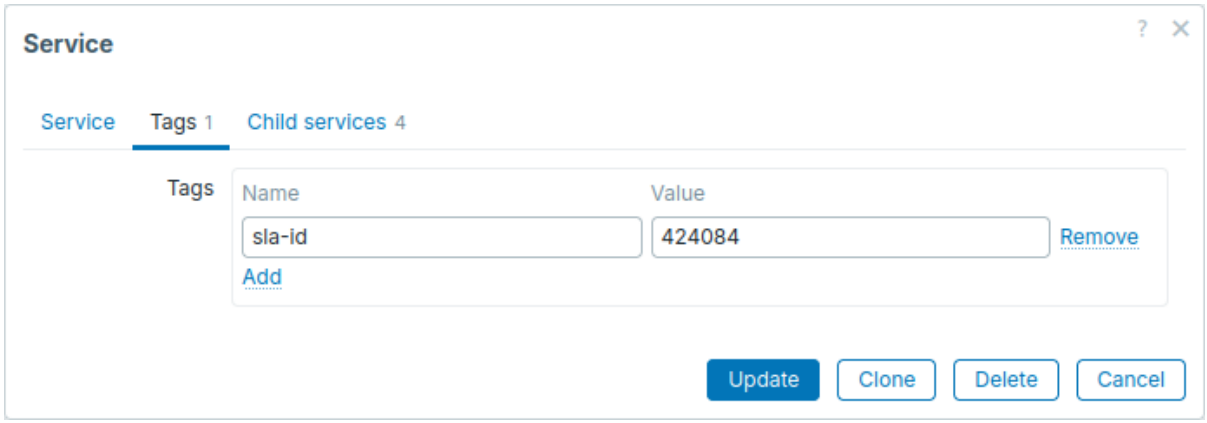
Tags Existem dois tipos diferentes de tags em services:

- Service tags
- Problem tags

Service tags

Service tags são usadas para associar services com service actions e SLAs. Essas tags são especificadas na aba de configuração *Tags* do service. Para o mapeamento de SLAs, é usada a lógica *OR*: um service será mapeado para um SLA se tiver pelo menos

uma tag correspondente. Em service actions, as regras de mapeamento são configuráveis e podem usar a lógica *AND*, *OR* ou *AND/OR*.

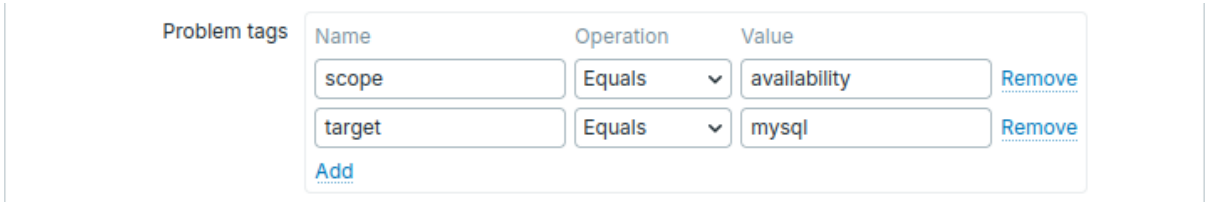


Tags de problema

As tags de problema são usadas para corresponder problemas e services. Essas tags são especificadas na guia de configuração do service primário.

Somente services filhos do nível mais baixo da hierarquia podem ter tags de problema definidas e ser correlacionados diretamente com problemas. Se as tags de problema corresponderem, o status do service mudará para o mesmo status do problema. No caso de vários problemas, um service terá o status do problema mais grave. O status de um service pai é então calculado com base nos status dos services filhos, de acordo com as regras de cálculo de status.

Se várias tags forem especificadas, a lógica *AND* é usada: um problema deve ter todas as tags especificadas na configuração do service para ser mapeado ao service.

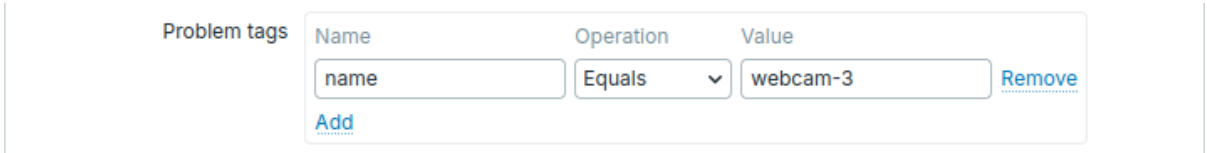


Note:
Um problema no Zabbix herda tags de toda a cadeia de templates, hosts, items, cenários web e triggers. Qualquer uma dessas tags pode ser usada para corresponder problemas a services.

Exemplo:

O problema *Web camera 3 is down* tem as tags `type:video-surveillance`, `floor:1` e `name:webcam-3` e status *Warning*

O service **Web camera 3** tem apenas a tag de problema especificada: `name:webcam-3`



O status do service mudará de *OK* para *Warning* quando esse problema for detectado.

Se o service **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 ficam visíveis somente quando a seção *Services* está no modo de ediçã

Para editar um serviço existente, clique no ícone de lápis ao lado do service.

Para clonar um serviço existente, clique no ícone de lápis para abrir sua configuração e, em seguida, clique no botão *Clone*. Quando um service é clonado, seus links de pai são preservados, enquanto os links de filho não são.

Para excluir um service, clique no ícone x ao lado dele. Quando você exclui um service pai, seus services filhos não serão excluídos e passarão para um nível acima na árvore de services (os filhos de 1º nível ficarão no mesmo nível do service pai excluído).

Dois botões abaixo da lista de services oferecem algumas opções de edição em massa:

- *Mass update* - atualizar em massa as propriedades do service
- *Delete* - excluir os services

Para usar essas opções, marque as caixas de seleção antes dos respectivos services e, em seguida, clique no botão desejado.

SLA

Visão geral Depois que os serviços forem criados, você poderá começar a monitorar se o desempenho deles está de acordo com o acordo de nível de serviço (SLA).

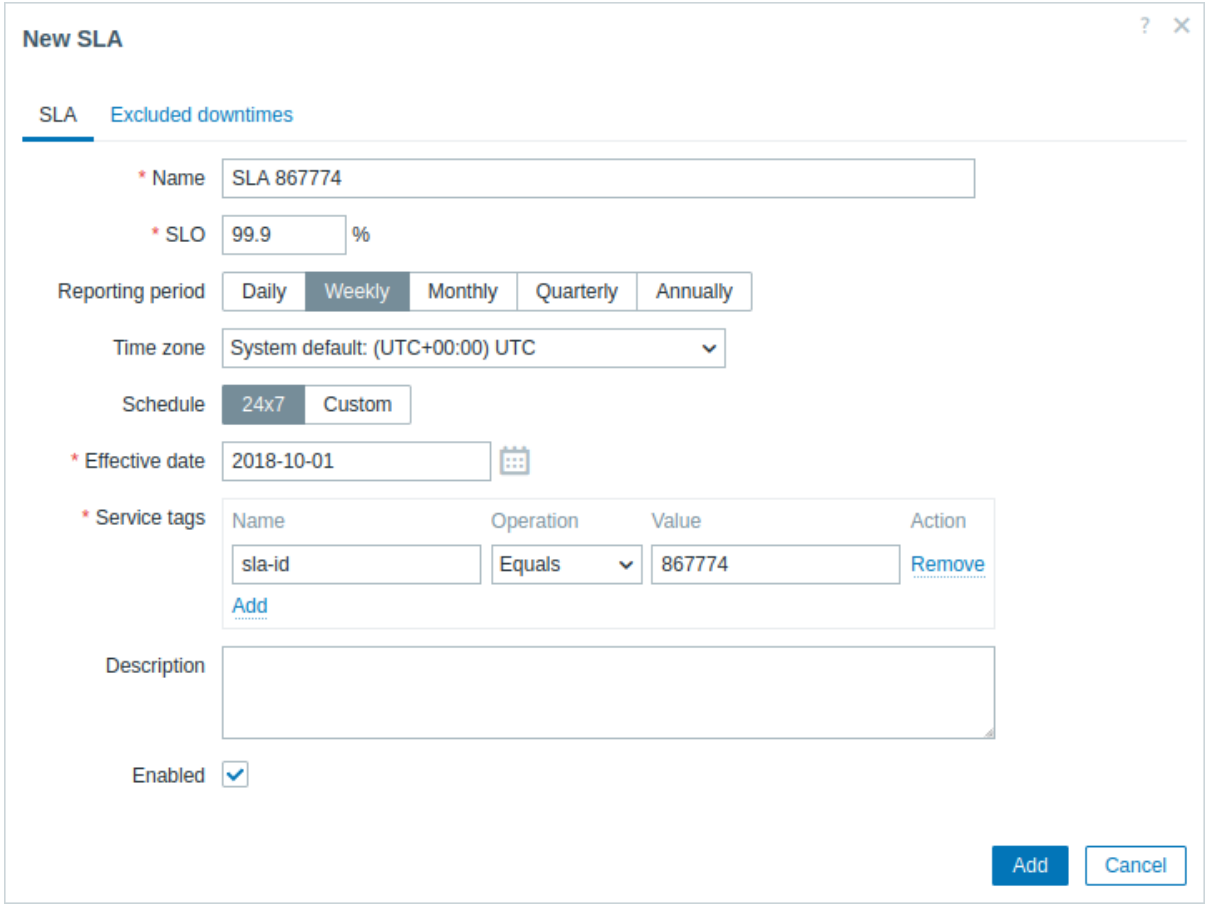
A seção de menu *Serviços* > *SLA* permite configurar SLAs para vários serviços. Um SLA no Zabbix define o objetivo de nível de serviço (SLO), a programação de disponibilidade esperada e os períodos de inatividade planejados.

Os SLAs e os serviços são associados por meio de 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. A verificação de que um serviço atende às metas do SLA é determinada pela comparação entre o SLO (disponibilidade esperada em %) e o SLI (disponibilidade real em %).

Configuração Para criar um novo SLA, clique no botão *Create SLA*.

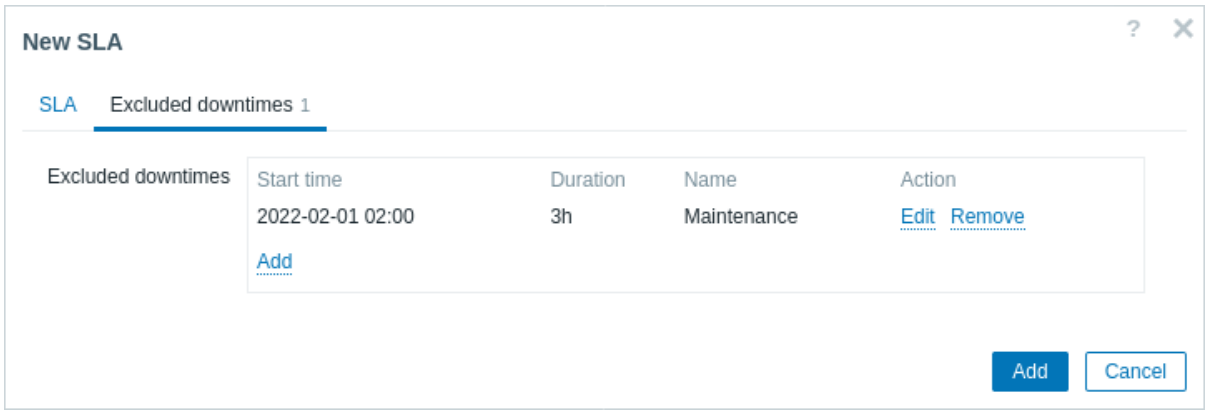
A aba *SLA* permite especificar os parâmetros gerais do SLA.



Parâmetro	Descrição
Name	Insira o nome do SLA.
SLO	Insira o objetivo de nível de serviço (SLO) como porcentagem.

Parâmetro	Descrição
Reporting period	Selecione o período de relatório do SLI usado para calcular os dados do SLI no relatório de SLA: **Daily** - cada dia, das 00:00:00 às 23:59:59. **Weekly** - cada semana, de domingo às 00:00:00 até sábado às 23:59:59. **Monthly** - cada mês, do primeiro dia às 00:00:00 até o último dia às 23:59:59. **Quarterly** - cada trimestre do calendário: Q1 (jan-mar), Q2 (abr-jun), Q3 (jul-set), Q4 (out-dez). **Annually** - cada ano do calendário, de 1º de janeiro às 00:00:00 até 31 de dezembro às 23:59:59.
Time zone	Selecione o fuso horário do SLA.
Schedule	Selecione a programação do SLA - 24x7 ou personalizada.
Effective date	Selecione a data de início do cálculo do SLA.
Service tags	Adicione tags de serviço para identificar os serviços aos quais este SLA deve ser aplicado. **Name** - nome da tag de serviço; deve corresponder exatamente e diferenciar maiúsculas de minúsculas. **Operation** - selecione *Equals* se o valor da tag precisar corresponder exatamente (diferenciando maiúsculas de minúsculas) ou *Contains* se parte do valor da tag precisar corresponder (sem diferenciar maiúsculas de minúsculas). **Value** - valor da tag de serviço a ser pesquisado de acordo com a operação selecionada. O SLA será aplicado a um serviço se pelo menos uma tag de serviço corresponder.
Description	Adicione uma descrição para o SLA.
Enabled	Marque a caixa de seleção para habilitar o cálculo do SLA.

A aba *Excluded downtimes* permite especificar períodos de indisponibilidade que serão excluídos do cálculo do SLA.



Clique em *Add* para configurar os períodos de indisponibilidade excluídos e, em seguida, insira o nome do período, a data de início e a duração.

Relatórios de SLA O desempenho de um service em comparação com um SLA é visível no relatório de SLA. Os relatórios de SLA podem ser visualizados:

- na seção *SLA*, clicando no hyperlink do relatório de SLA;
- na seção *Services*, clicando no nome do SLA na aba de informações;
- no widget *SLA report* do Dashboard.

Depois que um SLA é configurado, a aba *Info* na seção de services também exibirá algumas informações sobre o desempenho do service.

Exemplo de configuração

Visão geral Esta seção descreve uma configuração simples para monitorar o cluster de alta disponibilidade do Zabbix como um serviço.

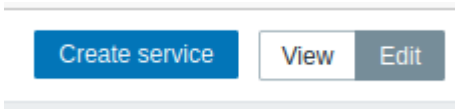
Pré-requisitos Antes de configurar o monitoramento de serviços, você precisa ter os hosts configurados:

- *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`

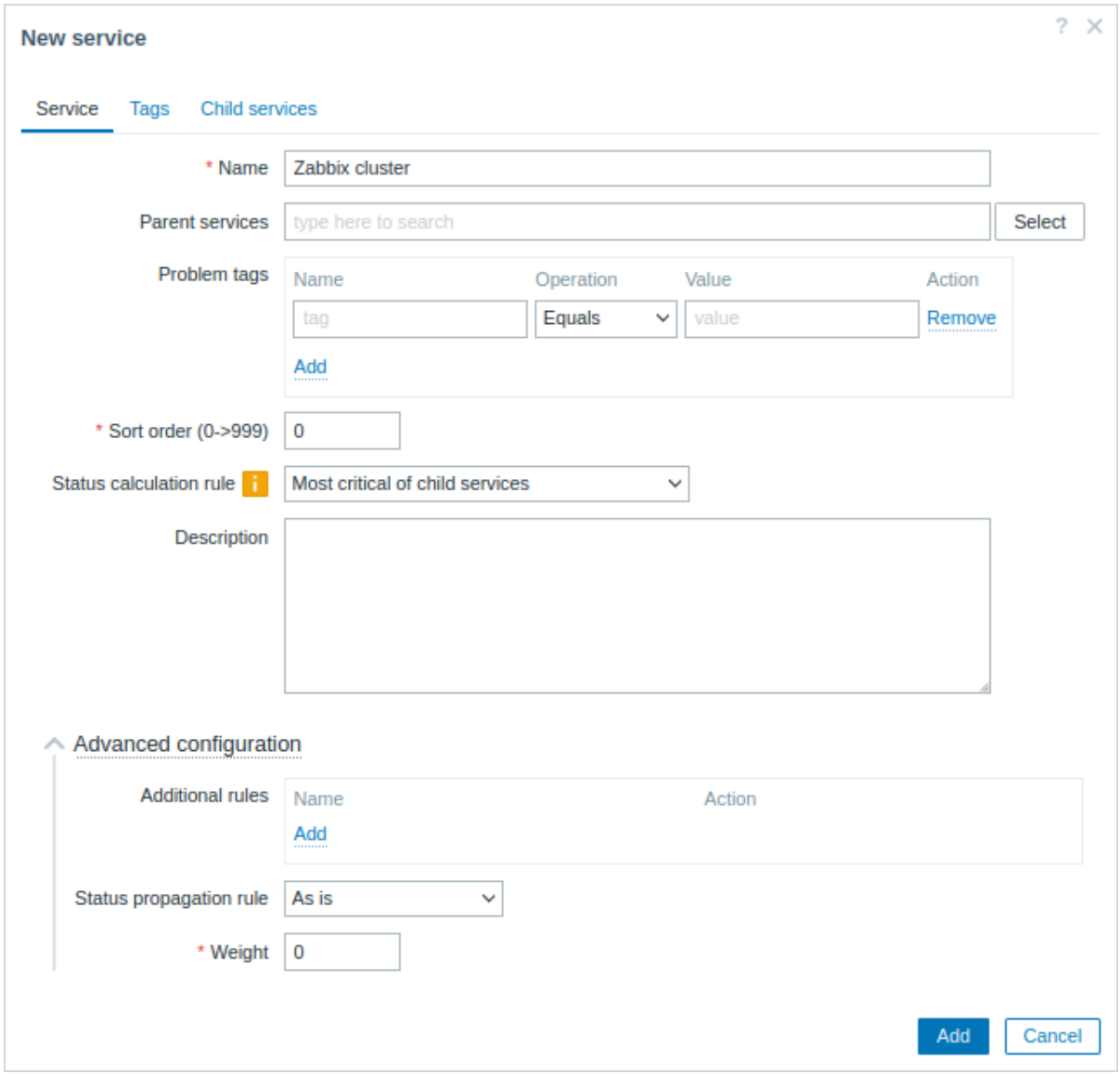
Árvore de serviços O próximo passo é montar a árvore de serviços. Neste exemplo, a infraestrutura é muito básica e consiste em três serviços: *Zabbix cluster* (pai) e dois serviços filhos *Zabbix server node 1* e *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

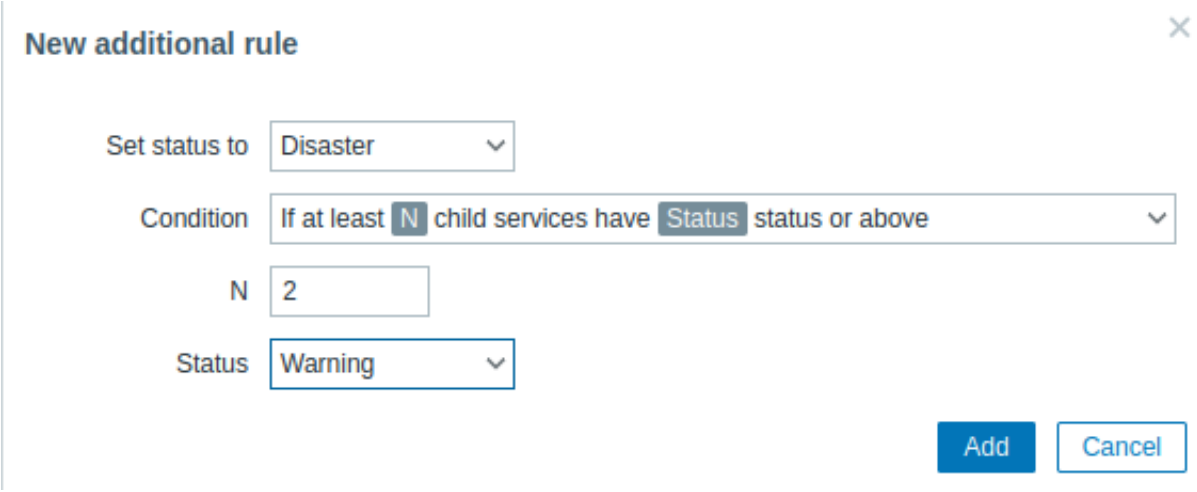
Na página Services, ative o modo *Edit* e clique em *Create service*:



Na janela de configuração do serviço, informe o nome *Zabbix cluster* e clique no cabeçalho *Advanced configuration* para exibir as opções de configuração avançada.



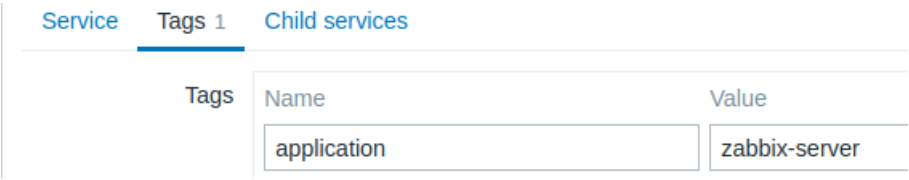
Configure uma regra adicional:



O Zabbix cluster terá dois serviços filhos - um para cada um dos nós HA. Se ambos os nós HA tiverem problemas com status de pelo menos *Warning*, o status do serviço pai deverá ser definido como *Disaster*. Para isso, a regra adicional deve ser configurada da seguinte forma:

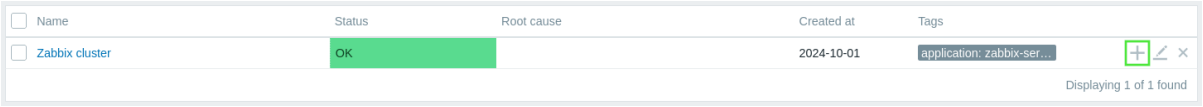
- Definir status como: Disaster
- Condição: Se pelo menos `N` serviços filhos tiverem status `Status` ou superior
- N: 2
- Status: Warning

Acesse a aba *Tags* e adicione a tag `application:zabbix-server`. Essa tag será usada posteriormente para ações de serviço e relatórios de SLA.



Salve o novo serviço.

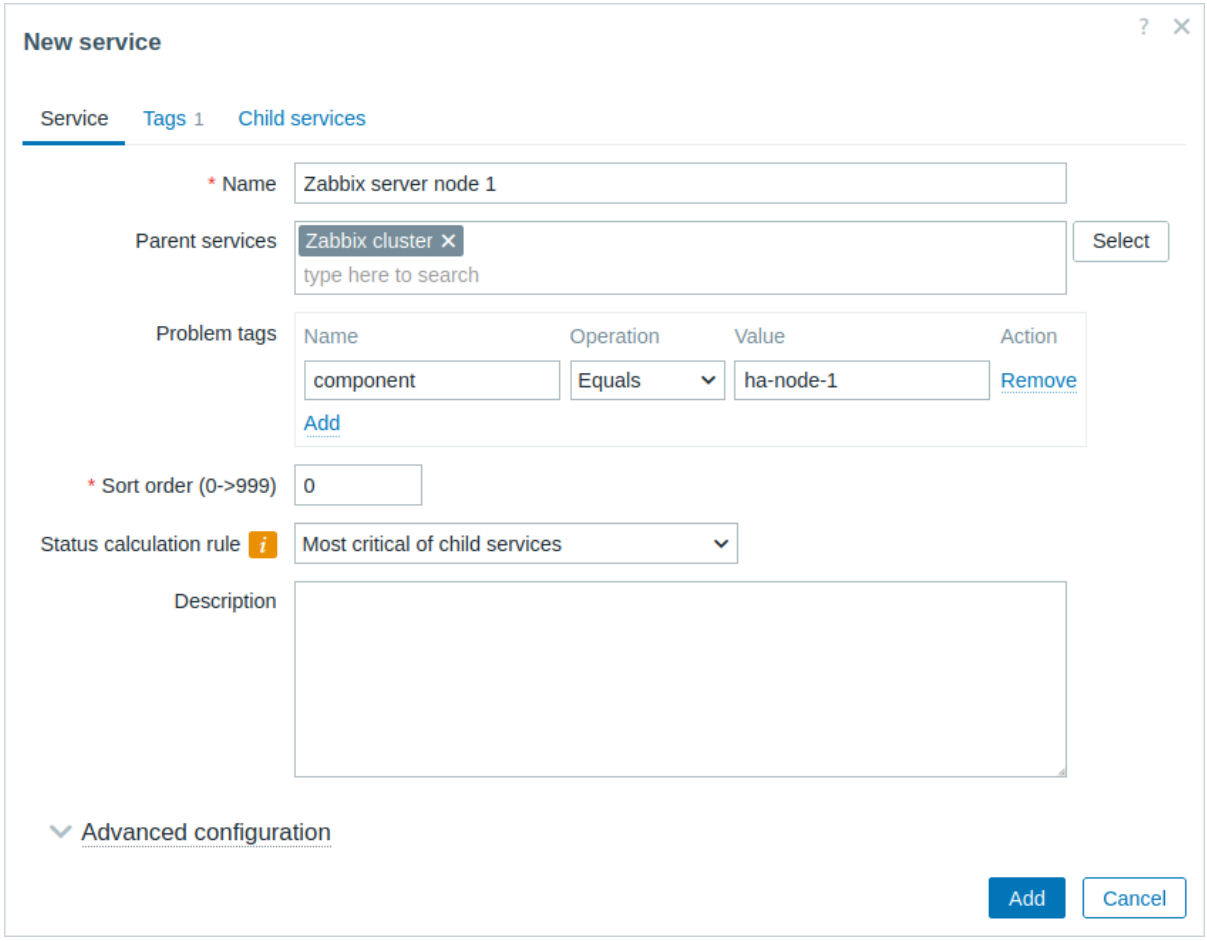
Para adicionar um serviço filho, clique no ícone de mais ao lado do serviço de cluster Zabbix (o ícone fica visível apenas no modo *Edit*).



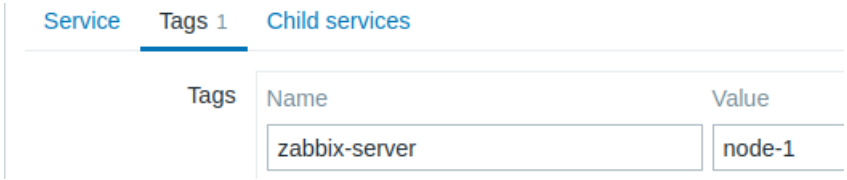
Na janela de configuração do serviço, informe o nome *Zabbix server node 1*. Observe que o parâmetro Parent services já está preenchido com *Zabbix cluster*.

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 Problem tags, informe:

- Name: component
- Operation: Equals
- Value: ha-node-1



Mude para a aba *Tags* e adicione uma tag de serviço: `zabbix-server:node-1`. Essa 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`.

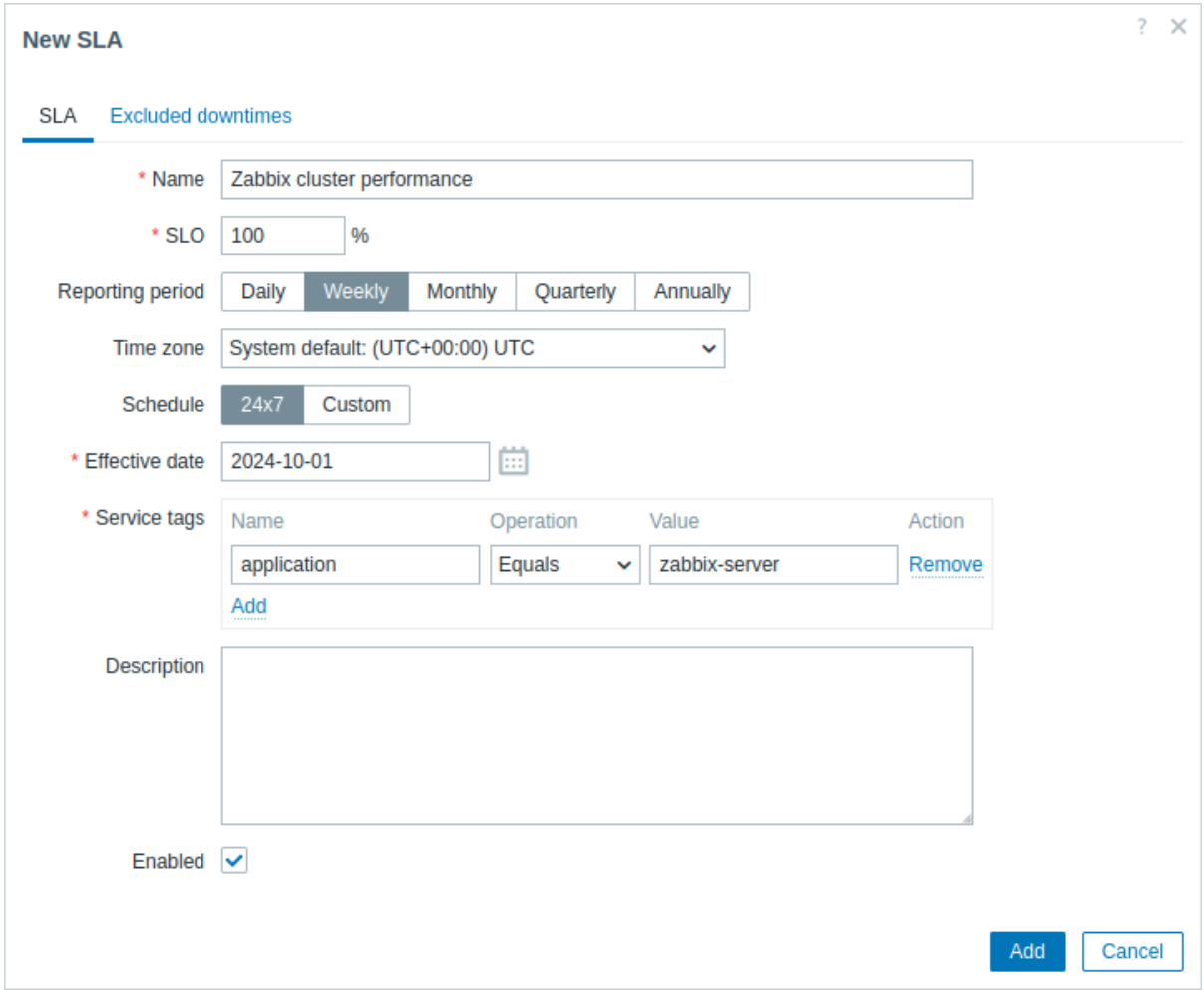
Salve o novo serviço.

SLA Neste exemplo, o desempenho esperado do cluster Zabbix é de 100%, excluindo um período de manutenção semestral de uma hora.

Primeiro, você precisa adicionar um novo SLA:

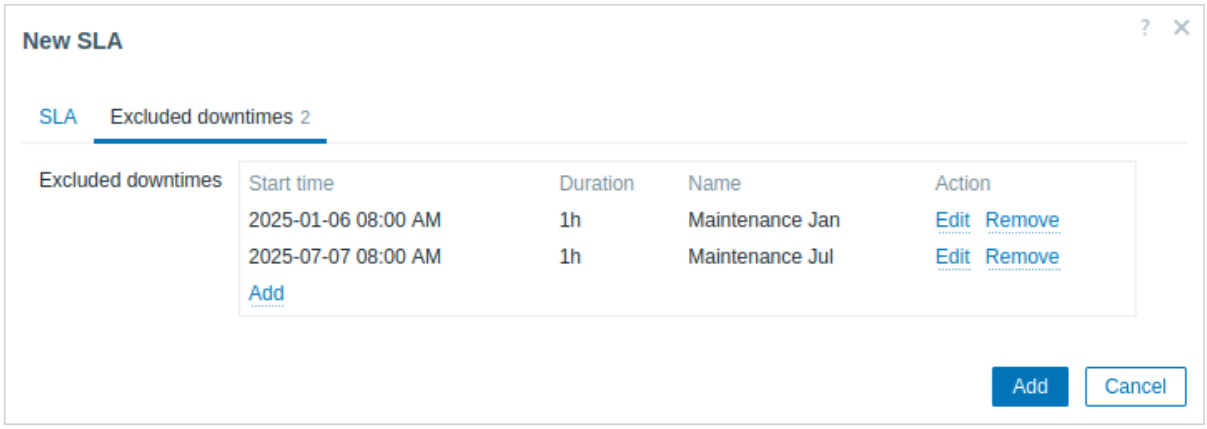
1. Acesse *Services* > *SLA*.
2. Clique em *Create SLA*.
3. Insira o nome *Zabbix cluster performance* e defina o SLO como 100%. O serviço Zabbix cluster tem a tag de serviço `application:zabbix-server`.
4. Para usar este SLA na medição do desempenho do cluster Zabbix, no parâmetro *Service tags*, especifique:

- Name: application
- Operation: Equals
- Value: zabbix-server



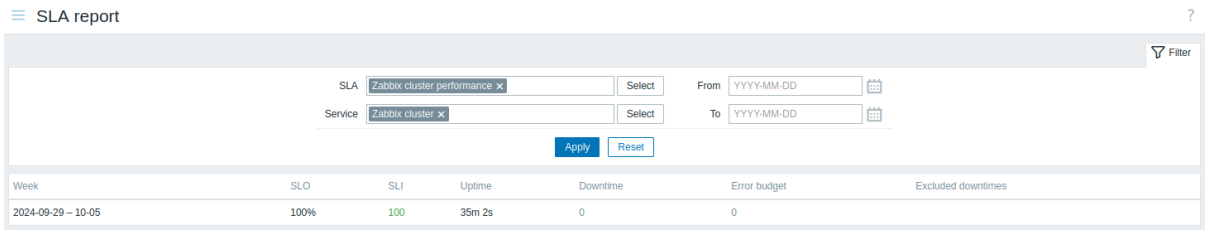
Em uma configuração real, você também pode atualizar o período de relatório desejado, o fuso horário e a data de início, ou alterar a programação de 24/7 para uma personalizada. Para este exemplo, as configurações padrão são suficientes.

5. Alterne para a aba *Excluded downtimes* e adicione períodos de inatividade correspondentes às manutenções programadas para excluí-los do cálculo do SLA.
6. Na seção *Excluded downtimes*, clique no link *Add*, insira o nome do período de inatividade, o horário de início planejado e a duração.

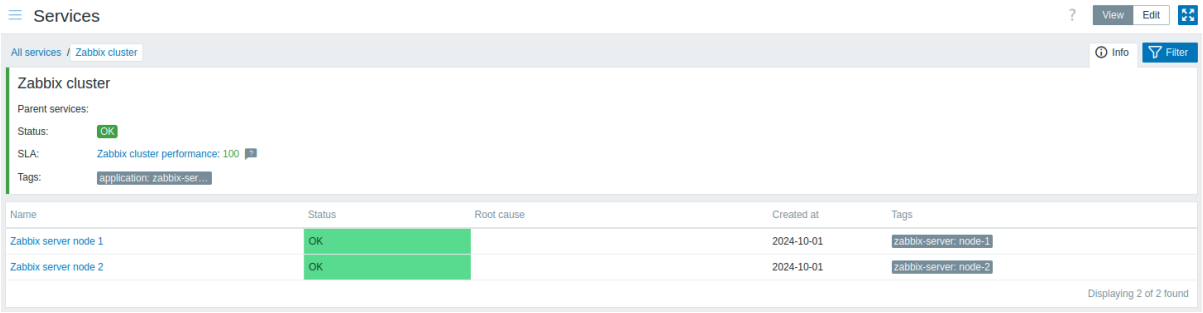


7. Clique em *Add* para salvar o novo SLA.

Alterne 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 *Services*.



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 mais antigas do Zabbix, os serviços tinham dois tipos de dependências: soft e hard. Após uma atualização, todas as dependências passarão a ser iguais.

Se um serviço "Child service" tiver sido vinculado anteriormente a "Parent service 1" por meio de uma dependência hard e, adicionalmente, a "Parent service 2" por meio de uma dependência soft, após a atualização o "Child service" terá dois serviços pai: "Parent service 1" e "Parent service 2".

O mapeamento entre problemas e serviços baseado em trigger foi substituído por mapeamento baseado em tags. No Zabbix 6.0 e versões mais recentes, o formulário de configuração de serviço tem um novo parâmetro *Problem tags*, que permite especificar um ou mais pares de nome e valor de tag para correspondência de problemas. Os triggers que foram vinculados a um serviço receberão uma nova tag `ServiceLink : <trigger ID>:<trigger name>` (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 "Status calculation algorithm" será aprimorado usando as seguintes regras:

- Não calcular → Definir o status como OK
- Problem, se pelo menos um filho tiver um problema → O mais crítico entre os serviços filhos
- Problem, se todos os filhos tiverem problemas → O mais crítico se todos os filhos tiverem problemas

SLAs Anteriormente, os alvos de SLA precisavam ser definidos separadamente para cada service. Desde o Zabbix 6.0, o SLA passou a ser uma entidade separada, que contém informações sobre a programação do service, o objetivo de nível de service esperado (SLO) e os períodos de indisponibilidade a serem excluídos do cálculo. Depois de configurado, um SLA pode ser atribuído a vários services por meio de service tags.

Durante uma atualização:

- SLAs idênticos definidos para cada service serão agrupados e um SLA por grupo será criado.
- Cada service afetado receberá uma tag especial `SLA:<ID>` e a mesma tag será especificada no parâmetro *Service tags* do SLA correspondente.
- O tempo de criação do service, uma nova métrica nos relatórios de SLA, será definido como 01/01/2000 00:00 para os services existentes.

Criptografia e segurança

Encryption O Zabbix oferece suporte a comunicações criptografadas entre os componentes do Zabbix usando o protocolo Transport Layer Security (TLS) v.1.2 e 1.3 (dependendo da biblioteca criptográfica). Há suporte para criptografia baseada em certificado e baseada em chave pré-compartilhada.

A criptografia pode ser configurada para conexões:

- Entre o Zabbix server, Zabbix proxy, Zabbix agent, Zabbix web service, zabbix_sender e os utilitários zabbix_get
- Com o banco de dados do 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 outros ainda podem continuar com comunicações sem criptografia (como antes).
- O server (proxy) pode usar diferentes configurações de criptografia para diferentes hosts.

Os programas daemon do Zabbix usam uma única porta de escuta para conexões de entrada criptografadas e não criptografadas. Adicionar criptografia não requer a abertura de novas portas nos 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 integrada 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; não há cache de sessão nem tickets implementados.
- Adicionar criptografia aumenta o tempo para verificações de item e ações, dependendo da latência da rede:
  - Por exemplo, se o atraso dos pacotes for de 100 ms, abrir uma conexão TCP e enviar uma solicitação sem criptografia leva cerca de 200 ms. Com criptografia, cerca de 1000 ms são adicionados para estabelecer a conexão TLS.
  - Pode ser necessário aumentar os timeouts; caso contrário, alguns items e ações que executam scripts remotos em agents podem funcionar com conexões sem criptografia, mas falhar por timeout com criptografia.
- A criptografia não é suportada por descoberta de rede. As verificações do Zabbix agent realizadas pela descoberta de rede serão sem criptografia e, se o Zabbix agent estiver configurado para rejeitar conexões sem criptografia, essas verificações não terão sucesso.

Compilando o Zabbix com suporte à criptografia Para oferecer suporte à criptografia, o Zabbix deve ser compilado e vinculado a uma das bibliotecas criptográficas compatíveis:

- 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 - 3.5.x
- LibreSSL - testado com as versões 2.7.4, 2.8.2:
  - LibreSSL 2.6.x não é compatível
  - LibreSSL é compatível como substituto do OpenSSL; as novas funções da API específicas do LibreSSL `tls_*()` não são usadas. Os componentes do Zabbix compilados com LibreSSL não poderão usar PSK; somente certificados poderão ser usados.

Note:
Para saber mais sobre como configurar SSL para o frontend do Zabbix, consulte estas práticas recomendadas.

A biblioteca é selecionada especificando a opção correspondente no script "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (também usado para LibreSSL)

Por exemplo, para configurar os fontes do server e do agent com *OpenSSL*, você pode usar algo semelhante a:

```
./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 criptográficas (por exemplo, um server com *OpenSSL* e 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 PSK. As bibliotecas *GnuTLS* e *OpenSSL 1.1.0* oferecem suporte a conjuntos de cifras PSK com Perfect Forward Secrecy. Versões mais antigas da biblioteca *OpenSSL* (1.0.1, 1.0.2c) também oferecem suporte a PSK, mas os conjuntos de cifras PSK disponíveis não fornecem Perfect Forward Secrecy.

Gerenciamento da criptografia de conexão As conexões no Zabbix podem usar:

- nenhuma criptografia (padrão)
- criptografia baseada em certificado RSA
- criptografia baseada em PSK

Há 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 conexão 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 do Zabbix (no modo ativo, especifica apenas as conexões com o server) e do agent do Zabbix (para verificações ativas). No frontend do Zabbix, o equivalente a `TLSConnect` é o campo *Conexões com o host* em *Coleta de dados* > *Hosts* > <algum host> > aba *Criptografia* e o campo *Conexões com 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 do Zabbix (no modo passivo, especifica apenas as conexões do server) e do agent do Zabbix (para verificações passivas). No frontend do Zabbix, o equivalente a 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. No entanto, pode ser necessário alterar o tipo de criptografia, por exemplo, de sem criptografia para baseada em certificado, com o mínimo de indisponibilidade e possibilidade de reversão. Para isso:

- Defina `TLSAccept=unencrypted,cert` no arquivo de configuração do agent e reinicie o Zabbix agent.
- Teste a conexão com o `zabbix_get` para o agent usando um certificado. Se funcionar, você poderá reconfigurar a criptografia desse agent no frontend do Zabbix, em *Coleta de dados* > *Hosts* > <algum host> > aba *Criptografia*, definindo *Conexões com o host* como *Certificado*.
- Quando o cache de configuração do server for atualizado (e a configuração do proxy também for atualizada se o host for monitorado por um proxy), as conexões com esse agent serão criptografadas.
- Se tudo funcionar conforme o esperado, defina `TLSAccept=cert` no arquivo de configuração do agent e reinicie o Zabbix agent. Agora o agent aceitará somente conexões criptografadas baseadas em certificado. Conexões sem criptografia e baseadas em PSK serão rejeitadas.

De forma semelhante, isso funciona no server e no proxy. Se, na configuração do host no frontend do Zabbix, *Conexões do host* estiver definido como "Certificado", somente conexões criptografadas baseadas em certificado serão aceitas do agent (verificações ativas) e do zabbix_sender (itens trapper).

É muito provável que você configure as conexões de entrada e saída para usar o mesmo tipo de criptografia ou nenhuma criptografia. No entanto, tecnicamente é possível configurá-las de forma assimétrica, por exemplo, usando criptografia baseada em certificado para conexões de entrada e baseada em PSK para conexões de saída.

A configuração de criptografia de cada host é exibida no frontend do Zabbix, em *Coleta de dados* > *Hosts*, na coluna *Criptografia do agent*. Por exemplo:

Exemplo	Conexões com 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	Criptografadas, baseadas em certificado	Criptografadas, baseadas em certificado	Sem criptografia e criptografadas baseadas em PSK
PSK NONE PSK CERT	Criptografadas, baseadas em PSK	Criptografadas, baseadas em PSK	Sem criptografia e criptografadas baseadas em certificado
PSK NONE PSK CERT	Criptografadas, baseadas em PSK	Sem criptografia e criptografadas baseadas em PSK	Criptografadas baseadas em certificado
CERT NONE PSK CERT	Criptografadas, baseadas em certificado	Sem criptografia, criptografadas baseadas em PSK ou em certificado	-

Attention:
As conexões não são criptografadas por padrão. A criptografia deve ser configurada individualmente para cada host e proxy.

zabbix_get e zabbix_sender com criptografia Veja as páginas de manual zabbix_get e zabbix_sender para usá-los com criptografia.

Conjuntos de cifras Os conjuntos de cifras são configurados internamente por padrão durante a inicialização do Zabbix.

Também há suporte a conjuntos de cifras configurados pelo usuário para GnuTLS e OpenSSL. Os usuários podem configurar conjuntos de cifras de acordo com suas políticas de segurança. O uso desse recurso é opcional (os conjuntos de cifras padrão integrados continuam funcionando).

Para bibliotecas criptográficas compiladas com as configurações padrão, as regras integradas do Zabbix normalmente resultam nos seguintes conjuntos de cifras (em ordem de maior para menor prioridade):

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Ciphersuites configuradas pelo usuário Os critérios de seleção de ciphersuite integrados podem ser substituídos por ciphersuites configuradas pelo usuário.

Attention:
Ciphersuites configuradas pelo usuário é um recurso destinado a usuários avançados que entendem ciphersuites TLS, sua segurança e as consequências de erros, e que se sentem à vontade com a solução de problemas de TLS.

Os critérios de seleção de ciphersuite integrados podem ser substituídos usando os seguintes parâmetros:

Escopo da substituição	Parâmetro	Valor	Descrição
Seleção de ciphersuite para certificados	TLSCipherCert13	Strings de cipher válidas do OpenSSL 1.1.1 cipher strings para o protocolo TLS 1.3 (seus valores são passados para a função `SSL_CTX_set_ciphersuites()` do OpenSSL).	Critérios de seleção de ciphersuite baseados em certificado para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherCert	Strings de cipher válidas do OpenSSL cipher strings para TLS 1.2 ou strings de prioridade válidas do GnuTLS priority strings. Seus valores são passados, respectivamente, para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critérios de seleção de ciphersuite baseados em certificado para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Seleção de ciphersuite para PSK	TLSCipherPSK13	Strings de cipher válidas do OpenSSL 1.1.1 cipher strings para o protocolo TLS 1.3 (seus valores são passados para a função `SSL_CTX_set_ciphersuites()` do OpenSSL).	Critérios de seleção de ciphersuite baseados em PSK para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherPSK	Strings de cipher válidas do OpenSSL cipher strings para TLS 1.2 ou strings de prioridade válidas do GnuTLS priority strings. Seus valores são passados, respectivamente, para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critérios de seleção de ciphersuite baseados em PSK para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Lista combinada de ciphersuite para certificado e PSK	TLSCipherAll13	Strings de cipher válidas do OpenSSL 1.1.1 cipher strings para o protocolo TLS 1.3 (seus valores são passados para a função `SSL_CTX_set_ciphersuites()` do OpenSSL).	Critérios de seleção de ciphersuite para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherAll	Strings de cipher válidas do OpenSSL cipher strings para TLS 1.2 ou strings de prioridade válidas do GnuTLS priority strings. Seus valores são passados, respectivamente, para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critérios de seleção de ciphersuite para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Para substituir a seleção de ciphersuite nos 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 ser vazio.

Se a configuração de um valor TLSCipher* na biblioteca criptográfica falhar, então o server, proxy ou agent não iniciará e um erro será registrado.

É importante entender quando cada parâmetro é aplicável.

Conexões de saída

O caso mais simples é o de 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 um 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 o **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 observado nas duas tabelas acima:

- TLSCipherAll e TLSCipherAll13 podem ser especificados somente se for usada uma lista combinada de cifras baseadas em certificado **e** em PSK. Há dois casos em que isso ocorre: server (proxy) com um certificado configurado (as cifras PSK são sempre configuradas no server, no proxy se a biblioteca criptográfica oferecer suporte a PSK), agent configurado para aceitar conexões de entrada baseadas em certificado e em PSK
- em outros casos, TLSCipherCert* e/ou TLSCipherPSK* são suficientes

As tabelas a seguir mostram os valores padrão internos de `TLSCipher*`. Eles podem ser um bom ponto de partida para seus próprios valores personalizados.

Parâmetro	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Parâmetro	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] Os valores padrão são diferentes para versões mais antigas do OpenSSL (1.0.1, 1.0.2, 1.1.0), para LibreSSL e se o OpenSSL for compilado sem suporte a PSK.

Exemplos de cifras configuradas pelo usuário

Veja abaixo os exemplos a seguir de cifras configuradas pelo usuário:

- Testando strings de cifra e permitindo apenas cifras PFS
- Alternando de AES128 para AES256

Testando strings de cifras e permitindo apenas conjuntos de cifras PFS

Para ver quais conjuntos de cifras foram selecionados, defina `DebugLevel=4` no arquivo de configuração ou use a opção `-vv` do zabbix_sender.

Pode ser necessário fazer alguns testes com os parâmetros `TLSCipher*` antes de obter os conjuntos de cifras desejados. É inconveniente reiniciar o Zabbix server, o proxy ou o agent várias vezes apenas para ajustar os parâmetros `TLSCipher*`. Opções mais convenientes são usar o zabbix_sender ou o comando `openssl`. Vamos mostrar ambas.

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 válidos da CA e do agent, além de uma PSK, para este exemplo. Ajuste os caminhos e nomes dos arquivos de certificado e PSK de acordo com o seu ambiente.

Se você não estiver usando certificados, mas apenas PSK, poderá criar um arquivo de teste mais simples:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Os conjuntos de cifras selecionados podem ser vistos 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 os conjuntos de cifras selecionados por padrão. Esses valores padrão são escolhidos para garantir a interoperabilidade com Zabbix agents em execução em sistemas com versões mais antigas do OpenSSL (a partir da 1.0.1).

Em sistemas mais recentes, você pode optar por aumentar a segurança, permitindo apenas alguns conjuntos de cifras, por exemplo, somente conjuntos de cifras com PFS (Perfect Forward Secrecy). Vamos configurar os parâmetros `TLSCipher*` para permitir apenas conjuntos de cifras compatíveis com PFS.

> **Attention:**
> O resultado não será interoperável com sistemas que usam OpenSSL 1.0.1 e 1.0.2 se a PSK for usada. A criptografia baseada em certificados 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 de "certificate ciphersuites" e "PSK ciphersuites" foram alteradas: elas estão mais curtas do que antes e contêm apenas conjuntos de cifras TLS 1.3 e conjuntos de cifras TLS 1.2 ECDHE-*, conforme esperado.

2. `TLSCipherAll` e `TLSCipherAll13` não podem ser testados com o zabbix_sender; eles não afetam o valor de "certificate and PSK ciphersuites" mostrado no exemplo acima. Para ajustar `TLSCipherAll` e `TLSCipherAll13`, é necessário fazer testes com o agent, o proxy ou o server.

Portanto, para permitir apenas conjuntos de cifras PFS, talvez seja necessário adicionar até três parâmetros

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

a zabbix_agentd.conf, zabbix_proxy.conf e zabbix_server.conf se cada um deles tiver um certificado configurado e o agent também tiver uma PSK.

Se o seu ambiente Zabbix usar apenas criptografia baseada em PSK e nenhum certificado, será necessário apenas um parâmetro:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Agora que você entende como isso funciona, pode testar a seleção de conjuntos de cifras mesmo fora do Zabbix, usando o comando openssl. Vamos testar os três valores de parâmetros 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 usar openssl ciphers com a opção -V para obter 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 do 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 integrado para os 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 conjuntos de cifras relacionados a certificados sejam usados, os parâmetros TLSCipherPSK* também são definidos para evitar seus valores padrão, que incluem cifras menos seguras para uma maior interoperabilidade. Os conjuntos de cifras PSK não podem ser completamente desativados 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
```

Solução de problemas de criptografia Recomendações gerais:

- Comece entendendo qual componente atua como cliente TLS e qual atua como servidor TLS no caso do problema.
 Zabbix server, proxies e agents, dependendo da interação entre eles, podem atuar tanto como servidores TLS quanto como clientes TLS.
 Por exemplo, o Zabbix server ao se conectar ao agent para uma verificação passiva atua como cliente TLS. O agent está no papel de servidor TLS.
 O Zabbix agent, ao solicitar uma lista de verificações ativas ao proxy, atua como cliente TLS. O proxy está no papel de servidor TLS.
 As utilidades `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 ao se conectar ao agent pode encerrar a conexão imediatamente se o certificado do agent for inválido. E vice-versa - o Zabbix agent ao aceitar uma conexão do server pode encerrar 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 frequentemente relata um erro mais geral (por exemplo, ”Connection closed by peer”, ”connection was non-properly terminated”).
- Às vezes, uma criptografia configurada incorretamente resulta em mensagens de erro confusas que não apontam de forma alguma 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 criptográficos (OpenSSL, GnuTLS) frequentemente produzem mensagens de erro diferentes nas mesmas situações de problema.
 Às vezes, as mensagens de erro dependem até mesmo da combinação específica de toolkits criptográficos em ambos os lados.

Solução de problemas de tipo de conexão ou problemas de permissão Server está configurado para se conectar com PSK ao agent, mas o agent aceita apenas conexões não criptografadas

No log do server ou proxy (com *GnuTLS* 3.3.16)

```
Falha ao obter valor do agent: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 A conexão TLS foi encerrada de forma incorreta.
```

No log do server ou proxy (com *OpenSSL* 1.0.2c)

```
Falha ao obter valor do agent: conexão TCP bem-sucedida, não foi possível estabelecer TLS com [[127.0.0.1]
    Conexão fechada pelo peer. Verifique os tipos de conexão permitidos e os direitos de acesso
```

Um lado se 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 para o Zabbix server/proxy compilado sem TLS No log do lado da conexão:

Linux:

```
...Em zbx_tls_init_child()
...Biblioteca OpenSSL (versão OpenSSL 1.1.1  11 Sep 2018) inicializada
...
...Em zbx_tls_connect(): psk_identity:"PSK test sender"
...Fim de zbx_tls_connect():FALHA erro:'conexão fechada pelo peer'
...erro ao enviar valor: TCP bem-sucedido, não foi possível estabelecer TLS com [[localhost]:10051]: conex
```

Windows:

```
...Biblioteca OpenSSL (versão OpenSSL 1.1.1a  20 Nov 2018) inicializada
...
...Em zbx_tls_connect(): psk_identity:"PSK test sender"
```

```
...zbx_psk_client_cb() solicitou a identidade PSK "PSK test sender"
...Fim de zbx_tls_connect():FALHA erro:'erro de E/S em SSL_connect(): [0x00000000] A operação foi concluíd
...erro ao enviar valor: TCP bem-sucedido, não foi possível estabelecer TLS com [[192.168.1.2]:10051]: err
```

No log do lado que aceita a conexão:

```
...falha ao aceitar uma conexão de entrada: de 127.0.0.1: o suporte a TLS não foi compilado
```

Um lado se conecta com PSK, mas o outro lado usa LibreSSL ou foi compilado sem suporte a criptografia O LibreSSL não oferece suporte a PSK.

No log do lado que está se conectando:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

No log do lado que está aceitando:

```
...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 se conecta com PSK, mas o outro lado usa OpenSSL com suporte a PSK desativado No log do lado que está se 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
```

Criptografia 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 com base em um certificado de 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 as extensões do seu 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.

Parameter	Mandatory	Description
TLSCAFile	yes	Caminho completo de um arquivo que contém 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	no	Caminho completo de um arquivo que contém Listas de Revogação de Certificados (CRL).
TLSCertFile	yes	Caminho completo de um arquivo que contém 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, por fim, pelos certificados da(s) CA(s) de nível superior.

Parameter	Mandatory	Description
TLSKeyFile	yes	Caminho completo de um arquivo que contém a chave privada. Certifique-se de que este arquivo seja legível apenas pelo usuário do Zabbix, definindo as permissões de acesso apropriadas.
TLSServerCertIssuer	no	Emissor permitido do certificado do server.
TLSServerCertSubject	no	Assunto permitido do certificado do server.

Exemplos de configuração Após configurar os certificados necessários, configure os componentes do Zabbix para usar criptografia baseada em certificado.

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 raiz da CA de nível superior e autoassinados. Por exemplo, se forem necessários certificados de duas CAs raiz independentes, coloque-os em um arquivo em `/home/zabbix/zabbix_ca_file.crt`:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
```

```
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Coloque o certificado/cadeia de certificados do Zabbix server em um arquivo, por exemplo, em `/home/zabbix/zabbix_server.crt`. O primeiro certificado é o certificado do Zabbix server, seguido pelo certificado da CA intermediária:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
Use somente os atributos mencionados acima para os certificados do cliente e do server, a fim de evitar impactos no processo de verificação do certificado. Por exemplo, o OpenSSL pode falhar ao estabelecer uma conexão criptografada se forem usadas as extensões *X509v3 Subject Alternative Name* ou *Netscape Cert Type*. Para obter mais informações, consulte Limitações do 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 de TLS no arquivo de configuração do server do Zabbix:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix proxy

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/a cadeia de certificados do Zabbix proxy e a chave privada, conforme descrito na seção Zabbix server. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do proxy do Zabbix conforme necessário.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do proxy do 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 obter mais informações, consulte Restringindo o emissor e o assunto do certificado permitidos.

Os parâmetros TLS no arquivo de configuração final do proxy podem ser semelhantes aos seguintes:

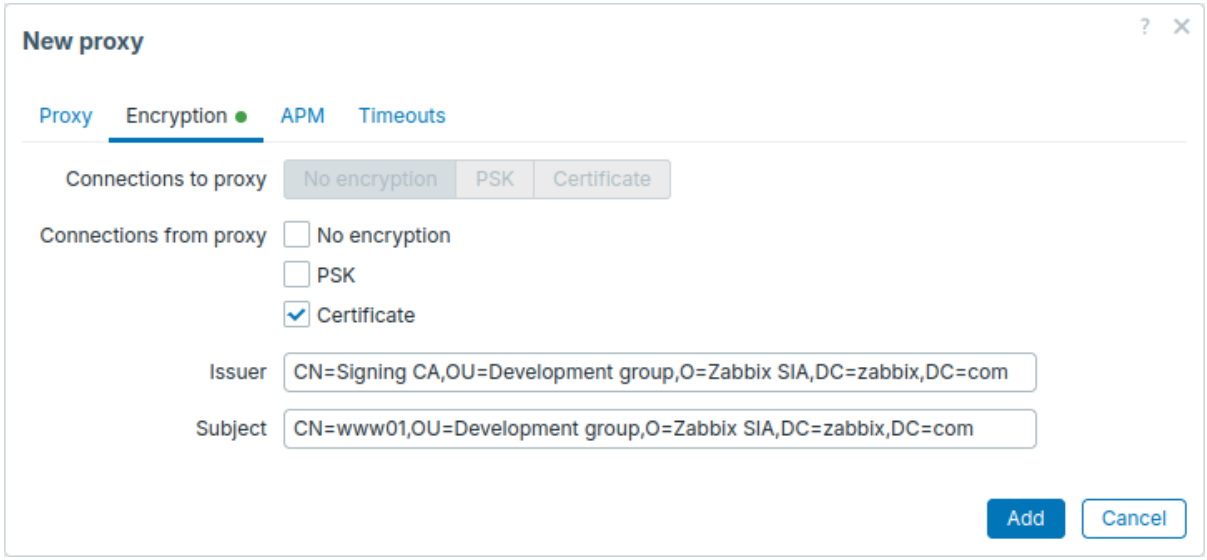
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configure a criptografia para este proxy no frontend do Zabbix:

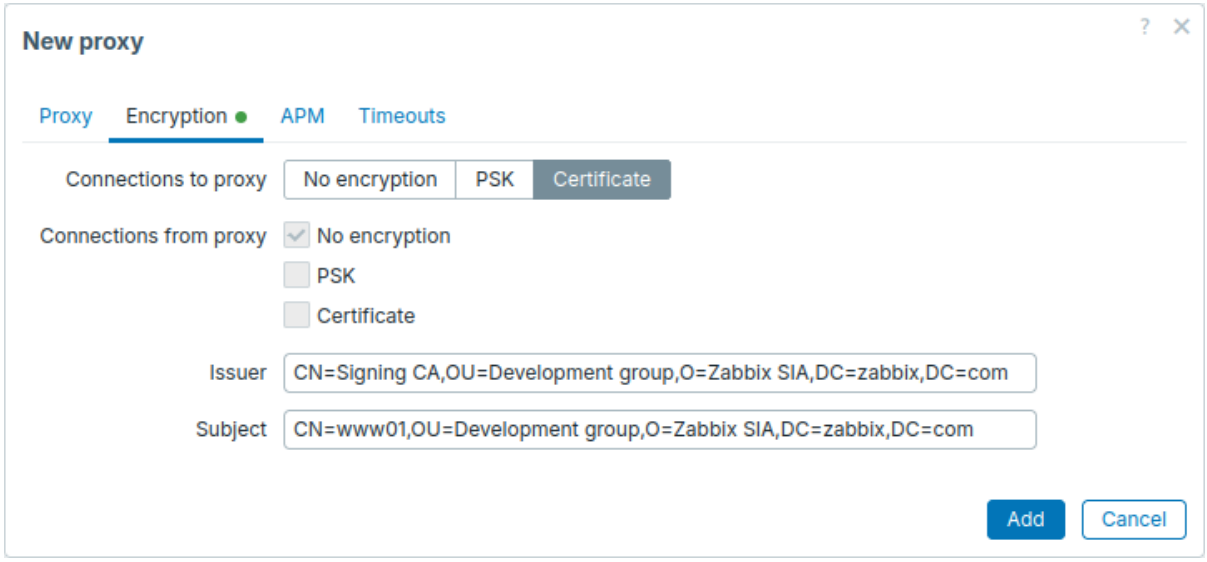
- Acesse: *Administration* > *Proxies*.
- Selecione o proxy e clique na aba *Encryption*.

Nos exemplos abaixo, os campos *Issuer* e *Subject* estão preenchidos. Para obter mais informações sobre por que e como usar esses campos, consulte Restringindo o emissor e o assunto do certificado permitidos.

Para proxy ativo:



Para proxy passivo:



Zabbix agent

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/a cadeia de certificados do Zabbix agent e a chave privada, conforme descrito na seção Zabbix server. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do agent do Zabbix conforme necessário.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do agent do Zabbix:

- Para agent ativo: `TLSConnect=cert`
- Para agent passivo: `TLSAccept=cert`

Note:
Para aumentar a segurança do agent, você pode definir os parâmetros `TLSServerCertIssuer` e `TLSServerCertSubject`. Para obter mais informações, consulte Restringindo o emissor e o assunto do certificado permitidos.

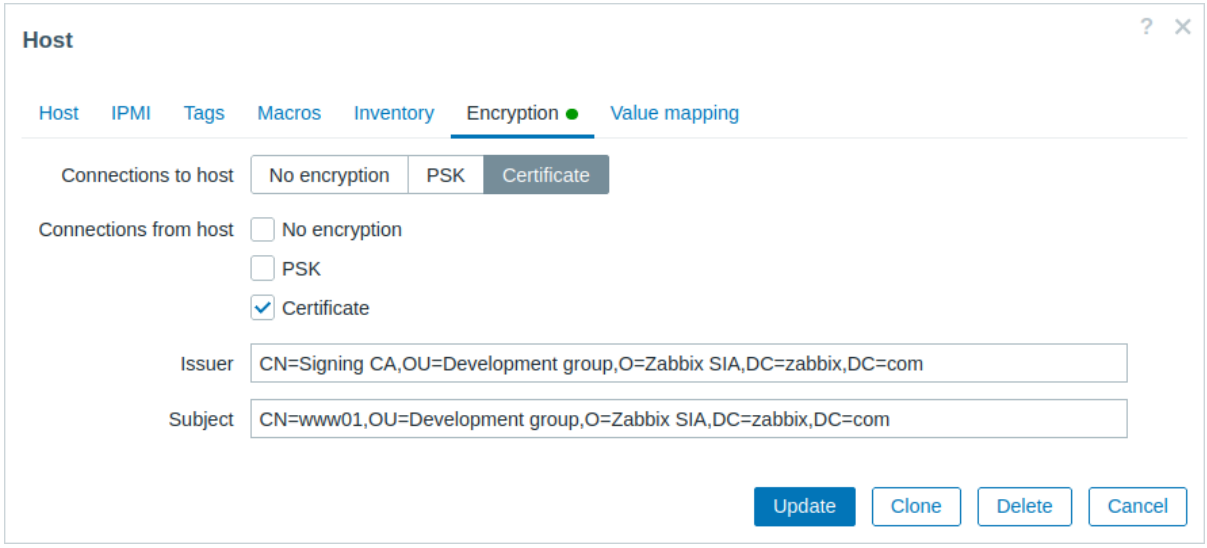
Os parâmetros TLS no arquivo de configuração final do agent podem ser semelhantes aos seguintes. Observe que o exemplo pressupõe que o host seja monitorado por um proxy; por isso, ele é especificado como o Subject:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. Configure a criptografia no frontend do Zabbix para o host monitorado por este agent.

- Acesse: *Coleta de dados* > *Hosts*.
- Selecione o host e clique na aba *Criptografia*.

No exemplo abaixo, os campos *Issuer* e *Subject* estão preenchidos. Para obter mais informações sobre por que e como usar esses campos, consulte Restringindo o emissor e o assunto do certificado permitidos.



Zabbix web service

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/a 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 web service do Zabbix conforme necessário.

2. Edite um parâmetro TLS adicional no arquivo de configuração do web service do Zabbix: `TLSAccept=cert`

Os parâmetros TLS no arquivo de configuração final do web service podem ser semelhantes aos seguintes:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Configure o Zabbix server para se conectar ao Zabbix web service configurado com TLS editando o parâmetro `WebServiceURL` no arquivo de configuração do server do Zabbix:

```
WebServiceURL=https://example.com:443/report
```

Restringindo o emissor e o assunto permitidos do certificado Quando dois componentes do Zabbix (por exemplo, o server e o agent) estabelecem uma conexão TLS, eles validam os certificados um do outro. Se o certificado do peer for assinado por uma CA confiável (com um certificado de nível superior pré-configurado em `TLSCAFile`), for válido, não estiver expirado e passar por outras verificações, a comunicação entre os componentes poderá prosseguir. Nesse caso mais simples, o emissor e o assunto do certificado não são verificados.

No entanto, isso apresenta um risco: qualquer pessoa com um certificado válido pode se passar por outra pessoa (por exemplo, um certificado de host poderia ser usado para se passar por um server). Embora isso possa ser aceitável em ambientes pequenos, nos quais os certificados são assinados por uma CA interna dedicada e o risco de personificação é baixo, talvez não seja suficiente em ambientes maiores ou mais sensíveis à segurança.

Se a CA de nível superior emitir certificados que não devem ser aceitos pelo Zabbix ou se você quiser reduzir o risco de personificação, poderá restringir os certificados permitidos especificando o emissor e o assunto.

Por exemplo, no arquivo de configuração do proxy do Zabbix, você poderia 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 do Zabbix cujo certificado tenha um emissor ou assunto diferente. Da mesma forma, um proxy passivo não aceitará solicitações desse 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 do RFC 4514 são implementados - caracteres que exigem escape (com uma barra invertida '\', U+005C):
  - em qualquer parte 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 no final 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 nível mais alto (DC) e terminam com o campo de nível mais baixo (CN). A formatação com espaços e separadores de campos também varia de acordo com as opções 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 então estará na ordem reversa, separada por vírgulas, e poderá ser usada 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, algumas 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 Restricting Allowed Certificate Issuer and Subject). 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 ela for usada, geralmente é necessário que os atributos *clientAuth* (para autenticação de cliente TLS WWW) e *serverAuth* (para autenticação de servidor TLS WWW) sejam especificados. Por exemplo:

- Em verificações passivas, nas quais o agent do Zabbix opera como um servidor TLS, o atributo *serverAuth* deve ser incluído no certificado do agent.
- Em verificações ativas, nas quais 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, ele normalmente permite que a comunicação continue apesar desses avisos.

Extensão Name Constraints

O suporte à extensão *Name Constraints* varia entre os toolkits criptográficos. Certifique-se de que o toolkit escolhido suporte essa extensão. Essa extensão pode impedir o Zabbix de carregar certificados CA se esta seção estiver 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
```

Nesse caso, `zabbix_crl_file.crt` pode conter CRLs de várias CAs e pode ter a seguinte aparência:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

O arquivo CRL é carregado somente 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`.

Solução de problemas de certificados OpenSSL usado com CRLs e, para algumas CA na cadeia de certificados, sua CRL não está incluída em `TLSCRLFile`

No log do servidor TLS no caso de um 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 um 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 que 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 reportado 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 as mesmas strings de Issuer e Subject, embora estivesse assinado por uma CA. Issuer e Subject são iguais no certificado CA de nível superior, mas não podem ser iguais no certificado do server. (O mesmo se aplica aos certificados de proxy e agent.)

Para verificar se um certificado contém as mesmas entradas de Issuer e Subject, execute:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

É aceitável que o certificado raiz (de nível superior) tenha valores idênticos para Issuer e Subject.

Criptografia usando chaves pré-compartilhadas

Visão geral Cada chave pré-compartilhada (PSK) no Zabbix é, na verdade, um par de:

- string de identidade da PSK não secreta;
- valor da string secreta da PSK.

A string de identidade da PSK é uma string UTF-8 não vazia. Por exemplo, "PSK ID 001 Zabbix agentd". É um nome exclusivo pelo qual essa PSK específica é referenciada pelos componentes do Zabbix. Não inclua informações confidenciais na string de identidade da PSK, pois ela é transmitida pela rede sem criptografia.

O valor da PSK é uma string difícil de adivinhar, composta por dígitos hexadecimais, por exemplo, `e560cb0d918d26d31b4f642181f5f570ad`

> **Warning:**
> Nunca use em ambientes de produção os valores de PSK de exemplo fornecidos na documentação. Em vez disso, gere uma PSK exclusiva e criptograficamente segura.

Limites de tamanho Há limites de tamanho para a identidade e o valor da PSK no Zabbix; em alguns casos, uma biblioteca criptográfica pode ter um limite menor:

Componente	Tamanho máximo da identidade PSK	Tamanho mínimo do valor da PSK	Tamanho máximo do valor da PSK
Zabbix	128 caracteres UTF-8	128 bits (PSK de 16 bytes, inserida como 32 dígitos hexadecimais)	2048 bits (PSK de 256 bytes, inserida como 512 dígitos hexadecimais)
GnuTLS	128 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserida como 512 dígitos hexadecimais)
OpenSSL 1.0.x, 1.1.0	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserida como 512 dígitos hexadecimais)

Componente	Tamanho máximo da identidade PSK	Tamanho mínimo do valor da PSK	Tamanho máximo do valor da PSK
OpenSSL 1.1.1	127 bytes (pode incluir caracteres UTF-8)	-	512 bits (PSK de 64 bytes, inserida como 128 dígitos hexadecimais)
OpenSSL 1.1.1a e posteriores	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (PSK de 256 bytes, inserida como 512 dígitos hexadecimais)

Attention:
O frontend do Zabbix permite configurar uma string de identidade PSK com até 128 caracteres e uma PSK de até 2048 bits, independentemente das bibliotecas criptográficas utilizadas.
 Se alguns componentes do Zabbix oferecerem suporte a limites menores, será responsabilidade do usuário configurar a identidade e o valor da PSK com o tamanho permitido por esses componentes.
 Exceder os limites de tamanho resulta em falhas de comunicação entre os componentes do Zabbix.

Antes de o server do Zabbix se conectar ao agent usando uma PSK, o server procura no banco de dados a identidade e o valor da PSK configurados para esse agent (na verdade, no cache de configuração). Ao receber uma conexão, o agent usa a identidade e o valor da PSK do seu arquivo de configuração. Se ambas as partes tiverem a mesma string de identidade PSK e o mesmo valor da PSK, a conexão poderá ser estabelecida.

Attention:
Cada identidade PSK deve estar associada a apenas um valor. É responsabilidade do usuário garantir que não existam duas PSKs com a mesma string de identidade, mas com valores diferentes. Não fazer isso pode causar erros imprevisíveis ou interrupções na comunicação entre os componentes do Zabbix que usam PSKs com essa string de identidade PSK.

Gerando PSK Por exemplo, uma PSK de 256 bits (32 bytes) pode ser gerada 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 a PSK associada. O Zabbix espera apenas uma PSK no arquivo PSK; portanto, a string de identidade e os dois-pontos (':') devem ser removidos do arquivo.

Configurando PSK para a comunicação entre server e agent (exemplo) No host do agent, grave o valor de PSK gerado anteriormente em um arquivo, por exemplo, `/home/zabbix/zabbix_agentd.psk`. O arquivo deve conter o valor de PSK na primeira linha.

Defina as permissões de acesso ao arquivo PSK: ele deve ser legível somente pelo usuário Zabbix.

Edite os parâmetros TLS no arquivo de configuração do agent do Zabbix (`zabbix_agentd.conf`). Por exemplo, defina:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

O agent se conectará ao server (verificações ativas) e aceitará do server e do `zabbix_get` somente conexões que usem PSK. A identidade PSK será "PSK 001".

Reinicie o agent. Agora você pode testar a conexão usando `zabbix_get`, por exemplo:

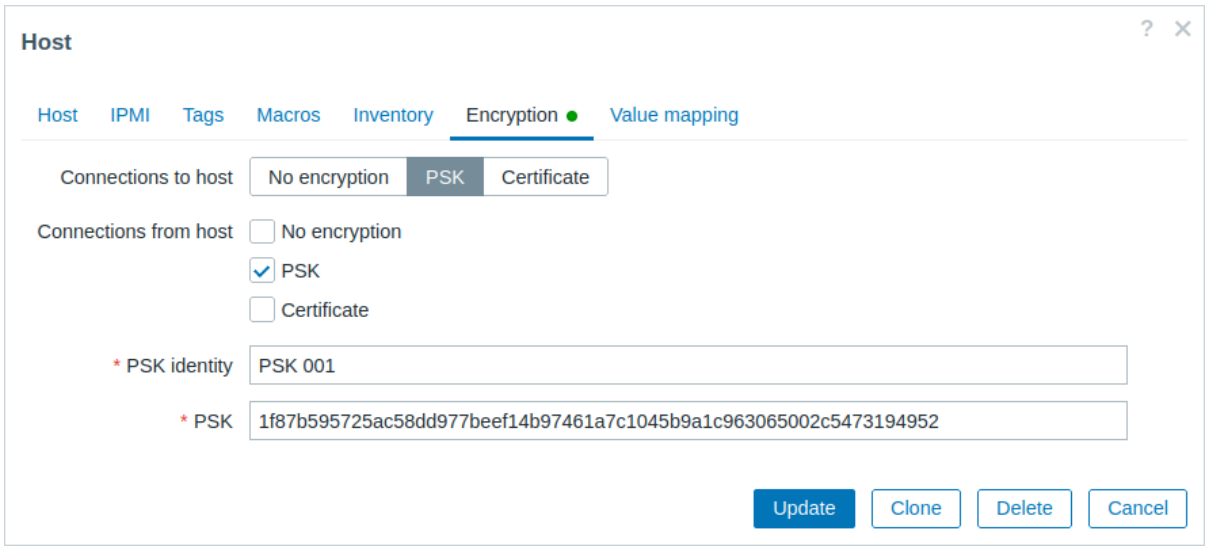
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Para minimizar o tempo de indisponibilidade, consulte como alterar o tipo de conexão em Gerenciamento da criptografia da conexão).

Configure a criptografia PSK para este agent no frontend do Zabbix:

1. Acesse: *Coleta de dados* > *Hosts*.
2. Selecione o host e clique na aba *Criptografia*.
3. Configure os detalhes da criptografia PSK.

Exemplo:



Todos os campos de entrada obrigatórios são 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 em busca de mensagens de erro.

Configurando PSK para a comunicação entre server e proxy ativo (exemplo) No proxy, grave 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 somente pelo usuário Zabbix.

Edite os parâmetros TLS no arquivo de configuração do proxy do Zabbix (`zabbix_proxy.conf`). Por exemplo, defina:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

O proxy se conectará ao server usando PSK. A identidade do PSK será "PSK 002".

(Para minimizar o tempo de inatividade, consulte como alterar o tipo de conexão em Gerenciamento da criptografia de conexão).

Configure o PSK para este proxy no frontend do Zabbix. Acesse *Administration* > *Proxies*, selecione o proxy e acesse a aba *Encryption*. Em "Connections from proxy", marque PSK. Cole "PSK 002" no campo "PSK identity" e "e560cb0d918d26d31b4f642181f5f570ad89a3909 no campo "PSK". Clique em "Update".

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 em busca de mensagens de erro.

Para um proxy passivo, o procedimento é muito semelhante. A única diferença é definir `TLSAccept=psk` no arquivo de configuração do proxy do Zabbix e definir *Connections to proxy* no frontend do Zabbix como PSK.

Solução de problemas de PSK PSK contém um número ímpar de dígitos hexadecimais

O proxy ou agent não inicia, mensagem no log do proxy ou agent:

```
PSK inválido no arquivo "/home/zabbix/zabbix_proxy.psk"
```

String de identidade PSK maior que 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 de PSK muito longo usado com OpenSSL 1.1.1

No log do lado que está conectando:

```
...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 que está aceitando:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Esse problema normalmente ocorre ao atualizar o OpenSSL da versão 1.0.x ou 1.1.0 para 1.1.1 e se o valor de PSK for maior que 512 bits (PSK de 64 bytes, informado como 128 dígitos hexadecimais).

Veja também: Limites de tamanho de valor

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:

Database	Zabbix components
MySQL	Zabbix frontend, Zabbix server, Zabbix proxy
PostgreSQL	Zabbix frontend, Zabbix server, Zabbix proxy

Para configurar a criptografia da conexão dentro do DBMS, consulte a documentação oficial do fornecedor para obter detalhes:

- MySQL: servidores de banco de dados de replicação de origem e réplica.
- MySQL: servidores de banco de dados de replicação em grupo etc.
- PostgreSQL opções de criptografia.

Todos os exemplos são baseados nas versões GA do MySQL CE (8.0) e PgSQL (13) disponíveis por meio dos repositórios oficiais usando CentOS 8.

Requisitos

O seguinte é necessário para configurar a criptografia:

- Sistema operacional com suporte do desenvolvedor e OpenSSL >=1.1.X ou alternativa.

> **Note:**
> É recomendável 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 distribuídos com versões desatualizadas de 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 server/proxy do Zabbix e do 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 que a identidade do banco de dados (CN) especificada por DBHost corresponde ao seu certificado

Configuração do Zabbix

Frontend para o banco de dados

Uma conexão segura com o banco de dados pode ser configurada durante a instalação do frontend:

- Marque a caixa de seleção *Criptografia TLS do banco de dados* na etapa Configurar conexão com o banco de dados para habilitar a criptografia de transporte.
- Marque a caixa de seleção *Verificar certificado do banco de dados*, que aparece quando o campo *Criptografia TLS* está marcado, para habilitar a criptografia com certificados.

Note:
Para o MySQL, a caixa de seleção *Criptografia TLS do banco de dados* fica desabilitada se *Host do banco de dados* estiver definido como localhost, pois uma conexão que usa um arquivo de socket (no Unix) ou memória compartilhada (no Windows) não pode ser criptografada.
 Para o PostgreSQL, a caixa de seleção *Criptografia TLS* fica desabilitada se o valor do campo *Host do banco de dados* começar com uma barra ou se o campo estiver vazio.

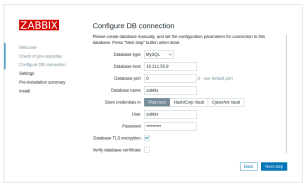
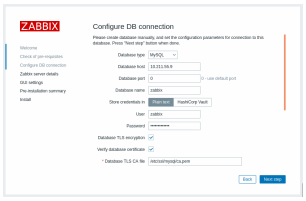
Os parâmetros a seguir ficam disponíveis no modo de criptografia TLS com certificados (se ambas as caixas de seleção estiverem marcadas):

Parâmetro	Descrição
Arquivo da CA TLS do banco de dados	Especifique o caminho completo para um arquivo válido de autoridade de certificação (CA) TLS.
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 de seleção para ativar a verificação do host. Desabilitado para o MySQL porque a biblioteca PHP do MySQL não permite ignorar a etapa de validação do certificado do peer.
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 somente para o MySQL.

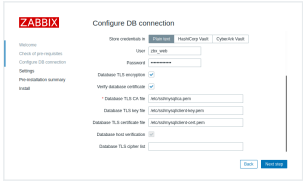
Attention:
Os parâmetros TLS devem apontar para arquivos válidos. Se apontarem para arquivos inexistentes ou inválidos, isso resultará em um erro de autorização.
 Se os arquivos de certificado forem graváveis, o frontend gerará um aviso no relatório Informações do sistema: "Os arquivos de certificado TLS devem ser somente leitura." (exibido somente se o usuário PHP for o proprietário do certificado).
Certificados protegidos por senhas não são compatíveis.

Casos de uso

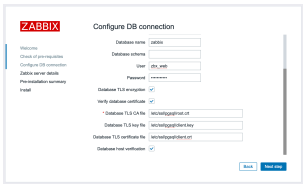
O frontend do Zabbix usa uma interface gráfica para definir as opções possíveis: `required`, `verify_ca`, `verify_full`. Especifique as opções necessárias na etapa *Configurar conexões com o banco de dados* do assistente de instalação. Essas opções são mapeadas para o arquivo de configuração do frontend (`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'] = ''; ...	1. Marque *Criptografia TLS do banco de dados* 2. 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 o *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 uma 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 uma lista de cifras definida - selecionada pelo server MySQL
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados*
		2. Especifique o caminho para o *arquivo de chave TLS do banco de dados*
		3. Especifique o caminho para o *arquivo CA TLS do banco de dados*
		4. Especifique o caminho para o *arquivo de certificado TLS do banco de dados*
		5. Especifique a *lista de cifras TLS do banco de dados* (opcional)
			Habilita o modo `verify_full` para MySQL.

Configurações da GUI	Arquivo de configuração	Descrição	Resultado

	...
	`$DB['ENCRYPTION'] = true;`
	`$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 o *arquivo de chave TLS do banco de dados*
		3. Especifique o caminho para o *arquivo CA TLS do banco de dados*
		4. Especifique o caminho para o *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

As conexões seguras com o banco de dados podem ser configuradas com os respectivos parâmetros no arquivo de configuração do server e/ou do proxy do Zabbix.

Configuração	Resultado
Nenhuma	Conexão com o banco de dados sem criptografia.
Defina `DBTLSConnect=required`	O server/proxy estabelece 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 de certificação TLS	O server/proxy estabelece 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 de certificação TLS	O server/proxy estabelece 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 de certificação 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.
Defina `DBTLSCipher` - a lista de cifras de criptografia permitidas pelo cliente para conexões que usam protocolos TLS até TLS 1.2 ou `DBTLSCipher13` - a lista de cifras de criptografia permitidas pelo cliente para conexões que usam o protocolo TLS 1.3	(MySQL) A conexão TLS é estabelecida usando uma cifra da lista fornecida. (PostgreSQL) A definição desta opção será considerada um erro.

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 de início 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. Nesse caso, uma conexão entre o frontend do Zabbix e o banco de dados usa um arquivo de socket (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. Há 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 server 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

As versões modernas do banco de dados já vêm prontas para uso com o modo de criptografia `required`. Um certificado no lado do server será criado após a configuração inicial e a inicialização.

Crie usuários e funções para os componentes principais (substitua `<strong_password>` por uma senha forte e exclusiva):

```
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 é configurado para usar somente conexões criptografadas. Consulte a documentação do MySQL para obter mais detalhes sobre a configuração de usuários.

Execute o comando a seguir para verificar a conexão (uma conexão por 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

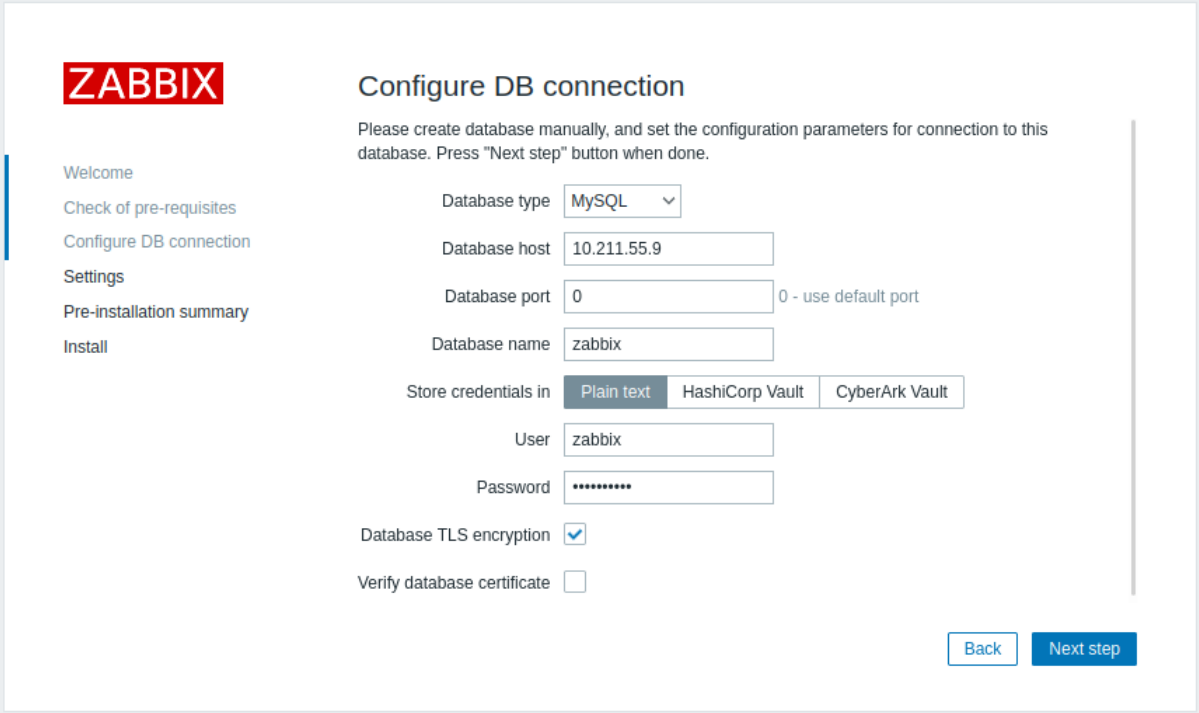

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

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 *Database TLS encryption*
- Deixe *Verify database certificate* desmarcado



Server

Para habilitar a criptografia somente de transporte para conexões entre o server e o banco de dados, edite o arquivo de configuração do server:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Criptografia com verificação da autoridade certificadora

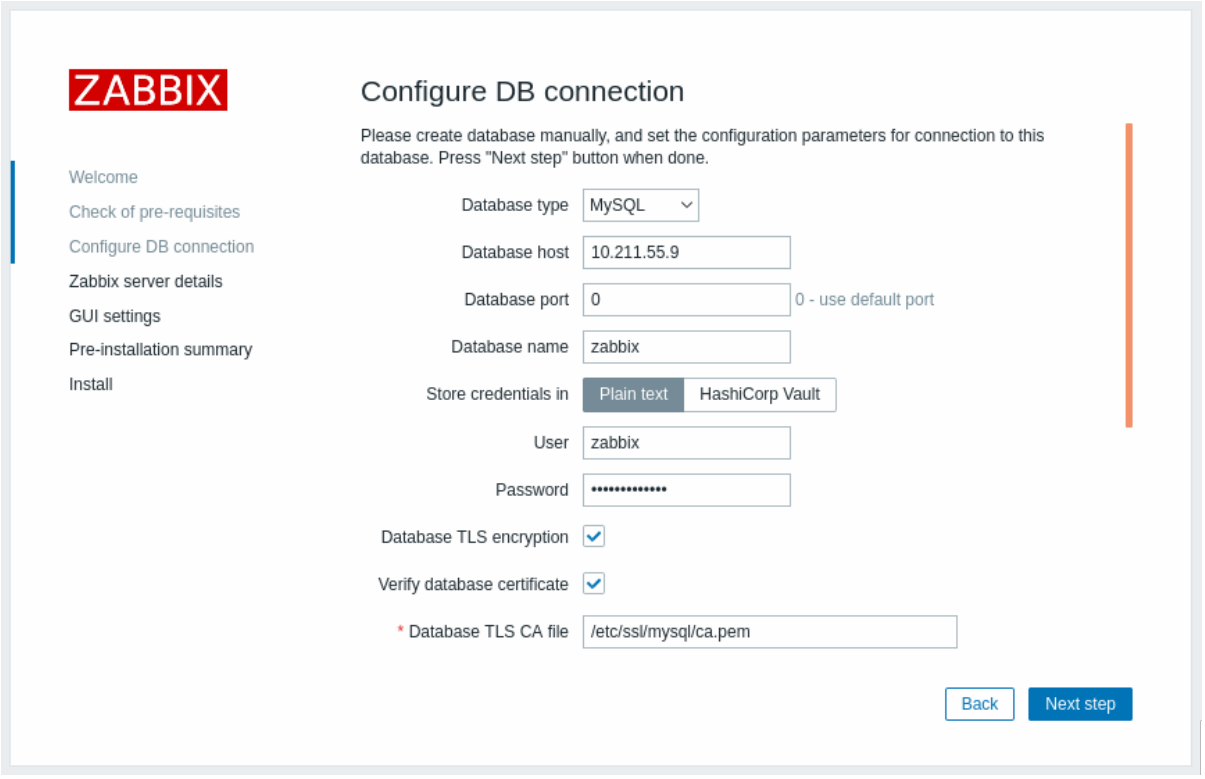
Copie a CA do MySQL necessária para o server frontend do Zabbix e atribua as permissões adequadas para permitir que o servidor web leia este arquivo.

> **Note:**
> Este modo não funciona no RHEL 7 devido a bibliotecas MySQL mais antigas.

frontend

Para habilitar a criptografia com verificação de certificado nas 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 CA do TLS do banco de dados



Como alternativa, isso pode ser definido no arquivo de configuração do frontend (`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 nas conexões entre o Zabbix server e o banco de dados, edite o arquivo de configuração do server:

```
...
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*) como:

```
[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: Creating SSL and RSA certificates and keys using MySQL ou Creating SSL certificates and keys using openssl

Attention:
O certificado do servidor MySQL deve conter o campo Common Name definido como 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
```

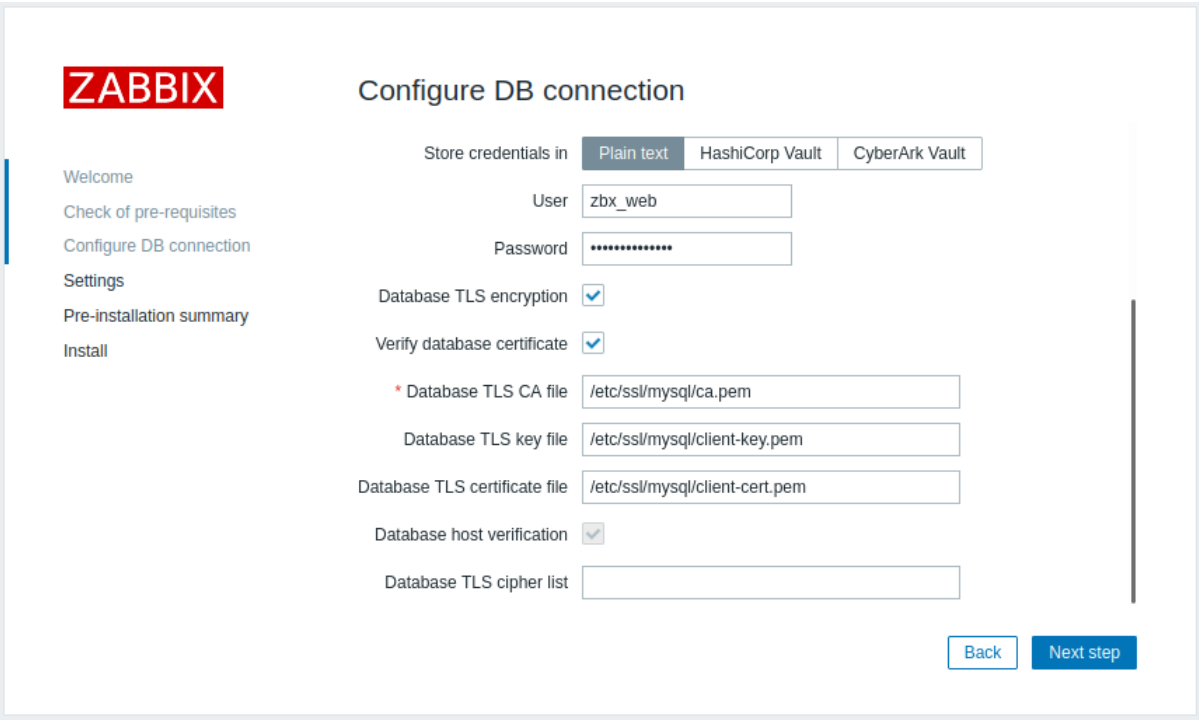
Frontend

Para habilitar a criptografia com verificação completa nas 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 de CA TLS do banco de dados*
- Especifique o caminho para o *arquivo de certificado TLS do banco de dados*

Observe que *Verificação do host do banco de dados* está marcada e desabilitada - essa etapa não pode ser ignorada para o MySQL.

Warning:
Se o campo *Lista de cifras TLS do banco de dados* estiver vazio, serão habilitadas as cifras comuns permitidas tanto pelo frontend (cliente) quanto pelo server. Como alternativa, as cifras podem ser definidas explicitamente, de acordo com os requisitos de configuração de cifras.



Como alternativa, isso pode ser definido no arquivo de configuração do frontend (`zabbix.conf.php`):

```
...
// Usado para conexão TLS com uma lista de cifras 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 uma lista de cifras definida - selecionada pelo server MySQL
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Para habilitar a criptografia com verificação completa das conexões entre o Zabbix server e o banco de dados, edite o arquivo de configuração do server:

```
...
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
...
```

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 desativados) 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 é configurado para aceitar conexões TLS por padrão. Siga as instruções da documentação do PostgreSQL para a preparação de certificados com postgresql.conf e também para controle de acesso de usuários por meio de `ph_hba.conf`.

Por padrão, o socket do PostgreSQL fica vinculado ao localhost; para conexões remotas pela rede, permita a escuta na interface de rede real.

As configurações do PostgreSQL para todos os modes 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

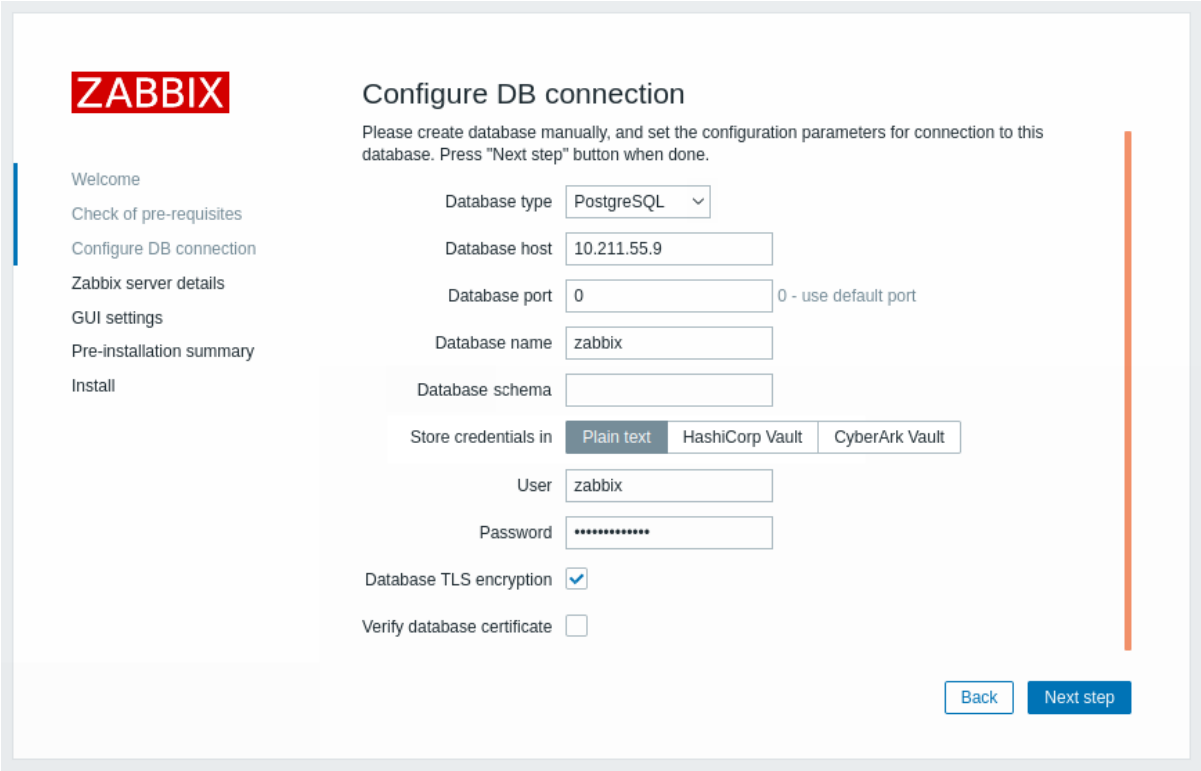
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Criptografia apenas para transporte

Frontend

Para habilitar a criptografia apenas de transporte para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Database TLS encryption*
- Deixe *Verify database certificate* desmarcado



server

Para habilitar a criptografia somente de transporte para conexões entre o server e o banco de dados, edite o arquivo de configuração do server:

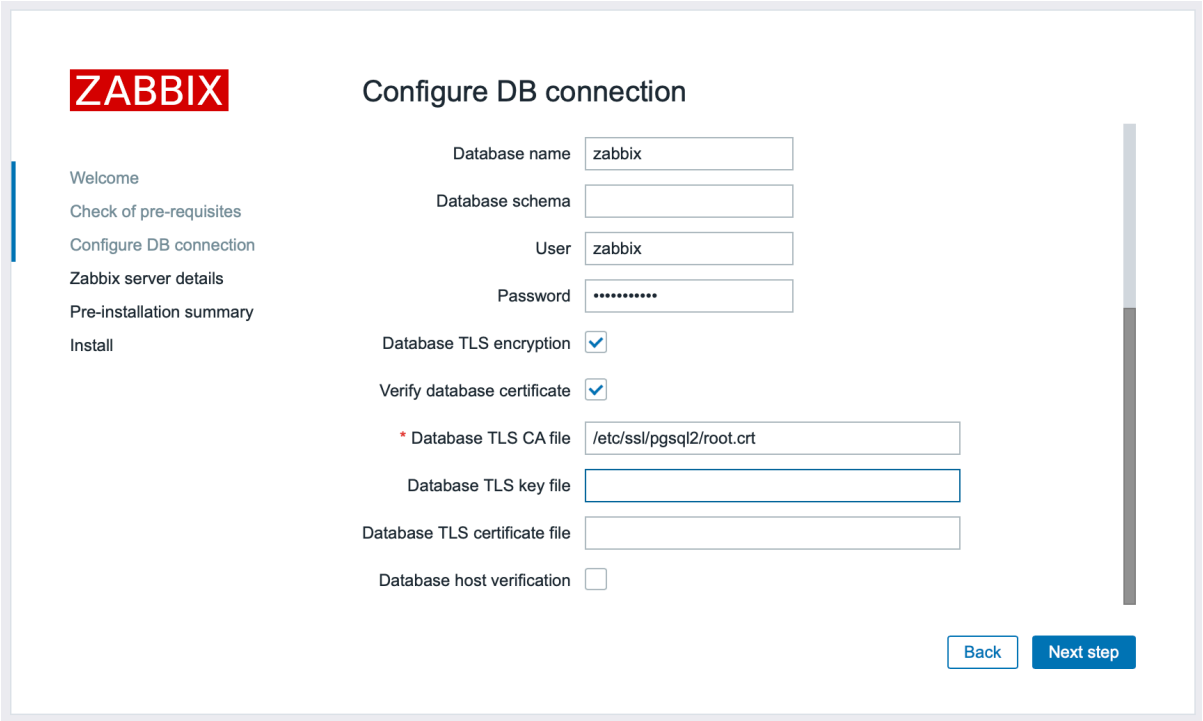
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Criptografia com verificação da autoridade certificadora

Frontend

Para habilitar a criptografia com verificação da autoridade certificadora nas 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*



Como alternativa, isso pode ser definido no arquivo de configuração do frontend (`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 nas conexões entre o Zabbix server e o banco de dados, edite o arquivo de configuração do server (substitua `<strong_password>` por uma senha forte e exclusiva):

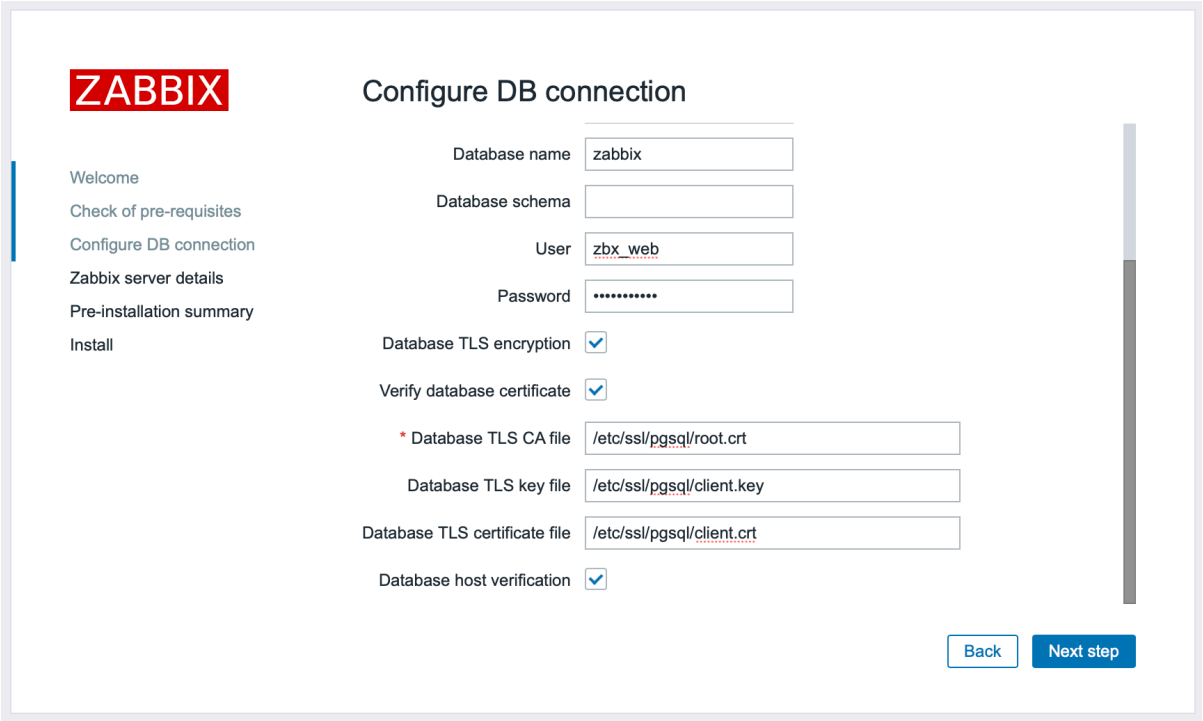
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Criptografia com verificação completa

Frontend

Para habilitar a criptografia com certificado e a verificação da identidade do host do banco de dados nas 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 de 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*



Como alternativa, isso pode ser definido no arquivo de configuração do frontend (`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 do certificado e da identidade do host do banco de dados nas conexões entre o Zabbix server e o banco de dados, edite o arquivo de configuração do server:

```
...
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
...
```

Conexão segura entre frontend e server

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 aumentar a segurança, habilite o TLS em ambos os lados. Abaixo está um exemplo da maneira mais simples de fazer isso.

1. Gere os certificados e as chaves.

Crie um diretório de trabalho:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Crie um certificado de 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 apropriadas.

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 arquivo de configuração do server (`zabbix_server.conf`), adicione:

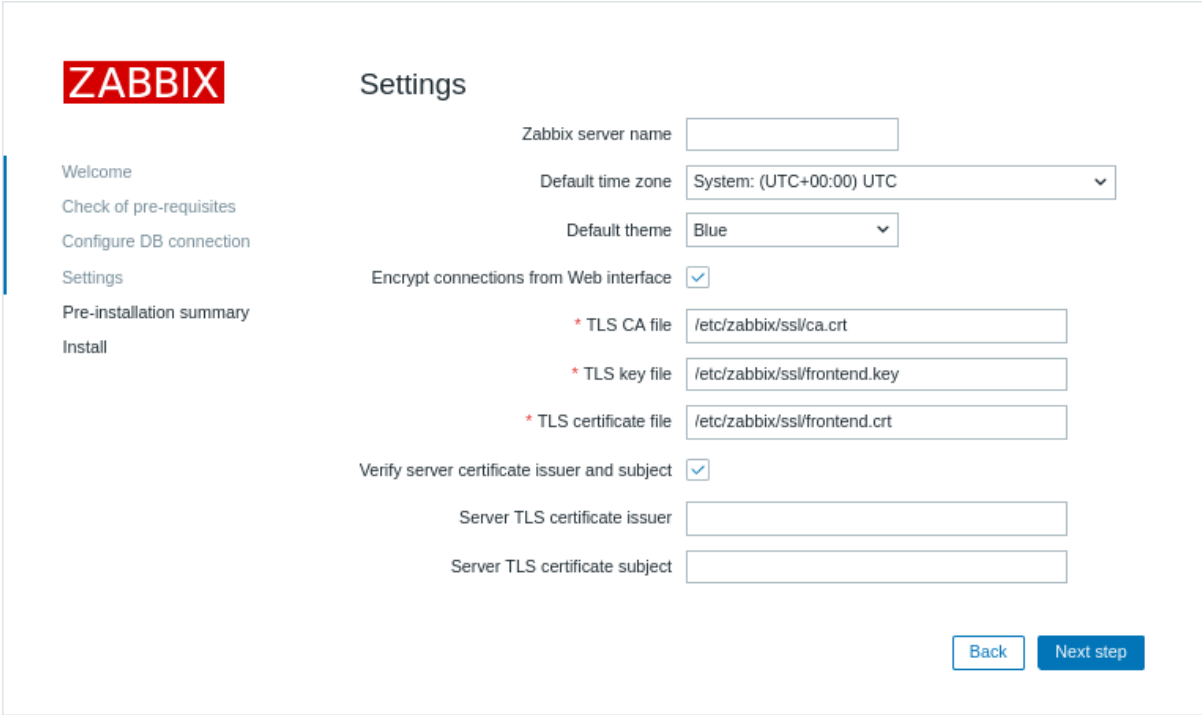
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### Opcionalmente:
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

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 server*, se necessário) e preencha os campos *Arquivo de CA do TLS*, *Arquivo de chave do TLS*, *Arquivo de certificado do TLS* (e os campos *Emissor do certificado TLS do server* e *Assunto do certificado TLS do server*, se necessário):



Parâmetro	Descrição
Arquivo de CA do TLS	Especifique o caminho completo para o arquivo de certificado da Autoridade de Certificação (CA) usado para verificar o certificado do server.
Arquivo de chave do TLS	Especifique o caminho completo para o arquivo de chave privada do cliente correspondente ao certificado do cliente.
Arquivo de certificado do TLS	Especifique o caminho completo para o arquivo de certificado do cliente se a autenticação TLS mútua for necessária.
Emissor do certificado TLS do server	Especifique um Nome Distinto (DN) do emissor para comparar com o certificado do server.
Assunto do certificado TLS do server	Especifique um Nome Distinto (DN) do assunto para comparar com o certificado do server.

Em instalações existentes, edite os seguintes parâmetros no arquivo de configuração do frontend:

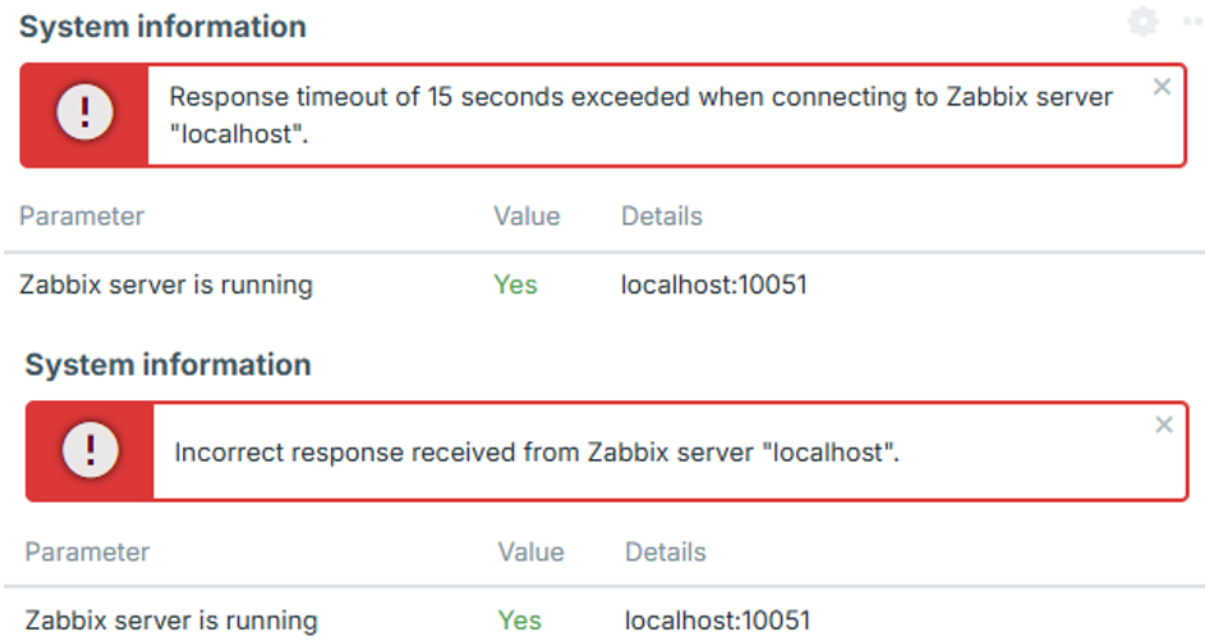
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Opcionalmente:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $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
```

Solução de problemas

Uma configuração TLS incorreta ou incompleta pode causar erros no frontend do Zabbix ou indicar que o Zabbix server está em execução, mas a conexão não foi estabelecida com sucesso:



Para identificar o problema exato de configuração, defina o nível de log do Zabbix server como 4, reproduza o erro no frontend e verifique o log do Zabbix server em busca da mensagem de erro correspondente.

Os casos a seguir descrevem erros comuns e suas possíveis causas.

O endereço IP do frontend não é permitido

O log do Zabbix server contém um erro semelhante a:

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

A conexão é encerrada após o handshake TCP porque o endereço IP do frontend não está autorizado.

Adicione o endereço IP do frontend ao parâmetro `FrontendAllowedIP` em `zabbix_server.conf` e reinicie o Zabbix server.

Conexão não criptografada não é permitida

O log do Zabbix server contém um erro semelhante a:

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

O Zabbix server está configurado para aceitar somente conexões TLS, por exemplo, com `TLSFrontendAccept=cert`, mas o frontend está tentando se conectar sem criptografia.

Habilite a criptografia do frontend e configure os arquivos TLS necessários em `zabbix.conf.php`, conforme descrito na etapa 4 de Configuração.

O frontend tenta usar TLS enquanto o server espera uma conexão não criptografada

O log do Zabbix server contém um erro semelhante a:

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

A mensagem de erro exata depende da versão do OpenSSL.

O Zabbix server está configurado para aceitar apenas conexões não criptografadas, por exemplo, com `TLSFrontendAccept=unencrypted`, enquanto o frontend tenta estabelecer uma conexão TLS.

Configure o Zabbix server para aceitar conexões TLS definindo `TLSFrontendAccept=cert` e especificando `TLSCertFile` e `TLSKeyFile` em `zabbix_server.conf`. Como alternativa, desative a criptografia do frontend em `zabbix.conf.php`.

O certificado do frontend não é confiável

O log do Zabbix server contém um erro semelhante a:

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

O frontend apresentou um certificado, mas o Zabbix server não confia na Autoridade de Certificação (CA) que o assinou.

O mesmo problema também pode aparecer nas informações do sistema do frontend como:

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

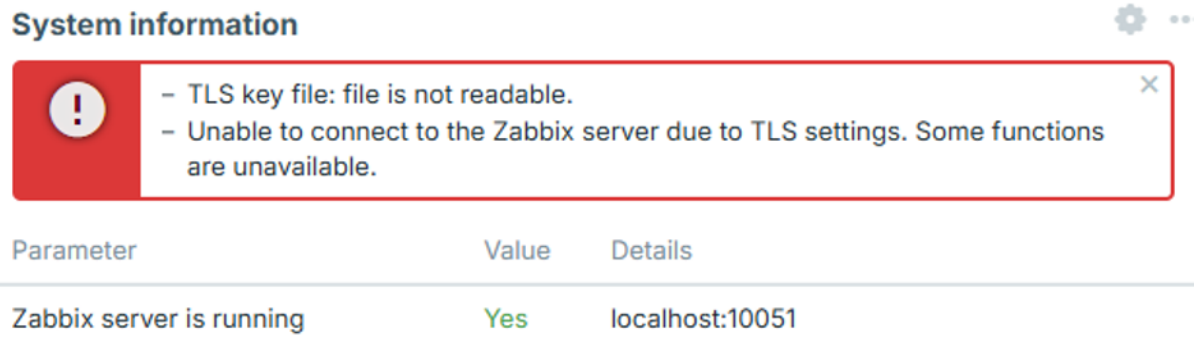
O frontend também pode exibir a mensagem ”Incorrect response received from Zabbix server ”localhost””.

Verifique se o certificado do frontend foi assinado pela CA especificada por `TLSCAFile` em `zabbix_server.conf`.

Arquivo de chave TLS não pode ser lido

O frontend pode exibir ”Arquivo de chave TLS: o arquivo não pode ser lido.”

Também pode exibir ”Não foi possível conectar-se ao server Zabbix devido às configurações de TLS. Algumas funções estão indisponíveis.”.

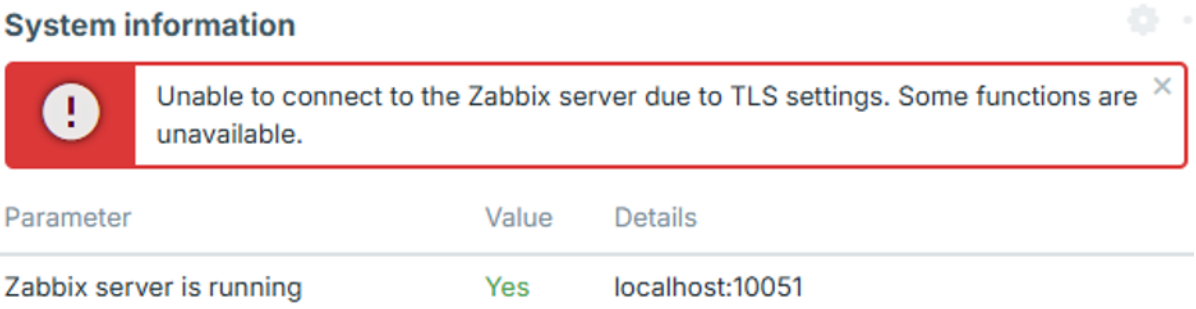


O usuário do web server não tem permissões suficientes para ler um ou mais arquivos de certificado, chave ou CA configurados em `zabbix.conf.php`. Isso pode ocorrer quando os arquivos de certificado estão armazenados em um diretório restrito, como /root/ ou no diretório inicial de um usuário.

Mova os arquivos de certificado para um diretório acessível ao web server e defina as permissões apropriadas conforme descrito na etapa 2 de Configuração.

Arquivos TLS inválidos ou corrompidos

O frontend pode exibir ”Não foi possível conectar-se ao Zabbix server devido às configurações de TLS. Algumas funções estão indisponíveis.”.



Isso pode ocorrer quando os arquivos de certificado ou chave configurados existem e podem ser lidos, mas seu conteúdo está vazio, corrompido ou é inválido por outro motivo. Nesse caso, o frontend não estabelece uma conexão com o Zabbix server, e nenhuma tentativa de conexão correspondente aparece no log do Zabbix server.

Verifique se os arquivos de certificado e chave contêm dados válidos e formatados corretamente e se a chave privada corresponde ao certificado.

Controle de acesso

Visão geral

Esta seção contém práticas recomendadas 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 possível de privilégios. Isso significa que as contas de usuário no frontend do Zabbix, os usuários do banco de dados ou o usuário dos 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 os privilégios das 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 a configuração e executar scripts personalizados.

Note:
Algumas informações estão disponíveis até mesmo para usuários sem privilégios. Por exemplo, embora *Alerts* → *Scripts* esteja disponível apenas para usuários *Super Admin*, os scripts também podem ser recuperados por meio da API do Zabbix. Nesse caso, limitar as permissões de 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 único usuário `zabbix`. Para impedir que o Zabbix agent/agent 2 (em execução na mesma máquina que o server/proxy) acesse detalhes sensíveis 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 seguros (por exemplo, `zabbix-agent`).
2. Defina esse usuário no parâmetro User do arquivo de configuração do agent.
3. Reinicie o agent para remover privilégios e passar a usar o novo usuário.

Para o Zabbix agent 2, a configuração deve ser aplicada no nível do service, porque o arquivo de configuração do agent 2 não suporta o parâmetro `User`. Para um exemplo, veja ZBX-26442.

Revogar o acesso de gravação à configuração SSL (Windows)

Se você compilou o agent Zabbix no Windows, com o OpenSSL localizado em um diretório desprotegido (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 possível vulnerabilidade de segurança.

Reforço da segurança dos componentes do Zabbix

Algumas funcionalidades podem ser desativadas para reforçar a segurança dos componentes do Zabbix:

- A execução global de scripts no Zabbix server pode ser desativada definindo `EnableGlobalScripts=0` na configuração do server.
- A execução global de scripts no Zabbix proxy é desativada por padrão (pode ser ativada definindo `EnableRemoteCommands=1` na configuração do proxy).
- A execução global de scripts nos Zabbix agents é desativada por padrão (pode ser ativada adicionando um parâmetro `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` para cada comando permitido na configuração do agent).
- O acesso à configuração da autenticação HTTP de usuários é ativado por padrão; pode ser desativado definindo `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` na configuração do frontend.
- O acesso à configuração de módulos é ativado por padrão; pode ser desativado definindo `$ZBX_FEATURE_FLAGS['modules_config_ = false` na configuração do frontend.
- A edição de tipos de mídia específicos pode ser restringida definindo `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` na configuração do frontend. Valores possíveis: `'email'`, `'script'`, `'sms'`, `'webhook'` e `'push'`. Um ou mais valores podem ser especificados.

A reinstalação do frontend executando `setup.php` é restrita se o arquivo de configuração `zabbix.conf.php` existir.

Acesso a caminhos UNC no Windows pelo Zabbix agent

Os Zabbix agents no Windows seguem caminhos UNC (compartilhamentos SMB, como `\\server\share\file.txt`) em itens como `vfs.file.*`, `vfs.dir.*`, `modbus.get` e `perf_counter*`. Isso pode representar um risco de segurança em alguns contextos.

Quando o Windows é solicitado a acessar um caminho UNC, ele tenta se autenticar nesse server. Isso significa que uma solicitação maliciosa ao Zabbix agent pode expor o hash NTLM ao server do solicitante. Os usuários podem reduzir esse risco configurando os parâmetros de configuração AllowKey, AllowKeyRegexp, DenyKey e DenyKeyRegexp conforme necessário.

Protegendo MySQL/MariaDB

Visão geral

Esta seção contém práticas recomendadas 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 completos no banco de dados do Zabbix. Esse 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, é recomendável criar funções e usuários adicionais do banco de dados com privilégios mínimos. Essas funções e usuários devem ser configurados com base no princípio do menor privilégio, ou seja, 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 a 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 a atualização das 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 depois que o banco de dados tiver 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';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

Após criar as funções, elas podem ser atribuídas aos usuários.

Atribuição de funções de usuário

Para atribuir as funções de usuário criadas, crie usuários e atribua a eles as funções relevantes.
Substitua <user>, <host>, <role> e <password> pelos valores apropriados.
Certifique-se de que <password> seja forte e exclusivo.

```
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 o proxy:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

Protegendo PostgreSQL/TimescaleDB

Visão geral

Esta seção contém práticas recomendadas 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 completos no banco de dados do Zabbix. Esse 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 adicionais do banco de dados com privilégios mínimos. Essas funções e usuários devem ser configurados com base no princípio do menor privilégio, ou seja, 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 do Zabbix; o proprietário do banco de dados é o usuário especificado na criação do banco:

```
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 nem revogado.

> **Attention:**
> Os comandos a seguir nesta página devem ser executados enquanto a conexão com o PostgreSQL estiver sendo feita especificamente para o banco de dados `zabbix`.

Crie o schema `zabbix` e defina o proprietário do banco de dados (`usr_owner`) como proprietário desse schema:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Configure um padrão de uso seguro do schema usage pattern:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Depois de 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 server e o proxy do Zabbix:

```
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 frontend e a API do Zabbix:

```
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.

Depois de criar as funções, elas podem ser atribuídas aos usuários.

Atribuição de funções de usuário

Para atribuir as funções de usuário criadas, crie usuários e atribua a eles as funções relevantes.
Substitua `<user>`, `<role>` e `<password>` pelos valores apropriados.
Certifique-se de que `<password>` seja forte e exclusivo.

```
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 o proxy:

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-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 server Zabbix requer privilégios de proprietário do banco de dados.

Se o schema PostgreSQL `zabbix` já tiver sido 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
```

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 informar o nome de domínio que deseja associar ao seu server. Você pode informar o endereço IP público, em vez disso, 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 do Apache para aplicar as alterações:

```
systemctl restart httpd.service
```

Protegendo o web server

Visão geral

Esta seção contém práticas recomendadas para configurar o web server de forma segura.

Forçando o redirecionamento da URL raiz para o Zabbix SSL

Em sistemas baseados em RHEL, adicione um host virtual à configuração do Apache (`/etc/httpd/conf/httpd.conf`) e defina um redirecionamento permanente da raiz do documento para a URL SSL do Zabbix. Observe que `example.com` deve ser substituído pelo nome real do server.

```
### Adicione as linhas:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Reinicie o serviço do Apache para aplicar as alterações:

```
systemctl restart httpd.service
```

Habilitando HTTP Strict Transport Security (HSTS) no web server

Para proteger o frontend do Zabbix contra ataques de downgrade de protocolo, recomendamos habilitar a política HSTS no web server.

Para habilitar a política HSTS para o seu frontend do Zabbix na configuração do Apache, siga estas etapas:

1. Localize o arquivo de configuração do seu virtual host:

- `/etc/httpd/conf/httpd.conf` em sistemas baseados em RHEL
- `/etc/apache2/sites-available/000-default.conf` em 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 do Apache para aplicar as alterações:

```
### Em sistemas baseados em RHEL:
systemctl restart httpd.service
```

```
### Em 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 aos cookies de sessão para aumentar a segurança e evitar ataques de falsificação de solicitação entre sites (CSRF). No entanto, aplicar `SameSite=Strict` pode causar problemas em certos cenários, como:

- Widgets de URL do dashboard exibindo "user not logged in" ao incorporar iframes do mesmo domínio.
- Usuários que acessam 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 Secure

Definir a flag `secure` garante que os cookies sejam transmitidos apenas por HTTPS, evitando exposição em conexões não criptografadas.

Para habilitar cookies secure no Zabbix, adicione ou modifique a seguinte configuração na configuração do web server:

Para Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Para Nginx:

```
proxy_cookie_path / "/; Secure";
```

Certifique-se de que seu frontend do Zabbix seja acessado via HTTPS; caso contrário, cookies com a flag `Secure` não serão enviados.

2. Configurando o atributo SameSite

As configurações do web server 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 maior especificidade
```

Habilitando a Content Security Policy (CSP) no web server

Para proteger o frontend do Zabbix contra Cross Site Scripting (XSS), injeção de dados e ataques semelhantes, recomendamos habilitar a Content Security Policy no web server. Para isso, configure o web server para retornar o HTTP header.

> **Attention:**
> A configuração de header CSP a seguir é válida 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 header CSP pode ser necessária se você estiver, por exemplo, configurando o widget *URL* para exibir conteúdo de subdomínios do site ou domínios externos, trocando de *OpenStreetMap* para outro mecanismo de mapa, ou adicionando CSS ou widgets externos. Se você estiver usando o método de autenticação multifator multi-factor authentication Duo Universal Prompt, certifique-se de adicionar "duo.com" à diretiva CSP no arquivo de configuração do seu virtual host.

Para habilitar CSP para o seu frontend do Zabbix na configuração do Apache, siga estas etapas:

1. Localize o arquivo de configuração do seu virtual host:

- `/etc/httpd/conf/httpd.conf` em sistemas baseados em RHEL
- `/etc/apache2/sites-available/000-default.conf` em Debian/Ubuntu

2. Adicione a seguinte diretiva ao arquivo de configuração do seu virtual host:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Reinicie o serviço Apache para aplicar as alterações:

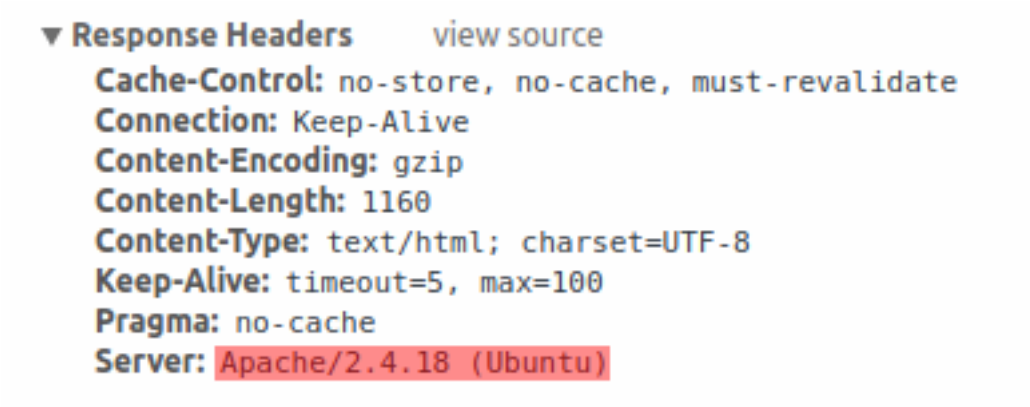
```
### Em sistemas baseados em RHEL:
systemctl restart httpd.service

### Em Debian/Ubuntu
systemctl restart apache2.service
```

Desabilitando a exposição de informações do web server

Para melhorar a segurança, é recomendável desabilitar todas as assinaturas do web server.

Por padrão, o web server 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 web server para que as alterações no arquivo de configuração sejam aplicadas.

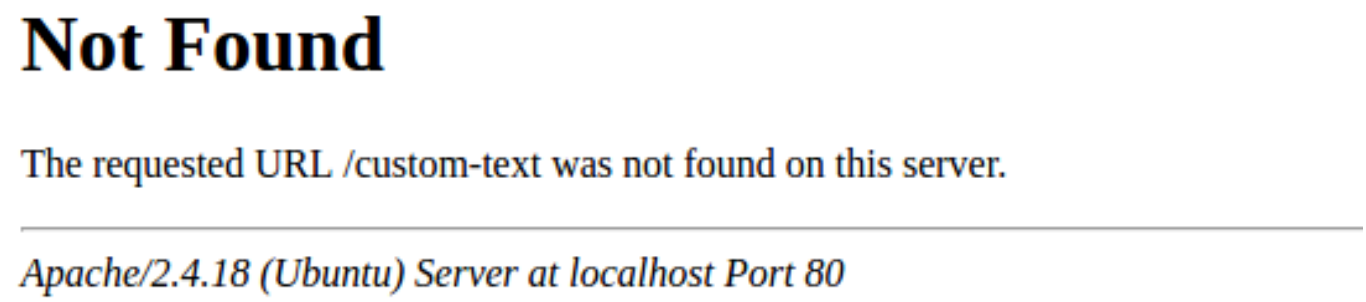
Para obter segurança adicional, você pode usar a ferramenta *mod_security* com o Apache (pacote *libapache2-mod-security2*). Essa ferramenta permite remover a assinatura do server, em vez de remover apenas a versão da assinatura do server. A assinatura do server pode ser alterada para qualquer valor definindo "SecServerSignature" com o valor desejado após a instalação do *mod_security*.

Consulte a documentação do seu web server para obter ajuda sobre como remover ou alterar assinaturas de software.

Desabilitando páginas de erro padrão do web server

Para evitar a exposição de informações, é recomendável desabilitar as páginas de erro padrão.

Por padrão, um web server usa páginas de erro integradas:



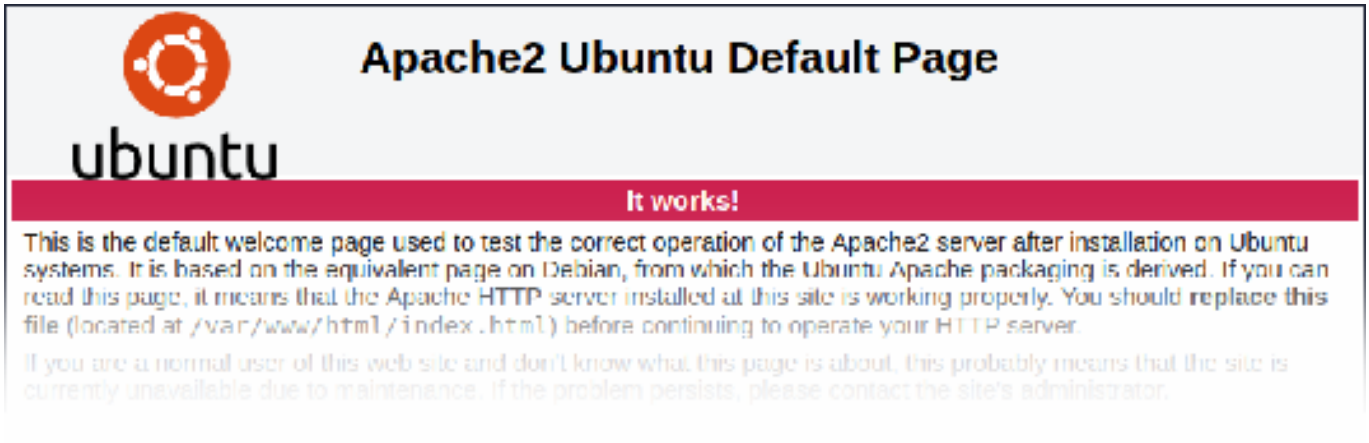
Essas páginas de erro padrão devem ser substituídas/removidas. Por exemplo, a diretiva "ErrorDocument" pode ser usada para definir uma página/texto de erro personalizado para o web server Apache.

Consulte a documentação do seu web server para obter ajuda sobre como substituir/remover as páginas de erro padrão.

Removendo a página de teste do web server

Para evitar a exposição de informações, é recomendável remover a página de teste do web server.

Por padrão, o webroot do web server Apache contém a página de teste `index.html`:



Consulte a documentação do seu web server 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 *Use X-Frame-Options HTTP header* 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 carregam conteúdo de URLs externas (ou seja, o widget de URL do dashboard) exibem o conteúdo recuperado em um sandbox com todas as restrições de sandbox habilitadas.

Essas configurações aumentam a segurança do frontend do Zabbix e fornecem proteção contra ataques de XSS e clickjacking. Usuários *Super admin* podem modificar os parâmetros *Use iframe sandboxing* e *Use X-Frame-Options HTTP header* conforme necessário. Avalie cuidadosamente os riscos e benefícios antes de alterar as configurações padrão. Não é recomendável 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 de ataques de força bruta de senha, é recomendável limitar o acesso ao arquivo `ui/data/top_passwords.tx`
Esse arquivo contém uma lista das senhas mais comuns e específicas de contexto e impede que os usuários definam essas senhas (se o parâmetro *Avoid easy-to-guess passwords* 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;
}
```

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 - Version 2, CyberArk Vault CV12.

Os segredos podem ser usados para recuperar:

- valores de macro 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 específica de um provedor de cofre, consulte:

- Configuração do HashiCorp
- Configuração do CyberArk

Armazenamento em 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 a 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 armazenados em cache de um vault, use a opção de linha de comando `secrets_reload`.

Para as credenciais do banco de dados do Zabbix frontend, o armazenamento em cache é desabilitado por padrão, mas pode ser habilitado definindo a opção `$DB['VAULT_CACHE'] = true` no arquivo de configuração do frontend. As credenciais serão armazenadas em um cache local usando o diretório de arquivos temporários do sistema de arquivos. O web server deve permitir a 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/invalidado, use a constante `ZBX_DATA_CACHE_TTL`.

Configuração de TLS Para configurar TLS para a comunicação entre os componentes do Zabbix e o vault, adicione um certificado assinado por uma autoridade certificadora (CA) ao armazenamento 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, em seguida, execute o command da CLI:

```
c_rehash .
```

Configuração do CyberArk

Esta seção explica como configurar o Zabbix para recuperar secrets do CyberArk Vault CV12.

O vault deve ser instalado e configurado conforme descrito na documentação oficial da CyberArk.

Para saber mais sobre a configuração de TLS no Zabbix, consulte *Storage of secrets*.

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 as credenciais do banco de dados do cofre para o server ou proxy do Zabbix, especifique os seguintes parâmetros de configuração no arquivo de configuração:

- `Vault` - qual provedor de cofre deve ser usado;
- `VaultURL` - URL HTTP(S) do server do cofre;
- `VaultDBPath` - consulta ao segredo do cofre que contém as credenciais do banco de dados, que serão recuperadas pelas chaves "Content" e "UserName" (essa opção só pode ser usada se DBUser e DBPassword não forem especificados);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - nomes dos arquivos do certificado e da chave SSL; a configuração dessas opções não é obrigatória, mas é altamente recomendada;
- `VaultPrefix` - prefixo personalizado para o caminho ou a consulta do cofre, dependendo do cofre; se não for especificado, será usado o padrão mais adequado.

> **Attention:**
> Os parâmetros de configuração `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` e `VaultPrefix` também são usados para a autenticação no cofre ao processar macros de segredo do cofre pelo server do Zabbix (e pelo proxy do Zabbix, se configurado). O server e os proxies do Zabbix não abrirão macros de segredo do cofre que contenham credenciais do banco de dados de VaultDBPath.

O server e o proxy do Zabbix leem os parâmetros de configuração relacionados ao cofre dos arquivos `zabbix_server.conf` e `zabbix_proxy.conf` durante a 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 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 as chaves ”Content” e ”UserName”:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

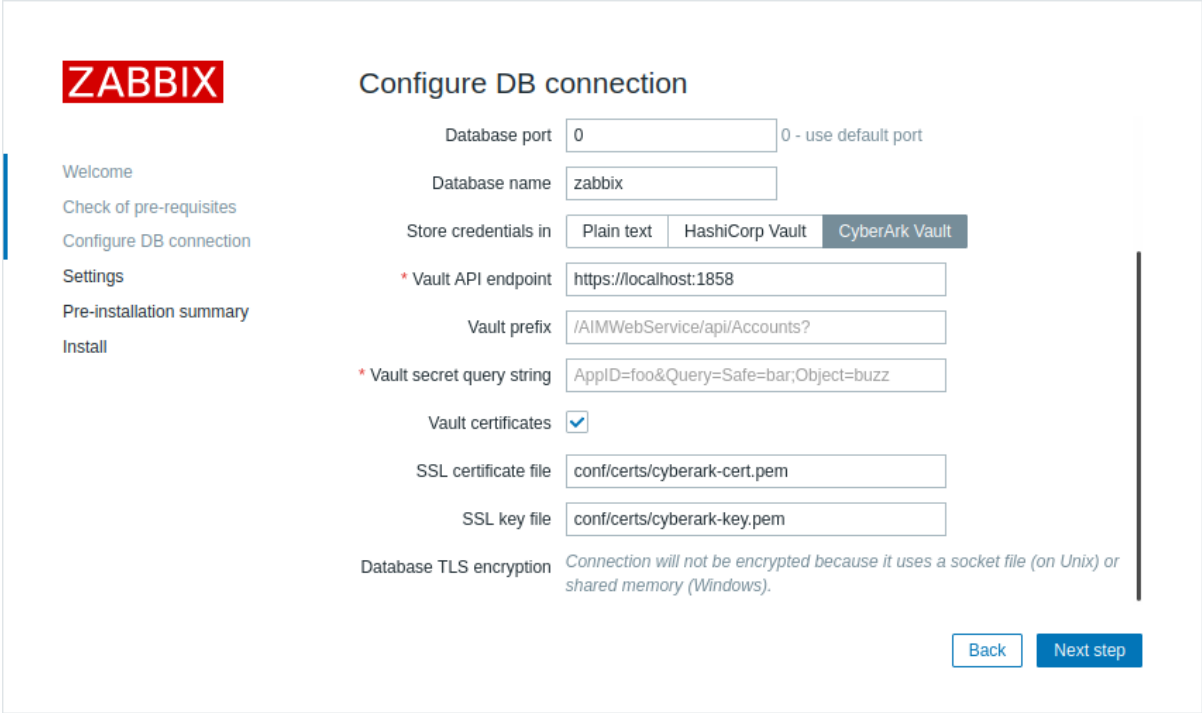
4. Como resultado, o Zabbix usará as seguintes credenciais para autenticação no banco de dados:

- Username: <username>
- Password: <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 *Configure DB Connection*, defina o parâmetro *Store credentials in* como *CyberArk Vault*.



2. Em seguida, preencha os parâmetros adicionais:

Parâmetro	Obrigatório	Valor padrão	Descrição
Vault API endpoint	sim	https://localhost:1858	Especifique a URL para conexão com o vault no formato `scheme://host:port`

Parâmetro	Obrigatório	Valor padrão	Descrição
Vault prefix	não	/AIMWebService/api/Accounts?	Forneça um prefixo personalizado para o caminho ou a consulta do vault. Se não for especificado, o valor padrão será usado.
Vault secret query string	sim		Uma consulta que especifica de onde as credenciais do banco de dados devem ser recuperadas. Exemplo: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	não		Após marcar a caixa de seleção, parâmetros adicionais serão exibidos, permitindo configurar a autenticação do cliente. Embora esse parâmetro seja opcional, é altamente recomendável habilitá-lo para a comunicação com o CyberArk Vault.
SSL certificate file	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 SSL key file vazio.
SSL key file	não	conf/certs/cyberark-key.pem	Nome do arquivo de chave privada SSL usado para a 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 esteja configurado para funcionar com o CyberArk Vault;
- o parâmetro *Vault provider* em *Administration > General > Other* esteja definido como "CyberArk Vault".



Note:
O Zabbix server (e o Zabbix proxy, se configurado) requer acesso aos valores de macro *Vault secret* no vault. O Zabbix frontend não precisa desse acesso.

O valor da macro deve conter uma consulta (como `query:key`).

Consulte *Vault secret macros* para obter informações detalhadas sobre o processamento do valor da macro pelo Zabbix.

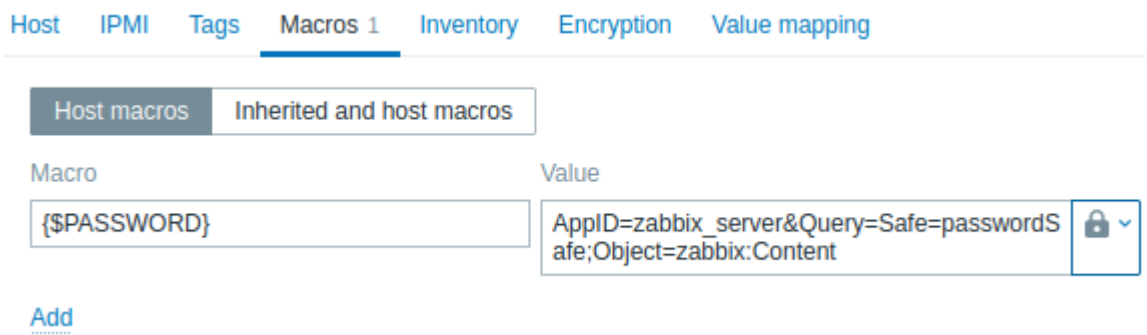
Sintaxe da consulta

O símbolo de dois-pontos (`:`) é reservado para separar a consulta da chave.

Se a própria consulta contiver uma barra (`/`) ou dois-pontos (`:`), esses símbolos deverão ser codificados na 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 ao vault:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. A resposta do vault conterá a chave `Content`:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Como resultado, o Zabbix resolverá a macro `{$PASSWORD}` para o valor - `<password>`

Atualizar a configuração existente

Para atualizar uma configuração existente para recuperar segredos de um CyberArk Vault:

1. Atualize os parâmetros do arquivo de configuração do Zabbix server ou proxy conforme descrito na seção *Credenciais do banco de dados*.
2. Atualize os parâmetros do arquivo de configuração do frontend do Zabbix:

```
$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 as macros de usuário conforme descrito na seção *Valores de macros de usuário*, se necessário.

Para atualizar uma configuração existente para recuperar segredos de um HashiCorp Vault, consulte *Configuração do HashiCorp*.

Configuração do HashiCorp

Visão geral

Esta seção explica como configurar o Zabbix para recuperar segredos do HashiCorp Vault KV Secrets Engine - Version 2.

O vault deve ser implantado e configurado conforme descrito na documentação oficial da HashiCorp.

Para saber mais sobre como configurar TLS no Zabbix, consulte *Storage of secrets*.

Recuperando credenciais do banco de dados

Para recuperar com êxito um segredo com credenciais do banco de dados, é necessário configurar ambos:

- Zabbix server/proxy
- Zabbix frontend

Server/proxy

Para configurar o server ou o proxy do Zabbix, 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 obter detalhes); não pode ser usado com `VaultAppRoleID`.
- `VaultAppRoleID` e `VaultAppSecretID` - ID da função e ID secreto de autenticação do vault (consulte o arquivo de configuração do Zabbix server/proxy para obter detalhes); não podem ser usados com `VaultToken`.
- `VaultURL` - URL HTTP(S) do server do vault.
- `VaultDBPath` - caminho para o segredo do vault que contém as credenciais do banco de dados (esta opção só pode ser usada se DBUser e DBPassword não forem especificados). O Zabbix server ou proxy recuperará as credenciais pelas chaves ”password” e ”username”.
- `VaultPrefix` - prefixo personalizado para o caminho ou a consulta do vault, dependendo do vault; se não for especificado, será usado o padrão mais adequado. Observe que o prefixo deve começar e terminar com `/`.

Attention:
Os parâmetros de configuração `Vault`, `VaultToken` (ou `VaultAppRoleID`/`VaultAppSecretID`), `VaultURL` e `VaultPrefix` também são usados para a autenticação no vault ao processar macros secretas do vault pelo Zabbix server (e pelo Zabbix proxy, se configurado). O Zabbix server e os proxies não abrirão macros secretas do vault que contenham credenciais do banco de dados de VaultDBPath.

É altamente recomendável usar tokens diferentes para proxies diferentes.

O Zabbix server e o Zabbix proxy leem os parâmetros de configuração relacionados ao vault de `zabbix_server.conf` e `zabbix_proxy.conf` durante a inicialização. Além disso, o Zabbix server e o Zabbix proxy leem a variável de ambiente `VAULT_TOKEN` uma vez durante a inicialização e a removem para que ela não fique disponível por meio de scripts bifurcados; é um erro se os parâmetros `VaultToken` e `VAULT_TOKEN` contiverem um valor.

Exemplo

1. Em `zabbix_server.conf`, especifique os seguintes parâmetros.

Usando VaultToken:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Ou, usando VaultAppRoleID:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. 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 o camin
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 server Zabbix recuperará as seguintes credenciais para autenticação no banco de dados:

- Username: zabbix
- Password: <password>

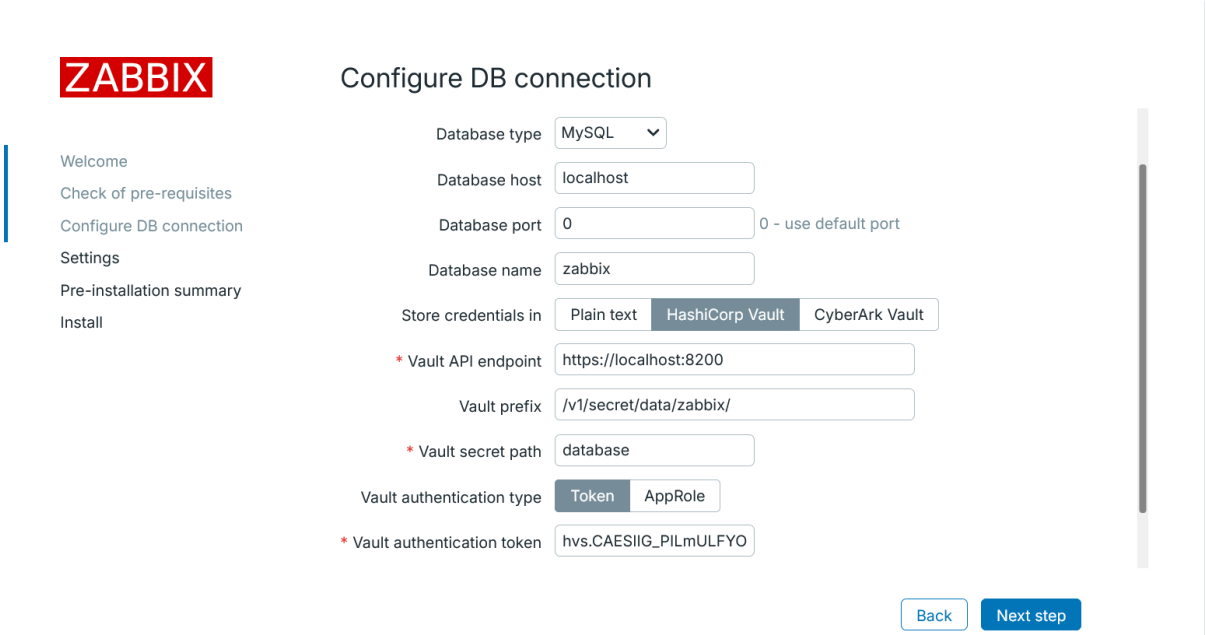
frontend

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`).

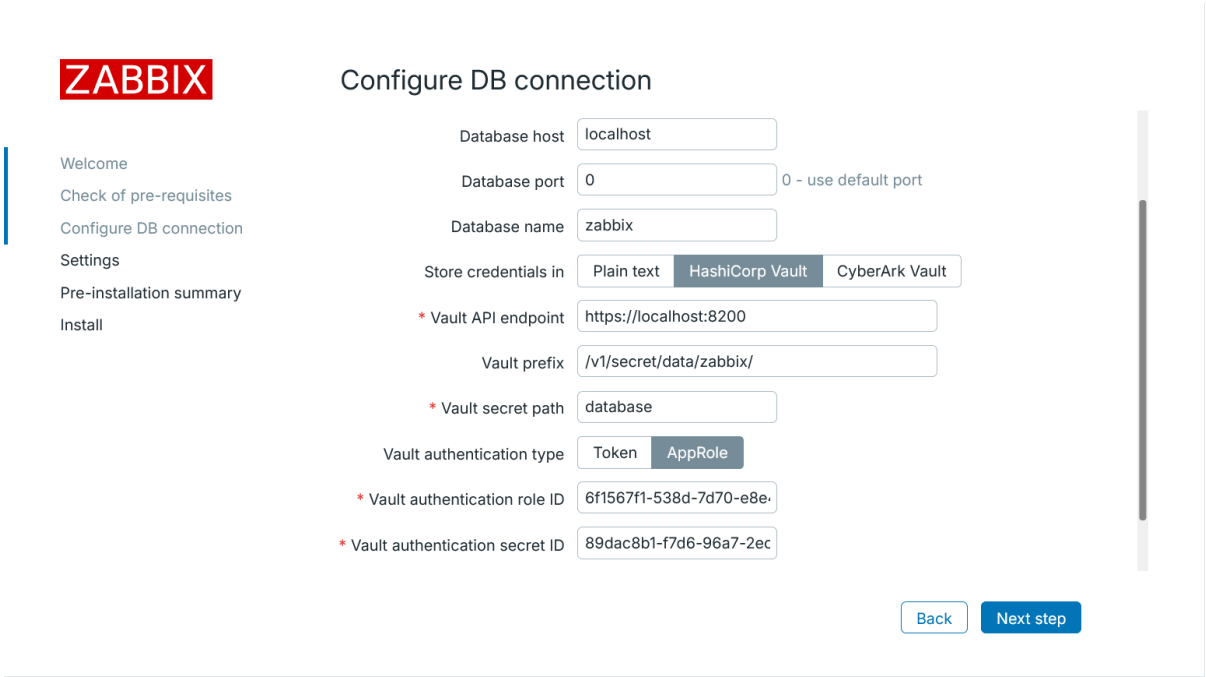
Attention:
Se as credenciais do vault tiverem sido alteradas desde a instalação anterior do frontend, atualize a configuração existente.

Durante a **instalação do frontend**, os parâmetros de conexão devem ser especificados na etapa *Configure DB Connection*.

Exemplo de configuração usando autenticação por token:



Exemplo de configuração usando autenticação AppRole:



Parâmetro	Obrigatório	Valor padrão	Descrição
Store credentials in	não		Selecione **HashiCorp Vault**.
Vault API endpoint	sim	https://localhost:8200	Especifique a URL para conexão com o vault no formato `scheme://host:port`
Vault prefix	não	/v1/secret/data/	Forneça um prefixo personalizado para o caminho ou a consulta do vault. Se não for especificado, o valor padrão será usado. Exemplo: `/v1/secret/data/zabbix/`
Vault secret path	sim		Um caminho para o segredo do qual as credenciais do banco de dados devem ser recuperadas por meio das chaves ”password” e ”username”. Exemplo: `database`
Vault authentication type	não		Selecione o tipo de autenticação do vault: **Token** - usar autenticação por token; **AppRole** - usar o método de autenticação AppRole.

Parâmetro	Obrigatório	Valor padrão	Descrição
Vault authentication token	sim		Forneça um token de autenticação com acesso somente leitura ao caminho do segredo. Consulte a documentação da HashiCorp para obter informações sobre como criar tokens e políticas do vault. Este campo só estará disponível se *Token* estiver selecionado como tipo de autenticação do vault.
Vault authentication role ID	sim		Forneça o ID da função do AppRole. Consulte a documentação da HashiCorp para obter informações sobre AppRoles. Este campo só estará disponível se *AppRole* estiver selecionado como tipo de autenticação do vault.
Vault authentication secret ID	sim		Forneça o ID secreto emitido para o AppRole. Consulte a documentação da HashiCorp para obter informações sobre AppRoles. Este campo só estará disponível se *AppRole* estiver selecionado como tipo de autenticação do vault.

Recuperando valores de macro de usuário

Para usar o HashiCorp Vault para armazenar valores de macro de usuário *Vault secret*, certifique-se de que:

- O Zabbix server/proxy esteja configurado para funcionar com o HashiCorp Vault.
- O parâmetro *Vault provider* em *Administration > General > Other* esteja definido como "HashiCorp Vault" (padrão).



Note:
O Zabbix server (e o Zabbix proxy, se configurado) requer acesso aos valores de macro *Vault secret* no 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.

Consulte *Vault secret macros* para obter informações detalhadas sobre o processamento do valor da macro pelo Zabbix.

Sintaxe do caminho

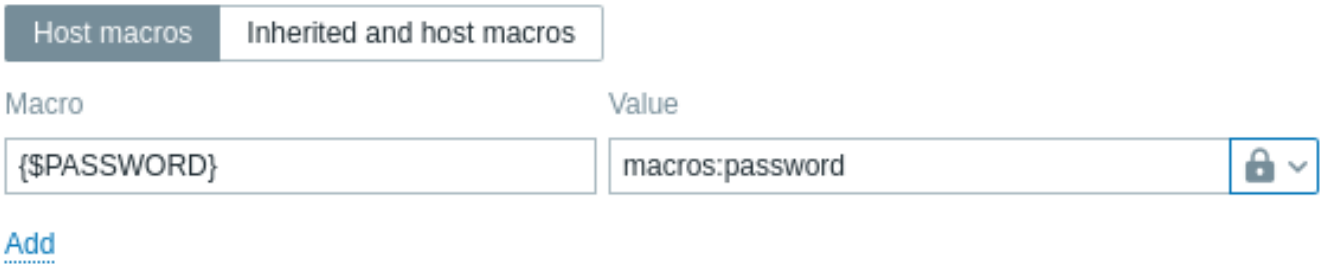
Os símbolos barra ("/") e dois-pontos (":") são reservados.

Uma barra só pode ser usada para separar um ponto de montagem de um caminho (por exemplo, *secret/zabbix*, em que o ponto de montagem é "secret" e o caminho é "zabbix"). No caso de macros do Vault, um dois-pontos só pode ser usado para separar um caminho/consulta de uma chave.

É possível codificar os símbolos barra e dois-pontos em URL se houver necessidade de criar um ponto de montagem com um nome que contenha uma barra (por exemplo, *foo/bar/zabbix*, em que o ponto de montagem é "foo/bar" e o caminho é "zabbix", pode ser codificado como "foo%2Fbar/zabbix") e se o nome de um ponto de montagem ou caminho precisar conter um 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 secret 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 secret com a chave "password" sob o ponto de montagem "secret/" e o caminho "zabbix/
vault kv put -mount=secret zabbix/macros password=<password>

#### Teste se o secret foi adicionado com sucesso.
vault kv get secret/zabbix/macros

#### 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/macros
```

3. Como resultado, o Zabbix resolverá a macro {$PASSWORD} para o valor: <password>

Atualização da 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 os parâmetros do arquivo de configuração do frontend do Zabbix:

Usando um token do Vault:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Ou usando o método AppRole do Vault:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. Configure as macros de usuário conforme descrito na seção *Valores de macros de usuário*, se necessário.

Para atualizar uma configuração existente para recuperar segredos de um CyberArk Vault, consulte *Configuração do CyberArk*.

Solução de problemas

1. No arquivo de log do Zabbix server/proxy:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Causa provável: o HashiCorp Vault foi (re)iniciado, mas não foi desbloqueado (necessário na configuração padrão, não necessário se Auto Unseal estiver configurado).

2. No arquivo de log do Zabbix server/proxy:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

ou

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Causa provável: VaultAppSecretID expirou ou atingiu seu limite de uso no HashiCorp vault.

Confiança no certificado TLS do Google Chrome

Visão geral

Esta página fornece etapas de configuração do Zabbix e exemplos de configuração para usar o Google Chrome com o frontend do Zabbix ou endpoints do zabbix-web-service protegidos por um certificado autoassinado ou por uma autoridade certificadora privada.

Estas instruções pressupõem que o servidor web de destino já esteja configurado para HTTPS. Para configurar TLS no frontend do Zabbix, consulte Secure connection to the frontend.

Configuração

O Google Chrome no Linux usa um banco de dados de certificados NSS por usuário para certificados confiáveis. Para fazer com que o Chrome confie em um certificado autoassinado ou em uma autoridade certificadora privada, adicione os certificados necessários ao banco de dados NSS da conta de usuário que executa o Chrome.

1. Instale os pacotes necessários.

Para Debian/Ubuntu:

```
sudo apt install ca-certificates libnss3-tools
```

Para sistemas baseados em RHEL:

```
sudo dnf install ca-certificates nss-tools
```

Ou:

```
sudo yum install ca-certificates nss-tools
```

2. Prepare os arquivos de certificado.

Para cada certificado necessário: crie um arquivo `.crt` baseado em PEM para o certificado autoassinado, ou crie arquivos `.crt` separados baseados em PEM para o certificado da autoridade certificadora raiz e, se necessário, para cada certificado intermediário.

3. Crie o diretório do banco de dados NSS.

Use o diretório home do usuário que executa o Chrome.

Por exemplo:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Se o Chrome no sistema de destino usar um local diferente para o banco de dados NSS, use esse diretório em vez disso.

Certifique-se de que os arquivos de certificado possam ser lidos pelo usuário que executa o Google Chrome. Se necessário, copie-os para um local acessível a esse usuário.

4. Importe os certificados.

Se estiver usando um certificado autoassinado:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

Se estiver usando uma autoridade certificadora privada, importe o certificado da autoridade certificadora raiz para o banco de dados:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

Se forem usados certificados intermediários, importe cada um separadamente:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. Verifique o banco de dados de certificados.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Teste o Chrome.

O Google Chrome usa o banco de dados de certificados da conta de usuário que o inicia. Se a conta de usuário não tiver um diretório home gravável, configure um home gravável e diretórios XDG antes de executar o teste.

Exemplo (substitua `google-chrome` por `chromium` se o Chromium for usado.):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Se a configuração estiver correta, o comando retornará o HTML da página solicitada.

Se o Chrome não confiar no certificado, o comando retornará uma página de erro de certificado contendo mensagens como `NET::ERR_CERT_AUTHORITY_INVALID`.

Solução de problemas

Error	Possible cause
`NET::ERR_CERT_AUTHORITY_INVALID`	O certificado foi importado para o perfil de usuário errado, o certificado da autoridade certificadora raiz está ausente ou um certificado intermediário está ausente.
`NET::ERR_CERT_COMMON_NAME_INVALID`	O certificado não corresponde ao nome do host usado na URL do navegador.
`NET::ERR_CERT_DATE_INVALID`	O certificado expirou, ainda não é válido ou o relógio do sistema está incorreto.
Chrome ainda exibe um aviso de certificado após a importação	O Chrome está usando um banco de dados NSS diferente para o usuário atual, ou os sinalizadores de confiança do certificado estão incorretos.

Práticas recomendadas de segurança

Visão geral Esta seção contém as melhores práticas para configurar o Zabbix de forma segura.

As práticas desta seção não são obrigatórias para o funcionamento do Zabbix, mas são recomendadas para melhorar a segurança do sistema.

Para conhecer as melhores práticas para manter o aplicativo móvel do Zabbix e o adaptador Zabbix bridge seguros, consulte Acesso móvel seguro.

Codificação UTF-8 UTF-8 é a única codificação suportada pelo Zabbix. Sabe-se que ela funciona sem falhas de segurança. Os usuários devem estar cientes de que há problemas de segurança conhecidos ao usar algumas das outras codificações.

Caminhos do instalador do Windows Ao usar instaladores do Windows, é recomendável usar os caminhos padrão fornecidos pelo instalador. Usar caminhos personalizados sem as permissões adequadas pode comprometer a segurança da instalação.

Macros em scripts globais definidos pelo usuário Para aumentar a segurança, é recomendável usar funções de macro em vez de macros simples em scripts globais definidos pelo usuário, pois as macros não são escapadas automaticamente.

Avisos de Segurança do Zabbix e banco de dados CVE Veja *Avisos de Segurança do Zabbix e banco de dados CVE*.

Templates de e-mail HTML Ao criar ou editar templates de mensagem usados para e-mails HTML, sempre envolva cada macro com a função de macro `htmlencode()`. Por exemplo:

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

O uso de `htmlencode()` garante que todos os caracteres HTML nos valores das macros sejam escapados e impede a injeção de HTML nas notificações (por exemplo, que um invasor insira um link malicioso ou de phishing em uma notificação).

Observação: as mensagens de e-mail HTML padrão fornecidas pelo Zabbix já aplicam `htmlencode()` às macros. Esta orientação se aplica ao editar templates existentes ou criar novos: verifique se as macros estão codificadas antes de usar um template para enviar e-mails HTML.

Interface web

Visão geral Para facilitar o acesso ao Zabbix de qualquer lugar e em qualquer plataforma, é fornecida uma interface baseada na web.

> **Note:**
> Se estiver usando mais de uma instância de frontend, certifique-se de que as localidades e as bibliotecas (LDAP, SAML etc.) estejam instaladas e configuradas de forma idêntica em todos os frontends.

Ajuda do frontend Um link de ajuda ? é disponibilizado nas seções e nos formulários do frontend do Zabbix, no canto superior direito, direcionando diretamente para as partes correspondentes da documentação.

Para campos de formulário com um ícone de ajuda, passar o cursor sobre o ícone exibe a dica correspondente.

Modo quiosque Nas seções do frontend que possuem o botão *Modo quiosque* (localizado no canto superior direito), clique no botão para alternar a seção para o modo quiosque. No modo quiosque, somente o conteúdo da seção é exibido.

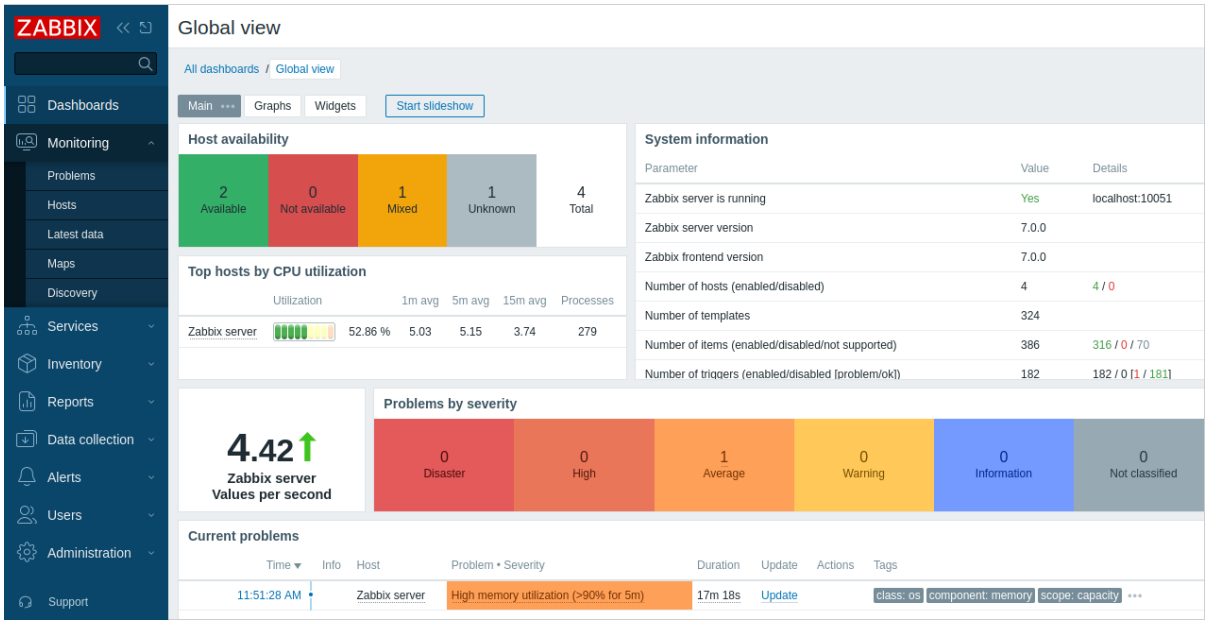
Para sair do modo quiosque, passe o cursor sobre o conteúdo e clique no botão *Visualização normal* que aparece no canto superior direito.

Menu

Visão geral

Um menu vertical em uma barra lateral fornece acesso a várias seções do frontend do Zabbix.

O menu é azul escuro no tema padrão.

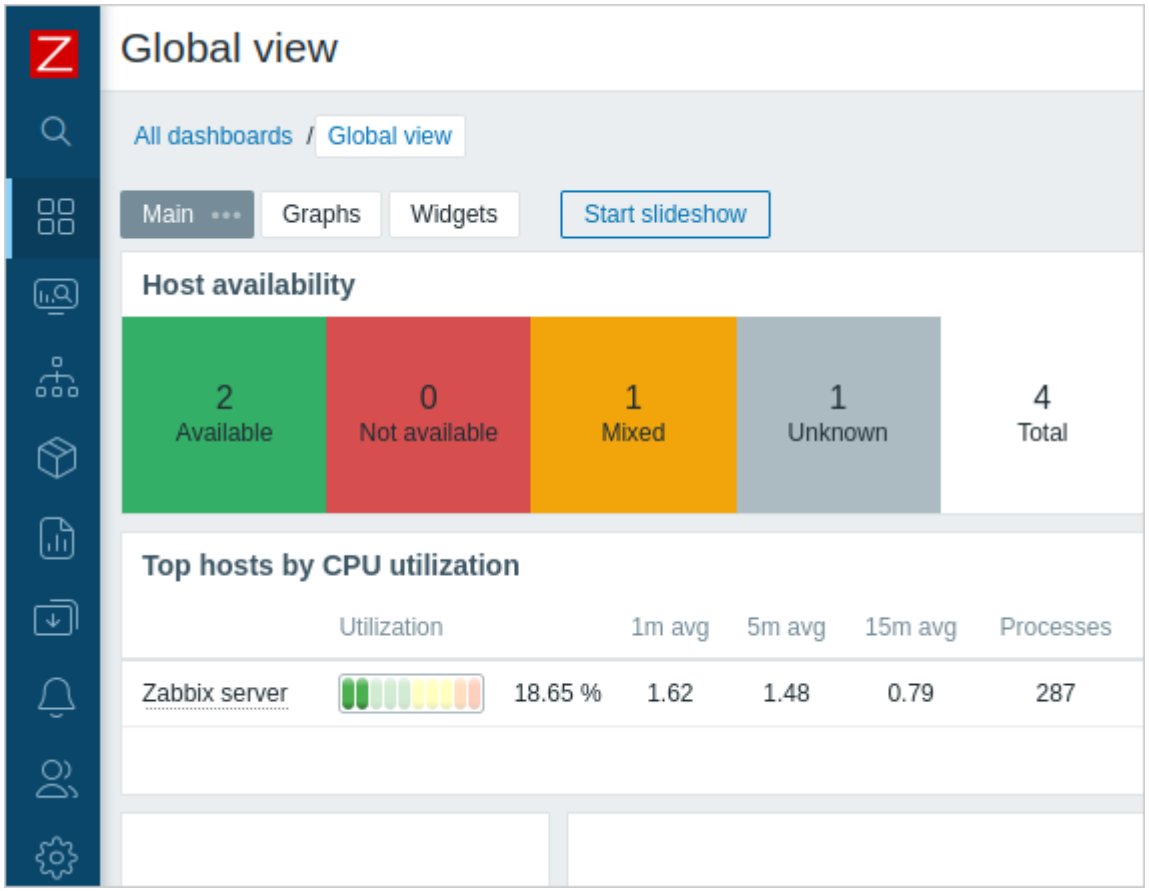


Trabalhando com o menu

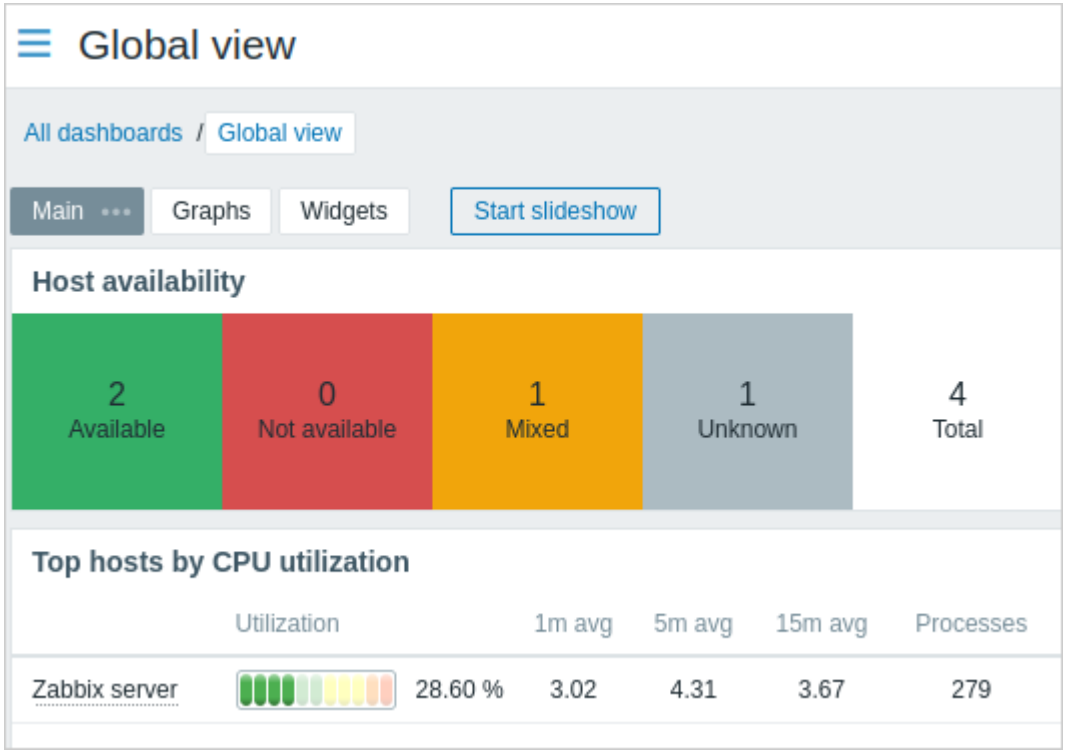
Uma caixa de pesquisa global está localizada abaixo do logotipo do Zabbix.

O menu pode ser recolhido ou ocultado completamente:

- Para recolher, clique em  ao lado do logotipo do Zabbix. No menu recolhido, apenas os ícones ficam visíveis.



- Para ocultar, clique em  ao lado do logotipo do Zabbix. No menu oculto, tudo fica escondido.



Menu recolhido

Quando o menu é recolhido, exibindo apenas ícones, o menu completo reaparece assim que o cursor do mouse é posicionado sobre ele. Observe que ele reaparece sobre o conteúdo da página; para mover o conteúdo da página para a direita, é necessário clicar no botão de expansão. Se o cursor do mouse for novamente posicionado fora do menu completo, o menu será recolhido outra vez após dois segundos.

Você também pode fazer o menu recolhido reaparecer completamente pressionando a tecla Tab. Pressionar a tecla Tab repetidamente permite focar no próximo elemento do menu.

Menu oculto

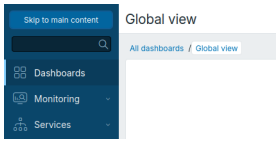
Mesmo quando o menu está completamente oculto, um menu completo está a apenas um clique do mouse de distância, ao clicar no ícone de hambúrguer . Observe que ele reaparece sobre o conteúdo da página; para mover o conteúdo da página para a direita, você precisa exibir o menu novamente clicando no botão mostrar barra lateral.

Níveis de menu

Existem até três níveis no menu.

Acessibilidade: Pular navegação

O link *Pular para o conteúdo principal* permite que usuários de teclado e leitores de tela ignorem a navegação repetitiva da barra lateral e vão diretamente para o conteúdo principal.



Ele permanece oculto durante o uso normal e aparece somente quando recebe foco, como quando a tecla Tab é pressionada após o carregamento da página ou quando é acessado usando um leitor de tela.

Menus de contexto

Além do menu principal, o Zabbix fornece menus de contexto para host, item e event para acesso rápido a entidades usadas com frequência, como os últimos valores, gráfico simples, formulário de configuração, scripts relacionados ou links externos.

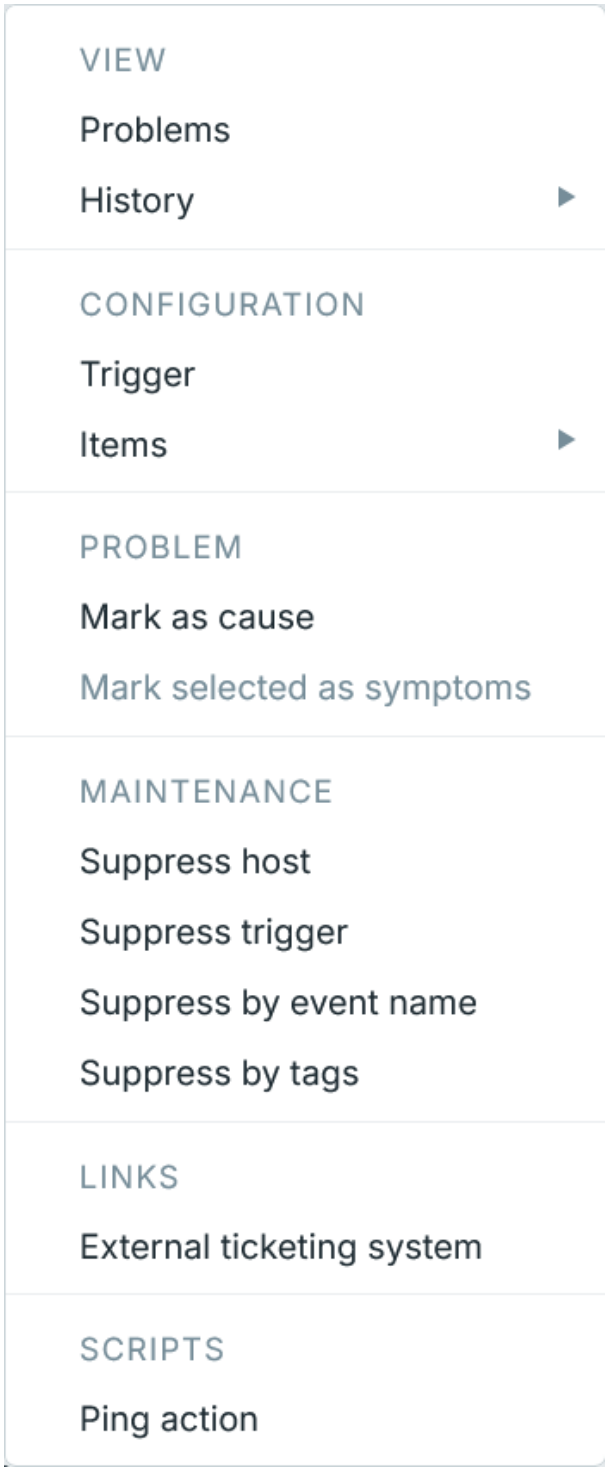
Os menus de contexto podem ser acessados clicando no nome do host, item ou problema/trigger em locais compatíveis.

Menu de eventos

Visão geral

O menu de eventos contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um evento.

O menu de eventos pode ser aberto clicando no nome do evento.



Conteúdo

O menu de contexto do evento contém várias seções: *Visualização*, *Ações*, *Configuração*, *Problema*, *Manutenção*, *Links* e *Scripts*. Para as entidades que não estão configuradas, os links ficam desabilitados e são exibidos em cinza. As seções *Scripts* e *Links* são exibidas se as respectivas entidades estiverem configuradas.

A seção *Visualização* contém links para:

- **Problemas** - abre a lista de problemas não resolvidos do trigger subjacente;
- **Histórico** - leva ao gráfico/histórico do item em *Dados mais recentes* para o(s) item(ns) subjacente(s). Se um trigger usar vários itens, haverá links disponíveis para cada um deles.

A seção *Ações* está disponível somente nos widgets *Visão geral dos triggers*. Ela contém um link para:

- **Atualizar problema** - abre a tela de atualização do problema.

A seção *Configuração* contém links para a configuração de:

- **Trigger** que gerou o problema;
- **Itens** usados na expressão do trigger.

> **Note:**
> Observe que a seção de configuração está disponível somente para usuários Admin e Super admin.

A seção *Problema* contém as opções para:

- **Marcar como causa** - marca o problema como causa;
- **Marcar selecionados como sintomas** - marca os problemas selecionados como sintomas deste problema.

A seção *Manutenção* contém opções para iniciar uma manutenção ad hoc:

- **Suprimir host** - inicia um período de manutenção para o host;
- **Suprimir trigger** - inicia um período de manutenção para o trigger;
- **Suprimir por nome do evento** - inicia um período de manutenção para o nome do evento;
- **Suprimir por tags** - inicia um período de manutenção para eventos com as tags correspondentes.

Ao selecionar a opção correspondente, um formulário de configuração do período de manutenção será aberto.

Um período de manutenção único de uma hora (configurável nas configurações e no perfil do usuário), com início a partir de agora, será pré-selecionado. O host/trigger/nome do evento/tags correspondente também será pré-selecionado. O nome da manutenção será preenchido previamente no formato `Ad-hoc: <nome do evento>`.

A seção *Links* contém links para:

- acessar uma URL do trigger configurada;
- acessar links personalizados configurados em Scripts globais (com escopo 'Ação manual do evento' e tipo 'URL');
- acessar um ticket externo configurado para o problema (consulte a opção *Incluir entrada no menu do evento* ao configurar webhooks).

A seção *Scripts* contém links para executar um script global (com escopo *Ação manual do evento*). Esse recurso pode ser útil para executar scripts usados no gerenciamento de tickets de problemas em sistemas externos.

Locais suportados

O menu de contexto do evento está acessível clicando no nome de um problema ou evento em várias seções do frontend, por exemplo:

- Dashboards widgets, como o widget *Problems*, o widget *Trigger overview*, etc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Detalhes do evento
- *Reports* > *Top 100 triggers* (scripts globais e acesso a ticket externo não são suportados neste local)

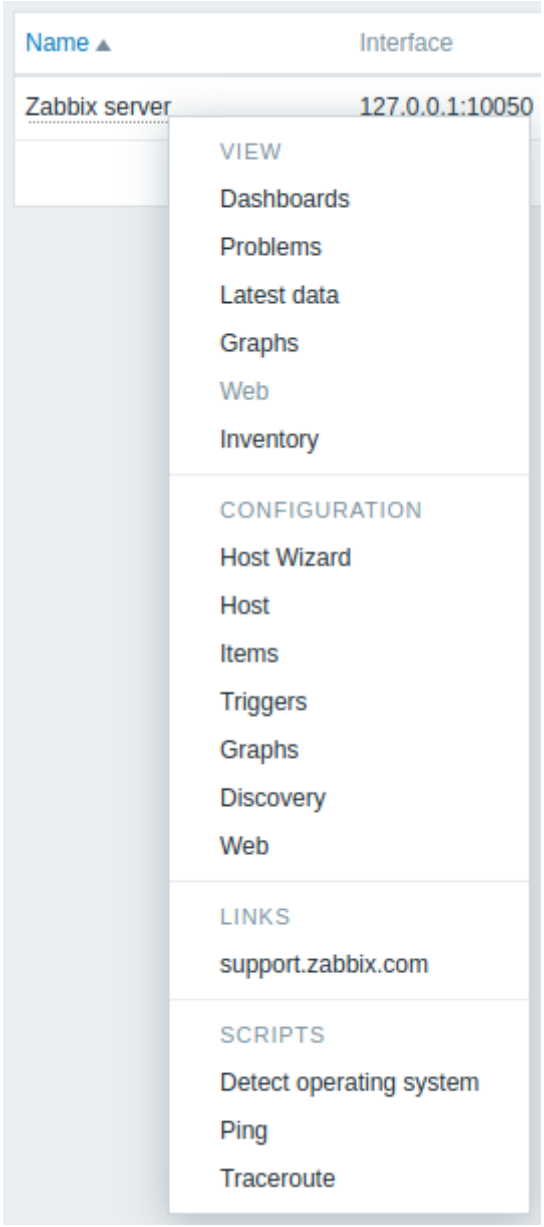
Menu do host

Visão geral

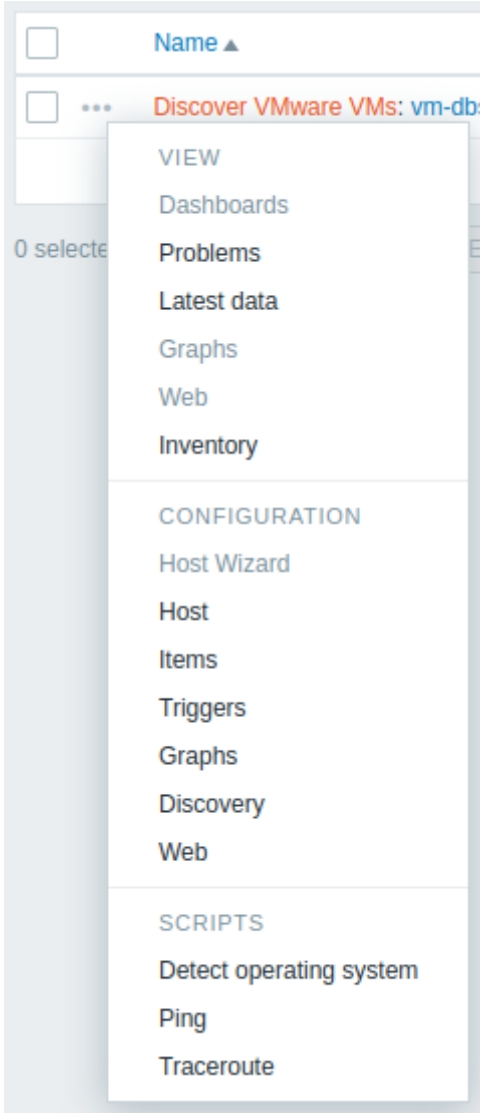
O menu do host contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um host.

O menu do host pode ser aberto clicando no nome do host ou no ícone de três pontos, dependendo da seção do frontend, por exemplo:

Monitoring > *Latest data*



Data collection > *Hosts*



Conteúdo

O menu de contexto do host tem quatro seções: *Visualização*, *Configuração*, *Links* e *Scripts*. Para as entidades que não estão configuradas, os links ficam desabilitados e são exibidos em cinza. As seções *Scripts* e *Links* são exibidas se as respectivas entidades estiverem configuradas.

A seção *Visualização* contém links para:

- **Dashboards** - abre widgets e gráficos.
- **Problemas** - abre a seção *Problemas* com a lista de problemas não resolvidos do trigger subjacente.
- **Dados mais recentes** - abre a seção *Dados mais recentes* com a lista de todos os dados mais recentes 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 itens 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 somente 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 o escopo *Ação manual de host* e o tipo 'URL').

A seção *Scripts* permite a execução de scripts globais configurados para o host atual. Para que estejam disponíveis no menu do host, esses scripts precisam ter o escopo definido como *Ação manual de host*.

Locais suportados

O menu do host pode ser acessado clicando no nome de um host em várias seções do frontend, por exemplo:

- Dashboards widgets, como *Problems*, *Top items*, *Trigger overview*, etc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Detalhes do evento
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

O menu do host pode ser acessado clicando no ícone de três pontos em Data collection > Hosts.

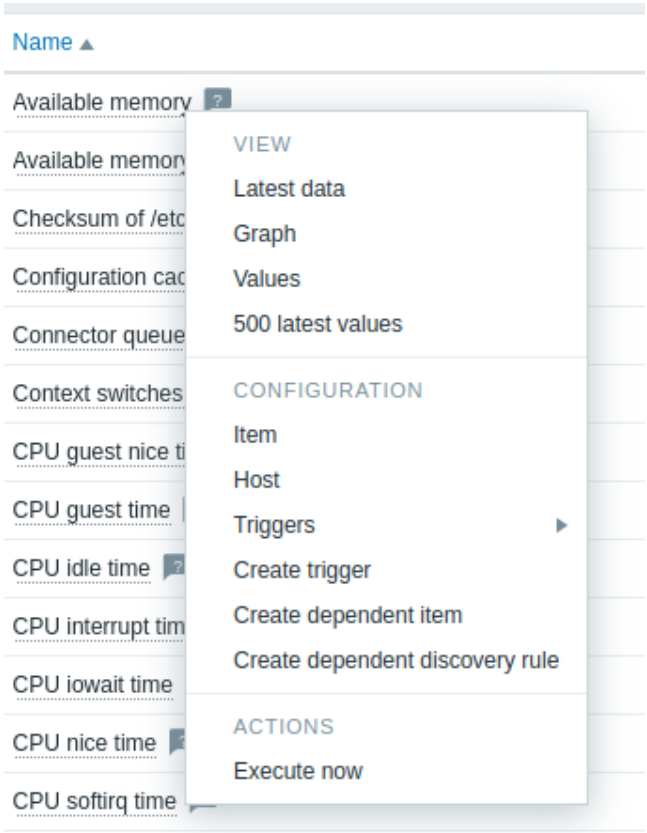
Menu do item

Visão geral

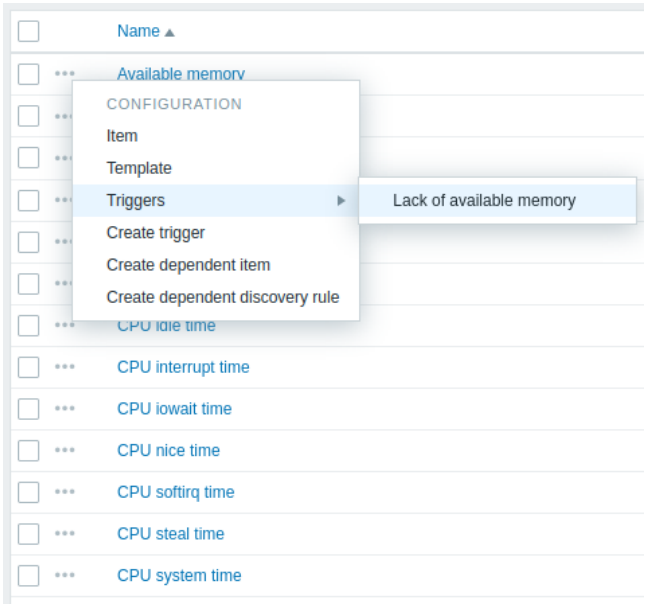
O menu do item contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um item.

O menu do item pode ser aberto clicando no nome do item ou no ícone de três pontos, dependendo da seção do frontend, por exemplo:

Monitoring > *Latest data*



Data Collection > *Templates* > *Items*



Conteúdo

O menu do item tem três seções: *Visualização*, *Configuração* e *Ações*.

A seção *Visualização* contém as seguintes opções:

- **Dados mais recentes** - abre a seção *Dados mais recentes* 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 valores mais recentes** - abre a lista dos 500 valores mais recentes do item atual.

A seção *Configuração* (disponível somente para usuários dos tipos *Admin* e *Super admin*) contém as seguintes opções:

- **Item** - abre o formulário de configuração do item atual;
- **Template** - abre o formulário de configuração do template ao qual o item pertence (disponível somente 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 dos triggers do item, se houver; clicar em um trigger abre seu formulário de configuração do trigger;
- **Criar trigger** - abre o formulário de configuração do trigger para criar um trigger para este item;
- **Criar item dependente** - abre o formulário de configuração do item para criar um item dependente, tendo 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, tendo o item atual como item mestre.

A seção *Ações* contém a seguinte opção:

- **Executar agora** - executa uma verificação imediatamente para obter um novo valor do item.

Locais suportados

O menu do item pode ser acessado clicando no nome de um item em várias seções do frontend, por exemplo:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items*
- *Data collection* > *Hosts* > *Discovery rules* > *Item prototypes*

O menu do item pode ser acessado clicando no valor de um item no widget de dashboard *Top items*.

Seções do frontend

Estrutura do menu O menu do frontend do Zabbix tem a seguinte estrutura:

- Painéis
- Monitoramento
  - Problemas
  - Hosts
  - Dados mais recentes
  - Mapas
  - Descoberta
- APM
  - Traces
  - Métricas
  - Logs
- Serviços
  - Serviços
  - SLA
  - Relatório de SLA
- Inventário
  - Visão geral
  - Hosts
- Relatórios
  - Informações do sistema
  - Relatórios agendados
  - Relatório de disponibilidade
  - 100 principais 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 autoregistro
    - Ações internas
  - Tipos de mídia
  - Scripts
- Usuários
  - Grupos de usuários
  - Funções de usuário
  - Usuários
  - Tokens de API
  - Autenticação
  - Dispositivos
- Administração
  - Geral
    - GUI
    - Autoregistro
    - Tempos limite
    - Imagens
    - Mapeamento de ícones

        * Expressões regulares
        * Opções de exibição de triggers
        * Mapas geográficos
        * Módulos
        * Conectores
        * Outros
    - Log de auditoria
    - Housekeeping
    - Grupos de proxies
    - Proxies
    - Macros
    - Fila
        * Visão geral da fila
        * Visão geral da fila por proxy
        * Detalhes da fila

Painéis

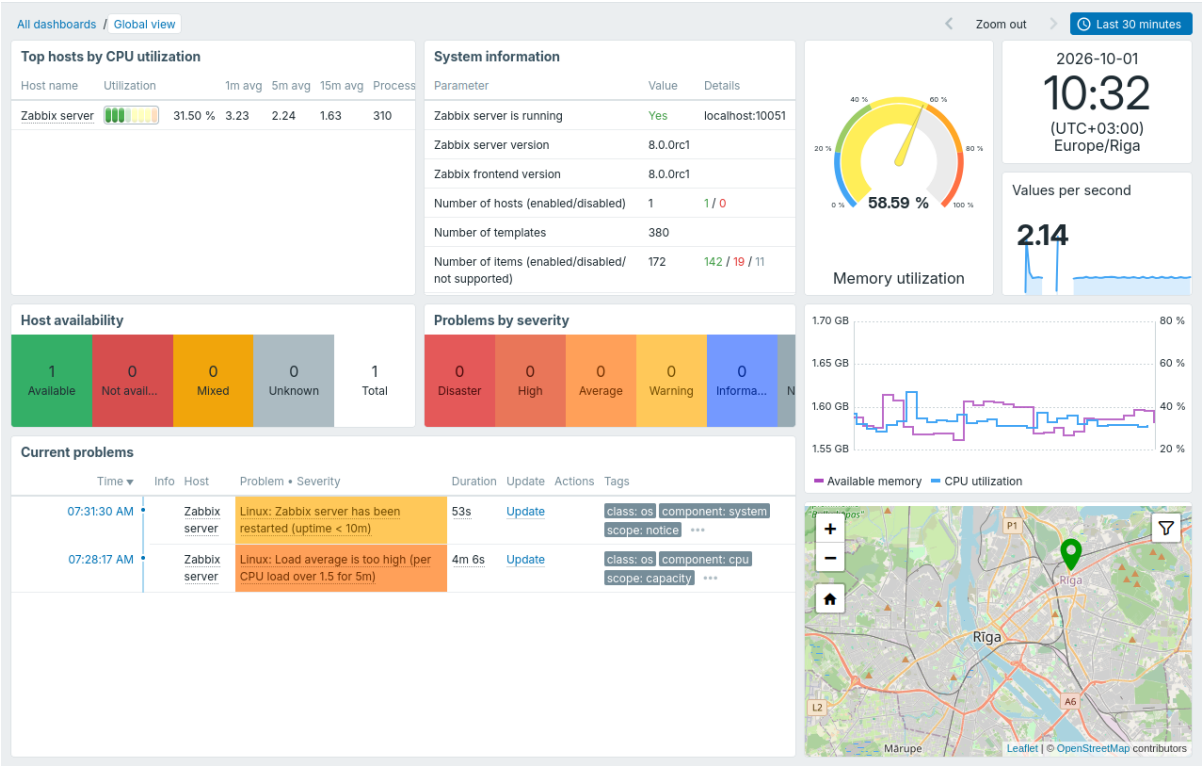
Os dashboards oferecem uma visão geral do seu ambiente monitorado. Você pode criá-los e personalizá-los de acordo com as suas necessidades.

Os blocos de construção são os widgets: um gráfico, um mapa, uma lista de problemas recentes e assim por diante, cada um projetado para exibir um tipo específico de informação.

Os widgets podem ser organizados em várias páginas do dashboard. Essas páginas podem ser visualizadas como uma apresentação de slides. Você também pode navegar para qualquer página clicando na respectiva aba.

Você pode criar vários dashboards, embora apenas um seja exibido por vez.

O Zabbix vem com alguns dashboards padrão. Um deles é o dashboard *Visão global*, que mostra a integridade e o status dos seus hosts monitorados, incluindo o Zabbix server:

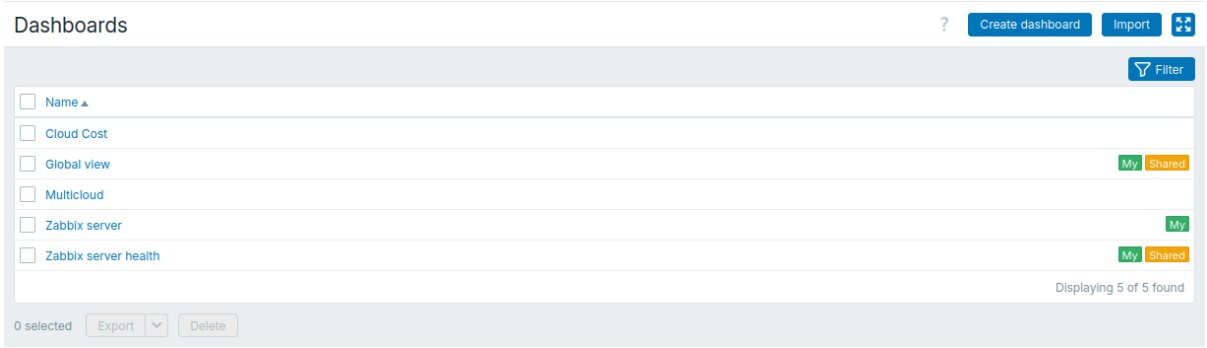


Você pode encontrar mais dashboards no repositório de dashboards. Esses dashboards foram projetados para funcionar com templates existentes do Zabbix e podem ser baixados e importados para o seu ambiente Zabbix. Uma descrição detalhada de cada dashboard, incluindo os pré-requisitos, está disponível no arquivo `README` do dashboard.

Lista de dashboards

Para abrir a lista de dashboards, você pode:

- Clicar em *Dashboards* no menu.
- Clicar em *Todos os dashboards* enquanto visualiza um dashboard para abrir a lista de dashboards.



O acesso ao menu *Dashboards* depende da função do usuário e da permissão *Dashboard* em *Acesso aos elementos da interface*.

A partir daqui, você pode: - Criar dashboards. - Excluir dashboards. Você pode excluir um dashboard selecionando sua caixa de seleção e clicando no botão *Excluir*. - Importar dashboards. - Exportar dashboards.

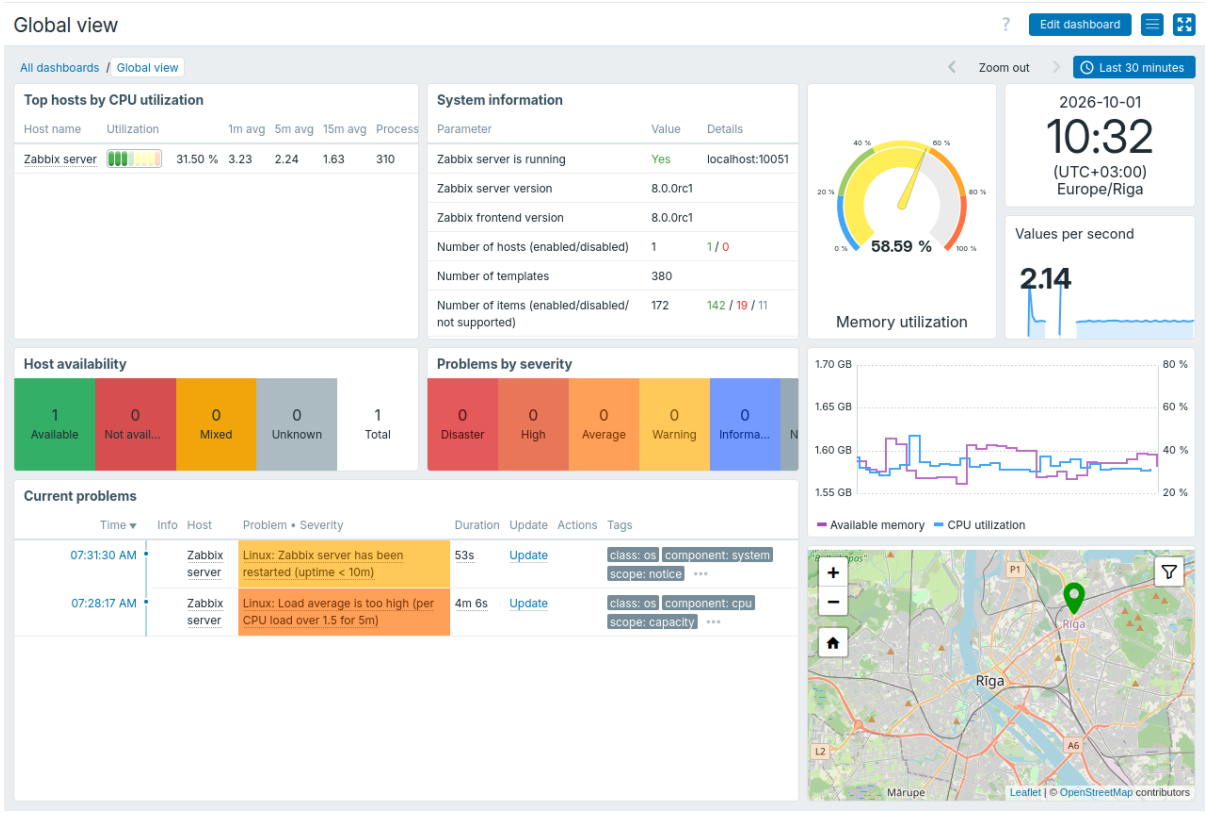
Os dashboards têm tags que indicam suas configurações de propriedade e compartilhamento:

- *Meus* - o dashboard pertence a você e está visível somente para você.
- *Meus + Compartilhado* - o dashboard pertence a você, mas é público ou compartilhado com usuários ou grupos de usuários específicos.
- Dashboards sem tags pertencem a outros usuários, mas são públicos ou compartilhados com você.

Você pode filtrar a lista pelo nome e pelo proprietário do dashboard.

Visualizar o dashboard

Clique no nome de um dashboard na lista de dashboards para visualizar o dashboard:



A interface de usuário do Zabbix memoriza o último dashboard aberto para a próxima sessão.

Note:
Os dados que você pode visualizar em um dashboard dependem do seu acesso aos hosts.

Para widgets que exibem gráficos, o seletor de período acima dos widgets define o intervalo de tempo. Ele mostra o intervalo de tempo atual como um rótulo. Clique no rótulo para abrir ou fechar o seletor.

A partir daqui, você pode gerenciar os dashboards da seguinte forma:

- Editar o dashboard.
- Executar as seguintes ações no menu *Actions* do dashboard :

  - *Sharing* - editar as configurações de compartilhamento do dashboard.
  - *Create new* - criar um novo dashboard.
  - *Clone* - criar um novo dashboard clonando o atual. Você pode editar as propriedades do dashboard antes de salvá-lo.
  - *Delete* - excluir o dashboard.
  - *Export* - exportar o dashboard.
  - *Create new report* - criar um relatório agendado. Essa opção fica desabilitada se você não tiver permissão para gerenciar relatórios agendados.
  - *View related reports* - visualizar uma lista de relatórios existentes para o dashboard atual. Essa opção fica desabilitada se não houver relatórios relacionados ou se você não tiver permissão para gerenciar relatórios agendados.
- Alternar para o modo quiosque.
- Iniciar ou interromper a apresentação de slides do dashboard.

No modo de visualização do dashboard, as seguintes opções estão disponíveis para os widgets:

- Editar o widget.
- Executar as seguintes ações no menu *Actions* do widget ⋮:
  - *Copy* - copiar o widget.
  - *Download image* - baixar o widget como uma imagem PNG. Essa opção está disponível para widgets que oferecem suporte a esse recurso (consulte as páginas individuais dos widgets).
  - *Refresh interval* - selecionar a frequência de atualização do conteúdo do widget.

Para dashboards com várias páginas, as seguintes ações estão disponíveis no menu *Actions* da página  no modo de visualização do dashboard:

- *Copy* - copiar a página.
- *Properties* - personalizar as propriedades da página, incluindo o nome e o intervalo da apresentação de slides.

Apresentação de slides

Se a caixa de seleção *Iniciar apresentação de slides* estiver selecionada nas propriedades do dashboard, a apresentação de slides será iniciada automaticamente quando o dashboard tiver mais de uma página. Caso contrário, você poderá iniciar a apresentação de slides clicando no botão *Iniciar apresentação de slides*.



As páginas mudam de acordo com os intervalos especificados nas propriedades do dashboard e das páginas individuais. Se a propriedade *Intervalo da apresentação de slides* estiver definida para uma página individual, o intervalo especificado para essa página será aplicado. Caso contrário, será aplicado o *Intervalo padrão da apresentação de slides* do dashboard.

Para interromper a apresentação de slides, clique em *Interromper apresentação de slides*.

No modo quiosque, use os seguintes botões:

-  para iniciar a apresentação de slides;
-  para interromper a apresentação de slides;
-  para navegar até a página anterior;
-  para navegar até a próxima página.

Compartilhar dashboard

Os dashboards podem ser públicos ou privados.

Os dashboards públicos ficam visíveis para todos os usuários.
Os dashboards privados ficam visíveis apenas para seus proprietários e para os usuários ou grupos de usuários com os quais são compartilhados.

Para alterar as configurações de compartilhamento de um dashboard, clique em *Compartilhamento* no menu *Ações* do dashboard enquanto visualiza o dashboard:

Parâmetro	Descrição
Tipo	Selecione o tipo de dashboard: **Privado** - o dashboard fica visível apenas para os grupos de usuários e usuários selecionados. **Público** - o dashboard fica visível para todos os usuários.

Parâmetro	Descrição
Lista de compartilhamentos com grupos de usuários	Selecione os grupos de usuários que podem acessar o dashboard. Você pode conceder acesso somente leitura ou leitura e gravação.
Lista de compartilhamentos com usuários	Selecione os usuários que podem acessar o dashboard. Você pode conceder acesso somente leitura ou leitura e gravação.

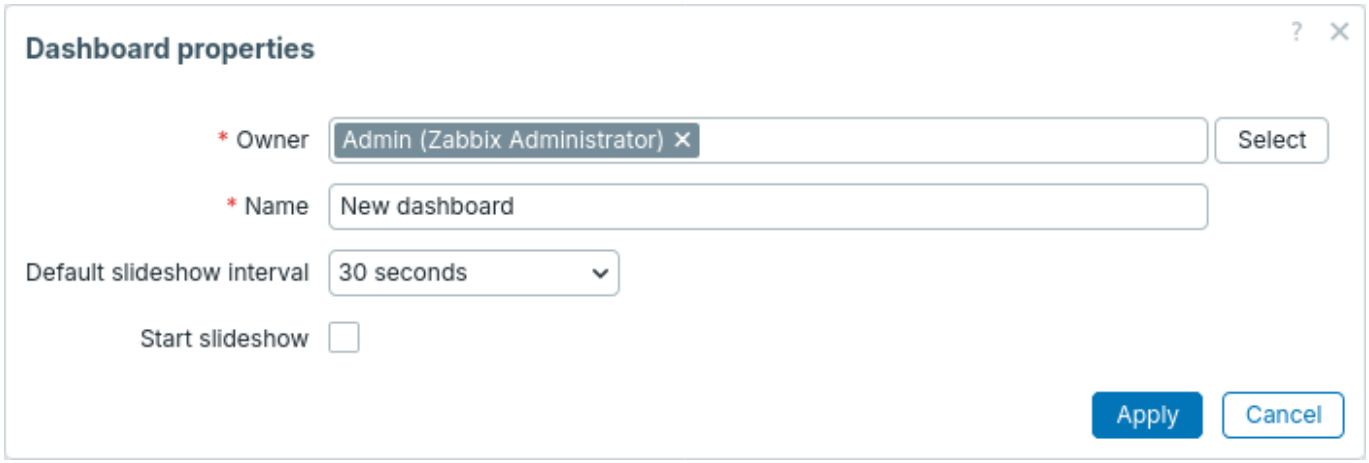
Consulte Permissões para dashboards para saber como as opções de compartilhamento limitam as ações disponíveis para os usuários.

Criar dashboard

Você pode criar um dashboard de duas maneiras:

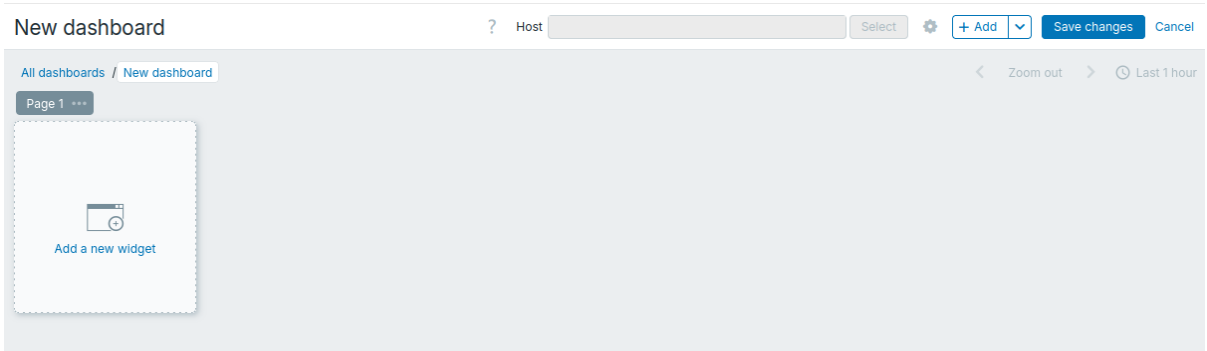
- Na lista de dashboards, clique no botão *Criar dashboard*.
- No modo de visualização do dashboard, clique em *Criar novo* no menu *Ações* do dashboard.

Em seguida, especifique as propriedades do dashboard:



Parâmetro	Descrição
Proprietário	Selecione o usuário que será o proprietário do dashboard.
Nome	Insira o nome do dashboard.
Intervalo padrão da apresentação de slides	Selecione por quanto tempo cada página do dashboard será exibida antes que a apresentação de slides avance para a próxima página.
Iniciar apresentação de slides	Marque esta caixa de seleção para iniciar a apresentação de slides automaticamente quando o dashboard tiver mais de uma página.

Em seguida, clique em *Aplicar*. O novo dashboard será aberto no modo de edição:



A partir daqui, você pode adicionar widgets e páginas ao dashboard.

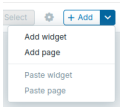
Observe que a largura mínima de um dashboard é de 1200 pixels; em janelas mais estreitas, uma barra de rolagem horizontal será exibida. Um dashboard mais largo sempre preenche a janela do navegador; seus widgets não podem ultrapassar as bordas da janela.

Clique no botão *Salvar alterações* para salvar o dashboard ou em *Cancelar* para retornar à tela anterior.

Adicionar widgets

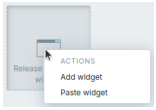
No modo de edição do dashboard, você pode adicionar widgets de duas maneiras:

- Clique no botão *Adicionar*. Como alternativa, clique na seta suspensa do botão *Adicionar* e selecione *Adicionar widget*.



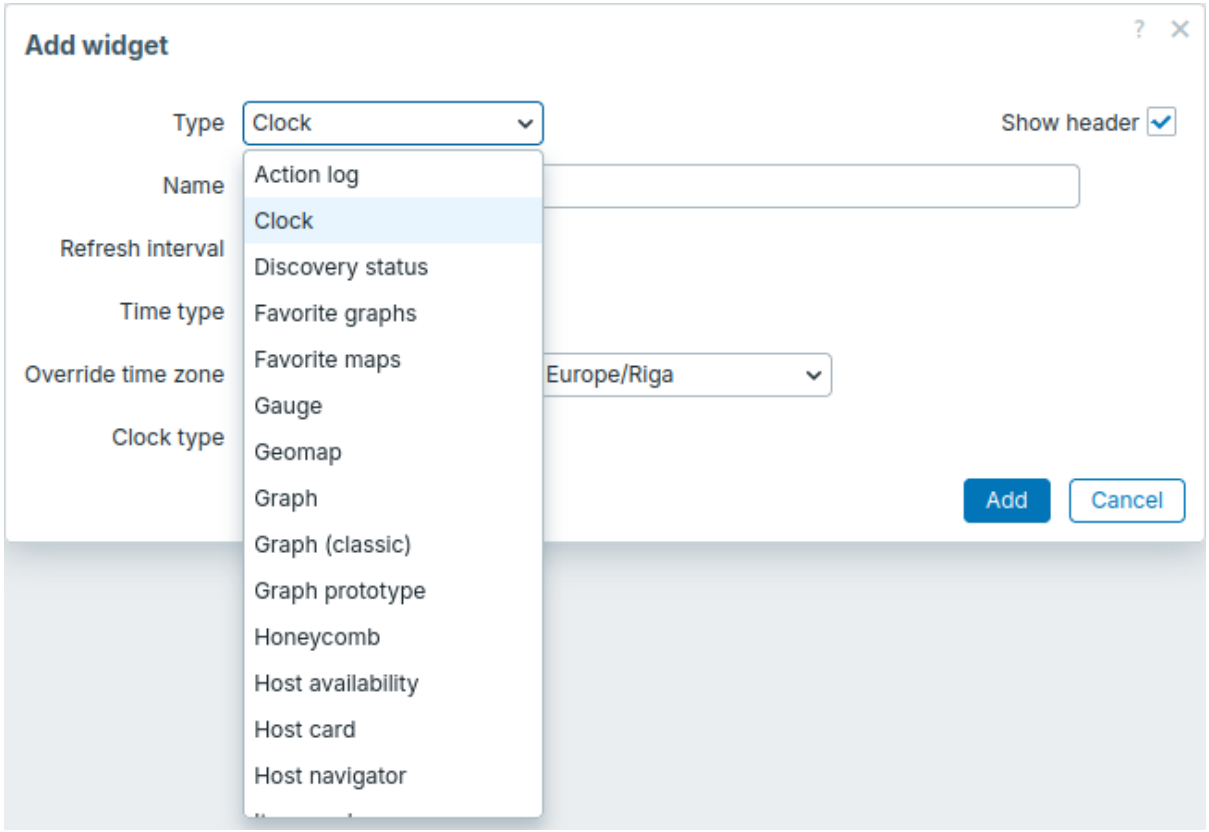
O espaço reservado do widget é adicionado com seu tamanho padrão ao primeiro espaço vazio disponível no dashboard, com o formulário de configuração do widget aberto.

- Mova o ponteiro do mouse sobre uma área vazia do dashboard. Em seguida, clique no espaço reservado do widget para abrir o formulário de configuração do widget. Como alternativa, clique e arraste o espaço reservado para redimensioná-lo e, em seguida, solte-o para abrir o formulário de configuração do widget; se um widget tiver sido copiado para a área de transferência, primeiro será solicitado que você escolha *Adicionar widget* ou *Colar widget*.



Em seguida, siga estas etapas:

1. No formulário de configuração do widget, selecione o tipo de widget.
2. Configure os parâmetros do widget.
3. Clique em *Adicionar* para adicionar o widget ao dashboard.



O widget é adicionado com seu tamanho padrão ou redimensionado para se ajustar ao espaço disponível. Você pode redimensionar e reposicionar widgets conforme descrito na seção *Editar widgets*.

Você também pode copiar e colar widgets para criar novos widgets com a mesma configuração, no mesmo dashboard ou entre dashboards abertos em diferentes abas do navegador.

Copiar/colar widgets

Para copiar e colar um widget:

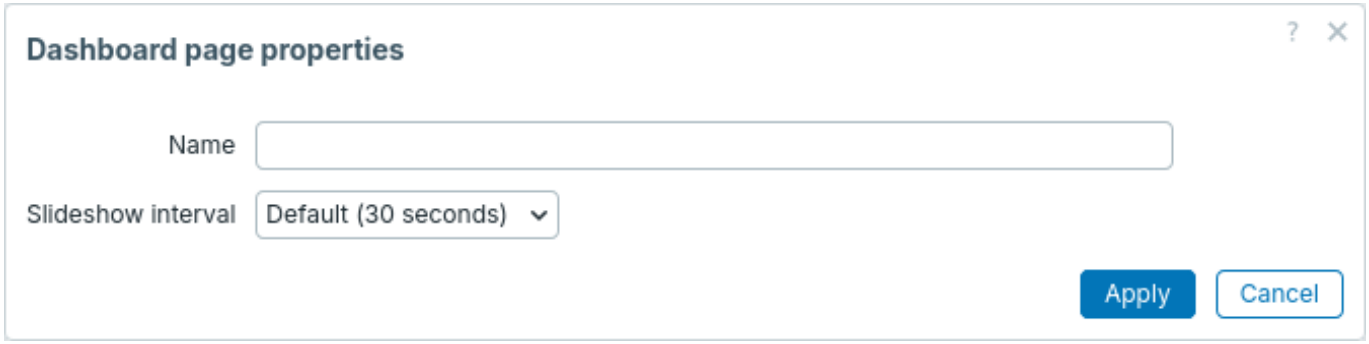
1. No menu *Ações* do widget ⋮, selecione *Copiar*.
2. No modo de edição do dashboard, clique na seta suspensa do botão *Adicionar* e selecione *Colar widget*. Como alternativa, clique em uma área vazia do dashboard e selecione *Colar widget*.

Você também pode colar um widget copiado sobre um widget existente usando a opção *Colar* no menu *Ações* dele.

Adicionar páginas

Para adicionar uma nova página a um dashboard:

1. No modo de edição do dashboard, clique na seta suspensa do botão *Adicionar* e selecione *Adicionar página*.
2. Preencha as propriedades da página:
    - *Nome* - insira o nome da página. Se você deixar o nome vazio, a página será adicionada com o nome *Página N*, em que 'N' é o número incremental da página.
    - *Intervalo da apresentação de slides* - selecione por quanto tempo a página será exibida antes que a apresentação de slides avance para a próxima página.



3. Clique em *Aplicar*.

Uma nova aba (por exemplo, *Página 2*) será adicionada para a nova página.



Você também pode copiar e colar páginas para criar novas páginas com as mesmas propriedades, dentro do mesmo dashboard ou entre dashboards abertos em abas diferentes.

Copiar/colar páginas

Para copiar e colar uma página:

1. No menu *Ações* da página , selecione *Copiar*.
2. No modo de edição do dashboard, clique na seta suspensa do botão *Adicionar* e selecione *Colar página*.

Editar dashboard

Para editar um dashboard, clique no botão *Editar dashboard* no canto superior direito. Como alternativa, clique no botão *Editar* de um widget. O dashboard é aberto no modo de edição.

A partir daqui, você pode gerenciar dashboards da seguinte forma:

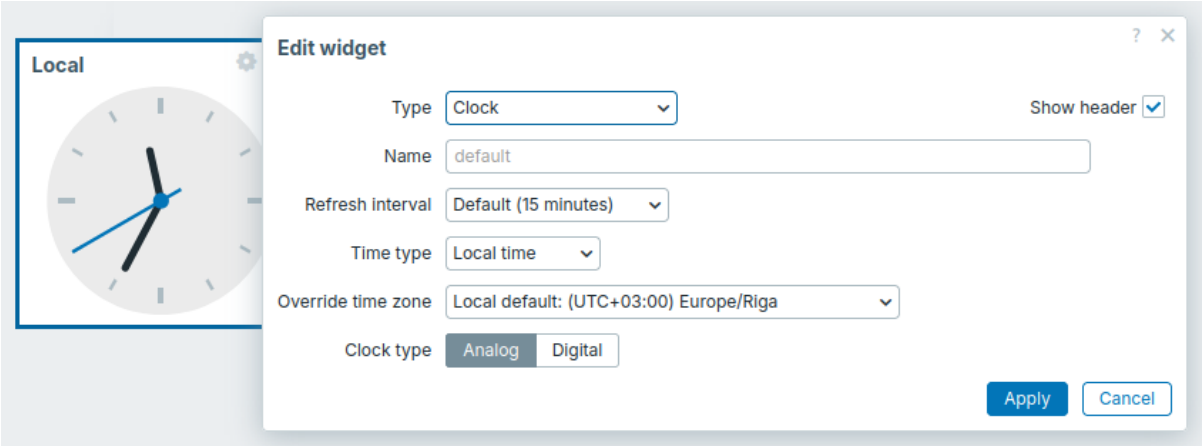
- Edite as propriedades do dashboard clicando no botão de engrenagem .
- Edite os widgets do dashboard.
- Acesse as seguintes ações do dashboard clicando na seta suspensa do botão *Adicionar*:
    - *Adicionar widget* - adicione um widget ao dashboard.
    - *Adicionar página* - adicione uma página ao dashboard.
    - *Colar widget* - cole um widget copiado. Essa opção ficará desabilitada se nenhum widget tiver sido copiado. Apenas um widget pode ser copiado por vez.
    - *Colar página* - cole uma página copiada. Essa opção ficará desabilitada se nenhuma página tiver sido copiada. Apenas uma página pode ser copiada por vez.

Depois de editar o dashboard, clique no botão *Salvar alterações* para salvar o dashboard ou em *Cancelar* para retornar à tela anterior.

Editar widgets

Um widget pode ser editado no modo de edição do dashboard clicando no botão *Edit* . Como alternativa, ao visualizar um dashboard, passe o cursor sobre um widget e clique no botão *Edit*.

Enquanto um widget está sendo editado, ele é destacado, e as alterações são visualizadas em tempo real.



Você pode reposicionar o widget e seu formulário de configuração arrastando suas barras de título e redimensionar o widget arrastando suas alças de redimensionamento.



As seguintes ações estão disponíveis no menu *Actions* do widget ... ao editar um widget:

- *Copy* - copiar o widget.
- *Paste* - colar um widget copiado sobre este widget. Essa opção fica desativada se nenhum widget tiver sido copiado.
- *Delete* - excluir o widget.

Se os parâmetros obrigatórios do widget não estiverem configurados, o widget entrará no estado *Widget is not fully configured*.

Após a edição, clique em *Apply* no formulário de configuração do widget e, em seguida, em *Save changes* no canto superior direito do dashboard para aplicar as alterações.

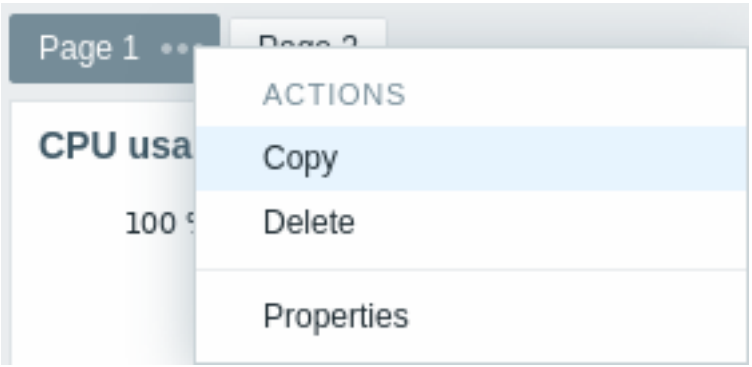
Editar páginas

As páginas podem ser editadas no modo de edição do dashboard.

As páginas podem ser reordenadas arrastando e soltando as abas das páginas. A reordenação não renomeia as páginas, exceto as páginas com nomes atribuídos automaticamente (*Página N*), que são renumeradas.

As seguintes ações estão disponíveis no menu *Ações* da página  ao editar uma página:

- *Copiar* - copiar a página.
- *Excluir* - excluir a página.
- *Propriedades* - personalizar as propriedades da página.



Permissões aos dashboards

As permissões aos dashboards para usuários comuns e usuários do tipo *Admin* são limitadas conforme descrito abaixo, desde que o acesso aos dashboards e a opção *Criar e editar dashboards* estejam habilitados para a função de usuário atribuída:

- Eles podem visualizar e clonar um dashboard público mesmo quando nenhum usuário ou grupo de usuários é adicionado às preferências de compartilhamento do dashboard.
- Eles podem visualizar e clonar um dashboard privado se tiverem pelo menos direitos de *Leitura* sobre ele, conforme especificado nas preferências de compartilhamento.
- Eles podem editar e excluir um dashboard somente se tiverem direitos de *Leitura e gravação* sobre ele, conforme especificado nas preferências de compartilhamento.

- Eles não podem alterar o proprietário do dashboard.

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.

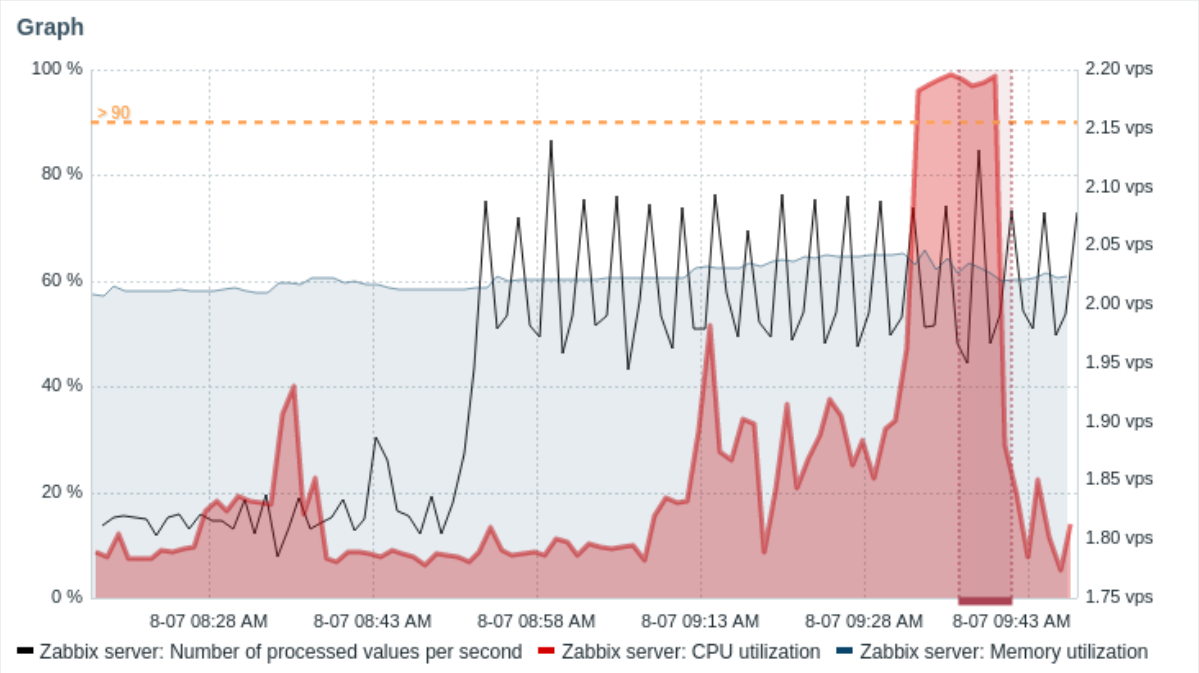
Para detalhes de configuração dos widgets, consulte Parâmetros dos widgets do dashboard.

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.



Veja também:

- 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 dispersão

O widget Gráfico de dispersão exibe a relação entre duas métricas, plotando pontos de dados individuais ao longo dos eixos X e Y. Isso ajuda a revelar padrões, agrupamentos, correlações e valores discrepantes no conjunto de dados.

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.

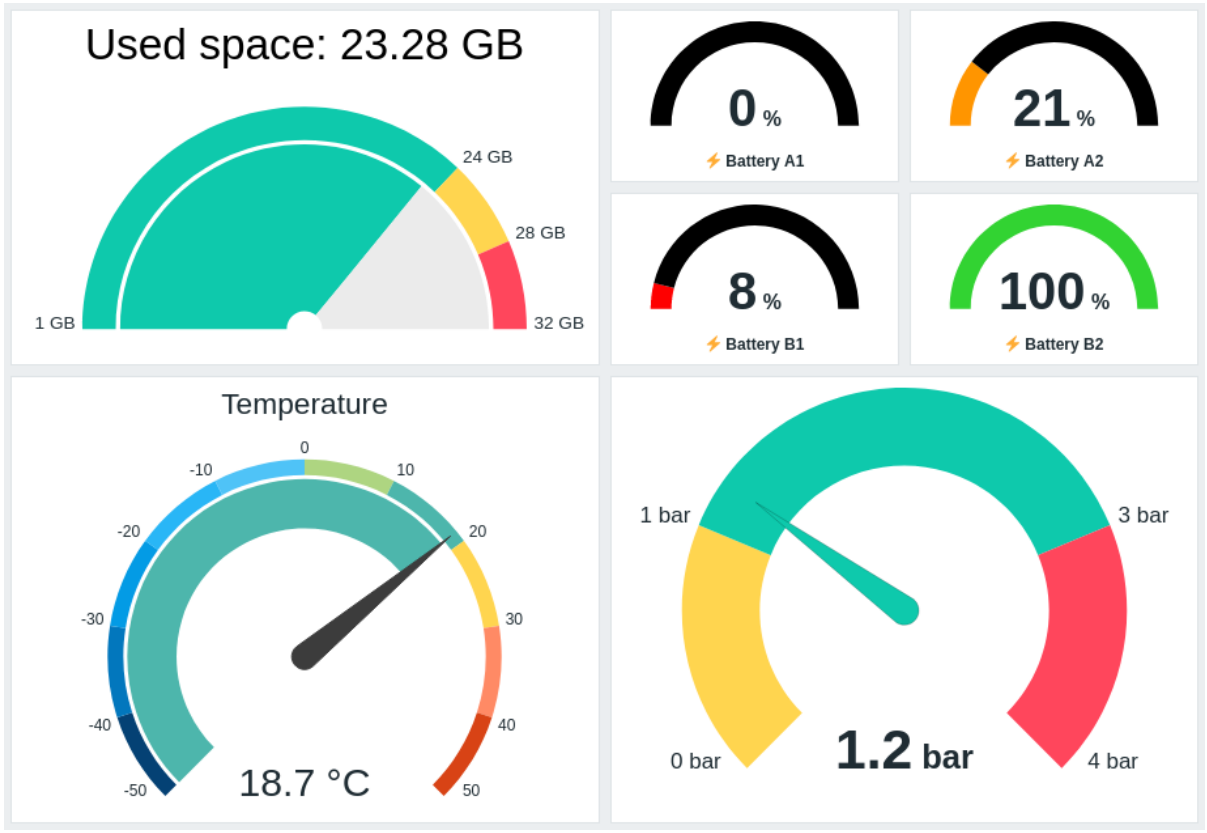
—

—

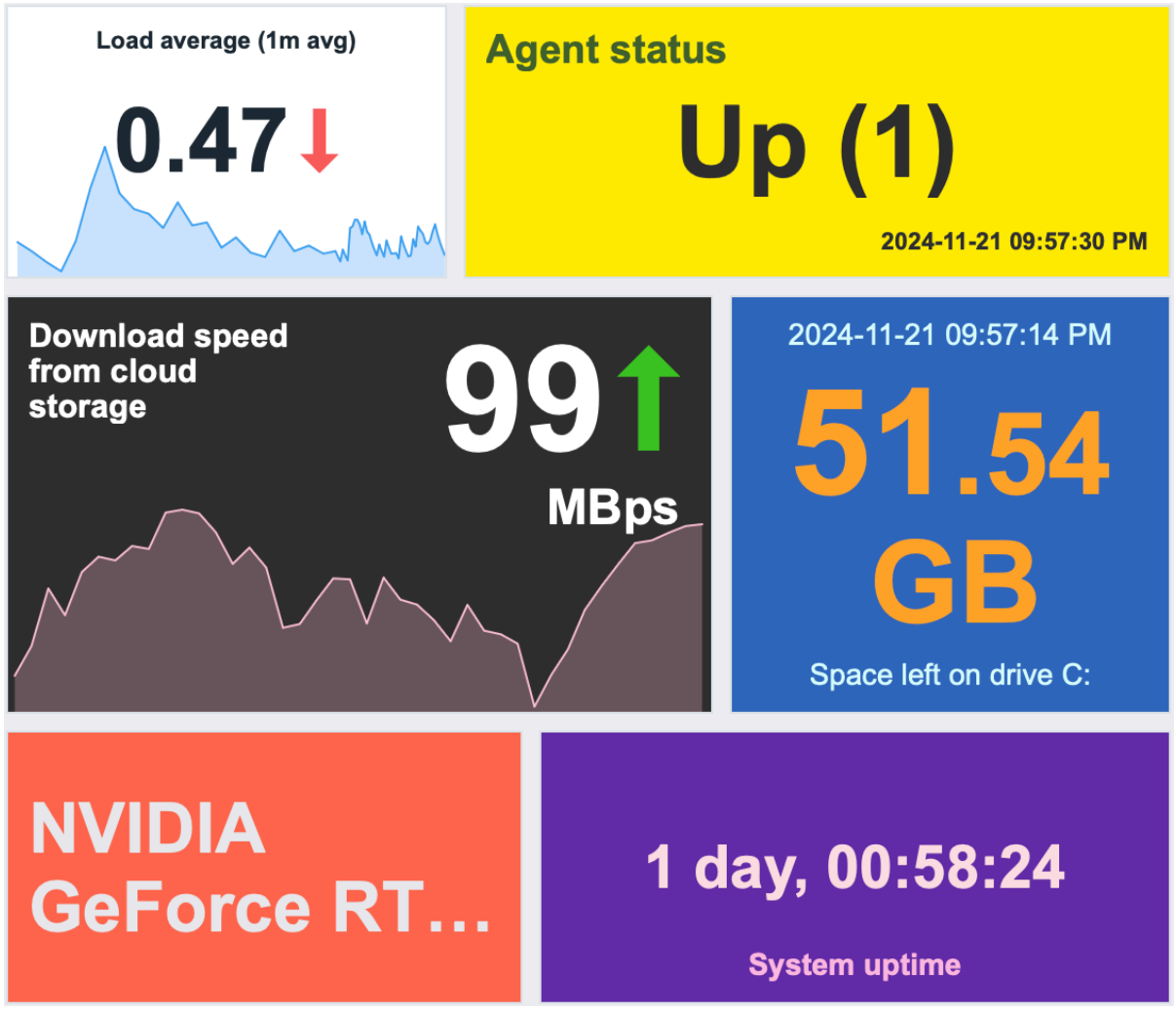
Medidor/Valor do item

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:

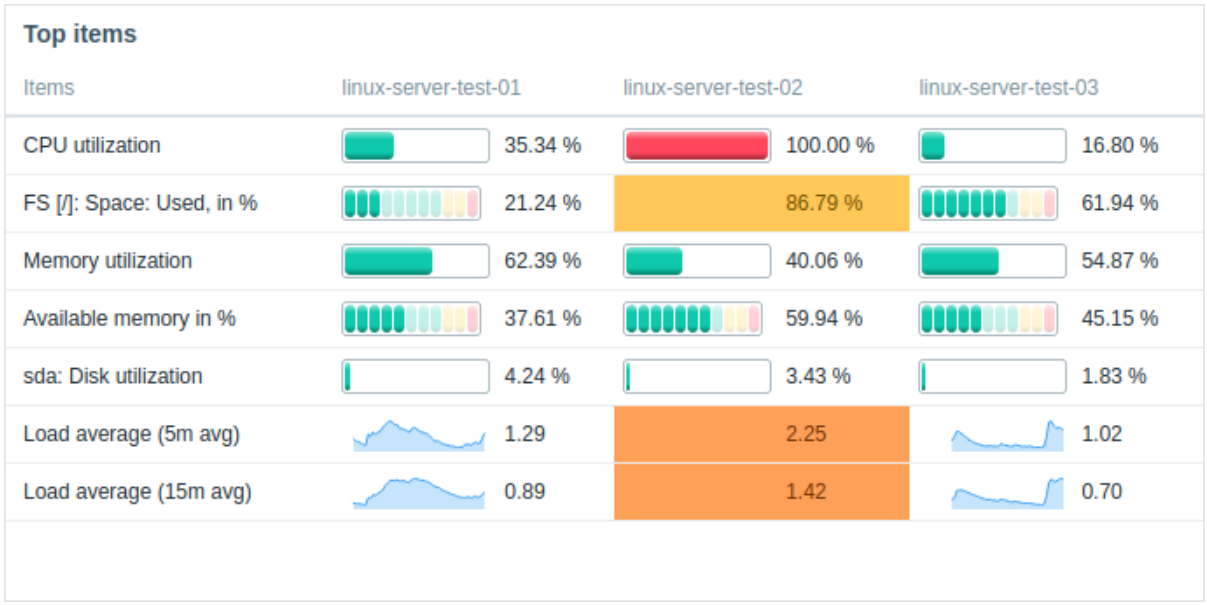


Histórico do item

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 de seu desempenho.

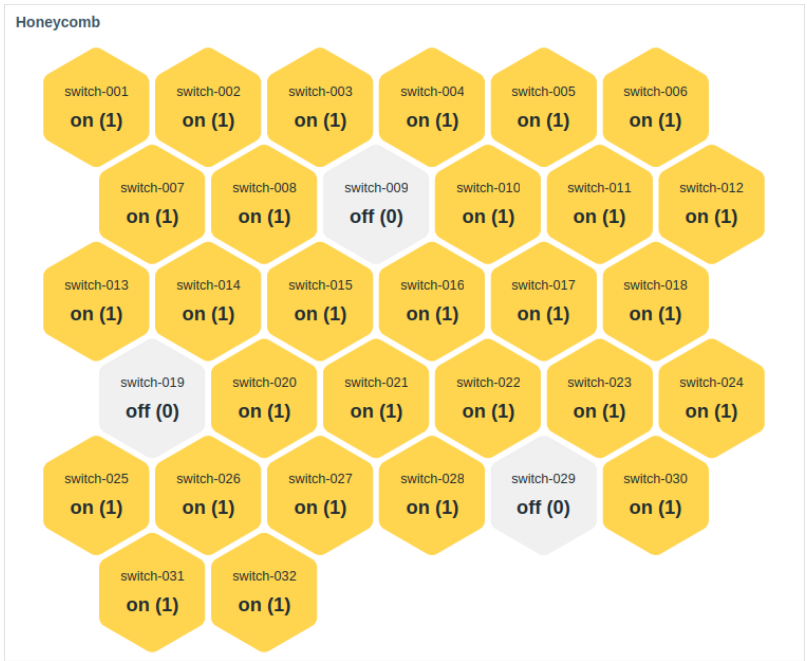


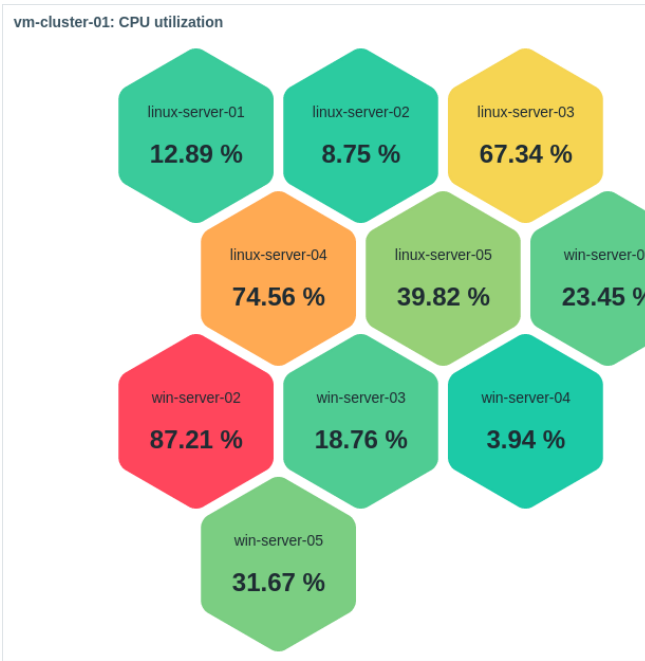
Veja também:

- 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.



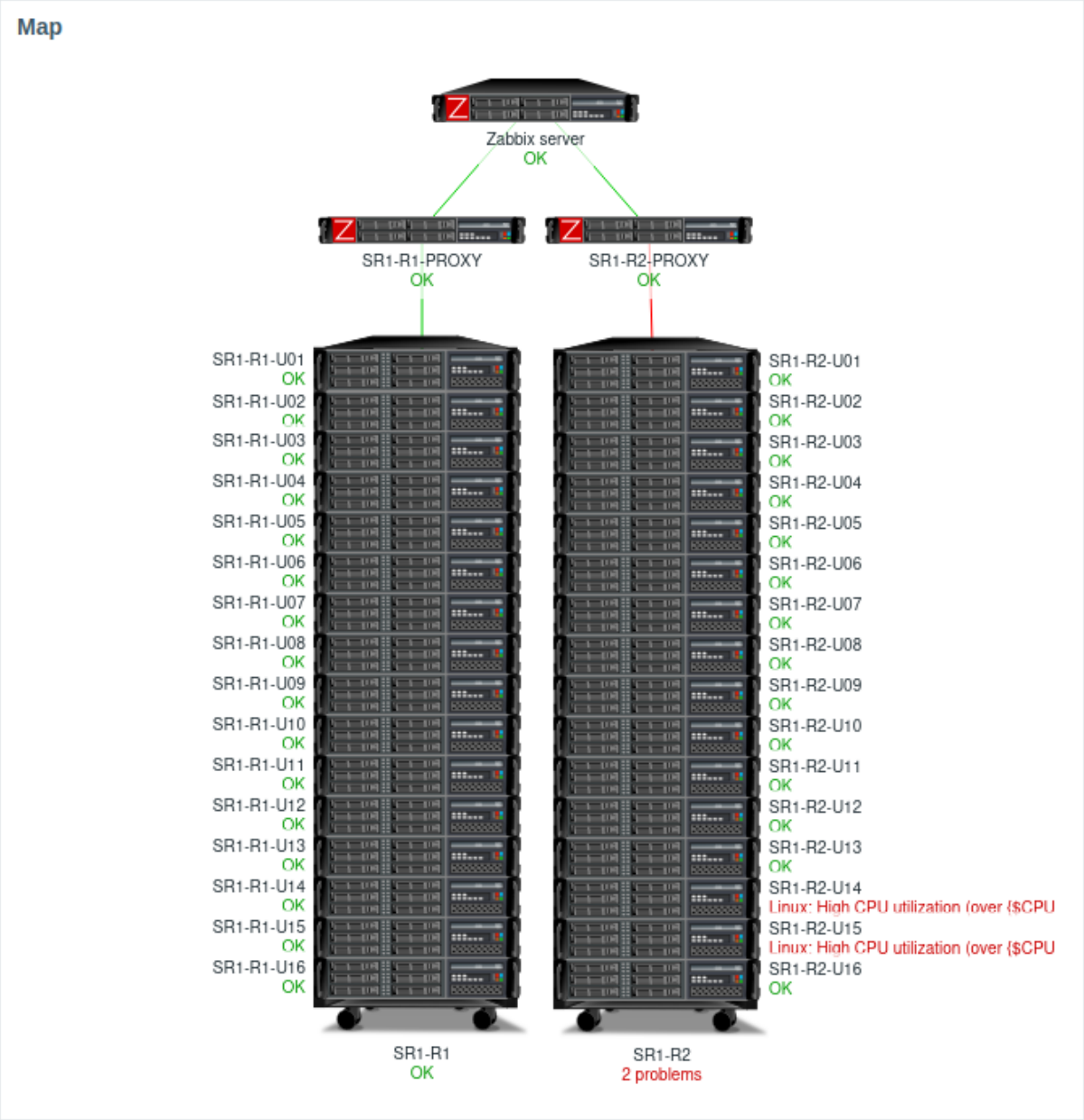


Mapas

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.



Veja também:

- 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

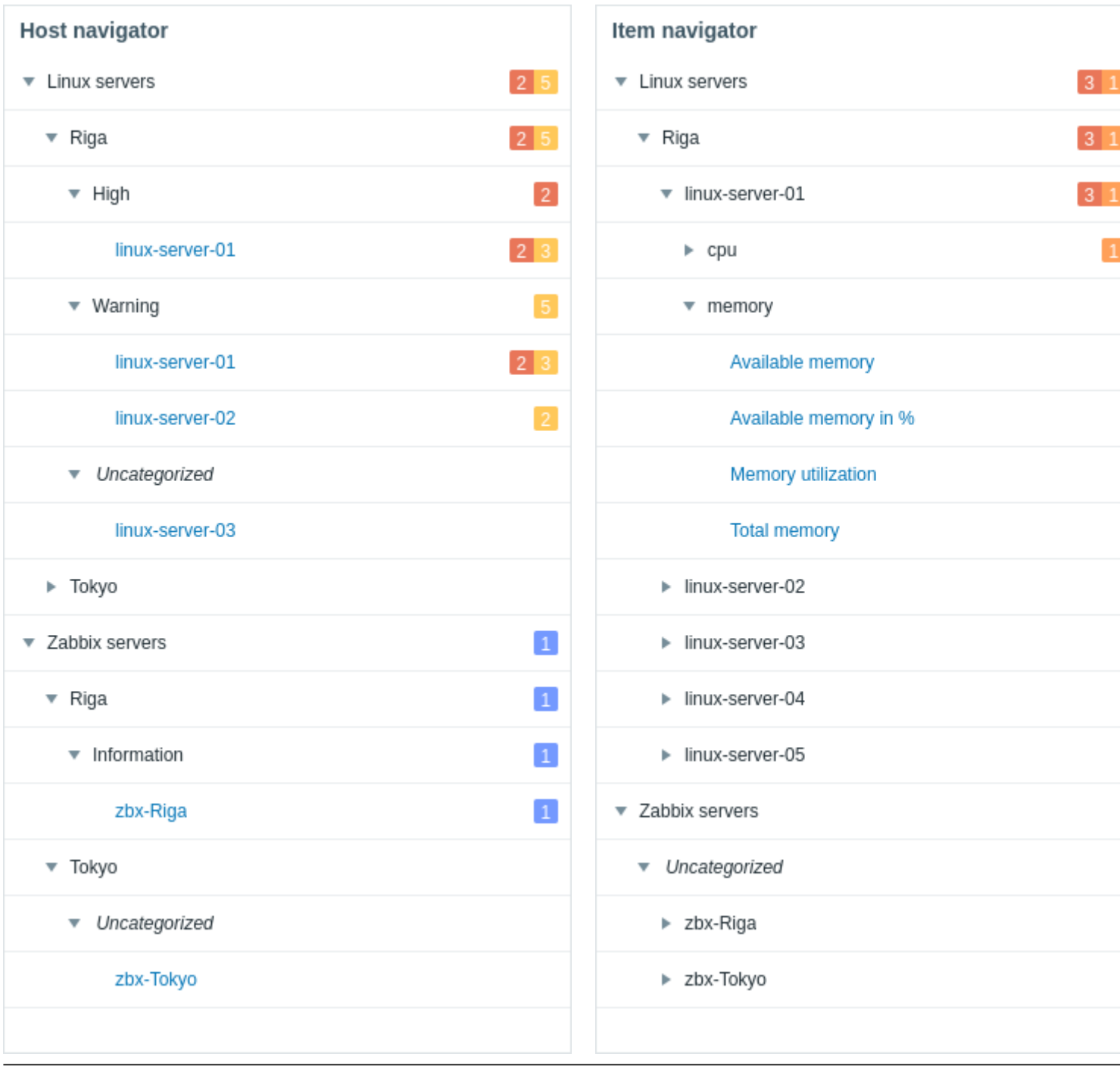
O widget Geomap exibe hosts como marcadores em um mapa interativo baseado em Leaflet, com as cores dos marcadores indicando o problema mais grave de cada host.

Detalhes do host/item

Esses widgets fornecem listas e visualizações detalhadas de hosts ou items.

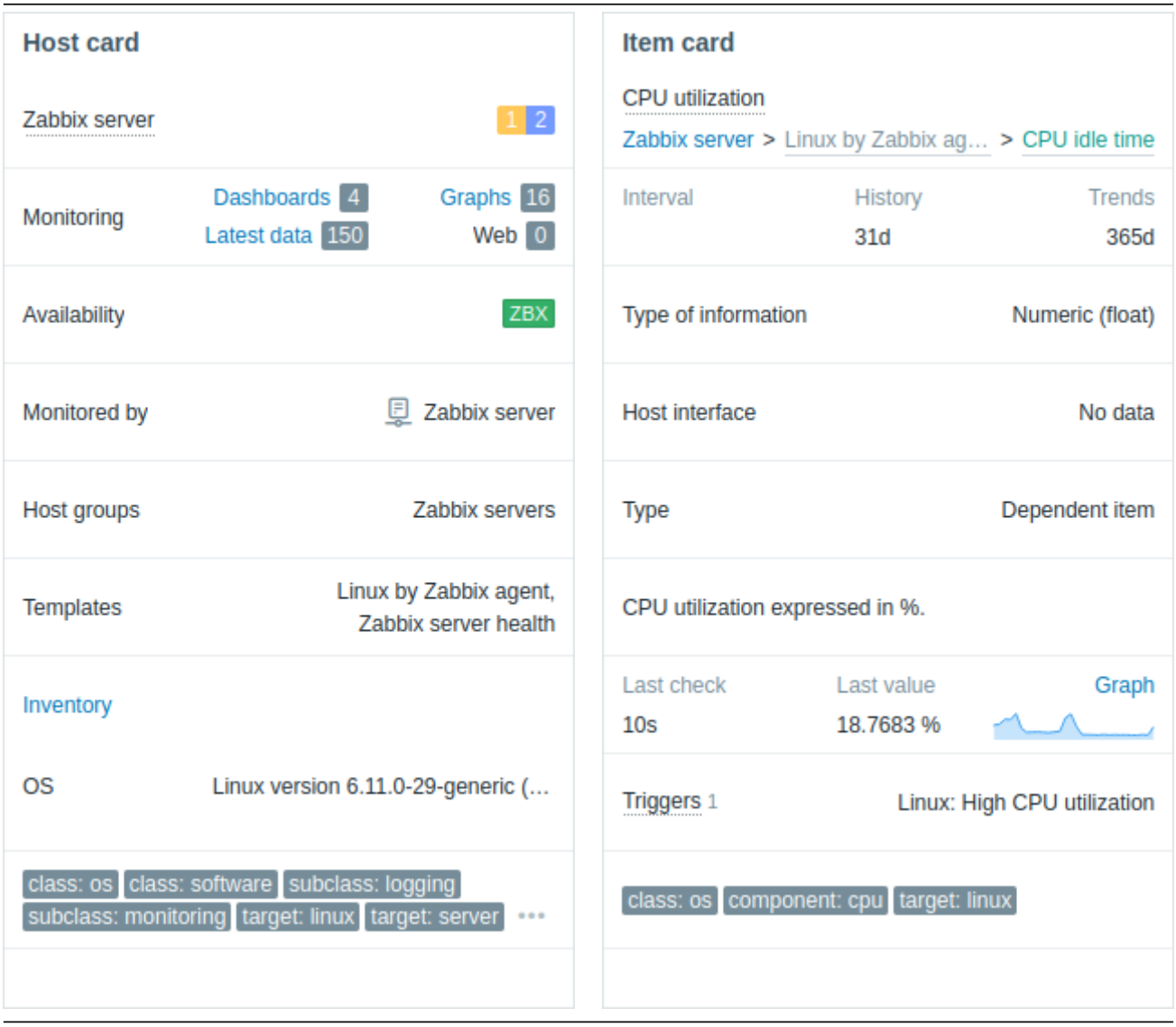
Navegador de Host/Item

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

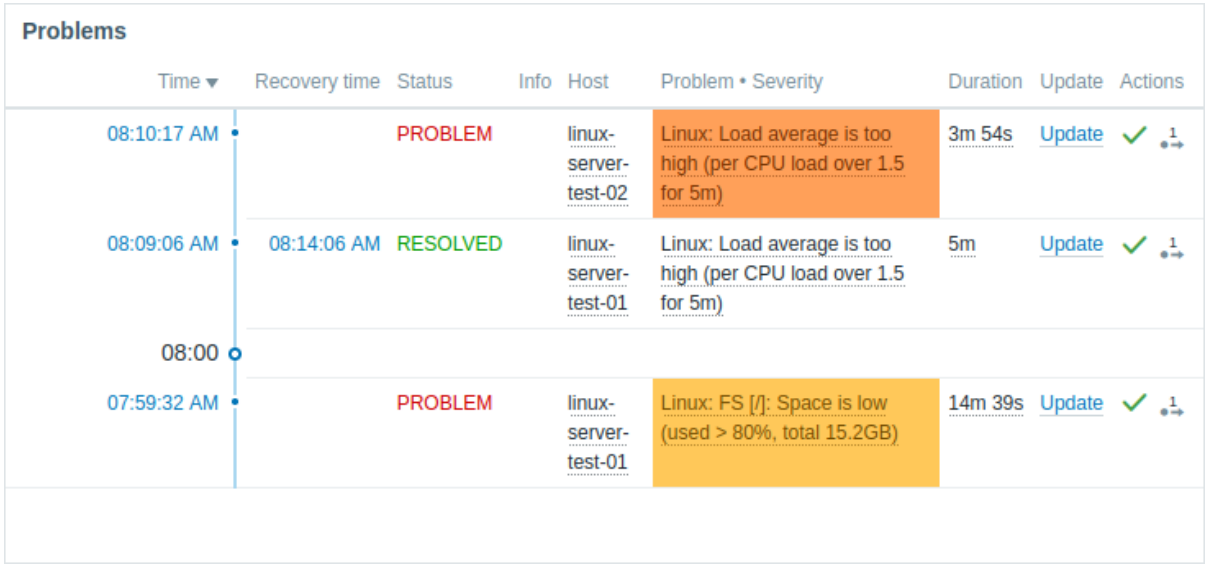
Esses widgets exibem dados de status (por exemplo, disponibilidade, triggers, problemas), bem como relatórios.

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.



Veja também:

- 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

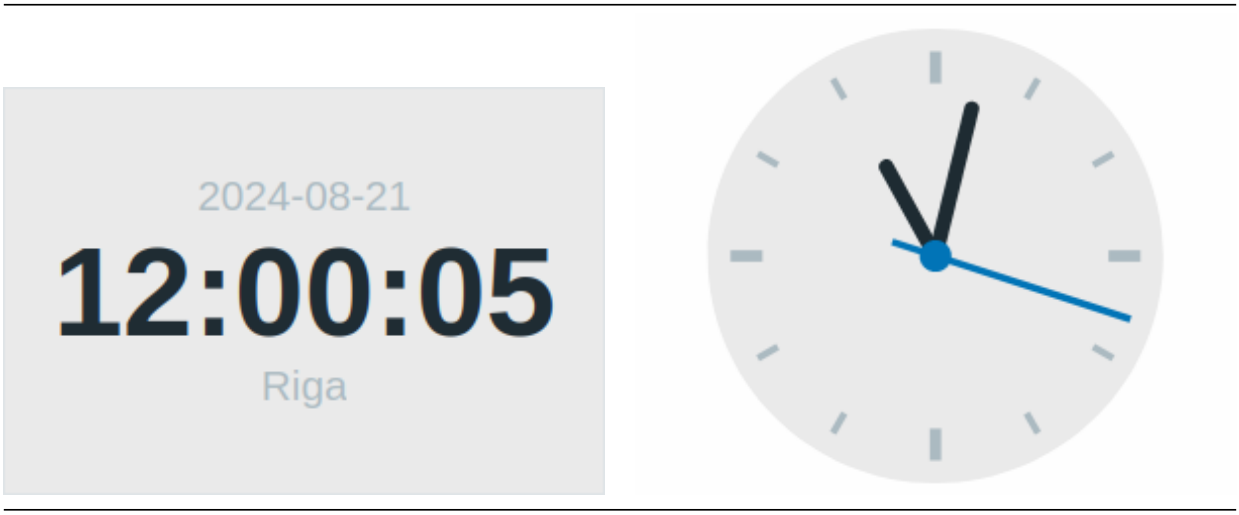
Esses widgets fornecem informações e ferramentas suplementares para aprimorar seus dashboards.

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.

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 parâmetros a seguir são comuns a todos os widgets:

Nome	Insira um nome para o widget.
Intervalo de atualização	Configure o intervalo de atualização padrão. Os intervalos de atualização padrão dos 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 como um valor padrão para todos os usuários. Alterne 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 no botão de três pontos em um widget e selecione o intervalo de atualização desejado na lista suspensa. Observe que o intervalo de atualização exclusivo de um usuário tem prioridade sobre a configuração do widget e é preservado mesmo quando a configuração do widget é modificada.
Mostrar cabeçalho	Marque a caixa de seleção para exibir permanentemente o cabeçalho do widget. Quando desmarcada, o cabeçalho fica oculto para economizar espaço e torna-se visível somente quando o mouse passa sobre o widget (tanto no modo de visualização quanto no modo de edição). O cabeçalho também fica parcialmente 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 de widgets

Alguns widgets podem transmitir dados de configuração para outros widgets, alguns podem escutar dados e alguns podem fazer ambos. Por exemplo:

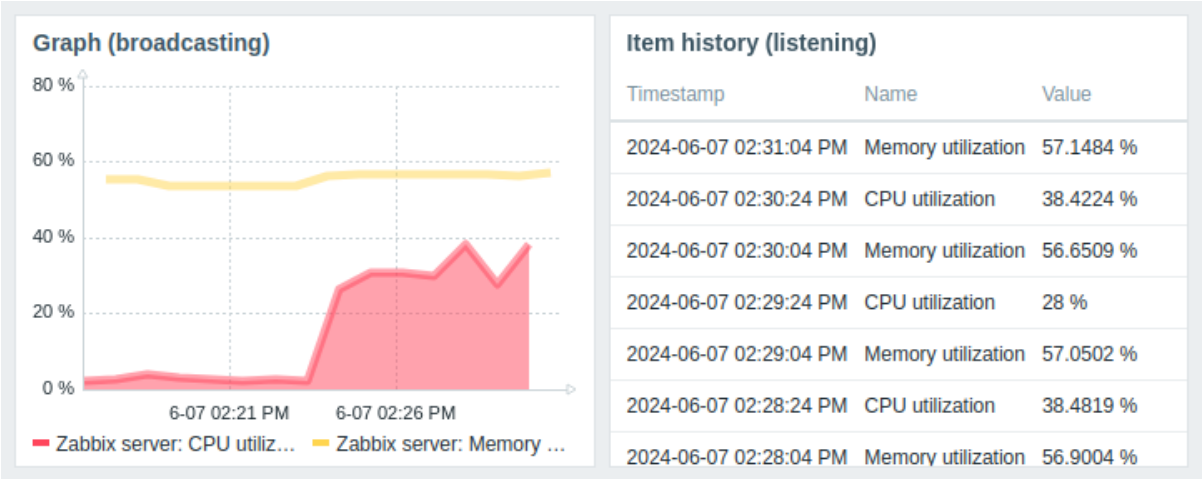
- O widget *Action log* só pode recuperar dados de período de tempo dos widgets *Graph*, *Graph (classic)* e *Graph prototype*.
- O widget *Geomap* pode transmitir dados de host para widgets que os escutam (*Honeycomb*, *Top items*, etc.) e também pode escutar dados de grupo de hosts e de host de widgets que os transmitem (*Honeycomb*, *Problem hosts*, etc.).
- O widget *Clock* não pode transmitir nem escutar nenhum dado.

A tabela a seguir descreve as capacidades de transmissão e escuta de cada widget.

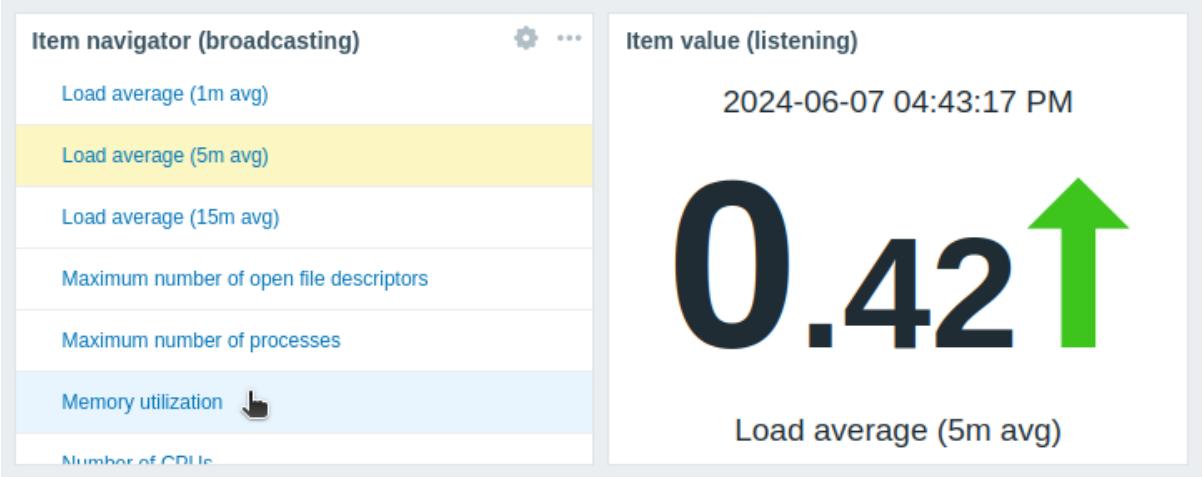
Widget	Transmite	Escuta
Action log	-	Período de tempo
Clock	-	-
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	Hosts, Items
Geomap	Hosts	Host groups, Hosts
Graph	Time period, Items	Hosts, Items, Time period
Graph (classic)	Time period	Hosts, Items, Graphs, Time period
Graph prototype	Time period	Hosts, Time period
Honeycomb	Hosts, Items	Host groups, Hosts
Host availability	-	Host groups
Host card	-	Hosts
Host navigator	Hosts	Host groups
Item card	-	Hosts, Items, Time period
Item history	Items	Hosts, Time period
Item navigator	Items	Host groups, Hosts
Item value	-	Hosts, Items, Time period
Map	Host groups, Hosts	Maps
Map navigation tree	Maps	-
Pie chart	-	Items, Time period
Problem hosts	Host groups	Host groups, Hosts
Problems	Events	Host groups, Hosts
Problems by severity	Host groups	Host groups, Hosts
Scatter plot	Time period, Items	Hosts, Items, Time period
SLA report	-	-
System information	-	-
Top hosts	Hosts	Host groups, Hosts
Top items	-	Host groups, Hosts
Top triggers	-	Time period
Trigger overview	-	Host groups, Hosts
URL	-	Hosts
Web monitoring	Host groups	Host groups, Hosts

Os widgets diferem em como **transmitem** dados para outros widgets.

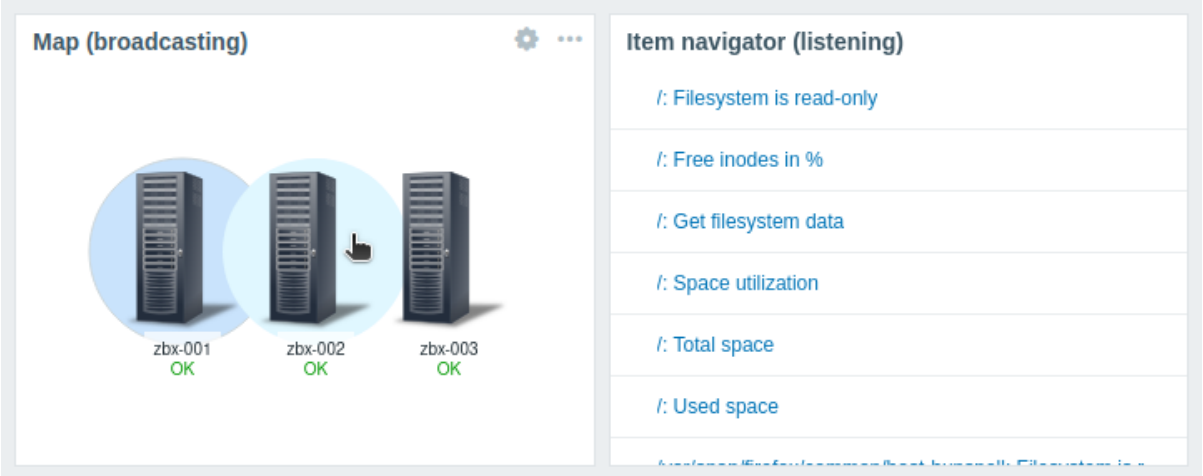
Os widgets que suportam transmissão começam a fazê-lo automaticamente quando são criados. Por exemplo, o widget *Gráfico* transmite imediatamente os dados do período de tempo para os widgets ouvintes.



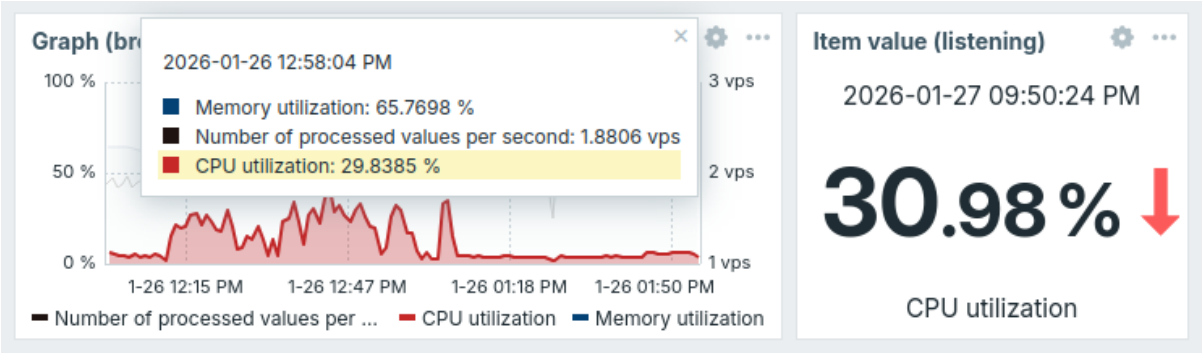
Os widgets que suportam seleção de entidade transmitem automaticamente os dados para a primeira entidade disponível. Por exemplo, o widget *Navegador de item* transmite dados para o primeiro item em sua lista de itens. Quando um item diferente é selecionado, ele transmite dados para esse item. Ao passar 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*, 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.

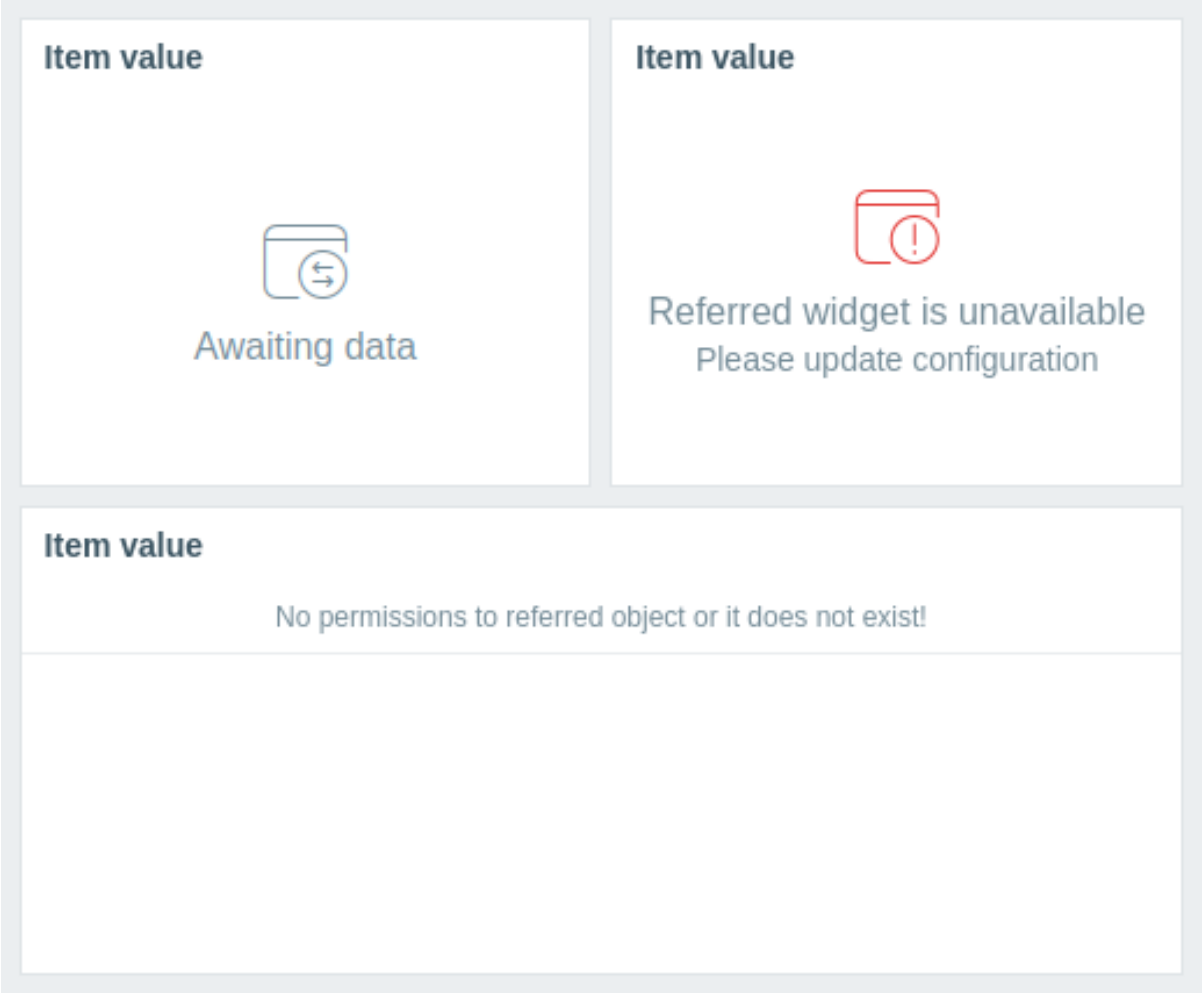


Widgets como *Gráfico* e *Gráfico de dispersão* podem transmitir dados de item quando um item é selecionado em suas dicas de ferramenta:



Os widgets também têm comportamentos específicos ao **ouvir** dados de outros widgets:

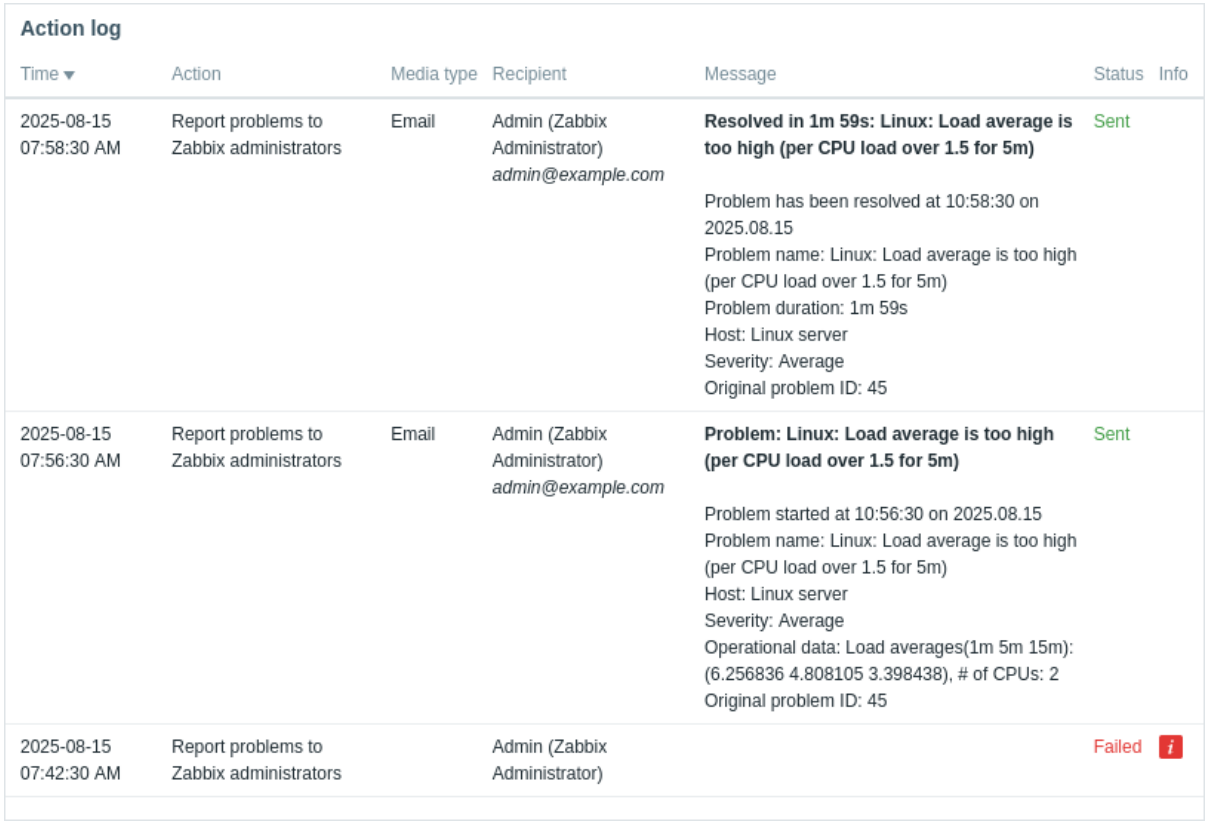
- Se o widget fonte de dados não estiver transmitindo dados, o widget ouvinte entra no estado *Aguardando dados*.
- Se o widget fonte de dados foi excluído, substituído por um widget incompatível ou movido para outra página do dashboard, o widget ouvinte entra no estado *Widget referenciado indisponível*.
- Se o host especificado na fonte de dados (widget ou dashboard) não 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 permissões para o objeto referenciado ou ele não existe!"*



Log de ação

Visão geral

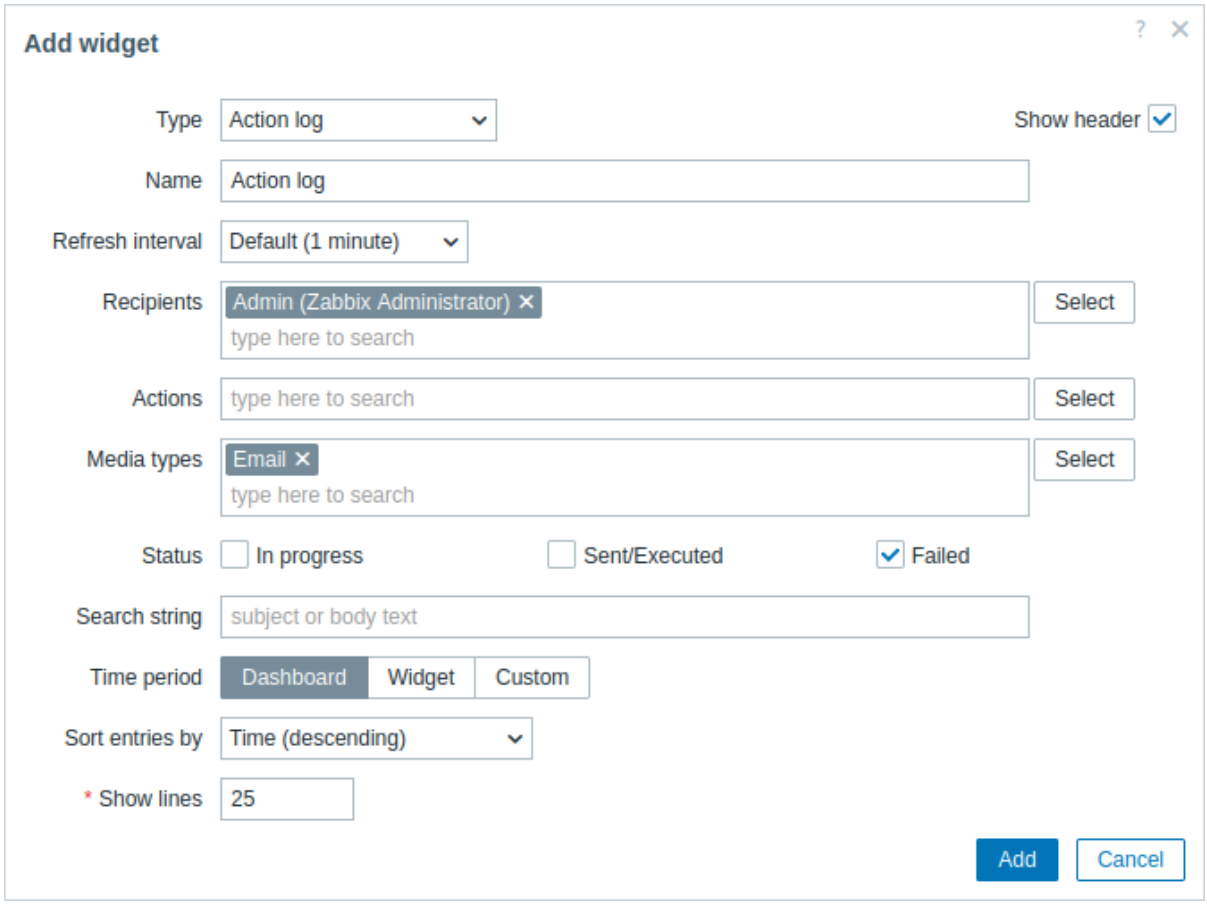
O widget *Log de ação* exibe detalhes das operações (notificações, comandos remotos) executadas dentro de uma ação. O log detalhado auxilia na auditoria, solução de problemas e monitoramento da execução de cada ação.



Ele exibe os mesmos dados de *Relatórios* > *Log de ação* e pode exibir até 1000 registros.

Configuração

Para configurar, selecione *Action log* como tipo:



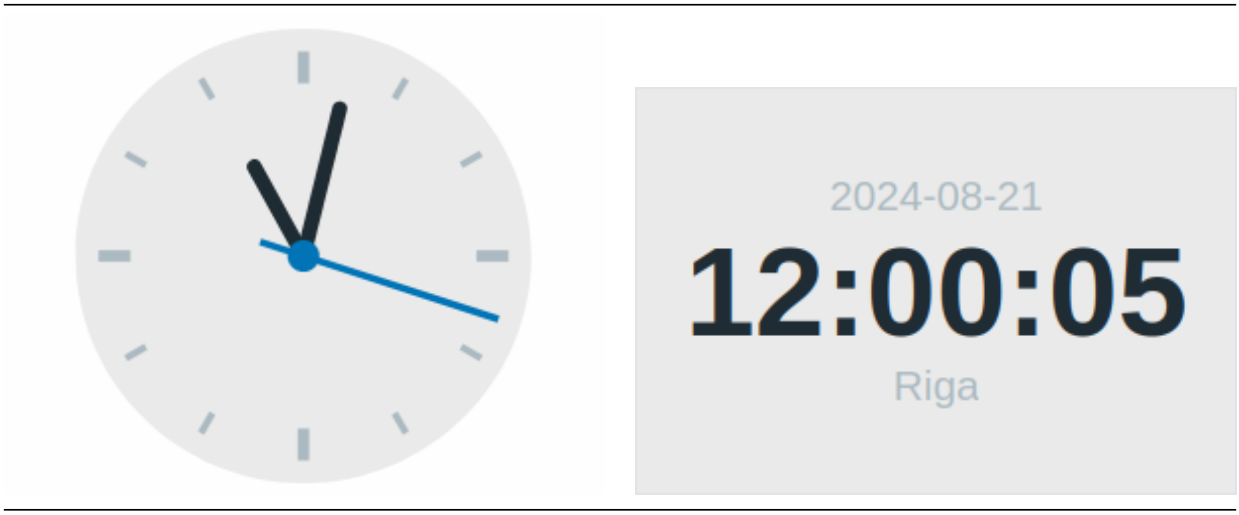
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

Relógio

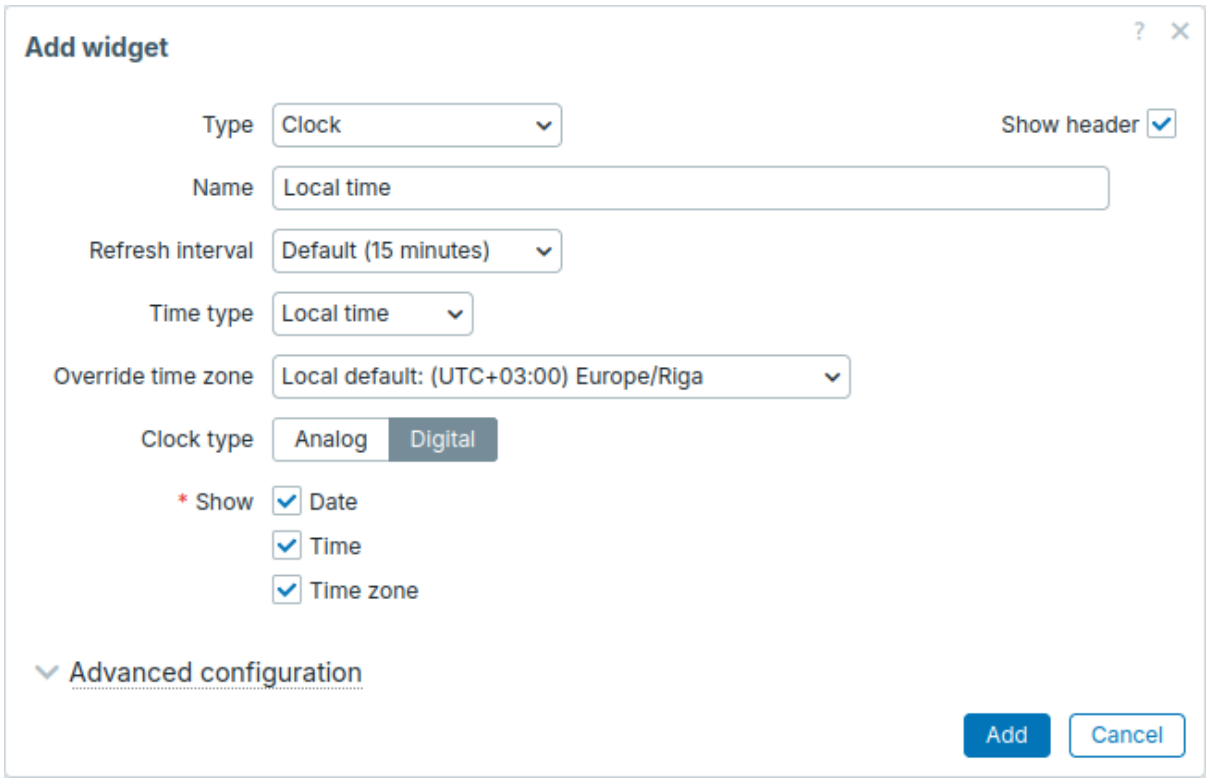
Visão geral

O widget *Relógio* exibe a hora local, do servidor ou do host no formato analógico ou digital.



Configuração

Para configurar, selecione *Clock* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

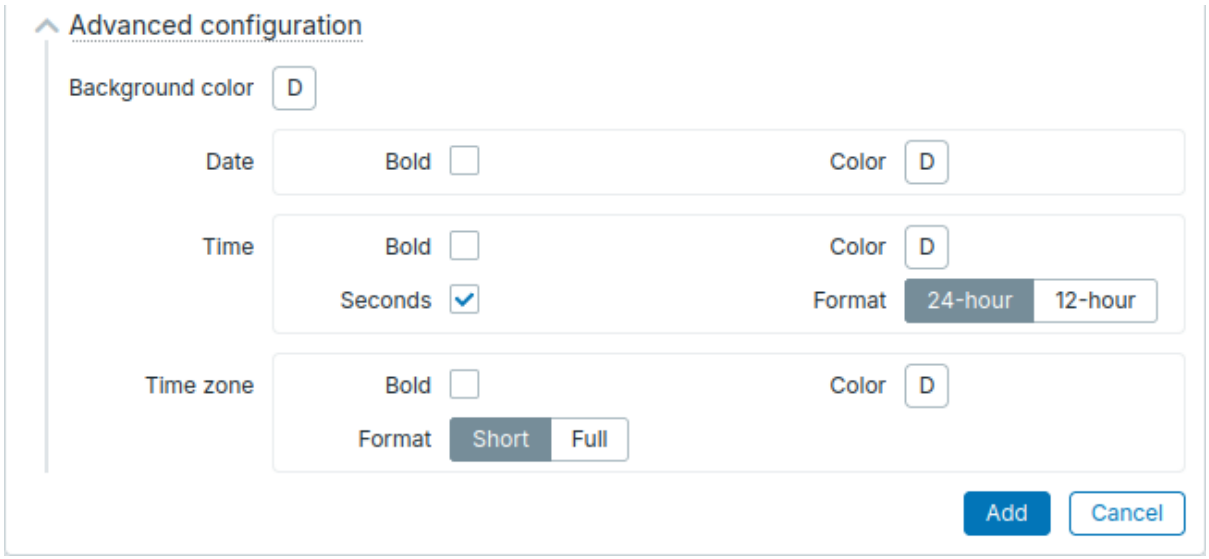
Time type	Selecione a fonte de horário: **Local time** - horário do sistema do dispositivo que executa o navegador; **Server time** - fuso horário definido globalmente ou para o usuário do Zabbix; **Host time** - horário recuperado de um item de um host.
Item	Selecione o item com a chave `system.localtime[local]` para exibir o horário do host. Este parâmetro está disponível quando *Time type* está definido como "Host time".
Clock type	Selecione o tipo de relógio (analógico ou digital).
Override time zone	Selecione o fuso horário. Este parâmetro está disponível para os tipos de horário *Local time* e *Server time*.
Show	Selecione uma ou mais unidades de informação (data, horário, 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 quando *Clock type* está definido como "Digital".

Advanced configuration	Clique no cabeçalho *Advanced configuration* para exibir as opções de configuração avançada do relógio digital. Esta seção está disponível quando *Clock type* está definido como ”Digital”.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*, e somente para os elementos selecionados no campo *Exibir* (consulte acima).

Além disso, a configuração avançada permite alterar a cor de fundo de todo o widget.

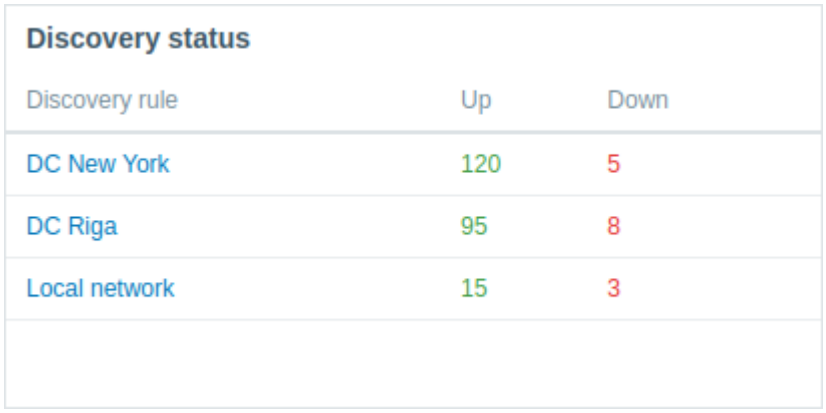


Cor de fundo		Selecione a cor de fundo no seletor de cores. D representa a cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Data		Esta seção está disponível quando a caixa de seleção *Data* está selecionada para o parâmetro *Exibir*.
	Negrito	Marque a caixa de seleção para exibir a data em negrito.
	Cor	Selecione a cor da data no seletor de cores. D representa a cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Hora		Esta seção está disponível quando a caixa de seleção *Hora* está selecionada para o parâmetro *Exibir*.
	Negrito	Marque a caixa de seleção para exibir a hora em negrito.
	Cor	Selecione a cor da hora no seletor de cores. D representa a 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, somente as horas e os minutos serão exibidos.
	Formato	Selecione a exibição da hora no formato de 24 ou 12 horas.
Fuso horário		Esta seção está disponível quando a caixa de seleção *Fuso horário* está selecionada para o parâmetro *Exibir*.
	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 representa a cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
	Formato	Selecione a exibição do fuso horário no formato abreviado (por exemplo, `New York`) ou no formato completo (por exemplo, `(UTC-04:00) America/New York`).

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.



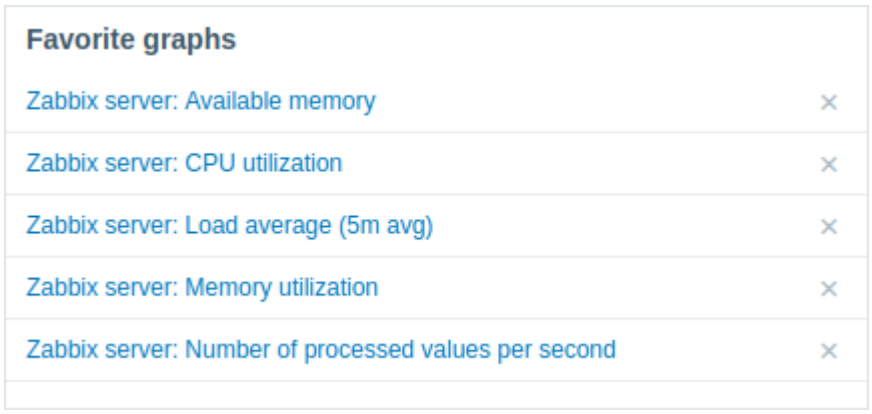
Configuração

Todos os parâmetros de configuração para este widget são comuns a todos os widgets.

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.

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*.

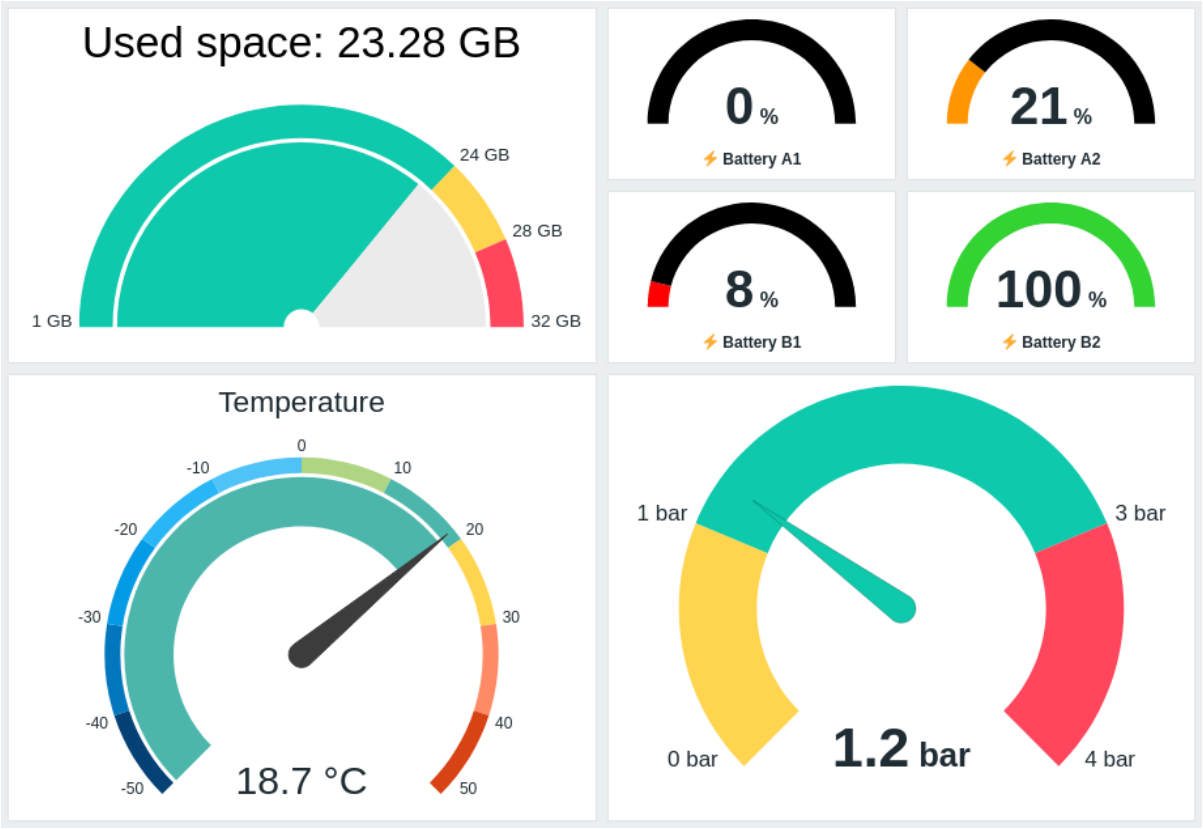
Configuração

Todos os parâmetros de configuração para este widget são comuns a todos os widgets.

Medidor

Visão geral

O widget *Gauge* exibe o valor de um único item numérico como um medidor. Ele é útil para acompanhar métricas importantes, visualizar limites e detectar alterações repentinas.



Você pode configurar o widget para exibir o seguinte:

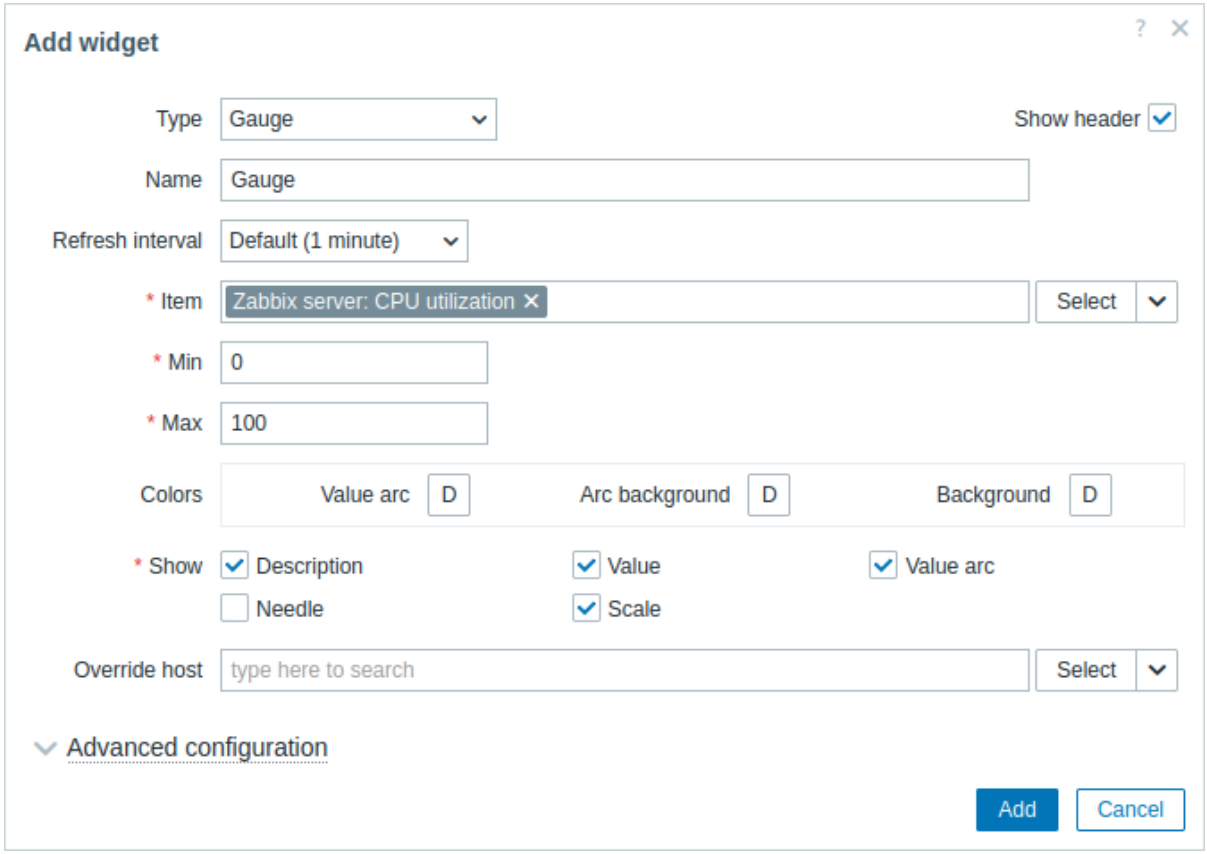
- Descrição do item (Espaço usado: {ITEM.LASTVALUE}, Temperatura)
- Valor e unidades do item (21 %, 18.7 °C, 1.2 bar)
- Escala do medidor (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Arco do medidor (arco do valor do medidor e arco dos limites do medidor)
- Ponteiro do medidor

Clicar no widget abre um gráfico simples para o item.

As informações exibidas no widget *Gauge* podem ser baixadas como uma imagem PNG selecionando a opção *Baixar imagem* no menu *Ações* do widget ⋮, que aparece quando você passa o cursor sobre o widget. Dependendo das configurações do seu navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Salvar* é aberta, permitindo escolher o local.

Configuração

Para configurar, selecione *Gauge* como tipo:



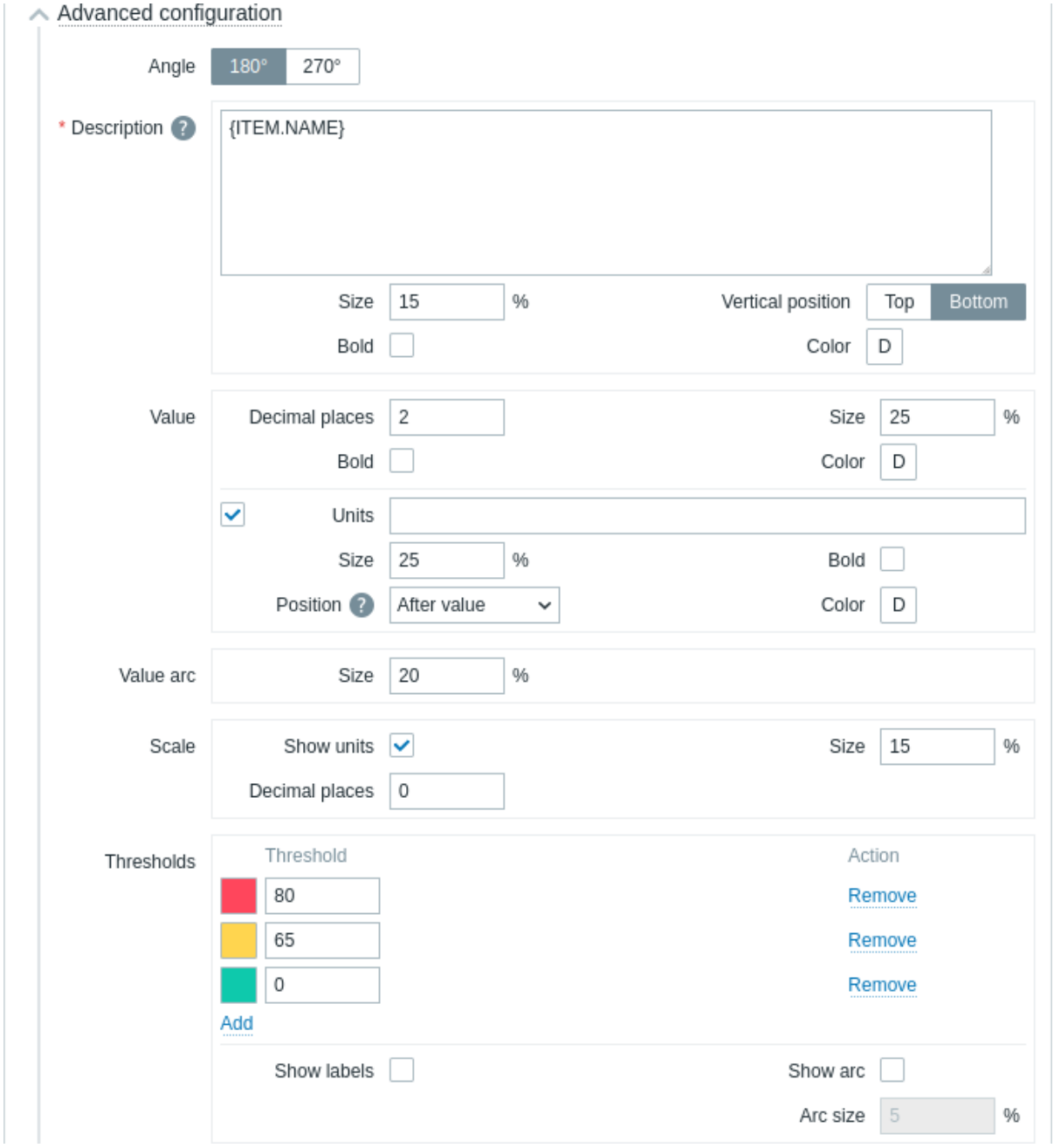
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Item	Selecione o item. Como alternativa, selecione um widget compatível como fonte de dados para itens. Este campo tem preenchimento automático, portanto, ao começar a digitar o nome de um item, será exibida uma lista suspensa com os itens correspondentes. Observe que você pode selecionar apenas itens que retornam dados numéricos (exceto dados binários).
Min	Insira o valor mínimo do medidor. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.
Max	Insira o valor máximo do medidor. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.
Colors	Selecione a cor no seletor de cores: **Arco do valor** - selecione a cor do arco do valor do medidor; **Fundo do arco** - selecione a cor de fundo do arco do valor do medidor e do arco dos limites do medidor; **Fundo** - selecione a cor de fundo do widget. ”D” representa a cor padrão, que depende do tema do frontend. Se *Thresholds* estiver configurado, a cor padrão de **Arco do valor** dependerá da cor do limite. Para retornar à cor padrão, clique no botão *Use default* no seletor de cores.
Show	Marque a caixa de seleção para exibir o respectivo elemento do medidor: descrição, valor, arco do valor, ponteiro, escala (o valor mínimo e máximo do medidor no início e no fim do arco do medidor). Desmarque para ocultar. Pelo menos um elemento deve ser selecionado. Observe que o ponteiro e a escala do medidor podem ser exibidos se o arco do valor do medidor ou o arco dos limites do medidor (consulte as opções de configuração avançada) estiverem exibidos. Observe também que, se o ponteiro do medidor estiver exibido, o valor será posicionado abaixo do ponteiro; se o ponteiro estiver oculto, o valor será alinhado com a parte inferior do arco do medidor.
Override 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.

Advanced configuration	Clique no cabeçalho *Advanced configuration* para exibir as opções de configuração avançada. É também aqui que você poderá ajustar os elementos do medidor selecionados no campo *Show*.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*:



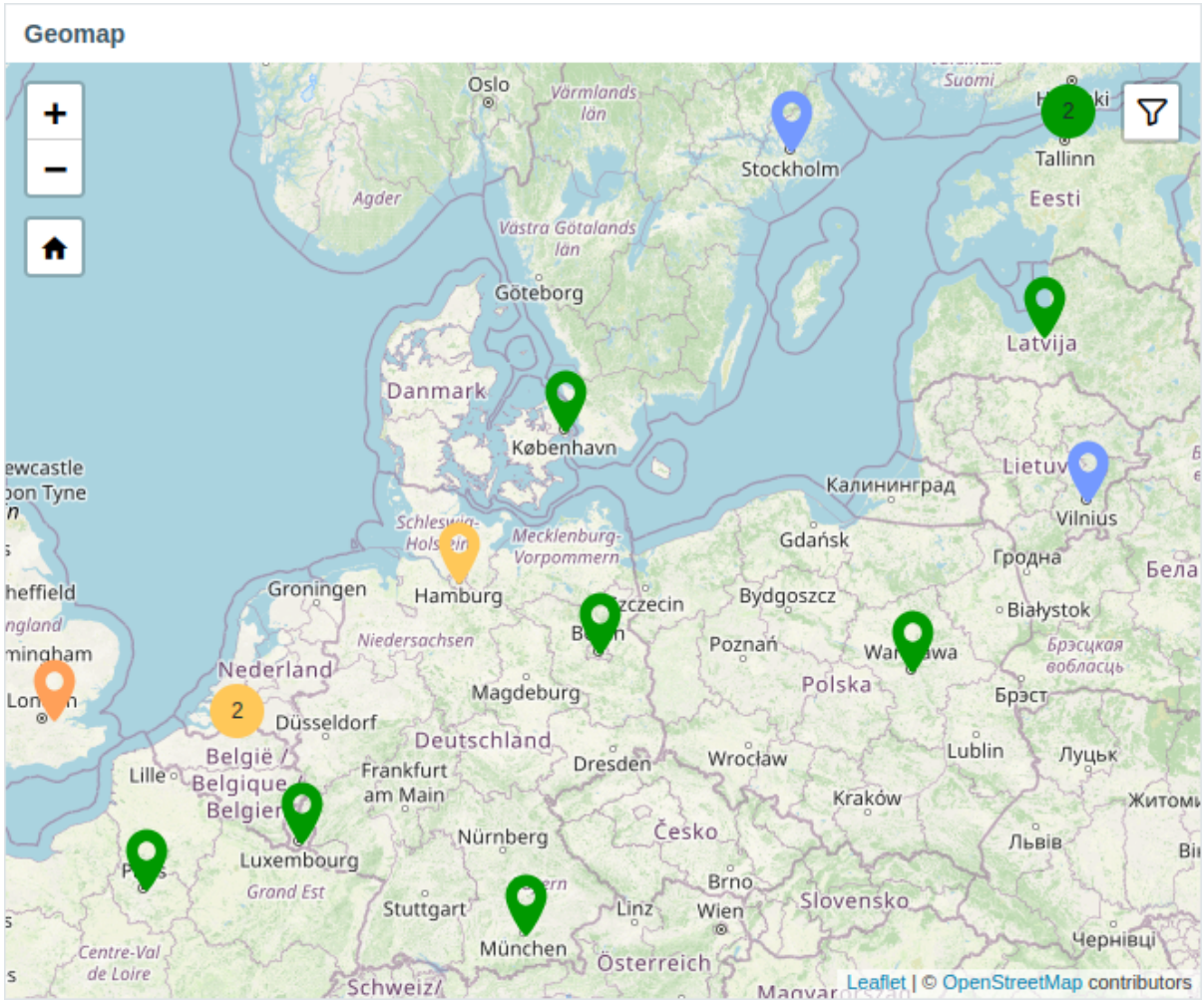
Ângulo	Selecione o ângulo do gauge (180° ou 270°).
Descrição	
Descrição	Insira a descrição do item. Esta descrição pode substituir o nome padrão do item. Descrições multilinha são suportadas. É possível uma combinação de texto e macros suportadas. {HOST.*}, {ITEM.*}, {INVENTORY.*} e macros de usuário 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).

Geomap

Visão geral

O widget *Geomap* exibe hosts como marcadores em um mapa interativo baseado no Leaflet, com as cores dos marcadores indicando o problema mais grave de cada host.

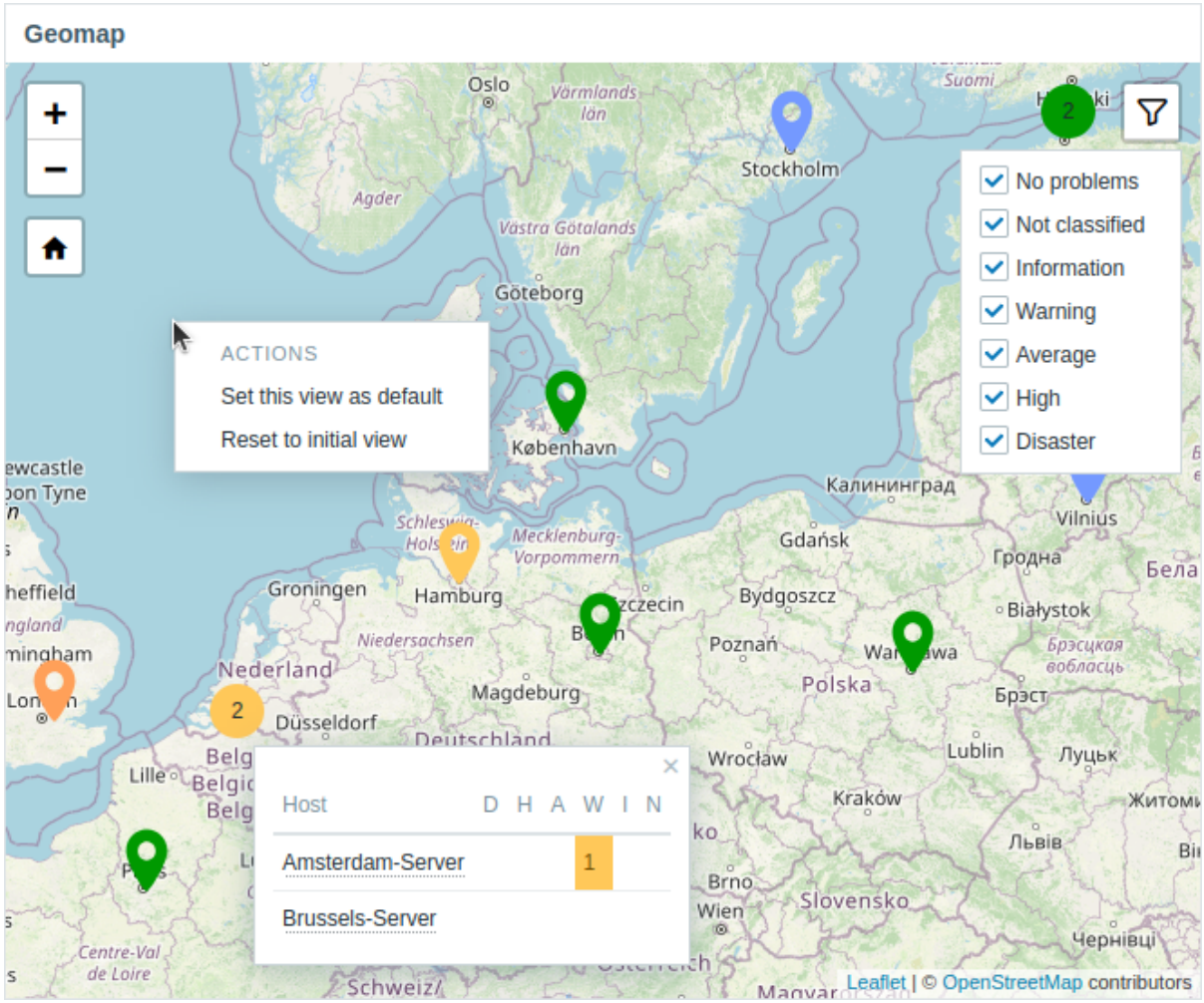


Note:
O Zabbix oferece vários provedores de mapas pré-definidos e permite adicionar provedores personalizados em *Administração* > *Geral* > *Mapas geográficos*.

Por padrão, o widget exibe todos os hosts habilitados que possuem coordenadas válidas (latitude de -90 a 90, longitude de -180 a 180) em seu inventário.

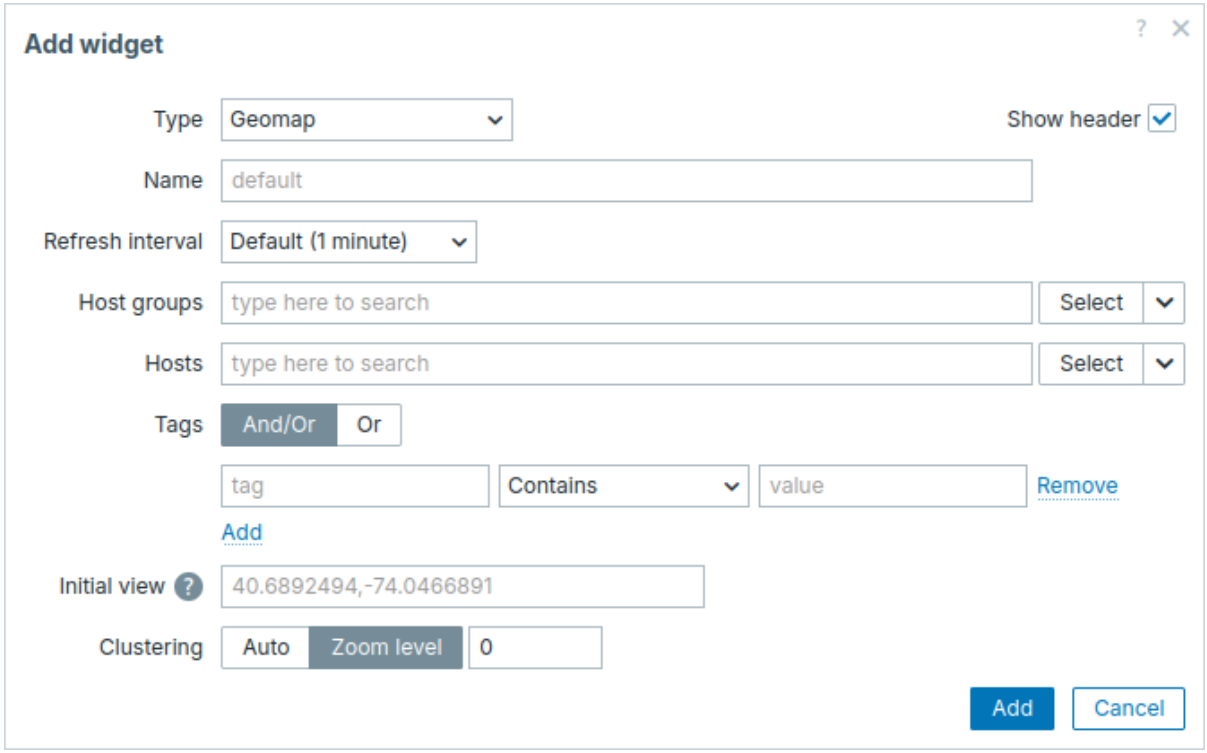
Use as seguintes opções para interagir com o widget:

- Clicar em um marcador exibe o nome do host e os 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:

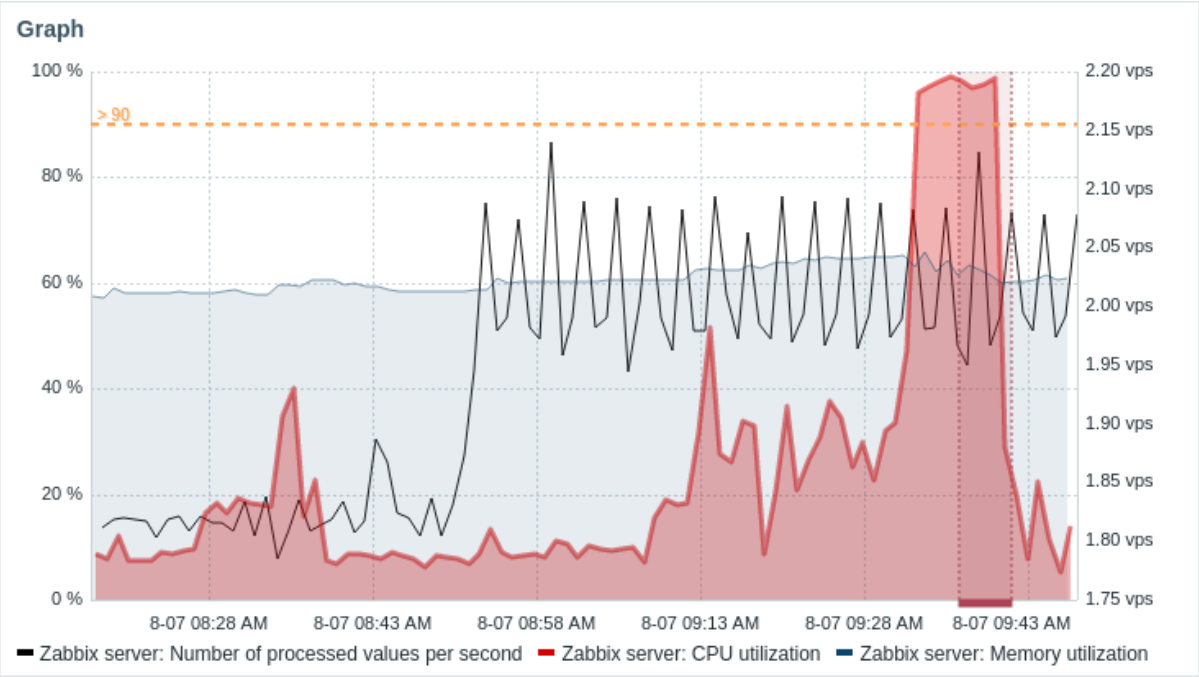
Grupos de hosts	Selecione os grupos de hosts a serem exibidos no mapa. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo possui preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será oferecida uma lista suspensa com os grupos correspondentes. Se nada for selecionado nos campos *Grupos de hosts* e *Hosts*, todos os hosts com coordenadas válidas serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione os hosts a serem exibidos no mapa. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo possui preenchimento automático, portanto, ao começar a digitar o nome de um host, será oferecida uma lista suspensa com os hosts correspondentes. Se nada for selecionado nos campos *Grupos de hosts* e *Hosts*, todos os hosts com coordenadas válidas serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tags	Especifique tags para limitar o número de hosts exibidos no widget. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tag especificados; **Igual a** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tag especificados em que os valores da tag contêm a string inserida (correspondência por substring, sem diferenciar maiúsculas de minúsculas); **Não existe** - exclui os nomes de tag especificados; **Diferente de** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tag especificados em que os valores da tag contêm a string inserida (correspondência por substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição *Or*; **Or** - 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 a ser exibido quando o widget for carregado inicialmente, no formato `<latitude>,<longitude>,<zoom>` Se o zoom inicial for especificado, o widget Geomap será carregado no nível de zoom informado. Caso contrário, o zoom inicial será calculado como metade do zoom máximo para o provedor de blocos específico. A visualização inicial será ignorada se a visualização padrão estiver definida (veja abaixo). Exemplos: 40.6892494,-74.0466891 40.6892494,-122.0466891
Agrupamento	Especifique como 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 - afastamento máximo, 30 - aproximação máxima). Por exemplo, se o parâmetro for definido como 0, os marcadores não serão combinados em nenhum nível de zoom; se for definido como 15, os marcadores serão combinados desde o afastamento máximo até o nível de zoom 15; se for definido como 30, os marcadores serão combinados em todos os níveis de zoom. Observe que o nível máximo de aproximação é determinado pelo provedor de blocos do mapa.

Gráfico

Visão geral

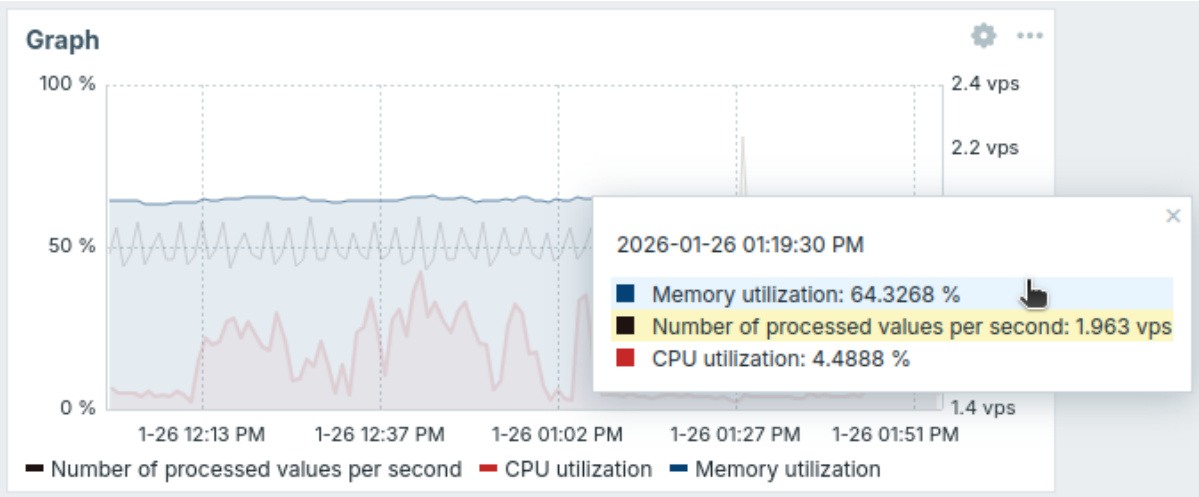
O widget *Graph* exibe dados numéricos de itens como um gráfico baseado em vetores. Ele pode ajudar a acompanhar tendências,

identificar problemas e comparar valores ao longo do tempo e entre hosts.

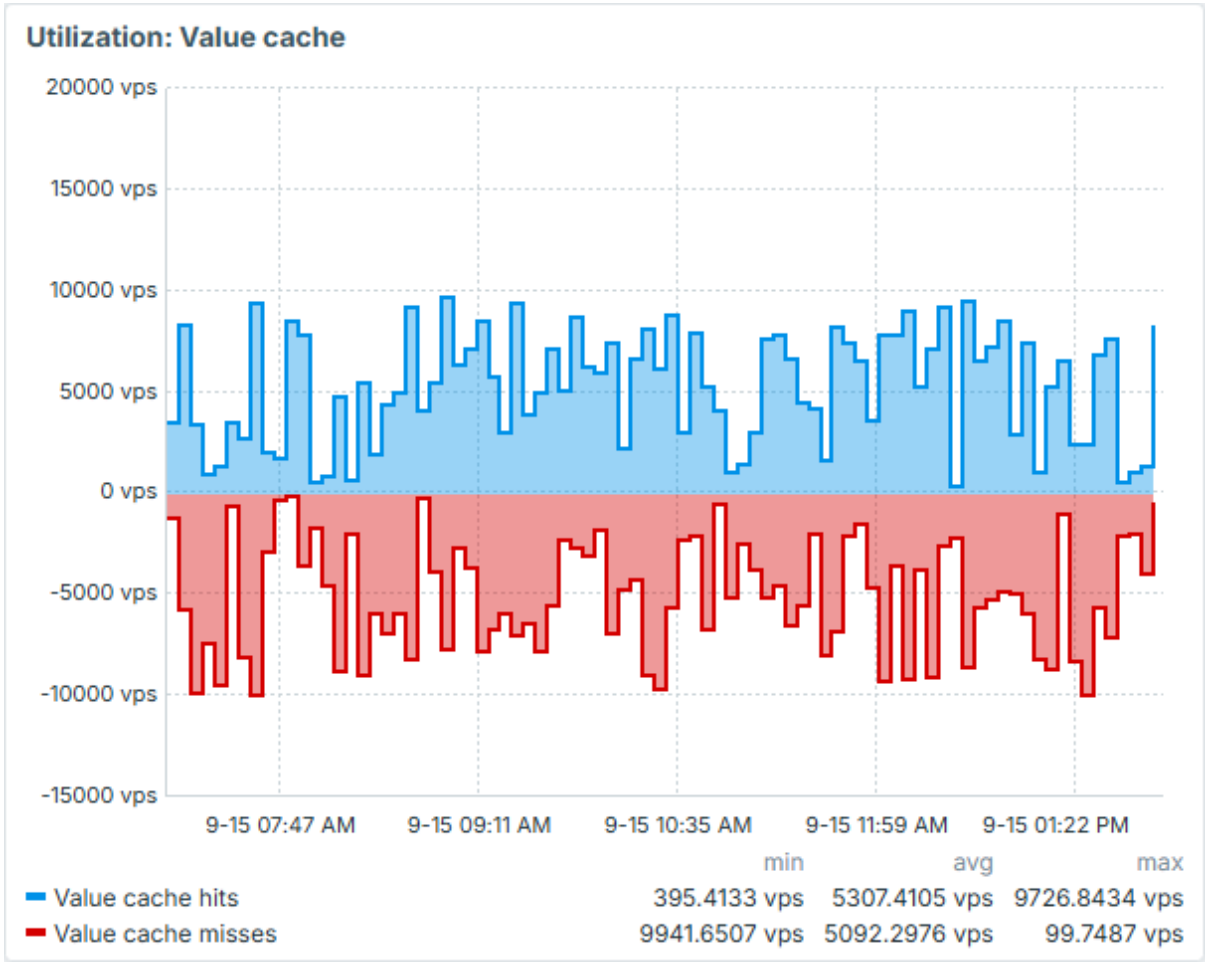


Ao passar o cursor sobre o gráfico, é exibida uma dica de ferramenta com uma lista de itens classificados por valor em ordem decrescente.

Ao passar o cursor sobre um item na dica de ferramenta, o gráfico correspondente é destacado (os demais ficam esmaecidos); ao selecionar o item, seus dados são transmitidos para outros widgets.



O gráfico pode ser exibido com os valores do eixo Y invertidos; isso afeta apenas a representação visual e não altera os valores originais.

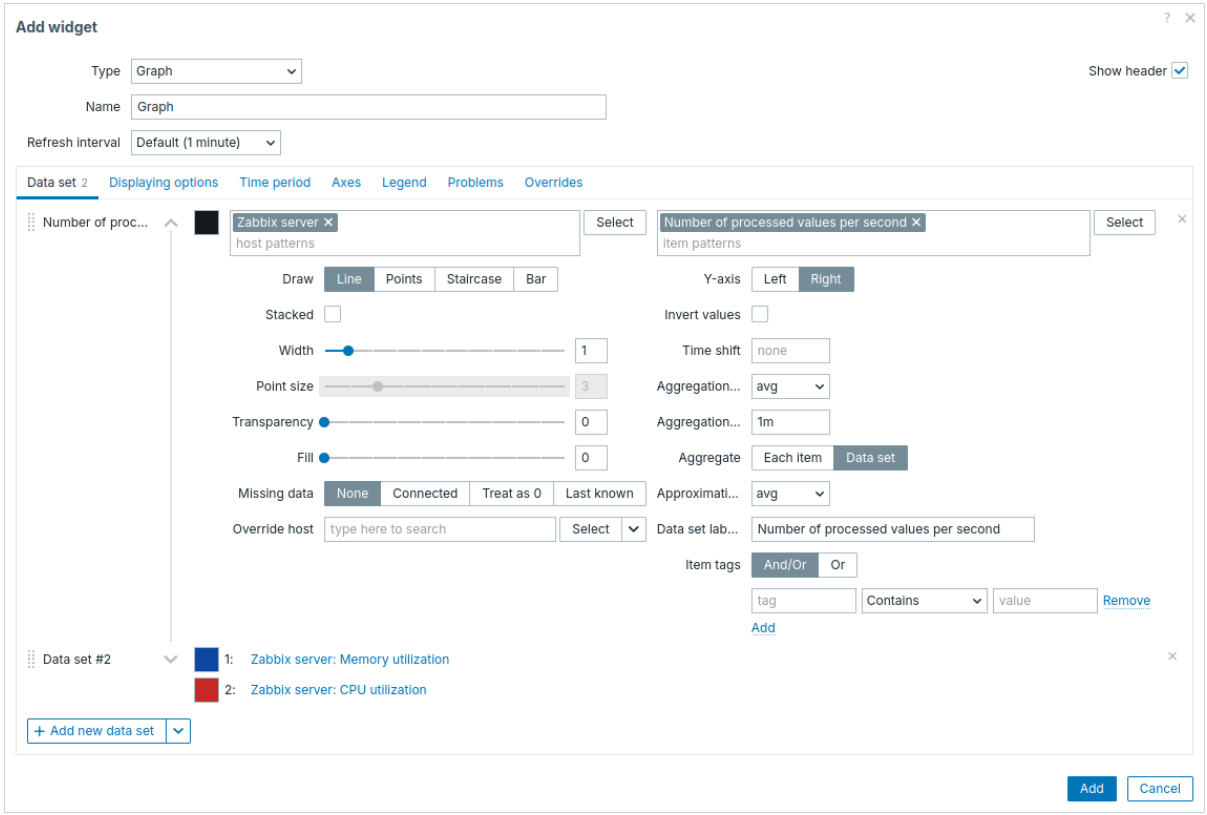


O widget legado Graph (classic) também está disponível.

As informações exibidas no widget *Graph* podem ser baixadas como uma imagem PNG selecionando a opção *Download image* no menu *Actions* ⋯ do widget, que aparece quando você passa o cursor sobre o widget. Dependendo das configurações do navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Save* é aberta, permitindo escolher o local.

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. É possível adicionar dois tipos de conjuntos de dados:

- *Padrões de item* - os dados dos itens correspondentes são exibidos. Você pode escolher uma única cor base ou selecionar uma linha da paleta para atribuir cores distintas a cada item correspondente.
- *Lista de itens* - os dados dos itens selecionados são exibidos. Você pode escolher individualmente a cor de cada item no seletor.

Por padrão, um conjunto de dados de *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 corresponderem a esses padrões serão exibidos no gráfico; até 50 itens podem ser exibidos. Padrões curinga podem ser usados para a 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 individualmente, por exemplo, um item chamado *item** se houver outros itens correspondentes (por exemplo, *item2*, *item3*). A especificação de padrões de host e de item é obrigatória para conjuntos de dados de ”Padrões de item”. Consulte também: Detalhes da configuração do conjunto de dados. Para o conjunto de dados **Lista de itens**: Selecione os itens para o gráfico clicando no botão *Adicionar item*. Você também pode selecionar widgets compatíveis como a fonte de dados dos itens clicando no botão *Adicionar widget*. A especificação de itens ou widgets é obrigatória para conjuntos de dados de ”Lista de itens”. Consulte também: Detalhes da configuração do conjunto de dados. Observe que somente 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 apenas a seleção de itens configurados no template.

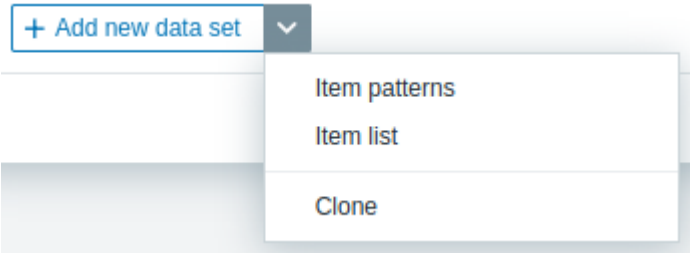
Desenho	Escolha o tipo de desenho da métrica. Tipos de desenho possíveis: *Linha* (definido por padrão), *Pontos*, *Escada* e *Barras*. Observe que, se houver apenas um ponto de dados no gráfico de linha/escada, ele será desenhado como um ponto, independentemente do tipo de desenho. O tamanho do ponto é calculado a partir da largura da linha, mas não pode ser menor que 3 pixels, mesmo que a largura da linha seja inferior.
Empilhado	Marque a caixa de seleção para exibir os dados empilhados (áreas preenchidas). Essa opção fica desabilitada quando o tipo de desenho *Pontos* é selecionado.
Largura	Defina a largura da linha. Essa opção está disponível quando o tipo de desenho *Linha* ou *Escada* é selecionado.
Tamanho do ponto	Defina o tamanho do ponto. Essa opção está disponível quando o tipo de desenho *Pontos* é selecionado.
Transparência	Defina o nível de transparência.
Preenchimento	Defina o nível de preenchimento. Essa opção está disponível quando o tipo de desenho *Linha* ou *Escada* é selecionado.
Dados ausentes	Selecione a opção para exibir dados ausentes: **Nenhum** - o intervalo permanece vazio; **Conectado** - os 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 aos tipos de desenho *Pontos* e *Barras*.
Substituir host	Selecione um widget compatível ou o seletor de host do dashboard como a fonte de dados para hosts. Esse 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.
Inverter valores	Marque a caixa de seleção para multiplicar todos os valores do eixo Y por -1 antes de exibi-los no gráfico. Isso afeta apenas a representação visual; os dados originais permanecem inalterados. Essa opção não afeta a dica de ferramenta, a legenda ou a linha de percentil. Se a opção *Triggers simples* for selecionada para o eixo, os limites do trigger serão exibidos invertidos.
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 para cada item ou para um conjunto de dados inteiro dentro do *Intervalo de agregação* selecionado: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **sum** - exibe a soma dos valores; **count** - exibe a quantidade de valores; **first** - exibe o primeiro valor; **last** - exibe o último valor. Se **não usado** for selecionado, todos os valores serão exibidos (sem agregação). Apenas dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em numéricos. Consulte também: Agregação em gráficos.
Intervalo de agregação	Especifique o intervalo para agregar os 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 *Automático*), 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 gerar resultados difíceis de interpretar.
Agregar	Especifique se os dados devem ser agregados: **Cada item** - cada item do 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 os valores mínimo, máximo e médio; **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio. Essa configuração é útil ao exibir um gráfico para um período longo com um 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 exibido na configuração *Conjunto de dados* do gráfico e na *Legenda* do gráfico (para conjuntos de dados agregados). Todos os conjuntos de dados são numerados, inclusive 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 nº 2”, ”Conjunto de dados nº 3” etc.). A numeração dos conjuntos de dados é recalculada após a reordenação/arraste dos conjuntos de dados. Rótulos de conjuntos de dados muito longos serão encurtados para caber no local em que são exibidos (por exemplo, ”Número de proc...”).
Tags de item	Especifique tags para filtrar os itens exibidos no widget. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência de nomes de tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tag especificados; **É igual a** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tag especificados quando os valores de tag contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Não existe** - exclui os nomes de tag especificados; **Não é igual a** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tag especificados quando os valores de tag contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

Detalhes da configuração do conjunto de dados

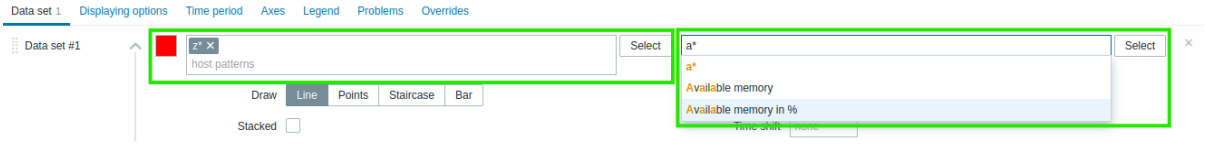
Os conjuntos de dados existentes são exibidos em uma lista. Você pode:

- Clicar no ícone  de movimentação e arrastar um conjunto de dados para um novo local na lista.
- Clicar no ícone de expansão para expandir os detalhes do conjunto de dados. Quando expandido, esse ícone se transforma em um ícone  de recolhimento.
- Clicar no ícone  de cor para abrir o seletor. Você pode inserir um código hexadecimal, escolher uma amostra de cor sólida ou alternar para a aba *Paleta* e selecionar uma linha de cores predefinidas. A cor escolhida é aplicada diretamente aos conjuntos de dados *Item list* ou usada como base para gerar tonalidades em *Item patterns*. Use Tab para alternar entre os controles da caixa de diálogo, as teclas de seta para navegar pelas amostras ou linhas da paleta, Enter para selecionar e Esc para cancelar.
- Clicar no botão *Add new data set* para adicionar um conjunto de dados vazio que permite selecionar padrões de host e item. Se você clicar no ícone apontando para baixo ao lado do botão *Add new data set*, será exibido um menu suspenso que permite adicionar um novo conjunto de dados *Item patterns* ou *Item list*, ou *Clone* o conjunto de dados aberto no momento. Se todos os conjuntos de dados estiverem recolhidos, a opção *Clone* não estará disponível.

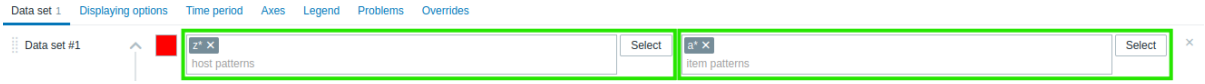


O conjunto de dados **Padrões de item** contém os campos *Padrões de host* e *Padrões de item* que reconhecem nomes completos ou padrões contendo um caractere curinga (*). Essa funcionalidade permite selecionar todos os nomes de host e nomes de item que contenham o padrão selecionado. Ao digitar o nome 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.

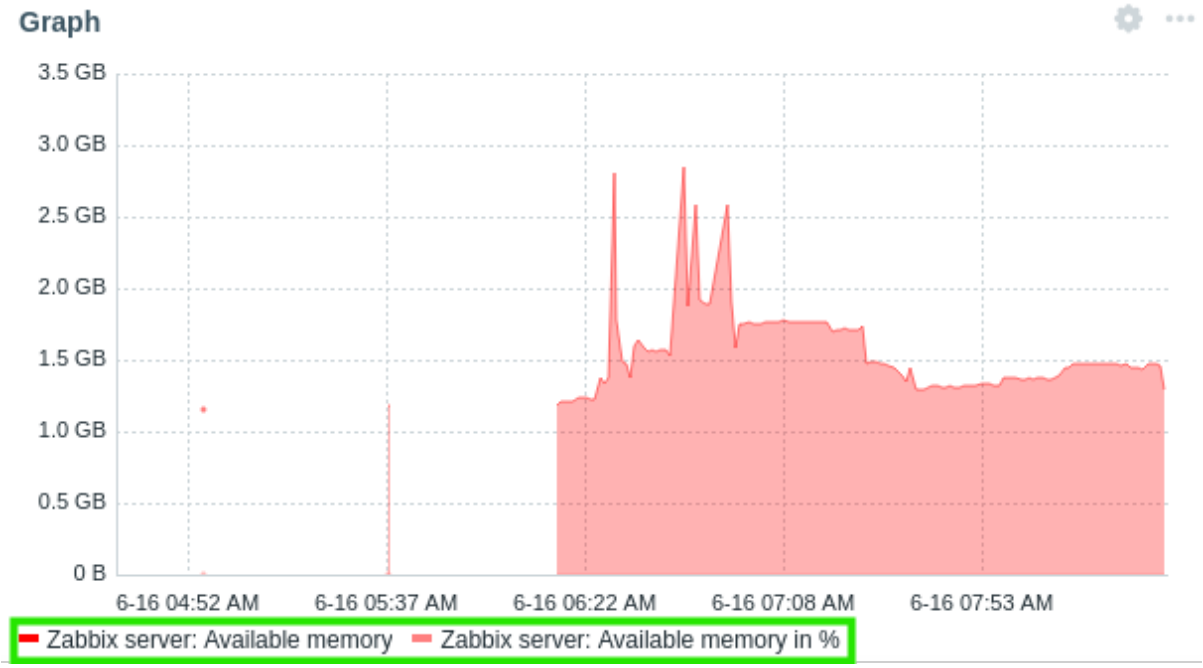
Por exemplo, ao digitar o padrão **z*** no campo *Padrões de host*, a lista suspensa exibe todos os nomes de host que contêm esse padrão: **z*, Zabbix server, Zabbix proxy**. Após pressionar *Enter*, esse padrão é aceito e exibido como **z***. Da mesma forma, ao digitar o padrão **a*** no campo *Padrões de item*, a lista suspensa exibe todos os nomes de item que contêm esse padrão: **a*, Available memory, Available memory in %**.



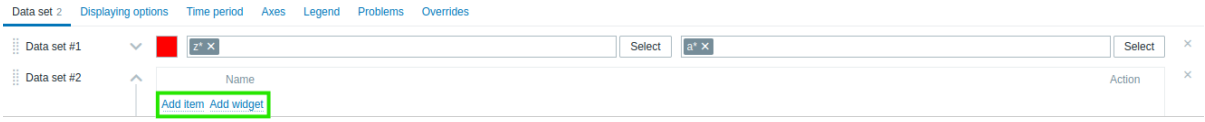
Após pressionar *Enter*, o padrão é aceito e exibido como **a***.



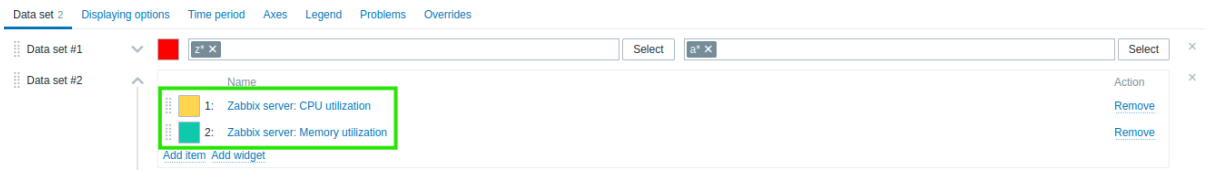
O gráfico então exibe todos os itens pertencentes ao(s) nome(s) de host selecionado(s).



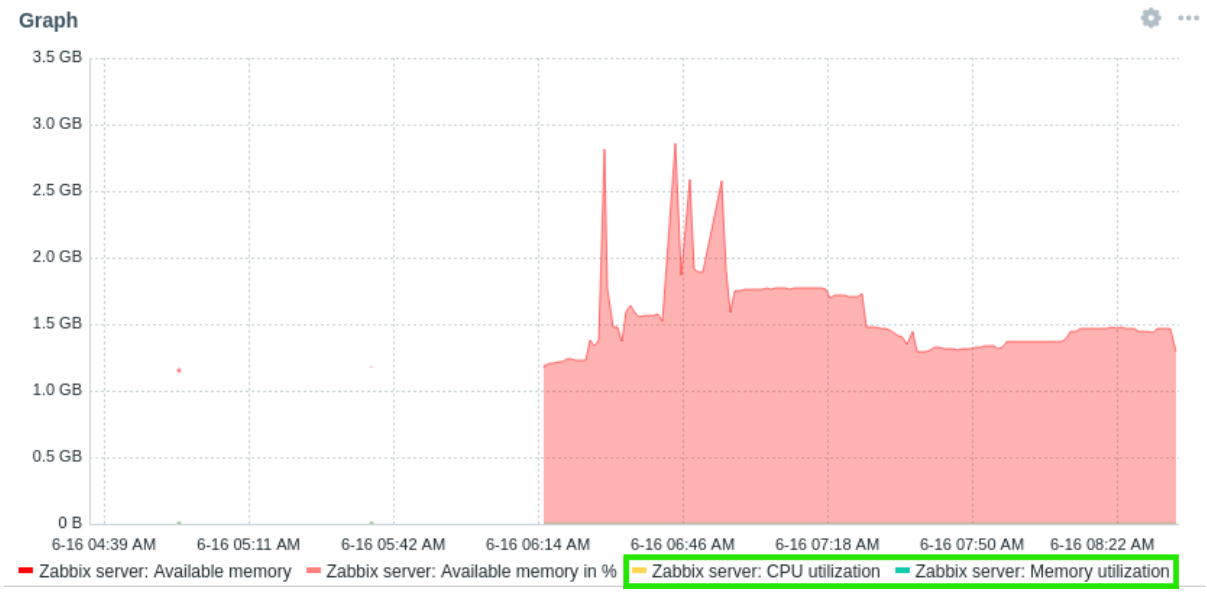
O conjunto de dados **Lista de items** contém o botão *Adicionar item* que permite adicionar items para serem exibidos no gráfico. Você também pode adicionar widgets compatíveis como fonte de dados para items clicando no botão *Adicionar widget*.



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.

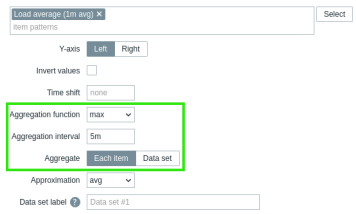


Agregação em gráficos

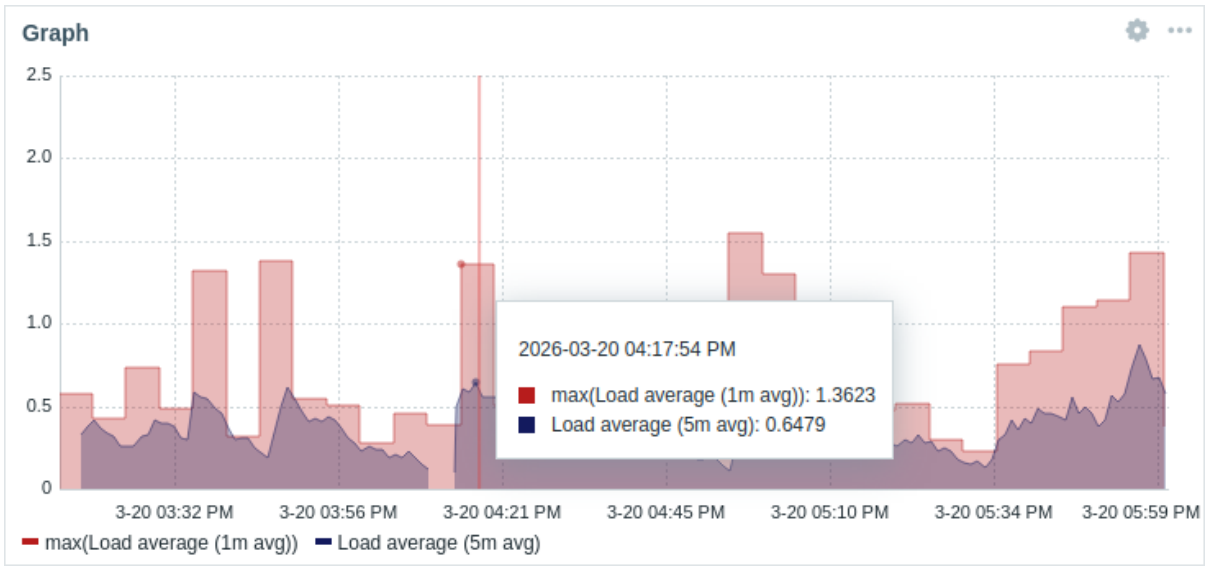
A agregação permite exibir um valor agregado (mínimo, máximo, média etc.) para o intervalo especificado (5 minutos, uma hora, um dia etc.), em vez de todos os valores individuais. Em um conjunto de dados, a agregação pode ser aplicada a cada item separadamente ou ao conjunto de dados inteiro.

Alguns casos de uso da função de agregação podem ser: - A contagem média diária de solicitações a um server NGINX. - O menor espaço em disco entre clusters por semana.

No exemplo a seguir, a função de agregação ”max” é aplicada ao item com um intervalo de agregação de cinco minutos:



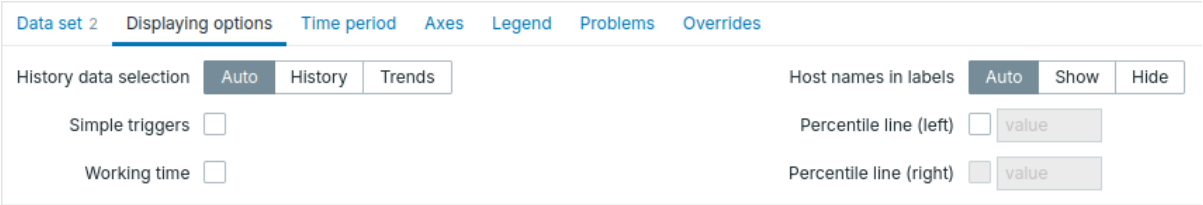
O exemplo demonstra a carga máxima da CPU ao longo de cinco minutos (agregada a partir de valores médios de um minuto), em comparação com a carga média da CPU ao longo de cinco minutos.



Quando *Show aggregation function* é selecionado para a legenda, os itens são exibidos entre parênteses, precedidos pela função de agregação na legenda e na dica de ferramenta. Se *Data set label* for especificado para um gráfico, o rótulo será exibido entre parênteses, precedido pela função de agregação.

Opções de exibição

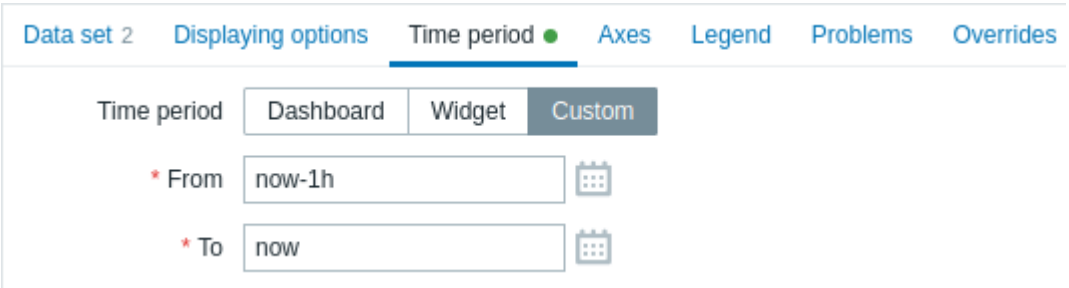
A aba **Opções de exibição** permite definir a seleção de dados históricos:



Seleção de dados históricos	Defina a fonte dos dados do gráfico: **Automático** - os dados são obtidos de acordo com o algoritmo clássico do gráfico (padrão); **Histórico** - dados do histórico; **Tendências** - dados das tendências.
Triggers simples	Marque a caixa de seleção para mostrar os limites dos triggers simples. Os limites serão desenhados como linhas tracejadas usando a cor da severidade do trigger. Um trigger simples é um trigger com uma função (somente `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 ficar 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 com um fundo branco, enquanto o período fora do horário de trabalho é exibido em cinza (com o tema frontend padrão *Original blue*).
Nomes dos hosts nos rótulos	Escolha se os nomes dos hosts devem ser exibidos na legenda do gráfico e na dica de ferramenta que aparece ao passar o cursor sobre os dados do gráfico: **Automático** - os nomes dos hosts são exibidos somente quando há mais de um host nos conjuntos de dados (padrão). **Mostrar** - os nomes dos hosts são exibidos. **Ocultar** - os nomes dos hosts ficam ocultos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
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. Por exemplo, se um percentil de 95% for definido, a linha do percentil ficará no nível abaixo do qual estão 95% dos valores.
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. Por exemplo, se um percentil de 95% for definido, a linha do percentil ficará no nível abaixo do qual estão 95% dos valores.

Período de tempo

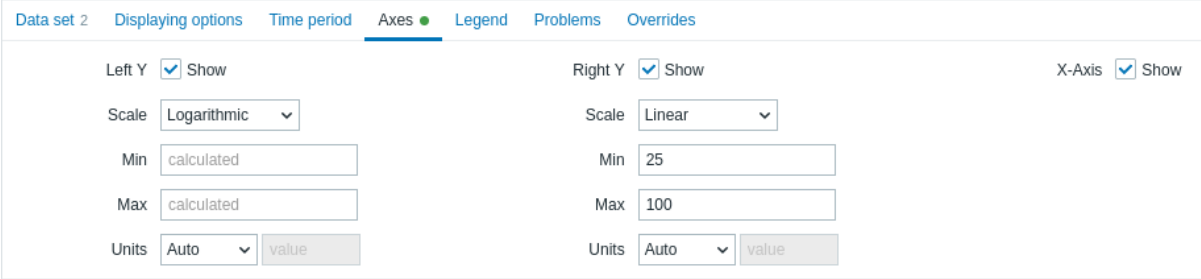
A aba **Período de tempo** permite definir um período de tempo para o qual os dados serão exibidos no gráfico:



Período de tempo	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*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido ao passar o mouse sobre ele. Observe que, independentemente da configuração de *Período de tempo* do widget, os 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 quando *Período de tempo* está definido como "Widget".
De	Insira ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".
Até	Insira ou selecione o fim do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".

Eixos

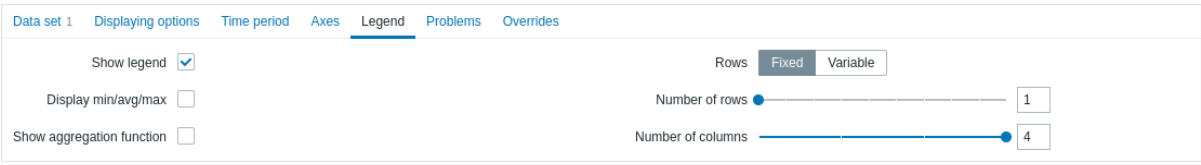
A aba **Eixos** permite personalizar como os eixos são exibidos:



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 dos eixos do gráfico na lista suspensa: **Linear** - os valores do eixo aumentam por uma quantidade fixa (por exemplo, 10, 20, 30), sendo adequada para dados que mudam de forma constante ou abrangem um intervalo pequeno a moderado; **Logarítmica** - os valores do eixo aumentam exponencialmente (por exemplo, 10, 100, 1000), sendo adequada para dados que mudam rapidamente ou abrangem um intervalo amplo.
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 dos eixos do gráfico na lista suspensa: **Automático** - os valores dos eixos são exibidos usando a unidade do primeiro item no conjunto de dados; **Estático** - os valores dos eixos são exibidos usando a unidade especificada no campo de entrada *valor*; se o campo for deixado em branco, somente os valores numéricos serão exibidos.

Legenda

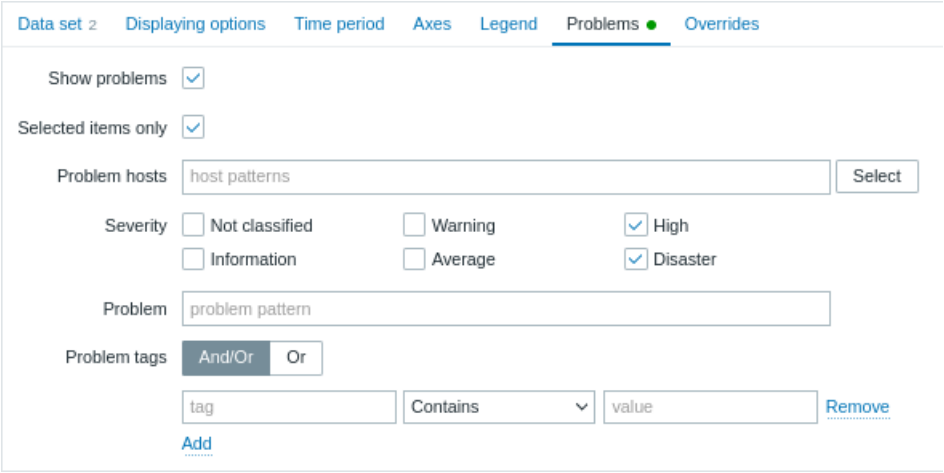
A aba **Legenda** permite personalizar a legenda do gráfico:



Mostrar legenda	Desmarque esta caixa de seleção para ocultar a legenda no gráfico (marcada por padrão).
Exibir mínimo/média/máximo	Marque esta caixa de seleção para exibir os valores mínimo, médio e máximo do item na legenda.
Mostrar função de agregação	Marque esta caixa de seleção para exibir a função de agregação na legenda.
Linhas	Selecione o modo de exibição das linhas da legenda: **Fixo** - o número de linhas exibidas é determinado pelo valor do parâmetro *Número de linhas*; **Variável** - o número de linhas exibidas é determinado pela quantidade de itens 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 estará disponível se *Exibir mínimo/média/máximo* estiver desmarcado.

Problemas

A aba **Problemas** permite personalizar a exibição de problemas:

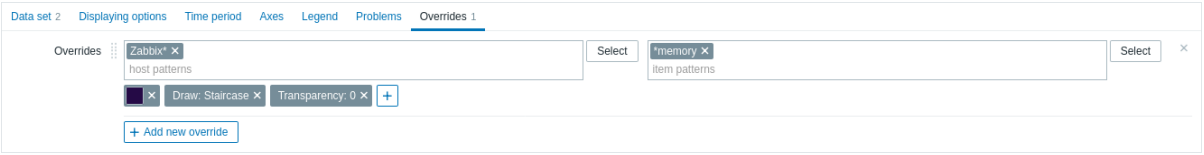


Mostrar problemas	Marque esta caixa de seleção para habilitar a exibição de problemas no gráfico (desmarcada, ou seja, desabilitada por padrão).
Somente itens selecionados	Marque esta caixa de seleção para incluir no gráfico somente os problemas dos itens selecionados.
Hosts com problemas	Selecione os hosts com problemas que serão exibidos no gráfico. Padrões curinga podem ser usados (por exemplo, ∗ retornará resultados que correspondam a zero ou mais caracteres). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. Enquanto você digita, observe como todos os hosts correspondentes são exibidos na lista suspensa. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Severidade	Marque as severidades dos problemas para filtrar os problemas que serão exibidos no gráfico. Se nenhuma severidade for marcada, todos os problemas serão exibidos.
Problema	Especifique o nome do problema que será exibido no gráfico.

Tags de problema	Especifique as tags de problema para limitar o número de problemas exibidos no widget. É possível incluir e excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **É igual a** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual a** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições que têm 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 itens são selecionados para um conjunto de dados usando o caractere curinga * e você deseja alterar a forma como os itens são exibidos por padrão (por exemplo, a cor base padrão ou qualquer outra propriedade).

As substituições existentes, se houver, são exibidas em uma lista.
Para adicionar uma nova substituição:

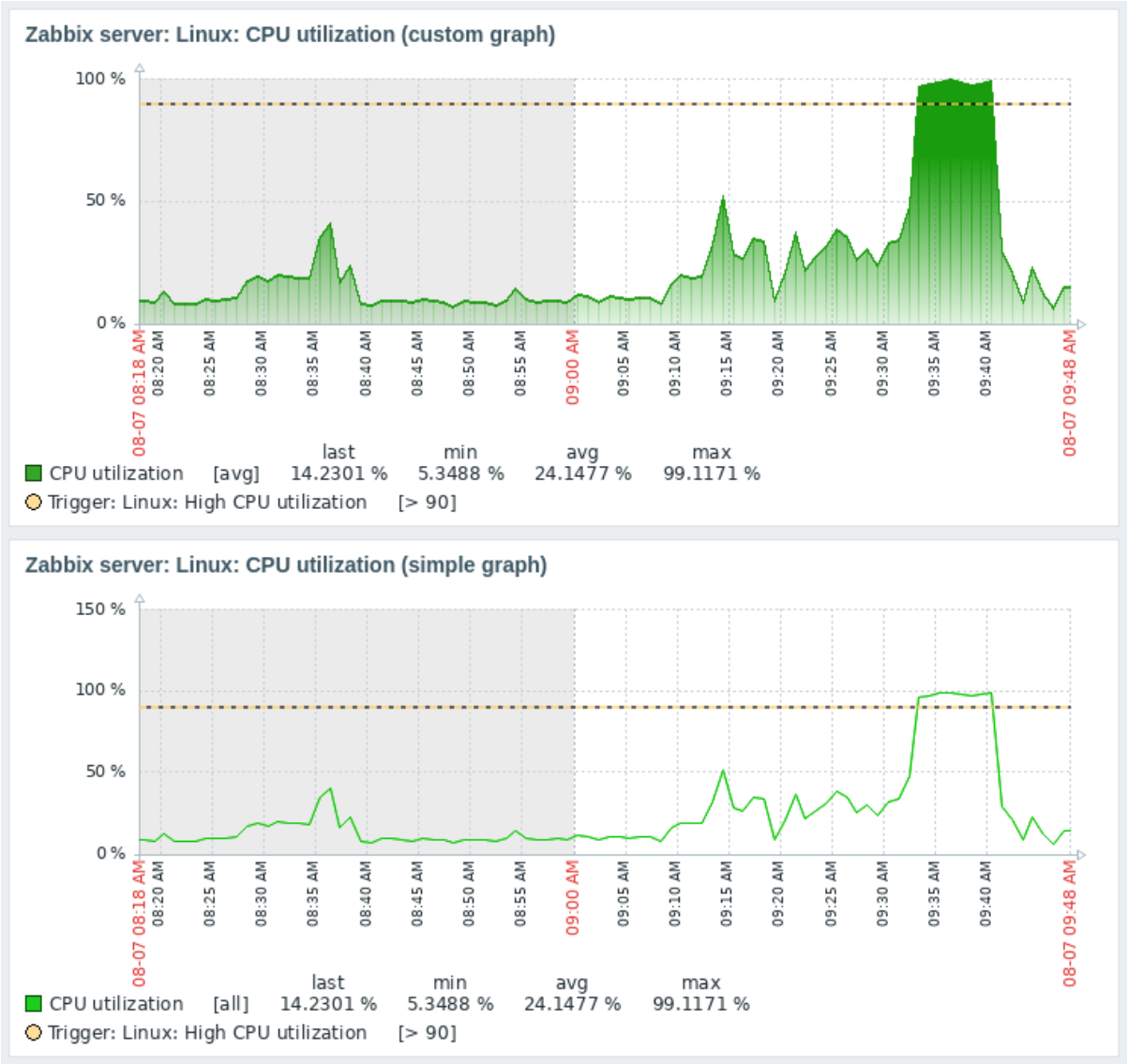
- Clique no botão 

- Selecione hosts e itens para a substituição.
  Como alternativa, você pode inserir padrões de host e item.
  Podem ser usados padrões curinga (por exemplo, * retornará resultados que correspondem a zero ou mais caracteres).
  Para especificar um padrão curinga, basta inserir a string manualmente e pressionar *Enter*.
  Enquanto você digita, observe como todos os hosts correspondentes são exibidos na lista suspensa.
  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).
  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 itens permite apenas a seleção de itens 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 obter descrições dos parâmetros, consulte a aba *Conjunto de dados* acima.

Gráfico (clássico)

Visão geral

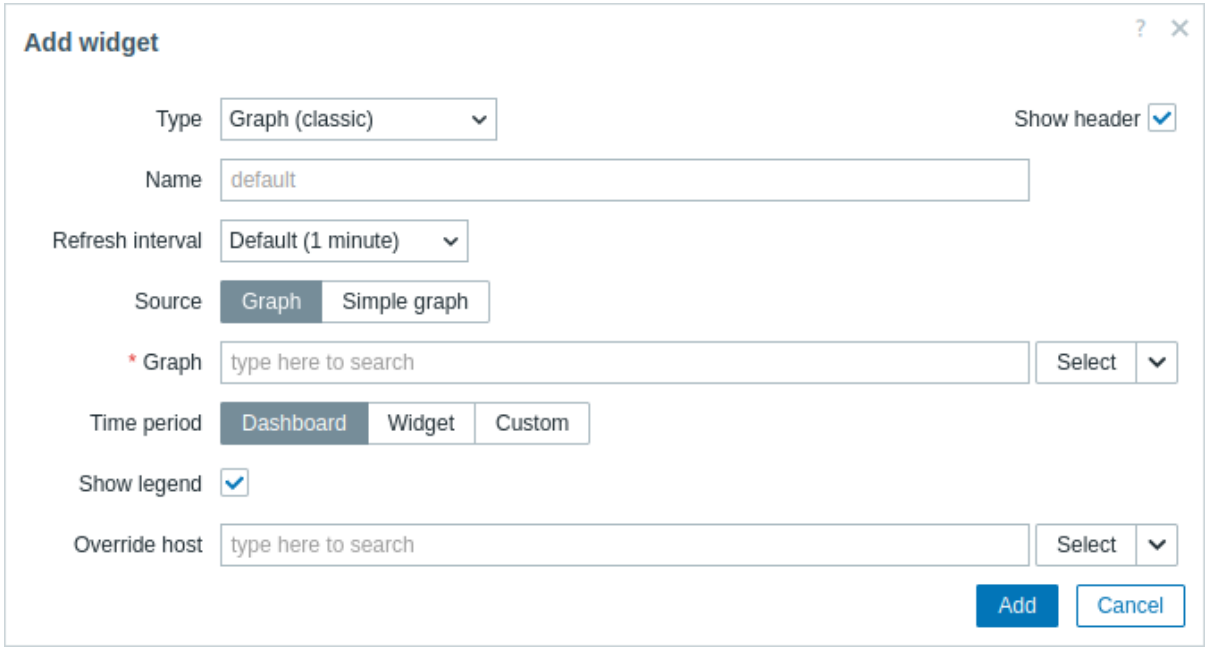
O widget *Graph (classic)* exibe dados numéricos de item como um gráfico personalizado ou gráfico simples baseado em imagem. Ele oferece suporte à seleção do período, linhas de trigger e exibição opcional de legenda.



As informações exibidas no widget *Graph (classic)* podem ser baixadas como uma imagem PNG selecionando a opção *Download image* no menu *Actions* do widget ⋮, que aparece quando você passa o cursor sobre o widget. Dependendo das configurações do seu navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Save* é aberta, permitindo escolher o local.

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. Como alternativa, 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. Como alternativa, selecione um widget compatível como fonte de dados para items. 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 do *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 de seleção 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:

- Items e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver algum associado aos items exibidos)

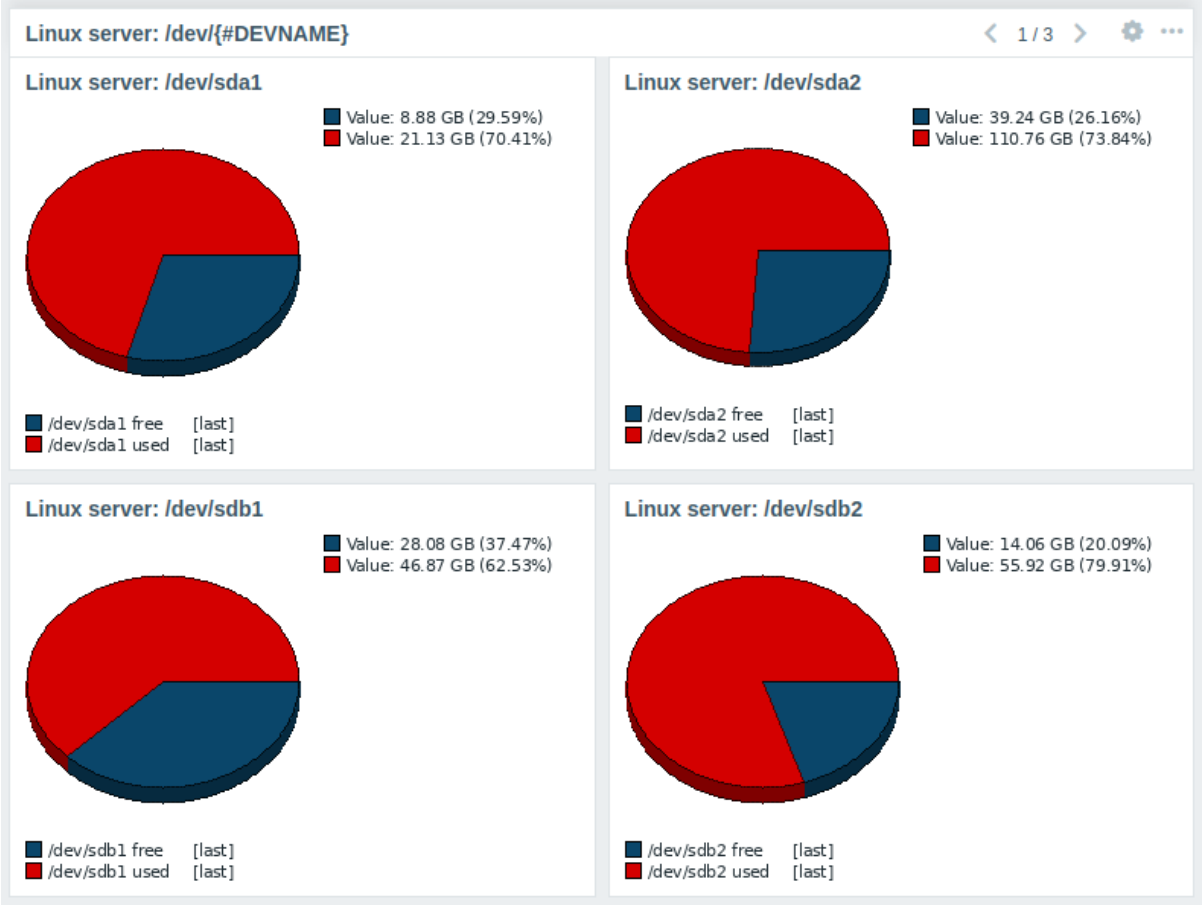
Se a altura do gráfico dentro do widget for insuficiente, a legenda pode não ser exibida ou pode ser exibida apenas parcialmente. Triggers e percentis são ocultados primeiro, seguidos pela legenda do item. Para exibir a legenda completa, aumente o tamanho vertical do widget.

Note:
Não mais que 3 linhas de trigger podem ser exibidas. Se houver mais triggers, as triggers com menor severidade terão prioridade para exibição.

Protótipo de gráfico

Visão geral

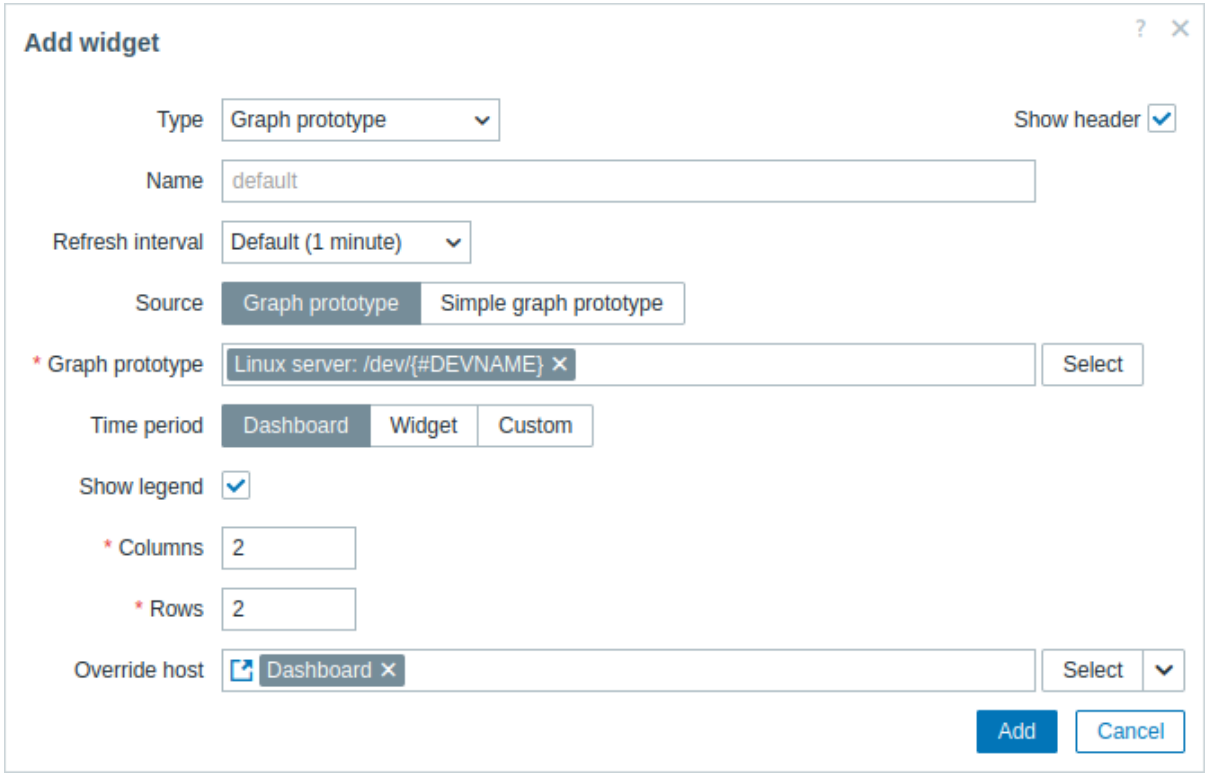
O widget *Protótipo de gráfico* exibe uma grade de gráficos personalizados criados automaticamente com base em protótipos de gráficos de regras de descoberta de baixo nível prototipos de gráficos ou prototipos de item.



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 *Graph prototype* como tipo de widget:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

Source	Selecione a origem dos gráficos: **Graph prototype** ou **Simple graph prototype**.
Graph prototype	Selecione um protótipo de gráfico para exibir os gráficos descobertos pelo protótipo de gráfico. Este parâmetro está disponível quando *Source* está definido como ”Graph prototype”.
Item prototype	Selecione um protótipo de item para exibir gráficos simples dos itens descobertos pelo protótipo de item. Este parâmetro está disponível quando *Source* está definido como ”Simple graph prototype”.
Time period	Defina um período para o qual os dados serão exibidos nos gráficos. Selecione a fonte de dados para o período: **Dashboard** - use o seletor de período do dashboard; **Widget** - use um widget compatível (definido no parâmetro *Widget*); **Custom** - use um período personalizado definido nos parâmetros *From* e *To*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido quando o mouse passar sobre ele. Observe que, independentemente da configuração de *Time period* do widget, os widgets compatíveis ainda podem usá-lo como fonte de dados para o período.
Widget	Insira ou selecione um widget compatível como fonte de dados para o período. Este parâmetro está disponível quando *Time period* está definido como ”Widget”.
From	Insira ou selecione o início do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Time period* está definido como ”Custom”.
To	Insira ou selecione o fim do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Time period* está definido como ”Custom”.
Show legend	Desmarque esta caixa de seleção para ocultar a legenda nos gráficos (marcada por padrão).
Override 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.
Columns	Insira o número de colunas de gráficos a serem exibidas em um widget de protótipo de gráfico.
Rows	Insira o número de linhas de gráficos a serem exibidas em um widget de protótipo de gráfico.

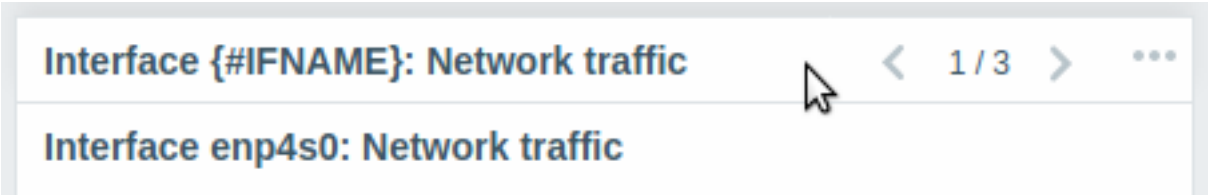
A legenda do gráfico consiste em três seções:

- Itens e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver algum associado aos itens exibidos)

Se a altura do gráfico dentro do widget for insuficiente, a legenda poderá não ser exibida ou poderá ser exibida apenas parcialmente. Triggers e percentis são ocultados primeiro, seguidos pela legenda dos itens. Para exibir a legenda completa, aumente o tamanho vertical do widget.

Note:
No máximo 3 linhas de trigger podem ser exibidas. Se houver mais triggers, aqueles com menor severidade terão prioridade na exibição.

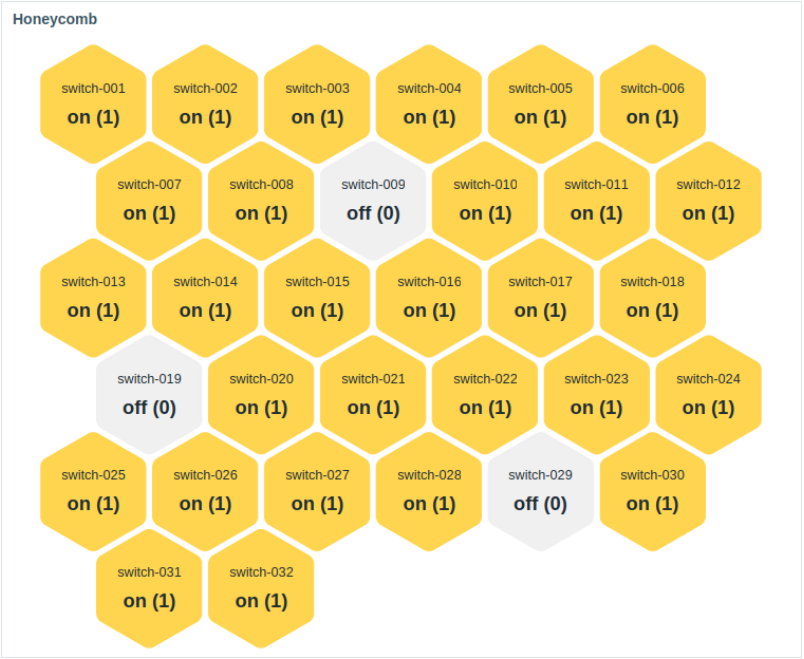
Embora os parâmetros *Columns* e *Rows* permitam ajustar mais de um gráfico no widget, ainda pode haver mais gráficos descobertos do que o número de colunas/linhas no widget. Nesse caso, a paginação fica disponível no widget, e um cabeçalho deslizante permite alternar entre as páginas usando as setas para a esquerda e para a direita:

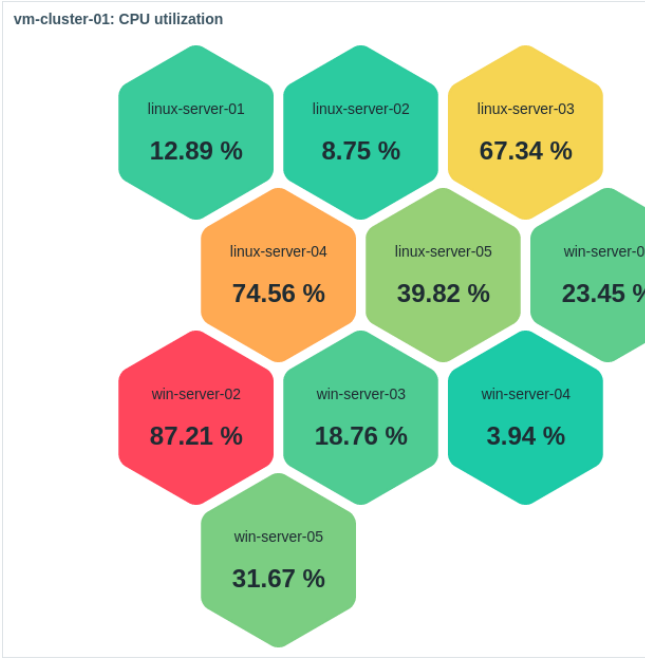


Honeycomb

Visão geral

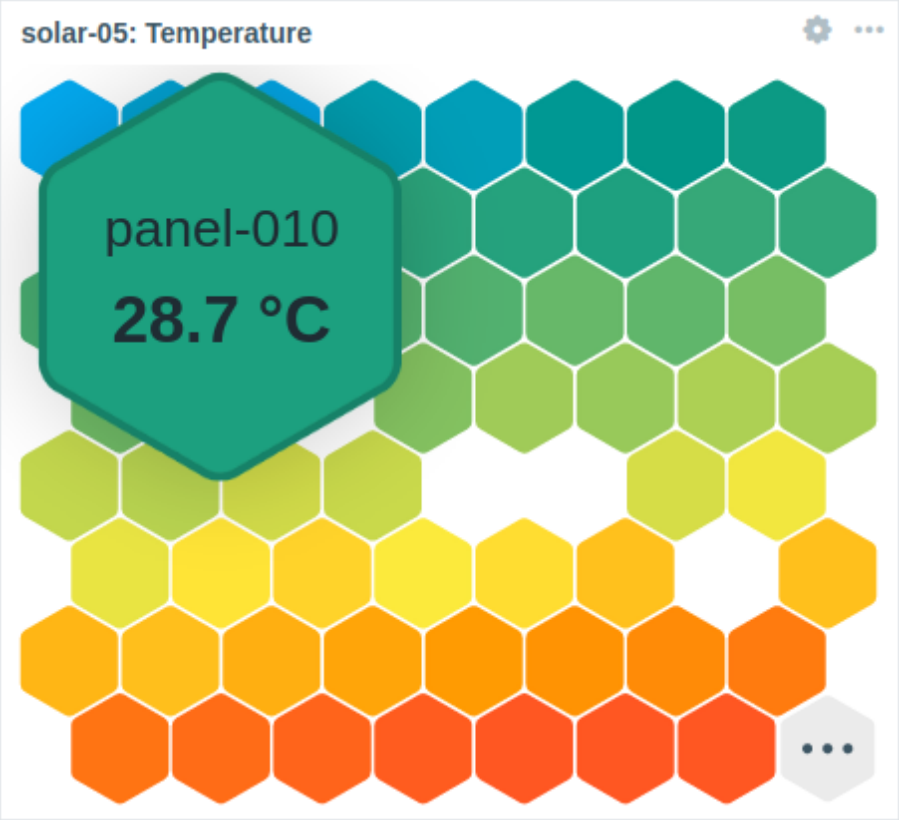
O widget *Honeycomb* exibe hosts ou itens como células hexagonais em uma grade. Esse layout facilita a detecção de hosts/itens com problemas, a identificação de agrupamentos de problemas, a comparação de grupos rapidamente e muito mais.





O número de células exibidas é limitado pelo tamanho do widget e pelo tamanho mínimo da célula (32px). Se não houver espaço para todas as células, a última célula exibirá reticências. O widget pode ser redimensionado, com a quantidade, o tamanho e o posicionamento das células sendo ajustados dinamicamente.

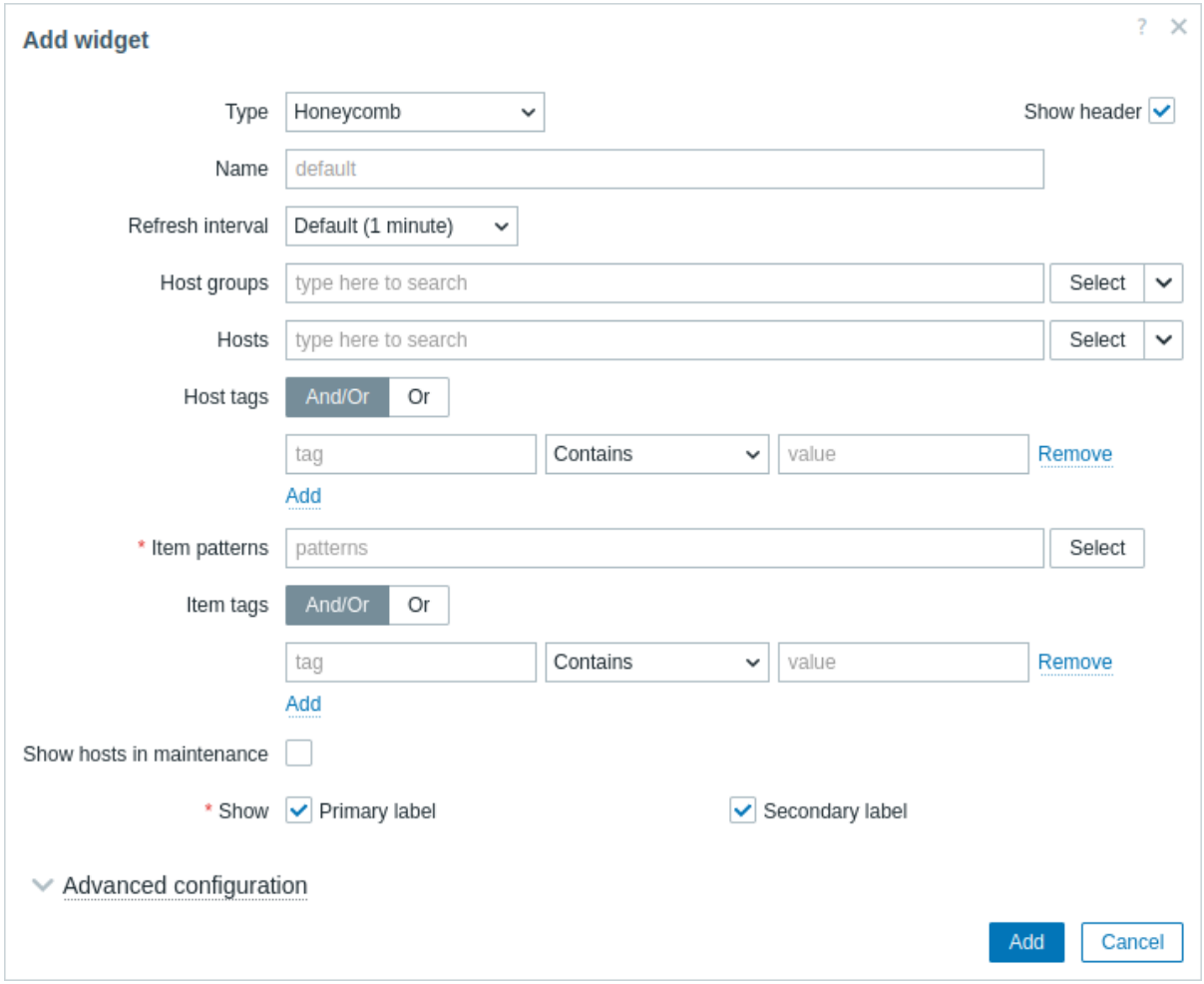
Ao passar o mouse sobre uma célula, a célula em foco aumenta para melhorar a visibilidade. Clicar em uma célula destaca sua borda até que outra célula seja selecionada.



As informações exibidas no widget *Honeycomb* podem ser baixadas como uma imagem PNG selecionando a opção *Baixar imagem* no menu *Ações* do widget ⋯, que aparece quando você passa o mouse sobre o widget. Dependendo das configurações do navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Salvar* é aberta, permitindo escolher o local.

Configuração

Para configurar, selecione *Honeycomb* como tipo:



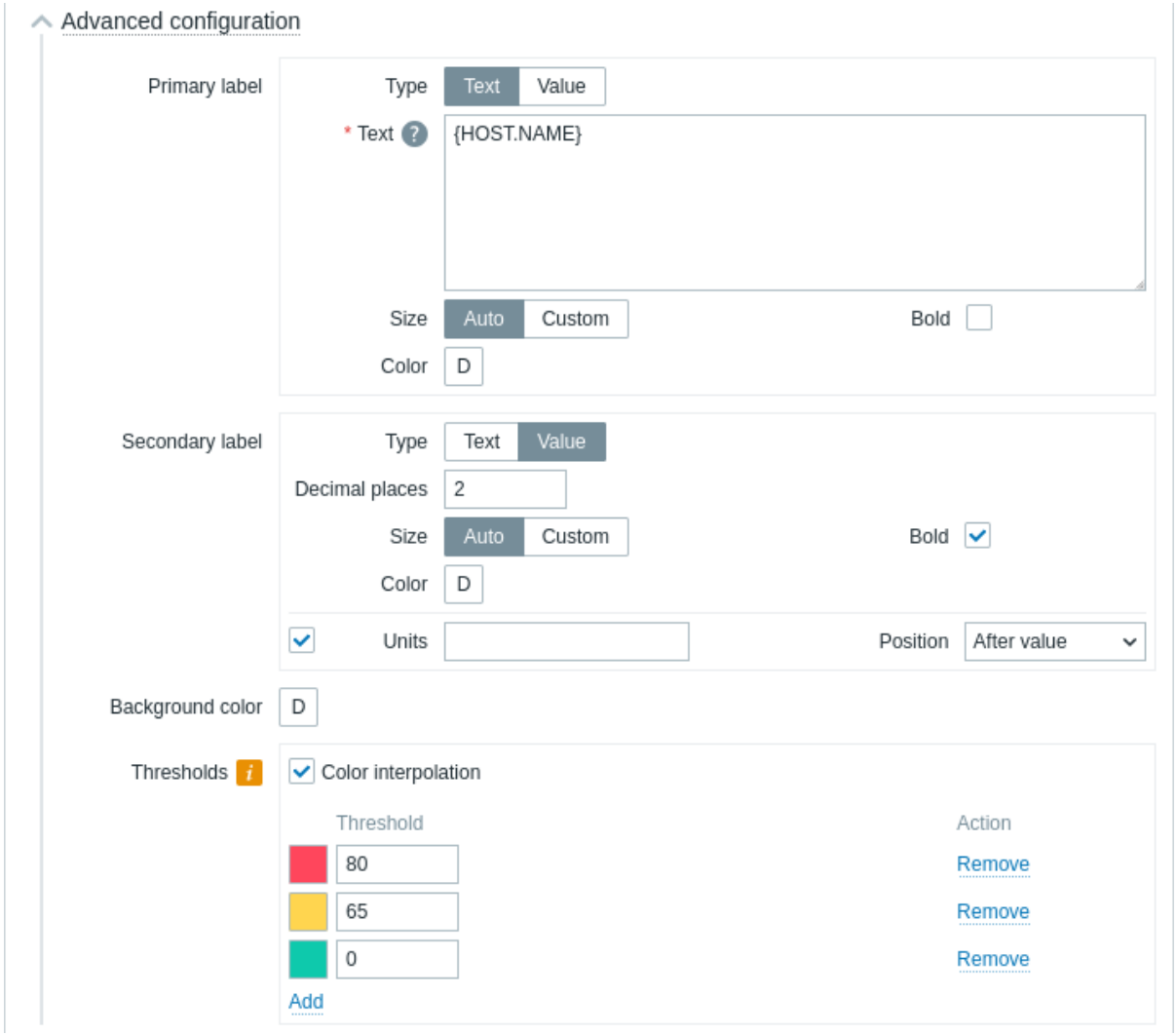
Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo possui preenchimento automático, 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 todos os grupos de hosts que contenham hosts com items correspondentes 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 possui preenchimento automático, 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 todos os hosts com items correspondentes 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, bem como excluir, tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - incluir os nomes de tag especificados; **Igual a** - incluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência por substring, sem diferenciar maiúsculas de minúsculas); **Não existe** - excluir os nomes de tag especificados; **Diferente de** - excluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Não contém** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência por substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - 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 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 a 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*. Quando você começar a digitar, uma lista suspensa mostrará os items correspondentes, limitada àqueles pertencentes aos *Hosts* selecionados ou aos 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, consulte *Tags de host* acima.
Mostrar hosts em manutenção	Marque esta caixa de seleção para exibir hosts em manutenção (neste caso, o ícone de manutenção será mostrado ao lado do nome do host). Este parâmetro é rotulado como *Mostrar dados em manutenção* ao configurar o widget em um dashboard de template.
Mostrar	Marque esta caixa de seleção para exibir o respectivo elemento da célula do honeycomb - rótulo principal, rótulo secundário. Pelo menos um elemento deve ser selecionado.
Configuração avançada	Clique no cabeçalho *Configuração avançada* para exibir as opções de configuração avançada.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada* e apenas para os elementos selecionados no campo *Mostrar* (veja acima), bem como para a cor de fundo ou os limites das células do favo de mel.



Rótulo primário/secundário	
Tipo	Selecione o tipo de rótulo: **Texto** - o rótulo exibirá o texto especificado para o parâmetro *Texto*; **Valor** - o rótulo exibirá o valor do item com as casas decimais especificadas para o parâmetro *Casas decimais*.
Texto	Insira o texto do rótulo. Esse texto pode substituir o nome padrão do item. Texto multilinha é suportado. É possível usar uma combinação de texto e macros suportadas. `{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}` e macros de usuário são suportadas. As células do favo de mel são ordenadas alfabeticamente por nome do host e, dentro de cada host, por nome do item. Este parâmetro está disponível se *Tipo* estiver definido como *Texto*.
Casas decimais	Insira 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: **Automático** - usar tamanho de rótulo ajustado automaticamente; **Personalizado** - inserir um tamanho de rótulo personalizado (em porcentagem, relativo ao tamanho da célula do favo de mel). Observe que os rótulos que não couberem no tamanho da célula do favo de mel serão truncados.
Negrito	Marque a caixa de seleção para exibir os rótulos em negrito.
Cor	Selecione a cor do rótulo no seletor de cores. "D" representa 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 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 *Valor*.

Posição	Selecione a posição das unidades do item (antes ou depois do valor do item). Este parâmetro é ignorado para as seguintes unidades relacionadas a tempo: unixtime, uptime, s. Este parâmetro está disponível se *Tipo* estiver definido como *Valor* e a opção *Unidades* estiver selecionada.
Cor de fundo	
Cor de fundo	Selecione a cor de fundo das células do favo de mel no seletor de cores. ”D” representa 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 cores	Marque a caixa de seleção para ativar a transição suave entre as cores dos limites para as células do favo de mel. Este parâmetro está disponível se dois ou mais limites estiverem 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 ser salva. 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.

Disponibilidade do host

Visão geral

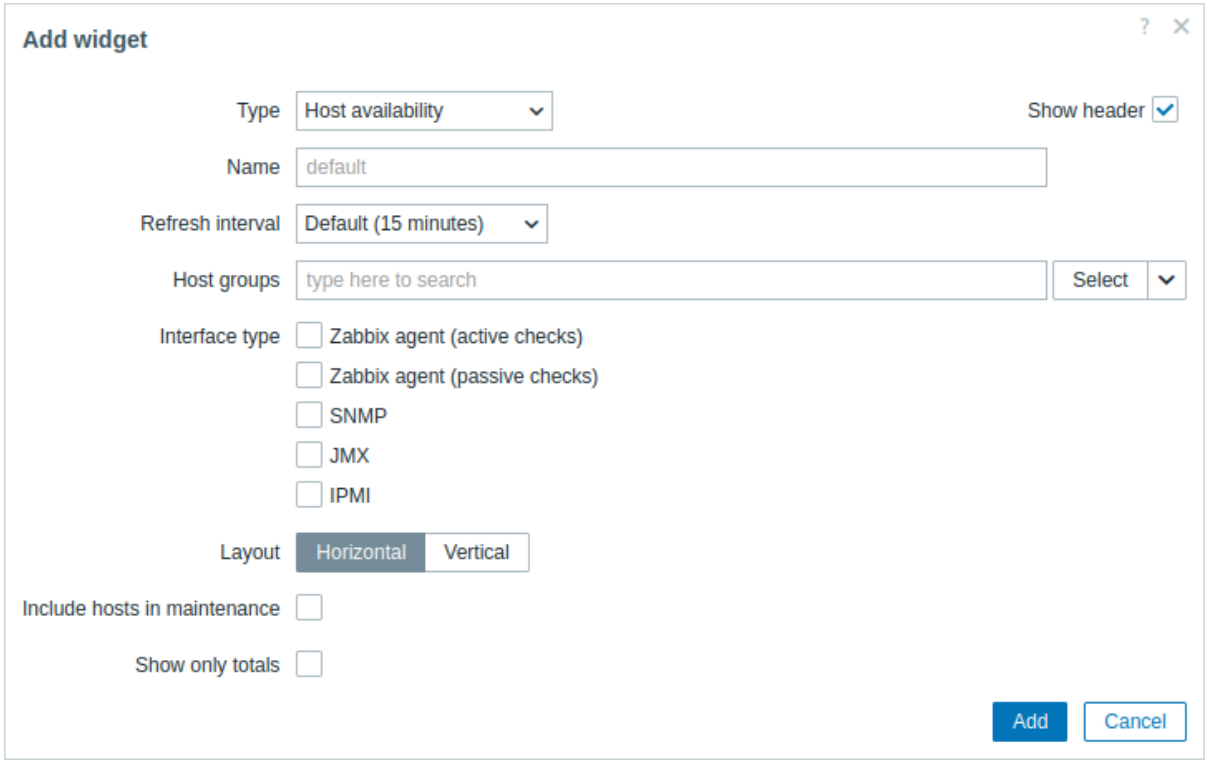
O widget *Disponibilidade do host* exibe a disponibilidade do host em grupos de hosts selecionados, permitindo monitorar o número de hosts que estão ativos ou inativos.

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:



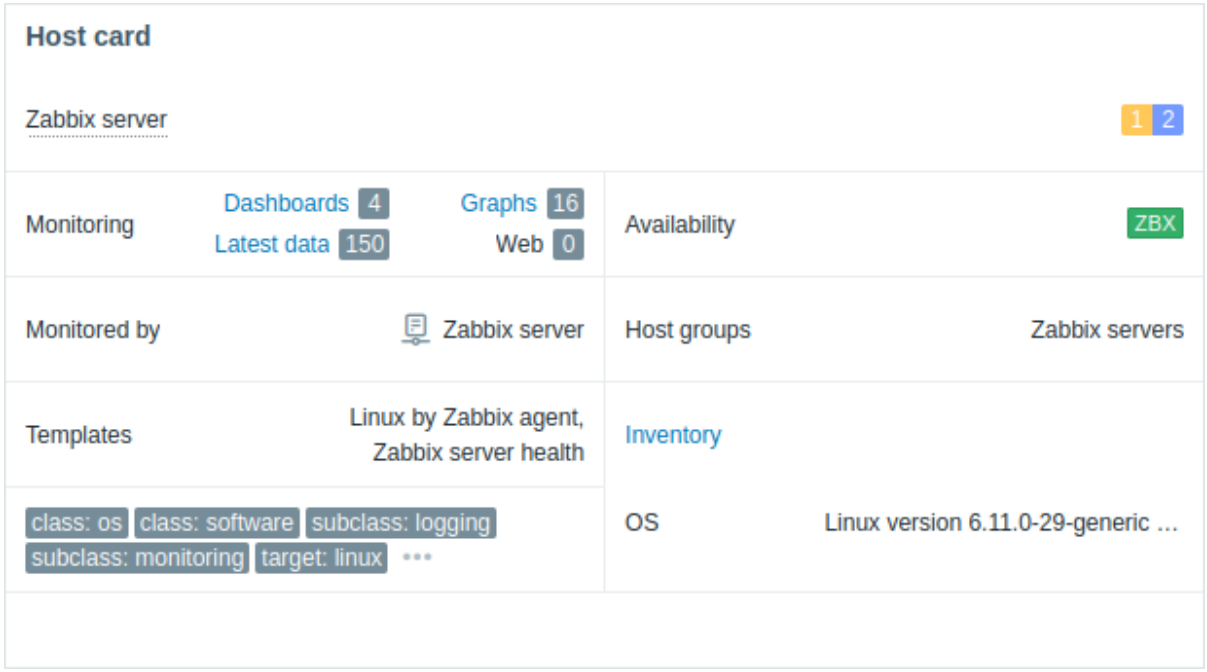
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

Cartão do 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.).

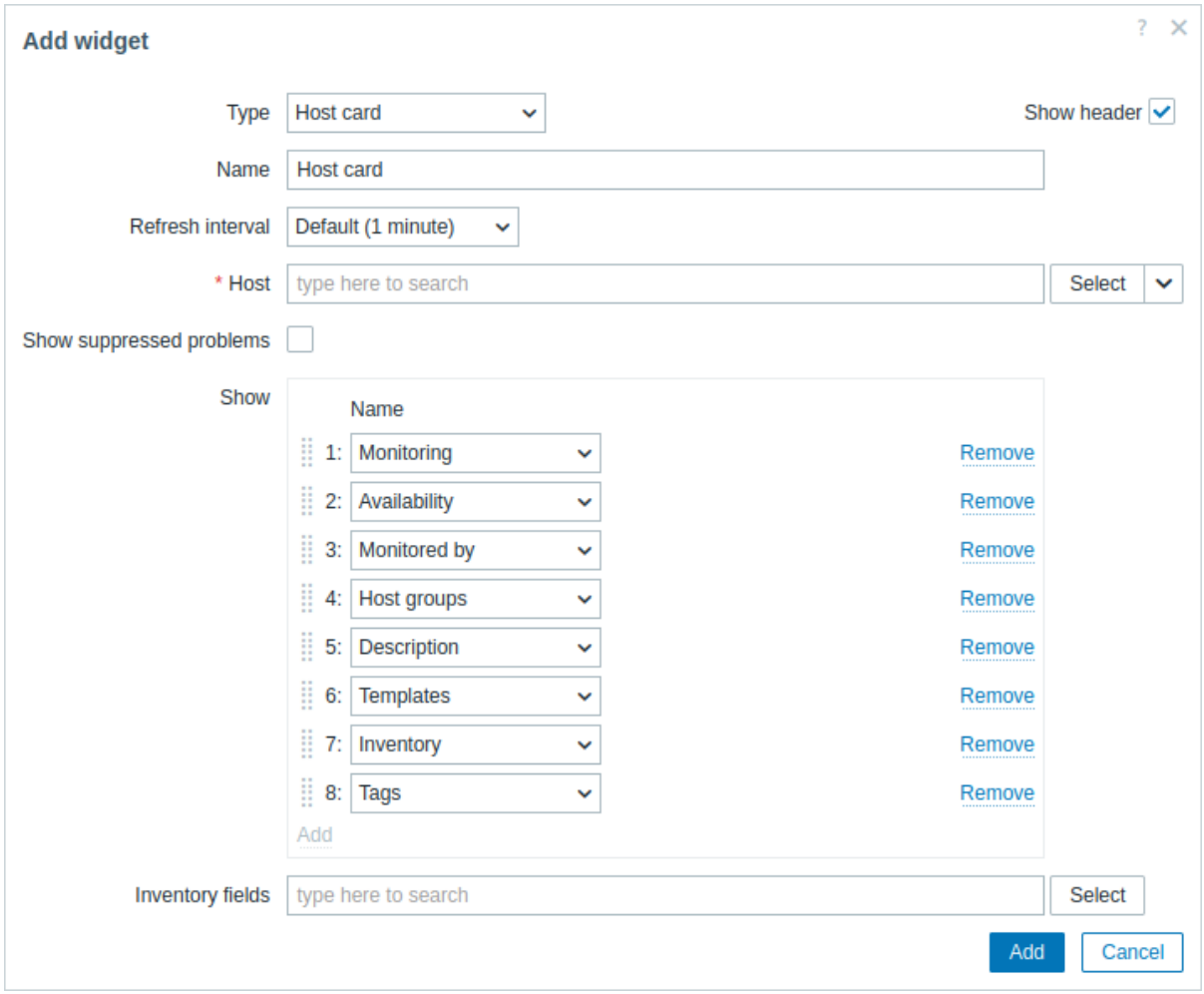


O host pode ser definido diretamente na configuração do widget ou selecionado a partir de um widget compatível ou do seletor de host no dashboard.

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*.

Navegador de host

Visão geral

O widget *Navegador de hosts* exibe uma lista de hosts com base em várias opções de filtragem e agrupamento.

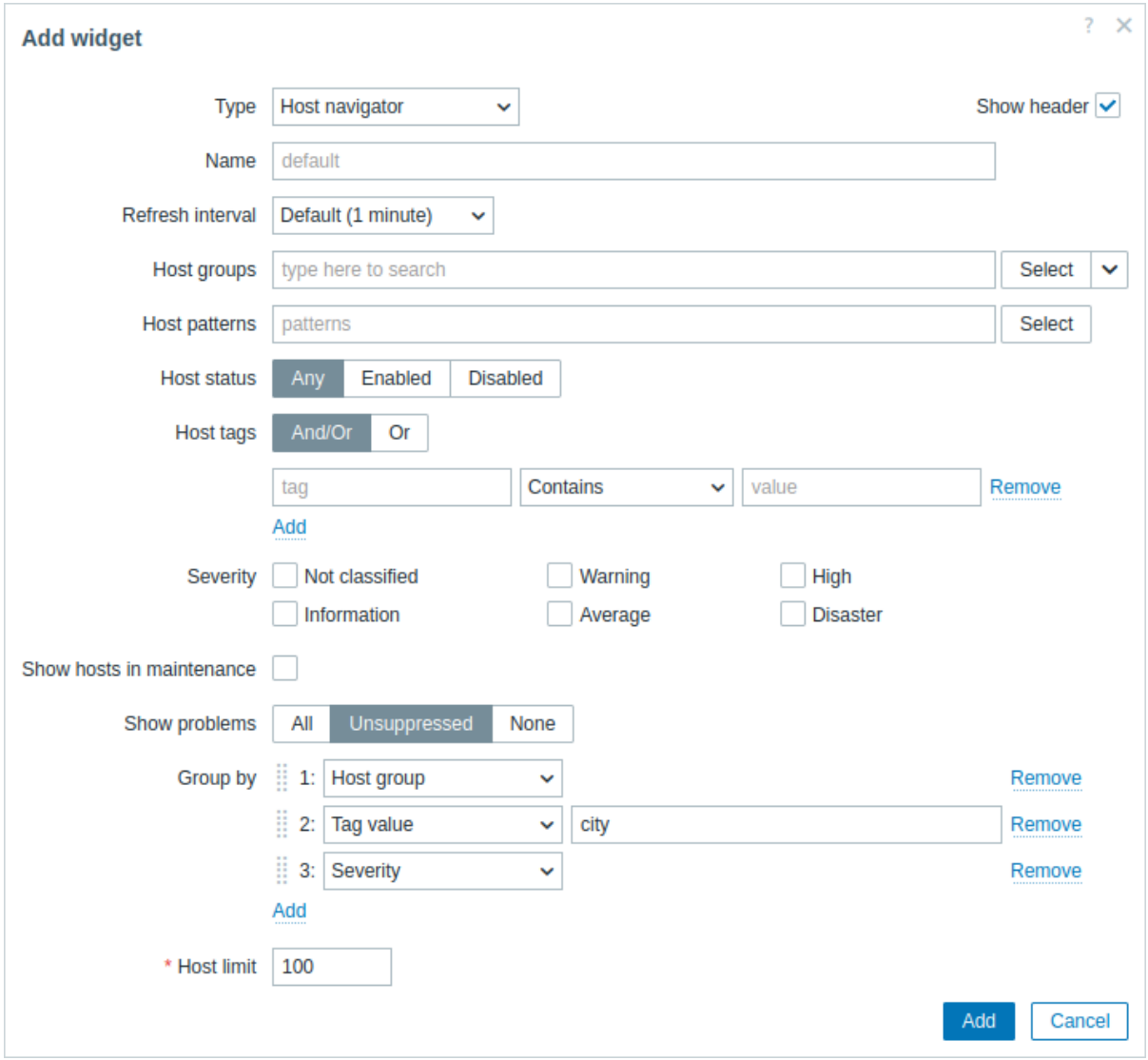
Os grupos pelos quais os hosts são organizados podem ser expandidos ou recolhidos.

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 comuns a todos os widgets, você pode definir as seguintes opções específicas:

Host groups	Selecione os grupos de hosts. Como alternativa, selecione um widget compatível como a fonte de dados para os grupos de hosts. Este campo oferece preenchimento automático; 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 hosts de todos os grupos de hosts. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Host patterns	Insira padrões de hosts ou selecione hosts existentes como padrões de hosts. Os hosts que corresponderem aos padrões especificados serão exibidos no host navigator. Este campo oferece preenchimento automático; 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. Padrões curinga podem ser usados para a seleção (por exemplo, ∗ retornará hosts que correspondam a zero ou mais caracteres; `Zabbix∗` retornará hosts que comecem com "Zabbix"). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. Ao começar a digitar, uma lista suspensa exibirá os hosts correspondentes, limitada aos hosts pertencentes aos hosts dentro dos *Host groups* selecionados, se houver. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Host status	Filtre os hosts a serem exibidos com base no status (qualquer, habilitado, desabilitado). Este parâmetro não está disponível ao configurar o widget em um template dashboard.

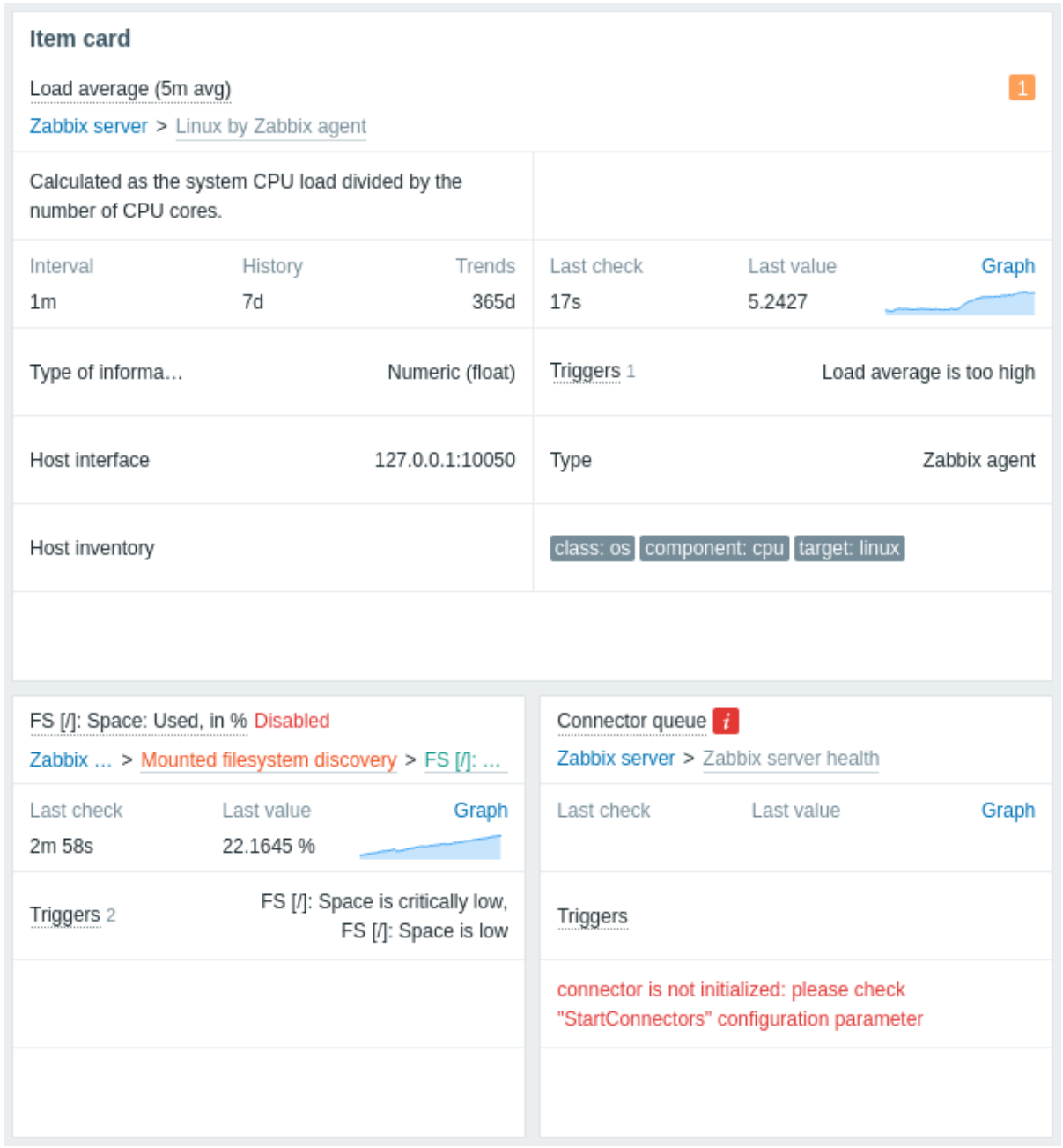
Host tags	Especifique tags para filtrar os hosts exibidos no widget. É possível incluir e excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tags especificados; **Equals** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contains** - inclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - exclui os nomes de tags especificados; **Does not equal** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Does not contain** - exclui os nomes de tags especificados quando os valores das tags contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - basta que uma condição seja atendida. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Severity	Marque as severidades de problemas 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.
Show hosts in maintenance	Marque esta caixa de seleção para exibir os hosts em manutenção (nesse caso, o ícone de manutenção será mostrado ao lado do nome do host). Este parâmetro é denominado *Show data in maintenance* ao configurar o widget em um template dashboard.
Show problems	Filtre os problemas a serem exibidos com base no status (todos, não suprimidos, nenhum). Os problemas são indicados por blocos coloridos no final da linha do host. As cores são baseadas na severidade do problema, que pode ser personalizada em *Administration* > *General* > *Trigger displaying options*. As severidades dos problemas podem ser alteradas ao atualizar problemas.

Group by	Adicione um atributo de agrupamento pelo qual agrupar os hosts selecionados: **Host group** - agrupa os hosts pelo grupo de hosts; **Tag value** - insira um nome de tag para agrupar os hosts pelos valores dessa tag (por exemplo, insira ”city” para agrupar os hosts pelos valores ”Riga”, ”Tokyo” etc.); **Severity** - agrupa os hosts pelas severidades dos problemas. Se *Show problems* estiver configurado para exibir problemas, eles serão exibidos da seguinte forma: - para cada grupo de severidade, somente a contagem de problemas correspondente será exibida; - para cada host, todas as contagens de seus problemas serão exibidas. Observe que os hosts serão agrupados somente pelas severidades marcadas no parâmetro *Severity*; se nenhuma severidade for marcada, todos os hosts serão agrupados por todas as severidades. Os atributos de agrupamento podem ser reordenados arrastando-os 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: color, 2: city) fará com que os hosts sejam agrupados por cor (red, blue etc.) e depois por cidade (Riga, Tokyo 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. Os hosts que não correspondem aos atributos de agrupamento configurados são exibidos no grupo *Uncategorized*. Podem ser especificados até 10 atributos de agrupamento, e todos devem ser exclusivos. Se nenhum atributo de agrupamento for especificado, os hosts não serão agrupados.
Host limit	Insira 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 mostrada abaixo dos hosts exibidos (por exemplo, ”100 of 100+ hosts are shown”). Observe que o limite de hosts configurado também afeta a exibição dos grupos configurados; por exemplo, se o limite de hosts estiver definido como 100 e os hosts forem agrupados por valores de tags (mais de 200), somente os primeiros 100 valores de tags, com os hosts correspondentes, serão exibidos no widget. Este parâmetro não é afetado pelo parâmetro *Limit for search and filter results* em *Administration > General > GUI*. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

Cartão do 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.).

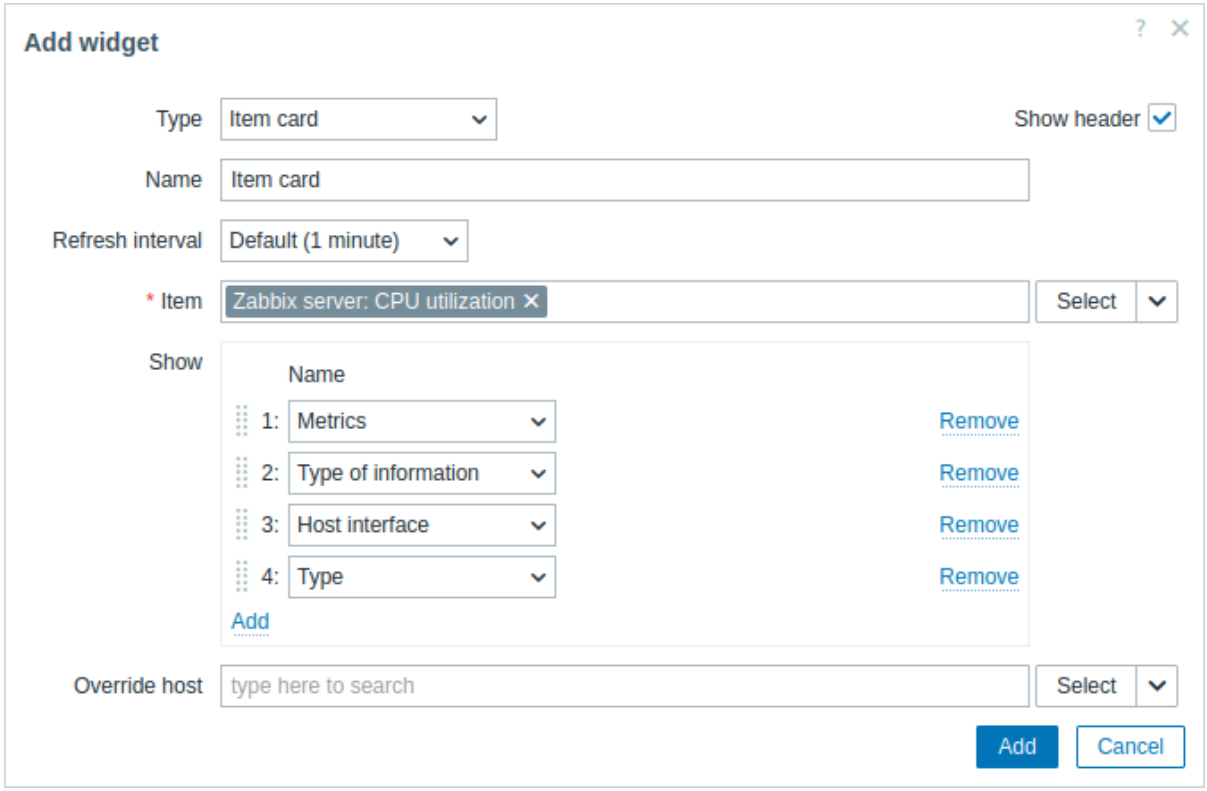


O item pode ser definido diretamente na configuração do widget ou selecionado a partir de um widget compatível no dashboard.

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:



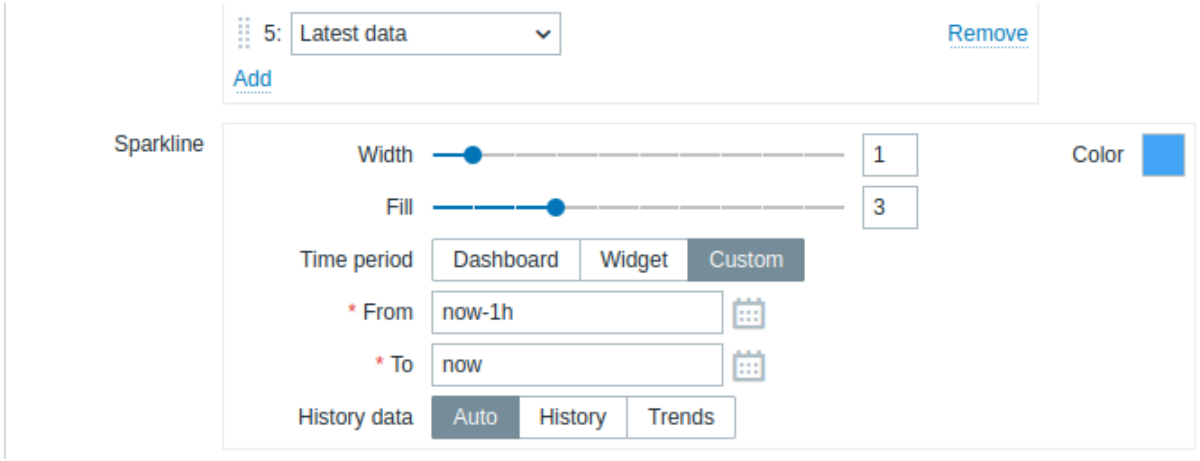
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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”.

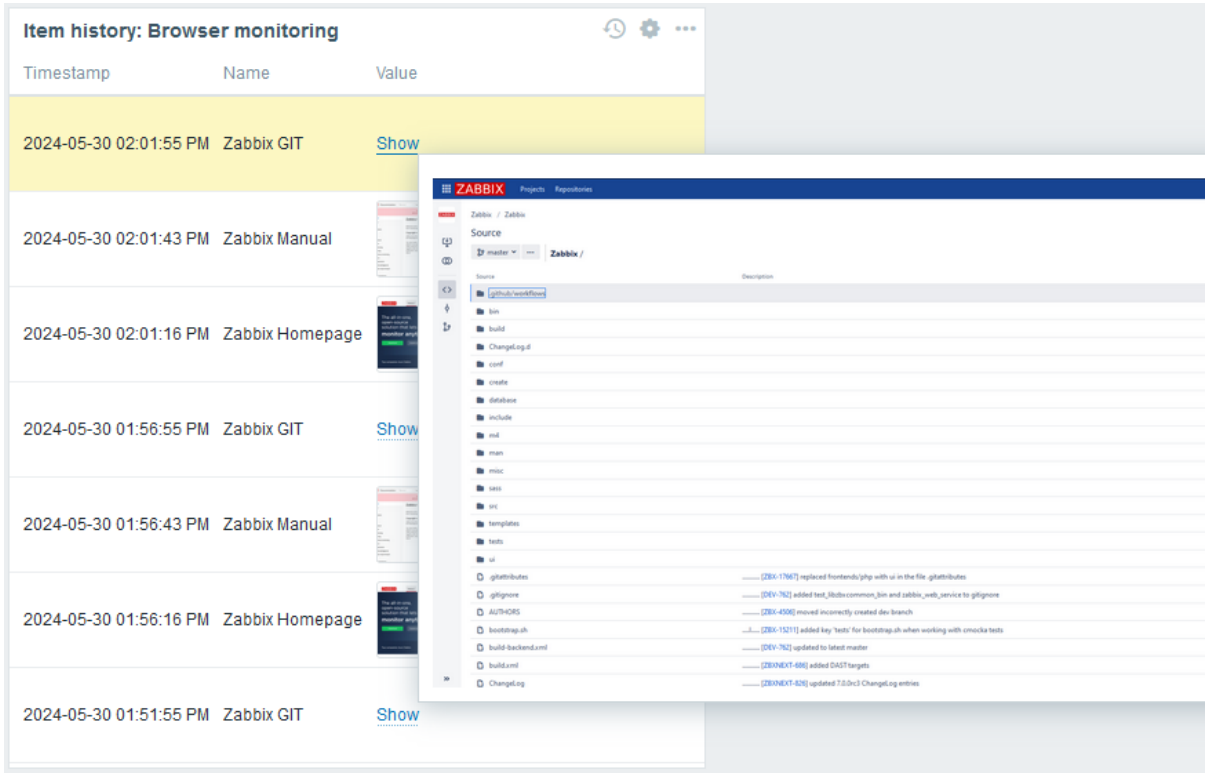
De	Digite ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. A sintaxe de tempo relativo (`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.

Histórico de item

Visão geral

O widget *Item history* exibe os dados mais recentes de todos os tipos de item (numérico, texto etc.) em formato de tabela. Ele pode mostrar barras de progresso e imagens (útil para browser items) e pode destacar valores (útil para monitoramento de arquivos de log).

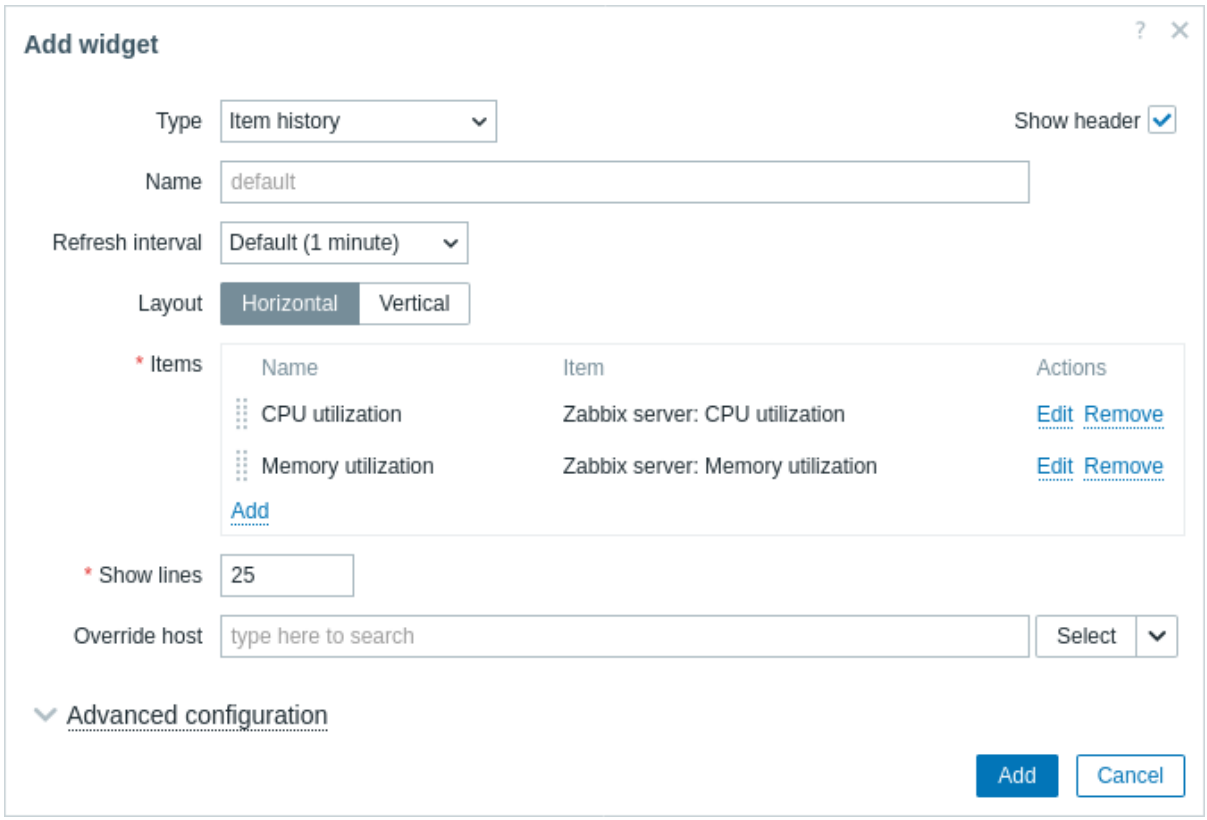




Até 1000 registros podem ser exibidos.

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 character, text, log e JSON:

Destaques	Clique em *Add* 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 em que a expressão regular especificada corresponder ao texto.

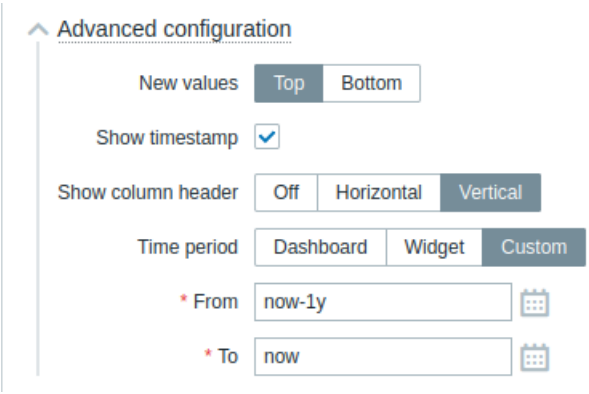
Exibição	Selecione como o valor do item deve ser exibido: **As is** - exatamente como recebido. As quebras de linha são preservadas se o valor contiver retornos de linha. A quebra de palavras é aplicada se *Layout* estiver definido como ”Vertical” e mais de uma coluna estiver configurada; **HTML** - como texto formatado em HTML; **Single line** - como uma única linha, truncada para um comprimento especificado (1-500 caracteres). Passar o mouse ou clicar no valor truncado abre uma janela 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 *Show timestamp* em advanced configuration também deve estar 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".

Navegador de item

Visão geral

O widget *Navegador de items* exibe uma lista de items com base em várias opções de filtragem e agrupamento.

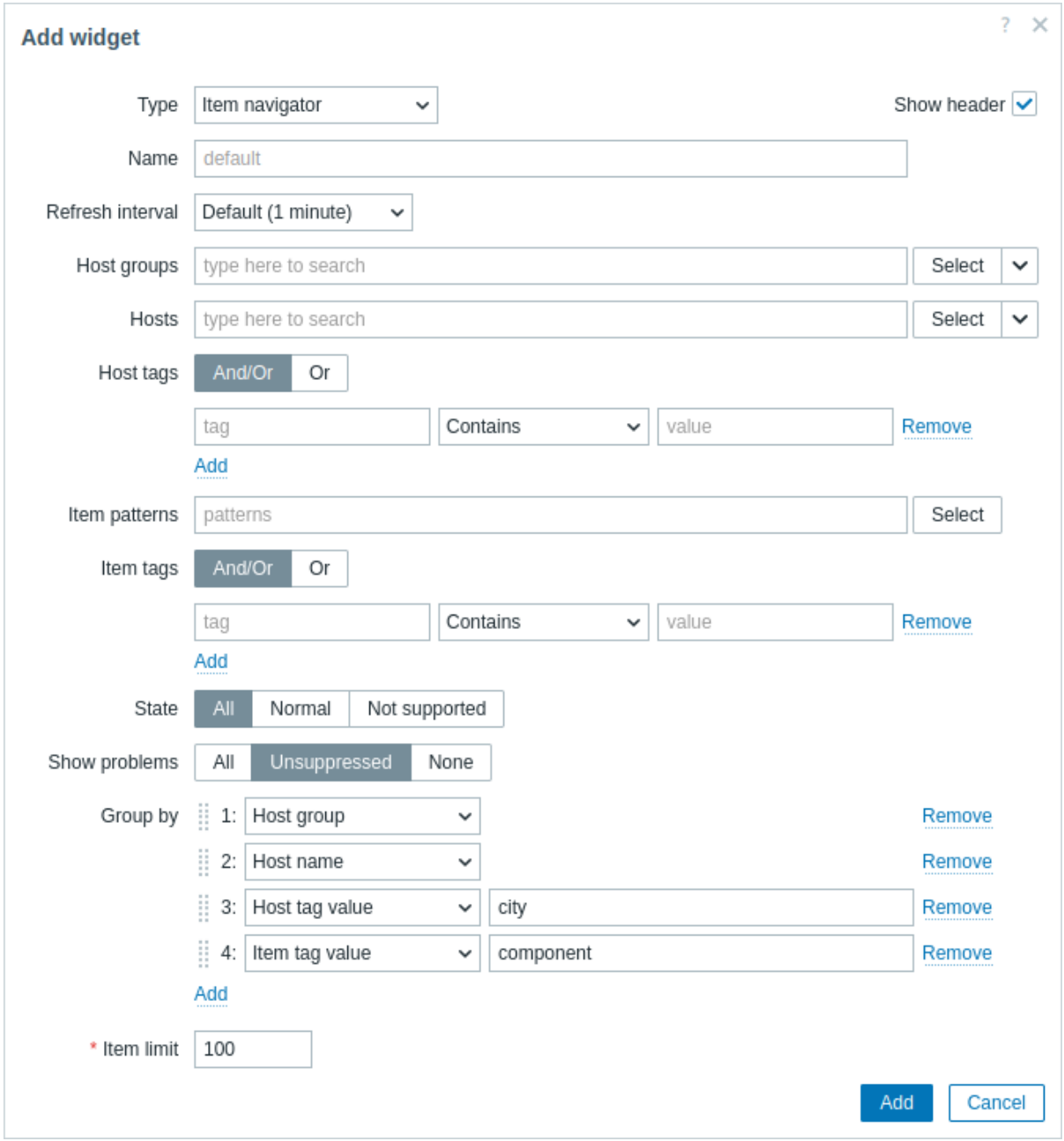
Os grupos pelos quais os items são organizados podem ser expandidos ou recolhidos.

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 itens* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione os grupos de hosts. Como alternativa, selecione um widget compatível como a fonte de dados para os grupos de hosts. Este campo oferece preenchimento automático; 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 os itens 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 os hosts. Como alternativa, selecione um widget compatível ou o dashboard como a fonte de dados para os hosts. Este campo oferece preenchimento automático; 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 os itens pertencentes a todos os hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

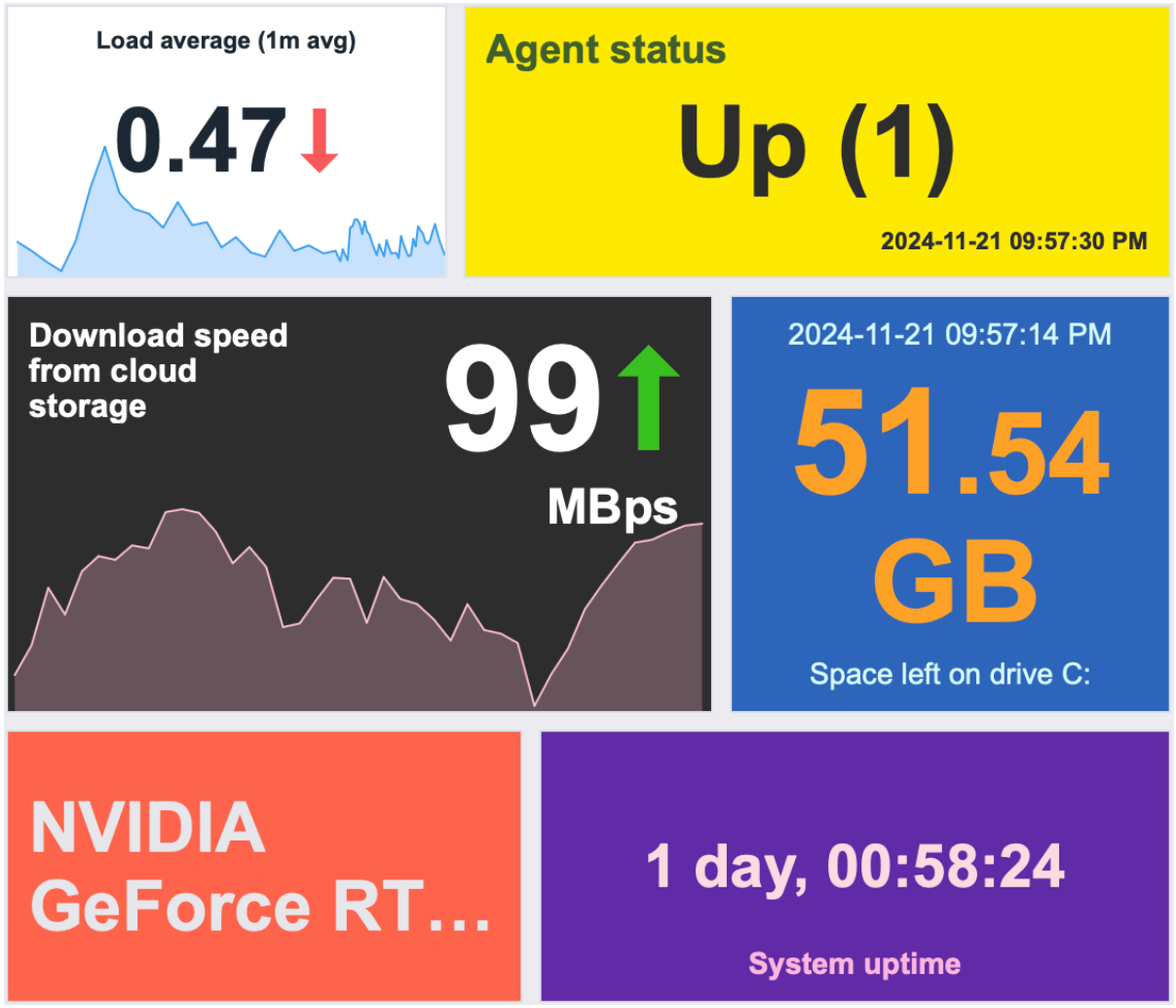
Tags de host	Especifique as tags de host para filtrar os itens exibidos no widget. É possível incluir e excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência de nomes de tags sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **É igual a** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual a** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições que têm 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 itens existentes como padrões de item. Os itens que corresponderem aos padrões especificados serão exibidos no navegador de itens. Padrões curinga podem ser usados para a seleção (por exemplo, `*` retornará itens que correspondem a zero ou mais caracteres; `Zabbix*` retornará itens que começam com "Zabbix"). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. Ao começar a digitar, uma lista suspensa exibirá os itens correspondentes, limitada àqueles pertencentes aos *Hosts* selecionados ou aos hosts dentro dos *Grupos de hosts* selecionados, se houver. O símbolo curinga é sempre interpretado; portanto, não é possível adicionar, por exemplo, individualmente um item chamado *item** se houver outros itens correspondentes (por exemplo, item2, item3). Ao configurar o widget em um dashboard de template, este parâmetro permite selecionar somente itens configurados no template.
Tags de item	Especifique as tags de item para filtrar os itens exibidos no widget. É possível incluir e excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência de nomes de tags sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **É igual a** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual a** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados cujos valores contêm a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições que têm o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.
Estado	Filtre os itens a serem exibidos com base no estado (todos, normal, não suportado).
Mostrar problemas	Filtre os problemas a serem exibidos com base no status (todos, não suprimidos, nenhum). Os problemas são indicados por 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 trigger*. As severidades dos problemas podem ser alteradas ao atualizar problemas.

Agrupar por	Adicione um atributo de agrupamento pelo qual os itens serão agrupados: **Grupo de hosts** - agrupa os itens pelos grupos de hosts de seus hosts; **Nome do host** - agrupa os itens pelos hosts aos quais pertencem; **Valor da tag de host** - insira um nome de tag para agrupar os itens pelos valores dessa tag de host (por exemplo, insira ”city” para agrupar os itens pelos valores ”Riga”, ”Tokyo” etc.); **Valor da tag de item** - insira um nome de tag para agrupar os itens pelos valores dessa tag de item (por exemplo, insira ”component” para agrupar os itens pelos valores ”cpu”, ”memory” etc.). Os atributos de agrupamento podem ser reordenados arrastando-os 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: color, 2: city) fará com que os itens sejam agrupados por cor (red, blue etc.) e, em seguida, 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 itens os seleciona e destaca em todos os grupos. Os itens que não correspondem aos atributos de agrupamento configurados são exibidos no grupo *Sem categoria*. Se *Mostrar problemas* estiver configurado para exibir problemas, eles serão exibidos da seguinte forma: - para cada grupo, é exibida a contagem de problemas de todos os itens subordinados; - para cada item, somente sua contagem de problemas é exibida. Podem ser especificados até 10 atributos de agrupamento, e todos devem ser exclusivos. Se nenhum atributo de agrupamento for especificado, os itens não serão agrupados.
Limite de itens	Insira o número máximo de itens a serem exibidos. Os valores possíveis variam de 1 a 9999. Quando houver mais itens disponíveis para exibição do que o limite definido, uma mensagem correspondente será exibida abaixo dos itens exibidos (por exemplo, ”100 de mais de 100 itens são exibidos”). Observe que o limite de itens configurado também afeta a exibição dos grupos configurados; por exemplo, se o limite de itens estiver definido como 100 e os itens forem agrupados por seus hosts (cada um contendo 200 itens), somente o primeiro host com seus 100 itens será exibido no widget. Este parâmetro não é afetado pelo parâmetro *Limite para resultados de pesquisa e filtro* em *Administração > Geral > GUI*.

Valor do item

Visão geral

O widget *Item value* exibe o valor de um único item numérico ou de string. Ele é útil para acompanhar métricas importantes, visualizar limites e detectar mudanças repentinas nos dados.



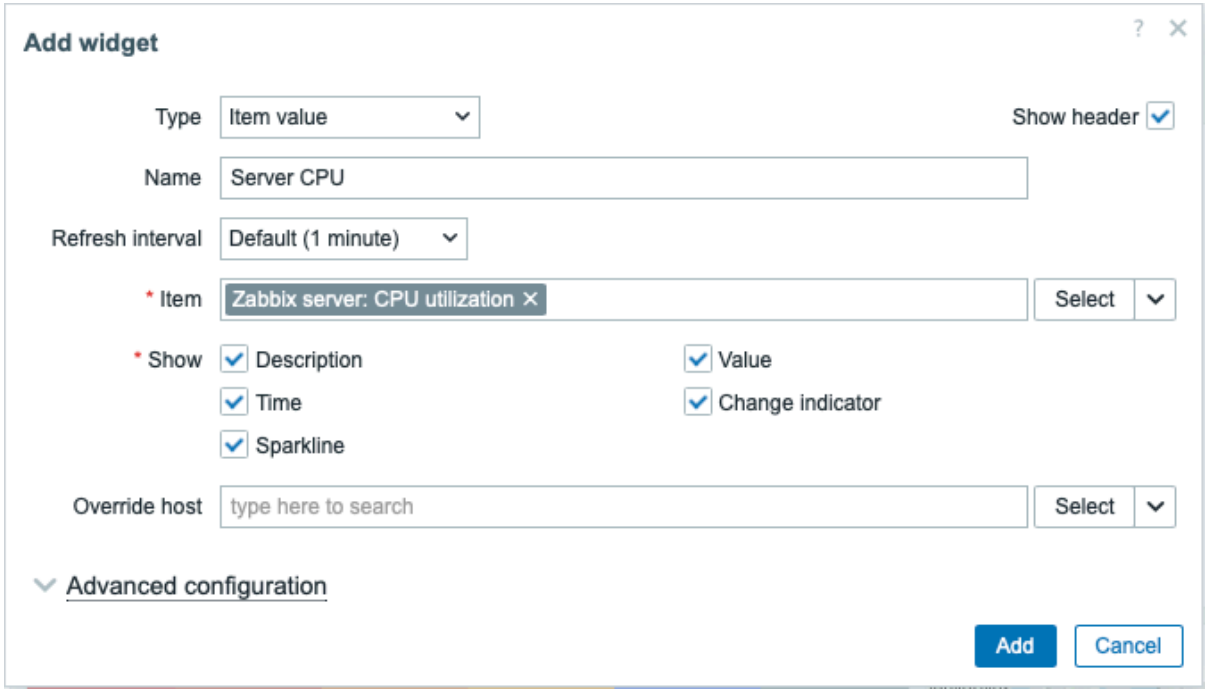
Você pode configurar o widget para exibir o seguinte:

- Carimbo de 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 sparkline para valores do período de tempo especificado

Ao clicar no widget, é aberto um gráfico ad hoc para itens numéricos ou os dados mais recentes para itens de string.

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 o item. Items 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 cabeçalho *Configuração avançada* para exibir as opções de configuração avançada.

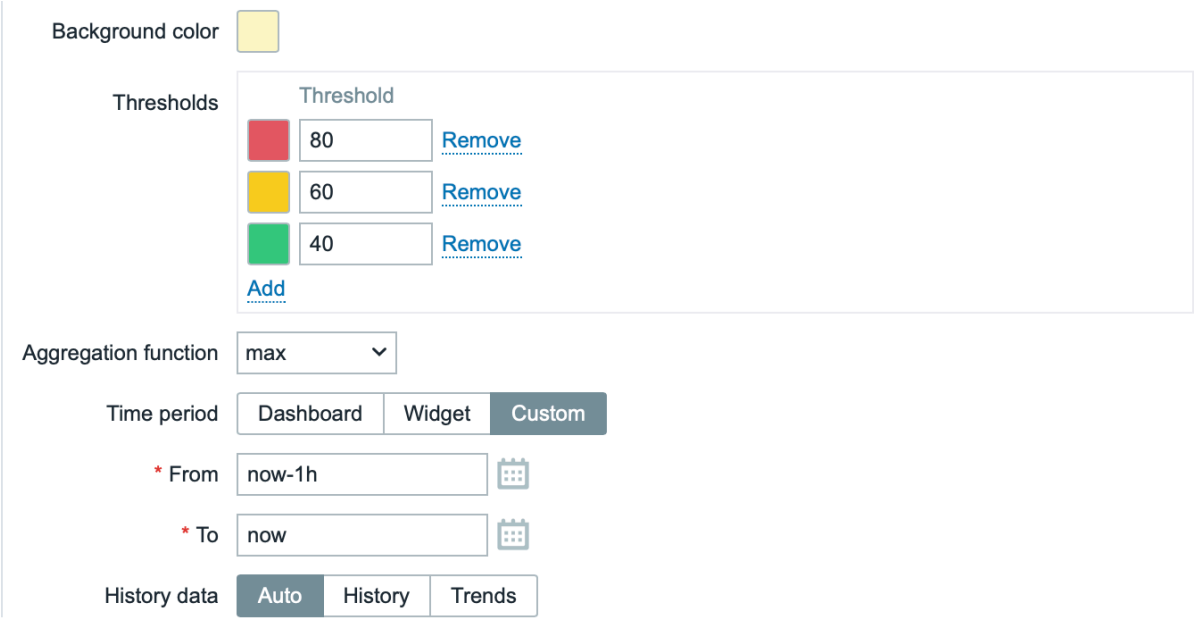
Configuração avançada

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) de todo o widget e uma função de agregação para exibir valores.

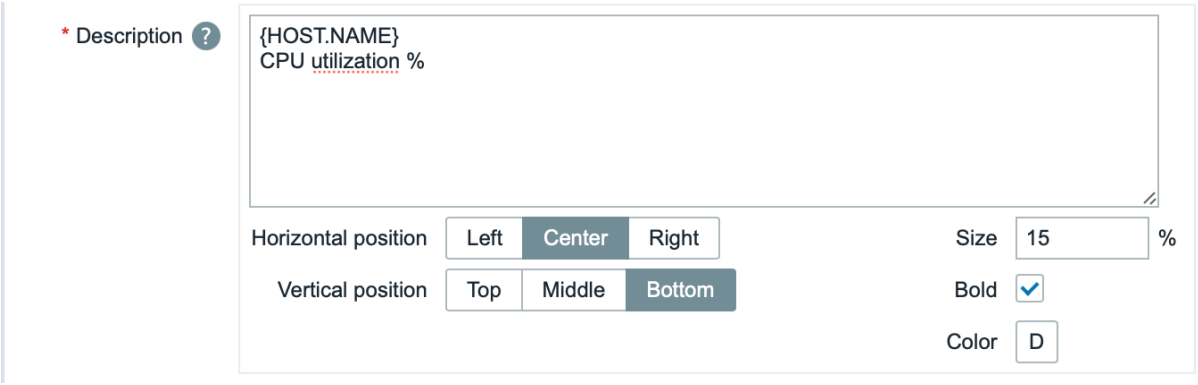


Cor de fundo	Selecione a cor de fundo de todo o widget no seletor de cores. D representa a cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Limiares	Configure a cor de fundo dinâmica de todo o widget. Clique em *Adicionar* para adicionar um limiar, selecione a cor de fundo no seletor de cores e especifique um valor numérico. Quando o valor do item for igual ou maior que o valor do limiar, a cor de fundo será alterada. A lista será classificada em ordem crescente ao ser salva. Observe que a cor de fundo dinâmica será exibida corretamente somente para itens numéricos.
Função de agregação	Especifique qual função de agregação usar no *Período de tempo* selecionado: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **count** - exibe a quantidade de valores; **sum** - exibe a soma dos valores; **first** - exibe o primeiro valor; **last** - exibe o último valor. Se **não usado** for selecionado, o valor mais recente será exibido (sem agregação). A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia), em vez do valor mais recente. Somente dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em 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*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido ao passar o mouse sobre ele. 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 fica oculto quando *Função de agregação* está definida como "não usado".
Widget	Insira ou selecione um widget compatível como fonte de dados para o período de tempo. Este parâmetro está disponível quando *Período de tempo* está definido como "Widget".
De	Insira ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".
Até	Insira ou selecione o fim do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".

Dados históricos	Obtenha dados do histórico ou das tendências: **Automático** - seleção automática; **Histórico** - obtém dados do histórico; **Tendências** - obtém dados de tendências. Essa configuração se aplica somente a dados numéricos. Os dados não numéricos sempre serão obtidos do histórico.

Descrição

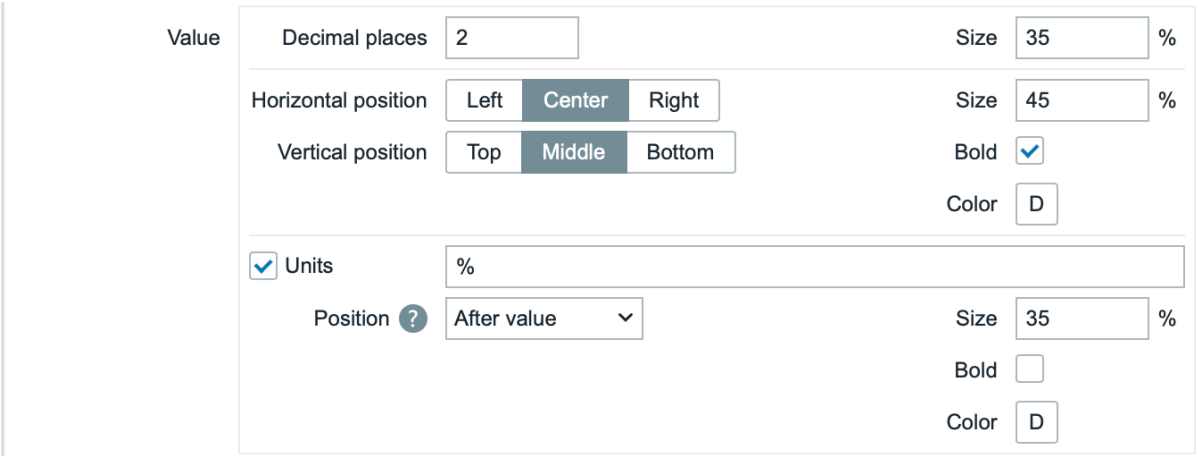
Esses parâmetros determinam como a descrição do item deve ser exibida.



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

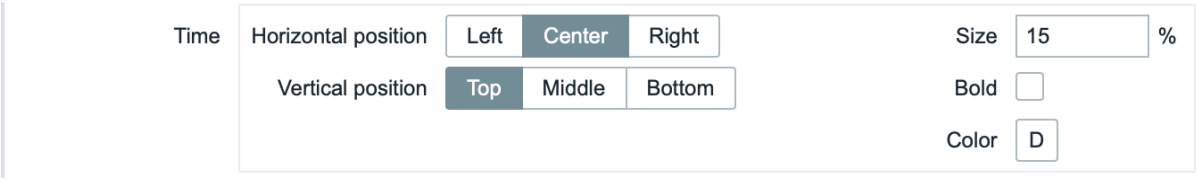
Esses parâmetros determinam como o valor do item deve ser exibido.



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

A seção permite selecionar a cor dos indicadores de alteração no seletor de cores.



Os indicadores de alteração são os seguintes:

- ↑ - o valor do item aumentou (para itens numéricos)
- ↓ - o valor do item diminuiu (para itens numéricos)
- ↕ - o valor do item foi alterado (para itens de string e itens com mapeamento de valores)

Observação:

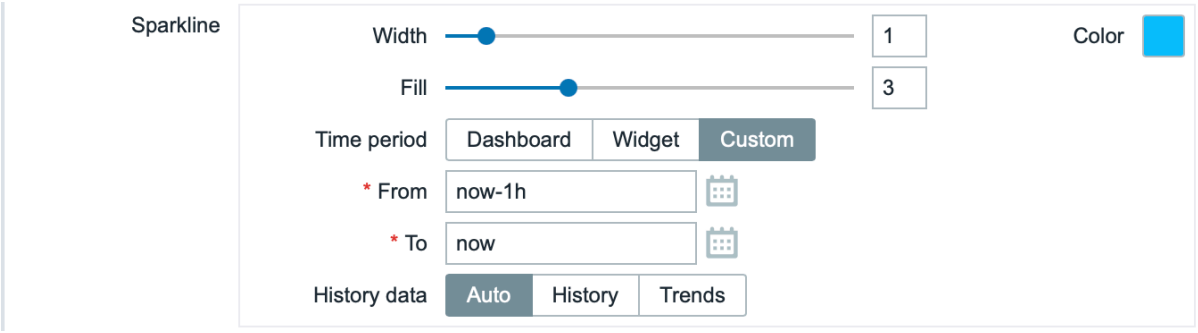
- Os indicadores de aumento (↑) e diminuição (↓) não são exibidos se houver apenas um valor.
- Os valores são sempre comparados dentro do mesmo escopo; por exemplo:
  - O valor mais recente é comparado ao valor anterior.
  - Um valor mensal é comparado ao mês anterior.
  - Para agregações, o período anterior tem a mesma duração que o período selecionado e termina imediatamente antes do início do período selecionado.

D representa a 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 (a parte inteira do valor para itens numéricos).

Sparkline

Esses parâmetros determinam como o gráfico sparkline deve ser exibido.

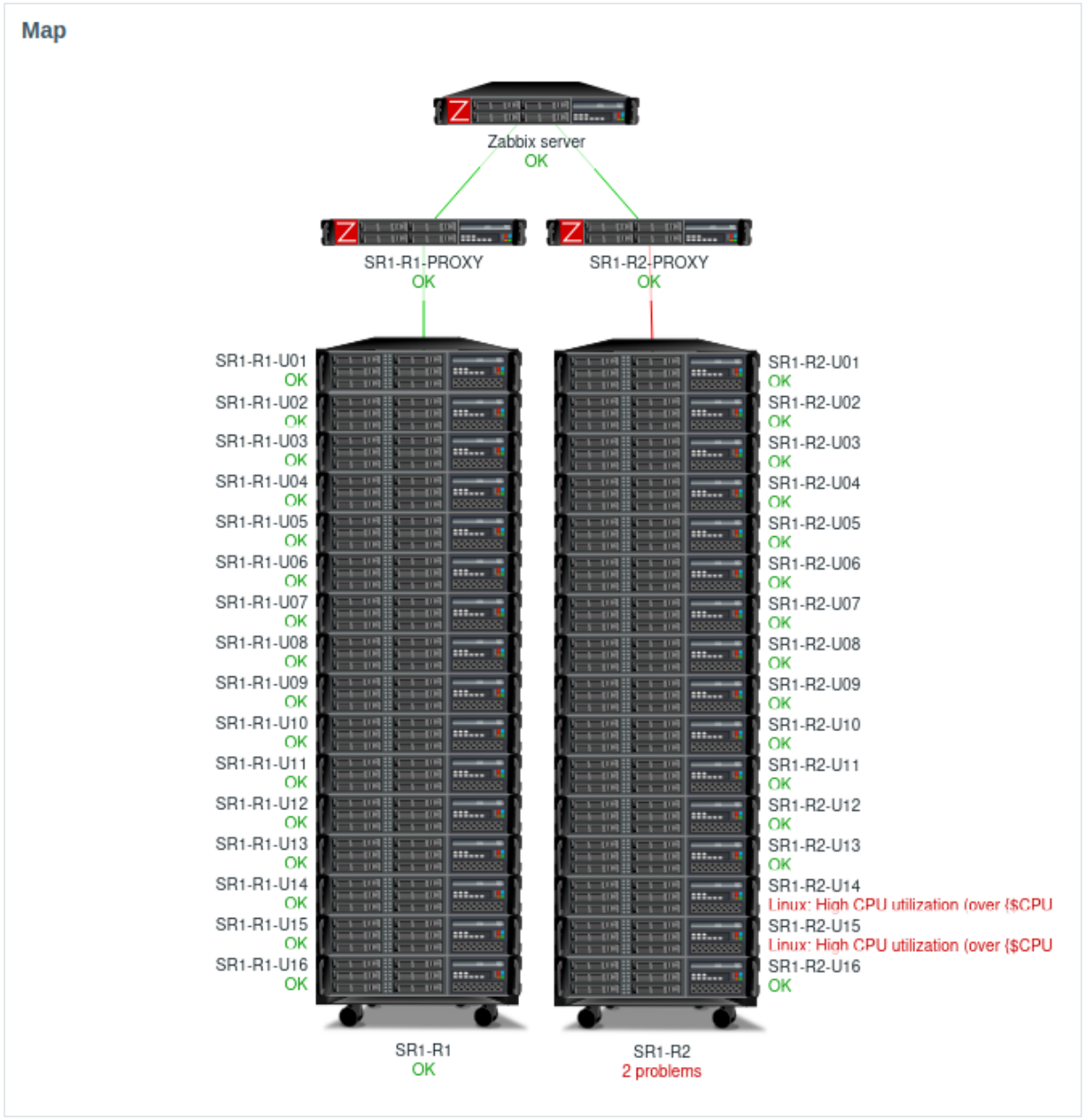


Largura	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Cor	Selecione a cor da linha e do preenchimento.
Preenchimento	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente 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.

Mapa

Visão geral

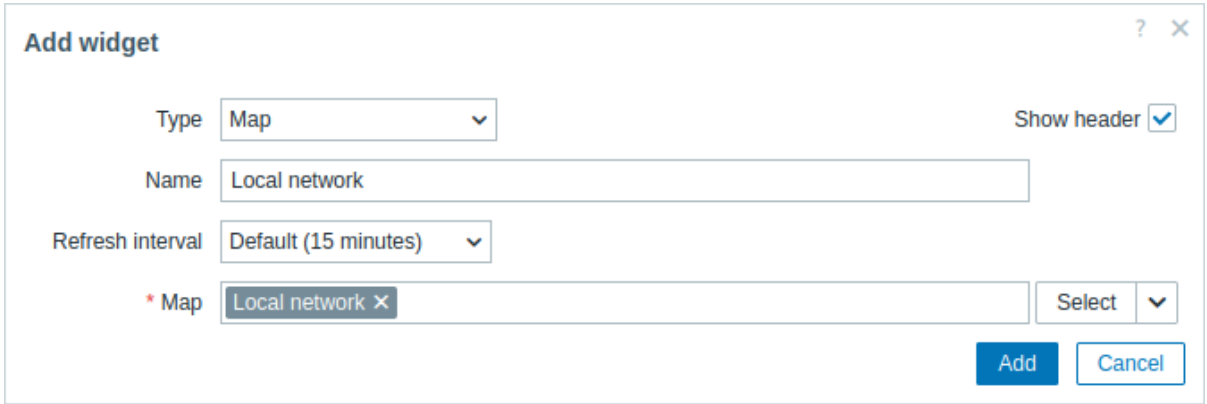
O widget *Mapa* pode exibir um mapa de rede ou uma visualização semelhante, fornecendo uma visão geral dinâmica da sua rede.



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:



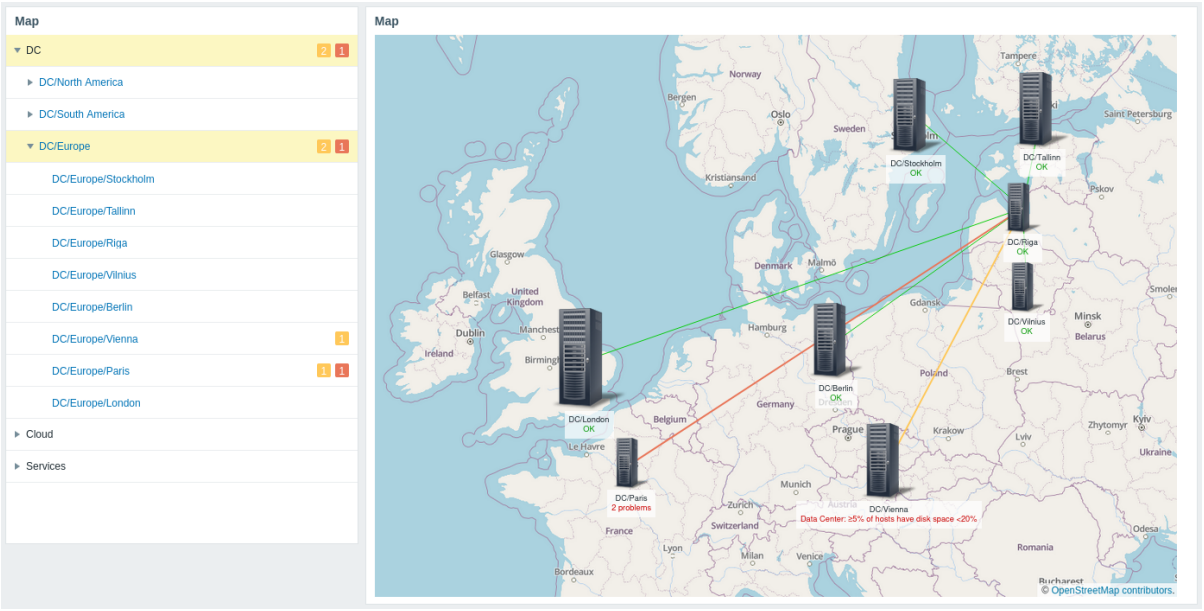
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

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.

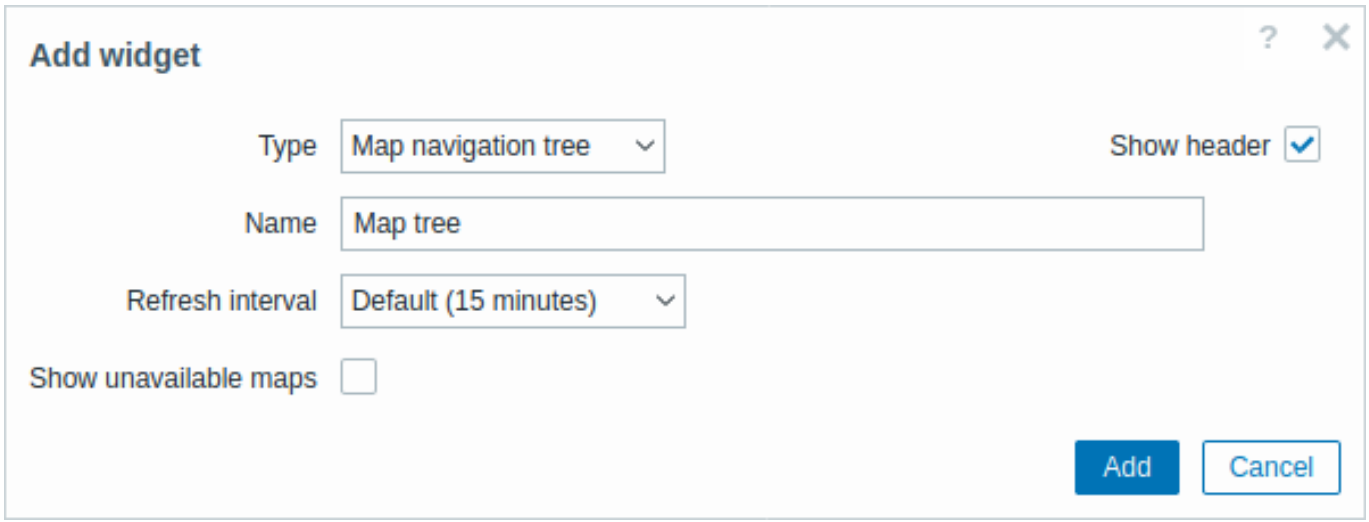


Você pode combiná-lo com o widget Mapa para exibir dinamicamente o mapa selecionado na árvore.

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:

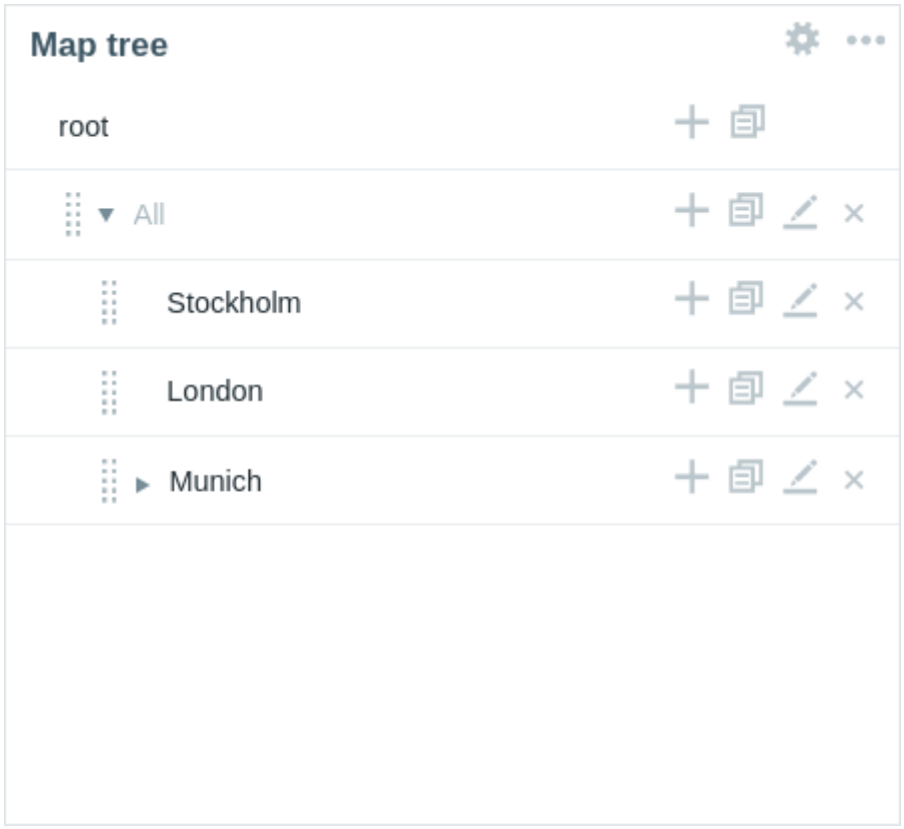


Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

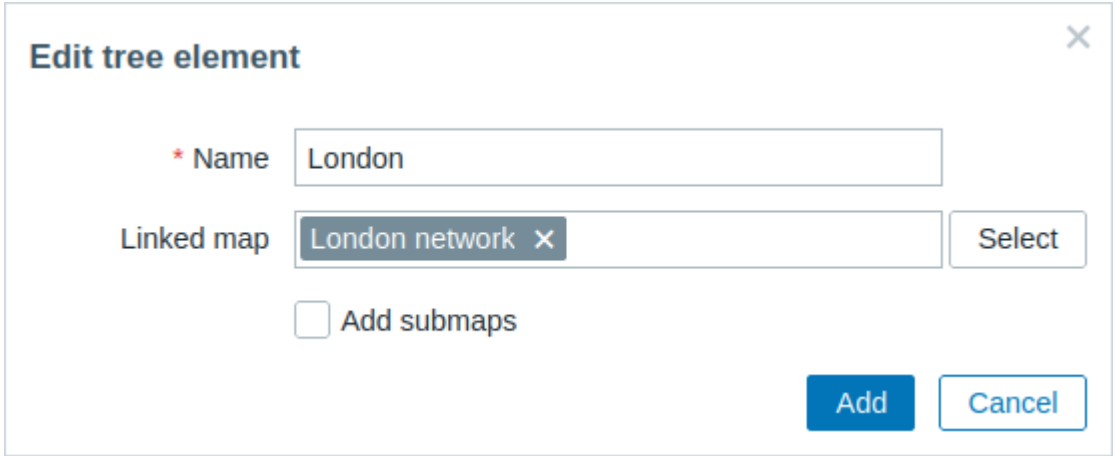
Os elementos da árvore de navegação são exibidos em uma lista. Você pode:

- 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.



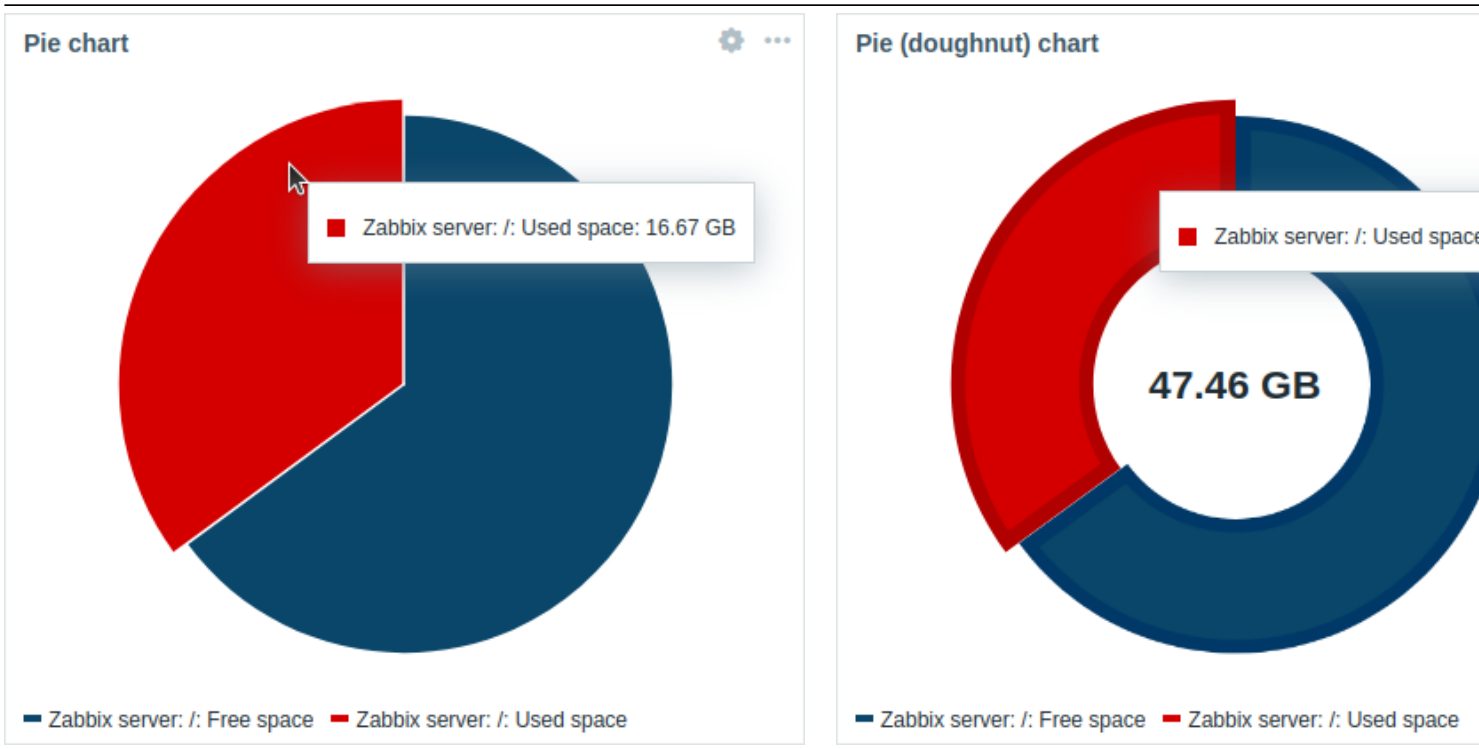
Os seguintes parâmetros de configuração do elemento da árvore de navegação estão disponíveis:

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.

Gráfico de pizza

Visão geral

O widget *Gráfico de pizza* exibe dados numéricos de item como um gráfico de pizza ou de rosca baseado em vetores. Esse widget pode ajudar você a visualizar como os itens ou hosts contribuem para o conjunto de dados geral.



Ao passar o mouse, o setor sobre o qual o cursor está posicionado se expande e exibe sua legenda; ao clicar no setor, ele permanece expandido e sua legenda continua visível até ser fechada.

As informações exibidas no widget *Gráfico de pizza* podem ser baixadas como uma imagem PNG selecionando a opção *Baixar imagem* no menu *Ações* do widget ..., que aparece quando você passa o mouse sobre o widget. Dependendo das configurações do navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Salvar* é aberta, permitindo que você escolha o local.

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. É possível adicionar dois tipos de conjuntos de dados:

- *Padrões de item* - os dados dos itens correspondentes são exibidos. Você pode escolher uma única cor base ou selecionar uma linha da paleta para atribuir cores distintas a cada item correspondente.
- *Lista de itens* - os dados dos itens selecionados são exibidos. Você pode escolher individualmente a cor de cada item no seletor.

Por padrão, um conjunto de dados de *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 corresponderem aos padrões inseridos serão exibidos no gráfico de pizza; até 50 itens podem ser exibidos. Padrões curinga podem ser usados para a 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*). A especificação de padrões de host e item é obrigatória para conjuntos de dados de ”Padrões de item”. Consulte 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 de 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 pode existir apenas um item ”Total” 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* (consulte abaixo) será desativado e definido como ”não utilizado”. Você também pode selecionar widgets compatíveis como a fonte de dados para os itens clicando no botão *Adicionar widget*. As mesmas opções aplicáveis a itens individuais também se aplicam aos widgets. A especificação de itens ou widgets é obrigatória para conjuntos de dados de ”Lista de itens”. Consulte também: Detalhes da configuração do conjunto de dados. Observe que somente 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 somente os itens configurados no template.
Função de agregação	Especifique qual função de agregação usar para cada item no conjunto de dados dentro do *Período* selecionado: **min** - exibir o menor valor; **max** - exibir o maior valor; **avg** - exibir o valor médio; **sum** - exibir a soma dos valores; **count** - exibir a contagem de valores; **first** - exibir o primeiro valor; **last** - exibir o último valor (padrão). A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia). Somente dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em numéricos.
Agregação do conjunto de dados	Especifique qual função de agregação usar para todo o conjunto de dados: **não utilizado** - sem agregação; os itens são exibidos separadamente (padrão); **min** - exibir o menor valor; **max** - exibir o maior valor; **avg** - exibir o valor médio; **sum** - exibir a soma dos valores; **count** - exibir a contagem de valores. A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia) selecionado na aba *Período* 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, inclusive 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 dos conjuntos de dados é recalculada após a reordenação/arraste dos conjuntos de dados. Rótulos de conjuntos de dados muito longos serão encurtados para caber no local em que forem exibidos (por exemplo, ”Número de proc...”).
Tags de item	Especifique tags para filtrar os itens exibidos no widget. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - incluir os nomes de tag especificados; **É igual a** - incluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - incluir os nomes de tag especificados nos quais os valores de tag contenham a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas); **Não existe** - excluir os nomes de tag especificados; **Não é igual a** - excluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Não contém** - excluir os nomes de tag especificados nos quais os valores de tag contenham a string inserida (correspondência de substring, sem diferenciação entre maiúsculas e minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

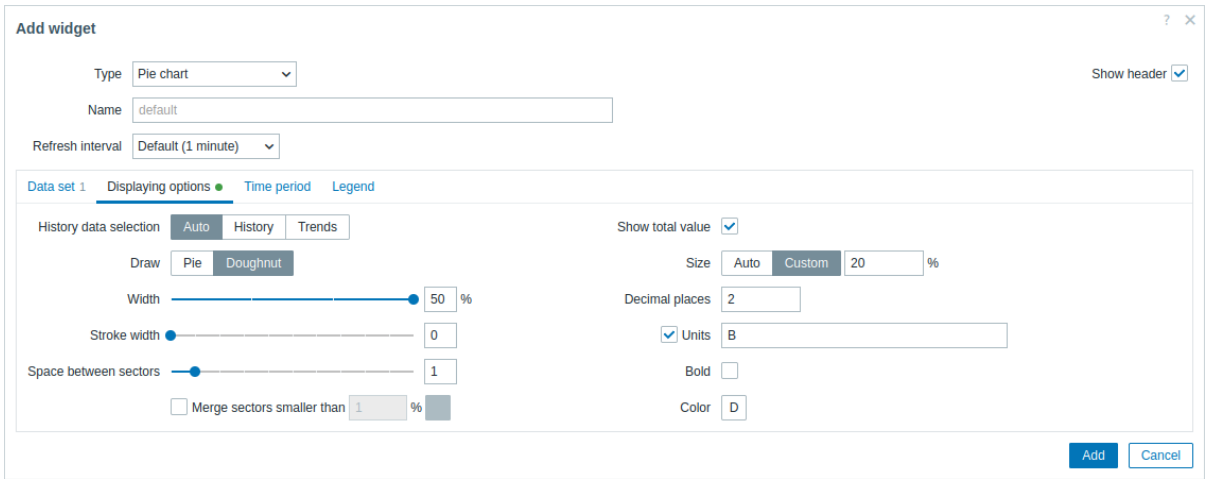
Detalhes da configuração do conjunto de dados

Os conjuntos de dados existentes são exibidos em uma lista. Você pode reorganizar, expandir/recolher, alterar cores e clonar esses conjuntos de dados.

Para mais informações, consulte detalhes da configuração do conjunto de dados no widget *Gráfico*. Esses detalhes também se aplicam ao widget *Gráfico de pizza*.

Opções de exibição

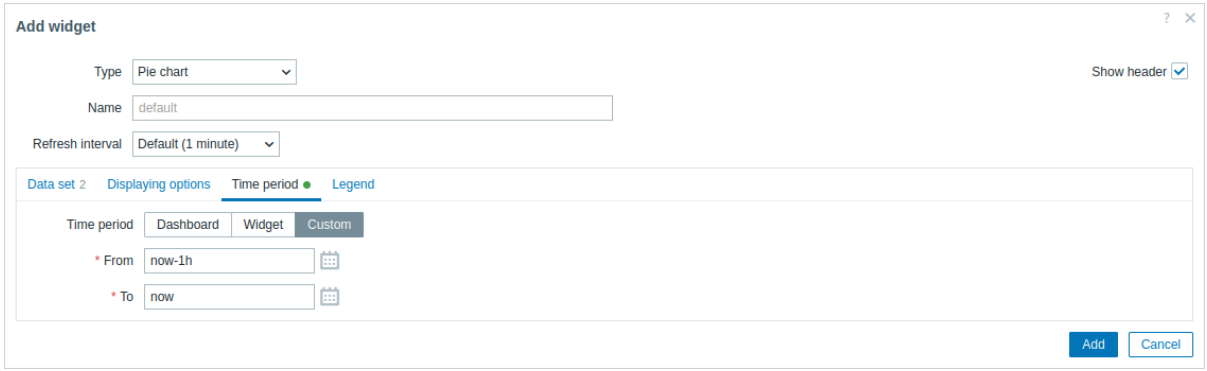
A aba **Opções de exibição** permite definir a seleção dos dados históricos e as opções de visualização do gráfico de pizza:



Seleção de dados históricos	Selecione a fonte de dados: **Automático** - 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); **Rosca** - uma pizza com um círculo vazio no meio (os setores ocupam até 50% do raio).
Espaço entre os setores	Selecione o tamanho do espaço (em unidades de 0 a 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 ativada, selecione a cor dos setores mesclados e o limite percentual (N) para mesclar setores pequenos.
Estilo de desenho:	
Rosca	
Largura	Selecione a largura da rosca: 20, 30, 40 ou 50% (padrão) do raio.
Largura do contorno	Selecione a largura da borda do setor da rosca (0 a 10).
Mostrar valor total	Marque a caixa de seleção para mostrar o valor total no meio do gráfico de rosca.
Tamanho	Selecione a opção de tamanho para o valor total: **Automático** - o tamanho do texto é ajustado automaticamente para caber de forma legível no meio da rosca; **Personalizado** - especifique o tamanho do texto como uma porcentagem da altura total do widget.
Casas decimais	Especifique o número de casas decimais para o valor total (0 a 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 do valor total.

Período de tempo

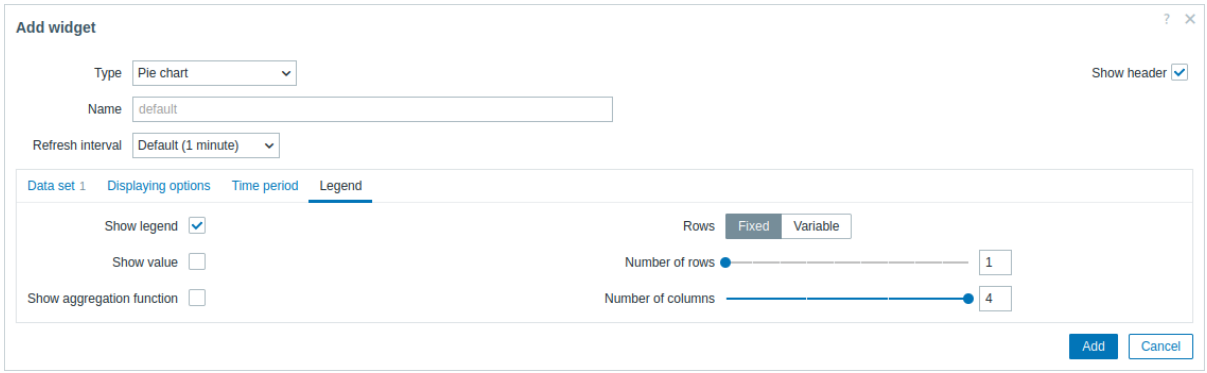
A aba **Período de tempo** permite definir um período de tempo personalizado para as configurações de agregação do gráfico de pizza:



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*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido ao passar o mouse sobre ele. Observe que, independentemente da configuração de *Período de tempo* do widget, os widgets compatíveis ainda podem usá-lo como fonte de dados para o período de tempo.
Widget	Insira ou selecione um widget compatível (*Gráfico*, *Gráfico (clássico)*, *Protótipo de gráfico*) como fonte de dados para o período de tempo. Este parâmetro está disponível quando *Período de tempo* está definido como "Widget".
De	Insira ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".
Até	Insira ou selecione o fim do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".

Legenda

A aba **Legenda** permite personalizar a legenda do gráfico de pizza:

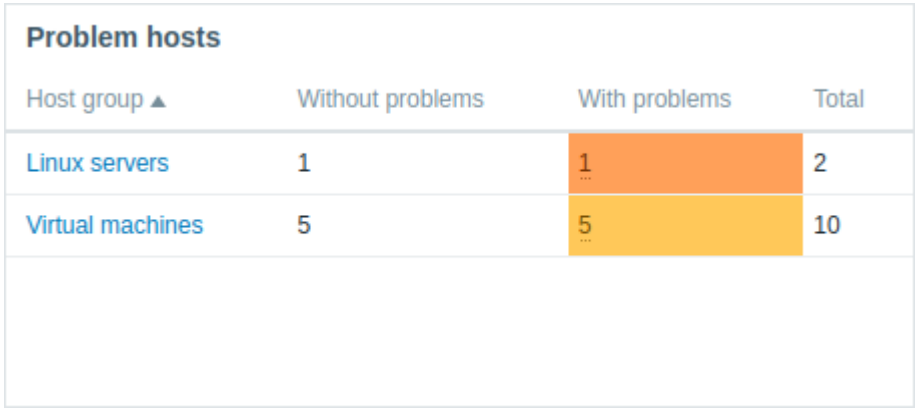


Mostrar legenda	Desmarque esta caixa de seleção para ocultar a legenda no gráfico de pizza (marcada por padrão).
Mostrar valor	Marque esta caixa de seleção para mostrar o valor do item na legenda.
Mostrar função de agregação	Marque esta caixa de seleção para mostrar a função de agregação na legenda.
Linhas	Selecione o modo de exibição das linhas da legenda: **Fixo** - o número de linhas exibidas é determinado pelo valor do parâmetro *Número de linhas*; **Variável** - o número de linhas exibidas é determinado pela quantidade de itens 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 estará disponível se *Mostrar valor* estiver desmarcado.

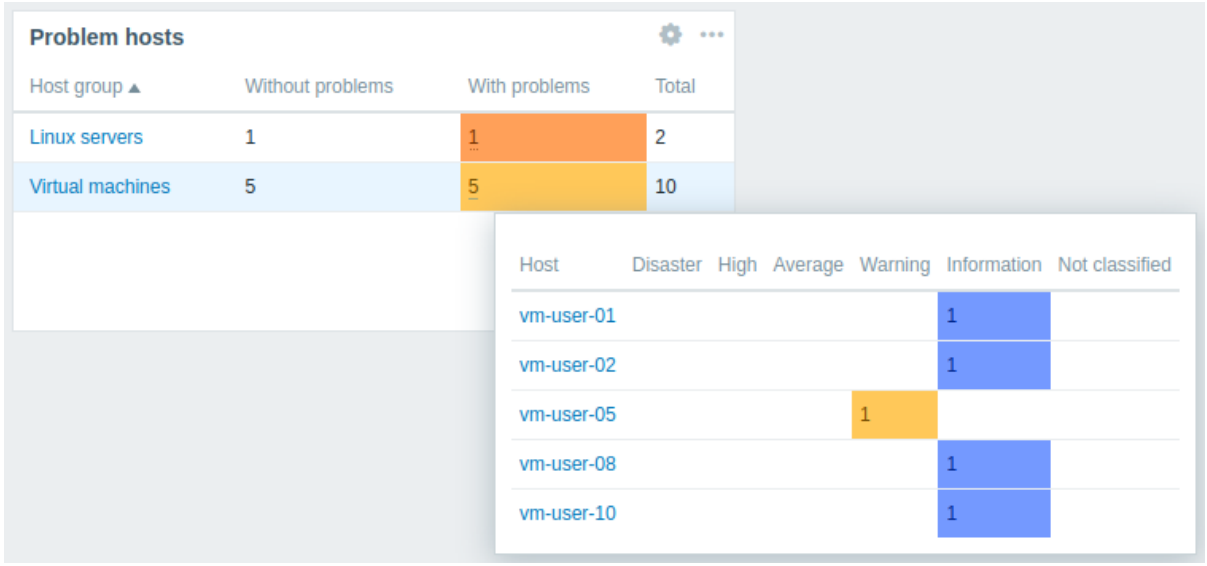
Hosts com problema

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.).

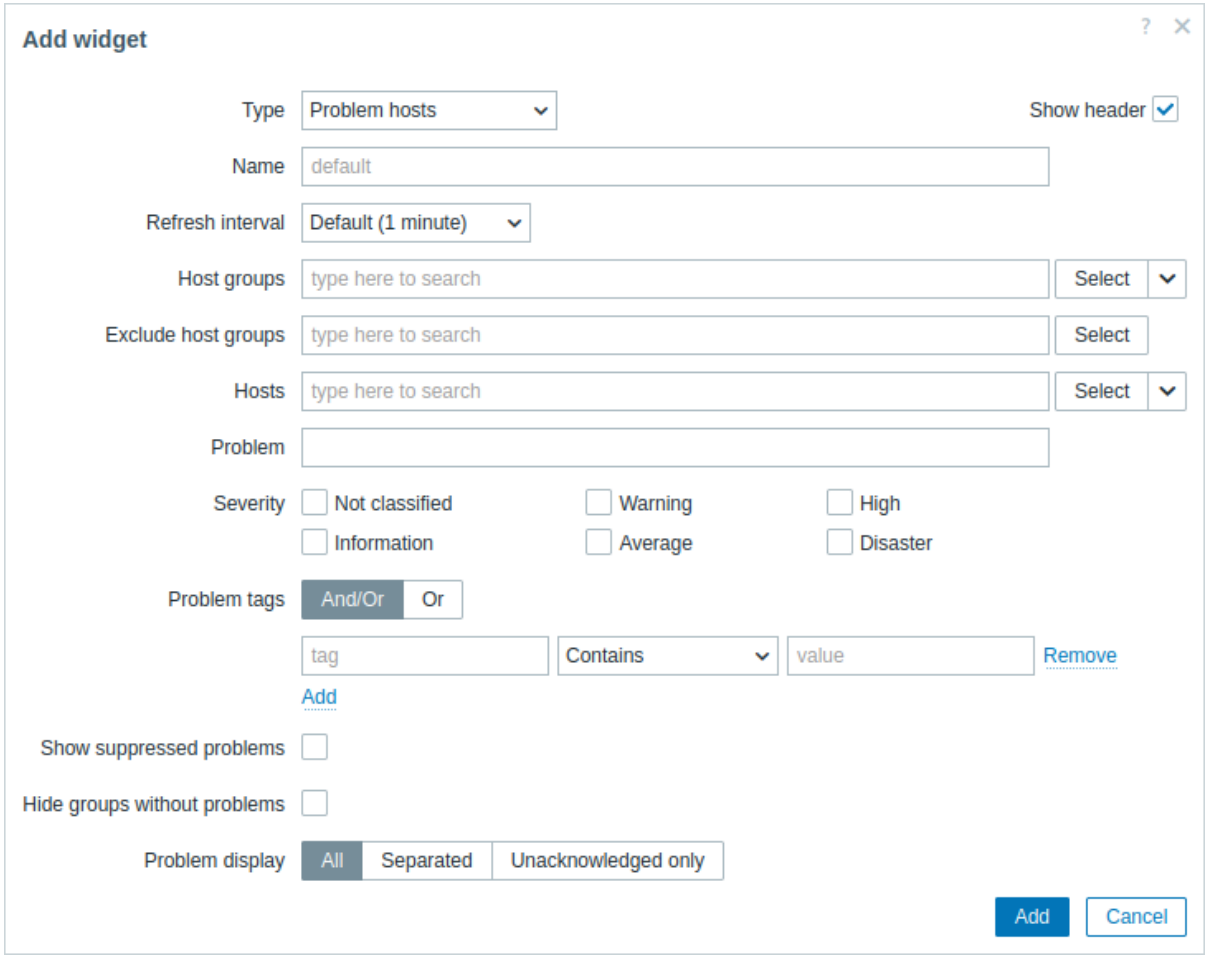


Ao passar o mouse sobre a contagem de problemas, é exibida a lista de hosts com problemas:



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:

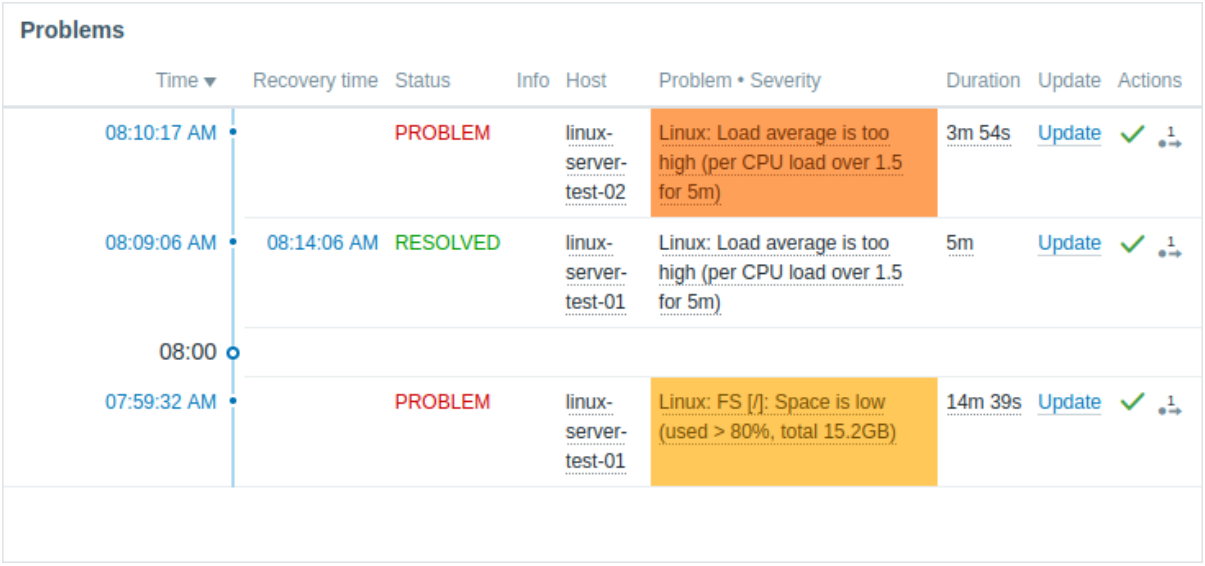
Grupos de hosts	Selecione grupos de hosts para exibir no widget. Alternativamente, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar 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.

Problemas

Visão geral

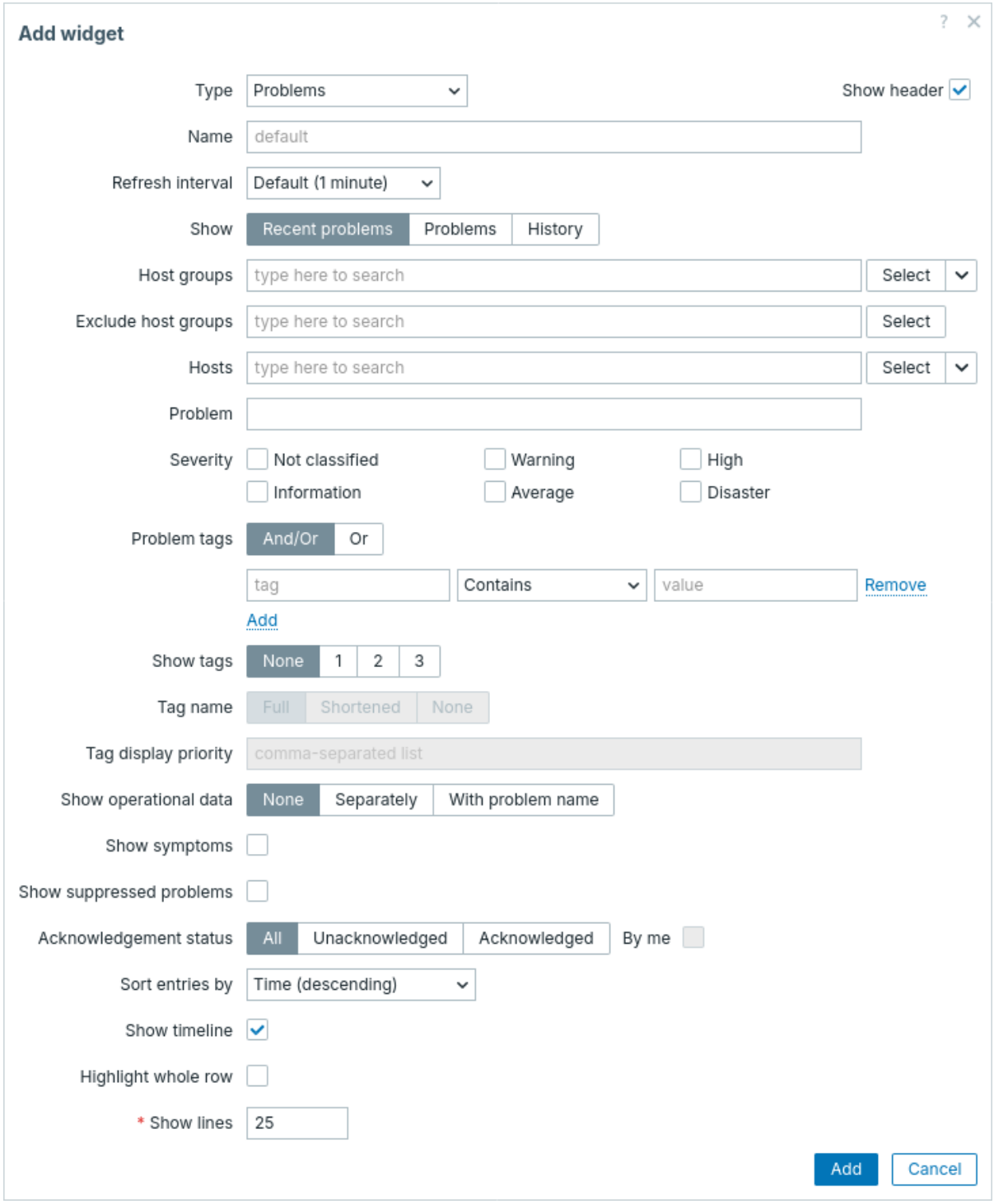
O widget *Problemas* exibe os problemas atuais filtrados por diferentes parâmetros (grupos de hosts, hosts, nomes de problemas, etc.), proporcionando uma visão clara do que precisa de atenção.



Ele exibe os mesmos dados de *Monitoramento* > *Problemas* e pode exibir até 1000 registros.

Configuração

Para configurar, selecione *Problems* como tipo:



Você pode limitar quantos problemas são exibidos no widget de várias maneiras - por status do problema, nome do problema, severidade, host group, host, tag de evento, status de reconhecimento, etc.

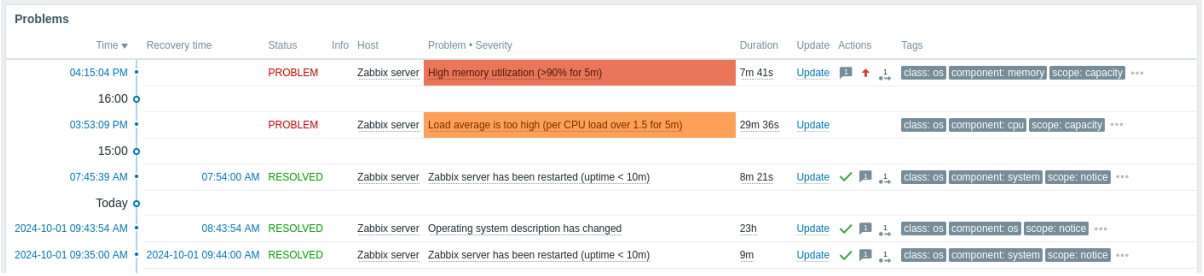
Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

Show	Filtrar por status do problema: **Recent problems** - problemas não resolvidos e recentemente resolvidos são exibidos (padrão); **Problems** - problemas não resolvidos são exibidos; **History** - o histórico de todos os eventos é exibido.
Host groups	Selecione os host groups dos quais deseja exibir problemas no widget. Como alternativa, selecione um widget compatível como fonte de dados para host groups. Este campo tem preenchimento automático, então, ao começar a digitar o nome de um grupo, será oferecida uma lista suspensa com os grupos correspondentes. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Os problemas desses host groups serão exibidos no widget; se nenhum host group for informado, os problemas de todos os host groups serão exibidos. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

Exclude host groups	Selecione os host groups cujos problemas devem ser ocultados no widget. Este campo tem preenchimento automático, então, ao começar a digitar o nome de um grupo, será oferecida uma lista suspensa com os grupos correspondentes. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Os problemas desses host groups não serão exibidos. Por exemplo, os hosts 001, 002, 003 podem estar no Group A e os hosts 002, 003 também no Group B. Se selecionarmos *show* Group A e *exclude* Group B ao mesmo tempo, apenas os problemas do host 001 serão exibidos no widget. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Hosts	Selecione os hosts dos quais deseja exibir problemas no widget. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo tem preenchimento automático, então, ao começar a digitar o nome de um host, será oferecida uma lista suspensa com os hosts correspondentes. Se nenhum host for informado, os problemas de todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Problem	Você pode limitar o número de problemas exibidos pelo nome. Se você inserir uma string aqui, somente os problemas cujo nome contiver a string informada serão exibidos. Macros não são expandidas.
Severity	Marque as severidades de problema para filtrar os problemas exibidos no widget. Se nenhuma severidade for marcada, todos os problemas serão exibidos.
Problem tags	Especifique tags de problema para limitar o número de problemas exibidos no widget. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tag especificados; **Equals** - inclui os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas); **Contains** - inclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas); **Does not exist** - exclui os nomes de tag especificados; **Does not equal** - exclui os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas); **Does not contain** - exclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas, e as condições com o mesmo nome de tag serão agrupadas pela condição *Or*; **Or** - 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 *Tag display priority* (veja abaixo).
Show tags	Selecione o número de tags exibidas: **None** - 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.
Tag name	Selecione o modo de exibição do nome da tag: **Full** - nomes e valores das tags são exibidos por completo; **Shortened** - os nomes das tags são abreviados para 3 símbolos, mas os valores das tags são exibidos por completo; **None** - somente os valores das tags são exibidos; sem nomes.
Tag display priority	Informe a prioridade de exibição das tags de 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 serão sempre exibidas primeiro, substituindo a ordenação natural por ordem alfabética.

Show operational data	Selecione o modo de exibição dos dados operacionais: **None** - nenhum dado operacional é exibido; **Separately** - os dados operacionais são exibidos em uma coluna separada; **With problem name** - adiciona os dados operacionais ao nome do problema, usando parênteses para os dados operacionais. Para detalhes, consulte a descrição do campo *Operational data* em configuração de trigger.
Show symptoms	Marque a caixa de seleção para exibir em sua própria linha os problemas classificados como sintomas.
Show suppressed problems	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não mostrados) devido à manutenção do host ou à supressão de problema única.
Acknowledgement status	Filtre para exibir todos os problemas, apenas problemas não reconhecidos ou apenas problemas reconhecidos. Marque a caixa de seleção adicional para filtrar os problemas que já foram reconhecidos por você.
Sort entries by	Classifique as entradas por: **Time** (decrescente ou crescente); **Severity** (decrescente ou crescente); **Problem name** (decrescente ou crescente); **Host** (decrescente ou crescente). A classificação por **Host** (decrescente ou crescente) não está disponível ao configurar o widget em um template dashboard.
Show timeline	Marque a caixa de seleção para exibir uma linha do tempo visual.
Highlight whole row	Marque a caixa de seleção para destacar a linha inteira para problemas não resolvidos. A cor da severidade do problema é usada para o destaque. *Highlight whole row* não está disponível nos temas de alto contraste.
Show lines	Especifique o número de linhas de problemas a serem exibidas.

Usando o widget

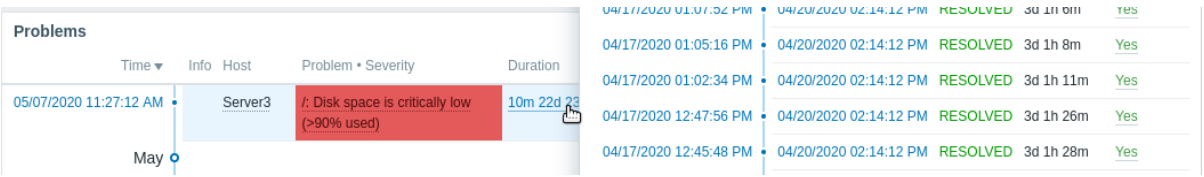


O widget Problems oferece acesso rápido a informações adicionais:

- Clique na data do problema (na coluna Time) ou na data de recuperação (na coluna Recovery Time) 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 do evento.
- Passe o mouse sobre ou clique na duração do problema para ver o popup de evento do problema.
- Clicar em *Update* abre uma janela de Atualizar problema.
- Passe o mouse sobre ou clique no ícone de seta cinza na coluna Actions para ver a lista de ações executadas.

Popup de evento do problema

O popup de evento do problema inclui a lista de eventos de problema para este trigger e, se definido, a descrição do trigger e uma URL clicável.



Para abrir o popup de evento do problema:

- Passe o mouse sobre a duração do problema na coluna *Duration* do widget *Problems*. O popup desaparece assim que você remover o mouse da duração.

- Clique na duração na coluna *Duration* do widget *Problems*. O popup desaparece somente se você clicar na duração novamente.

Problemas por severidade

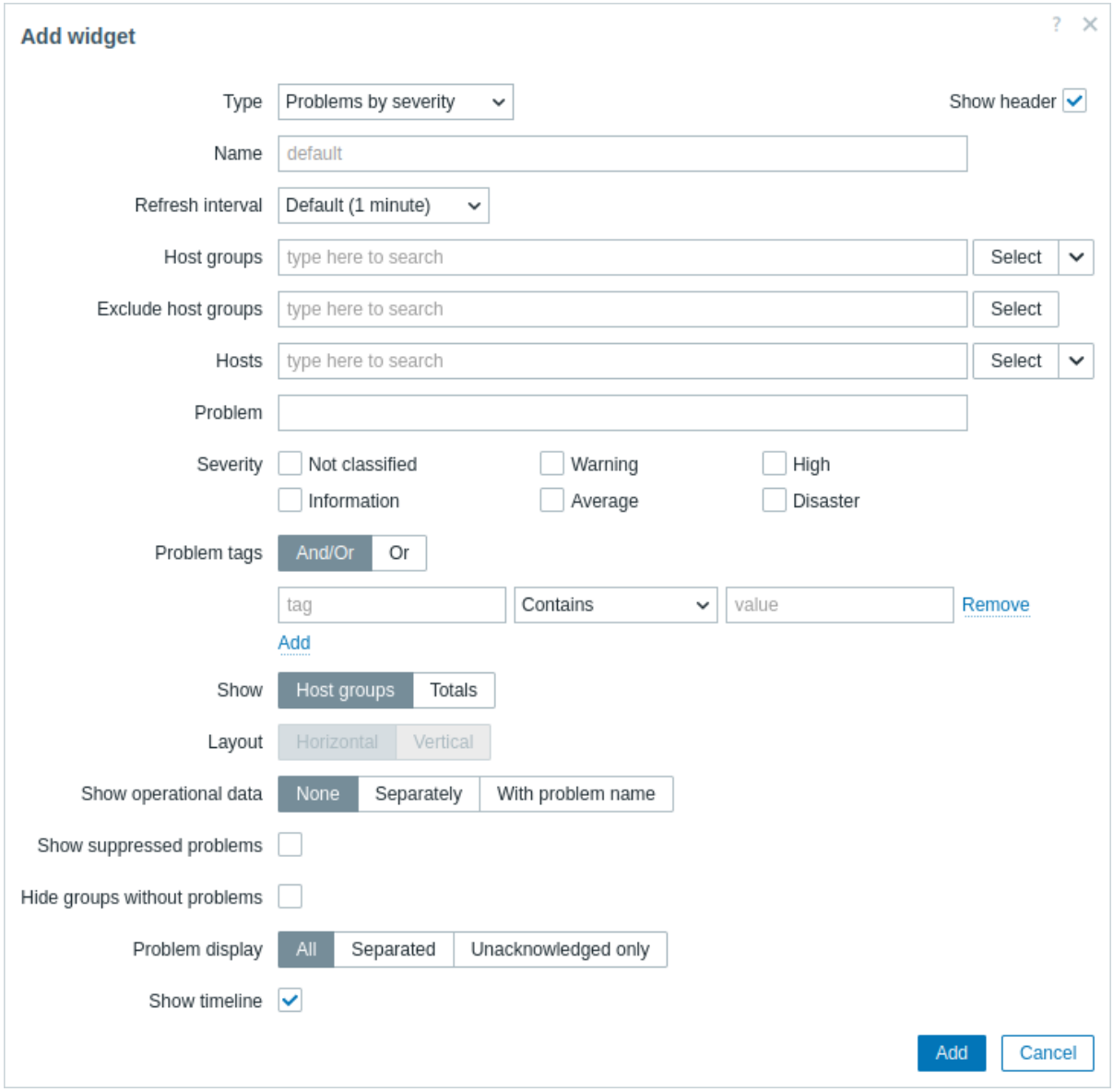
Visão geral

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 *Problems by severity* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

Host groups	Selecione os host groups a serem exibidos no widget. Como alternativa, selecione um widget compatível como fonte de dados para host groups. Este campo tem preenchimento automático, então começar a digitar o nome de um grupo exibirá uma lista suspensa com os grupos correspondentes. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Os dados de host desses host groups serão exibidos no widget; se nenhum host group for informado, todos os host groups serão exibidos. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

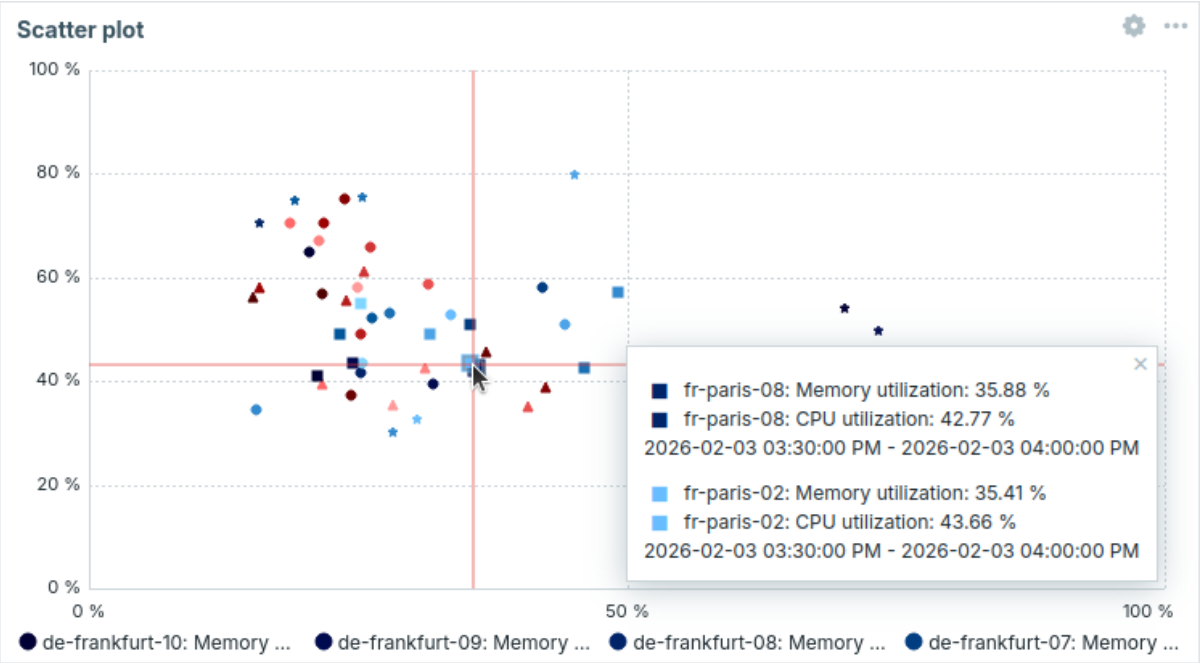
Exclude host groups	Selecione os host groups a serem ocultados do widget. Este campo tem preenchimento automático, então começar a digitar o nome de um grupo exibirá uma lista suspensa com os grupos correspondentes. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Os dados de host desses host groups 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 ao mesmo tempo *mostrar* o Grupo A e *excluir* o Grupo B, apenas os dados do host 001 serão exibidos no dashboard. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
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 tem preenchimento automático, então começar a digitar o nome de um host exibirá uma lista suspensa com os hosts correspondentes. Se nenhum host for informado, todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Problem	Você pode limitar o número de hosts com problemas exibidos pelo nome do problema. Se você inserir uma string aqui, somente os hosts com problemas cujo nome contenha a string informada serão exibidos. Macros não são expandidas.
Severity	Marque as severidades de problema para filtrar os problemas a serem exibidos no widget. Se nenhuma severidade for marcada, todos os problemas serão exibidos.
Problem tags	Especifique tags de problema para limitar o número de problemas exibidos no widget. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tag especificados; **Equals** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contains** - inclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - exclui os nomes de tag especificados; **Does not equal** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Does not contain** - exclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas, as condições com o mesmo nome de tag serão agrupadas pela condição *Or*; **Or** - basta que uma condição seja atendida.
Show	Selecione a opção de exibição: **Host groups** - exibe problemas por host group; **Totals** - exibe o total de problemas para todos os host groups selecionados em blocos coloridos correspondentes à severidade do problema. Este parâmetro não está disponível ao configurar o widget em um template dashboard, 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 *Show* estiver definido como ”Totals”.
Show operational data	Exibir os dados operacionais como: **None** - os dados operacionais não serão exibidos **Separately** - os dados operacionais serão exibidos como uma linha separada **With problem name** - os dados operacionais serão exibidos junto com o nome do problema. Para detalhes, consulte a descrição do campo *Operational data* em trigger configuration.
Show suppressed problems	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não mostrados) devido à manutenção do host.
Hide groups without problems	Marque a opção *Hide groups without problems* para ocultar no widget os dados de host groups sem problemas. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

Problem display	Exibir a contagem de problemas como: **All** - a contagem total de problemas será exibida; **Separated** - a contagem de problemas não reconhecidos será exibida separada como um número da contagem total de problemas; **Unacknowledged only** - somente a contagem de problemas não reconhecidos será exibida.
Show timeline	Marque a caixa de seleção para exibir uma linha do tempo visual.

Gráfico de dispersão

Visão geral

O widget *Gráfico de dispersão* exibe a relação entre duas métricas plotando pontos de dados individuais ao longo dos eixos X e Y. Isso ajuda a revelar padrões, agrupamentos, correlações e valores discrepantes no conjunto de dados.



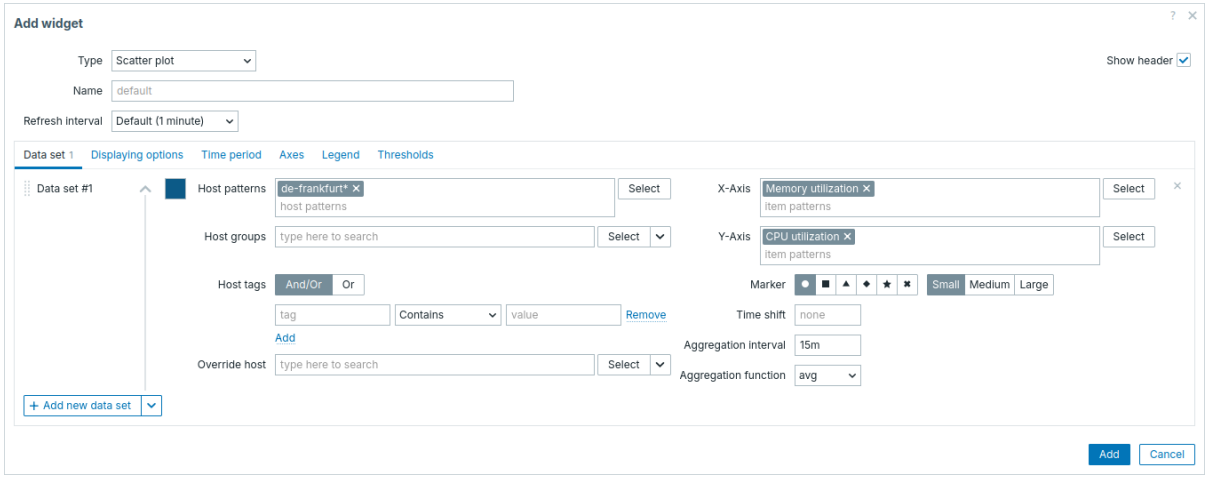
Ao passar o cursor sobre um ponto de dados, é exibida uma dica de ferramenta com uma lista de itens desse ponto de dados e de outros pontos dentro de um raio de 6 px; primeiro ordenados pelos valores do eixo X do gráfico de dispersão em ordem crescente e, em seguida, pelos valores do eixo Y em ordem decrescente.

A seleção de um item na dica de ferramenta transmite seus dados para outros widgets.

As informações exibidas no widget *Gráfico de dispersão* podem ser baixadas como uma imagem PNG selecionando a opção *Baixar imagem* no menu *Ações* do widget ..., que aparece quando você passa o cursor sobre o widget. Dependendo das configurações do navegador, a imagem é salva automaticamente na pasta `Downloads` ou uma caixa de diálogo *Salvar* é aberta, permitindo escolher o local.

Configuração

Para configurar, selecione *Gráfico de dispersão* como tipo:



Conjunto de dados

A aba **Conjunto de dados** permite selecionar dados para o gráfico de dispersão adicionando conjuntos de dados. Dois tipos de conjuntos de dados podem ser adicionados:

- *Padrões de item* - hosts com itens correspondentes são exibidos como pontos de dados. Você pode escolher uma única cor base ou selecionar uma linha da paleta para atribuir cores distintas a cada host com os itens correspondentes.
- *Lista de itens* - os itens selecionados são exibidos como pontos de dados, em que cada ponto representa um par de itens (X e Y). Você pode escolher individualmente a cor de cada conjunto de dados no seletor.

Os pontos de dados são criados somente para os pares de hosts/itens que tenham pelo menos um valor em ambos os eixos.

Por padrão, um conjunto de dados do tipo *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 corresponderem a esses padrões serão exibidos no gráfico de dispersão. Observe que, se o número de pontos de dados exceder 8 mil, `memory_limit` deverá ser aumentado no arquivo de configuração `php.ini`. O limite de memória recomendado para 20 mil pontos de dados é de 512 MB. Padrões curinga podem ser usados para a 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 individualmente, por exemplo, um item chamado *item** se houver outros itens correspondentes (por exemplo, *item2*, *item3*). A especificação de padrões de host e de item é obrigatória para conjuntos de dados do tipo ”Padrões de item”. Consulte também: Detalhes da configuração do conjunto de dados. Para o conjunto de dados **Lista de itens**: Selecione itens para o gráfico de dispersão clicando no botão *Adicionar item*. Você também pode selecionar widgets compatíveis como a fonte de dados dos itens clicando no botão *Adicionar widget*. A especificação de itens ou widgets é obrigatória para conjuntos de dados do tipo ”Lista de itens”. Consulte também: Detalhes da configuração do conjunto de dados. Observe que somente 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 somente os itens configurados no template.

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como a fonte de dados dos grupos de hosts. Este campo oferece preenchimento automático; ao começar a digitar o nome de um grupo, será exibida uma lista suspensa com os grupos correspondentes. A especificação de 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 está disponível somente para conjuntos de dados do tipo *Padrões de item* e não está disponível ao configurar o widget em um dashboard de template.
Tags de host	Especifique tags para filtrar os hosts exibidos no widget. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tag especificados; **É igual a** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tag especificados quando os valores de tag contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Não existe** - exclui os nomes de tag especificados; **Não é igual a** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tag especificados quando os valores de tag contêm a string inserida (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida. Este parâmetro está disponível somente para conjuntos de dados do tipo *Padrões de item* e não está disponível ao configurar o widget em um dashboard de template.
Substituir host	Selecione um widget compatível ou o seletor de host do dashboard como a fonte de dados dos hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Marcador	Escolha o tipo e o tamanho do marcador para os pontos de dados dos hosts no conjunto de dados.
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 para cada item no conjunto de dados dentro do *Intervalo de agregação* selecionado: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **sum** - exibe a soma dos valores; **count** - exibe a quantidade de valores (se existir pelo menos um valor em ambos os eixos); **first** - exibe o primeiro valor; **last** - exibe o último valor. A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia), em vez de todos os valores. Somente dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em numéricos.

Intervalo de agregação	Especifique o intervalo para agregar os 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 do histórico* estiver definida como *Tendências* ou *Automático*), 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 gerar resultados difíceis de interpretar. A especificação de um intervalo de agregação é obrigatória. O intervalo de agregação determina o intervalo de tempo e as coordenadas de um ponto de dados; por exemplo, se os itens do eixo X tiverem 4 valores e os itens do eixo Y tiverem 1 valor, as coordenadas refletirão a agregação de 4 valores no eixo X e 1 no eixo Y dentro do intervalo especificado.

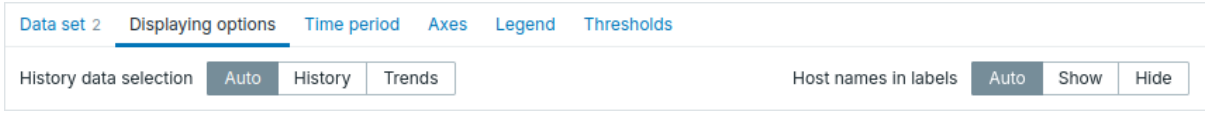
Detalhes da configuração do conjunto de dados

Os conjuntos de dados existentes são exibidos em uma lista. Você pode reorganizar, expandir/recolher, alterar cores e clonar esses conjuntos de dados.

Para mais informações, consulte detalhes da configuração do conjunto de dados no widget *Gráfico*. Esses detalhes também se aplicam ao widget *Gráfico de dispersão*.

Opções de exibição

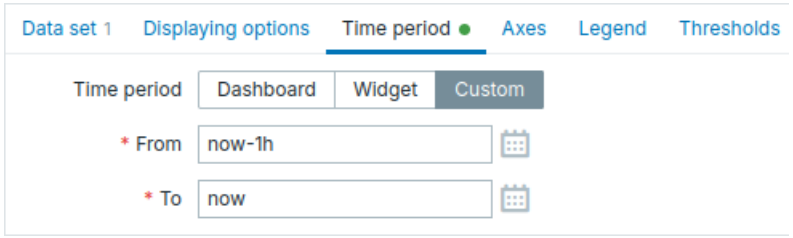
A aba **Opções de exibição** permite definir a seleção de dados históricos:



Seleção de dados históricos	Defina a origem dos dados do gráfico de dispersão: **Automático** - os dados são obtidos de acordo com o algoritmo clássico de gráficos (padrão). **Histórico** - dados do histórico. **Tendências** - dados das tendências.
Nomes dos hosts nos rótulos	Escolha se os nomes dos hosts devem ser exibidos na legenda do gráfico e na dica de ferramenta que aparece ao passar o cursor sobre os dados do gráfico: **Automático** - os nomes dos hosts são exibidos somente quando há mais de um host nos conjuntos de dados (padrão). **Mostrar** - os nomes dos hosts são exibidos. **Ocultar** - os nomes dos hosts ficam ocultos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Período de tempo

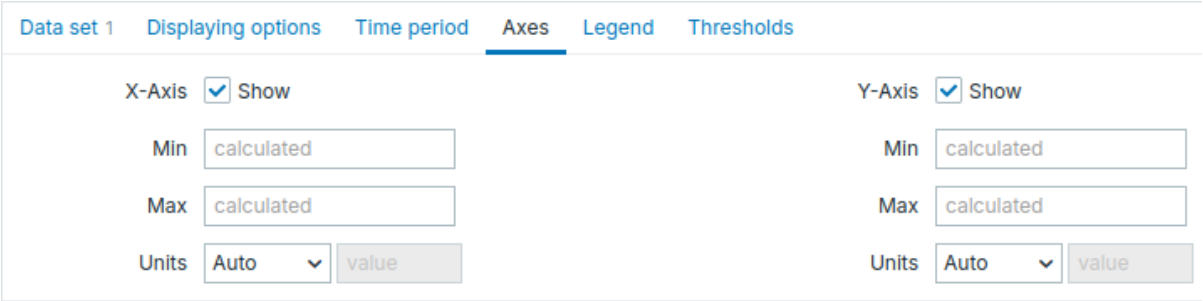
A aba **Período de tempo** permite definir um período de tempo para o qual os dados serão exibidos no gráfico de dispersão:



Período de tempo	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*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido ao passar o mouse sobre ele. 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 quando *Período de tempo* está definido como "Widget".
De	Insira ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".
Até	Insira ou selecione o fim do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período de tempo* está definido como "Personalizado".

Eixos

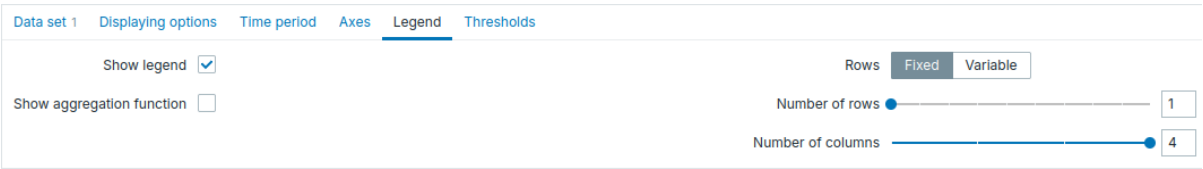
A aba **Eixos** permite personalizar a forma como os eixos são exibidos:



Eixo X/Eixo Y	Desmarque esta caixa de seleção para ocultar o eixo X/eixo Y (marcada por padrão).
Mín.	Defina o valor mínimo do eixo correspondente.
Máx.	Defina o valor máximo do eixo correspondente.
Unidades	Escolha a unidade dos valores dos eixos do gráfico de dispersão na lista suspensa: **Automático** - os valores dos eixos são exibidos usando a unidade do primeiro item no conjunto de dados; **Estático** - os valores dos eixos são exibidos usando a unidade especificada no campo de entrada *valor*; se o campo for deixado em branco, somente valores numéricos serão exibidos.

Legenda

A aba **Legenda** permite personalizar a legenda do gráfico de dispersão:

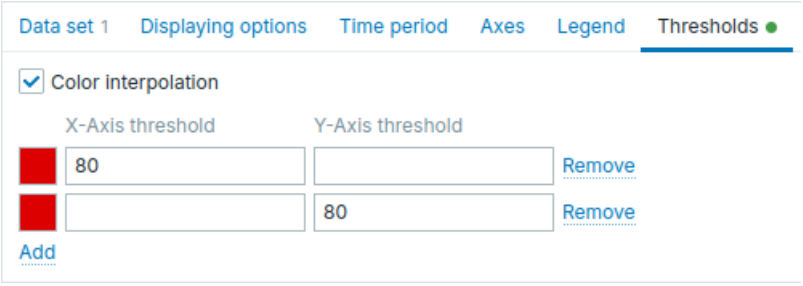


Mostrar legenda	Desmarque esta caixa de seleção para ocultar a legenda no gráfico de dispersão (marcada por padrão).
Mostrar função de agregação	Marque esta caixa de seleção para mostrar a função de agregação na legenda.
Linhas	Selecione o modo de exibição das linhas da legenda: **Fixo** - o número de linhas exibidas é determinado pelo valor do parâmetro *Número de linhas*; **Variável** - o número de linhas exibidas é determinado pela quantidade de itens 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 mín/méd/máx* estiver desmarcado.

Limites

A aba **Limites** permite configurar limites que determinam as cores dos marcadores de host:



Interpolação de cores	Marque a caixa de seleção para habilitar uma transição suave entre as cores dos limites para os marcadores de host. Este parâmetro se aplica quando dois ou mais limites são definidos para o eixo X, o eixo Y ou ambos os eixos simultaneamente.
Limite do eixo X/eixo Y	Clique em *Adicionar* para adicionar um limite, selecione uma cor para o limite no seletor de cores e especifique um valor numérico. Os limites podem ser aplicados a um único eixo, a ambos os eixos separadamente ou a ambos os eixos simultaneamente. Por exemplo, um único limite com X=80 e Y=80 colorirá os marcadores quando ambas as condições forem atendidas. A lista de limites será classificada em ordem crescente ao ser salva. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são compatíveis. Mapeamentos de valores são compatíveis.

Relatório de SLA

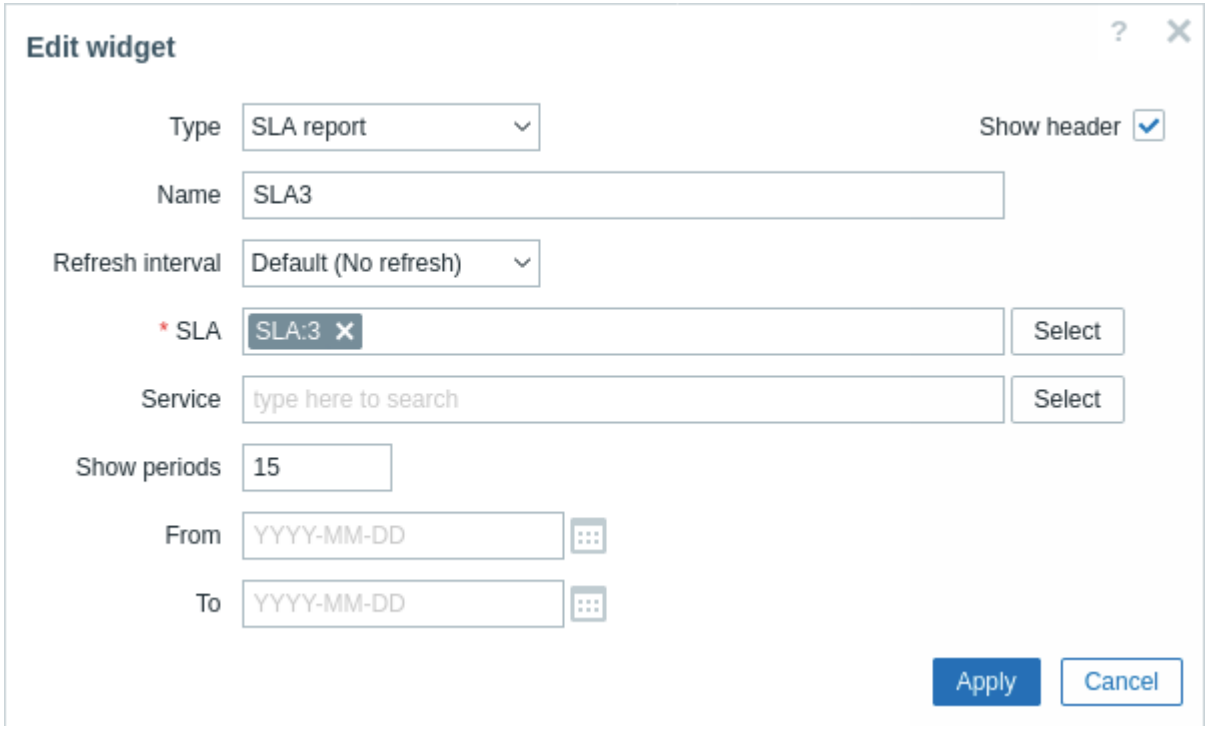
Visão geral

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:



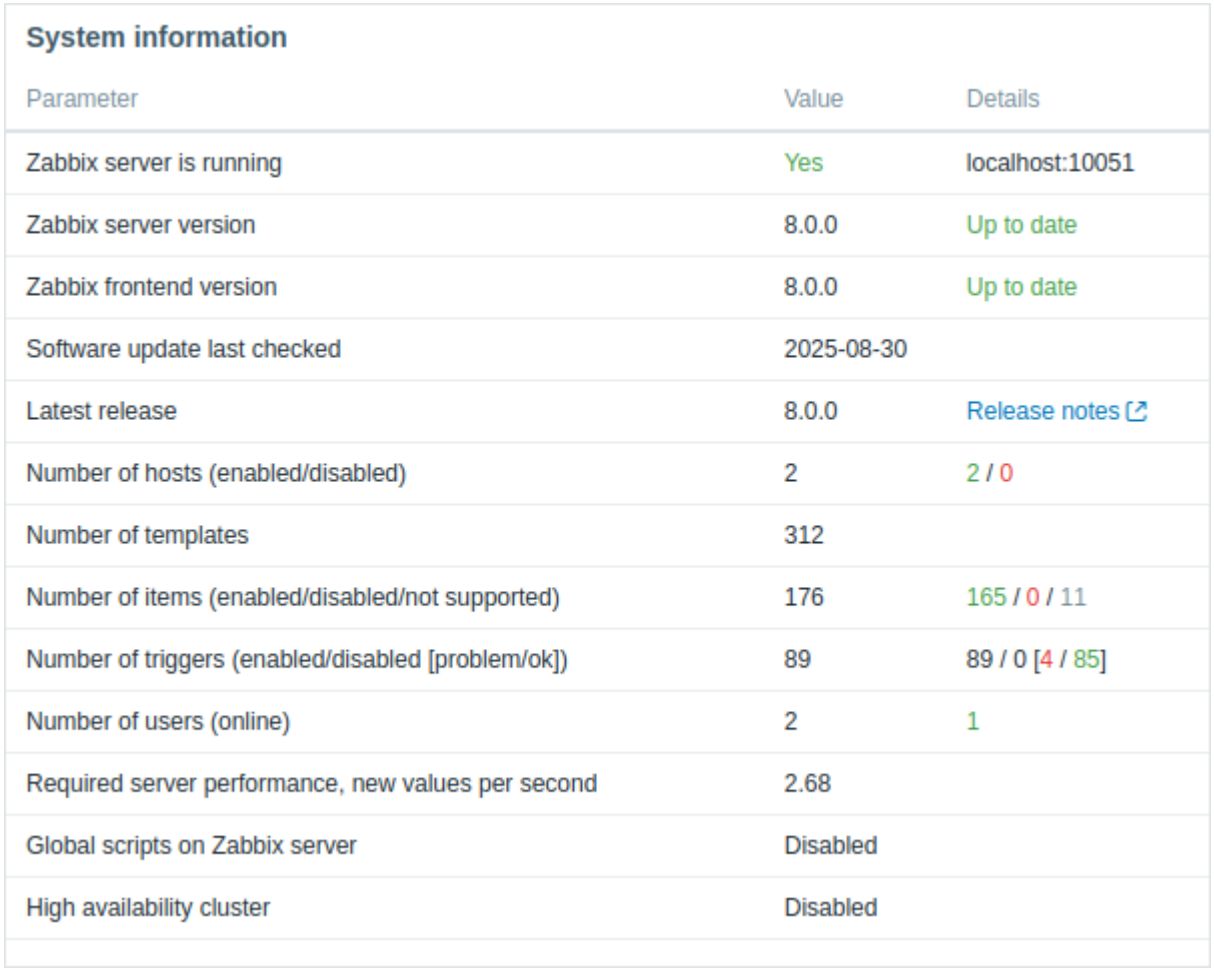
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

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.

Informações do sistema

Visão geral

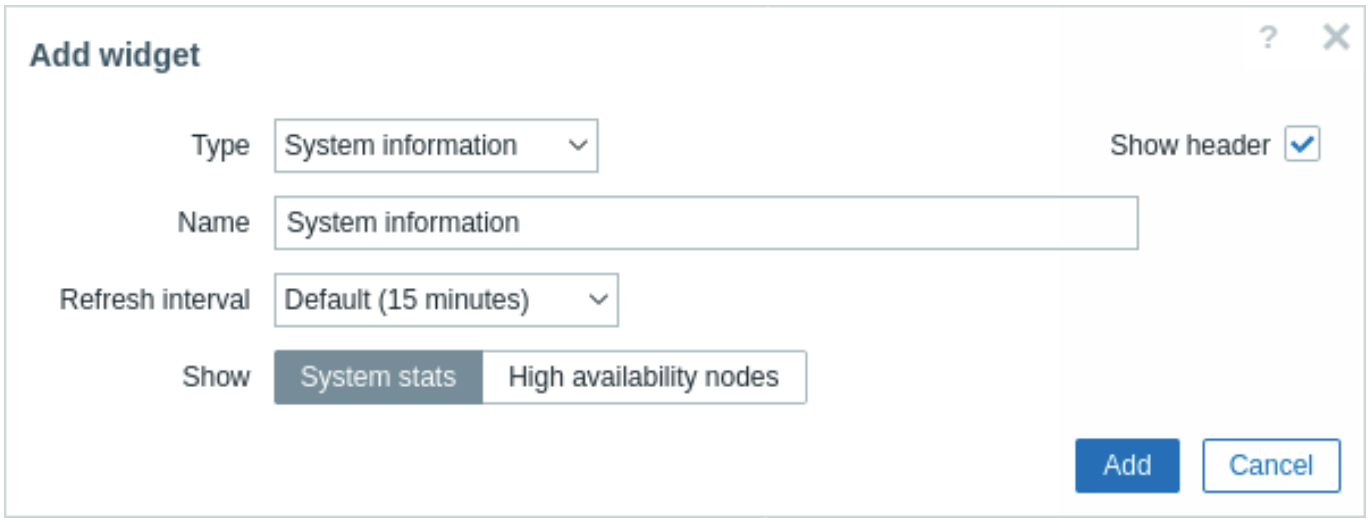
O widget *System information* exibe um resumo dos principais dados do server Zabbix e do sistema e, quando a alta disponibilidade está habilitada, detalhes dos nós de alta disponibilidade.



Ele exibe os mesmos dados que *Reports* > *System information*, exceto *Zabbix server ID*.

Configuração

Para configurar, selecione *Informações do sistema* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

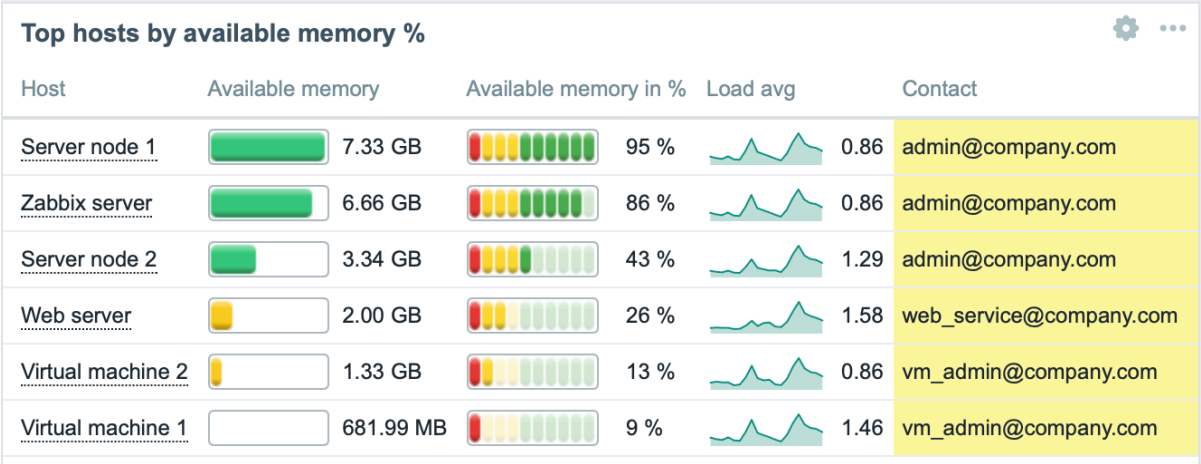
Exibir	Selecione o que exibir: **Estatísticas do sistema** - exibe um resumo dos principais dados do Zabbix server e do sistema; **Nós de alta disponibilidade** - exibe o status dos nós de alta disponibilidade (se o cluster de alta disponibilidade estiver habilitado).

Exibir detalhes da verificação de atualização do software	Marque a caixa de seleção para exibir os detalhes da verificação de atualização do software do Zabbix. Essa opção só estará disponível se a verificação de atualização do software estiver habilitada na configuração do server do Zabbix e se ”Estatísticas do sistema” estiver selecionado no campo *Exibir*.

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.

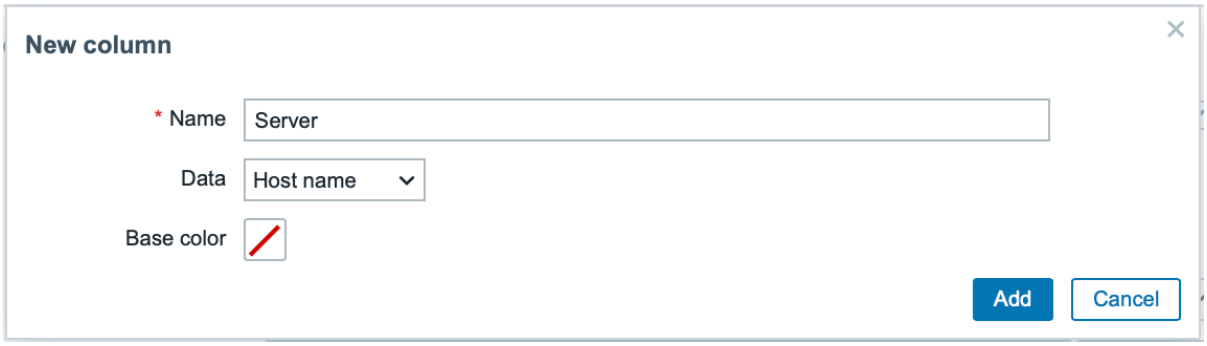
Tags de host	Especifique tags para limitar o número de hosts exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **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.

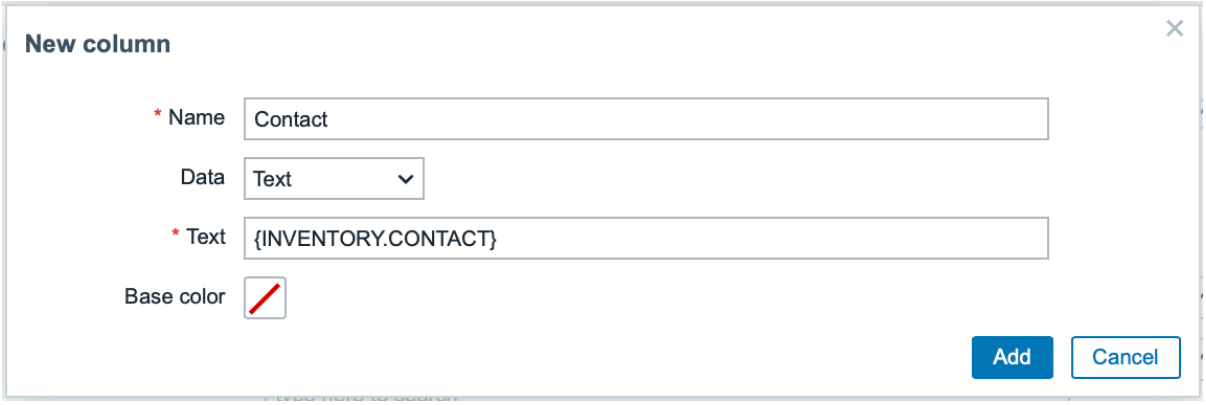


Parâmetros suportados:

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.

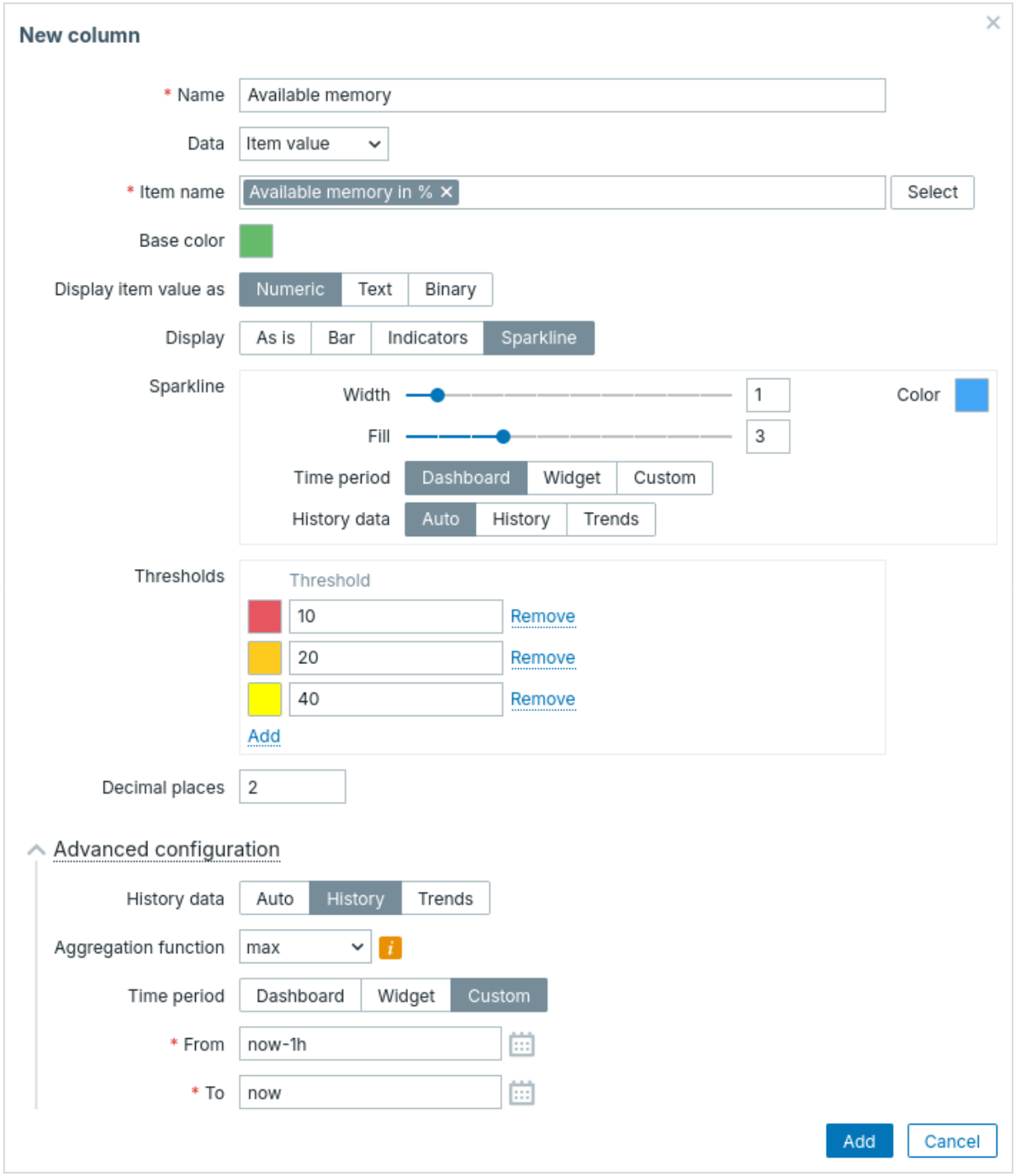


Parâmetros suportados:

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 de inventário macros.

Coluna de valor do item

A coluna de valor do item é usada para exibir o valor do item especificado; ela oferece suporte a vários formatos e opções de exibição de valores.



Parâmetros compatíveis:

Name	Nome da coluna.
Data	Tipo de dados a ser exibido na coluna; selecione **Item value**.
Item name	Selecione um item; o nome dele é usado para localizar e exibir itens com o mesmo nome em todos os hosts selecionados. Se um host tiver vários itens desse tipo, o widget exibirá o item cuja chave aparecer primeiro 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 template dashboard, somente itens configurados no template poderão ser selecionados.
Base color	Cor de fundo da coluna; cor de preenchimento se os dados de *Item value* forem exibidos como barra/indicadores. A cor padrão pode ser substituída por uma cor personalizada se o valor do item ultrapassar um dos limites especificados.
Display item value as	Formato para exibir o valor do item: **Numeric**, **Text** ou **Binary**. A opção selecionada determina quais parâmetros adicionais estarão disponíveis. Consulte a lista de parâmetros de cada formato abaixo.
Advanced configuration	

History data	Obter dados do histórico ou das tendências: **Auto** - seleção automática; **History** - obter dados do histórico; **Trends** - obter dados das tendências. Este parâmetro está disponível somente para valores numéricos de itens. Para valores de texto e binários, os dados serão sempre obtidos do histórico.
Aggregation function	Especifique qual função de agregação usar no *Time period* selecionado: **min** - exibir o menor valor; **max** - exibir o maior valor; **avg** - exibir o valor médio; **count** - exibir a quantidade de valores; **sum** - exibir a soma dos valores; **first** - exibir o primeiro valor; **last** - exibir o último valor. Se **not used** for selecionado, o valor mais recente será exibido (sem agregação). A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia), em vez do valor mais recente. Somente dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em numéricos.
Time period	Especifique o período a ser usado para agregar valores. Selecione a fonte de dados para o período: **Dashboard** - usar o seletor de período do dashboard; **Widget** - usar um widget compatível (definido no parâmetro *Widget*); **Custom** - usar um período personalizado definido nos parâmetros *From* e *To*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido quando o mouse passar sobre ele. Observe que, independentemente da configuração de *Time period* do widget, widgets compatíveis ainda poderão usá-lo como fonte de dados para o período. Este parâmetro não está disponível se *Aggregation function* estiver definido como ”not used”.
Widget	Insira ou selecione um widget compatível como fonte de dados para o período. Este parâmetro estará disponível se *Time period* estiver definido como ”Widget”.
From	Insira ou selecione o início do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro estará disponível se *Time period* estiver definido como ”Custom”.
To	Insira ou selecione o fim do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro estará disponível se *Time period* estiver definido como ”Custom”.

Parâmetros de **valor numérico do item**:

Display	Defina como o valor deve ser exibido: **As is** - como texto comum; **Bar** - como uma barra sólida preenchida com cor; **Indicators** - como uma barra segmentada preenchida com cor; **Sparkline** - mini gráfico de linhas.
Thresholds	Especifique os valores de limite nos quais a cor de fundo/preenchimento deverá mudar. A lista será classificada em ordem crescente ao ser salva. Para sparklines, os limites serão aplicados somente ao último valor do item.
Decimal places	Especifique quantas casas decimais serão exibidas com o valor.
Bar/Indicators	
Min	Valor mínimo.
Max	Valor máximo.
Sparkline	
Width	Defina a espessura da linha do gráfico usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.
Color	Selecione a cor da linha e do preenchimento.
Fill	Defina o nível de transparência da cor de preenchimento usando o controle deslizante ou inserindo manualmente um valor no intervalo de 0 a 10.

Time period	Especifique o período dos valores a serem incluídos no gráfico sparkline. Selecione a fonte de dados para o período: **Dashboard** - usar o seletor de período do dashboard **Widget** - usar um widget compatível (definido no parâmetro *Widget*); **Custom** - usar um período personalizado definido nos parâmetros *From* e *To*; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido quando o mouse passar sobre ele. Observe que, independentemente da configuração de *Time period* do widget, widgets compatíveis ainda poderão usá-lo como fonte de dados para o período.
Widget	Insira ou selecione um widget compatível como fonte de dados para o período. Este parâmetro estará disponível se *Time period* estiver definido como "Widget".
From	Insira ou selecione o início do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro estará disponível se *Time period* estiver definido como "Custom".
To	Insira ou selecione o fim do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro estará disponível se *Time period* estiver definido como "Custom".
History data	Obter dados do histórico ou das tendências: **Auto** - seleção automática; **History** - obter dados do histórico; **Trends** - obter dados das tendências.

Parâmetros de **valor de texto do item**:

Highlights	Especifique as expressões regulares cuja correspondência deverá alterar a cor de fundo/preenchimento.

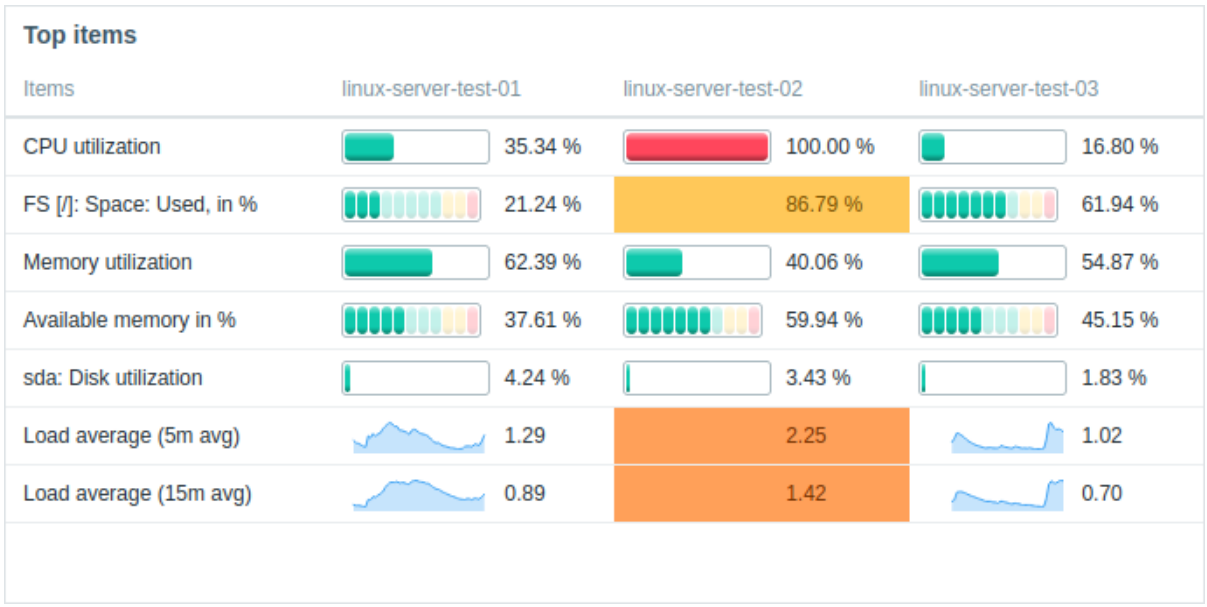
Parâmetros de **valor binário do item**:

Show thumbnail	Especifique se deve ser criada e exibida uma miniatura da imagem que contém dados binários ou se deve ser exibido um hiperlink *Show* que leva à imagem em tamanho completo na coluna de valor.

Principais itens

Visão geral

O widget *Top items* exibe os maiores/menores valores para os items selecionados, fornecendo uma visão rápida do seu desempenho.

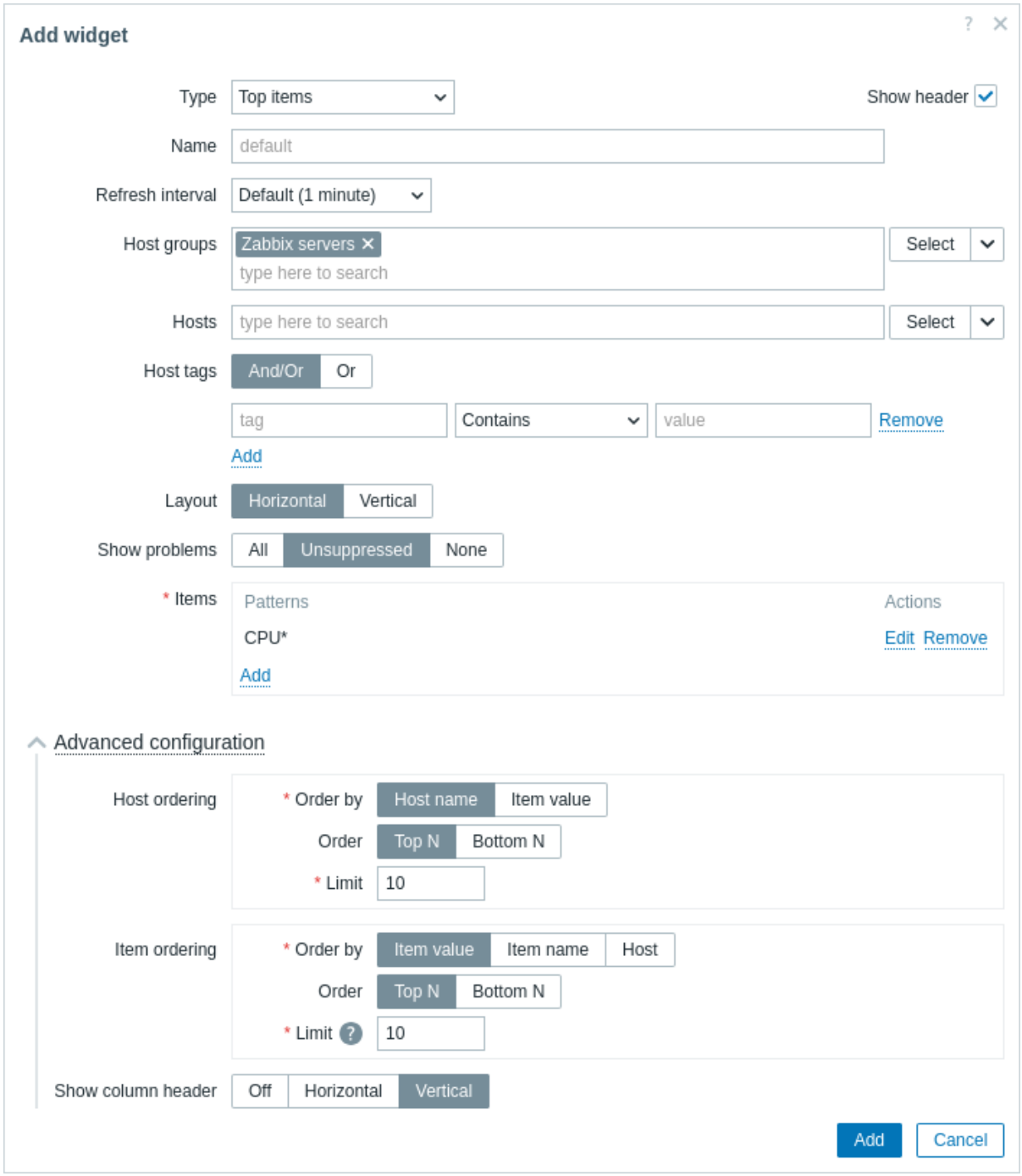


Por padrão, o widget exibe valores das últimas 24 horas. Esse período pode ser alterado em *Administração* > *Geral* > *GUI* (*Período máximo de exibição do histórico*).

Clicar no valor do item abre o menu do item.

Configuração

Para configurar, selecione *Top items* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

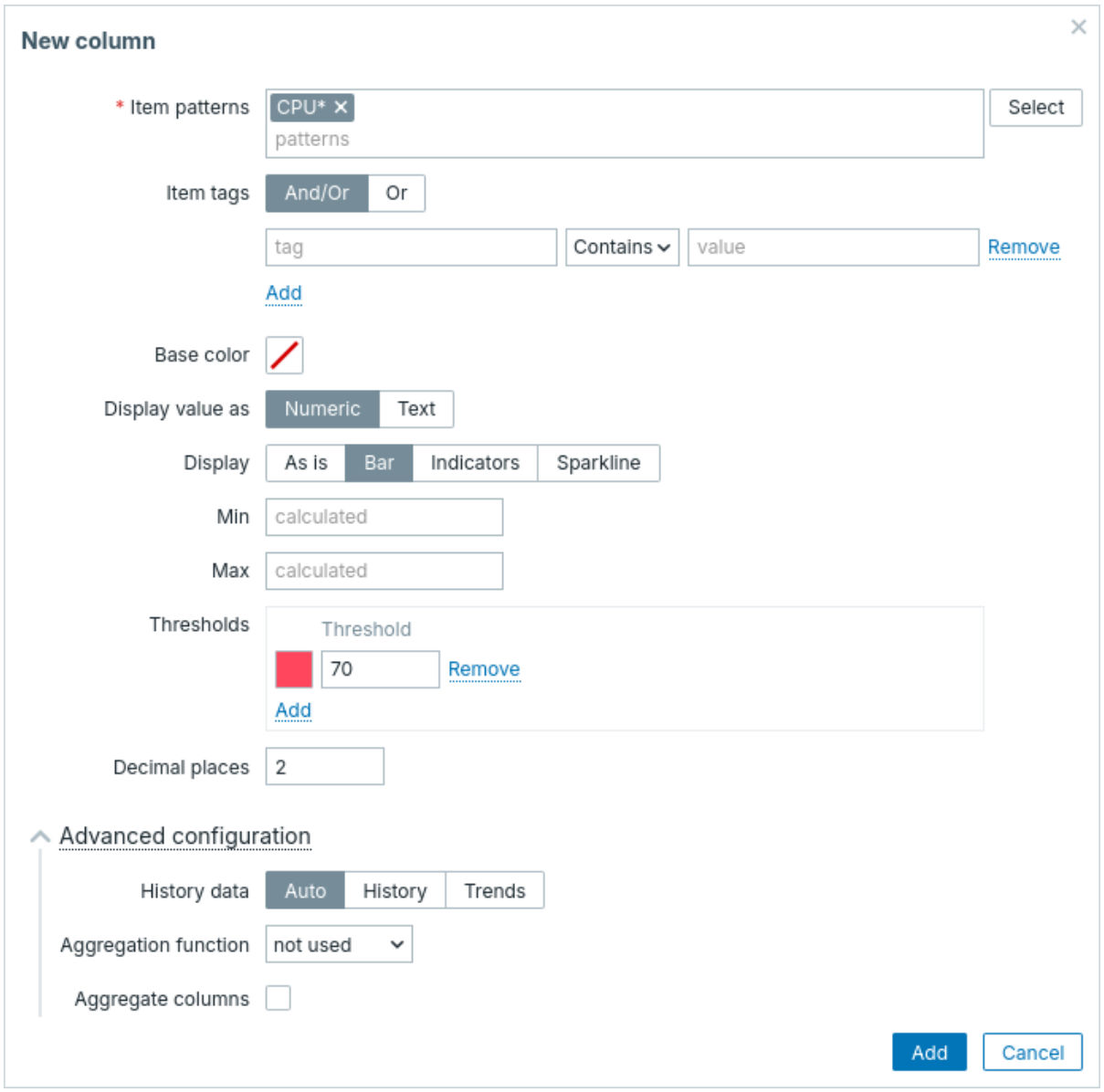
Host groups	Selecione host groups. Alternativamente, selecione um widget compatível como fonte de dados para host groups. Este campo possui auto-completar, então começar a digitar o nome de um grupo exibirá uma lista suspensa com os grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

Hosts	Selecione hosts. Alternativamente, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo possui auto-completar, então começar a digitar o nome de um host exibirá uma lista suspensa com os hosts correspondentes. Este parâmetro não está disponível ao configurar o widget em um template dashboard.
Host tags	Especifique tags para limitar a quantidade de dados de host exibidos no widget. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tag especificados; **Equals** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contains** - inclui os nomes de tag especificados em que os valores da tag contêm a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - exclui os nomes de tag especificados; **Does not equal** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Does not contain** - exclui os nomes de tag especificados em que os valores da tag contêm a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição *Or*; **Or** - 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.
Show problems	Filtre quais problemas exibir com base no status deles (all, unsuppressed, none). Os problemas são indicados com valores de item coloridos, substituindo as configurações de exibição da coluna item. As cores são baseadas na severidade do problema, que pode ser personalizada em *Administration > General > Trigger displaying options*. As severidades dos problemas podem ser alteradas ao atualizar problemas.
Items	Adicione padrões de item ou items específicos para exibição (veja Configuração de coluna).
Advanced configuration	
Host ordering	Selecione as opções de ordenação para a coluna/linha de host.
Order by	Ordenar hosts por: **Host name** - os hosts serão ordenados pelo nome do host; **Item value** - os hosts serão ordenados pelo valor dos items selecionados.
Order	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 *Limit*.
Limit	Informe o limite de hosts exibíveis (intervalo 1-1000; padrão 10). Esse valor se tornará o valor de N no campo *Order*.
Item ordering	Selecione as opções de ordenação para a coluna/linha de item.

Order by	Ordenar items por: **Item value** - os items serão ordenados pelo valor do item; **Item name** - os items serão ordenados pelo nome do item; **Host** - os items serão ordenados pelo padrão de host selecionado.
Order	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 *Limit*.
Limit	Informe o limite de items exibíveis (intervalo 1-1000; padrão 10). Esse valor se tornará o valor de N no campo *Order*.
Show column header	Selecione as opções de exibição do cabeçalho da coluna: **Off** - não exibir o cabeçalho da coluna; **Horizontal** - exibir o texto horizontalmente no cabeçalho; **Vertical** - exibir 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:

Exibição	Defina como o valor deve ser exibido: **Como está** - como texto comum; **Barra** - como uma barra sólida preenchida com cor; **Indicadores** - como uma barra segmentada preenchida com cor; **Sparkline** - mini gráfico de linhas.
Mín.	Valor mínimo para a exibição de barras/indicadores.
Máx.	Valor máximo para a exibição de barras/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 está disponível somente para a exibição de sparklines.
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 está disponível somente para a exibição de sparklines.
Cor	Selecione a cor da linha e do preenchimento. Este parâmetro está disponível somente para a exibição de sparklines.
Consulte *Configuração avançada* para obter a descrição dos campos relacionados ao período e à seleção de dados históricos.	
Limiares	Especifique os valores de limiar nos quais a cor do plano de fundo/preenchimento deve mudar. A lista será classificada em ordem crescente ao ser salva. Para sparklines, os limiares são aplicados somente ao último valor do item.
Casas decimais ***Configuração avançada***	Especifique quantas casas decimais serão exibidas com o valor.

Dados históricos	Obtenha dados do histórico ou das tendências: **Automático** - seleção automática; **Histórico** - obter dados históricos; **Tendências** - obter dados de tendências. Este parâmetro está disponível somente para valores numéricos de item. Para valores de texto, os dados sempre serão obtidos do histórico.
Função de agregação	Especifique qual função de agregação usar no *Período* selecionado: **min** - exibir o menor valor; **max** - exibir o maior valor; **avg** - exibir o valor médio; **count** - exibir a quantidade de valores; **sum** - exibir a soma dos valores; **first** - exibir o primeiro valor; **last** - exibir o último valor. Se **não usar** for selecionado, o valor mais recente será exibido (sem agregação). A agregação permite exibir um valor agregado para o intervalo especificado (5 minutos, uma hora, um dia), em vez do valor mais recente. Somente dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, os dados não numéricos serão convertidos em numéricos.
Período	Especifique o período a ser usado para agregar valores. Selecione a fonte de dados para o período: **Dashboard** - usar o seletor de período do dashboard; **Widget** - usar um widget compatível (definido no parâmetro *Widget*); **Personalizado** - usar um período personalizado definido nos parâmetros *De* e *Até*; quando definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o horário definido ao passar o mouse sobre ele. Observe que, independentemente da configuração de *Período* do widget, os widgets compatíveis ainda podem usá-lo como fonte de dados para o período. Este parâmetro não está disponível se *Função de agregação* estiver definida como ”não usar”.
Widget	Insira ou selecione um widget compatível como fonte de dados para o período. Este parâmetro está disponível quando *Período* está definido como ”Widget”.
De	Insira ou selecione o início do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período* está definido como ”Personalizado”.
Até	Insira ou selecione o fim do período. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é compatível. Este parâmetro está disponível quando *Período* está definido como ”Personalizado”.
Agregar colunas	Quando selecionado, agrupa todos os itens que correspondem aos padrões de item da coluna em uma única coluna/linha combinada. Os valores dos itens correspondentes serão agregados usando a *Função de agregação da coluna* selecionada e exibidos sob o *Nome da coluna combinada*. Padrão: desativado (não selecionado).Este parâmetro é exibido somente quando *Exibir valor como* está definido como **Numérico**.
Função de agregação da coluna	Escolha como os valores dos padrões de item agrupados serão combinados. Opções disponíveis: **min**, **max**, **avg**, **count**, **sum** (padrão). Este parâmetro é exibido somente quando *Agregar colunas* está selecionado.
Nome da coluna combinada	Insira o nome de exibição da coluna agregada. Este campo é obrigatório quando *Agregar colunas* está selecionado e se torna o cabeçalho da coluna para os valores combinados.

Parâmetros de coluna para valores de texto:

Destaques	Especifique as expressões regulares cuja correspondência deve fazer com que a cor do plano de fundo/preenchimento seja alterada.

Principais triggers

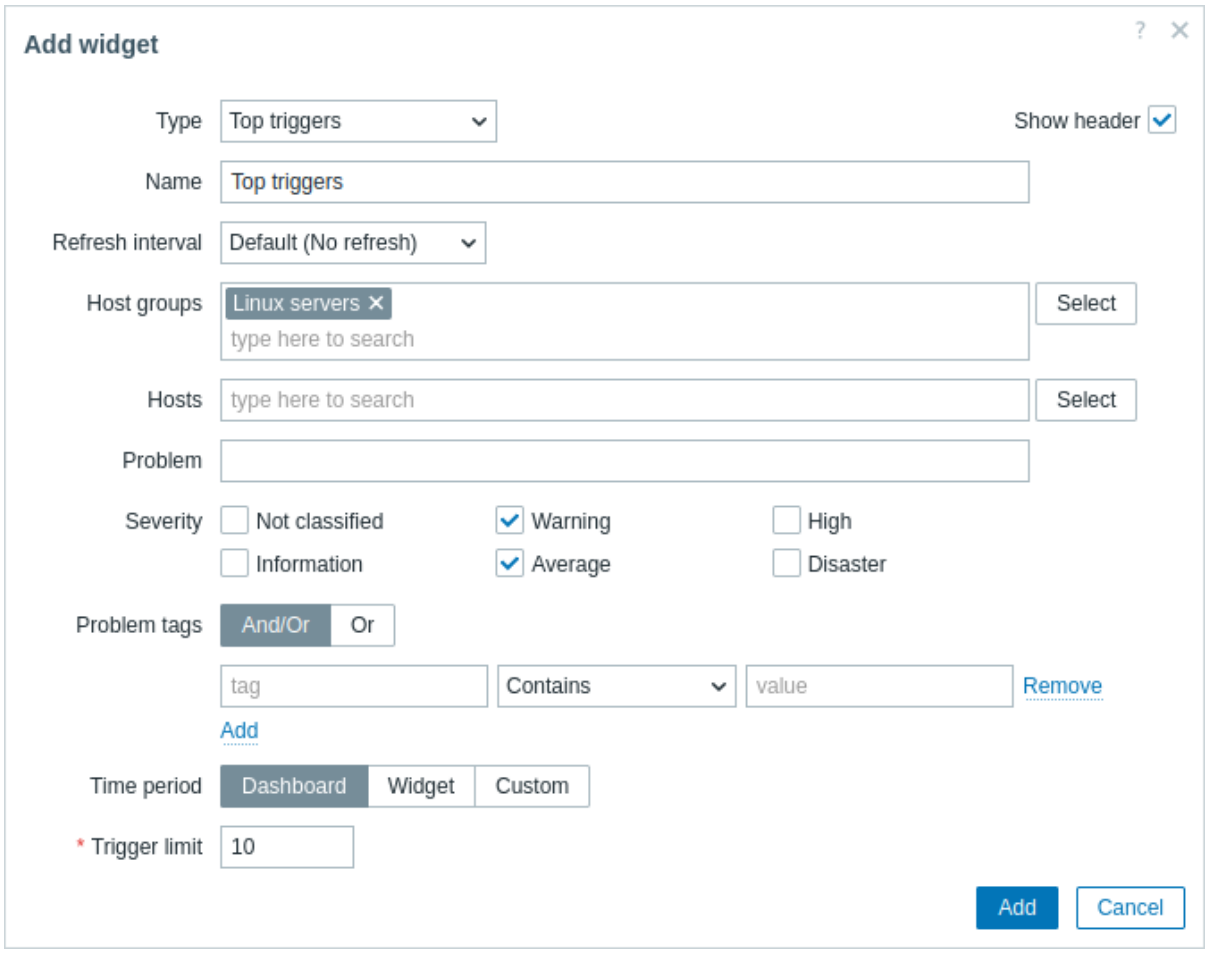
Visão geral

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.

Visão geral de trigger

Visão geral

O widget *Trigger overview* exibe os estados atuais dos triggers do host como uma tabela de blocos coloridos. Ele fornece uma visão visual rápida da saúde e da atividade dos triggers em diferentes hosts.



O widget usa vários elementos visuais para transmitir detalhes dos triggers:

- As cores dos blocos para triggers em PROBLEM correspondem às severidades do problema.
- Blocos piscando indicam alterações recentes no estado do trigger (nos últimos 2 minutos).
- Setas cinzas indicam triggers com dependências (passe o mouse sobre elas para ver os detalhes).
- Ícones de caixa de seleção indicam problemas reconhecidos; todos os problemas ativos ou resolvidos do trigger devem ser reconhecidos para que esse ícone apareça.

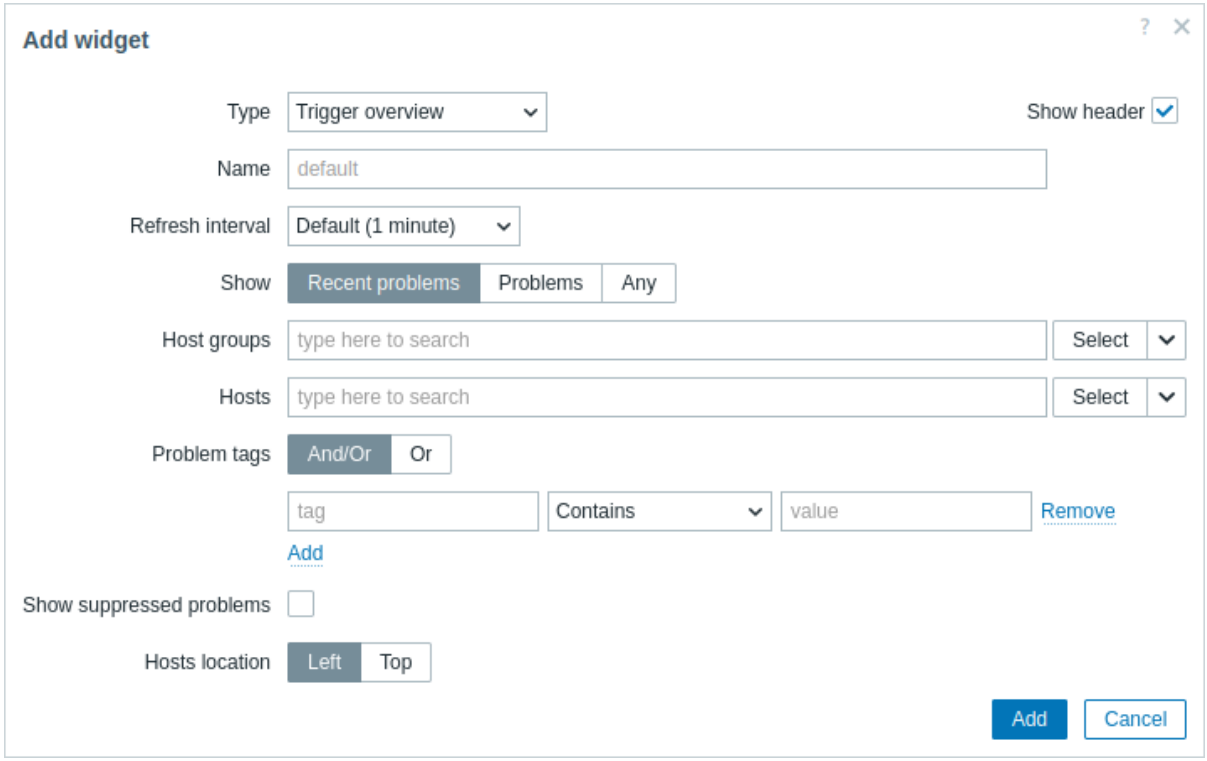
Note:
As cores de severidade do trigger podem ser ajustadas em *Administration* > *General*. As severidades do trigger podem ser modificadas ao atualizar problemas.

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 *Administration* > *General* > *GUI* em *Max number of columns and rows in overview tables*). Se houver mais triggers, uma mensagem solicitará 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.

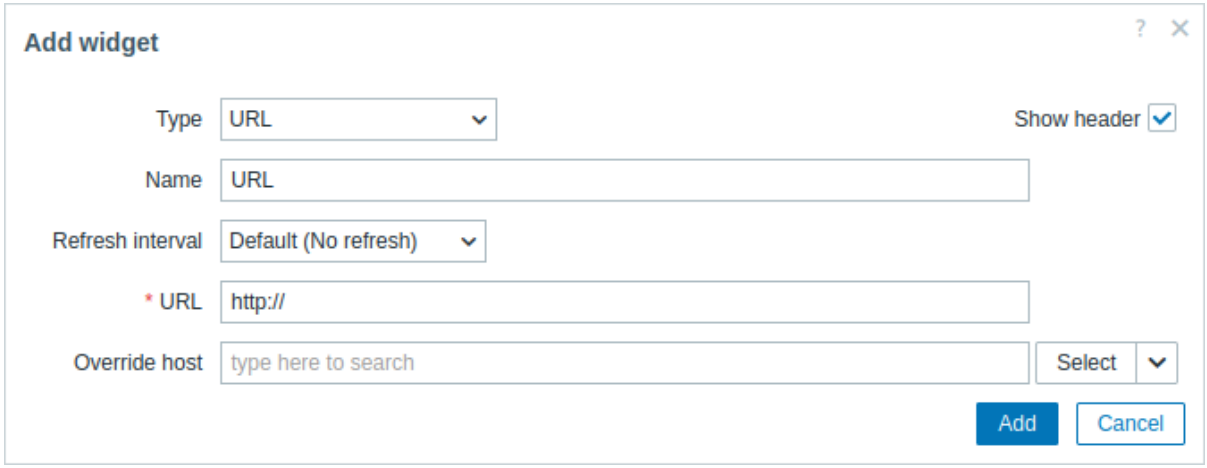
URL

Visão geral

O widget *URL* exibe o conteúdo recuperado de uma URL.

Configuração

Para configurar, selecione *URL* como tipo:



Além dos parâmetros comuns a todos os widgets, você pode definir as seguintes opções específicas:

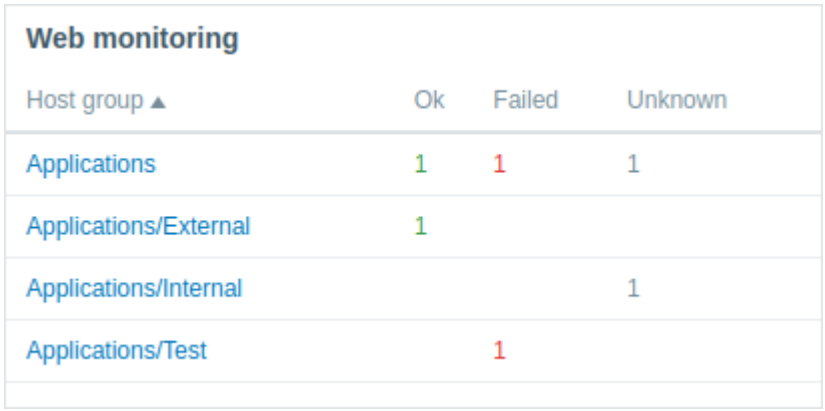
URL	Digite a URL a ser exibida (até 2048 caracteres). URLs externas devem começar com `http://` ou `https://`. URLs internas suportam caminhos relativos (por exemplo, `zabbix.php?action=report.status`). Macros `{HOST.*}` são suportadas.
Override host	Selecione um widget compatível ou o seletor de host do dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um template dashboard.

Attention:
Os navegadores podem não carregar uma página HTTP configurada no widget se o frontend do Zabbix for acessado via HTTPS. Além disso, talvez seja necessário modificar o cabeçalho HTTP de resposta X-Frame-Options para exibir conteúdo externo ao frontend do Zabbix.

Monitoramento web

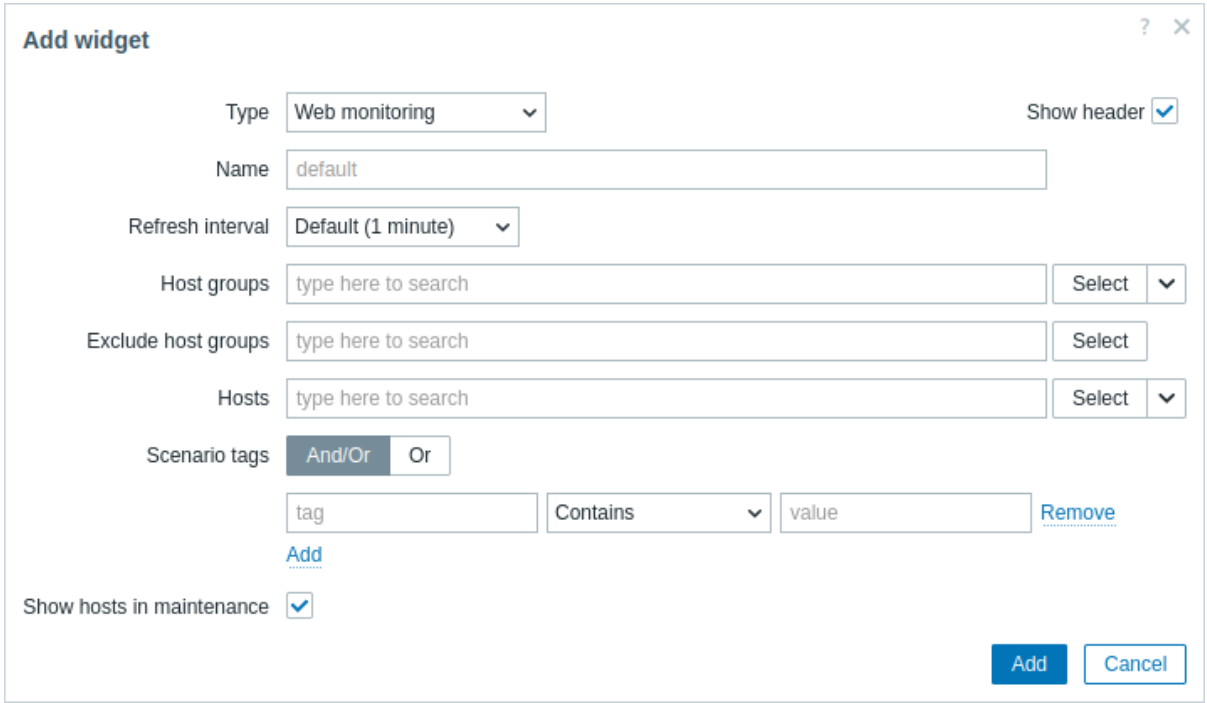
Visão geral

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.

Monitoramento

Visão geral

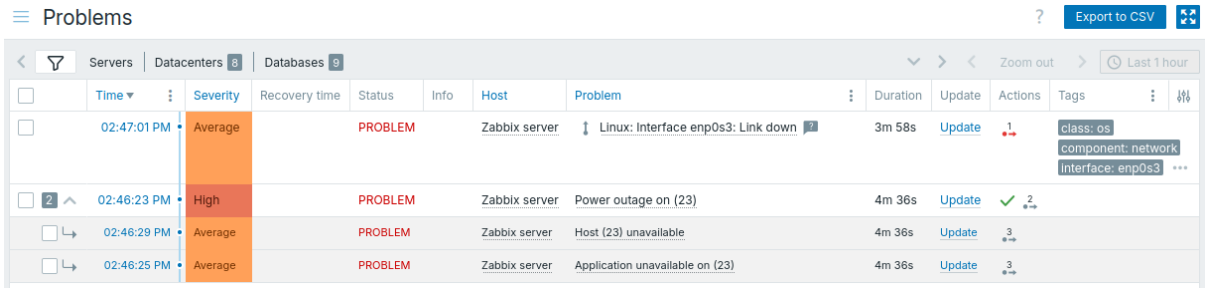
O menu Monitoring é dedicado à exibição de dados.
Todas as informações que o Zabbix estiver configurado para coletar, visualizar e processar serão exibidas nas várias seções do menu *Monitoring*.

Problemas

Visão geral

Em *Monitoring* > *Problems* você pode ver quais problemas estão ativos no momento. Problems são os triggers que estão no estado "Problem".

Por padrão, todos os novos problems são classificados como cause problems. É possível reclassificar manualmente determinados problems como symptom problems do cause problem. Para mais detalhes, consulte cause and symptom events.



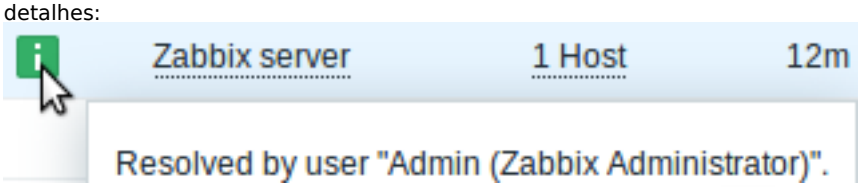
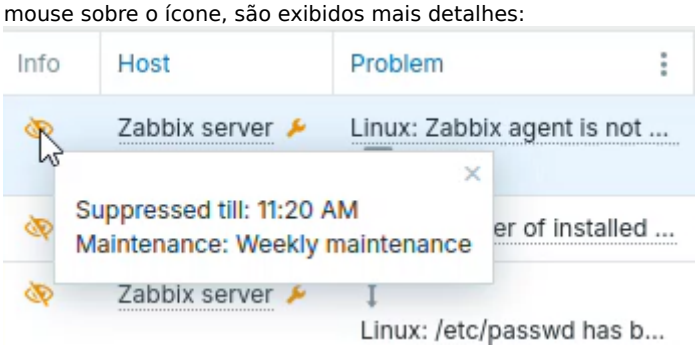
Os detalhes do problem são exibidos em uma tabela personalizável. O botão *Customize table* no cabeçalho da tabela permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Estas opções também estão disponíveis:

- *Compact view* - reduz a altura das linhas para exibir mais problems na tela.
- *Highlight full row* - usa a cor de severidade do problem para destacar problems não resolvidos (não disponível em temas de alto contraste).

Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefinir. Todas as personalizações são aplicadas imediatamente e sempre ficam salvas para o usuário atual dentro da aba ativa de filter (padrão ou salva).

As seguintes colunas estão disponíveis:

Column	Description
Checkbox	Caixas de seleção para escolher problems são exibidas. Os ícones ao lado das caixas de seleção têm o seguinte significado: 2 - o número de eventos de symptom para o cause problem; - expandir para mostrar eventos de symptom; - recolher para ocultar eventos de symptom; - este é um evento de symptom.
Time	O horário de início do problem é exibido. Clicar no botão *Options* no cabeçalho da coluna dá acesso à opção *Show timeline*, que alterna a linha do tempo visual e o agrupamento de problems (não disponível quando *Compact view* está habilitado).
Severity	A severidade do problem é exibida. A severidade do problem é originalmente baseada na severidade do trigger do problem subjacente; no entanto, após o evento ocorrer, ela pode ser atualizada usando a tela *Update problem* screen. A cor da severidade do problem é usada como plano de fundo da célula durante o período do problem.
Recovery time	O horário de resolução do problem é exibido Esta coluna não está disponível quando *Show* no filter está definido como *Problems*.
Status	O status do problem é exibido: **Problem** - problem não resolvido **Resolved** - problem resolvido recentemente. Você pode ocultar problems resolvidos recentemente usando o filter. Problems novos e resolvidos recentemente piscam por 2 minutos. Problems resolvidos são exibidos por 5 minutos no total. Ambos os valores são configuráveis em *Administration* > *General* > *Trigger displaying options*. Esta coluna não está disponível quando *Show* no filter está definido como *Problems*.

Column	Description
Info	Um ícone verde de informação é exibido se um problem for encerrado por correlação global ou manualmente ao atualizar o problem. Ao passar o mouse sobre o ícone, são exibidos mais detalhes:  Se um problem suprimido for exibido, o seguinte ícone é mostrado: . Quando o evento é suprimido por manutenção do host, o ícone é exibido em laranja: . Se manutenção e supressão manual se aplicarem, a supressão manual tem prioridade na cor do ícone. Ao passar o mouse sobre o ícone, são exibidos mais detalhes: 
Host	O host do problem é exibido. Clicar no nome do host abre o host menu.
Problem	O nome do problem é exibido. O nome do problem é baseado no nome do trigger do problem subjacente. Macros no nome do trigger são resolvidas no momento em que o problem ocorre e os valores resolvidos não são mais atualizados. Clicar no nome do problem abre o event menu. Ao passar o mouse sobre o ícone após o nome do problem, será exibida a descrição do trigger (para os problems que a possuem). Clicar no botão *Options* no cabeçalho da coluna dá acesso a duas opções: - *Show operational data* exibe os dados operacionais após o nome do problem, entre parênteses (somente se o campo *Operational data* não estiver vazio na trigger configuration). - *Show trigger expression* exibe a expressão do trigger do problem abaixo do nome do problem (não disponível quando *Compact view* está habilitado).
Operational data	São exibidos dados dinâmicos e em tempo real do problem. O conteúdo depende do campo *Operational data* na trigger configuration: se uma string com macros estiver definida, ela será mostrada aqui; se o campo estiver vazio, serão mostrados os valores mais recentes de todos os items na expressão do trigger.
Duration	A duração do problem é exibida. Veja também: Negative problem duration
Update	Clique no link *Update* para ir para a tela de problem update, onde várias ações podem ser executadas no problem, incluindo comentar e reconhecer o problem.

Column	Description
Actions	O histórico de atividades sobre o problem é exibido usando ícones simbólicos:  - o problem foi reconhecido. Este ícone é sempre exibido primeiro.  - foram feitos comentários. O número de comentários também é exibido.  - a severidade do problem aumentou (por exemplo, Information > Warning).  - a severidade do problem diminuiu (por exemplo, Warning > Information).  - a severidade do problem foi alterada, mas voltou ao nível original (por exemplo, Warning > Information > Warning).  - o problem foi suprimido.  - a supressão do problem foi removida.  - o problem foi suprimido devido à manutenção do host.  - a supressão do problem foi removida após a manutenção do host.  - ações foram executadas. O número de ações também é exibido.  - ações foram executadas, e pelo menos uma está em andamento. O número de ações também é exibido.  - ações foram executadas, e pelo menos uma falhou. O número de ações também é exibido. Ao passar o mouse sobre os ícones, são exibidos pop-ups com detalhes sobre a atividade. Consulte viewing details para saber mais sobre os ícones usados no pop-up para ações executadas.
Tags	tags do problem são exibidas. Além disso, tags de um sistema externo de tickets também podem ser exibidas (veja a opção *Process tags* ao configurar webhooks). Ao passar o mouse ou clicar no ícone de três pontos ••• após as tags exibidas, abre-se um pop-up com todas as tags. Clicar no botão ⋮ *Options* no cabeçalho da coluna permite definir o número de tags exibidas (1-3), escolher como os nomes das tags são exibidos (completos, abreviados ou nenhum), definir a prioridade de exibição das tags ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).
Tag value	Os valores das tag do problem são exibidos. Clicar no botão ⋮ *Options* no cabeçalho da coluna permite renomear a coluna, escolher quais valores de tag exibir ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).
Custom text	É exibido texto personalizado, valores de macro ou uma combinação de ambos. Clicar no botão ⋮ *Options* no cabeçalho da coluna permite renomear a coluna, inserir o texto personalizado ou macros, ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas). Supported macros: `{HOST.*}` (exceto `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (exceto `{ITEM.STATE.ERROR}`), `{TRIGGER.ID}`, `{EVENT.ID}` e user macros.

Duração negativa do problema

Na verdade, é possível em algumas situações comuns ter uma duração negativa do problema, ou seja, quando o tempo de resolução do problema é anterior ao tempo de criação do problema, por exemplo:

- Se algum host for monitorado por proxy e ocorrer um erro de rede, levando a nenhum dado recebido do proxy por um tempo, o trigger nodata(/host/key) será acionado pelo server. Quando a conexão for restaurada, o server receberá dados do item do proxy com um tempo do passado. Então, o problema nodata(/host/key) será resolvido e terá uma duração negativa do problema;
- Quando os dados do item que resolvem o evento do problema são enviados pelo Zabbix sender e contêm um timestamp anterior ao tempo de criação do problema, uma duração negativa do problema também será exibida.

Note:
A duração negativa do problema não afeta o cálculo de SLA ou o relatório de disponibilidade de um trigger específico de forma alguma; ela não reduz nem expande o tempo do problema.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Atualização em massa* - atualiza os problemas selecionados navegando para a tela de atualização de problema

Para usar esta opção, marque as caixas de seleção antes dos respectivos problemas e, em seguida, clique no botão *Atualização em massa*.

Botões

O botão à direita oferece a seguinte opção:

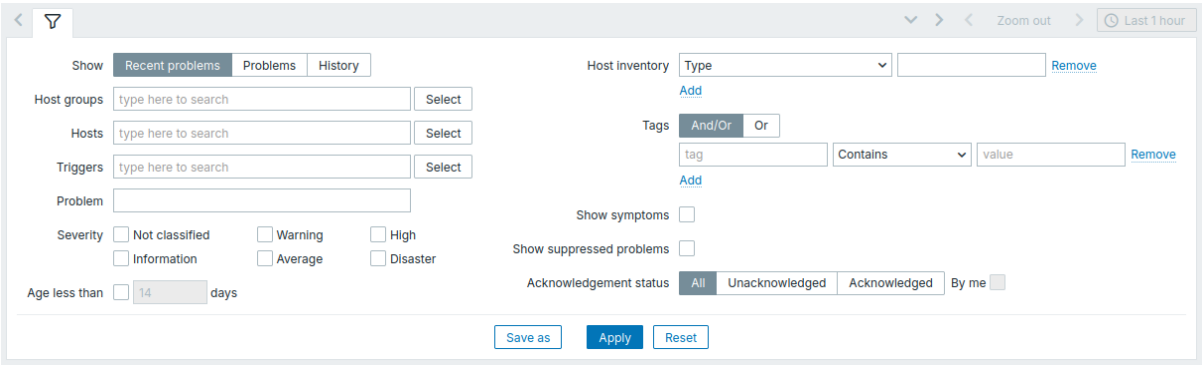
Export to CSV	Exportar o conteúdo de todas as páginas para um arquivo CSV.

Os botões do modo de visualização, por serem comuns a todas as seções, são descritos na página Monitoring.

Usando o filtro

Você pode usar o filtro para exibir apenas os problemas de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro fica localizado acima da tabela. As configurações favoritas do filtro podem ser salvas como abas e, em seguida, acessadas rapidamente clicando em as abas acima do filtro.



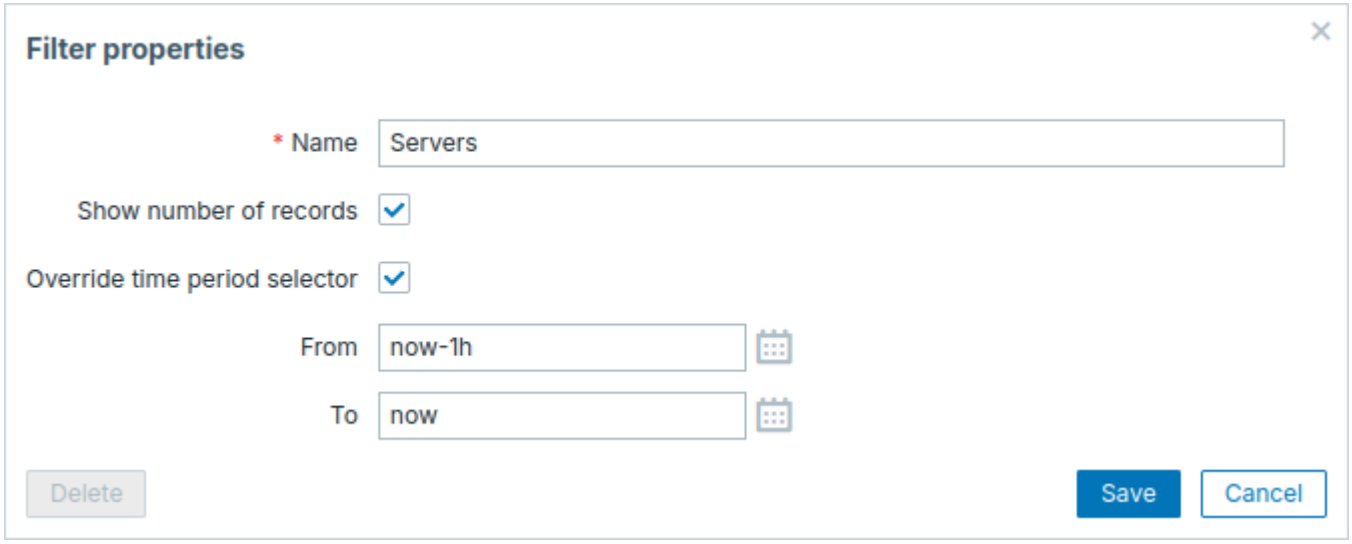
Parameter	Description
Show	Filtrar por status do problema: **Recent problems** - problemas não resolvidos e recentemente resolvidos são exibidos (padrão) **Problems** - problemas não resolvidos são exibidos **History** - o histórico de todos os eventos é exibido
Host groups	Filtrar por um ou mais grupos de host. Especificar um grupo de host pai seleciona implicitamente todos os grupos de host aninhados.
Hosts	Filtrar por um ou mais hosts.
Triggers	Filtrar por um ou mais triggers.
Problem	Filtrar pelo nome do problema.
Severity	Filtrar pela severidade do trigger (problema).
Age less than	Filtrar pela idade do problema.
Host inventory	Filtrar por tipo e valor do inventário.

Parameter	Description
Tags	Filtrar por nome e valor de event tag. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Contains** - incluir os nomes de tag especificados em que os valores da tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Does not contain** - excluir os nomes de tag especificados em que os valores da tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - 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 *Tag display priority* (veja abaixo).
Show symptoms	Marque a caixa de seleção para exibir em sua própria linha os problemas classificados como sintomas.
Show suppressed problems	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos e ocultados devido à manutenção do host ou suprimidos manualmente por um usuário.
Acknowledgement status	Filtrar para exibir todos os problemas, apenas problemas não reconhecidos ou apenas problemas reconhecidos. Marque a caixa de seleção adicional para filtrar os problemas que já foram reconhecidos por você.

Abas para filtros favoritos

Conjuntos de parâmetros de filtro usados com frequência podem ser salvos em abas.

Para salvar um novo conjunto de parâmetros de filtro, abra a aba principal, configure as definições do filtro e clique no botão *Save as*. Em uma nova janela pop-up, defina as *Filter properties*.

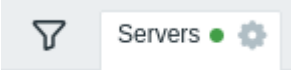


Parameter	Description
Name	O nome do filtro a ser exibido na lista de abas.
Show number of records	Marque se quiser que o número de problemas seja exibido ao lado do nome da aba.
Override time period selector	Marque para definir um período padrão específico para este conjunto de filtros. Se definido, você só poderá alterar o período para esta aba atualizando as definições do filtro. Para abas sem um período personalizado, o intervalo de tempo pode ser alterado clicando no botão seletor de tempo no canto superior direito (o nome do botão depende do intervalo de tempo selecionado: This week, Last 30 minutes, Yesterday, etc.). Esta opção está disponível apenas para filtros em *Monitoring* > *Problems*.

Parameter	Description
From/To	Início e fim do período de tempo em sintaxe de tempo absoluto (`Y-m-d H:i:s`) ou relativa (`now-1d`). Disponível se *Set custom time period* estiver marcado.

Quando salvo, o filtro é criado como uma aba de filtro nomeada e ativado imediatamente.

Para editar as propriedades de filtro de um filtro existente, clique no símbolo de engrenagem ao lado do nome da aba ativa.



Notas:

- Para ocultar a área de filtro, clique no nome da aba atual. Clique novamente no nome da aba ativa para abrir a área de filtro outra vez.
- 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. Como alternativa, o botão apontando para baixo abre um menu suspenso com todos os filtros salvos e você pode clicar no que precisar.
- Se as definições de um filtro salvo tiverem sido alteradas (mas não salvas), um ponto verde é exibido após o nome do filtro. Para atualizar o filtro de acordo com as novas definições, clique no botão *Update*, que é exibido em vez do botão *Save as*.
- O layout das colunas da tabela é salvo por aba de filtro. Para personalizar o layout de um filtro salvo, primeiro abra essa aba de filtro e depois personalize as colunas.
- As abas de filtro, as definições do filtro e o layout das colunas da tabela de cada aba são lembrados no perfil do usuário e restaurados quando o usuário abre a página novamente.

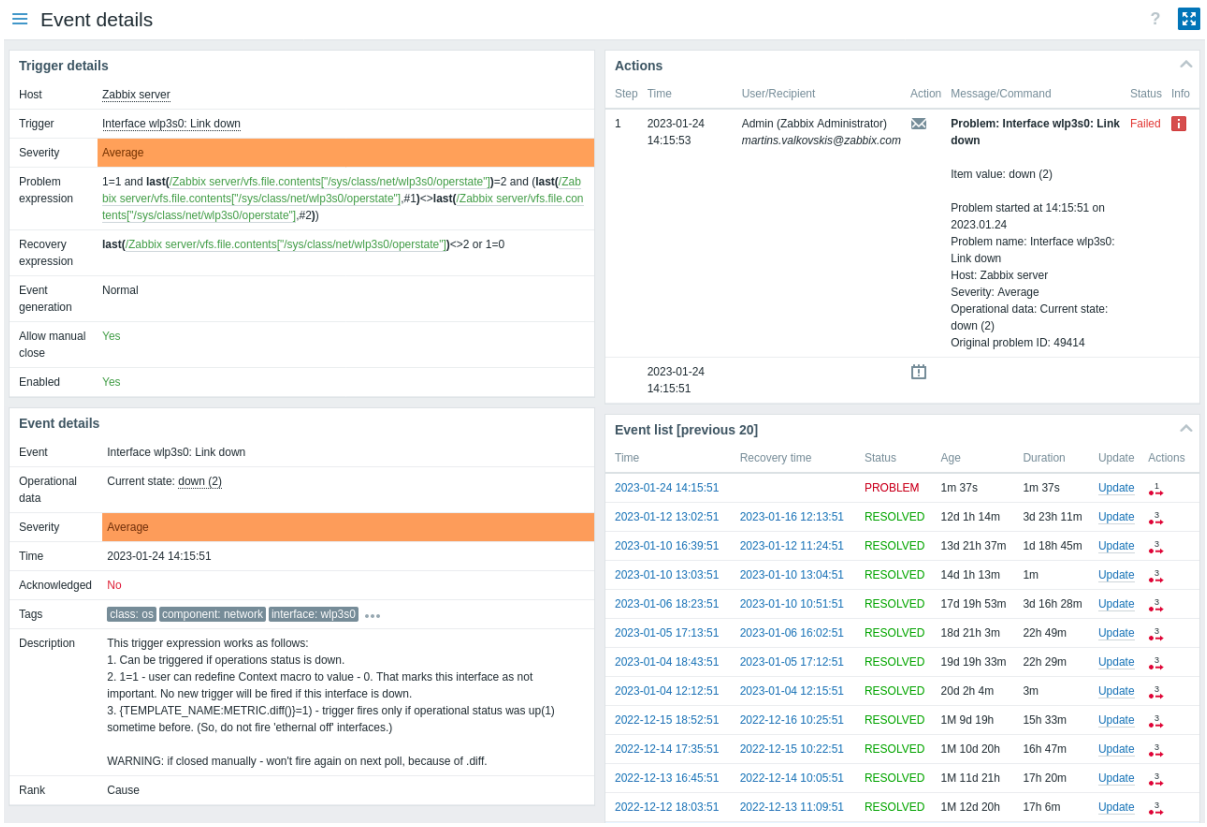
Note:
Para compartilhar filtros, copie e envie a outras pessoas uma URL de um filtro ativo. Após abrir essa URL, outros usuários poderão salvar esse conjunto de parâmetros como um filtro permanente em sua conta Zabbix.
 Veja também: Page parameters.

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 *Monitoring* > *Problems* são links. Ao clicar neles, são exibidos mais detalhes do evento.



Nos detalhes do evento, você pode ver se o evento foi suprimido por manutenção ou suprimido manualmente por um usuário.

Observe que a severidade 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 *Update problem*.

Na lista de ações, os seguintes ícones são usados para indicar o tipo de atividade:

-  - evento de problema gerado.
-  - mensagem foi enviada.
-  - evento de problema reconhecido.
-  - evento de problema não reconhecido.
-  - um comentário foi adicionado.
-  - a severidade do problema foi aumentada (por exemplo, Information > Warning).
-  - a severidade do problema foi reduzida (por exemplo, Warning > Information).
-  - a severidade do problema foi alterada, mas retornou ao nível original (por exemplo, Warning > Information > Warning).
-  - um comando remoto foi executado.
-  - o evento de problema foi recuperado.
-  - o problema foi fechado manualmente.
-  - o problema foi suprimido.
-  - a supressão do problema foi removida.
-  - o problema foi suprimido devido à manutenção do host.
-  - a supressão do problema foi removida após a manutenção do host.
-  - o problema foi convertido em um problema sintoma.
-  - o problema foi convertido em um problema causa.

Causa e problemas de 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.

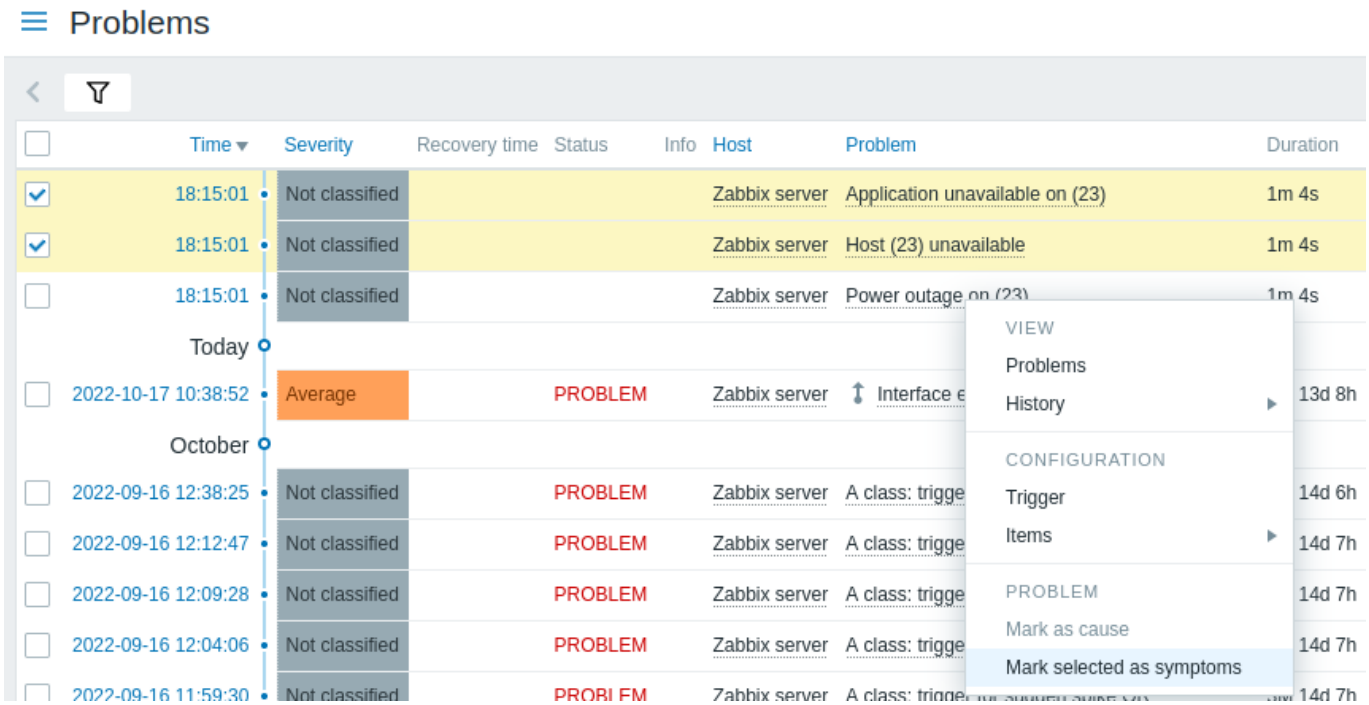
Apenas problemas de causa são contabilizados nos totais de problemas em mapas, widgets do dashboard como *Problemas por severidade* ou *Hosts com problemas*, etc. No entanto, a classificação de problemas não afeta os serviços.

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 problema sintoma, primeiro selecione-o na lista de problemas. Um ou vários problemas podem ser selecionados.

Em seguida, vá até o problema causa e, no menu de contexto dele, clique na opção *Marcar selecionados como sintomas*.



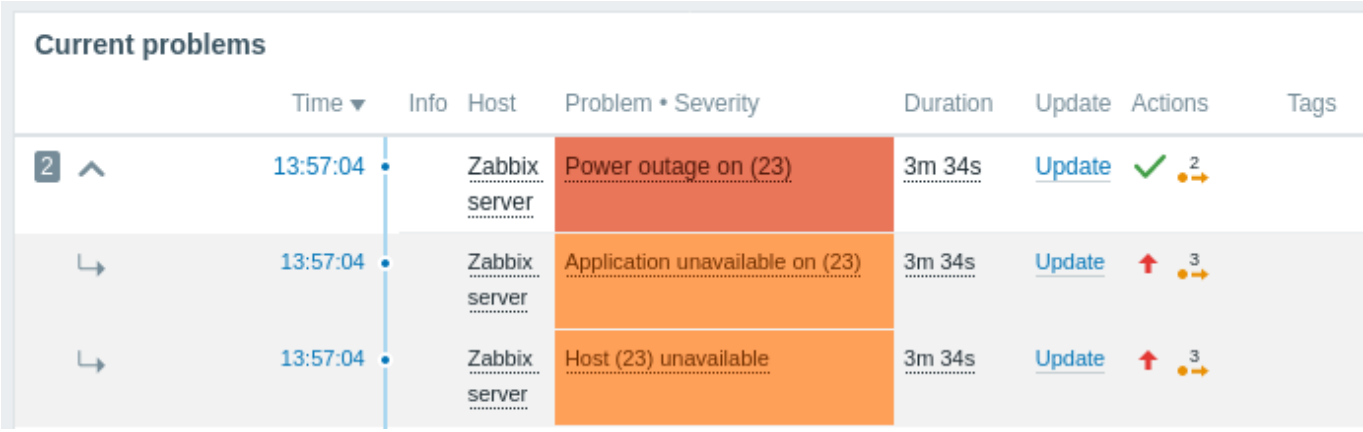
Depois disso, os problemas selecionados serão atualizados pelo server para problemas sintoma do problema causa.

Enquanto o status do problema estiver sendo atualizado, ele é exibido de uma das duas formas:

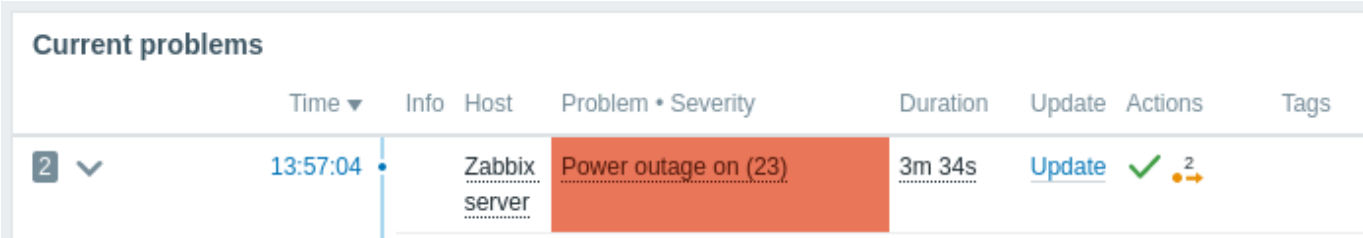
- Um status piscante "UPDATING" é exibido na coluna Status;
- Um ícone piscante ou na coluna Info (isso se aplica se apenas *Problems* estiverem selecionados no filtro e, portanto, a coluna Status não for exibida).

Exibição

Os problemas de sintoma são exibidos abaixo do problema de causa e marcados de forma correspondente em *Monitoring* > *Problems* (e no widget do painel *Problems*) - com um ícone, fonte menor e plano de fundo diferente.



Na visualização recolhida, apenas o problema de causa é exibido; 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 *Show symptoms* nas configurações do filtro ou na configuração do widget.

Revertendo para problema de causa

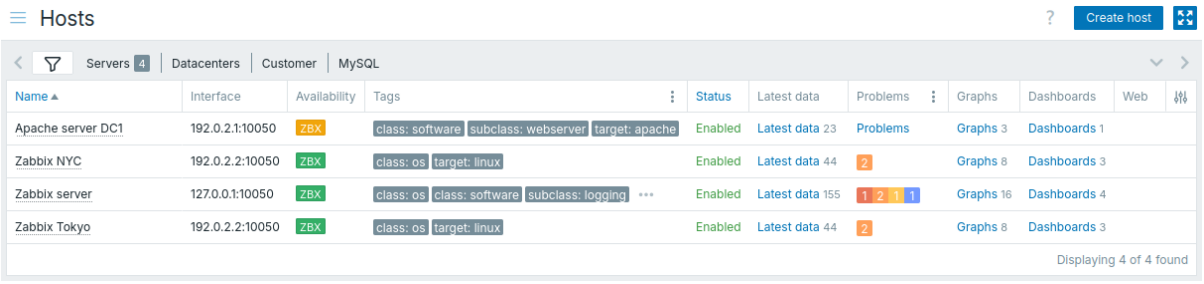
Um problema de sintoma pode ser revertido para um problema de causa. Para fazer isso, escolha uma das opções:

- clique na opção *Marcar como causa* no menu de contexto do problema de sintoma;
- marque a opção *Converter em causa* na tela de atualização do problema e clique em *Atualizar* (essa opção também funcionará se vários problemas forem selecionados).

Hosts

Visão geral

A seção *Monitoring* > *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 dados mais recentes do host, histórico de problemas, gráficos, dashboards e cenários web.



Os detalhes do host são exibidos em uma tabela personalizável. O botão *Customize table* no cabeçalho da tabela permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Você também pode redimensionar cada coluna arrastando sua borda direita; dê um clique duplo na borda para redefinir. Todas as personalizações são aplicadas imediatamente e sempre ficam salvas para o usuário atual dentro da aba ativa de filtro (padrão ou salva).

As seguintes colunas estão disponíveis:

Column	Description
Name	O nome visível do host. Clicar no nome abre 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.
Availability	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, a disponibilidade será atualizada depois que o Zabbix server sincronizar as alterações de configuração. Da mesma forma, se você habilitar um host, a disponibilidade será atualizada depois que o Zabbix server sincronizar as alterações de configuração e consultar o host. Os ícones de disponibilidade representam apenas os tipos de interface configurados (Agent, SNMP, IPMI, JMX). 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 do host é determinado pela conexão entre um item habilitado que usa a interface e o host. O status pode ser: **Available** - a conexão com o host foi bem-sucedida; **Not available** - a conexão com o host falhou (timeout, problemas de firewall etc.); **Unknown** - 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 Unknown interface status e Unreachable/unavailable host interface settings. O status de todas as interfaces de um 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: **Green** - todas as interfaces estão disponíveis; **Yellow** - pelo menos uma interface não está disponível, e pelo menos uma está disponível ou desconhecida; **Red** - todas as interfaces não estão disponíveis; **Gray** - 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 de 60 segundos, intervalo de 0-3600). As verificações ativas são consideradas indisponíveis quando o heartbeat da verificação ativa é mais antigo que 2 x HeartbeatFrequency segundos. **Nota:** agents do Zabbix anteriores à versão 6.2.x não enviam heartbeats de verificação ativa, portanto a disponibilidade de seus hosts permanece desconhecida.
Tags	As tags do host são exibidas. Ao passar o mouse ou clicar no ícone de três pontos após as tags exibidas, é aberto um pop-up com todas as tags. Clicar no botão *Options* no cabeçalho da coluna permite definir o número de tags exibidas (1–3), escolher como os nomes das tags são exibidos (completo, abreviado ou nenhum), definir a prioridade de exibição das tags ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).
Tag value	Os valores das tags do host são exibidos. Clicar no botão *Options* no cabeçalho da coluna permite renomear a coluna, escolher quais valores de tag exibir ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).

Column	Description
Status	Status do host - *Enabled* ou *Disabled*. Clique no cabeçalho da coluna para classificar os hosts por status em ordem crescente ou decrescente.
Latest data	Clicar no link abrirá a seção *Monitoring* > *Latest data* com todos os dados mais recentes coletados do host. O número de items com dados mais recentes é exibido em cinza.
Problems	O número de problemas abertos de um host é exibido, classificado por severidade. A cor de cada ícone de problema mostra a severidade, enquanto o número mostra a quantidade de problemas para aquela severidade. Clicar em um ícone abre *Monitoring* > *Problems* para esse host; se um host não tiver problemas, um link *Problems* é exibido no lugar. Clicar no botão ⋮ *Options* no cabeçalho da coluna dá acesso à opção *Show suppressed problems*, que inclui problemas que, de outra forma, seriam suprimidos (ocultos) devido à manutenção do host ou à supressão individual de problem.
Graphs	Clicar no link exibirá os gráficos configurados para o host. O número de gráficos é exibido em cinza. Se um host não tiver gráficos, nenhum link será exibido.
Dashboards	Clicar no link exibirá os dashboards configurados para o host. O número de dashboards é exibido em cinza. Se um host não tiver dashboards, nenhum link será 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, nenhum link será exibido.
Custom text	Texto personalizado, valores de macro ou uma combinação de ambos é exibido. Clicar no botão ⋮ *Options* no cabeçalho da coluna permite renomear a coluna, inserir o texto personalizado ou macros, ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas). Macros suportadas: `{HOST.*}` (exceto `{HOST.METADATA}`), `{INVENTORY.*}` e user macros.

Botões

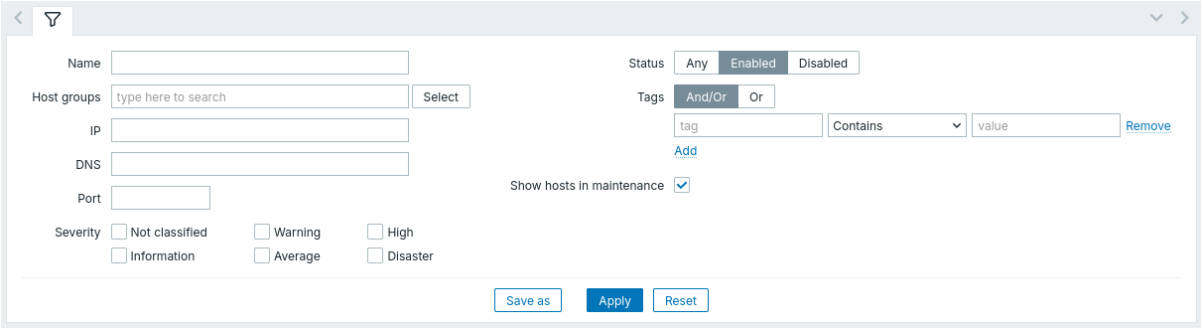
Criar host permite criar um novo host. Este botão está disponível somente para usuários Admin e Super Admin.

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 de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro fica localizado acima da tabela. É possível filtrar hosts por nome, grupo de host, IP ou DNS, porta da interface, tags, severidade do problema, status (enabled/disabled/any); você também pode selecionar se deseja exibir hosts que estão atualmente em manutenção.



Parameter	Description
Name	Filtrar pelo nome visível do host.
Host groups	Filtrar por um ou mais grupos de host. Especificar um grupo de host pai seleciona implicitamente todos os grupos de host aninhados.
IP	Filtrar por endereço IP.
DNS	Filtrar por nome DNS.
Port	Filtrar por número da porta.

Parameter	Description
Severity	Filtrar por severidade do problema. Por padrão, problemas de todas as severidades são exibidos. Os problemas são exibidos se não estiverem suprimidos.
Status	Filtrar por status do host.
Tags	Filtrar por 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 e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida
Show hosts in maintenance	Marque a caixa de seleção para exibir hosts que estão em manutenção (exibido por padrão).

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.

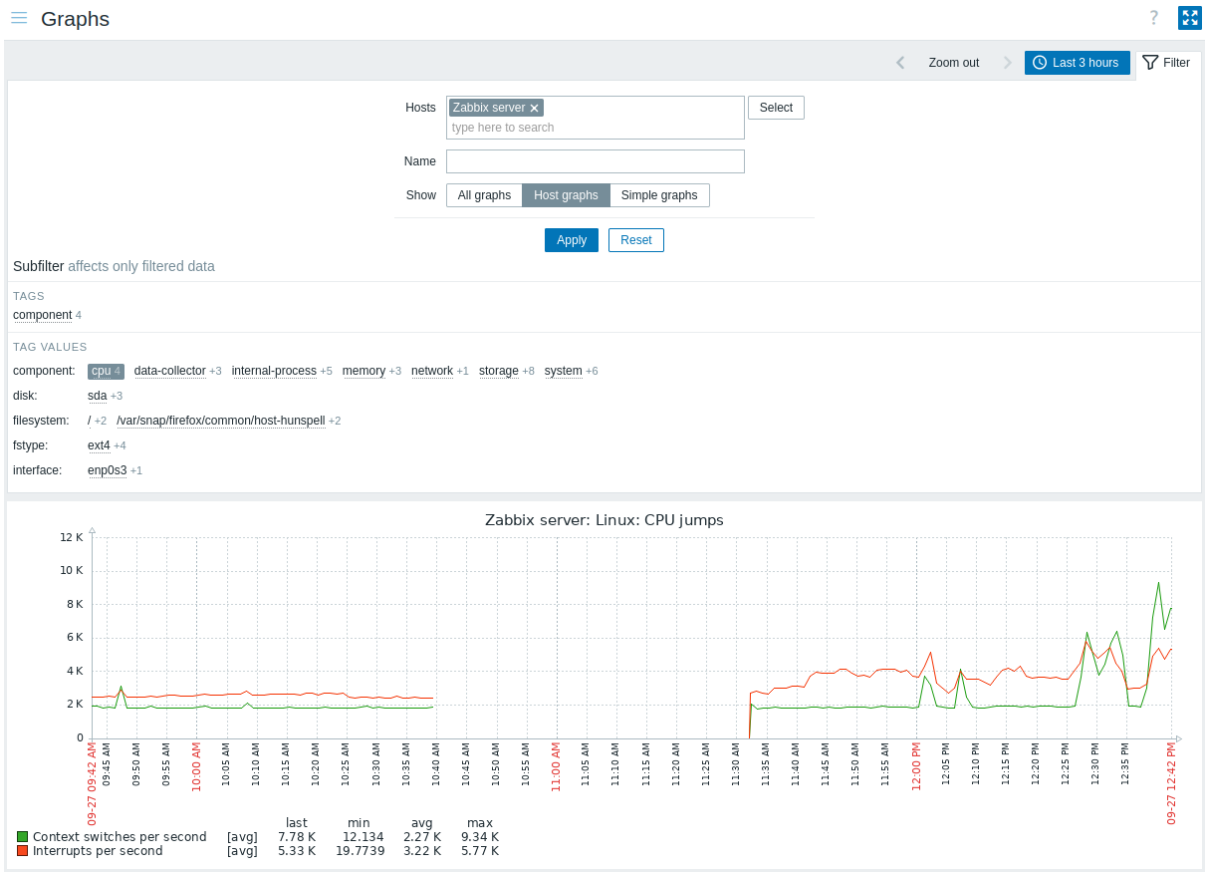
Veja mais detalhes sobre salvando filtros.

Gráficos

Visão geral

Os gráficos do host podem ser acessados em *Monitoring* > *Hosts* clicando em Graphs para o respectivo host.

Qualquer custom graph que tenha sido configurado para o host pode ser exibido, assim como qualquer simple graph.



Os gráficos são classificados por:

- nome do gráfico (custom graphs)
- nome do item (simple graphs)

Os gráficos de hosts desabilitados também estão acessíveis.

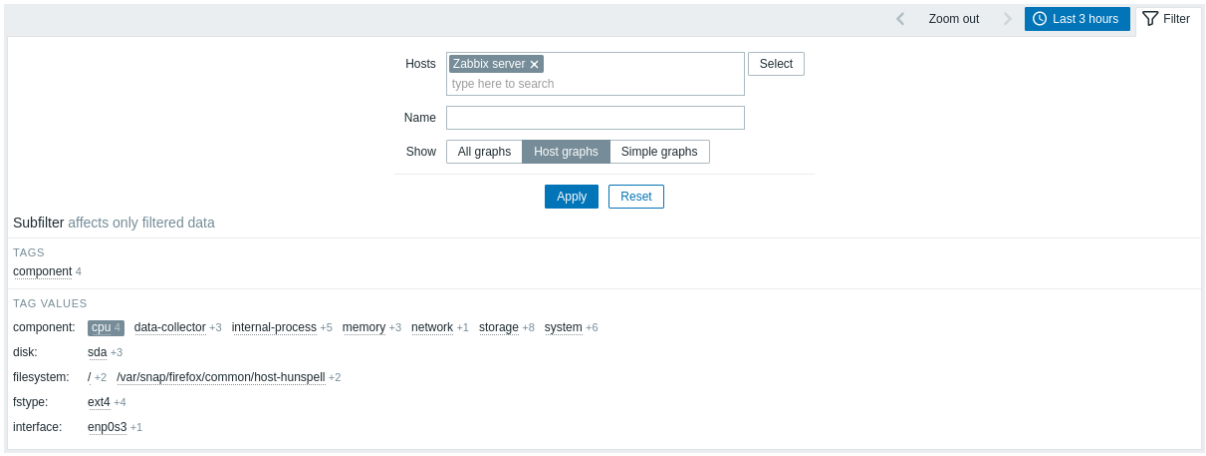
Seletor de período de tempo

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 o subfiltro

O subfiltro permite modificar ainda mais a filtragem aplicada pelo filtro principal.

Ele contém links clicáveis para acesso rápido aos 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 a filtragem.

Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se quiser ver mais registros no subfiltro, aumente o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração* > *Geral* > *Interface gráfica*).

Diferentemente 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 contagem.

O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de três pontos será exibido no final; ele não é clicável. As listas verticais (como tags com seus valores) são limitadas a 20 entradas. Se houver mais, um ícone de três pontos será 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.

Depois que uma entidade é selecionada, os números das 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, são descritos na página Monitoring.

Painéis do host

Visão geral

Os dashboards de host são semelhantes aos dashboards globais; no entanto, os dashboards de host não têm um proprietário e exibem dados somente para o host selecionado.



Ao visualizar os dashboards de host, você pode alternar entre os dashboards configurados clicando em:

- nas abas dos dashboards;
- 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 *Monitoring* > *Hosts*, clique no link de navegação *All 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.

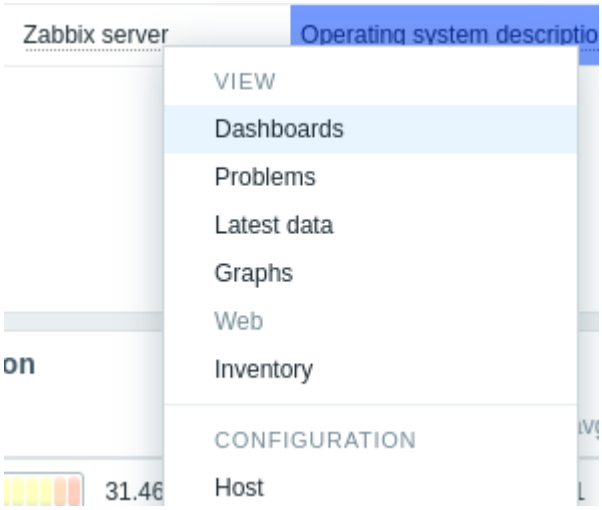
Os widgets dos dashboards de host também só podem ser configurados no nível de template, exceto para alteração do intervalo de atualização. Além disso, os widgets dos dashboards de host só podem ser copiados para outros dashboards de host dentro do mesmo template. Observe que widgets de dashboards globais *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 do host podem ser acessados:

- após pesquisar por um nome de host em global search (clique no link *Dashboards* fornecido nos resultados da pesquisa);
- após clicar em um nome de host em *Inventory* > *Hosts* (clique no link *Dashboards* fornecido na visão geral do host);
- no menu do host, clicando em *Dashboards*.

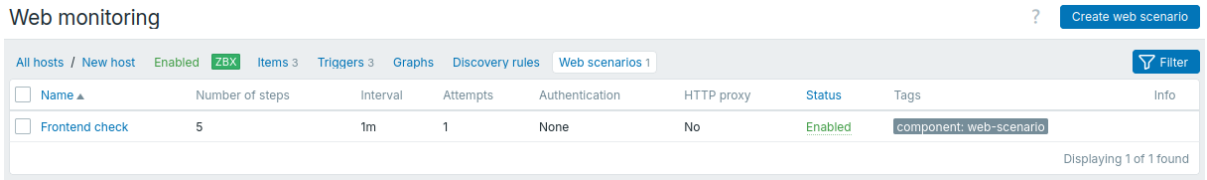


Observe que os dashboards do host *não podem* ser acessados diretamente na seção *Dashboards*, que é reservada para dashboards globais.

Cenários web

Visão geral

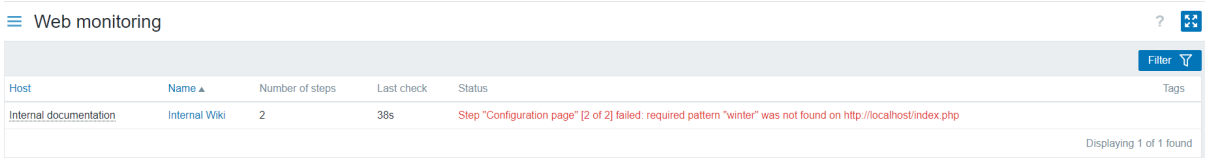
As informações do web scenario do host podem ser acessadas em *Monitoring* > *Hosts* clicando em Web para o host correspondente.



Clicar no nome do host abre o menu do host. Os dados de hosts desabilitados também estão acessíveis. O nome de um host desabilitado é exibido em vermelho.

O status de um web scenario pode ser:

- OK (verde) - todas as etapas foram concluídas com sucesso e o Zabbix coletou dados para o cenário.
- Erro (vermelho) - uma ou mais etapas falharam (consulte a mensagem de erro para obter dicas sobre quais parâmetros podem precisar de correção).
- Desconhecido (cinza) - ainda não há status de etapa disponível ou nenhum dado foi coletado.



O número máximo de cenários exibidos por página depende da configuração de perfil de usuário *Rows per page* [/manual/web_interface/user_profile#user-profile).

Por padrão, apenas valores que se enquadram nas últimas 24 horas são exibidos. Esse limite foi introduzido com o objetivo de melhorar os tempos de carregamento inicial de páginas grandes com os dados mais recentes. Você pode estender esse período alterando o valor do parâmetro *Max history display period* no menu *Administration > General > 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 *Monitoring > Hosts*, selecione esse host ou grupo no filtro. Você também pode filtrar cenários com base em tags.

Botões

Os botões de modo de visualização, comuns a todas as seções, são descritos na página Monitoring.

Dados mais recentes

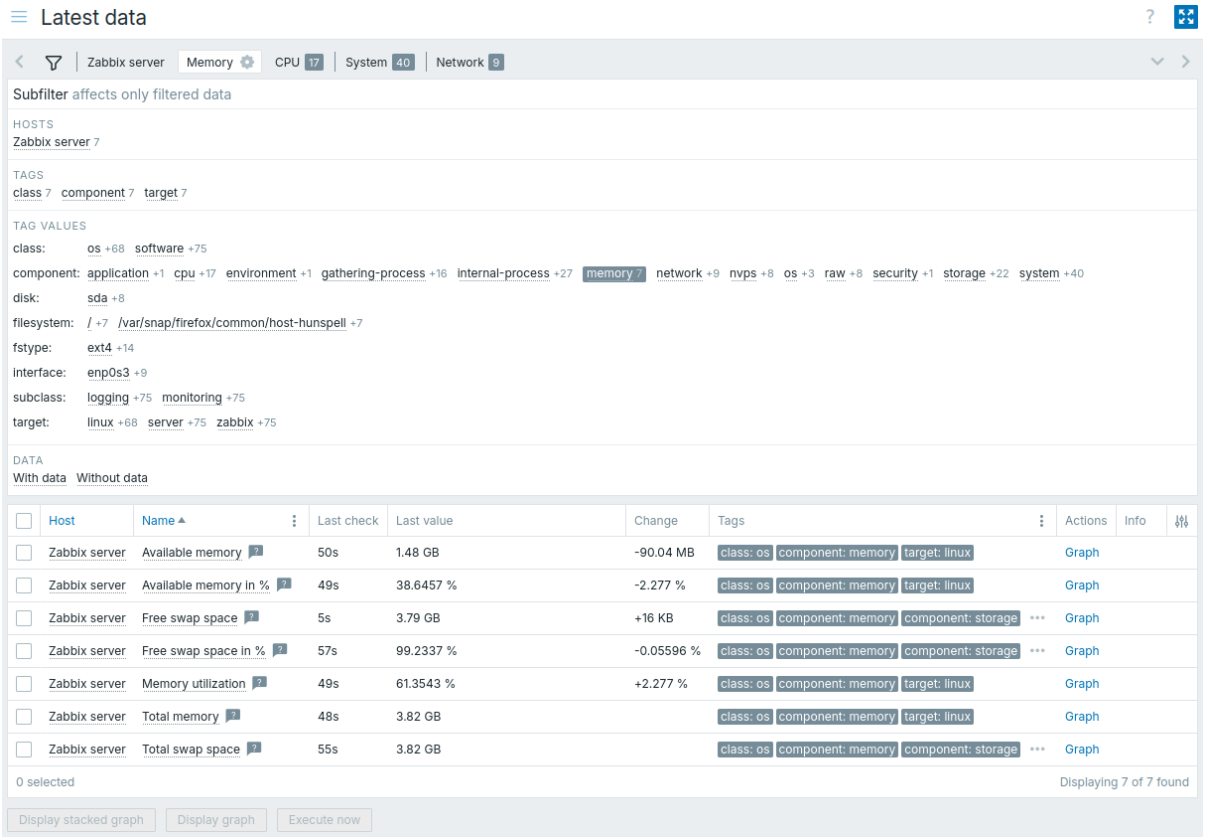
Visão geral

A seção *Monitoring > Latest data* exibe os valores mais recentes coletados por items.

Esta seção contém os seguintes elementos:

- Filter
- Subfilter
- Lista de items

Note:
O subfilter e a lista de items são exibidos somente se o filter estiver definido e houver resultados para exibir.



Os detalhes do valor mais recente são exibidos em uma tabela personalizável. O botão *Customize table* no cabeçalho da tabela permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefinir. Todas as personalizações são aplicadas imediatamente e sempre lembradas para o usuário atual dentro da aba ativa de filter (padrão ou salva).

As seguintes colunas estão disponíveis:

Column	Description
Host	Nome do host ao qual o item pertence. Clicar no nome abre o menu de contexto do host. Se um host estiver em manutenção, um ícone laranja de chave inglesa é exibido após o nome do host. Se um host estiver desativado, o nome do host é exibido em vermelho. Observe que os dados de hosts desativados (incluindo gráficos e listas de valores de items) estão acessíveis na seção *Latest data*.
Name	Nome do item. Clicar no nome abre o menu do item. Um ícone de ponto de interrogação é exibido ao lado do nome do item para todos os items que têm uma descrição. Passe o mouse sobre o ícone para exibir uma dica de ferramenta com a descrição do item. Clicar no botão *Options* no cabeçalho da coluna dá acesso à opção *Show item key*, que exibe a chave do item abaixo do nome de cada item.
Interval	Intervalo de atualização do item.
History	Período de armazenamento do histórico do item.
Trends	Período de armazenamento de tendências do item.
Type	Tipo do item.
Last check	Tempo desde a última verificação do item.

Column	Description
Last value	Valor mais recente do item. Os valores são exibidos com conversão de unidade e mapeamento de valor aplicados. Passe o mouse sobre o valor para exibir os dados brutos (truncados para 8192 caracteres). Valores binários são exibidos como um placeholder (*binary value*) em vez do valor real. Por padrão, apenas os valores recebidos nas últimas 24 horas são exibidos. Esse limite melhora os tempos de carregamento inicial para páginas grandes de latest data; para ampliá-lo, atualize o valor do parâmetro *Max history display period* em *Administration* > *General* > *GUI*.
Change	Diferença entre o valor anterior e o valor mais recente. Para items com frequência de atualização de 1 dia ou mais, o valor da mudança nunca será exibido (com a configuração padrão). Nesse caso, o último valor não será exibido de forma alguma se tiver sido recebido há mais de 24 horas.
Tags	Tags associadas ao item. Clicar em uma tag a habilita no subfilter, filtrando assim a lista de items por essa tag (e por quaisquer outras tags selecionadas no subfilter). Depois de clicadas, essas tags deixam de ser clicáveis; para ajustar a filtragem por tags, o subfilter deve ser usado. Clicar no botão *Options* no cabeçalho da coluna permite definir o número de tags exibidas (1-3), escolher como os nomes das tags são exibidos (completos, abreviados ou nenhum), definir a prioridade de exibição das tags ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).
Tag value	Valores da tag do item são exibidos. Clicar no botão *Options* no cabeçalho da coluna permite renomear a coluna, escolher quais valores de tag exibir ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas).
Actions	Link para simple graph/history dos valores do item.
Info	Informações adicionais sobre o item. Se um item tiver erros (por exemplo, tiver se tornado não suportado), um ícone de informação é exibido. Passe o mouse sobre o ícone para ver os detalhes.
Custom text	Texto personalizado, valores de macro ou uma combinação de ambos é exibida. Clicar no botão *Options* no cabeçalho da coluna permite renomear a coluna, inserir o texto personalizado ou macros, ou duplicar a coluna (colunas duplicadas podem ser renomeadas ou excluídas). Macros suportadas: `{HOST.*}` (exceto `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (exceto `{ITEM.STATE.ERROR}`) e user macros.

Botões

Os botões do modo de visualização, comuns a todas as seções, são descritos na página Monitoring.

Ações em massa

Os botões abaixo da lista oferecem ações em massa com um ou vários itens selecionados:

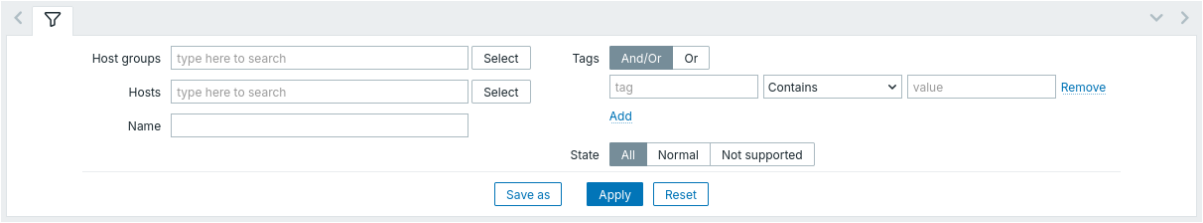
- *Exibir gráfico empilhado* - exibe um gráfico ad hoc empilhado.
- *Exibir gráfico* - exibe um gráfico ad hoc simples.
- *Executar agora* - executa imediatamente uma verificação para novos valores de item. Compatível apenas com verificações **passivas** (veja mais detalhes). Esta opção está disponível apenas para hosts com acesso de leitura e gravação. O acesso a esta opção para hosts com permissões somente leitura depende da opção de função de usuário chamada *Invocar "Executar agora" em hosts somente leitura*.

Para usar essas opções, marque as caixas de seleção antes dos respectivos itens e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os itens nos quais está interessado. Para melhorar o desempenho da pesquisa, os dados são pesquisados com as macros não resolvidas.

O ícone  do filtro está localizado acima da lista de itens e do subfiltro. Clique nele para expandir o filtro.



O filtro permite restringir a lista de itens por grupo de hosts, host, nome do item, tags e estado. Especificar um grupo de hosts pai no filtro seleciona implicitamente todos os grupos de hosts aninhados. Consulte *Monitoramento* > *Problemas* para obter detalhes sobre a filtragem por tags.

A opção de filtro *Mostrar detalhes* permite expandir as informações exibidas para os itens. Marque-a para exibir detalhes como o intervalo de atualização do item, as configurações de histórico e tendências, o tipo de item e os erros do item (normal/não suportado).

Salvando filtro

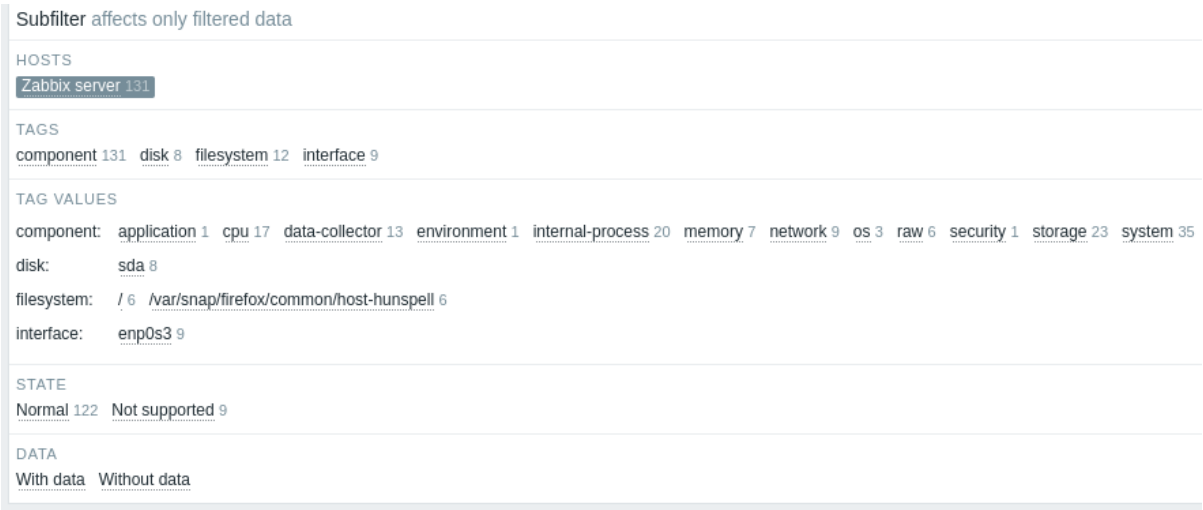
As configurações de filtro favorito podem ser salvas como abas e, em seguida, acessadas rapidamente clicando na respectiva aba acima do filtro.

Veja mais detalhes sobre salvando filtros.

Usando o subfiltro

O subfiltro permite modificar ainda mais a filtragem do filtro principal.

Ele contém links clicáveis para acesso rápido a itens relacionados. Os itens são relacionados por uma entidade em 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 a filtragem.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se houver 20 hosts, cada um com 100 itens (ou seja, 2000 registros no total), apenas metade dos hosts ficará visível no subfiltro. Para ver mais registros no subfiltro, aumente o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração* > *Geral* > *Interface gráfica*).

Diferentemente do filtro principal, o subfiltro é atualizado a cada solicitação de atualização da tabela, para sempre exibir informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

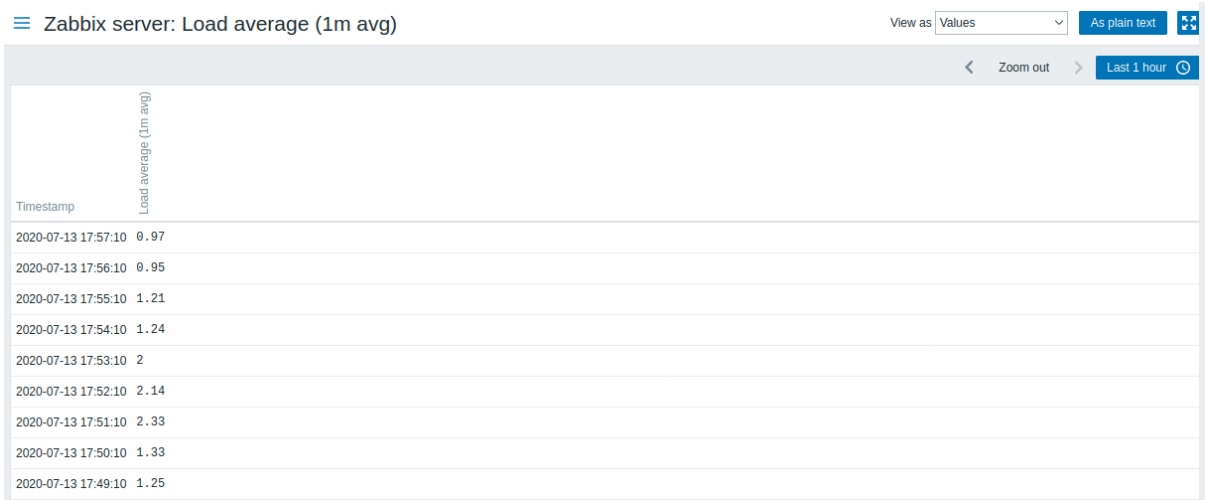
Para cada grupo de entidades (hosts, tags, valores de tags etc.), são exibidas até 10 linhas de entidades. Se houver mais entidades, essa lista poderá ser expandida para exibir no máximo 1000 entradas (o valor de SUBFILTER_VALUES_PER_GROUP em definições do frontend), clicando no ícone de três pontos ••• no final da lista. Para *Valores de tags*, a lista pode ser expandida para exibir no máximo 200 nomes de tags com seus valores correspondentes. Observe que, depois de totalmente expandida, a lista não poderá 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 *Graph/History* na lista de itens oferece os seguintes links:

- **History** - para todos os itens textuais, levando a listagens (*Values/500 latest values*) que exibem o histórico de valores anteriores do item.
- **Graph** - para todos os itens numéricos, levando a um gráfico simples. Observe que, quando o gráfico é exibido, um menu suspenso no canto superior direito oferece também a possibilidade de alternar para *Values/500 latest values*.



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 *Limit for search and filter results*, configurado em *Administration* > *General* > *GUI*.

Mapas

Visão geral

Na seção *Monitoring* > *Maps*, você pode configurar, gerenciar e visualizar network maps.

Ao abrir esta seção, você verá o último mapa que acessou 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 os quais o mapa foi compartilhado.

Listagem de mapas



Dados exibidos:

Column	Description
Name	Nome do mapa. Clique no nome para visualizar o mapa.
Width	A largura do mapa é exibida.
Height	A altura do mapa é exibida.
Actions	Duas ações estão disponíveis: **Properties** - define as propriedades gerais do mapa **Edit** - acessa a grade para adicionar elementos do mapa

Para configurar um novo mapa, clique no botão *Create map* no canto superior direito. Para importar um mapa de um arquivo YAML, XML ou JSON, clique no botão *Import* 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:

- *Export* - exporta os mapas para um arquivo YAML, XML ou JSON
- *Delete* - exclui os mapas

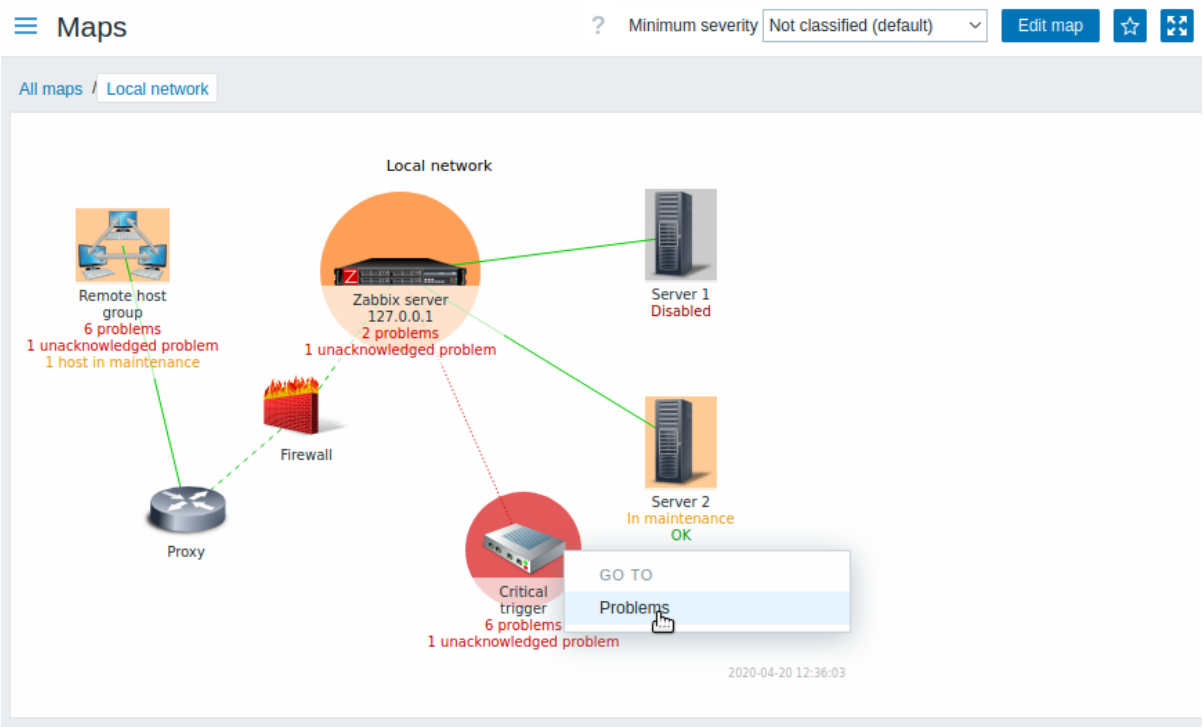
Para usar essas opções, marque as caixas de seleção antes dos respectivos mapas e, em seguida, clique no botão desejado.

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 no nome dele na lista de todos os mapas.



Você pode usar a lista suspensa na barra de título do mapa para selecionar o nível de severidade mais baixo dos triggers problemáticos a serem exibidos. A severidade marcada como *default* é o nível definido na configuração do mapa. Se o mapa contiver um submapa, ao navegar para o submapa será mantida a severidade do mapa de nível superior (exceto se ela for *Not classified*; nesse caso, ela não será passada para o submapa).

Destaque de ícones

Se um elemento do mapa estiver em estado de problema, ele será destacado com um círculo redondo. A cor de preenchimento do círculo corresponde à cor da severidade do problema. Somente os problemas com nível de severidade igual ou superior ao selecionado serão exibidos com o elemento. Se todos os problemas forem reconhecidos, uma borda verde espessa ao redor do círculo será exibida.

Além disso:

- um host em manutenção é destacado com um quadrado preenchido em laranja. Observe que o destaque de manutenção tem prioridade sobre o destaque da severidade do problema, se o elemento do mapa for um host.
- um host desabilitado (não monitorado) é destacado com um quadrado preenchido em cinza.

O destaque será exibido se a caixa de seleção *Destaque de ícones* estiver marcada na configuração do mapa.

Marcadores de alterações recentes

Triângulos vermelhos apontando para dentro ao redor de um elemento indicam uma alteração recente no estado do trigger - ocorrida nos últimos 30 minutos. Esses triângulos são exibidos quando a caixa de seleção *Marcar elementos na alteração do status do trigger* está 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 Monitoring.

Resumo legível nos mapas

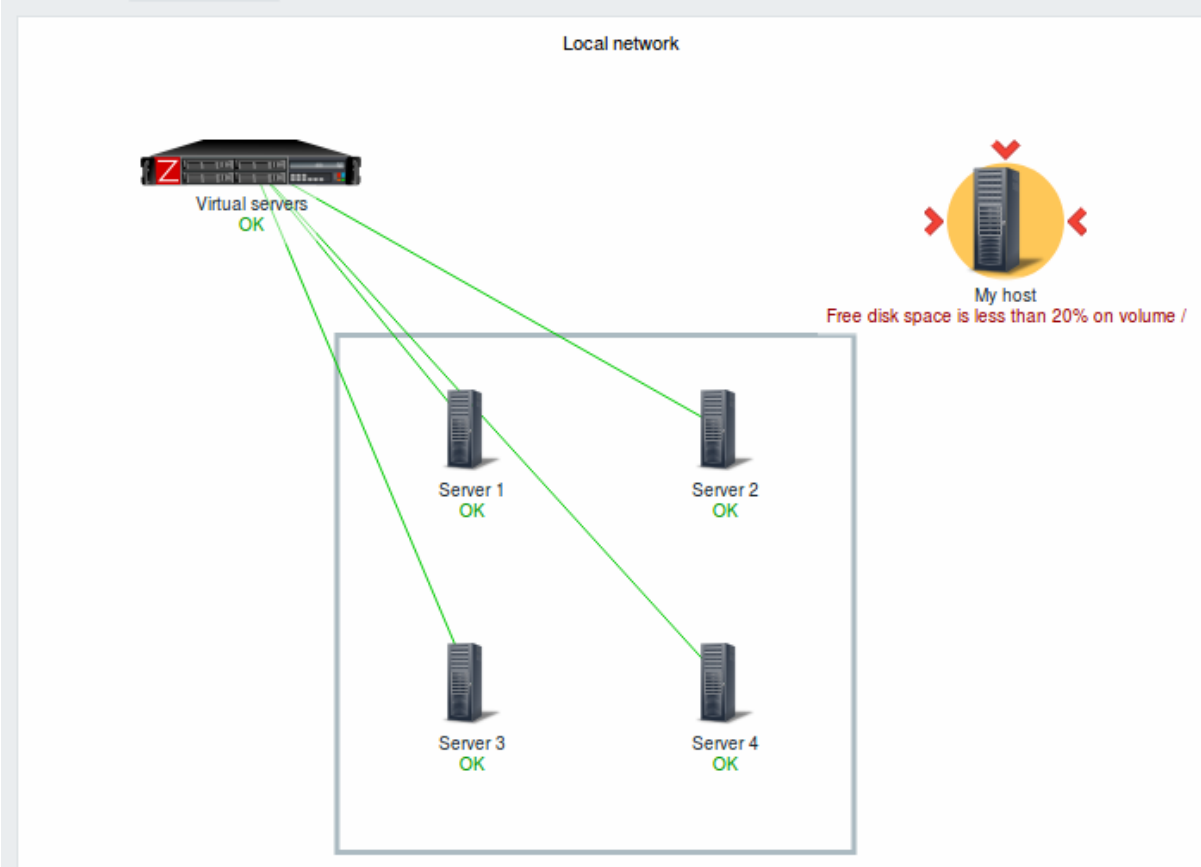
Uma propriedade ”aria-label” oculta está disponível, permitindo que as informações do mapa sejam lidas por um leitor de tela. A descrição geral do mapa e a descrição de cada elemento estão disponíveis no seguinte formato:

- para a descrição do mapa: `<Nome do mapa>, <* de * itens no 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, a seguinte descrição está disponível:

`'Rede local, 1 de 6 elementos no estado de problema, 1 problema no total. Host, Status problema, Meu host,`

para o seguinte mapa:



Referenciando um mapa de rede

Os mapas de rede podem ser referenciados tanto pelos parâmetros GET `sysmapid` quanto `mapname`. Por exemplo,

`http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network`

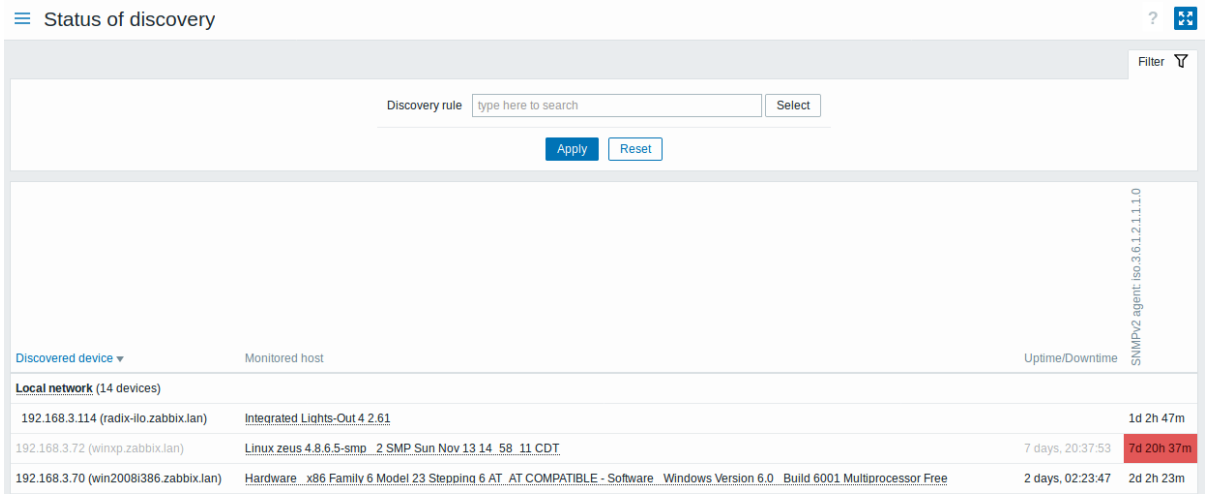
abrirá o mapa com esse nome (Local network).

Se ambos `sysmapid` (ID do mapa) e `mapname` (nome do mapa) forem especificados, `mapname` terá prioridade.

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 e agrupados pela regra de descoberta. Ao clicar na regra de descoberta, é exibido o menu da regra, que contém o link para o formulário de configuração da regra de descoberta. Se um dispositivo tiver vários endereços IP (conforme determinado pelos *critérios de exclusividade do dispositivo* da regra de descoberta), eles serão listados juntos: primeiro o endereço IP principal do dispositivo, seguido pelos demais endereços IP (com recuo). O endereço IP principal é o endereço IP mais baixo que possui um host listado na coluna *Host monitorado* ou, se não houver tal endereço, o endereço IP mais baixo do dispositivo. Os dispositivos podem ser classificados pelo endereço IP principal em ordem crescente ou decrescente; os demais endereços IP de um dispositivo são sempre listados abaixo do endereço IP principal, em ordem crescente.
Host monitorado	Se um dispositivo já estiver sendo monitorado, o nome do host será listado nesta coluna. Ao clicar no nome do host, será exibido o menu do host.
Tempo em atividade/indisponibilidade	A duração desde que o dispositivo foi descoberto ou perdido após a descoberta anterior é exibida nesta coluna. Para um dispositivo com vários endereços IP, a duração é exibida somente para o endereço IP principal.
Verificação de descoberta	O estado do serviço individual (verificação de descoberta) de cada dispositivo descoberto é exibido. Uma célula vermelha indica que o serviço está indisponível. O tempo em atividade ou de indisponibilidade do serviço é incluído na célula. Esta coluna é exibida somente se o serviço tiver sido encontrado em pelo menos um dispositivo descoberto.

Note:
O número de regras de descoberta e o número total de dispositivos descobertos exibidos nesta seção são limitados pelo parâmetro *Limite para resultados de pesquisa e filtro* em *Administração > Geral > Interface gráfica do usuário*. Regras ou dispositivos que excederem o limite não serão exibidos, e nenhuma indicação disso será mostrada.

Botões

Os botões do modo de visualização, que são comuns a todas as seções, são descritos na página Monitoring.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta de seu interesse. Para melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

Quando nada estiver selecionado no filtro, todas as regras de descoberta habilitadas serão exibidas. Para selecionar uma regra de descoberta específica para exibição, comece a digitar o nome dela no filtro. Todas as regras de descoberta habilitadas correspondentes serão listadas para seleção. Mais de uma regra de descoberta pode ser selecionada.

APM

Visão geral

O menu *APM* é destinado aos recursos de monitoramento de desempenho de aplicações (APM) do Zabbix. Ele exibe dados de telemetria enviados por suas aplicações e serviços usando o OpenTelemetry, um padrão aberto para a coleta de dados de telemetria. Esses dados ajudam você a visualizar como as solicitações percorrem suas aplicações e serviços, medir o desempenho deles e encontrar a causa de operações lentas ou com falha.

O menu contém as seguintes seções, que correspondem aos tipos de sinais do OpenTelemetry:

- Traces
- Métricas
- Logs

Diferentemente de *Monitoramento* > *Dados mais recentes*, que exibe valores coletados pelos itens do Zabbix, as seções de *APM* exibem dados de telemetria diretamente da fonte de dados, portanto, não é necessário criar nenhum item. Para exibir dados nas seções de *APM*:

1. Especifique uma fonte de dados de telemetria em *Administração* > *Fonte de dados*.
2. Certifique-se de que os dados de telemetria sejam gravados na fonte de dados pelo proxy do Zabbix com o APM habilitado ou por um coletor do OpenTelemetry. Observe que o server do Zabbix não pode receber dados de telemetria; somente o proxy do Zabbix pode.

Para usar dados de telemetria em triggers ou gráficos, use o item consulta de telemetria.

Rastreamentos

Visão geral

A seção *APM* > *Traces* exibe os traces coletados de seus aplicativos e serviços.

Um **trace** representa o caminho percorrido por uma solicitação através de seus aplicativos e serviços. Por exemplo, quando um usuário clica no botão *Fazer pedido* em uma loja online, a solicitação pode passar pelo serviço frontend, pelos serviços de checkout e pagamento e chegar ao banco de dados.

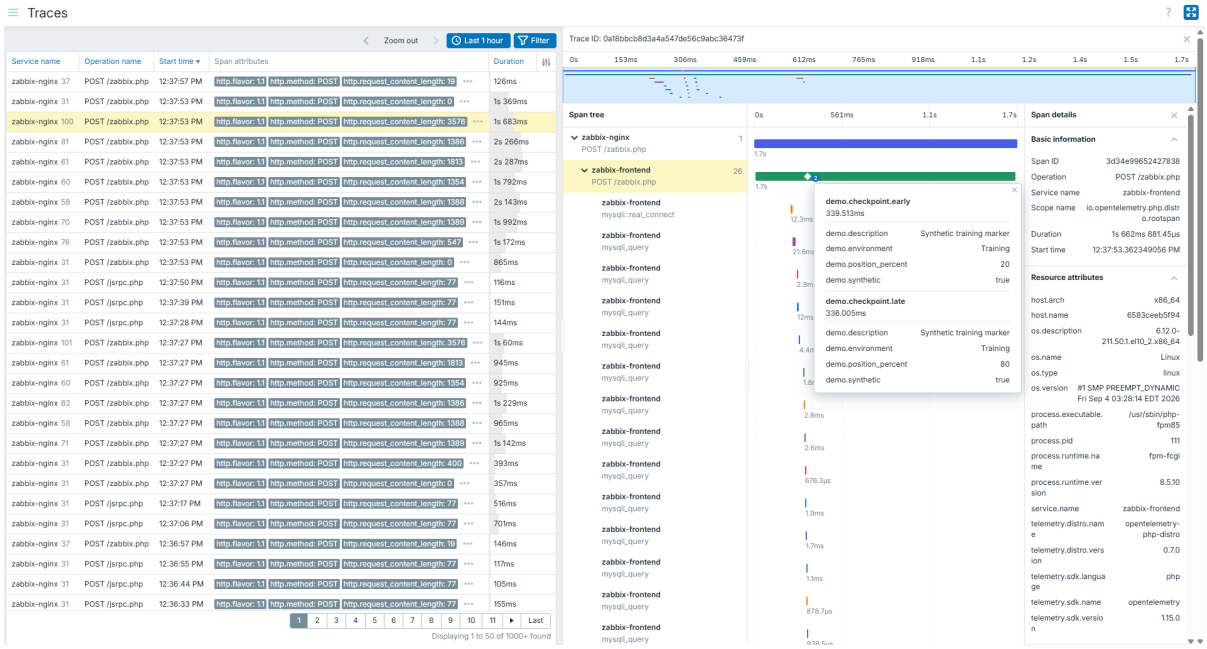
Um trace consiste em **spans**. Cada span representa uma única operação, como uma solicitação HTTP que o serviço frontend envia ao serviço de checkout, uma consulta ao banco de dados que salva o pedido ou uma chamada de função que calcula o custo de envio.

Os spans formam uma hierarquia. O **span raiz** representa a primeira operação da solicitação. Todos os outros spans são **spans filhos** do span durante o qual sua operação foi iniciada. Por exemplo, o serviço frontend que recebe a solicitação *Fazer pedido* é o span raiz, e o serviço de checkout que processa o pedido é seu span filho.

Cada span é produzido por um **serviço** (por exemplo, `zabbix-frontend`; nomeado pelo atributo de recurso `service.name` do OpenTelemetry) e por um escopo de instrumentação, a parte do código que o criou (por exemplo, uma biblioteca que rastreia solicitações HTTP).

Para fornecer contexto sobre cada operação, os spans também têm **atributos** de chave-valor: os atributos de span descrevem a própria operação (por exemplo, `http.method: POST`), enquanto os atributos de recurso descrevem o serviço e o ambiente no qual ele é executado (por exemplo, `host.name: eabf8c2a5ddc`).

Para obter mais detalhes sobre traces, consulte a documentação do OpenTelemetry.



A seção *APM* > *Traces* contém os seguintes elementos:

- Botão Modo quiosque - exibe apenas o conteúdo da página, sem o menu e o cabeçalho.
- Seletor de período - seleciona o período para o qual os traces serão exibidos.
- Filtro - exibe apenas os traces de seu interesse.
- Lista de traces - exibe os traces, descritos por seus spans raiz.
- Painel de trace - exibe todos os spans do trace selecionado em uma linha do tempo.

Lista de traces

Cada linha na lista de traces representa um trace; as colunas mostram os dados do span raiz.

Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793	2s 900ms

Os traces são exibidos em uma tabela personalizável. O botão *Personalizar tabela*, no cabeçalho da tabela, permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefini-la. Todas as personalizações são aplicadas imediatamente e lembradas para o usuário atual.

A lista de traces pode ter as seguintes colunas:

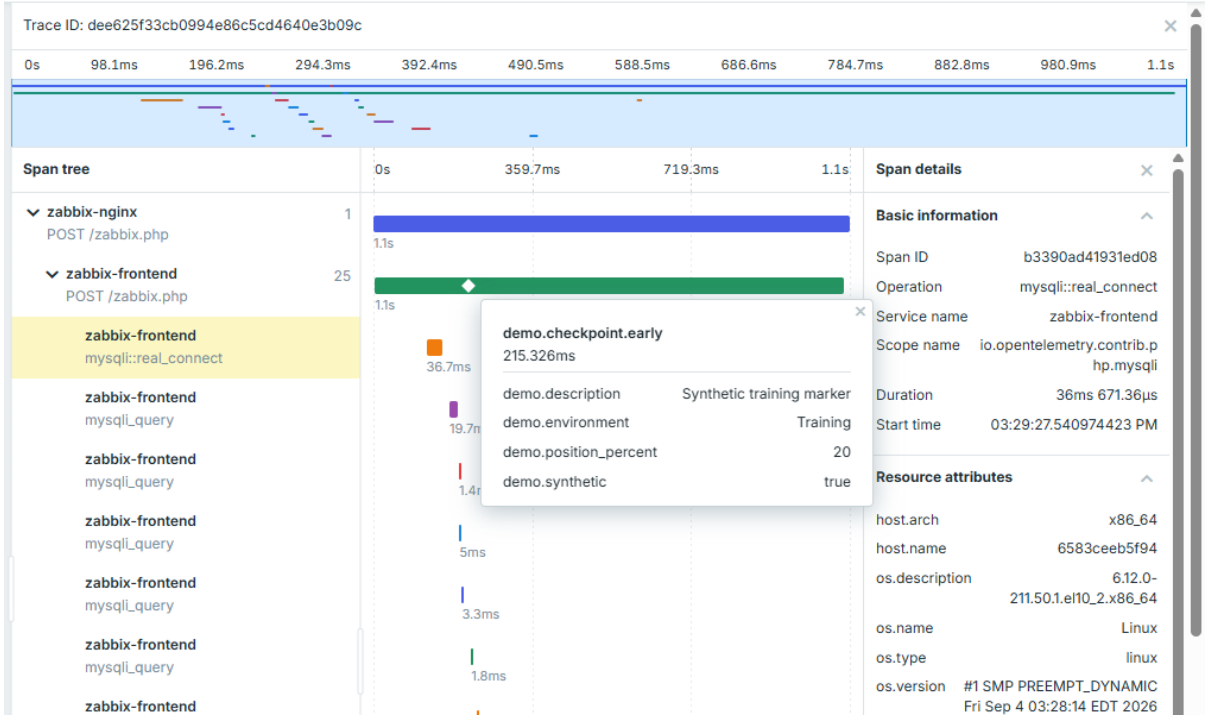
Note:
Cada coluna corresponde a uma coluna na tabela `otel_traces` da fonte de dados de telemetria; para ver a lista de colunas, consulte Colunas de dados de telemetria compatíveis. A maioria das colunas tem o mesmo nome que na fonte de dados (por exemplo, *ID do trace* corresponde à coluna `TraceId`). Quando os nomes diferem, a coluna da fonte de dados é especificada na tabela abaixo.

Coluna	Descrição
ID do trace	ID do trace, compartilhado por todos os spans que pertencem ao trace.
ID do span	ID do span raiz.

Coluna	Descrição
Estado do trace	Dados adicionais, se houver, que as ferramentas de tracing associam ao trace (consulte W3C Trace Context).
Nome do serviço	Serviço que produziu o span raiz, seguido pelo número de spans no trace. Se algum span tiver o código de status *Error* (consulte abaixo), um ícone vermelho mostrará o número deles.
Nome da operação	Operação representada pelo span raiz (por exemplo, `POST /zabbix.php`). Coluna da fonte de dados: `SpanName`.
Tipo do span	Função do span raiz: **Unspecified** - não especificada. **Internal** - operação executada dentro de um aplicativo. **Server** - tratamento de uma solicitação recebida. **Client** - solicitação de saída. **Producer** - criação de uma operação assíncrona (por exemplo, envio de uma mensagem). **Consumer** - processamento de uma operação assíncrona.
Nome do escopo	Escopo de instrumentação que produziu o span raiz.
Versão do escopo	Versão do escopo de instrumentação.
Código de status	Status da operação representada pelo span raiz: **Unset** - nenhum status explícito foi atribuído ao span raiz. **Ok** - a operação foi concluída com sucesso. **Error** - a operação foi concluída com um erro.
Mensagem de status	Informações adicionais, se houver, sobre o status.
Hora de início	Data e hora em que o span raiz foi iniciado. Coluna da fonte de dados: `Timestamp`.
Atributos do span	Atributos do span raiz. Clique no botão de três pontos ••• para exibir todos os atributos.
Atributos do recurso	Atributos do recurso do span raiz. Clique no botão de três pontos ••• para exibir todos os atributos.
Duração	Duração do trace; a barra cinza mostra a duração em relação aos outros traces.

Painel de trace

Clique em um trace na lista de traces para abrir o painel de trace, que mostra todos os spans do trace em uma linha do tempo: quais operações foram realizadas, em que ordem e quanto tempo cada uma levou.

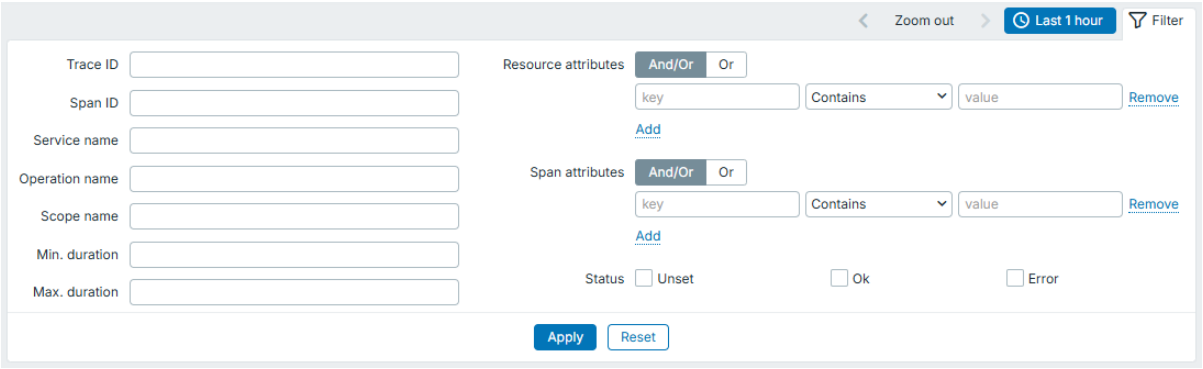


O painel de trace pode conter os seguintes elementos:

Elemento	Descrição
ID do trace	ID do trace selecionado.
Visão geral da linha do tempo do trace	O trace completo, com spans em até 8 linhas (vários spans podem compartilhar uma linha). Arraste as bordas do intervalo selecionado, ou o próprio intervalo, para ampliar a linha do tempo dos spans nessa parte do trace.
Árvore de spans	Spans organizados em sua hierarquia; cada um mostra seu serviço (em negrito), o nome da operação e o número de spans filhos diretos.
Linha do tempo dos spans	Spans representados como barras: a posição e o comprimento mostram quando cada span foi iniciado e quanto tempo levou. Os marcadores mostram eventos (por exemplo, exceções); passe o cursor sobre um marcador ou clique nele para ver os detalhes.
Detalhes do span	Clique em um span na árvore de spans para ver seus detalhes e atributos.

Usando o filtro

Você pode usar o filtro para exibir apenas os traces nos quais está interessado.



Para filtrar por atributos, insira a chave do atributo, selecione o operador e insira o valor do atributo.

Os seguintes operadores são compatíveis:

- **Contém** - *(padrão)* o valor do atributo contém a string inserida.
- **É igual a** - o valor do atributo é igual à string inserida.
- **Não contém** - o valor do atributo não contém a string inserida.
- **Não é igual a** - o valor do atributo não é igual à string inserida.
- **Existe** - o atributo com a chave inserida existe.
- **Não existe** - o atributo com a chave inserida não existe.

Se você adicionar várias condições, selecione como avaliá-las:

- **E/Ou** - *(padrão)* todas as condições devem ser atendidas; as condições com a mesma chave de atributo são combinadas com *Ou*.
- **Ou** - pelo menos uma condição deve ser atendida.

Métricas

Visão geral

A seção *APM* > *Métricas* exibe pontos de dados de métricas coletados de seus aplicativos e serviços.

Uma **métrica** é uma medição (por exemplo, duração de uma solicitação, número de pedidos processados ou uso de memória). Os aplicativos enviam seus valores como **pontos de dados** que contêm o valor (ou valores agregados) medido em um momento específico. Uma métrica é semelhante a um item do Zabbix, e um ponto de dados é semelhante a um valor desse item.

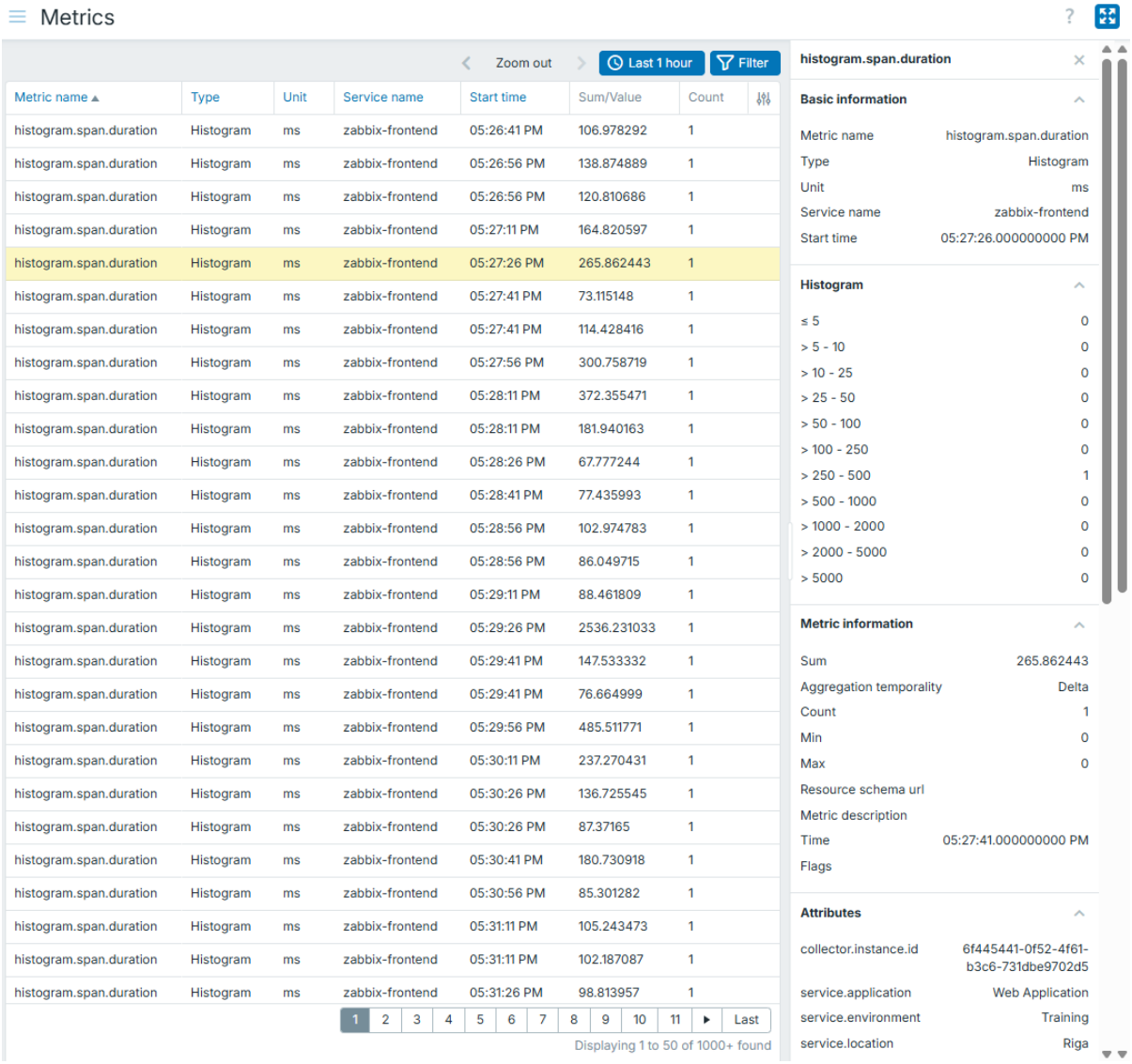
Um ponto de dados pode ser de um dos seguintes tipos:

- **Gauge** - um valor medido em um momento específico (por exemplo, uso atual de memória).
- **Sum** - uma soma de medições ao longo de um período (por exemplo, número total de solicitações).
- **Histograma** - o número de medições ao longo de um período que se enquadram em cada uma de várias faixas de valores fixas, chamadas de buckets (por exemplo, quantas solicitações levaram de 0 a 100 ms, de 100 a 250 ms ou de 250 a 500 ms).
- **Histograma exponencial** - semelhante a um histograma, mas cada bucket é mais amplo que o anterior pelo mesmo fator (por exemplo, 1-2 ms, 2-4 ms, 4-8 ms).

Cada ponto de dados é produzido por um **serviço** (por exemplo, `zabbix-frontend`; nomeado pelo atributo de recurso `service.name` do OpenTelemetry) e por um escopo de instrumentação, a parte do código que o criou (por exemplo, uma biblioteca que mede a duração de solicitações HTTP).

Para fornecer contexto sobre cada medição, os pontos de dados também têm **atributos** de chave-valor: atributos do ponto de dados descrevem o que foi medido e distinguem pontos de dados da mesma métrica (por exemplo, `http.request.method: POST`), enquanto atributos de recurso descrevem o serviço e o ambiente em que ele é executado (por exemplo, `host.name: eabf8c2a5ddc`).

Para obter mais detalhes sobre métricas, consulte a documentação do OpenTelemetry.

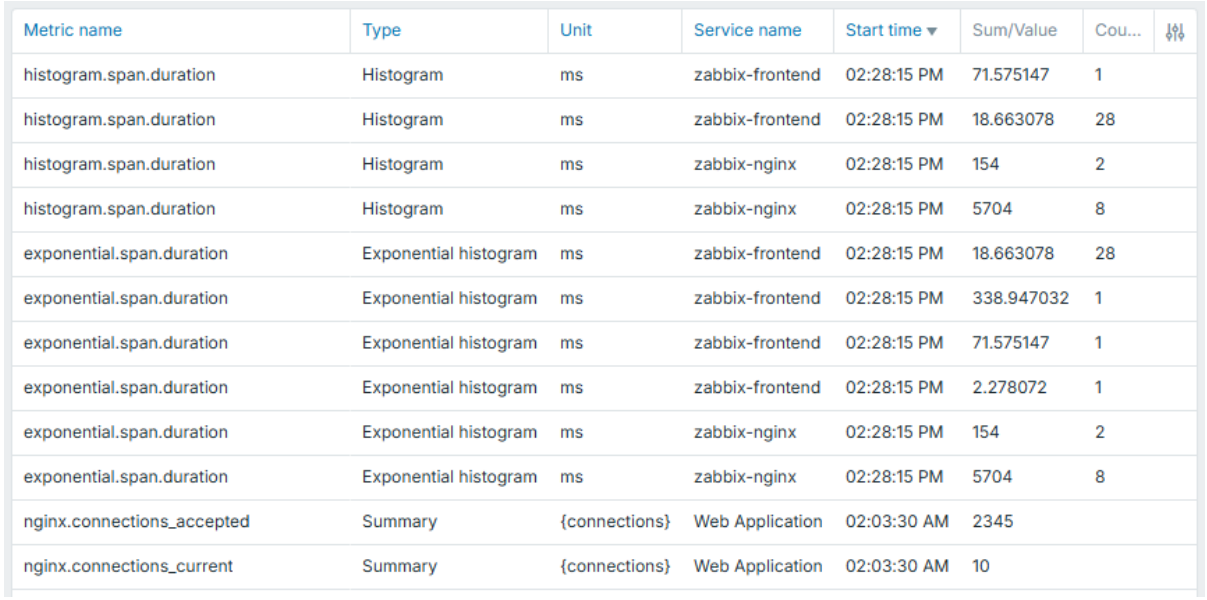


A seção *APM* > *Métricas* contém os seguintes elementos:

- Botão Modo quiosque - exibe apenas o conteúdo da página, sem o menu e o cabeçalho.
- Seletor de período - seleciona o período para o qual os pontos de dados serão exibidos.
- Filtro - exibe apenas os pontos de dados de seu interesse.
- Lista de métricas - exibe os pontos de dados.
- Detalhes da métrica - exibe os detalhes do ponto de dados selecionado.

Lista de métricas

Cada linha na lista de métricas representa um ponto de dados.



Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

Os pontos de dados são exibidos em uma tabela personalizável.

O botão *Personalizar tabela*, no cabeçalho da tabela, permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão.

Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefinir o tamanho.

Todas as personalizações são aplicadas imediatamente e lembradas para o usuário atual.

A lista de métricas pode conter as seguintes colunas:

Note:
Cada coluna, exceto *Tipo*, corresponde a uma coluna nas tabelas de métricas (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` e `otel_metrics_exponential_histogram`) da fonte de dados de telemetria; para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis. A maioria das colunas tem o mesmo nome que na fonte de dados (por exemplo, *Nome da métrica* corresponde à coluna `MetricName`). Quando os nomes diferem, a coluna da fonte de dados é especificada na tabela abaixo.

Coluna	Descrição
Nome da métrica	Métrica à qual o ponto de dados pertence (por exemplo, `http.server.duration`).
Tipo	Tipo do ponto de dados: *Gauge*, *Sum*, *Histogram* ou *Exponential histogram*. Determinado pela tabela de métricas que contém o ponto de dados.
Unidade	Unidade, se houver, dos valores da métrica (por exemplo, `s` ou `By`; consulte UCUM). Coluna da fonte de dados: `MetricUnit`.
Nome do serviço	Serviço que produziu o ponto de dados.
Nome do escopo	Escopo de instrumentação que produziu o ponto de dados.
Hora de início	Data e hora em que o período de medição do ponto de dados começou. Coluna da fonte de dados: `StartTimeUnix`.
Soma/Valor	Soma de todas as medições (*Histogram*, *Exponential histogram*) ou o valor (*Gauge*, *Sum*) do ponto de dados.
Contagem	Número de medições em um ponto de dados *Histogram* ou *Exponential histogram*.
Hora	Data e hora do ponto de dados; para pontos de dados *Sum*, *Histogram* e *Exponential histogram*, o fim do período de medição. Coluna da fonte de dados: `TimeUnix`.
Flags	Flags do ponto de dados: **0** - nenhuma flag está definida. **1** - nenhum valor foi registrado para o ponto de dados.
Temporalidade da agregação	Como os valores de um ponto de dados *Sum*, *Histogram* ou *Exponential histogram* são agregados ao longo do tempo: **Não especificada** - não especificada. **Delta** - agregados durante o período desde o ponto de dados anterior. **Cumulativa** - agregados durante o período desde a hora de início.
Descrição da métrica	Descrição, se houver, da métrica.
Atributos	Atributos do ponto de dados. Clique no botão de três pontos ••• para exibir todos os atributos.

Coluna	Descrição
Atributos do recurso	Atributos do recurso do ponto de dados. Clique no botão de três pontos ••• para exibir todos os atributos.
Atributos do escopo	Atributos do escopo de instrumentação. Clique no botão de três pontos ••• para exibir todos os atributos.

Detalhes da métrica

Clique em um ponto de dados na lista de métricas para abrir o painel de detalhes da métrica, que exibe todas as informações sobre o ponto de dados.

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

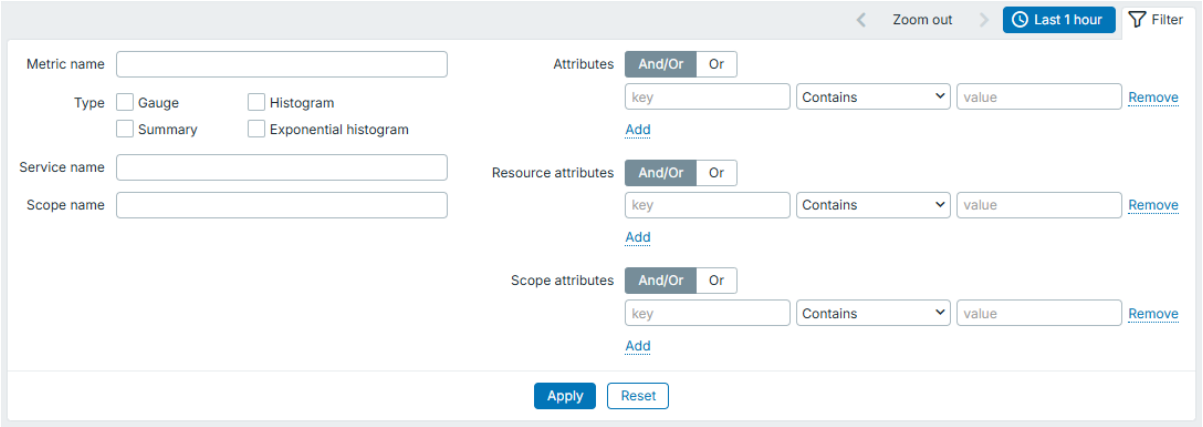
Scope

O painel de detalhes da métrica pode conter as seguintes seções:

Seção	Descrição
Informações básicas	Nome da métrica, tipo, unidade, nome do serviço e hora de início.
Histograma	Buckets (por exemplo, `> 5 – 10`), cada um com o número de medições nele. Exibido somente para pontos de dados do tipo *Histograma*.
Histograma exponencial	Buckets (por exemplo, `>= 249.7885 – 249.7887`), cada um com o número de medições nele. Exibido somente para pontos de dados do tipo *Histograma exponencial*.
Informações da métrica	Valores do ponto de dados, que dependem do tipo, além da descrição da métrica, hora, flags e URL do esquema do recurso.
Atributos	Atributos do ponto de dados.
Atributos do recurso	Atributos do recurso.
Atributos do escopo	Atributos, se houver, do escopo de instrumentação.
Escopo	Nome, versão e URL do esquema do escopo de instrumentação.

Usando o filtro

Você pode usar o filtro para exibir apenas os pontos de dados nos quais está interessado.



Para filtrar por atributos, insira a chave do atributo, selecione o operador e insira o valor do atributo.

Os seguintes operadores são compatíveis:

- **Contém** - *(padrão)* o valor do atributo contém a string inserida.
- **É igual a** - o valor do atributo é igual à string inserida.
- **Não contém** - o valor do atributo não contém a string inserida.
- **Não é igual a** - o valor do atributo não é igual à string inserida.
- **Existe** - o atributo com a chave inserida existe.
- **Não existe** - o atributo com a chave inserida não existe.

Se você adicionar várias condições, selecione como avaliá-las:

- **E/Ou** - *(padrão)* todas as condições devem ser atendidas; as condições com a mesma chave de atributo são combinadas com *Ou*.
- **Ou** - pelo menos uma condição deve ser atendida.

Logs

Visão geral

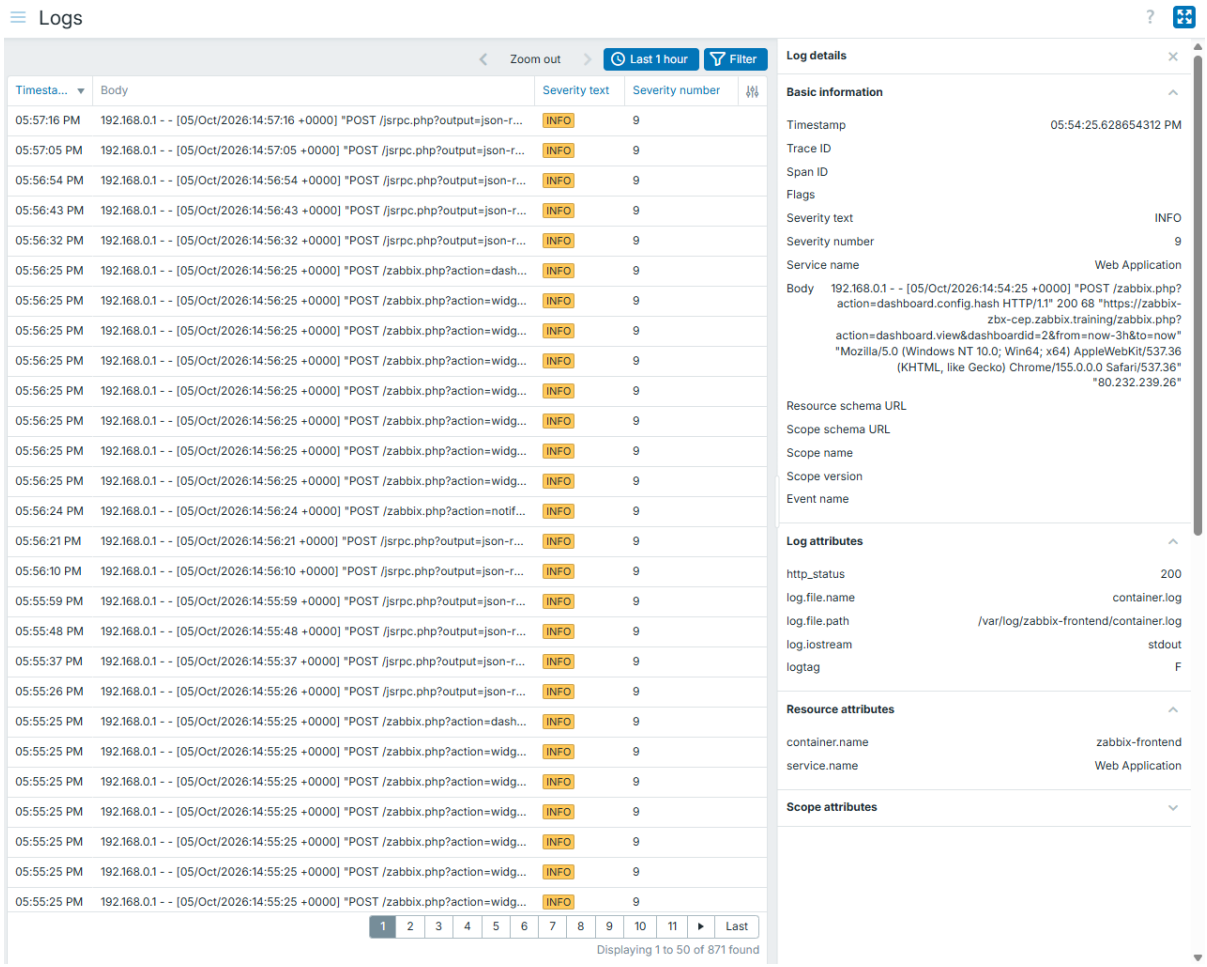
A seção *APM > Logs* exibe registros de log coletados de seus aplicativos e serviços.

Um **registro de log** é um registro de texto de um evento (por exemplo, um pagamento com falha ou um tempo limite do banco de dados). Cada registro de log tem uma gravidade e uma mensagem (corpo). Se um registro de log foi produzido durante uma operação que faz parte de um trace, ele também contém os IDs desse trace e span.

Cada registro de log é produzido por um **serviço** (por exemplo, `zabbix-frontend`; nomeado pelo atributo de recurso `service.name` do OpenTelemetry) e por um escopo de instrumentação, a parte do código que o criou (por exemplo, uma biblioteca de logging).

Para fornecer contexto sobre cada evento, os registros de log também têm **atributos** de chave-valor: os atributos de log descrevem o próprio evento (por exemplo, `log.file.path: /var/log/zabbix/zabbix_frontend.log`), enquanto os atributos de recurso descrevem o serviço e o ambiente no qual ele é executado (por exemplo, `host.name: eabf8c2a5ddc`).

Para obter mais detalhes sobre logs, consulte a documentação do OpenTelemetry.

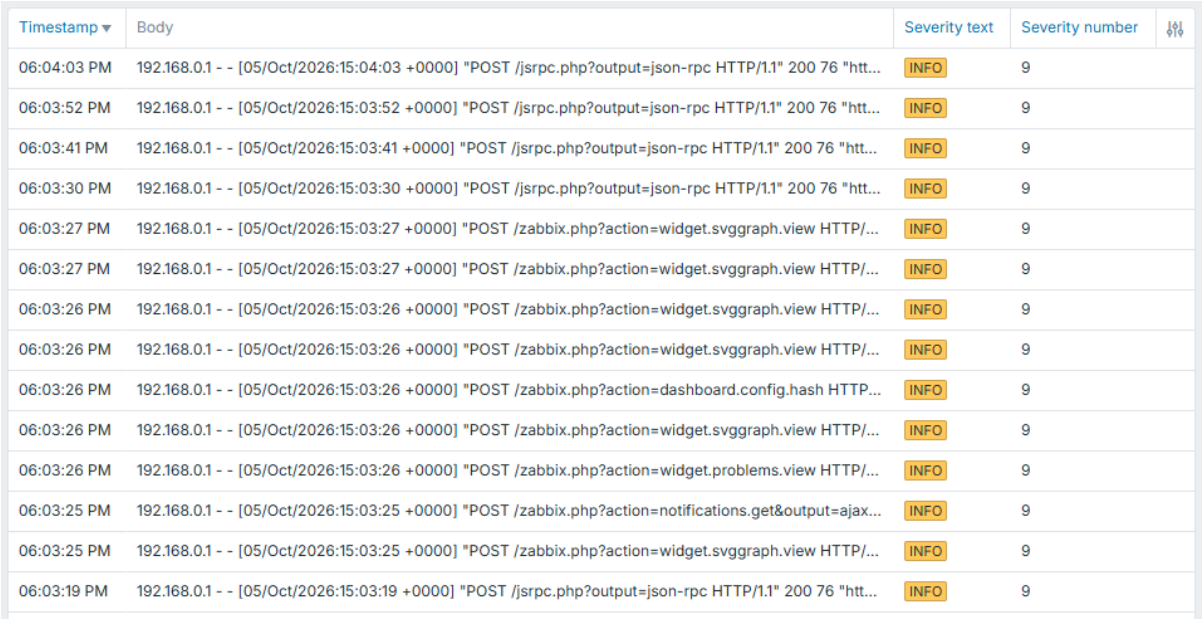


A seção *APM* > *Logs* contém os seguintes elementos:

- Botão Modo quiosque - exibe apenas o conteúdo da página, sem o menu e o cabeçalho.
- Seletor de período - seleciona o período para o qual os registros de log serão exibidos.
- Filtro - exibe apenas os registros de log de seu interesse.
- Lista de logs - exibe os registros de log.
- Detalhes do log - exibe os detalhes do registro de log selecionado.

Lista de logs

Cada linha na lista de logs representa um registro de log.



Timestamp ▾	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

Os registros de log são exibidos em uma tabela personalizável. O botão *Personalizar tabela*, no cabeçalho da tabela, permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefini-la. Todas as personalizações são aplicadas imediatamente e lembradas para o usuário atual.

A lista de logs pode conter as seguintes colunas:

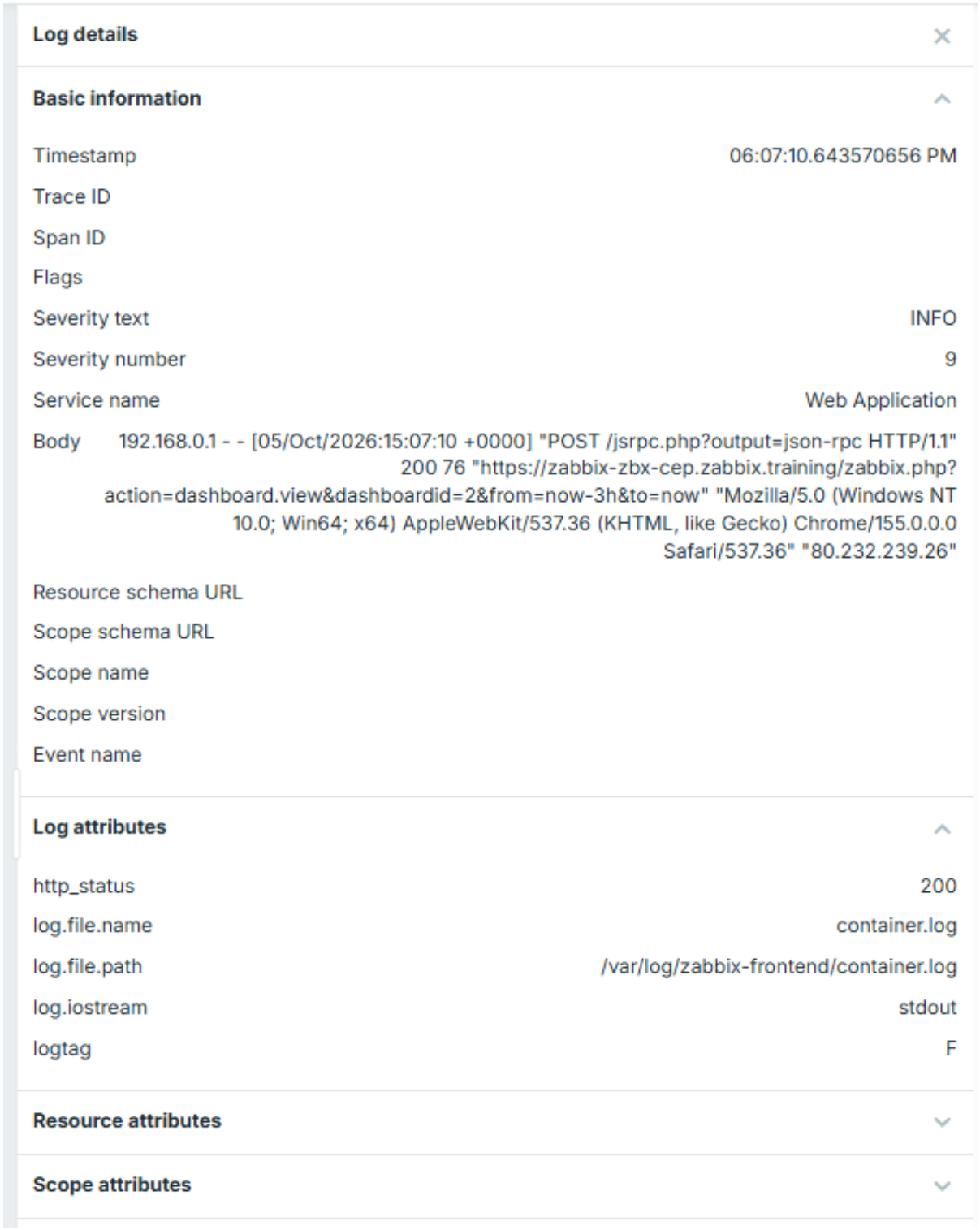
Note:
Cada coluna corresponde a uma coluna na tabela `otel_logs` da fonte de dados de telemetria; para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis. A maioria das colunas tem o mesmo nome que na fonte de dados (por exemplo, *Texto da severidade* corresponde à coluna `SeverityText`). Quando os nomes são diferentes, a coluna da fonte de dados é especificada na tabela abaixo.

Coluna	Descrição
Timestamp	Data e hora em que o evento ocorreu.
Corpo	Mensagem do registro de log.
Texto da severidade	Nível de severidade fornecido pelo aplicativo (por exemplo, `INFO`, `WARN` ou `ERROR`). A cor do rótulo depende da severidade.
Número da severidade	Nível de severidade como um número: **0** - NÃO ESPECIFICADO. **1-4** - TRACE. **5-8** - DEBUG. **9-12** - INFO. **13-16** - WARN. **17-20** - ERROR. **21-24** - FATAL.
ID do trace	ID, se houver, do trace durante o qual o registro de log foi produzido.
ID do span	ID, se houver, do span durante o qual o registro de log foi produzido.
Flags	Flags de trace do registro de log: **0** - o trace não é amostrado. **1** - o trace é amostrado (registrado). Coluna da fonte de dados: `TraceFlags`.
Nome do serviço	Serviço que produziu o registro de log.
Nome do escopo	Escopo de instrumentação que produziu o registro de log.
Versão do escopo	Versão do escopo de instrumentação.
Nome do evento	Nome, se houver, do tipo de evento que o registro de log representa.
Atributos do log	Atributos de log do registro de log. Clique no botão de três pontos ••• para exibir todos os atributos.
Atributos do recurso	Atributos de recurso do registro de log. Clique no botão de três pontos ••• para exibir todos os atributos.

Coluna	Descrição
Atributos do escopo	Atributos do escopo de instrumentação. Clique no botão de três pontos ••• para exibir todos os atributos.

Detalhes do log

Clique em um registro de log na lista de logs para abrir o painel *Detalhes do log*, que exibe todas as informações sobre o registro de log.



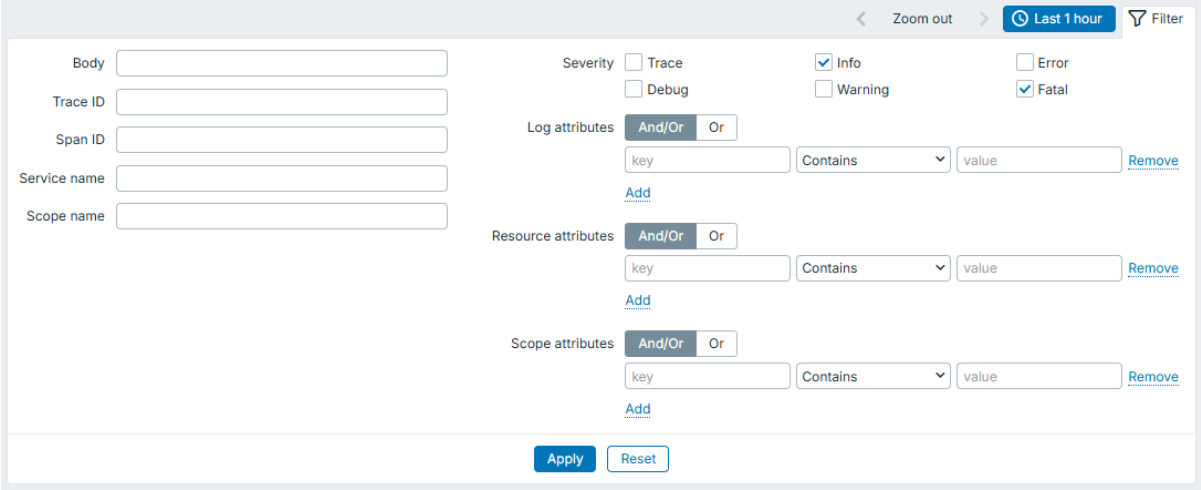
O painel *Detalhes do log* pode conter as seguintes seções:

Seção	Descrição
Informações básicas	ID do trace, ID do span e flags de trace do registro de log.
Atributos do log	Atributos de log do registro de log.

Seção	Descrição
Atributos do recurso	URL do esquema e atributos do recurso.
Atributos do escopo	URL do esquema, nome, versão e atributos do escopo de instrumentação.

Usando o filtro

Você pode usar o filtro para exibir apenas os registros de log nos quais está interessado.



Para filtrar por atributos, insira a chave do atributo, selecione o operador e insira o valor do atributo.

Os seguintes operadores são compatíveis:

- **Contém** - *(padrão)* o valor do atributo contém a string inserida.
- **É igual a** - o valor do atributo é igual à string inserida.
- **Não contém** - o valor do atributo não contém a string inserida.
- **Não é igual a** - o valor do atributo não é igual à string inserida.
- **Existe** - o atributo com a chave inserida existe.
- **Não existe** - o atributo com a chave inserida não existe.

Se você adicionar várias condições, selecione como avaliá-las:

- **E/Ou** - *(padrão)* todas as condições devem ser atendidas; as condições com a mesma chave de atributo são combinadas com *Ou*.
- **Ou** - pelo menos uma condição deve ser atendida.

Serviços

Visão geral

O menu Serviços é destinado às funções de monitoramento de serviços do Zabbix.

Serviços

Visão geral

Nesta seção, você pode ver o status geral de todos os serviços que foram configurados no Zabbix, com base na sua infraestrutura.

Um serviço pode ser uma hierarquia composta por vários níveis de outros serviços, chamados serviços ”filhos”, que contribuem para o status geral do serviço (consulte também uma visão geral da funcionalidade de monitoramento de serviços).

As principais categorias do estado do serviço são *OK* ou *Problema*, sendo que o estado *Problema* é expresso pelo nome e pela cor correspondentes à severidade do problema.

Embora o modo de visualização permita monitorar os serviços juntamente com seus status e outros detalhes, você também pode configurar a hierarquia de serviços nesta seção (adicionar/editar serviços e serviços filhos) alternando para o modo de edição.

Para alternar do modo de visualização para o modo de edição (e vice-versa), clique no botão correspondente no canto superior direito:

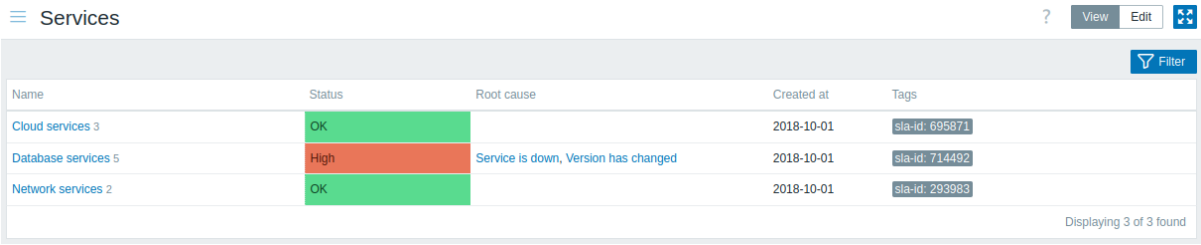
- 

- visualizar serviços

- Edit - adicionar/editar serviços e serviços filhos

Observe que o acesso à edição depende das configurações da função de usuário.

Visualizando services



Uma lista dos services existentes é exibida.

Dados exibidos:

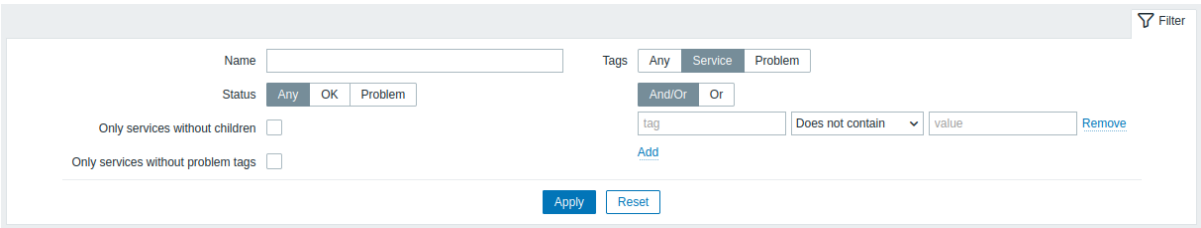
Parameter	Description
Name	Nome do service. O nome do service é um link para detalhes do service. O número após o nome indica quantos child services o service possui.
Status	Estado do service: **OK** - sem problemas **<problem color and severity>** - indica um problema e sua severidade. Em caso de múltiplos problemas, a cor e a severidade do problema mais crítico são exibidas.
Root cause	Problemas subjacentes que afetam direta ou indiretamente o estado do service são listados. Os mesmos problemas são listados conforme retornado pela macro {SERVICE.ROOTCAUSE} macro. Clique no nome do problema para ver mais detalhes sobre ele em *Monitoring* > *Problems*. Problemas que não afetam o estado do service não aparecem na lista.
Created at	O horário em que o service foi criado é exibido.
Tags	Tags do service são exibidas. As tags são usadas para identificar um service em ações e SLAs.

Botões

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 serviços do seu interesse.



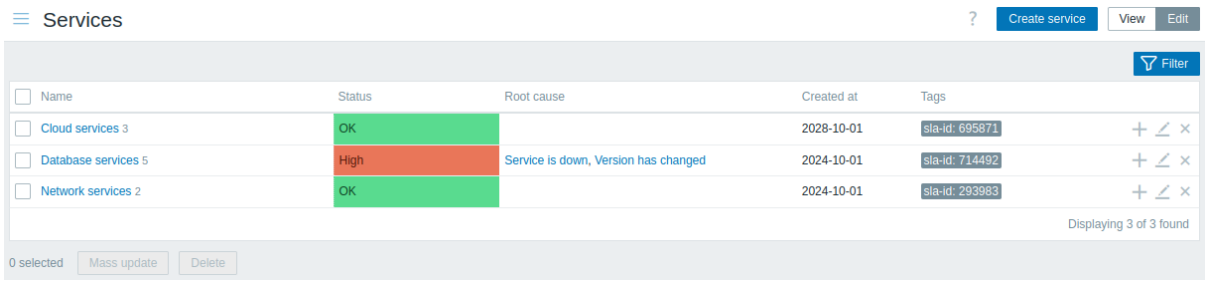
Parâmetro	Descrição
Nome	Filtrar pelo nome do serviço.
Status	Filtrar pelo estado do serviço.
Apenas serviços sem filhos	Marque a caixa de seleção para exibir apenas serviços sem serviços filhos. Este parâmetro está disponível apenas no modo de edição do serviço.
Apenas serviços sem tags de problema	Marque a caixa de seleção para exibir apenas serviços sem tags de problema. Este parâmetro está disponível apenas no modo de edição do serviço.

Parâmetro	Descrição
Tags	Filtrar pelo nome e valor da tag do serviço, ou pelo nome e valor da tag de problema do serviço (no modo de edição do serviço). 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.

Editando serviços

Clique no botão *Edit* 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 com estas opções adicionais:

-  - adicionar um serviço filho a este serviço
-  - editar este serviço
-  - excluir este serviço

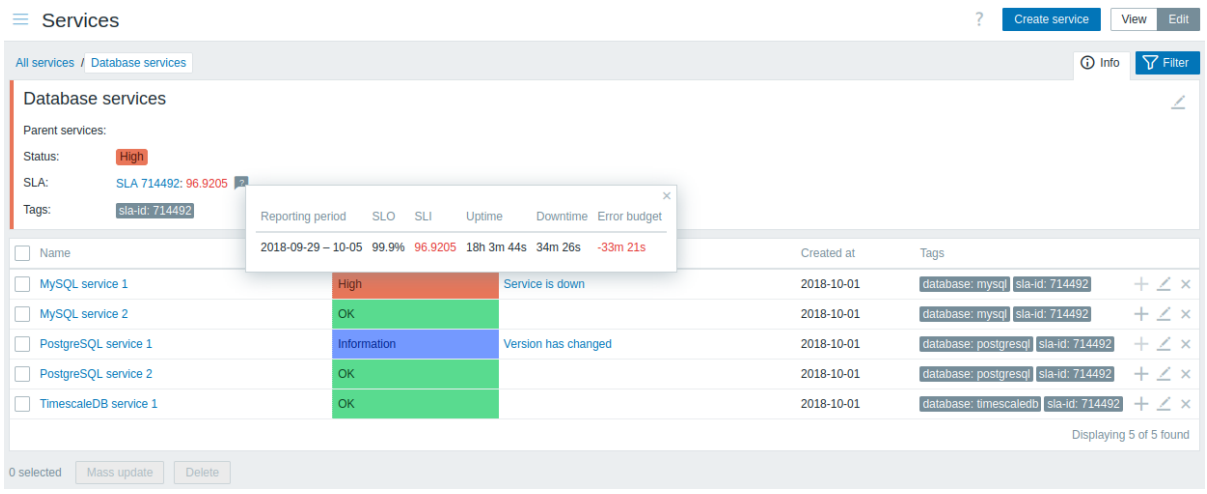


Para configurar um novo serviço, clique no botão *Create service* no canto superior direito.

Detalhes do serviço

Para acessar os detalhes do serviço, clique no nome do serviço. Para retornar à lista de todos os serviços, clique em *All services*.

Os detalhes do serviço incluem a caixa de informações e a lista de serviços filhos.



Para acessar a caixa de informações, clique na aba *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 `SLA name:service level indicator`. 'SLA name' também é um link para o relatório de SLA deste serviço. Se você posicionar o mouse sobre a caixa de informações ao lado do indicador de nível de serviço (SLI), será exibida uma lista pop-up com os 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 aba *Filter*.

Quando estiver no modo de edição, a listagem de serviços filhos é complementada com opções adicionais de edição:

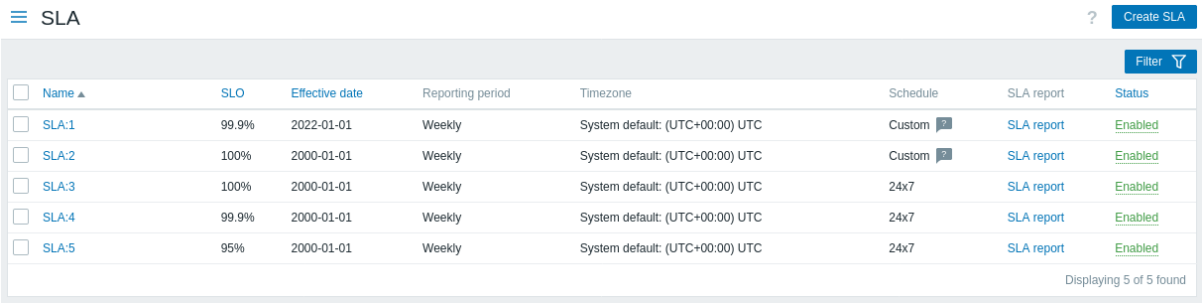
-  - adicionar um serviço filho a este serviço
-  - editar este serviço
-  - excluir este serviço

SLA

Visão geral

Esta seção permite visualizar e configurar SLAs.

SLAs



Uma lista dos SLAs configurados é exibida. *Observe* que somente os SLAs relacionados aos serviços acessíveis ao usuário serão exibidos (como somente leitura, a menos que *Manage SLA* esteja habilitado para a função do usuário).

Dados exibidos:

Parameter	Description
Name	O nome do SLA é exibido. O nome é um link para SLA configuration.
SLO	O objetivo de nível de serviço (SLO) é exibido.
Effective date	A data de início do cálculo do SLA é exibida.
Reporting period	O período usado no relatório de SLA é exibido - *daily*, *weekly*, *monthly*, *quarterly*, or *annually*.
Time zone	O fuso horário do SLA é exibido.
Schedule	A programação do SLA é exibida - 24x7 ou personalizada.
SLA report	Clique no link para ver o relatório de SLA deste SLA.
Status	O status do SLA é exibido - habilitado ou desabilitado.

Relatório de SLA

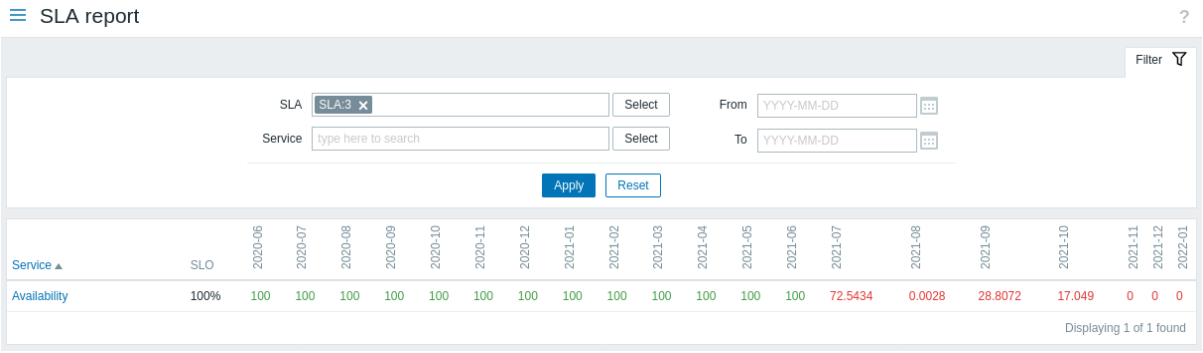
Visão geral

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 e também no nome do serviço. Também é possível limitar o período exibido.

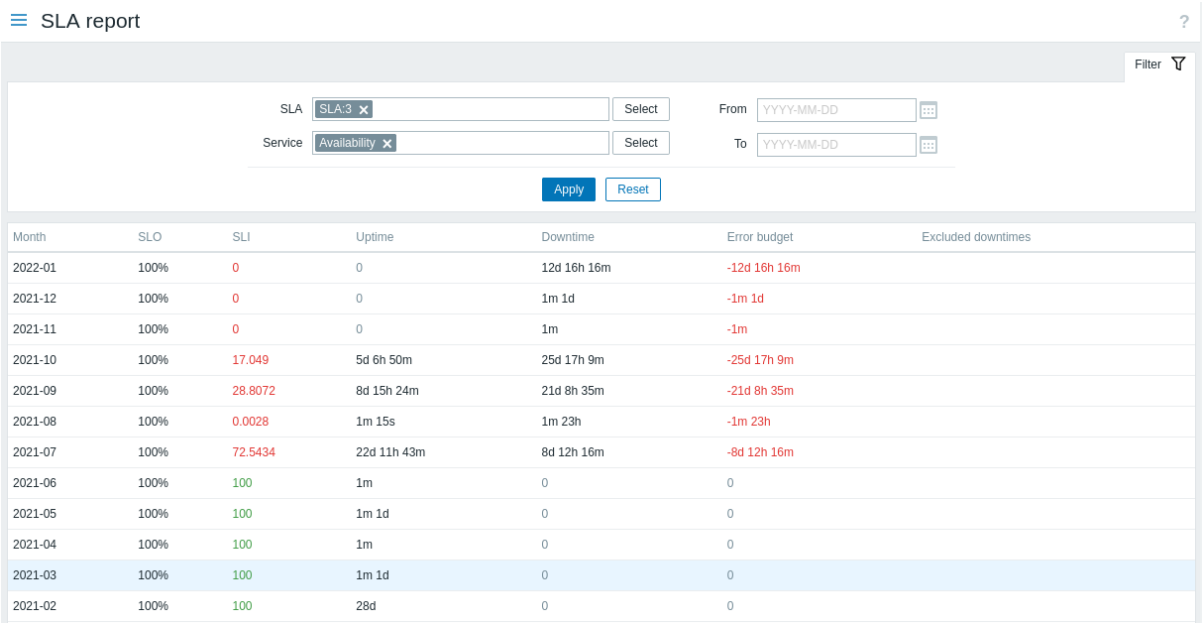


Cada coluna (período) exibe o SLI correspondente a esse período. Os SLIs que violam o SLO definido são destacados em vermelho.

O relatório exibe 20 períodos. É possível exibir no máximo 100 períodos se 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 duração negativa do problema não afeta o cálculo ou a geração de relatórios de SLA.

Inventário

Visão geral

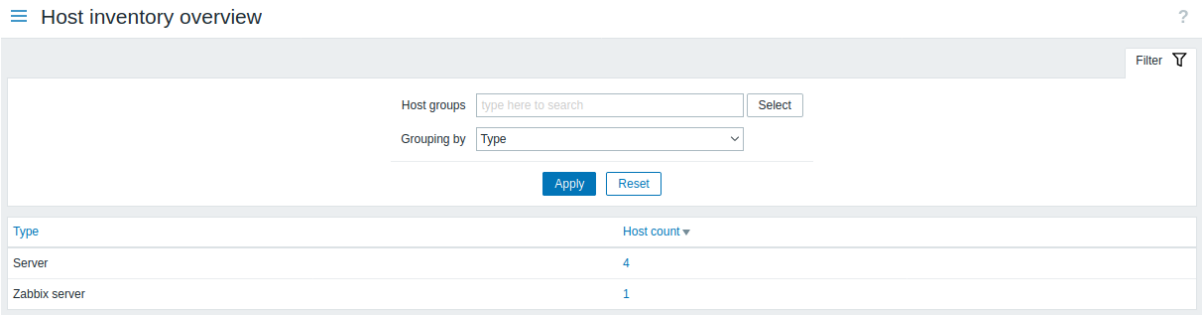
O menu Inventário apresenta seções que fornecem uma visão geral dos dados de inventário do host por um parâmetro escolhido, bem como a capacidade de visualizar detalhes do inventário do host.

Visão geral

Visão geral

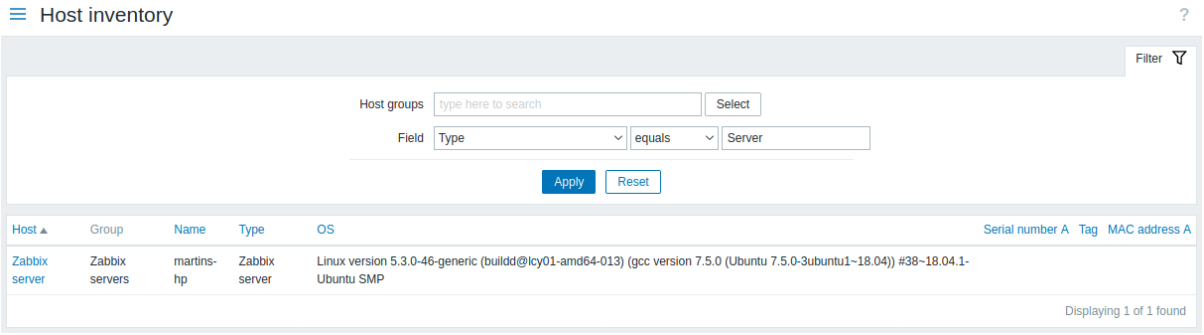
A seção *Inventory* > *Overview* fornece maneiras de obter uma visão geral dos dados de inventário de host.

Para que uma visão geral seja exibida, escolha grupos de hosts (ou nenhum) e o campo de inventário pelo qual os dados serão exibidos. 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 para os hosts.

Os números na coluna *Host count* são links; eles levam à filtragem desses hosts na tabela *Host Inventories*.

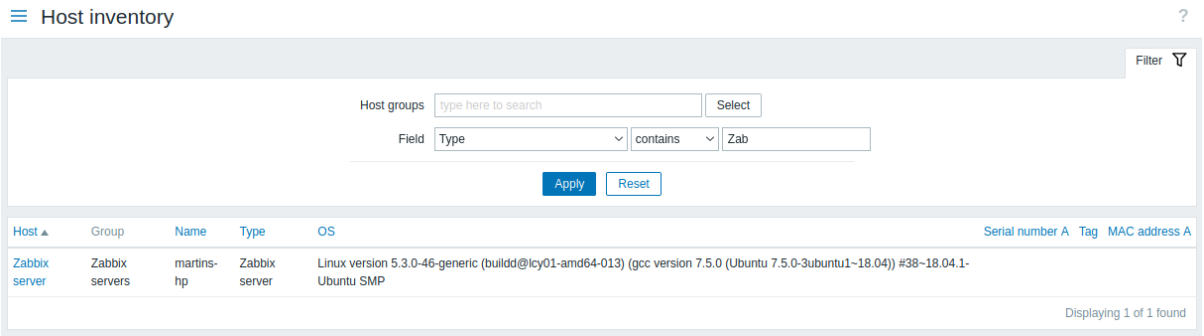


Hosts

Visão geral

Na seção *Inventory* > *Hosts* inventory data dos hosts são exibidos.

Você pode filtrar os hosts por grupo(s) de host e por qualquer campo de inventory para exibir apenas os hosts de seu interesse.

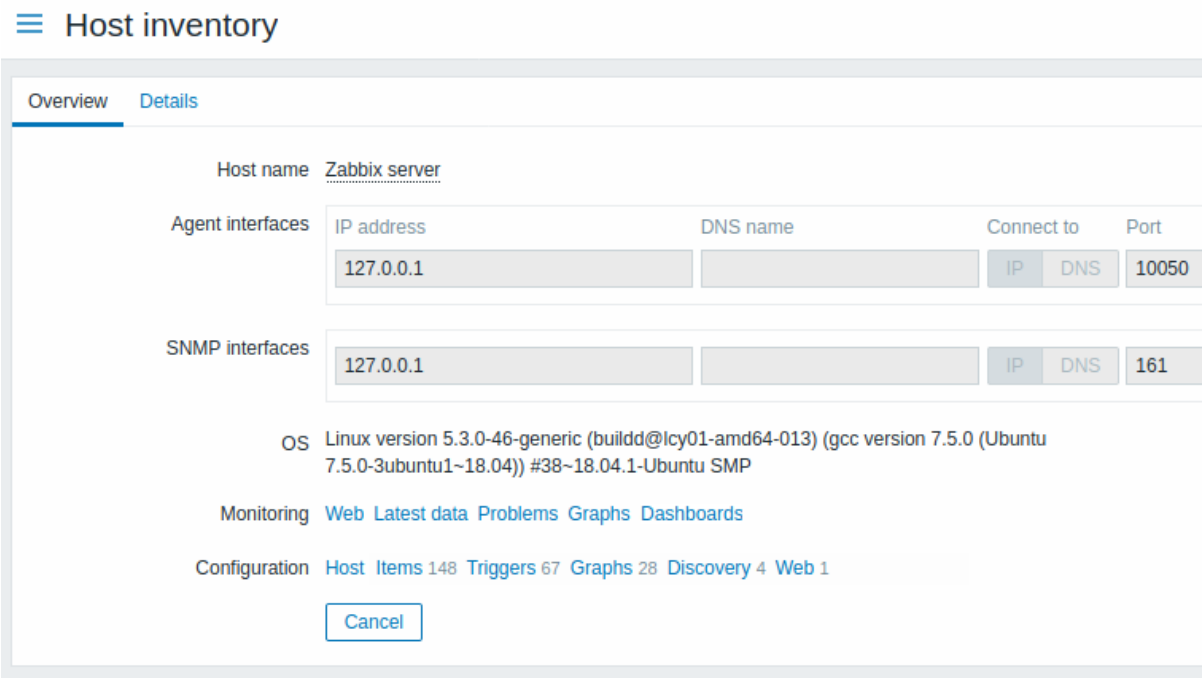


Para exibir todos os inventories de host, clique no botão *Reset*.

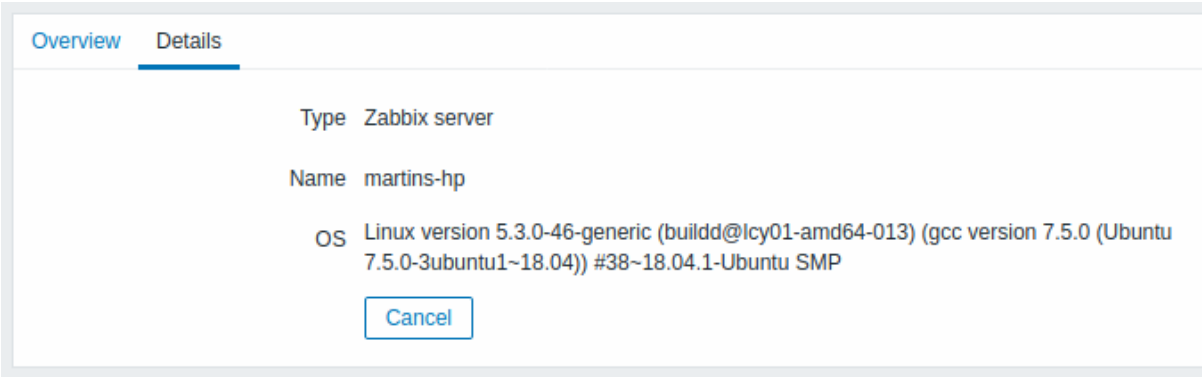
Embora apenas alguns campos principais de inventory sejam exibidos na tabela, você também pode visualizar todas as informações de inventory disponíveis para esse host. Para fazer isso, clique no nome do host na primeira coluna.

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.

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.

Informações do sistema

Visão geral

Em *Relatórios* > *Informações do sistema*, é exibido um resumo dos principais dados do Zabbix server e do sistema. Os dados do sistema são coletados usando itens internos.

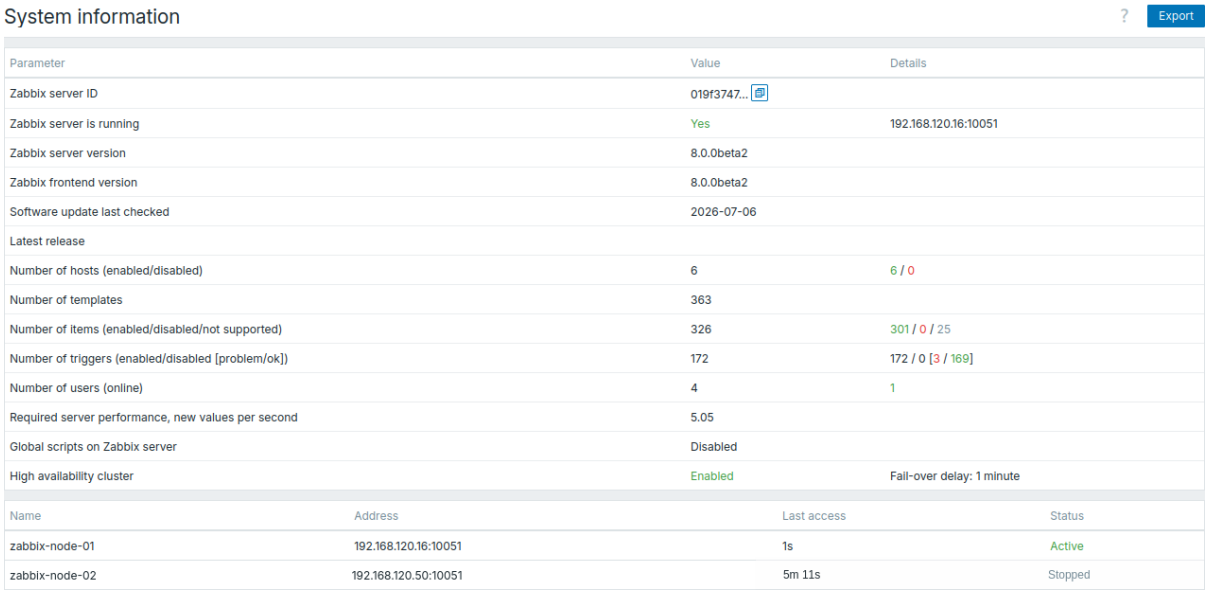
O relatório de informações do sistema fica visível somente para usuários *Super Admin* e *Admin* do Zabbix.

Observe que, em uma configuração de alta disponibilidade, é possível redirecionar a origem das informações do sistema (instância do server). Para fazer isso, edite o arquivo de configuração do frontend - remova o comentário e defina `$ZBX_SERVER` ou ambos `$ZBX_SERVER` e `$ZBX_SERVER_PORT` para um server diferente daquele mostrado como ativo. Observe que, ao definir somente `$ZBX_SERVER`, será usado um valor padrão (10051) para `$ZBX_SERVER_PORT`.

Com a configuração de alta disponibilidade ativada, um bloco separado é exibido abaixo das estatísticas do sistema, com detalhes dos nós de alta disponibilidade.

As *Informações do sistema* também estão disponíveis como um widget do dashboard.

Estatísticas do sistema



Dados exibidos:

Parâmetro	Valor	Detalhes
ID do Zabbix server	Identificador exclusivo do Zabbix server (UUIDv7).	O ID do server é gerado na primeira inicialização do server. Os primeiros 8 caracteres são exibidos, seguidos por reticências. Ao passar o cursor sobre ou clicar em um valor de ID do server, uma dica de ferramenta é exibida com o valor completo. Clicar no ícone ao lado do valor copia o ID para a área de transferência. Se a alta disponibilidade estiver habilitada, cada node terá seu próprio ID de node para endereçamento pela linha de comando, mas todos os nodes serão agrupados sob um único ID de server gerado pelo node que foi iniciado primeiro (ou que anteriormente era um server independente), independentemente de qual node esteja ativo no momento.
O Zabbix server está em execução	Status do Zabbix server: **Sim** - o server está em execução **Não** - o server não está em execução *Observação:* Para exibir o restante das informações, o web frontend precisa que o server esteja em execução e deve haver pelo menos um processo trapper iniciado no server (o 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. *Observação:* É exibido somente quando o Zabbix server está em execução.	O status da versão do server é exibido: **Atualizado** - a versão mais recente está em uso; **Nova atualização disponível** - uma versão mais recente está disponível; **Desatualizado** - o período completo de suporte para esta versão expirou. Essas informações estão disponíveis somente se a verificação de atualização de software estiver habilitada na configuração do server do Zabbix. Nada é exibido se a última verificação de atualização de software tiver sido realizada há mais de uma semana ou se não houver dados sobre a versão atual.

Parâmetro	Valor	Detalhes
Versão do Zabbix frontend	O número da versão do Zabbix frontend é exibido.	O status da versão do Zabbix frontend é exibido: **Atualizado** - a versão mais recente está em uso; **Nova atualização disponível** - uma versão mais recente está disponível; **Desatualizado** - o período completo de suporte para esta versão expirou. Essas informações estão disponíveis somente se a verificação de atualização de software estiver habilitada na configuração do server do Zabbix. Nada é exibido se a última verificação de atualização de software tiver sido realizada há mais de uma semana ou se não houver dados sobre a versão atual.
Última verificação de atualização de software	A data da última verificação de atualização do software do Zabbix é exibida. Essas informações estão disponíveis somente se a verificação de atualização de software estiver habilitada na configuração do server do Zabbix.	
Versão mais recente	O número de uma versão mais recente (se disponível) para a versão atual do Zabbix é exibido. Essas informações estão disponíveis somente se a verificação de atualização de software estiver habilitada na configuração do server do Zabbix. Nada é exibido se a última verificação de atualização de software tiver sido realizada há mais de uma semana ou se não houver dados sobre a versão atual.	Um link para as notas de versão da versão mais recente disponível do Zabbix é exibido.
Número de hosts	O número total de hosts configurados é exibido.	Número de hosts monitorados/não monitorados.
Número de templates	O número total de templates é exibido.	
Número de itens	O número total de itens é exibido.	Número de itens no nível do host monitorados/desabilitados/não suportados. Os itens em hosts desabilitados são contabilizados como desabilitados.
Número de triggers	O número total de triggers é exibido.	Número de triggers no nível do host habilitados/desabilitados; divisão dos triggers habilitados de acordo com os estados ”Problema”/”OK”. Os triggers listados no estado ”OK” incluem triggers com o status ”Desconhecido”. Os triggers dependentes de itens desabilitados ou atribuídos a hosts desabilitados são contabilizados como desabilitados.
Número de usuários	O número total de usuários configurados é exibido.	Número de usuários online.

Parâmetro	Valor	Detalhes
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 orientação. Para obter números precisos dos valores processados, use o item interno `zabbix[wcache,values,all]`. Itens habilitados de hosts monitorados são incluídos no cálculo. Itens de log são contabilizados como um valor por intervalo de atualização do item. Valores de intervalos regulares são contabilizados; valores de intervalos flexíveis e de agendamento não são. O cálculo não é ajustado durante um período de manutenção ”nodata”. Itens trapper não são contabilizados.
Scripts globais no Zabbix server	**Desabilitado** será exibido neste campo se os scripts globais estiverem desabilitados no Zabbix server pela configuração de EnableGlobalScripts=0 na configuração do server.	
Cluster de alta disponibilidade	Status do cluster de alta disponibilidade do Zabbix server: **Desabilitado** - server independente **Habilitado** - existe pelo menos um node de alta disponibilidade	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 a ordenação necessários (UTF-8).
- A versão do banco de dados está abaixo ou acima do intervalo suportado (disponível somente para usuários com o tipo de função *Super admin*).
- A manutenção do TimescaleDB está configurada incorretamente (as tabelas de histórico ou de tendências contêm blocos compactados, mas as opções *Substituir período do histórico do item* ou *Substituir período de tendências do item* estão desabilitadas).

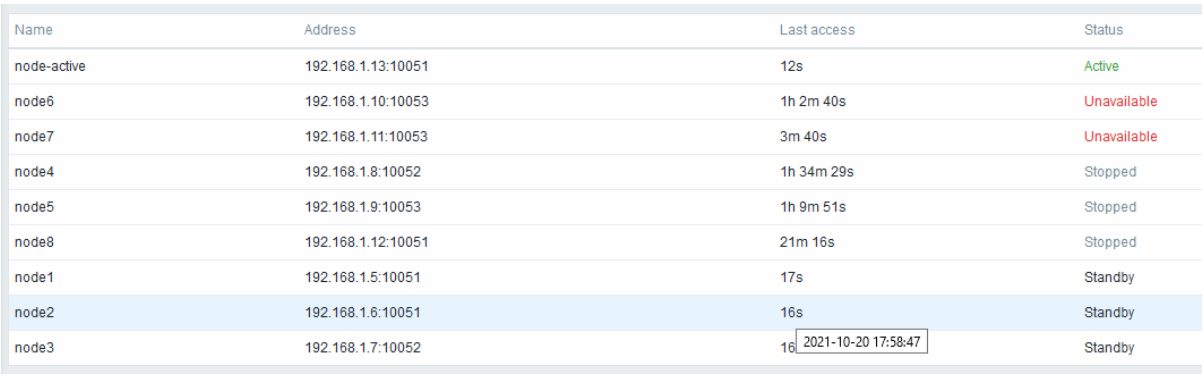
Botões

O botão no canto superior direito da página oferece a seguinte opção:

Export	Exportar informações do sistema para um arquivo JSON. O arquivo é salvo como `Zabbix-system-information-{time}.json`, em que `{time}` é o timestamp Unix que indica quando os dados foram preparados no server. Se a alta disponibilidade estiver habilitada, os detalhes dos nós de alta disponibilidade também serão exportados.

Nós de alta disponibilidade

Se o cluster de alta disponibilidade estiver habilitado, um bloco adicional de dados será exibido mostrando 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

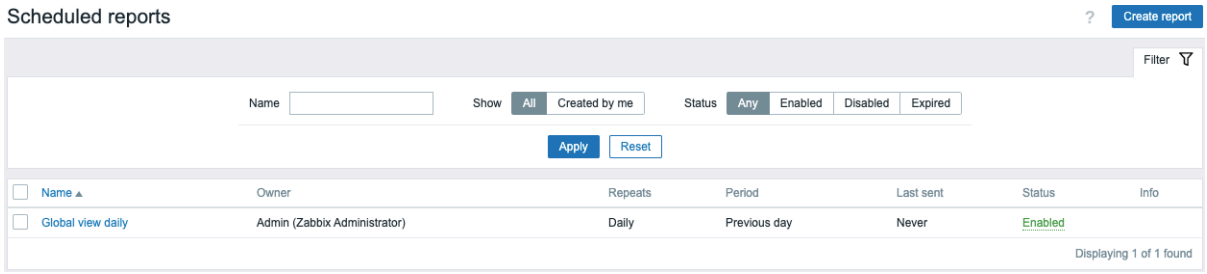
Dados exibidos:

Column	Description
Name	Nome do nó, conforme definido na configuração do server.
Address	Endereço IP e porta do nó.
Last access	Hora do último acesso ao nó. Ao passar o mouse sobre a célula, é exibido o carimbo de data e hora do último acesso em formato longo.
Status	Status do nó. As linhas da tabela são classificadas por esses status na seguinte ordem de prioridade: **Active** - o nó está ativo e funcionando **Unavailable** - o nó não foi visto por mais tempo do que o atraso de failover (talvez seja útil descobrir o motivo) **Stopped** - o nó foi interrompido ou não pôde iniciar (talvez seja útil iniciá-lo ou excluí-lo) **Standby** - o nó está ativo e aguardando

Relatórios agendados

Visão geral

Em *Relatórios* > *Relatórios agendados*, os usuários com permissões suficientes podem configurar a geração agendada de versões em PDF dos dashboards, que serão enviadas por e-mail aos destinatários especificados.



A tela inicial exibe informações sobre os relatórios agendados, que podem ser filtradas para facilitar a navegação - consulte a seção Usando o filtro abaixo.

Dados exibidos:

Coluna	Descrição
Nome	Nome do relatório. Clicar nele abre o formulário de configuração do relatório.
Proprietário	Usuário que criou o relatório.
Repetições	Frequência de geração do relatório (diária/semanal/mensal/anual).
Período	Período para o qual o relatório é preparado (dia, semana, mês ou ano anterior).
Último envio	Data e hora em que o relatório mais recente foi enviado.
Status	Status atual do relatório (habilitado/desabilitado/expirado). Usuários com permissões suficientes podem alterar o status clicando nele - de ”Habilitado” para ”Desabilitado” (e vice-versa); de ”Expirado” para ”Desabilitado” (e vice-versa). Para usuários sem direitos suficientes, o status não pode ser clicado.
Informações	Exibe ícones informativos: Um ícone vermelho indica que a geração do relatório falhou; ao passar o cursor sobre ele, uma dica de ferramenta será exibida com informações sobre o erro. Um ícone amarelo indica que um relatório foi gerado, mas o envio para alguns (ou todos) os destinatários falhou ou que um relatório está expirado; ao passar o cursor sobre ele, uma dica de ferramenta será exibida 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*)

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 resolvidos.

O filtro está localizado abaixo do nome da seção *Availability report*. Ele pode ser aberto e recolhido clicando na guia *Filter* 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.

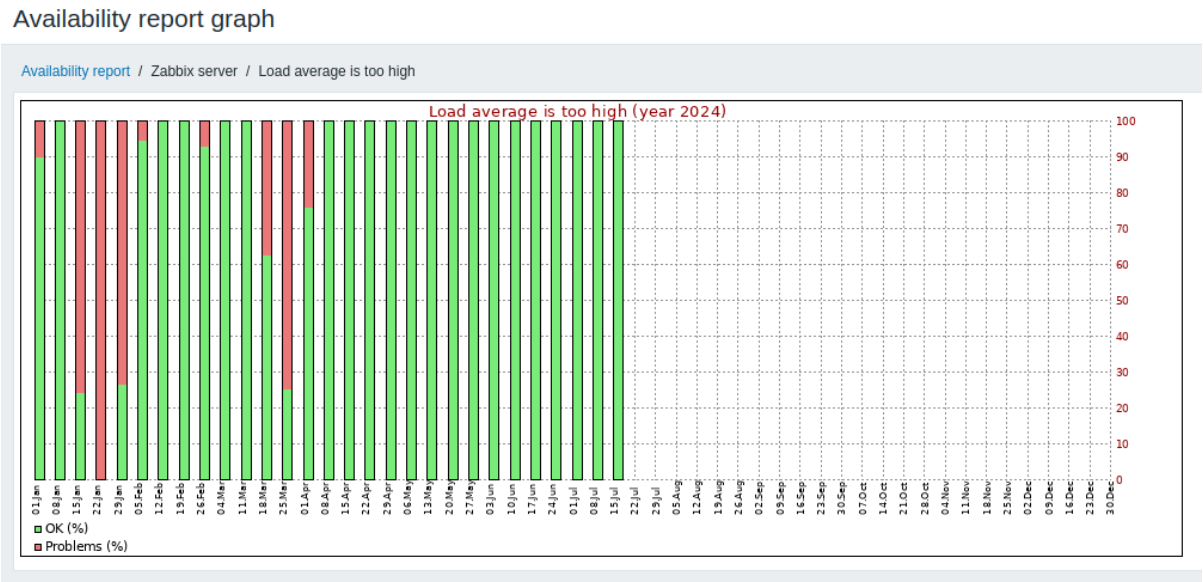
Filtrando por host

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 usados com frequência com um único clique. O seletor pode ser expandido e recolhido clicando na aba *Período de tempo* ao lado do filtro.

Clicar em *Mostrar* na coluna *Gráfico* exibe as informações de disponibilidade em um gráfico de barras, no qual cada barra representa uma semana transcorrida do ano atual.



A cor verde de uma barra representa o tempo em que tudo estava OK, e a cor vermelha representa o tempo em que houve problemas.

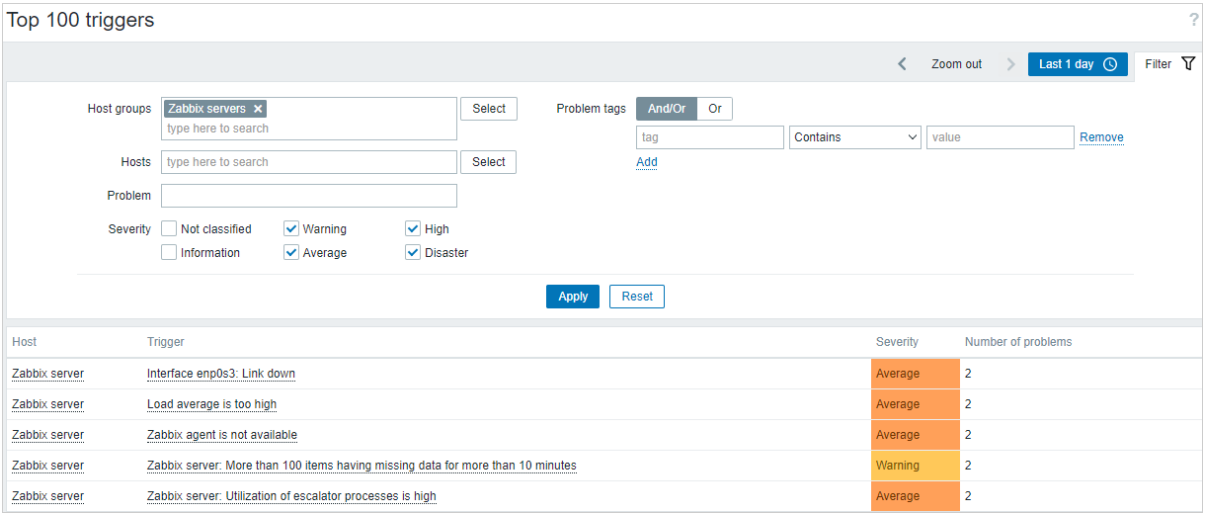
Efeito dos períodos de manutenção

Maintenance não exclui automaticamente o tempo do relatório de disponibilidade. Somente a maintenance configurada como *No data collection* interrompe a coleta (portanto, nenhum problema é gerado) e, assim, mantém a porcentagem de disponibilidade inalterada para os triggers afetados.

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.



As entradas das colunas de host e trigger são links que oferecem algumas opções úteis:

- para host - clicar no nome do host abre o menu do host
- para trigger - clicar no nome do trigger abre links para os eventos mais recentes, o gráfico simples de 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 melhorar o desempenho da pesquisa, os dados são pesquisados com as macros não resolvidas.

O filtro está localizado abaixo do nome da seção *Top 100 triggers*. Ele pode ser aberto e recolhido clicando na aba *Filtro* à direita.

Seletor de período

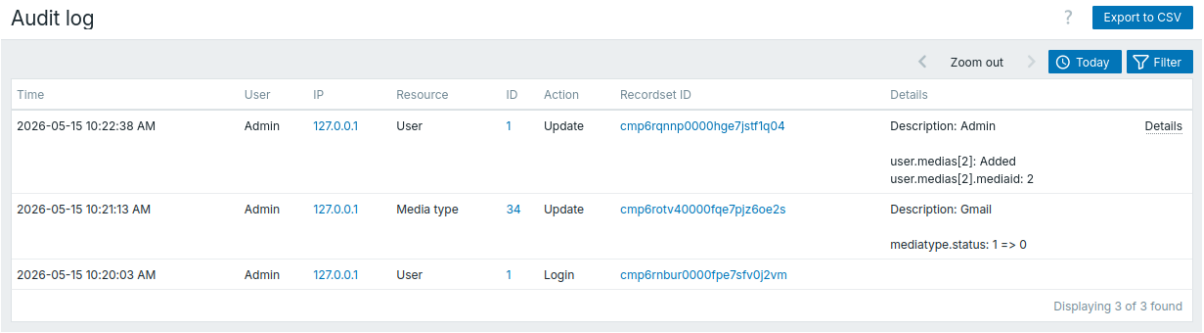
O seletor de período ao lado do filtro permite selecionar períodos frequentemente necessários com um clique do mouse. Para obter mais informações, consulte Seletores de período e host.

Log de auditoria

Visão geral

Na seção *Reports* > *Audit log*, os registros da atividade de usuários e do sistema podem ser visualizados.

Note:
Para que os registros de auditoria sejam coletados e exibidos, a caixa de seleção *Enable audit logging* deve estar marcada na seção *Administration* > *Audit log*. Sem essa configuração habilitada, o histórico de atividades não será gravado no banco de dados e não será exibido no audit log.



O audit log exibe os seguintes dados:

Column	Description
Time	Carimbo de data e hora do registro de auditoria.
User	Usuário que executou a atividade.
IP	IP de onde a atividade foi iniciada. Clicar no hiperlink resultará na filtragem dos registros do audit log por este IP.
Resource	Tipo do recurso afetado (*API token*, *Action*, *Authentication*, *Autoregistration*, etc.).
ID	ID do recurso afetado. Clicar no hiperlink resultará na filtragem dos registros do audit log por este ID de recurso.
Action	Tipo da atividade (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	ID compartilhado para todos os registros do audit log criados como resultado da mesma operação. Por exemplo, ao vincular um template a um host, um registro separado do audit log é criado para cada entidade de template herdada (item, trigger, etc.) - todos esses registros terão o mesmo *Recordset ID*. Clicar no hiperlink resultará na filtragem dos registros do audit log por este *Recordset ID*.
Details	Descrição do recurso e informações detalhadas sobre a atividade executada. Se um registro contiver mais de duas linhas, um link adicional *Details* será exibido. Clique nesse link para ver a lista completa de alterações.

Note:
Quando um trapper item ou um HTTP agent item (com trapping habilitado) recebe alguns dados, uma entrada no audit log será adicionada somente se os dados tiverem sido enviados usando o método de API `history.push`, e não o utilitário Zabbix sender.

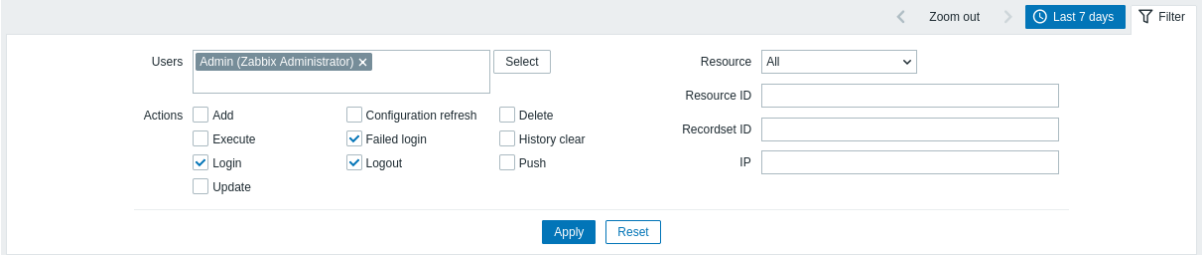
Botões

O botão no canto superior direito da página oferece a seguinte opção:

Export to CSV	Exporta os registros do log de auditoria de todas as páginas para um arquivo CSV. Se um filtro for aplicado, somente os registros filtrados serão exportados.

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.

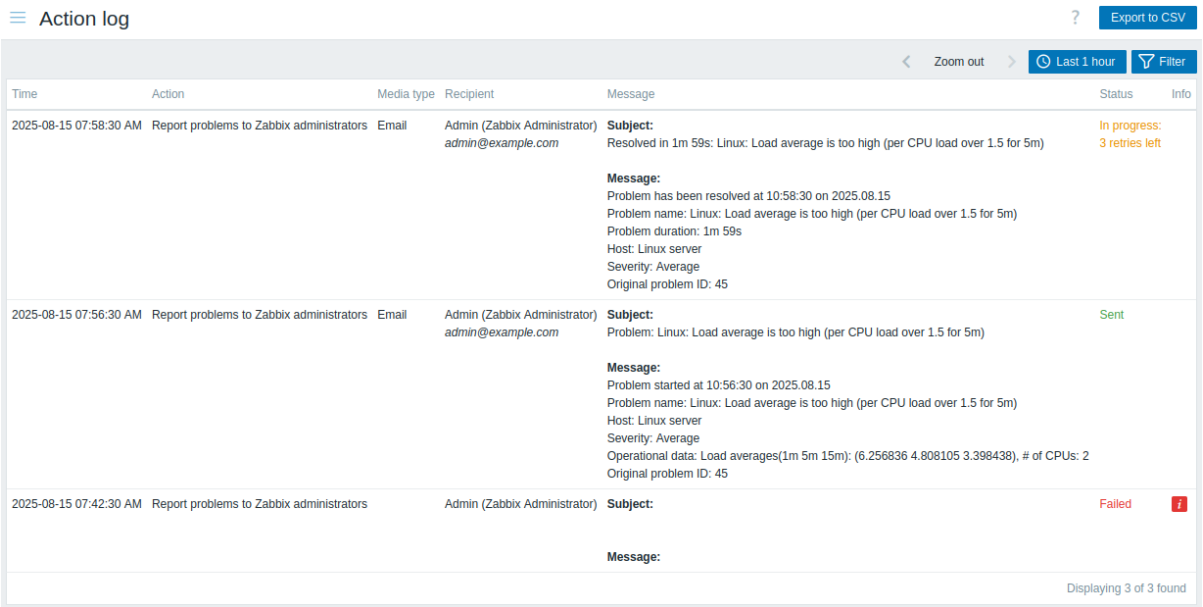
Seletor de período de tempo

O seletor de período de tempo ao lado do filtro permite selecionar períodos frequentemente necessários com um único clique do mouse. Para obter mais informações, consulte Seletores de período de tempo e host.

Log de ação

Visão geral

Em *Reports* > *Action log*, você pode visualizar detalhes de operations (notificações, comandos remotos) executadas dentro de uma action.



Dados exibidos:

Column	Description
Time	Carimbo de data e hora da operação.
Action	Nome da action que está causando operations.
Media type	Tipo de mídia (por exemplo, Email, Jira etc.) usado para enviar uma notificação. Para operations que executaram comandos remotos, esta coluna ficará vazia.
Recipient	Informações sobre o destinatário da notificação - nome de usuário, nome e sobrenome (entre parênteses) e informações adicionais dependendo do tipo de mídia (email, nome de usuário etc.). Para operations que executaram comandos remotos, esta coluna ficará vazia.
Message	O conteúdo da mensagem, do comando remoto ou do nome do script global. Um comando remoto é separado do host de destino por um símbolo de dois-pontos: `<host>:<command>`. Por exemplo, se o comando remoto foi executado no Zabbix server, então a informação terá o seguinte formato: `Zabbix server:<command>`.

Column	Description
Status	Status da operação: *In progress* - a operação de envio de uma notificação está em andamento (o número restante de vezes que o server tentará enviar a notificação também é exibido) *Sent* - a notificação foi enviada *Executed* - o comando remoto foi executado *Failed* - a operação não foi concluída
Info	Informações de erro (se houver) relacionadas à execução da operação.

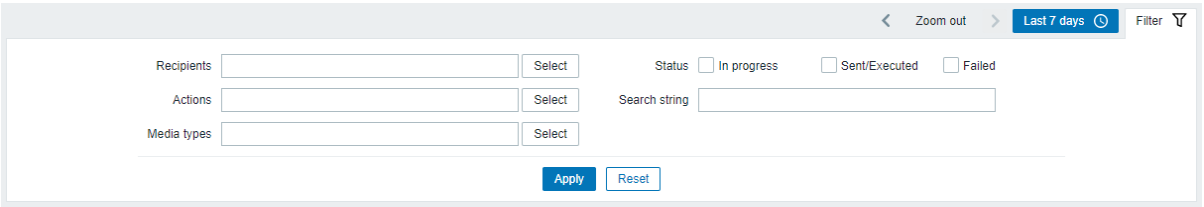
Botões

O botão no canto superior direito da página oferece a seguinte opção:

Export to CSV	Exporta os registros do log de ações da página atual para um arquivo CSV. Se um filtro for aplicado, apenas os registros filtrados serão exportados. No arquivo CSV exportado, as colunas ”Destinatário” e ”Mensagem” são divididas em várias colunas - ”Nome de usuário Zabbix do destinatário”, ”Nome do destinatário”, ”Sobrenome do destinatário”, ”Destinatário” e ”Assunto”, ”Mensagem”, ”Comando”.

Usando o filtro

O filtro está localizado abaixo da barra *Action log*. Ele pode ser aberto e recolhido clicando na guia *Filter* 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 (*Search string*). Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

Seletor de período de tempo

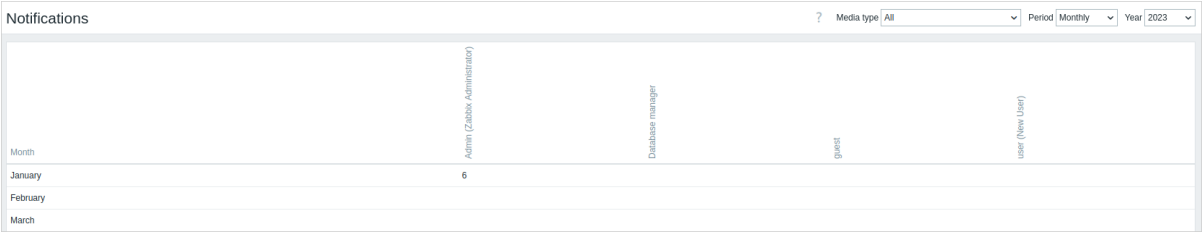
O seletor de período de tempo ao lado do filtro permite selecionar períodos frequentemente necessários com um único clique do mouse. Para obter mais informações, consulte Seletores de período de tempo e host.

Notificações

Visão geral

Na seção *Reports* > *Notifications*, 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 das notificações enviadas.



Cada coluna exibe os totais por um usuário do sistema.

Coleta de dados

Visão geral

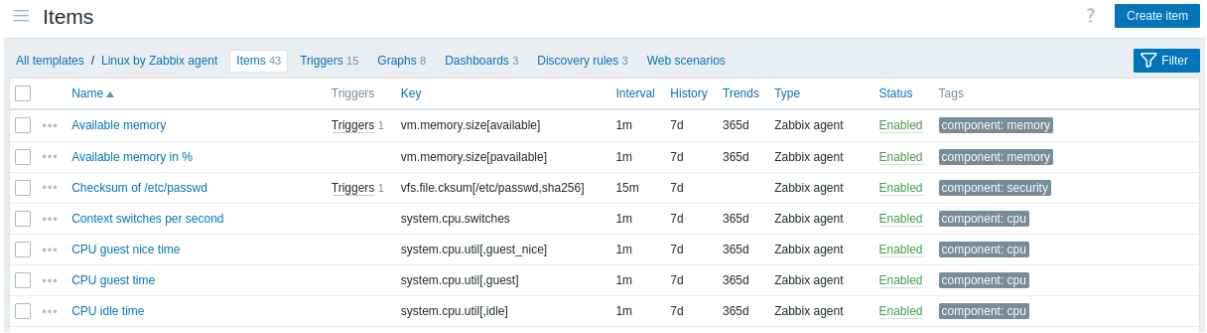
Este menu apresenta seções relacionadas à configuração da coleta de dados.

Itens

Visão geral

A lista de itens de um template pode ser acessada em *Coleta de dados* > *Templates*, clicando em *Itens* para o template correspondente.

Uma lista dos itens existentes é exibida.



Dados exibidos:

Coluna	Descrição
Menu do item	Clique no ícone de três pontos para abrir o menu do item.
Template	Template ao qual o item pertence. Clicar no nome do template abre o formulário de configuração do template. Esta coluna é exibida somente quando vários templates ou nenhum template estão 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 será exibido antes do nome do item como um link cinza. Clicar no link do template abre a lista de itens nesse nível do template.
Triggers	Ao passar o mouse sobre Triggers, uma caixa de informações com os triggers associados ao item será exibida. O número de triggers é exibido em cinza.
Chave	A chave do item é exibida.
Intervalo	A frequência da verificação é exibida.
Histórico	É exibido por quantos dias o histórico de dados do item será mantido.
Tendências	É exibido por quantos dias o histórico de tendências do item será mantido.
Tipo	O tipo de item é exibido (Zabbix agent, SNMP agent, verificação simples etc.).
Status	O status do item é exibido: *Habilitado* ou *Desabilitado*. Ao clicar 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, será exibido um link "..." que permite visualizar todas as tags ao passar o mouse sobre ele.

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:

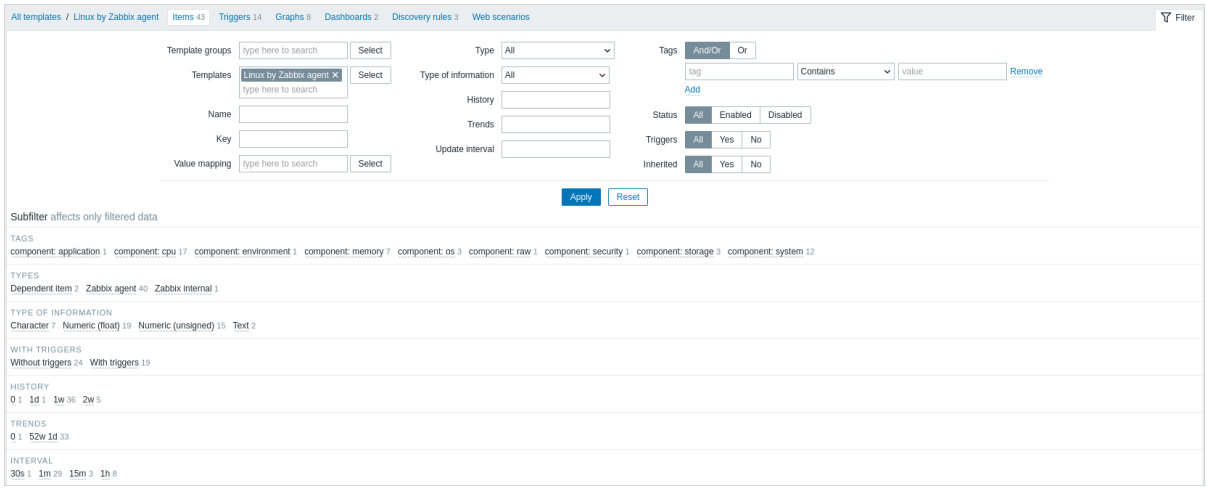
- *Enable* - altera o status do item para *Enabled*.
- *Disable* - altera o status do item para *Disabled*.
- *Copy* - copia os itens para outros hosts ou templates.
- *Mass update* - atualiza várias propriedades de vários itens de uma vez.
- *Delete* - exclui os itens.

Para usar essas opções, marque as caixas de seleção antes dos respectivos itens e, em seguida, 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 *Filter* está disponível no canto superior direito. Clicar nele abrirá um filtro onde você pode especificar os critérios de filtragem desejados.

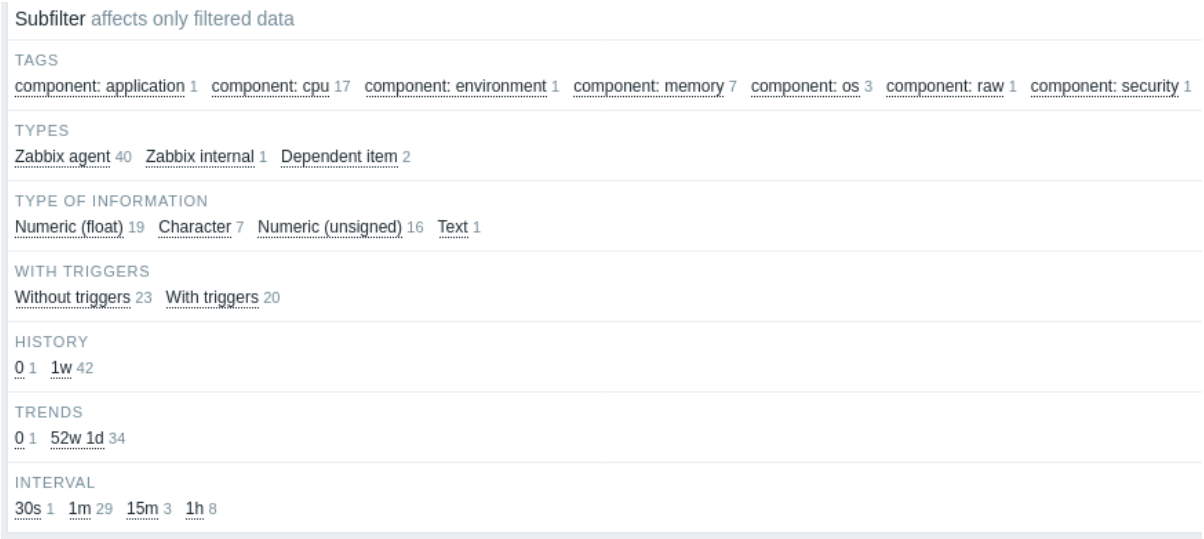


Parameter	Description
Template groups	Filtrar por um ou mais grupos de template. Especificar um grupo de template pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.
Name	Filtrar pelo nome do item.
Key	Filtrar pela chave do item.
Value mapping	Filtrar pelo mapeamento de valor usado. Este parâmetro não é exibido se a opção *Templates* estiver vazia.
Type	Filtrar pelo tipo do item (Zabbix agent, SNMP agent, etc.).
Type of information	Filtrar pelo tipo de informação (Unsigned numérico, float, etc.).
History	Filtrar por quanto tempo o histórico do item é mantido.
Trends	Filtrar por quanto tempo as tendências do item são mantidas.
Update interval	Filtrar pelo intervalo de atualização do item.
Tags	Especifique tags para limitar o número de items exibidos. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida
Status	Filtrar pelo status do item - *Enabled* ou *Disabled*.
Triggers	Filtrar items com (ou sem) triggers.
Inherited	Filtrar items herdados (ou não herdados) de templates vinculados.

Usando o subfiltro

O subfiltro permite modificar ainda mais a filtragem aplicada pelo filtro principal.

Ele contém links clicáveis para acesso rápido a itens relacionados. Os itens são relacionados por uma entidade comum, como 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 filtrados imediatamente (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outra entidade adiciona essa entidade aos resultados filtrados. Clicar novamente na entidade remove a filtragem.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se quiser ver mais registros no subfiltro, aumente 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 exibir informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

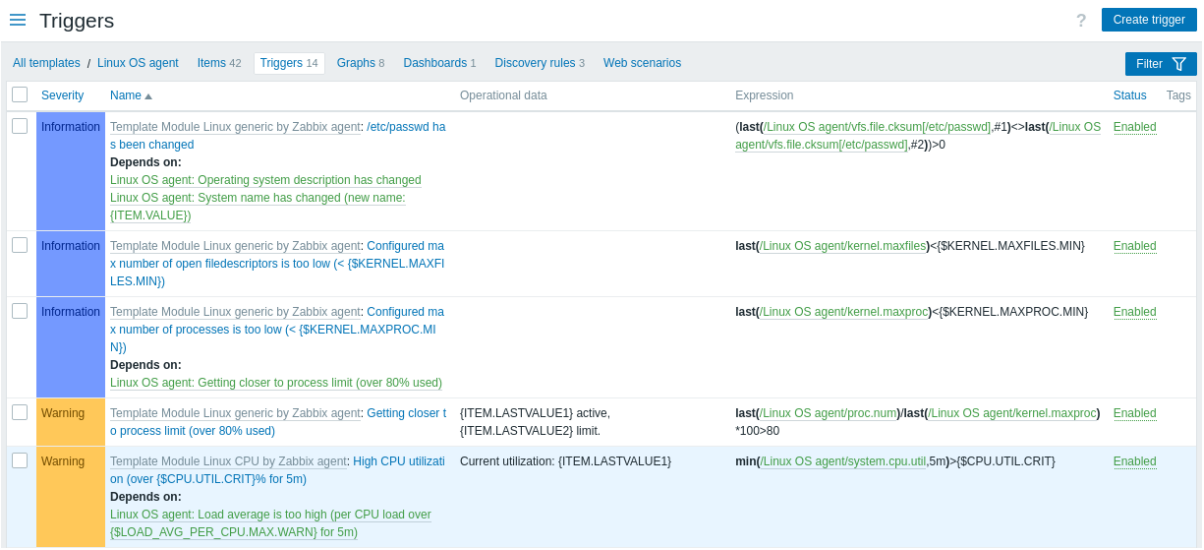
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais entidades, um ícone de três pontos será exibido no final; ele não é clicável.

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 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.

Triggers

Visão geral

A lista de triggers de um template pode ser acessada em *Coleta de dados* > *Templates*, clicando em *Triggers* para o template correspondente.



Dados exibidos:

Coluna	Descrição
Severidade	A severidade do trigger é exibida pelo nome e 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 somente quando vários templates ou nenhum template estão 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 será exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de triggers nesse nível do template.
Dados operacionais	Definição dos dados operacionais do trigger, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* > *Problemas*.
Expressão	A expressão do trigger é exibida. A parte da expressão referente ao item do template é exibida como um link que leva ao formulário de configuração do item.
Status	O status do trigger é exibido: *Habilitado* ou *Desabilitado*. Ao clicar 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:

- *Enable* - altera o status do trigger para *Enabled*
- *Disable* - altera o status do trigger para *Disabled*
- *Copy* - copia os triggers para outros hosts ou templates
- *Mass update* - atualiza várias propriedades de vários triggers de uma vez
- *Delete* - exclui os triggers

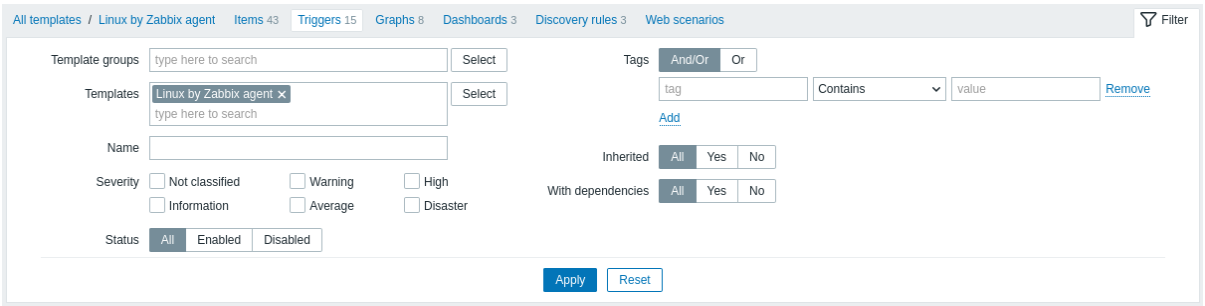
Para usar essas opções, marque as caixas de seleção antes dos respectivos triggers e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os triggers de seu interesse.

Para um melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filter* está disponível no canto superior direito. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados.



Parameter	Description
Template groups	Filtrar por um ou mais grupos de template. Especificar um grupo de template pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates. Se grupos de template já tiverem sido selecionados acima, a seleção de templates fica limitada a esses grupos.
Name	Filtrar pelo nome do trigger.
Severity	Selecione para filtrar por uma ou várias severidades de trigger.
Status	Filtrar pelo status de ativação do trigger (Enabled/Disabled).

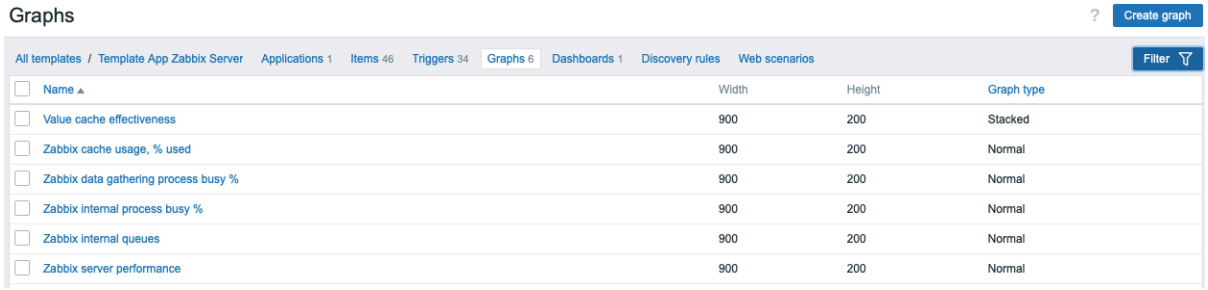
Parameter	Description
Tags	Filtrar pelo nome e valor da tag do trigger. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (case-sensitive) **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (case-sensitive) **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida Macros e funções de macro são suportadas nos campos de nome e valor da tag.
Inherited	Filtrar triggers herdados (ou não herdados) de templates vinculados.
With dependencies	Filtrar triggers com (ou sem) dependências.

Gráficos

Visão geral

A lista de gráficos personalizados de um template pode ser acessada em *Coleta de dados* > *Templates*, clicando em *Gráficos* para o template correspondente.

Uma lista dos 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 somente quando vários templates ou nenhum template estão selecionados no filtro.
Nome	Nome do gráfico personalizado, exibido como um link azul para os detalhes do gráfico. Clicar no link com o nome do gráfico abre o formulário de configuração do gráfico. Se o gráfico for herdado de outro template, o nome do template será exibido antes do nome do gráfico como um link cinza. Clicar no link do template abre a lista de gráficos nesse nível do 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:

- *Copy* - copiar os gráficos para outros hosts ou templates
- *Delete* - excluir os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos, depois clique no botão desejado.

Usando o filtro

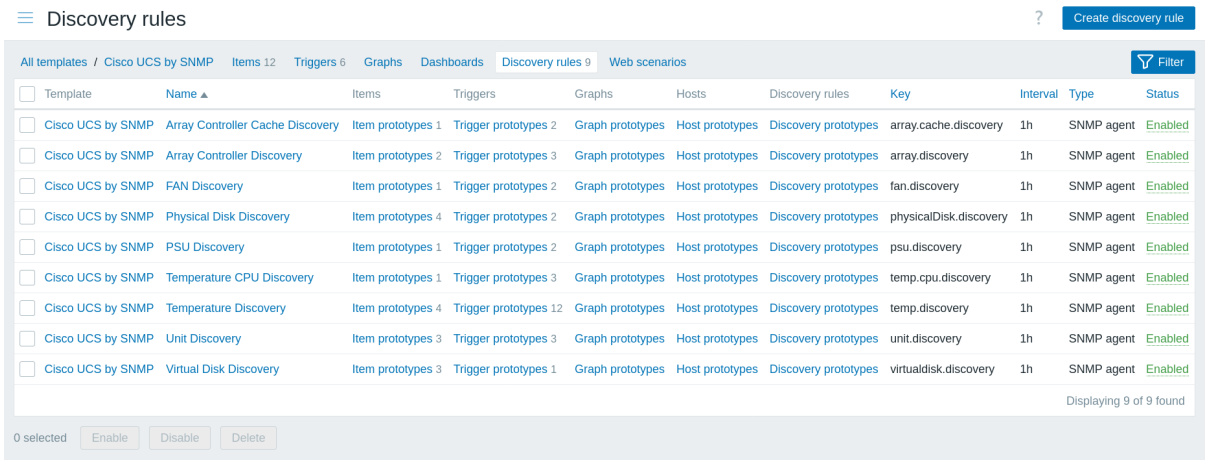
Você pode filtrar gráficos por grupo de template e template. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Regras de descoberta

Visão geral

A lista de regras de descoberta de baixo nível de um template pode ser acessada em *Coleta de dados* > *Templates*, clicando em *Descoberta* para o template correspondente.

Uma lista das regras de descoberta de baixo nível existentes é exibida. Também é possível visualizar 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 nesse nível do template.
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.
Chave	É exibida a chave do item usada para a descoberta.
Intervalo	É exibida a frequência de execução da descoberta.
Tipo	É exibido o tipo de item usado para a descoberta (Zabbix agent, SNMP agent etc.).
Status	O status da regra de descoberta é exibido: *Habilitado* ou *Desabilitado*. Ao clicar no status, é possível 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:

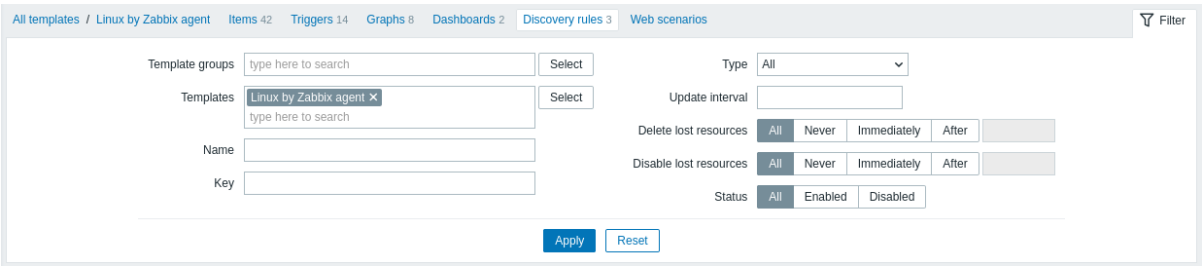
- *Enable* - altera o status da regra de descoberta de baixo nível para *Enabled*
- *Disable* - altera o status da regra de descoberta de baixo nível para *Disabled*
- *Delete* - exclui as regras de descoberta de baixo nível

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de descoberta e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta de seu interesse. Para melhor desempenho da busca, os dados são pesquisados com macros não resolvidas.

O ícone *Filter* está disponível no canto superior direito. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados, como template, nome da regra de descoberta, chave do item, tipo do item etc.



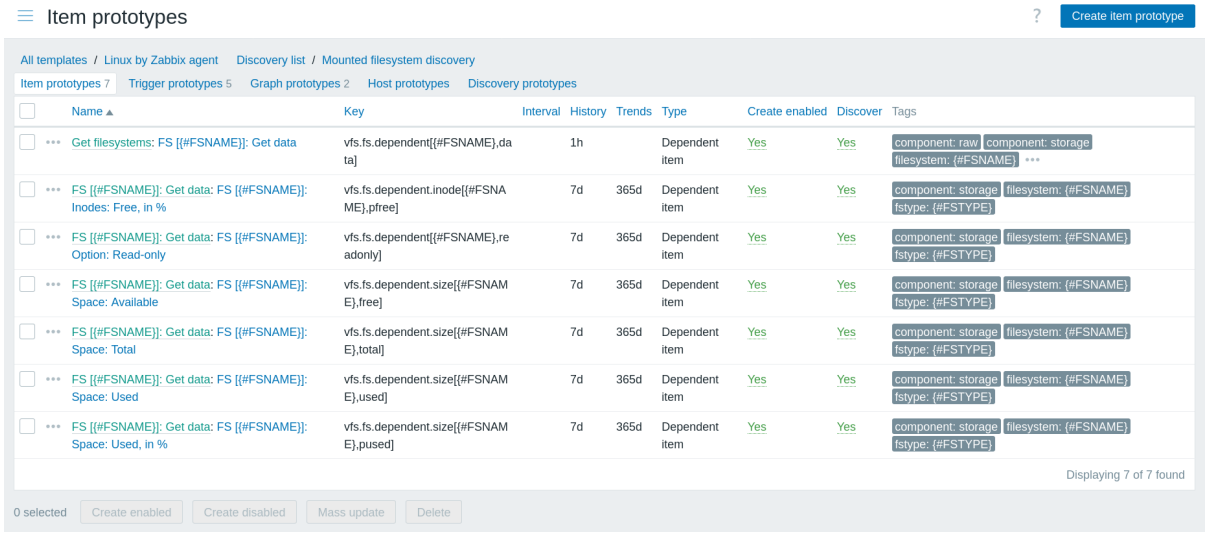
Parameter	Description
Template groups	Filtrar por um ou mais grupos de template. Especificar um grupo de template pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.
Name	Filtrar pelo nome da regra de descoberta.
Key	Filtrar pela chave do item de descoberta.
Type	Filtrar pelo tipo do item de descoberta.
Update interval	Filtrar pelo intervalo de atualização. Não disponível para Zabbix trapper e itens dependentes.
Delete lost resources	Filtrar pelo período de *Delete lost resources*.
Disable lost resources	Filtrar pelo período de *Disable lost resources*.
Status	Filtrar pelo status da regra de descoberta (All/Enabled/Disabled).

Protótipos de item

Visão geral

Nesta seção, são exibidos os protótipos de item configurados de uma regra de descoberta de baixo nível no template.

Se o template estiver vinculado ao host, os protótipos de item serão a base para a criação de itens reais no host durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Name	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 ”Dependent item”, 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).
Key	A chave do protótipo de item é exibida.
Interval	A frequência da verificação é exibida.
History	É exibido por quantos dias o histórico de dados do item será mantido.
Trends	É exibido por quantos dias o histórico de tendências do item será mantido.
Type	O tipo do protótipo de item é exibido (Zabbix agent, SNMP agent, verificação simples etc.).
Create enabled	Cria o item com base neste protótipo como: **Yes** - habilitado **No** - desabilitado. Você pode alternar entre ’Yes’ e ’No’ clicando neles.
Discover	Descobre o item com base neste protótipo: **Yes** - descobrir **No** - não descobrir. Você pode alternar entre ’Yes’ e ’No’ clicando neles.
Tags	As tags do protótipo de item são exibidas.

Para configurar um novo protótipo de item, clique no botão *Create item prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Create enabled* - criar estes items como *Enabled*
- *Create disabled* - criar estes items como *Disabled*
- *Mass update* - atualizar em massa estes protótipos de item
- *Delete* - excluir estes protótipos de item

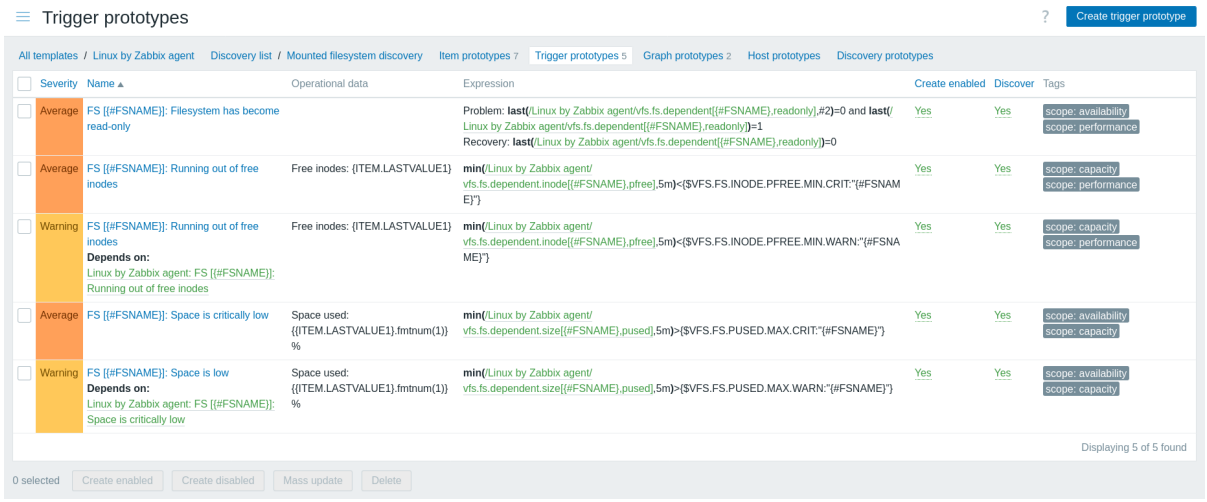
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de item e, em seguida, clique no botão desejado.

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 servirão de base para a criação de triggers reais no host durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Severity	A severidade do trigger é exibida pelo nome e pela cor de fundo da célula.
Name	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.
Expression	A expressão do trigger é exibida. A parte da expressão referente ao item do template é exibida como um link que leva ao formulário de configuração do item.
Operational data	É exibido o formato dos dados operacionais do trigger, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoring* > *Problems*.
Create enabled	Criar o trigger com base neste protótipo como: **Yes** - habilitado **No** - desabilitado. Você pode alternar entre 'Yes' e 'No' clicando neles.
Discover	Descobrir o trigger com base neste protótipo: **Yes** - descobrir **No** - não descobrir. Você pode alternar entre 'Yes' e 'No' clicando neles.
Tags	As tags do protótipo de trigger são exibidas.

Para configurar um novo protótipo de trigger, clique no botão *Create trigger prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria estes triggers como *Habilitado*
- *Criar desabilitado* - cria estes triggers como *Desabilitado*
- *Atualização em massa* - atualiza em massa estes protótipos de trigger
- *Excluir* - exclui estes protótipos de trigger

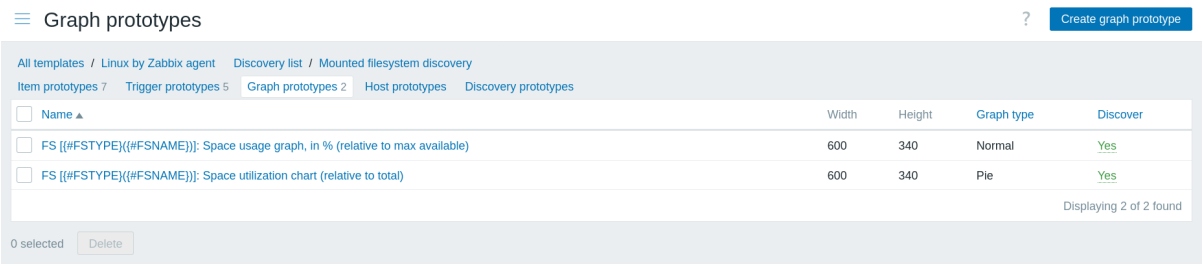
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de trigger e, em seguida, clique no botão desejado.

Protótipos de gráfico

Visão geral

Nesta seção, são exibidos os protótipos de gráfico configurados de uma regra de descoberta de baixo nível no template.

Se o template estiver vinculado ao host, os protótipos de gráfico servirão de base para a criação de gráficos reais do host durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Name	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 gráfico no nível do template vinculado.
Width	A largura do protótipo de gráfico é exibida.
Height	A altura do protótipo de gráfico é exibida.
Type	O tipo do protótipo de gráfico é exibido: *Normal*, *Stacked*, *Pie* ou *Exploded*.
Discover	Descobrir o gráfico com base neste protótipo: **Yes** - descobrir **No** - não descobrir. Você pode alternar entre 'Yes' e 'No' clicando nessas opções.

Para configurar um novo protótipo de gráfico, clique no botão *Create graph prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Delete* - exclui estes protótipos de gráfico

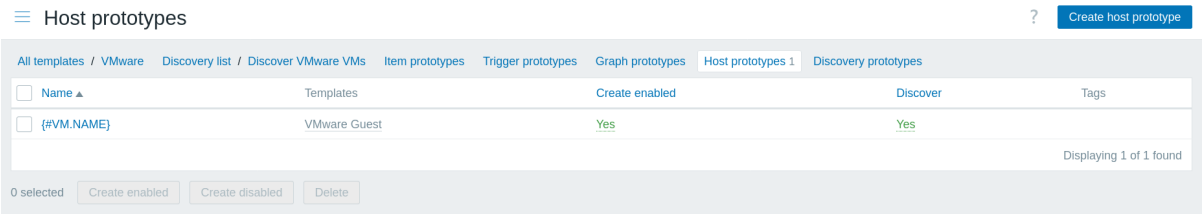
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráfico e, em seguida, clique no botão desejado.

Protótipos de host

Visão geral

Nesta seção são exibidos os protótipos de host configurados de uma regra de descoberta de baixo nível no template.

Se o template estiver vinculado ao host, os protótipos de host serão a base para a criação de hosts reais durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Name	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	São exibidos os templates do protótipo de host.
Create enabled	Criar o host com base neste protótipo como: **Yes** - habilitado **No** - desabilitado. Você pode alternar entre 'Yes' e 'No' clicando neles.

Column	Description
Discover	Descobrir o host com base neste protótipo: **Yes** - descobrir **No** - não descobrir. Você pode alternar entre 'Yes' e 'No' clicando neles.
Tags	São exibidas as tags do protótipo de host.

Para configurar um novo protótipo de host, clique no botão *Create host prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria estes hosts como *Habilitado*
- *Criar desabilitado* - cria estes hosts como *Desabilitado*
- *Excluir* - exclui estes protótipos de host

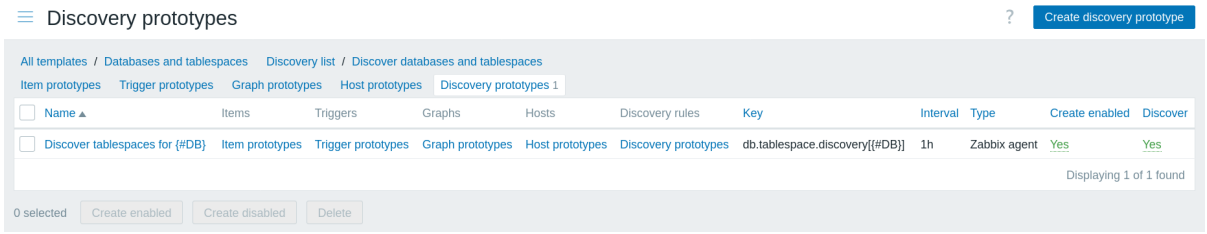
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e, em seguida, clique no botão desejado.

Protótipos de descoberta

Visão geral

Nesta seção, são exibidos os protótipos de descoberta configurados de uma regra de descoberta de baixo nível no template.

Se o template estiver vinculado ao host, os protótipos de descoberta servirão de 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	É exibida a chave do item usada para a descoberta.
Intervalo	É exibida a frequência de execução da descoberta.
Tipo	É exibido o tipo de item usado para a descoberta (Zabbix agent, SNMP agent etc.).
Criação habilitada	É exibido o status de criação do protótipo de descoberta: criação habilitada (*Sim*) ou criação desabilitada (*Não*). Ao clicar no status, é possível alterá-lo de Sim para Não (e vice-versa).
Descobrir	É exibido o status de descoberta do protótipo de descoberta: descobrir (*Sim*) ou não descobrir (*Não*). Ao clicar no status, é possível 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* - cria estes protótipos de descoberta como *Habilitado*
- *Criar desabilitado* - cria estes protótipos de descoberta como *Desabilitado*
- *Excluir* - exclui estes protótipos de descoberta

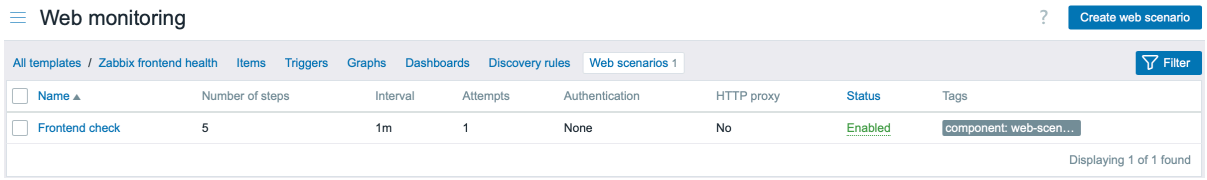
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de descoberta e, em seguida, clique no botão desejado.

Cenários web

Visão geral

A lista de cenários web de um template pode ser acessada em *Coleta de dados* > *Templates*, clicando em *Web* para o template correspondente.

Uma lista dos 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 será exibido antes do nome do cenário web como um link cinza. Clicar no link do template abre a lista de cenários web nesse nível do template.
Número de etapas	O número de etapas que o cenário contém.
Intervalo de atualização	Com que frequência o cenário é executado.
Tentativas	Quantas tentativas de execução das etapas do cenário web são realizadas.
Autenticação	O método de autenticação é exibido: Basic, NTLM, Kerberos, Digest ou None.
Proxy HTTP	Exibe o proxy HTTP ou "Não" se não for utilizado.
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, será exibido um link "..." que permite visualizar todas as tags ao passar o mouse sobre ele.

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:

- *Enable* - altera o status do cenário para *Enabled*
- *Disable* - altera o status do cenário para *Disabled*
- *Delete* - exclui os web scenarios

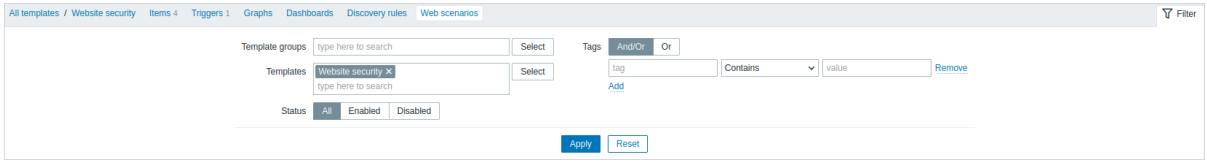
Para usar essas opções, marque as caixas de seleção antes dos respectivos web scenarios e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os cenários nos quais você está interessado.

Para um melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de cenários da web. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar cenários por grupo de template, template, status e tags.

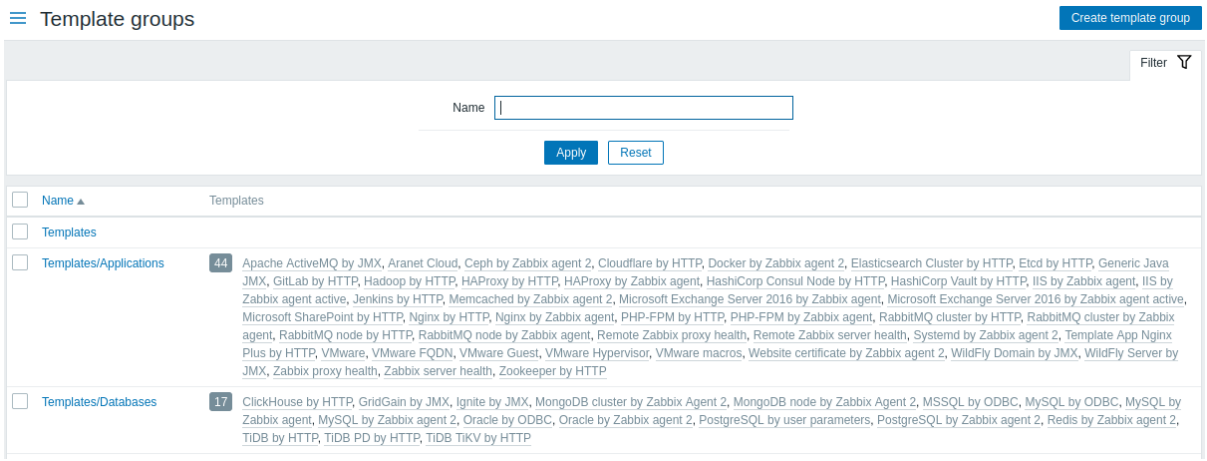


Grupos de template

Visão geral

Na seção *Coleta de dados* > *Grupos de templates*, os usuários podem configurar e manter grupos de templates.

É exibida uma lista dos grupos de templates existentes com seus detalhes. Você pode pesquisar e filtrar grupos de templates por nome.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de templates. Clicar no nome do grupo abre o formulário de configuração do grupo de templates.
Templates	Número de templates no grupo (exibido em cinza), seguido pela lista de membros do grupo. Clicar no nome de um template abre o formulário de configuração do template. Clicar no número abre a lista de templates neste grupo.

Opções de edição em massa

Para excluir vários grupos de template de uma vez, marque as caixas de seleção antes dos respectivos grupos e, em seguida, clique no botão Delete abaixo da lista.

Usando o filtro

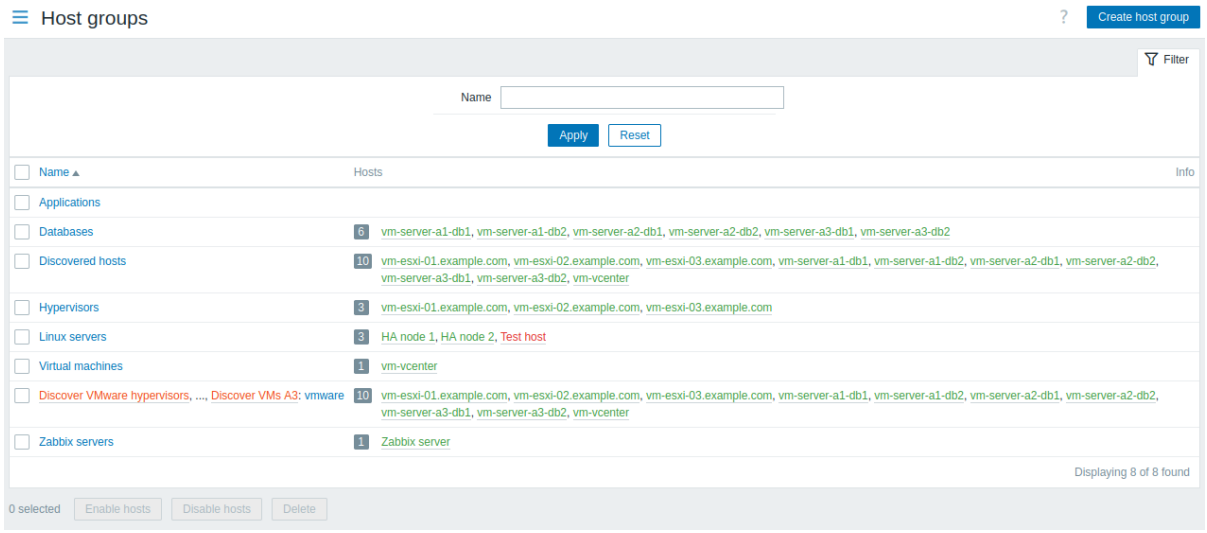
Você pode usar o filtro para exibir apenas os grupos de templates que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Grupos de hosts

Visão geral

Na seção *Coleta de dados* > *Grupos de hosts*, os usuários podem configurar e manter grupos de hosts.

É exibida uma lista dos grupos de hosts existentes com seus detalhes. Você pode pesquisar e filtrar grupos de hosts por nome.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de hosts. Clicar no nome do grupo abre o formulário de configuração do grupo de hosts. Os 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 os 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 de um host abre o formulário de configuração do host. Clicar no número filtra, na lista completa de hosts, aqueles que pertencem ao grupo.
Informações	As 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:

- *Enable hosts* - altera o status de todos os hosts no grupo para "Monitored"
- *Disable hosts* - altera o status de todos os hosts no grupo para "Not monitored"
- *Delete* - exclui os grupos de hosts

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de hosts e, em seguida, clique no botão desejado.

Usando o filtro

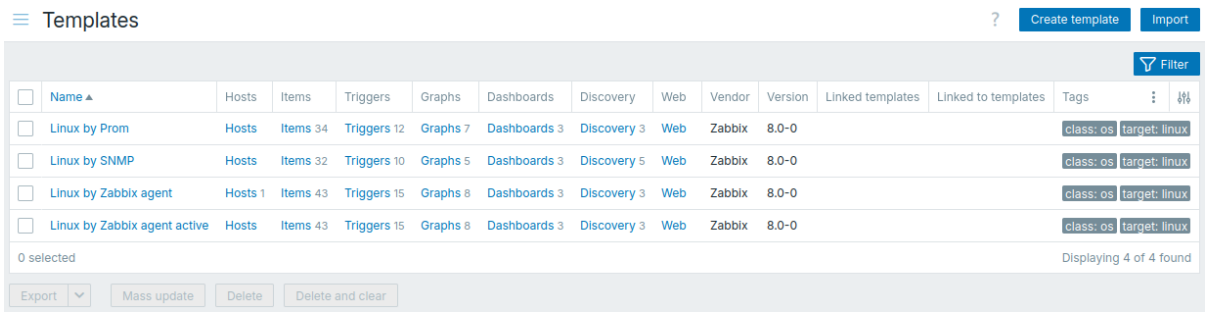
Você pode usar o filtro para exibir apenas os grupos de hosts que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Templates

Visão geral

Na seção *Coleta de dados* > *Templates*, os usuários podem configurar e manter templates.

É exibida uma lista dos templates existentes com seus detalhes.



Os detalhes do template são exibidos em uma tabela personalizável. O botão *Personalizar tabela*, no cabeçalho da tabela, permite mostrar, ocultar ou reordenar colunas, além de redefinir o layout para o padrão. Também é possível redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefinir o tamanho. Todas as personalizações são aplicadas imediatamente e sempre são lembradas para o usuário atual.

As seguintes colunas estão disponíveis:

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 são incluídos. Clicar em *Hosts* abre a lista de hosts filtrada para exibir somente os hosts vinculados ao template.
Entidades (Items, Triggers, Gráficos, Dashboards, Descoberta, Web)	Número das respectivas entidades no template (exibido em cinza). Clicar no nome da entidade filtra, na lista completa dessa entidade, 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; exibidos se a configuração do template contiver essas informações, e somente para templates prontos para uso, templates importados ou templates modificados por meio da API de Template. Para templates prontos para uso, a versão é exibida da seguinte forma: versão principal do Zabbix, delimitador ("-"), número da revisão (incrementado 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.
Tags	As tags do template são exibidas, com as macros não resolvidas. Passar o cursor sobre ou clicar no ícone de três pontos após as tags exibidas abre uma janela pop-up com todas as tags. Clicar no botão *Opções*, no cabeçalho da coluna, permite definir o número de tags exibidas (1–3), escolher como os nomes das tags serão exibidos (completos, abreviados ou nenhum), definir a prioridade de exibição das tags ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas).
Valor da tag	Os valores das tags do template são exibidos, com as macros não resolvidas. Clicar no botão *Opções*, no cabeçalho da coluna, permite renomear a coluna, escolher quais valores de tags exibir ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas).
Texto personalizado	É exibido texto personalizado, valores de macros ou uma combinação de ambos. Clicar no botão *Opções*, no cabeçalho da coluna, permite renomear a coluna, inserir o texto personalizado ou as macros, ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas). Macros compatíveis: `{HOST.ID}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.DESCRIPTION}` e macros de usuário.

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.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Export* - exporta o template para um arquivo YAML, XML ou JSON;
- *Mass update* - atualiza várias propriedades de vários templates de uma vez;
- *Delete* - exclui o template, mantendo suas entidades vinculadas (items, triggers etc.) nos hosts;
- *Delete and clear* - exclui o template e suas entidades vinculadas dos hosts.

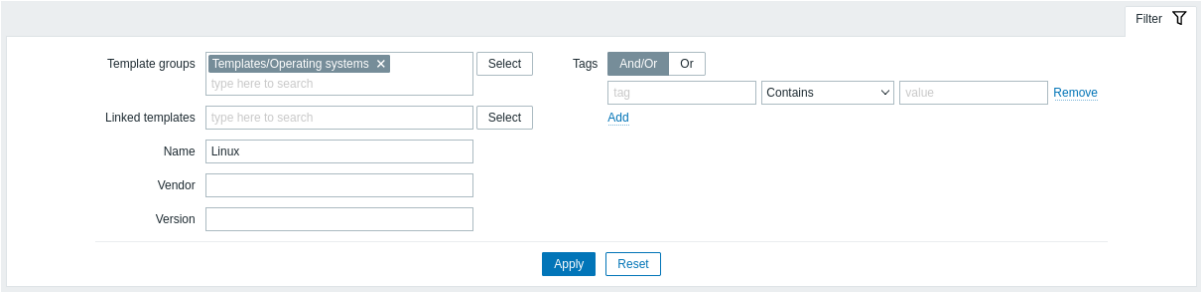
Para usar essas opções, marque as caixas de seleção antes dos respectivos templates e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os templates de seu interesse.

Para um melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível abaixo dos botões *Create template* e *Import*. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar templates por grupo de template, templates vinculados, nome e tags.



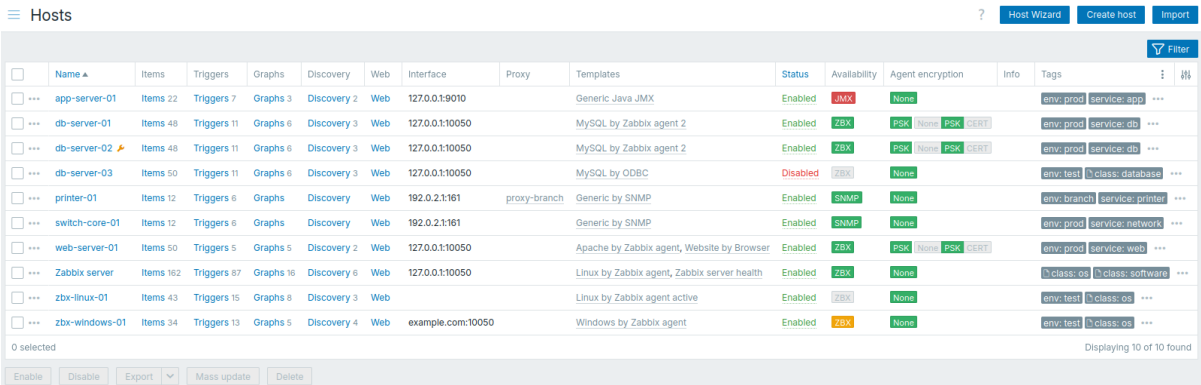
Parameter	Description
Template groups	Filtrar por um ou mais grupos de template. Especificar um grupo de template pai seleciona implicitamente todos os grupos aninhados.
Linked templates	Filtrar por templates vinculados diretamente.
Name	Filtrar por nome do template.
Vendor	Filtrar por fornecedor do template.
Version	Filtrar por versão do template.
Tags	Filtrar por nome e valor da tag do template. A filtragem é possível apenas por tags no nível do template (não as herdadas). É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados; **Equals** - incluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - excluir os nomes de tag especificados; **Does not equal** - excluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - basta que uma condição seja atendida.

Hosts

Visão geral

Na seção *Coleta de dados* > *Hosts*, os usuários podem configurar e manter hosts.

É exibida uma lista dos hosts existentes com seus detalhes.



Os detalhes dos hosts são exibidos em uma tabela personalizável. O botão *Personalizar tabela*, no cabeçalho da tabela, permite mostrar, ocultar ou reordenar colunas e redefinir o layout para o padrão. Você também pode redimensionar cada coluna arrastando sua borda direita; clique duas vezes na borda para redefinir. Todas as personalizações são aplicadas imediatamente e sempre são lembradas para o usuário atual.

As seguintes colunas estão disponíveis:

Coluna	Descrição
Nome	Nome do host. Clicar no nome do host abre o formulário de configuração do host. Um ícone de chave inglesa laranja  após o nome do host indica que este host está em manutenção. Os detalhes da manutenção são exibidos quando o ponteiro do mouse é posicionado sobre o ícone.
Entidades (Itens, Triggers, Gráficos, Descoberta, Web)	Clicar no nome da entidade exibirá os itens, 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 proxies, ou o Zabbix server não atribuiu um proxy para monitorar o host; Nada - o host não é monitorado por um proxy nem por um 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 de ferramenta fornece detalhes sobre o 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, a disponibilidade será atualizada depois que o Zabbix server sincronizar as alterações de configuração. Da mesma forma, se você habilitar um host, a disponibilidade será atualizada depois que o Zabbix server sincronizar as alterações de configuração e consultar o host. Os ícones de disponibilidade representam somente os tipos de interface (Agent, SNMP, IPMI, JMX) que estão configurados. Ao passar o mouse sobre o ícone, é exibida uma janela pop-up com uma lista de todas as interfaces do mesmo tipo, incluindo detalhes, status e erros. Para a interface Agent, a janela pop-up exibe interfaces (passivas) e verificações ativas. Se um host tiver somente 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 do host é determinado pela conexão entre um item habilitado que usa a interface e o host. O status pode ser: **Disponível** - a conexão com o host foi bem-sucedida; **Indisponível** - a conexão com o host não foi bem-sucedida (tempo limite, problemas de firewall etc.); **Desconhecido** - a conexão com o host não foi tentada ou o resultado é desconhecido. Para obter detalhes adicionais sobre como o Zabbix server determina o status da interface, consulte Status de interface desconhecido e Configurações de interface de host inacessível/indisponível. O status de todas as interfaces do host de um único tipo (Agent, SNMP, IPMI, JMX) é determinado pelas interfaces 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 está indisponível e pelo menos uma está disponível ou desconhecida; **Vermelho** - todas as interfaces estão indisponíveis; **Cinza** - pelo menos uma interface está desconhecida, mas nenhuma está indisponível. **Disponibilidade de verificações ativas.** Se pelo menos uma verificação ativa estiver habilitada no host, as verificações ativas também afetarão a disponibilidade total da interface Agent, conforme descrito acima. Para determinar a disponibilidade das verificações ativas, mensagens de heartbeat são enviadas na thread de verificações ativas do agent. A frequência das mensagens de heartbeat é controlada pelo parâmetro `HeartbeatFrequency` na configuração do agent ou do agent 2 do Zabbix (padrão de 60 segundos, intervalo de 0 a 3600). As verificações ativas são consideradas indisponíveis quando o heartbeat da verificação ativa é mais antigo que 2 x HeartbeatFrequency segundos. **Observação:** agents do Zabbix anteriores à versão 6.2.x não enviam heartbeats de verificações ativas; portanto, a disponibilidade de seus hosts permanece desconhecida.
Criptografia do agent	O status da criptografia das conexões com o host e do host é exibido: **Nenhuma** - sem criptografia; **PSK** - usando chave pré-compartilhada; **Cert** - usando certificado.
Informações	As informações de erro, se houver, relacionadas ao host são exibidas.
Tags	As tags do host são exibidas, com as macros não resolvidas. Passar o mouse sobre ou clicar no ícone de três pontos após as tags exibidas abre uma janela pop-up com todas as tags. Clicar no botão *Opções* no cabeçalho da coluna permite definir o número de tags exibidas (1–3), escolher como os nomes das tags serão exibidos (completos, abreviados ou nenhum), definir a prioridade de exibição das tags ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas).

Coluna	Descrição
Valor da tag	Os valores das tags do host são exibidos, com as macros não resolvidas. Clicar no botão ⋮ *Opções* no cabeçalho da coluna permite renomear a coluna, escolher quais valores de tags serão exibidos ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas).
Texto personalizado	Texto personalizado, valores de macros ou uma combinação de ambos são exibidos. Clicar no botão ⋮ *Opções* no cabeçalho da coluna permite renomear a coluna, inserir o texto personalizado ou as macros ou duplicar a coluna (as colunas duplicadas podem ser renomeadas ou excluídas). Macros compatíveis: `{HOST.*}` (exceto `{HOST.METADATA}`), `{INVENTORY.*}` e macros de usuário.

Para abrir o menu do host, clique no botão ⋮ *Opções* antes do nome do host.

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:

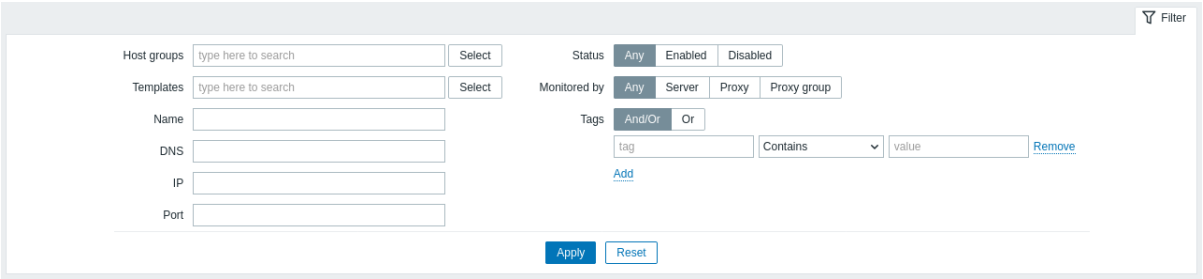
- *Enable* - altera o status do host para *Monitored*;
- *Disable* - altera o status do host para *Not monitored*;
- *Export* - exporta os hosts para um arquivo YAML, XML ou JSON;
- *Mass update* - atualiza várias propriedades de vários hosts de uma vez;
- *Delete* - exclui os hosts.

Para usar essas opções, marque as caixas de seleção antes dos respectivos hosts e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os hosts de seu interesse. Para um melhor desempenho na busca, os dados são pesquisados com macros não resolvidas.

O ícone *Filter* está disponível no canto superior direito. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados.



Parameter	Description
Host groups	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.
Name	Filtrar pelo nome visível do host.
DNS	Filtrar pelo nome DNS.
IP	Filtrar pelo endereço IP.
Port	Filtrar pelo número da porta.
Status	Filtrar pelo status do host.
Monitored by	Filtrar hosts monitorados por Zabbix server, proxy ou grupo de proxy.
Proxies	Filtrar hosts monitorados pelos proxies especificados aqui. Este campo só está disponível se ”Proxy” estiver selecionado no campo *Monitored by*.
Proxy groups	Filtrar hosts monitorados pelos grupos de proxy especificados aqui. Este campo só está disponível se ”Proxy group” estiver selecionado no campo *Monitored by*.

Parameter	Description
Tags	Filtrar por nome e valor da tag do host. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados; **Equals** - incluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - excluir os nomes de tag especificados; **Does not equal** - excluir os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas, as condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - basta que uma condição seja atendida.

Status de interface desconhecido

O Zabbix server exibe o status ”Unknown” para uma interface de host (Agent, SNMP, IPMI, JMX) nos seguintes casos:

- O host está desabilitado.
- O host está configurado para ser monitorado por proxy, por um proxy diferente, ou pelo server, se anteriormente era monitorado por 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 itens 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 ”Unknown” após a sincronização do cache de configuração do Zabbix server.

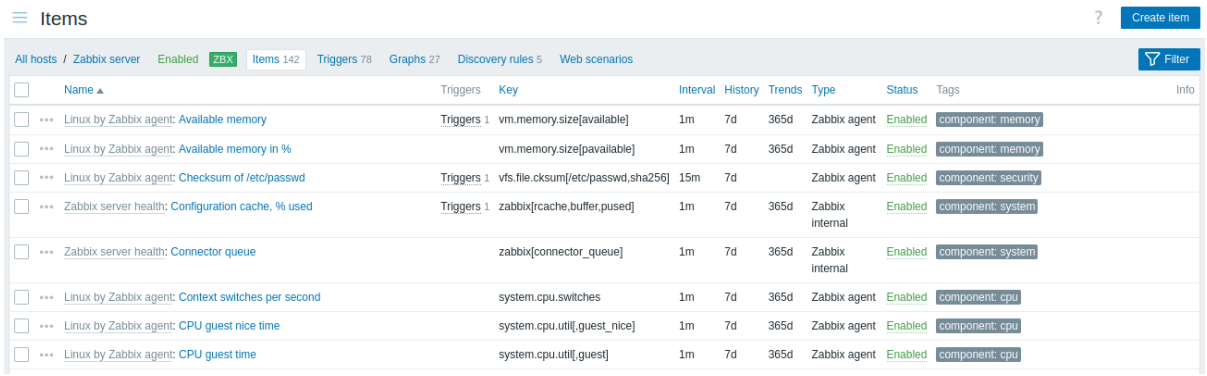
A disponibilidade da interface (Available/Not available) em hosts monitorados por proxies é restaurada após a sincronização do cache de configuração do proxy.

Itens

Visão geral

A lista de itens de um host pode ser acessada em *Coleta de dados* > *Hosts*, clicando em *Itens* para o host correspondente.

Uma lista de itens 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 somente quando vários hosts ou nenhum host está selecionado 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 será exibido antes do nome do item como um link cinza. Clicar no link do template abrirá a lista de itens no nível do template. Se o item tiver sido criado a partir de um protótipo de item, seu nome será 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, uma caixa de informações com os triggers associados ao item será exibida. O número de triggers é exibido em cinza.
Chave	A chave do item é exibida.
Intervalo	A frequência da verificação é exibida. *Observação*: itens passivos também podem ser verificados imediatamente pressionando o botão *Executar agora*.
Histórico	É exibido por quantos dias o histórico de dados do item será mantido.
Tendências	É exibido por quantos dias o histórico de tendências do item será mantido.
Tipo	O tipo de item é exibido (Zabbix agent, SNMP agent, verificação 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). Itens descobertos que foram perdidos são marcados com um ícone de informação. O texto da dica de ferramenta fornece detalhes sobre o status.
Tags	As tags do item são exibidas. Até três tags (pares nome:valor) podem ser exibidas. Se houver mais tags, será exibido um link "..." que permite visualizar todas as tags ao passar o mouse sobre ele.
Informações	Se o item estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, será exibido um ícone quadrado com a letra "i". Passe o mouse sobre o ícone para ver uma dica de ferramenta 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:

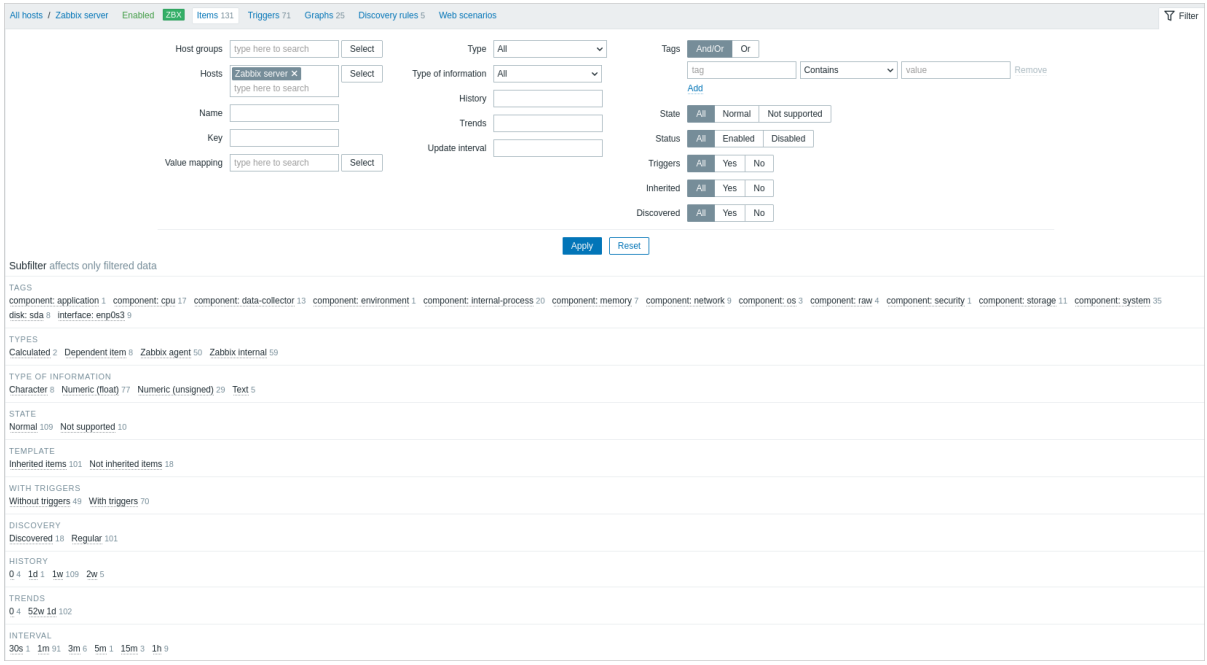
- *Enable* - altera o status do item para *Enabled*
- *Disable* - altera o status do item para *Disabled*
- *Execute now* - executa imediatamente uma verificação para novos valores do item. Suportado apenas para verificações **passivas** (veja mais detalhes). Observe que, ao verificar valores imediatamente, o cache de configuração não é atualizado; portanto, os valores não refletirão alterações muito recentes na configuração do item.
- *Clear history and trends* - exclui os dados de histórico e tendências dos items.
- *Copy* - copia os items para outros hosts ou templates.
- *Mass update* - atualiza várias propriedades de vários items de uma vez.
- *Delete* - exclui os items.

Para usar essas opções, marque as caixas de seleção antes dos respectivos items e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os items de seu interesse. Para um melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filter* está disponível no canto superior direito. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados.

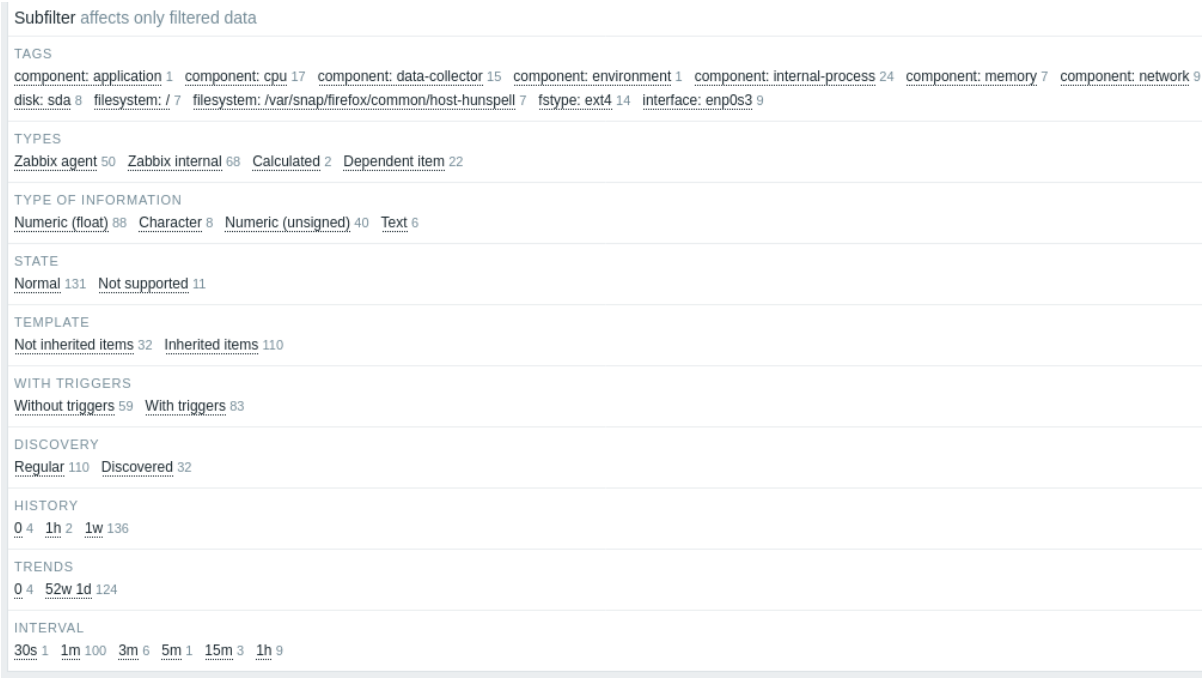


Parameter	Description
Host groups	Filtrar por um ou mais host groups. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Host groups que contêm apenas templates não podem ser selecionados.
Hosts	Filtrar por um ou mais hosts.
Name	Filtrar pelo nome do item.
Key	Filtrar pela key do item.
Value mapping	Filtrar pelo value map usado. Este parâmetro não é exibido se a opção *Hosts* estiver vazia.
Type	Filtrar pelo tipo do item (Zabbix agent, SNMP agent, etc.).
Type of information	Filtrar pelo tipo de informação (Numeric unsigned, float, etc.).
History	Filtrar por quanto tempo o histórico do item é mantido.
Trends	Filtrar por quanto tempo as tendências do item são mantidas.
Update interval	Filtrar pelo intervalo de atualização do item.
Tags	Especifique tags para limitar o número de items exibidos. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas) **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem distinção entre maiúsculas e minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida
State	Filtrar pelo estado do item - *Normal* ou *Not supported*.
Status	Filtrar pelo status do item - *Enabled* ou *Disabled*.
Triggers	Filtrar items com (ou sem) triggers.
Inherited	Filtrar items herdados (ou não herdados) de um template.
Discovery	Filtrar items descobertos (ou não descobertos) por low-level discovery.

Usando o subfiltro

O subfiltro modifica ainda mais a filtragem aplicada pelo filtro principal.

Ele contém links clicáveis para acesso rápido a itens relacionados. Os itens são relacionados por uma entidade comum, como 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 filtrados imediatamente (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outra entidade adiciona-a aos resultados filtrados. Clicar novamente na entidade remove a filtragem.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se quiser ver mais registros no subfiltro, aumente 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 apresentar informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

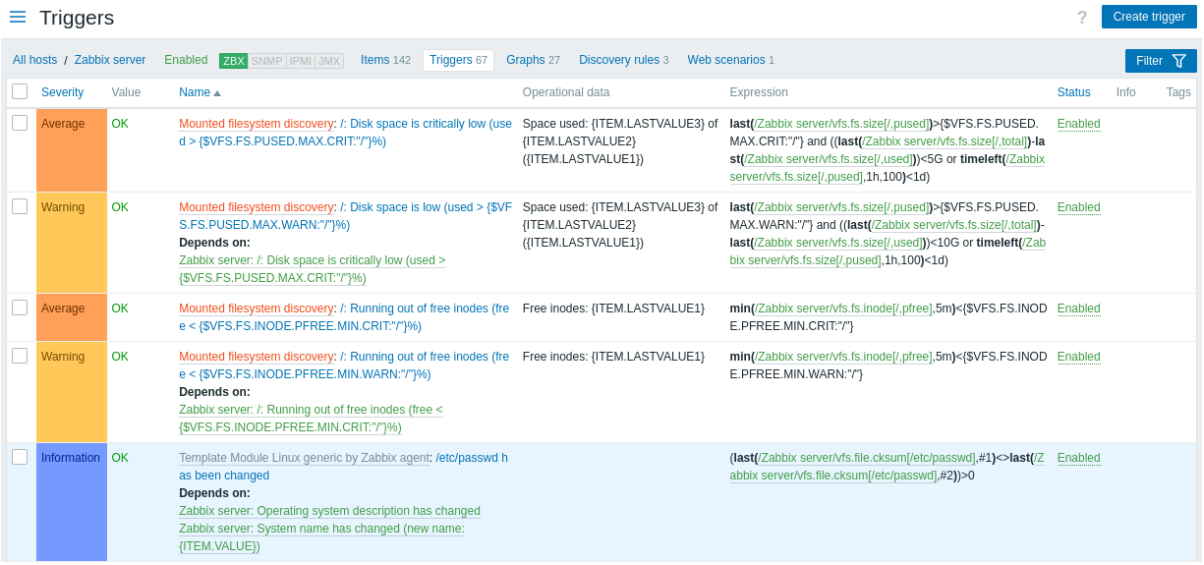
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais entidades, um ícone de três pontos será exibido no final; ele não é clicável.

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 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.

triggers

Visão geral

A lista de triggers de um host pode ser acessada em *Coleta de dados* > *Hosts*, clicando em *Triggers* para o host correspondente.



Dados exibidos:

Coluna	Descrição
Severidade	A severidade do trigger é exibida pelo nome e pela cor de fundo da célula.
Valor	O estado do trigger é exibido: **OK** - estado OK **PROBLEM** - estado de problema
Host	Host do trigger. Clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida somente quando vários hosts ou nenhum host estão 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 tiver sido criado a partir de um protótipo de trigger, seu nome será precedido pelo nome da regra de descoberta de baixo nível, em laranja. Clicar no nome da regra de descoberta abrirá a lista de protótipos de trigger.
Dados operacionais	Definição dos dados operacionais do trigger, contendo strings e macros arbitrárias 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 que leva ao formulário de configuração do item.
Status	O status do trigger é exibido: *Habilitado*, *Desabilitado* ou *Desconhecido*. Ao clicar no status, você pode alterá-lo manualmente: de Habilitado para Desabilitado (e vice-versa); de Desconhecido para Desabilitado (e vice-versa). Os problemas de um trigger desabilitado não são mais exibidos no frontend, mas não são excluídos. Os triggers descobertos que foram perdidos são marcados com um ícone de informação. O texto da dica de ferramenta fornece detalhes sobre o status.
Informações	Se tudo estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, será exibido um ícone quadrado com a letra "i". Passe o cursor 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:

- *Enable* - altera o status do trigger para *Enabled*.
- *Disable* - altera o status do trigger para *Disabled*.
- *Copy* - copia os triggers para outros hosts ou templates.
- *Mass update* - atualiza várias propriedades de vários triggers de uma vez.
- *Delete* - exclui os triggers.

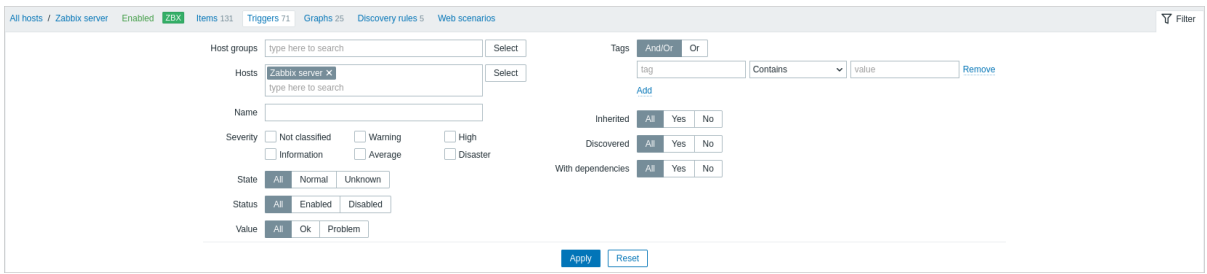
Para usar essas opções, marque as caixas de seleção antes dos respectivos triggers e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os triggers de seu interesse.

Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filter* está disponível no canto superior direito. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados.



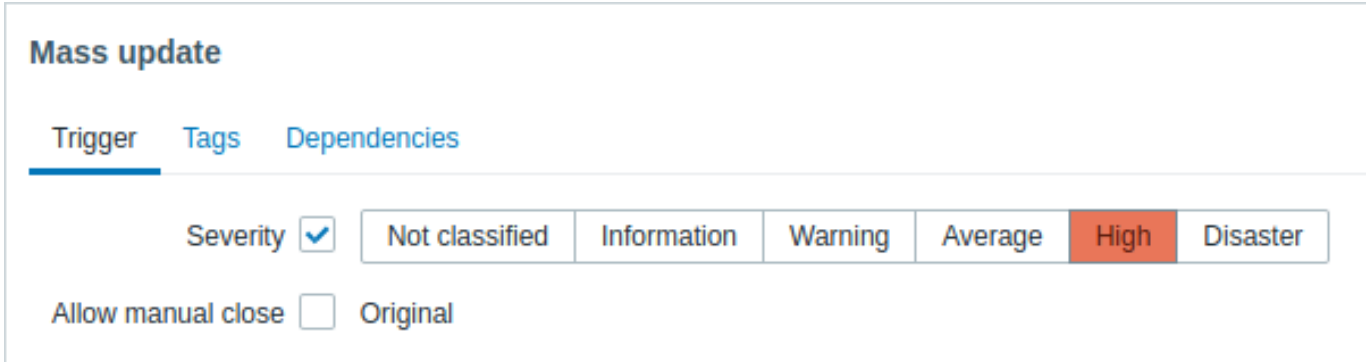
Parameter	Description
Host groups	Filtrar por um ou mais host groups. Especificar um host group pai seleciona implicitamente todos os host groups aninhados. Host groups que contêm apenas templates não podem ser selecionados.
Hosts	Filtrar por um ou mais hosts. Se host groups já estiverem selecionados acima, a seleção de hosts fica limitada a esses grupos.
Name	Filtrar pelo nome do trigger.
Severity	Selecione para filtrar por uma ou várias severidades do trigger.
State	Filtrar pelo estado da expressão do trigger (Normal/Unknown).
Status	Filtrar pelo status de ativação do trigger (Enabled/Disabled).
Value	Filtrar pelo valor do trigger.
Tags	Filtrar pelo nome e valor da tag do trigger. É possível incluir e excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - incluir os nomes de tag especificados **Equals** - incluir os nomes e valores de tag especificados (case-sensitive) **Contains** - incluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) **Does not exist** - excluir os nomes de tag especificados **Does not equal** - excluir os nomes e valores de tag especificados (case-sensitive) **Does not contain** - excluir os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas) Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; condições com o mesmo nome de tag serão agrupadas pela condição Or **Or** - basta que uma condição seja atendida Macros e funções de macro são suportadas tanto nos campos de nome da tag quanto nos campos de valor da tag.
Inherited	Filtrar triggers herdados (ou não herdados) de um template.
Discovered	Filtrar triggers descobertos (ou não descobertos) por low-level discovery.
With dependencies	Filtrar triggers com (ou sem) dependências.

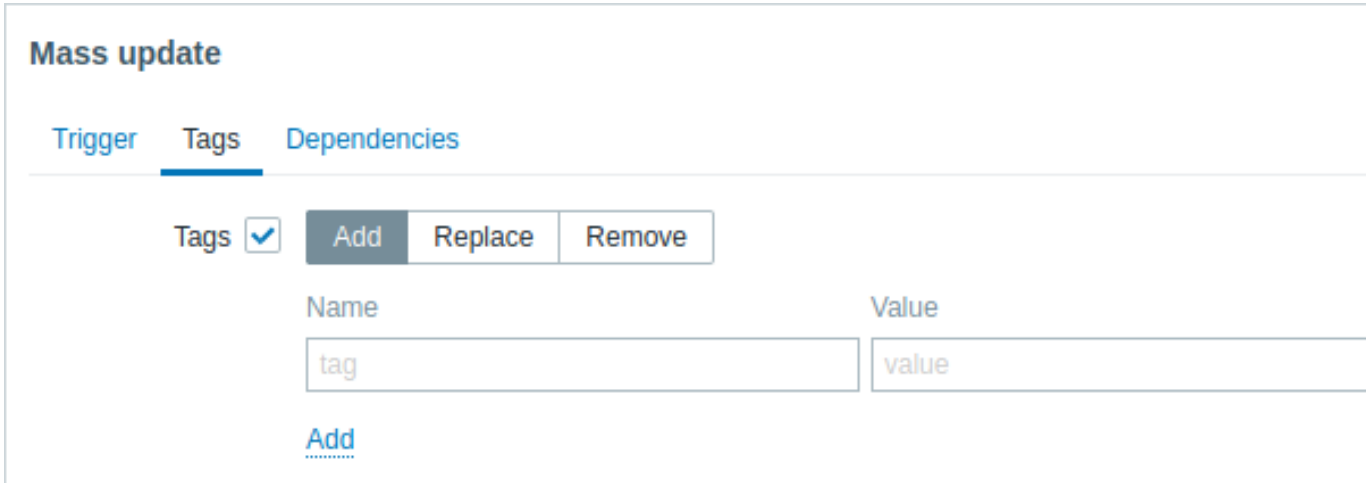
Usando a atualização em massa

Com a atualização em massa, você pode alterar algum atributo de vários triggers de uma só vez, evitando a necessidade de abrir cada trigger individualmente para edição.

Para atualizar vários triggers em massa, faça o seguinte:

- Marque as caixas de seleção dos triggers que deseja atualizar na lista
- Clique em *Atualização em massa* abaixo da lista
- Navegue até a aba com os atributos necessários (*Trigger*, *Tags* ou *Dependências*)
- Marque as caixas de seleção dos atributos que deseja atualizar

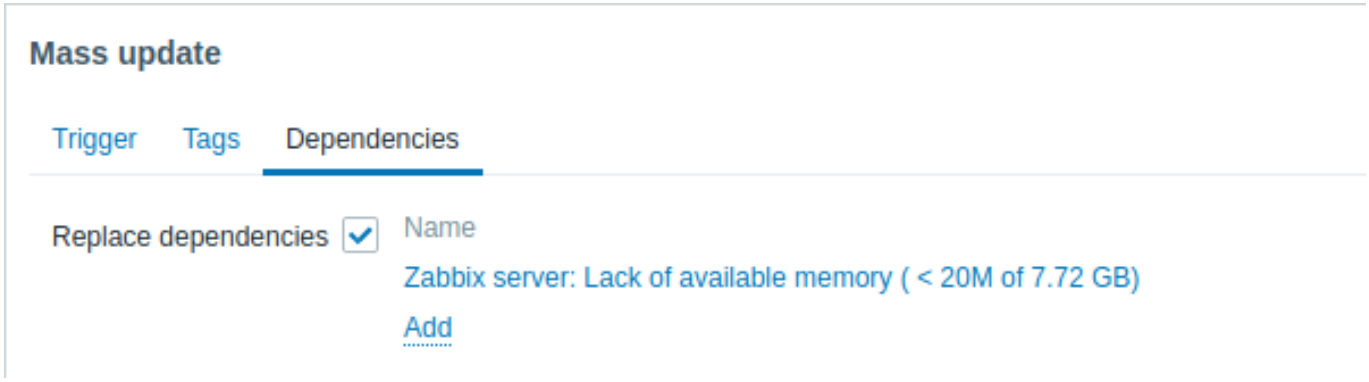




As seguintes opções estão disponíveis ao selecionar o botão correspondente para a atualização de tags:

- *Adicionar* - adiciona novas tags aos triggers;
- *Substituir* - remove todas as tags existentes do trigger e as substitui pela(s) tag(s) especificada(s) abaixo;
- *Remover* - remove as tags especificadas dos triggers.

Observe que tags com o mesmo nome, mas com valores diferentes, não são consideradas ”duplicadas” e podem ser adicionadas ao mesmo trigger.



Substituir dependências - remove todas as dependências existentes do trigger e as substitui pela(s) dependência(s) especificada(s).

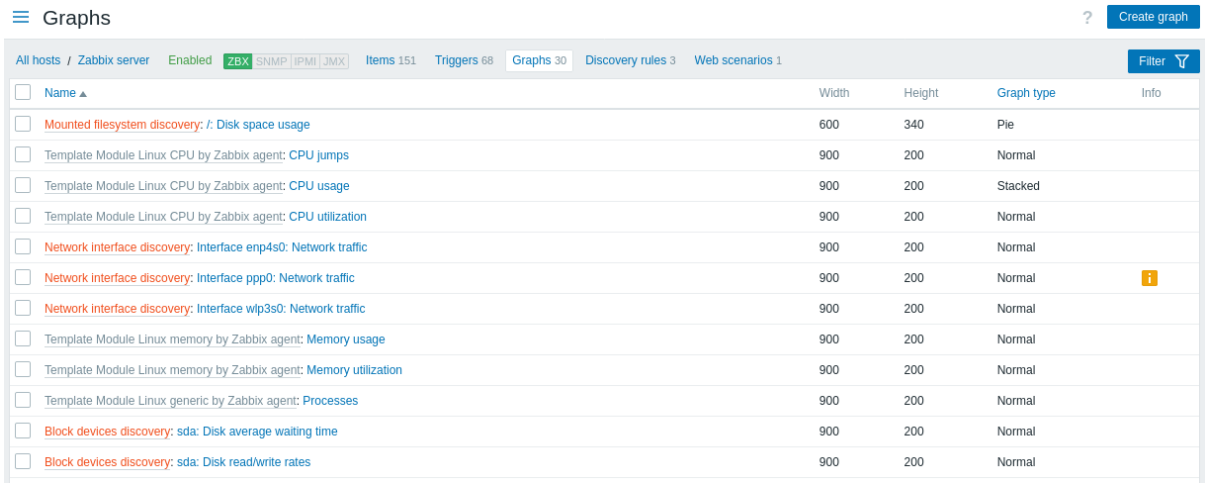
Clique em *Atualizar* para aplicar as alterações.

Gráficos

Visão geral

A lista de gráficos personalizados de um host pode ser acessada em *Coleta de dados* > *Hosts*, clicando em *Gráficos* para o host correspondente.

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 somente quando vários hosts ou nenhum host está selecionado no filtro.
Nome	Nome do gráfico personalizado, exibido como um link azul para os detalhes do gráfico. Clicar no link do nome do gráfico abre o formulário de configuração do gráfico. Se o gráfico do host pertencer a um template, o nome do template será exibido antes do nome do gráfico como um link cinza. Clicar no link do template abrirá a lista de gráficos no nível do template. Se o gráfico tiver sido criado a partir de um protótipo de gráfico, seu nome será precedido pelo nome da regra de descoberta de baixo nível, em laranja. Clicar no nome da regra de descoberta abrirá a lista de protótipos de gráficos.
Largura	A largura do gráfico é exibida.
Altura	A altura do gráfico é exibida.
Tipo de gráfico	O tipo de gráfico é exibido: *Normal*, *Empilhado*, *Pizza* ou *Explodido*.
Informações	Se o gráfico estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, será exibido um ícone quadrado com a letra "i". Passe o cursor 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:

- *Copy* - copiar os gráficos para outros hosts ou templates
- *Delete* - excluir os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos e, em seguida, clique no botão desejado.

Usando filtro

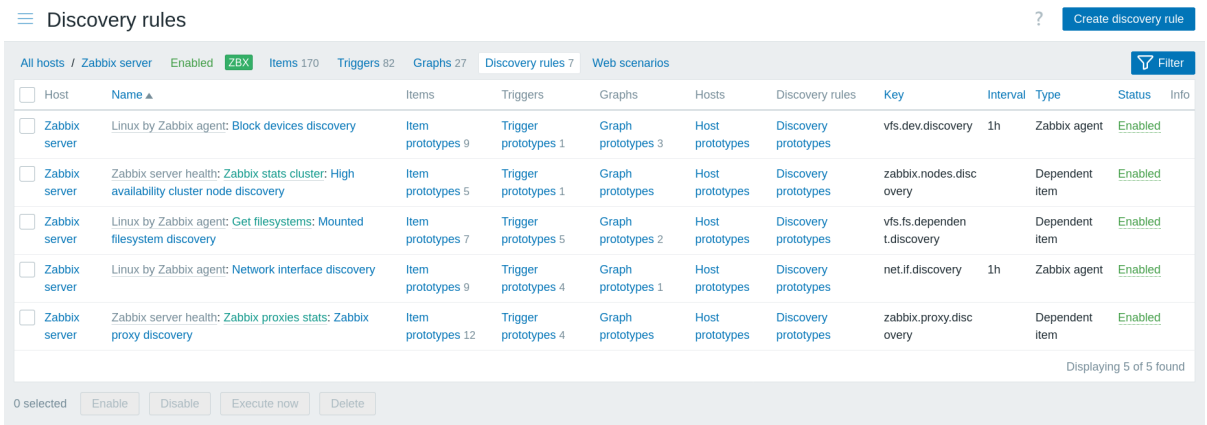
Você pode filtrar gráficos por grupo de hosts e host. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Regras de descoberta

Visão geral

A lista de regras de descoberta de baixo nível de um host pode ser acessada em *Coleta de dados* > *Hosts*, clicando em *Descoberta* para o host correspondente.

Uma lista das regras de descoberta de baixo nível existentes é exibida. Também é possível visualizar 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 de host visível, o nome técnico do host é exibido.
Nome	O nome da regra é 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 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 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 a descoberta é exibida.
Intervalo	A frequência de execução da descoberta é exibida. *Observação*: a descoberta também pode ser executada imediatamente pressionando o botão *Executar agora* abaixo da lista.
Tipo	O tipo de item usado para a descoberta é exibido (Zabbix agent, SNMP agent etc.).
Status	O status da regra de descoberta é exibido: *Habilitado*, *Desabilitado* ou *Não suportado*. Ao clicar no status, é possível alterá-lo: de Habilitado para Desabilitado (e vice-versa); de Não suportado para Desabilitado (e vice-versa).
Informações	Se tudo estiver correto, nenhum ícone será exibido nesta coluna. Em caso de erros, será exibido um ícone quadrado com a letra "i". Passe o cursor 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:

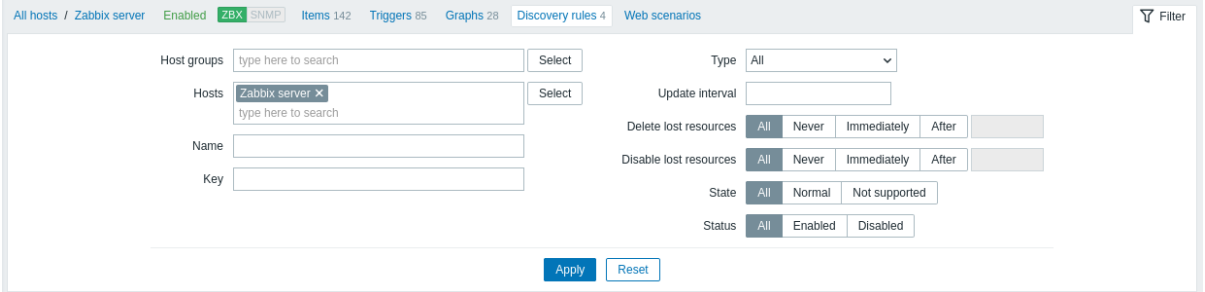
- *Enable* - altera o status da regra de descoberta de baixo nível para *Enabled*.
- *Disable* - altera o status da regra de descoberta de baixo nível para *Disabled*.
- *Execute now* - executa a descoberta com base nas regras de descoberta imediatamente. Veja mais detalhes. Observe que, ao executar a descoberta imediatamente, o cache de configuração não é atualizado; portanto, o resultado não refletirá alterações muito recentes na configuração da regra de descoberta.
- *Delete* - exclui as regras de descoberta de baixo nível.

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de descoberta e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta de seu interesse. Para melhor desempenho da busca, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de regras de descoberta. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar regras de descoberta por grupo de host, host, nome, chave do item, tipo do item e outros parâmetros.

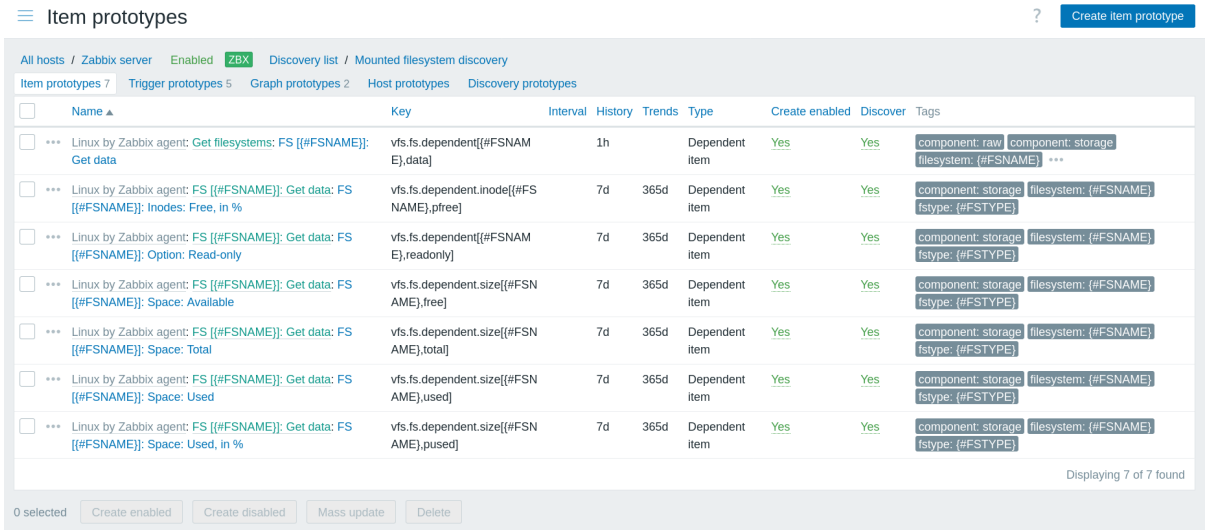


Parameter	Description
Host groups	Filtrar por um ou mais grupos de host. Especificar um grupo de host pai seleciona implicitamente todos os grupos de host aninhados.
Hosts	Filtrar por um ou mais hosts.
Name	Filtrar pelo nome da regra de descoberta.
Key	Filtrar pela chave do item de descoberta.
Type	Filtrar pelo tipo do item de descoberta.
Update interval	Filtrar pelo intervalo de atualização. Não disponível para Zabbix trapper e itens dependentes.
Delete lost resources	Filtrar pelo período de *Delete lost resources*.
Disable lost resources	Filtrar pelo período de *Disable lost resources*.
SNMP OID	Filtrar pelo SNMP OID. Disponível apenas se *SNMP agent* estiver selecionado como tipo.
State	Filtrar pelo estado da regra de descoberta (All/Normal/Not supported).
Status	Filtrar pelo status da regra de descoberta (All/Enabled/Disabled).

Protótipos de item

Visão geral

Nesta seção, são exibidos os protótipos de item de uma regra de descoberta de baixo nível no host. Os protótipos de item são a base dos itens reais do host, 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 ”Dependent item”, 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	O número de dias durante os quais o histórico de dados do item deve ser mantido é exibido.
Tendências	O número de dias durante os quais o histórico de tendências do item deve ser mantido é exibido.
Tipo	O tipo do protótipo de item é exibido (Zabbix agent, SNMP agent, verificação simples etc.).
Criação habilitada	Cria o item com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre ”Sim” e ”Não” clicando nessas opções.

Coluna	Descrição
Descobrir	Descobre o item com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre ”Sim” e ”Não” clicando nessas opções.
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:

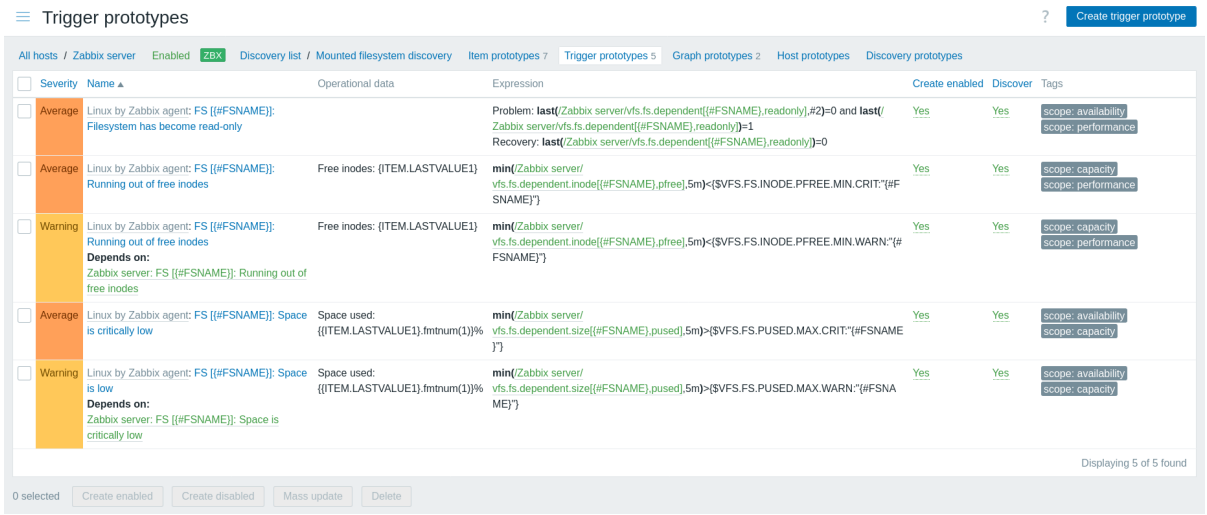
- *Criar habilitado* - cria estes itens como *Habilitado*
- *Criar desabilitado* - cria estes itens como *Desabilitado*
- *Atualização em massa* - atualiza em massa estes protótipos de item
- *Excluir* - exclui estes protótipos de item

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de item e, em seguida, clique no botão desejado.

Protótipos de trigger

Visão geral

Nesta seção, são exibidos os protótipos de trigger de uma regra de descoberta de baixo nível no host. Os protótipos de trigger são a base dos triggers reais do host, que são criados durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de trigger, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de trigger. Se o protótipo de trigger pertencer a um template vinculado, o nome do template será exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de protótipos de trigger no nível do template vinculado.
Dados operacionais	É exibido o formato dos dados operacionais do trigger, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* > *Problemas*.
Criação habilitada	Cria o trigger com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Descobrir	Descobre o trigger com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Tags	São exibidas as tags do protótipo de trigger.

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:

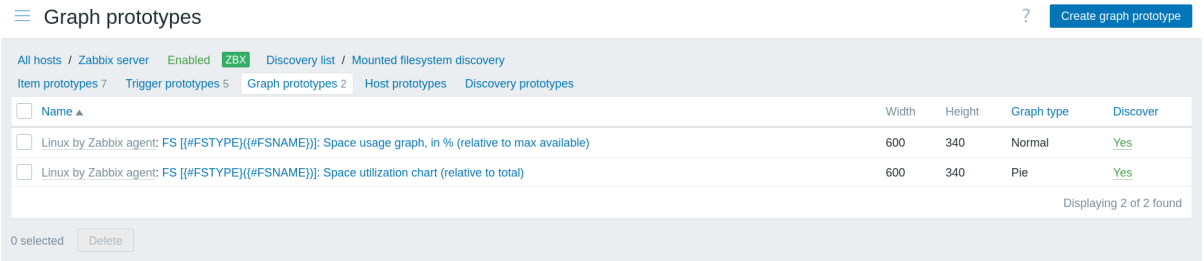
- *Criar habilitado* - criar estes triggers como *Habilitado*
- *Criar desabilitado* - criar estes triggers como *Desabilitado*
- *Atualização em massa* - atualizar em massa estes protótipos de trigger
- *Excluir* - excluir estes protótipos de trigger

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de trigger e, em seguida, clique no botão desejado.

Protótipos de gráfico

Visão geral

Nesta seção, são exibidos os protótipos de gráfico de uma regra de descoberta de baixo nível no host. Os protótipos de gráfico são a base dos gráficos reais do host, criados durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Name	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 gráfico no nível do template vinculado.
Width	A largura do protótipo de gráfico é exibida.
Height	A altura do protótipo de gráfico é exibida.
Type	O tipo do protótipo de gráfico é exibido: *Normal*, *Stacked*, *Pie* ou *Exploded*.
Discover	Descobrir o gráfico com base neste protótipo: **Yes** - descobrir **No** - não descobrir. É possível alternar entre 'Yes' e 'No' clicando nessas opções.

Para configurar um novo protótipo de gráfico, clique no botão *Create graph prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

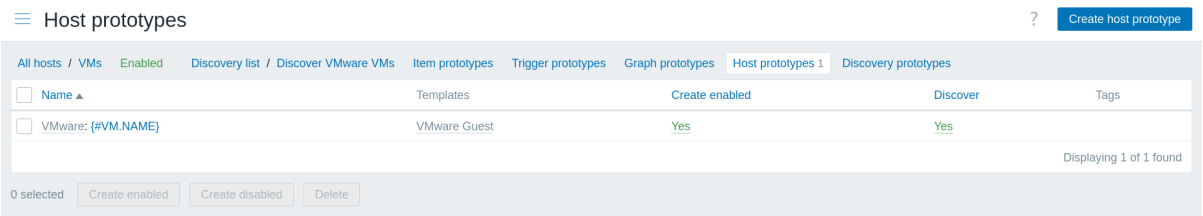
- *Excluir* - exclui estes protótipos de gráfico

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráfico e, em seguida, clique no botão desejado.

Protótipos de host

Visão geral

Nesta seção, são exibidos os protótipos de host de uma regra de descoberta de baixo nível no host. Os protótipos de host são a base dos hosts reais criados durante a descoberta de baixo nível.



Dados exibidos:

Column	Description
Name	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	São exibidos os templates do protótipo de host.
Create enabled	Criar o host com base neste protótipo como: **Yes** - habilitado **No** - desabilitado. Você pode alternar entre 'Yes' e 'No' clicando nessas opções.
Discover	Descobrir o host com base neste protótipo: **Yes** - descobrir **No** - não descobrir. Você pode alternar entre 'Yes' e 'No' clicando nessas opções.
Tags	São exibidas as tags do protótipo de host.

Para configurar um novo protótipo de host, clique no botão *Create host prototype* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

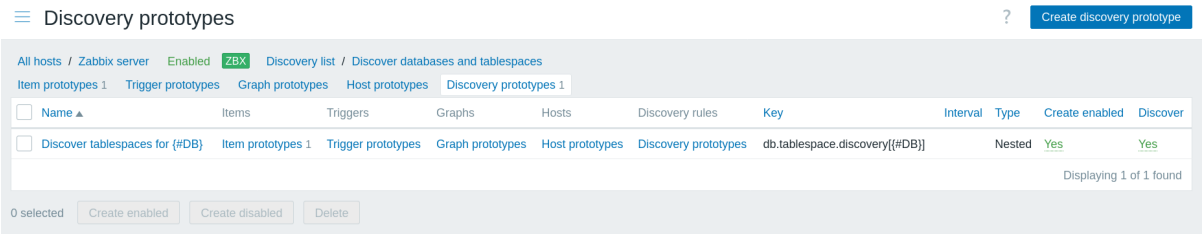
- *Criar habilitado* - cria esses hosts como *Habilitado*
- *Criar desabilitado* - cria esses hosts como *Desabilitado*
- *Excluir* - exclui esses protótipos de host

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e clique no botão desejado.

Protótipos de descoberta

Visão geral

Nesta seção, são exibidos os protótipos de descoberta de uma regra de descoberta de baixo nível no host. 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 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 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.

Coluna	Descrição
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	É exibida a chave do item usada para a descoberta.
Intervalo	É exibida a frequência de execução da descoberta. *Observação*: a descoberta também pode ser executada imediatamente clicando no botão *Executar agora* abaixo da lista.
Tipo	É exibido o tipo de item usado para a descoberta (Zabbix agent, agente SNMP etc.).
Criação habilitada	É exibido o status de criação do protótipo de descoberta: criação habilitada (*Sim*) ou criação desabilitada (*Não*). Ao clicar no status, é possível alterá-lo de Sim para Não (e vice-versa).
Descobrir	É exibido o status de descoberta do protótipo de descoberta: descobrir (*Sim*) ou não descobrir (*Não*). Ao clicar no status, é possível 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* - cria estes protótipos de descoberta como *Habilitado*
- *Criar desabilitado* - cria estes protótipos de descoberta como *Desabilitado*
- *Excluir* - exclui estes protótipos de descoberta

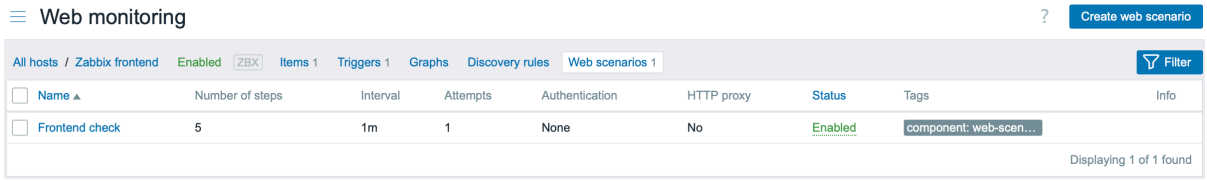
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de descoberta e, em seguida, clique no botão desejado.

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 host correspondente.

Uma lista dos 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, Kerberos, Digest ou None.
Proxy HTTP	Exibe o proxy HTTP ou 'No' se não for utilizado.
Status	O status do cenário web é exibido: *Enabled* ou *Disabled*. 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, será exibido um link "..." que permite visualizar todas as tags ao passar o mouse sobre ele.
Informações	Se tudo estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, será exibido um ícone quadrado com a letra "i". 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:

- *Enable* - altera o status do cenário para *Enabled*
- *Disable* - altera o status do cenário para *Disabled*
- *Clear history and trends* - limpa os dados de histórico e tendências dos cenários
- *Delete* - exclui os web scenarios

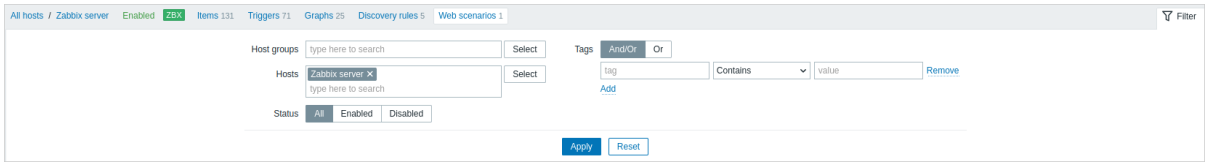
Para usar essas opções, marque as caixas de seleção antes dos respectivos web scenarios e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os cenários de seu interesse.

Para melhor desempenho na busca, a pesquisa de dados é feita com macros não resolvidas.

O link *Filter* está disponível acima da lista de web scenarios. Ao clicar nele, será aberto um filtro no qual você pode especificar os critérios de filtragem desejados.



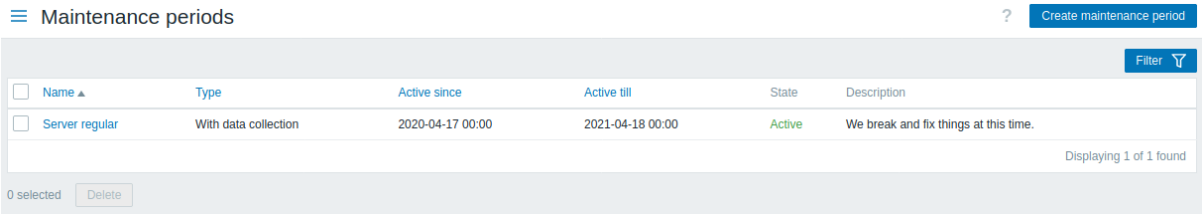
Parameter	Description
Host groups	Filtrar por um ou mais host groups. Especificar um host group pai seleciona implicitamente todos os host groups aninhados.
Hosts	Filtrar por um ou mais hosts. Se host groups já estiverem selecionados acima, a seleção de hosts fica limitada a esses grupos.
Status	Filtrar por status do web scenario.
Tags	Filtrar por nome e valor da tag do web scenario. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag sempre diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Exists** - inclui os nomes de tag especificados; **Equals** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Contains** - inclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas); **Does not exist** - exclui os nomes de tag especificados; **Does not equal** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas); **Does not contain** - exclui os nomes de tag especificados cujos valores de tag contenham a string informada (correspondência de substring, sem diferenciar maiúsculas de minúsculas). Há dois tipos de cálculo para as condições: **And/Or** - todas as condições devem ser atendidas; as condições com o mesmo nome de tag serão agrupadas pela condição Or; **Or** - basta que uma condição seja atendida.

Manutenção

Visão geral

Na seção *Coleta de dados* > *Manutenção*, os usuários podem configurar e manter períodos de manutenção para hosts.

É exibida uma lista dos períodos de manutenção existentes com seus detalhes.



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 a hora em que a execução dos períodos de manutenção se torna ativa. Observação: esse horário não ativa um período de manutenção; os períodos de manutenção precisam ser definidos separadamente.
Ativo até	A data e a hora em que a execução dos períodos de manutenção deixa de estar ativa.
Estado	O estado do período de manutenção: **Próximo** - será ativado em breve **Ativo** - está ativo **Expirado** - não está mais ativo
Descrição	A 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:

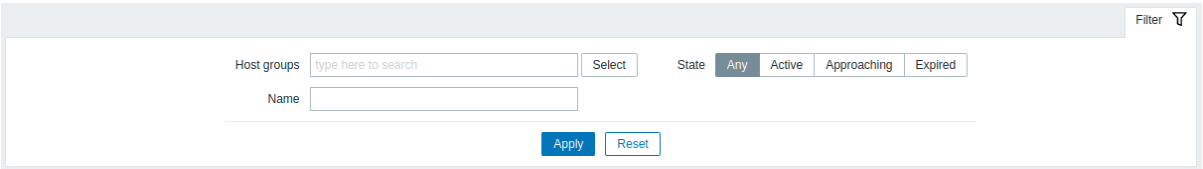
- *Delete* - excluir os períodos de manutenção

Para usar essa opção, marque as caixas de seleção antes dos respectivos períodos de manutenção e clique em *Delete*.

Usando o filtro

Você pode usar o filtro para exibir apenas os períodos de manutenção de seu interesse. Para melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de períodos de manutenção. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar os períodos de manutenção por grupo de host, nome e estado.



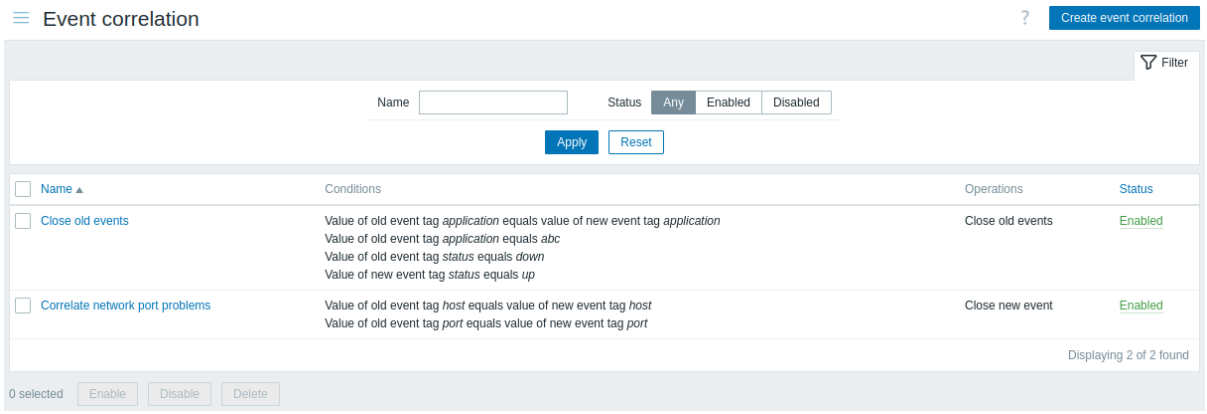
Cálculo de filas durante a manutenção

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

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:

Column	Description
Nome	Nome da regra de correlação. Clicar no nome da regra de correlação abre o formulário de configuração da correlação de eventos.
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 - *Habilitada* ou *Desabilitada*. Clicando 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:

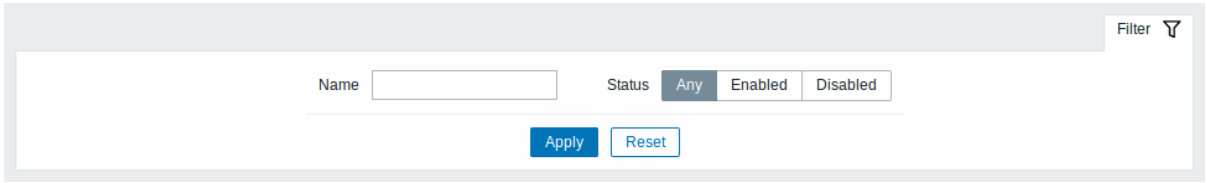
- *Enable* - altera o status da regra de correlação para *Enabled*
- *Disable* - altera o status da regra de correlação para *Disabled*
- *Delete* - exclui as regras de correlação

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de correlação e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de correlação de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de regras de correlação. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar as regras de correlação por nome e status.

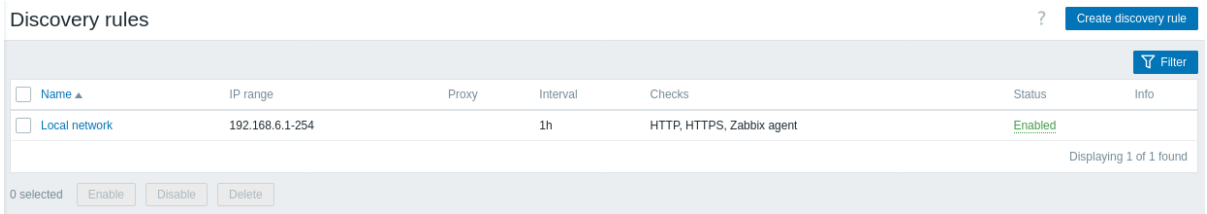


Descoberta

Visão geral

Na seção *Coleta de dados* > *Descoberta*, os usuários podem configurar e manter regras de descoberta.

É exibida uma lista das regras de descoberta existentes com seus detalhes.



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	É exibido o intervalo de endereços IP a ser usado para a varredura da rede.
Proxy	O nome do proxy é exibido se a descoberta for realizada pelo proxy.
Intervalo	É exibida a frequência de realização da descoberta.
Verificações	São exibidos os tipos de verificações usados para a descoberta.
Status	É exibido o status da regra de descoberta - *Habilitada* ou *Desabilitada*. Clicando no status, você pode alterá-lo.
Informações	Se tudo estiver funcionando corretamente, nada será exibido nesta coluna. Em caso de erros, será exibido um ícone vermelho de informações com a letra "i". Passe o cursor 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:

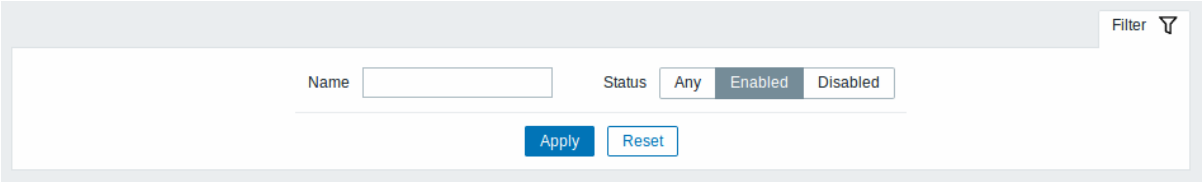
- *Enable* - altera o status da regra de descoberta para *Enabled*
- *Disable* - altera o status da regra de descoberta para *Disabled*
- *Delete* - exclui as regras de descoberta

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de descoberta e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta nas quais você está interessado. Para melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de regras de descoberta. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar as regras de descoberta por nome e status.



Alertas

Visão geral

Este menu apresenta seções relacionadas à configuração de alertas no Zabbix.

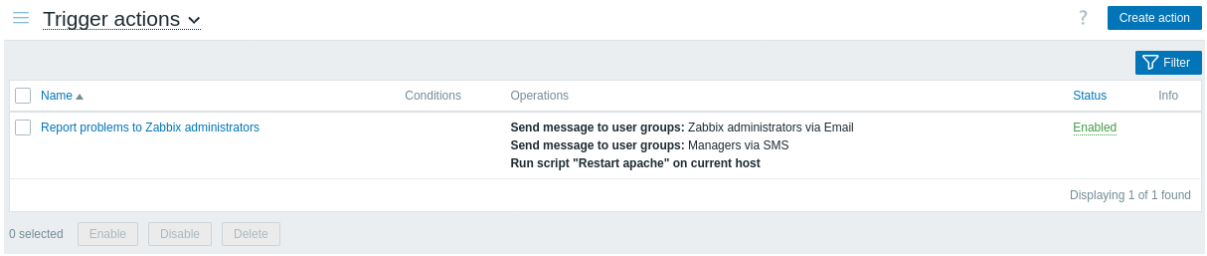
Ações

Visão geral

Na seção *Alertas* > *Ações*, os usuários podem configurar e manter ações.

As ações exibidas são as ações atribuídas à fonte de evento selecionada (trigger, serviços, descoberta, registro automático, ações internas).

Para mudar para outra fonte de evento, clique em *Ações* na seção do menu *Alertas*. Também é possível alternar entre as fontes usando a lista suspensa 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 a notificação, bem como o nome e o sobrenome (entre parênteses após o nome de usuário) do destinatário da notificação. A operação da ação pode ser uma notificação ou um comando remoto, dependendo do tipo de operação selecionado.
Status	O status da ação é exibido: *Habilitada* ou *Desabilitada*. Clicando no status, você pode alterá-lo. Consulte a seção Escalonamentos para obter mais detalhes sobre o que acontece quando uma ação é desabilitada durante um escalonamento em andamento.
Informações	Se tudo estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, como operações ou condições de ação ausentes após uma atualização ou descoberta de baixo nível, um ícone de aviso será exibido. Passe o cursor 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 administrador, as ações são exibidas de acordo com as configurações de permissões. Isso significa que, em alguns casos, um usuário sem direitos de Super administrador não consegue visualizar a lista completa de ações devido a determinadas restrições de permissão. Uma ação é exibida para o usuário sem direitos de Super administrador se as seguintes condições forem atendidas:

- O usuário tem acesso de leitura e gravação a grupos de hosts, hosts, templates e triggers nas condições da ação
- O usuário tem acesso de leitura e gravação a grupos de hosts, hosts e templates nas operações da ação, operações de recuperação e operações de atualização
- O usuário tem acesso de leitura a grupos de usuários e usuários nas operações da ação, operações de recuperação e operações de atualização

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Enable* - altera o status da ação para *Enabled*
- *Disable* - altera o status da ação para *Disabled*
- *Delete* - exclui as ações

Para usar essas opções, marque as caixas de seleção antes das respectivas ações e, em seguida, 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.

Tipos de mídia

Visão geral

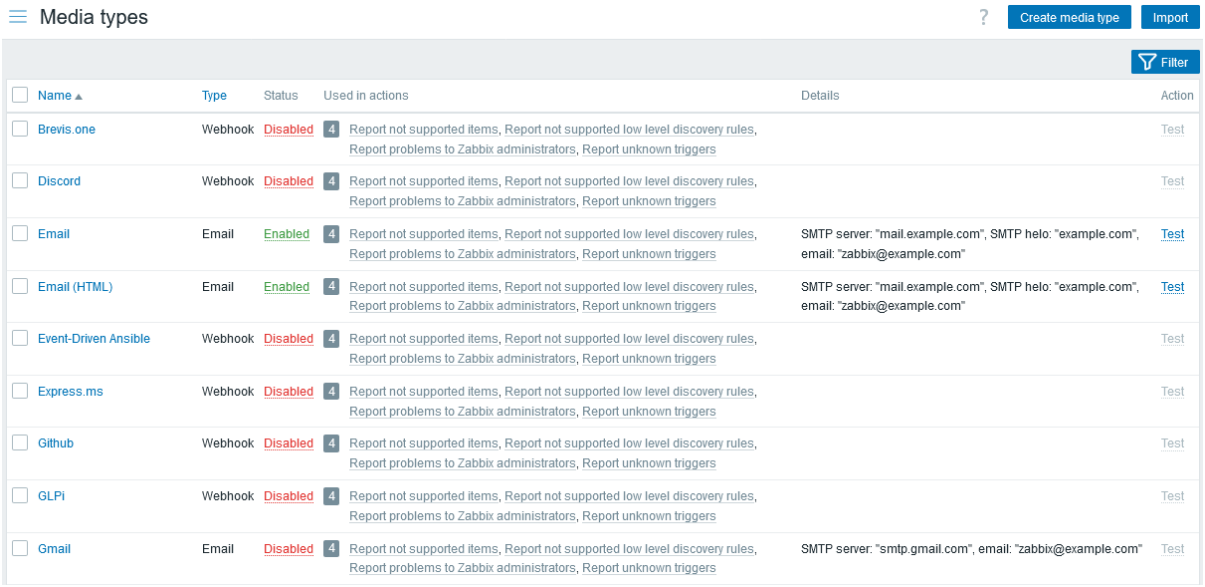
Na seção *Alertas* > *Tipos de mídia*, os usuários podem configurar e manter informações sobre tipos de mídia.

Este menu está disponível somente para usuários do tipo SuperAdmin.

As informações sobre o tipo de mídia contêm instruções gerais para usar uma mídia como canal de entrega de notificações. Detalhes específicos, como os endereços de e-mail individuais para os quais uma notificação deve ser enviada, são mantidos com os usuários individuais.

É exibida uma lista dos tipos de mídia existentes com seus detalhes.

Note:
Se os métodos de entrega (*E-mail*, *SMS*, *Webhook*, *Script* ou *Push*) forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend, os tipos de mídia pré-configurados correspondentes serão desabilitados nesta seção.



Dados exibidos:

Coluna	Descrição
Nome	Nome do tipo de mídia. Clicar no nome abre o formulário de configuração do tipo de mídia.
Tipo	O tipo de mídia (e-mail, SMS etc.) é exibido.
Status	O status do tipo de mídia é exibido: *Habilitado* ou *Desabilitado*. Clicar no status permite alterá-lo.
Usado em ações	São exibidas as ações nas quais o tipo de mídia é usado, 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	São exibidas informações detalhadas sobre o tipo de mídia.
Ações	A seguinte ação está disponível: **Testar** - clique para abrir um formulário de teste no qual 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. Consulte também: Teste de tipo 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:

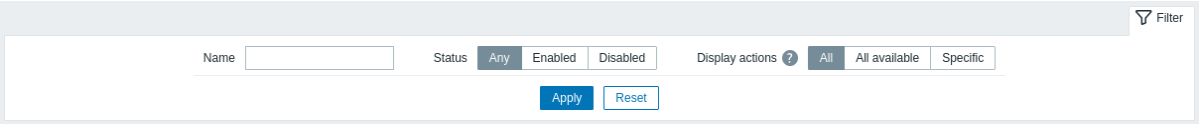
- *Enable* - altera o status do tipo de mídia para *Enabled*
- *Disable* - altera o status do tipo de mídia para *Disabled*
- *Export* - exporta os tipos de mídia para um arquivo YAML, XML ou JSON
- *Delete* - exclui os tipos de mídia

Para usar essas opções, marque as caixas de seleção antes dos respectivos tipos de mídia e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os tipos de mídia de seu interesse. Para melhor desempenho na busca, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de tipos de mídia. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar os tipos de mídia por nome e status. Além disso, você pode usar o filtro para exibir ações na coluna *Used in actions* com base no escopo do uso do tipo de mídia (definido pelo parâmetro *Send to media type* em detalhes da operação da ação).



Scripts

Visão geral

Na seção *Alertas* > *Scripts*, é possível configurar e manter scripts globais definidos pelo usuário.

Este menu está disponível somente para usuários do tipo SuperAdmin.

Os scripts globais, dependendo do escopo configurado e também das permissões do usuário, estão disponíveis para execução:

- No menu do host, em vários locais do frontend (*Dashboard*, *Problemas*, *Dados recentes*, *Mapas* etc.).
- No menu de eventos.
- Podem ser executados como uma operação de uma ação.

Os scripts são executados pelo Zabbix agent, pelo Zabbix server (proxy) ou somente pelo Zabbix server. Consulte também Execução de comandos.

Os comandos remotos são desabilitados por padrão tanto no Zabbix agent quanto no Zabbix proxy. Eles podem ser habilitados da seguinte forma:

- Para comandos remotos executados pelo Zabbix agent, adicione um parâmetro `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` para cada comando permitido na configuração do agent, em que `*` corresponde a qualquer um dos modos de execução (`wait` ou `nowait`).
- Para comandos remotos executados pelo Zabbix proxy, defina o parâmetro `EnableRemoteCommands` como 1 na configuração do proxy. Observe que não é necessário habilitar comandos remotos no Zabbix proxy quando eles são executados pelo Zabbix agent em um host monitorado pelo proxy.

Attention:
Não use `AllowKey=system.run[*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` a menos que seja necessário, pois isso permite a execução de qualquer comando no host. Por motivos de segurança, recomenda-se permitir somente comandos exatos, enquanto `system.run[*]` é negado por padrão (`DenyKey=system.run[*]`).

A execução de scripts globais no Zabbix server pode ser desabilitada definindo `EnableGlobalScripts=0` na configuração do server. Em novas instalações a partir do Zabbix 7.0, a execução de scripts globais no Zabbix server é desabilitada por padrão.

É exibida uma lista dos scripts existentes com seus detalhes.



Dados exibidos:

Coluna	Descrição
Nome	Nome do script. Clicar no nome do script abre o formulário de configuração do script.

Coluna	Descrição
Escopo	Escopo do script - operação de ação, ação manual do host ou ação manual do evento. Esta configuração determina onde o script está disponível - nos comandos remotos das operações de uma ação, no menu do host ou no menu de eventos, respectivamente.
Usado nas ações	São exibidas todas as ações nas quais o script é usado, 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	É exibido o tipo de script - comando *URL*, *Webhook*, *Script*, *SSH*, *Telnet* ou *IPMI*.
Executar em	É exibido se o script será executado no Zabbix agent, no Zabbix proxy ou no server, ou somente no Zabbix server.
Comandos	São exibidos todos os comandos a serem executados no script. Nada é exibido aqui para webhooks.
Grupo de usuários	É exibido o grupo de usuários para o qual o script está disponível (ou *Todos* para todos os grupos de usuários).
Grupo de hosts	É exibido o grupo de hosts para o qual o script está disponível (ou *Todos* para todos os grupos de hosts).
Acesso ao host	É exibido o nível de permissão para o grupo de hosts - *Leitura* ou *Escrita*. Somente 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:

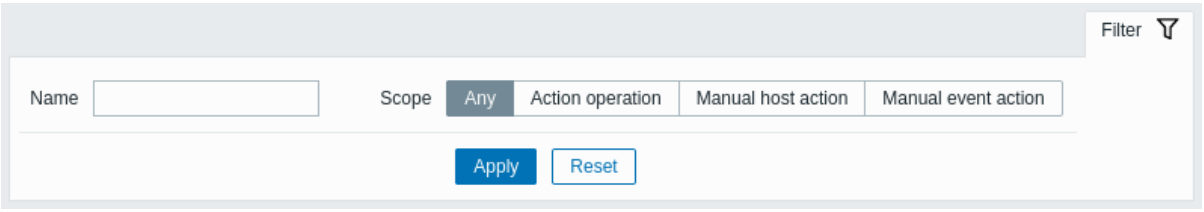
- *Delete* - excluir os scripts

Para usar essa opção, marque as caixas de seleção antes dos respectivos scripts e clique em *Delete*.

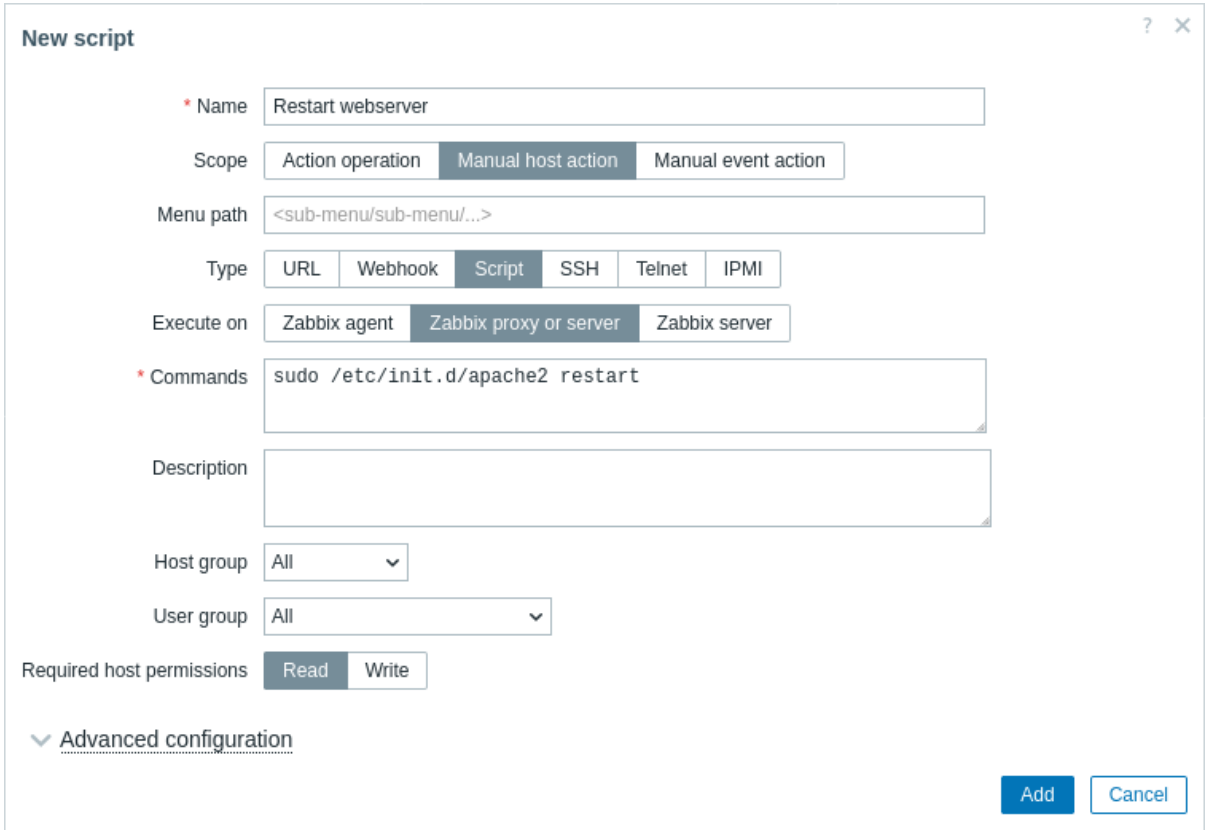
Usando o filtro

Você pode usar o filtro para exibir apenas os scripts de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidos.

O link *Filter* está disponível acima da lista de scripts. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar scripts por nome e escopo.



Configurando um script global



Note:
Recomenda-se usar funções de macro em vez de macros simples em scripts globais para aumentar a segurança, pois as macros não são escapadas automaticamente.

Atributos do script:

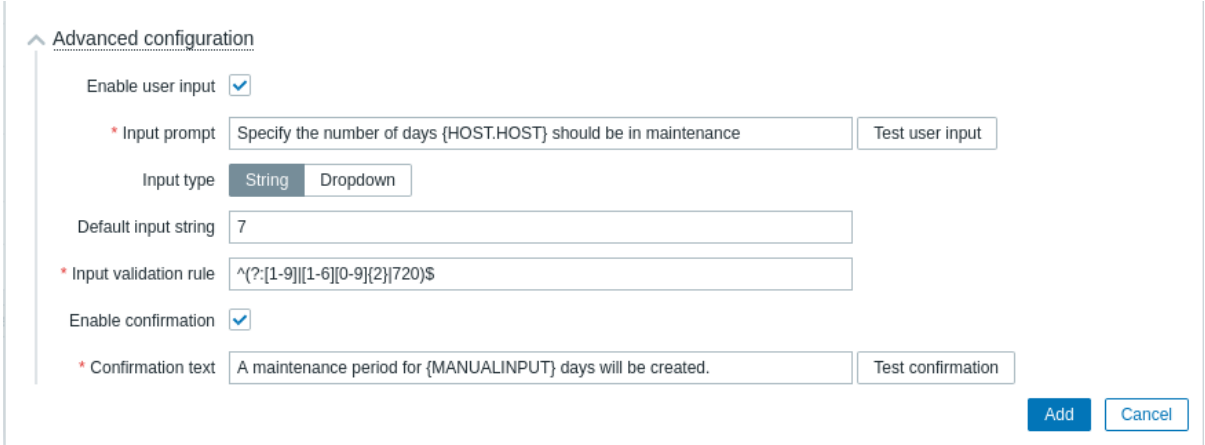
Parâmetro	Descrição
Nome	Nome exclusivo do script. Por exemplo, `Clear /tmp filesystem`
Escopo	Escopo do script: operação de ação, ação manual de host ou ação manual de evento. Essa configuração determina onde o script pode ser usado: em comandos remotos de operações de ação, no menu do host ou no menu do 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 usado em uma ação, seu escopo não poderá ser alterado de "operação de ação". **Suporte a macros** O escopo afeta o conjunto de macros disponíveis. Por exemplo, macros relacionadas ao usuário (`{USER.*}`) são compatíveis com scripts para permitir a transmissão de informações sobre o usuário que iniciou o script. No entanto, elas não são compatíveis quando o escopo do script é operação de ação, pois as operações de ação são executadas automaticamente. Uma macro `{MANUALINPUT}` permite especificar uma entrada manual no momento da execução do script. Ela é compatível com scripts de ação manual de host e de ação manual de evento. Para saber quais outras macros são compatíveis, pesquise por "Trigger-based notifications and commands/Trigger-based commands", "Manual host action scripts" e "Manual event action scripts" na tabela de macros compatíveis. Observe que, se uma macro puder ser resolvida para um valor com espaços (por exemplo, nome do host), não se esqueça de colocá-la entre aspas conforme necessário.
Caminho do menu	Caminho de menu desejado para o script. Por exemplo, `Default` ou `Default/` exibirá o script no diretório correspondente. Os menus podem ser aninhados, por exemplo, `Main menu/Sub menu1/Sub menu2`. Ao acessar scripts pelo menu do host/evento nas seções de monitoramento, eles serão organizados de acordo com os diretórios especificados. Este campo é exibido somente quando "Ação manual de host" ou "Ação manual de evento" está selecionado como *Escopo*.

Parâmetro	Descrição
Tipo	Clique no botão correspondente para selecionar o tipo de script: comando **URL**, **Webhook**, **Script**, **SSH**, **Telnet** ou **IPMI**. O tipo **URL** está disponível somente quando "Ação manual de host" ou "Ação manual de evento" está selecionado como *Escopo*.
Tipo de script: **URL**	
URL	Especifique a URL para acesso rápido pelo menu do host ou menu do evento. Macros e macros de usuário personalizadas são compatíveis. O suporte a macros depende do escopo do script (consulte *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 para URL.
Abrir em uma nova janela	Determina se a URL deve ser aberta em uma nova aba do navegador ou na mesma aba.
Tipo de script: **Webhook**	
Parâmetros	Especifique as variáveis do webhook como pares atributo-valor. Consulte também: configuração de mídia Webhook. Macros e macros de usuário personalizadas são compatíveis nos valores dos parâmetros. O suporte a macros depende do escopo do script (consulte *Escopo* acima).
Script	Insira o código JavaScript no editor modal que é aberto ao clicar no campo do parâmetro ou no ícone de lápis ao lado dele. O suporte a macros depende do escopo do script (consulte *Escopo* acima). Consulte também: configuração de mídia Webhook, Objetos JavaScript adicionais.
Tempo limite	Tempo limite de execução do JavaScript (1-60s, padrão de 30s). Sufixos de tempo são compatíveis (por exemplo, `30s`, `1m`).
Tipo de script: **Script**	
Executar em	Clique no botão correspondente para executar o script de 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 de o host ser monitorado por proxy ou server. Ele será executado no proxy se habilitado por EnableRemoteCommands. Ele será executado no server se os scripts globais estiverem habilitados pelo parâmetro do server EnableGlobalScripts. **Zabbix server** - o script será executado somente pelo Zabbix server. Esta opção não estará disponível se os scripts globais estiverem desabilitados pelo parâmetro do server EnableGlobalScripts.
Comandos	Insira o caminho completo dos comandos a serem executados no script. O suporte a macros depende do escopo do script (consulte *Escopo* acima). Macros de usuário personalizadas são compatíveis.
Tipo de script: **SSH**	
Método de autenticação	Selecione o método de autenticação: senha ou chave pública.
Nome de usuário	Insira o nome de usuário.
Senha	Insira a senha. Este campo está disponível se "Senha" estiver selecionado como método de autenticação.
Arquivo de chave pública	Insira o caminho para o arquivo de chave pública. Este campo está disponível se "Chave pública" estiver selecionado como método de autenticação.
Arquivo de chave privada	Insira o caminho para o arquivo de chave privada. Este campo está disponível se "Chave pública" estiver selecionado como método de autenticação.
Frase secreta	Insira a frase secreta. Este campo está disponível se "Chave pública" estiver selecionado como método de autenticação.
Porta	Insira a porta do serviço SSH remoto no host de destino à qual o Zabbix se conectará.
Comandos	Insira os comandos. O suporte a macros depende do escopo do script (consulte *Escopo* acima). Macros de usuário personalizadas são compatíveis.
Tipo de script: **Telnet**	

Parâmetro	Descrição
Nome de usuário	Insira o nome de usuário.
Senha	Insira a senha.
Porta	Insira a porta do serviço Telnet remoto no host de destino à qual o Zabbix se conectará.
Comandos	Insira os comandos. O suporte a macros depende do escopo do script (consulte *Escopo* acima). Macros de usuário personalizadas são compatíveis.
Tipo de script: **IPMI**	
Comando	Insira o comando IPMI. O suporte a macros depende do escopo do script (consulte *Escopo* acima). Macros de usuário personalizadas são compatíveis.
Descrição	Insira 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 somente quando "Ação manual de host" ou "Ação manual de evento" está selecionado como *Escopo*.
Permissões necessárias do host	Selecione o nível de permissão para o grupo de hosts: *Leitura* ou *Gravação*. Somente usuários com o nível de permissão necessário terão acesso à execução do script. Este campo é exibido somente quando "Ação manual de host" ou "Ação manual de evento" está selecionado como *Escopo*.
Configuração avançada	Clique no cabeçalho *Configuração avançada* para exibir as opções de configuração avançada. Este campo é exibido somente quando "Ação manual de host" ou "Ação manual de evento" está selecionado como *Escopo*.

Configuração avançada

As opções de configuração avançada estão disponíveis em uma seção recolhível de *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. Consulte também: Entrada manual do usuário.
Prompt de entrada	Insira um texto personalizado solicitando a entrada personalizada do usuário. Esse texto será exibido acima do campo de entrada no pop-up *Entrada manual*. Para visualizar o pop-up *Entrada manual*, clique em *Testar entrada do usuário*. A visualização também permite testar se a string de entrada está em conformidade com a regra de validação de entrada (consulte os parâmetros abaixo). O suporte a macros e macros de usuário depende do escopo do script (consulte *Escopo* nos parâmetros gerais de configuração do script).
Tipo de entrada	Selecione o tipo de entrada manual: **String** - string única; **Lista suspensa** - o valor é selecionado entre várias opções de lista suspensa.

Parâmetro	Descrição
Opções da lista suspensa	Insira valores exclusivos para a lista suspensa de entrada do usuário em uma lista delimitada por vírgulas. Para incluir uma opção vazia na lista suspensa, adicione uma vírgula extra no início, no meio ou no final da lista. Esse campo é exibido somente se *Lista suspensa* estiver selecionado como *Tipo de entrada*.
String de entrada padrão	Insira a string padrão para a entrada do usuário (ou nenhuma). Esse 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 pop-up *Entrada manual*. Esse campo é exibido somente se *String* estiver 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 compatíveis. Esse campo é exibido somente se *String* estiver selecionado como *Tipo de entrada*.
Habilitar confirmação	Marque a caixa de seleção para exibir uma mensagem de confirmação antes de executar o script. Esse recurso pode ser especialmente útil para operações potencialmente perigosas (como um script de reinicialização) ou operações que podem levar muito tempo.
Texto de confirmação	Insira um texto de confirmação personalizado para o pop-up de confirmação habilitado com a caixa de seleção acima (por exemplo, *O sistema remoto será reiniciado. Tem certeza?*). Para ver como o texto será exibido, clique em *Testar confirmação* ao lado do campo. Macros e macros de usuário personalizadas são compatíveis. *Observação:* as macros não serão expandidas ao testar a mensagem de confirmação.

Se a entrada manual do usuário e uma mensagem de confirmação estiverem configuradas, elas serão exibidas em janelas pop-up 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 diferença em apenas um parâmetro.

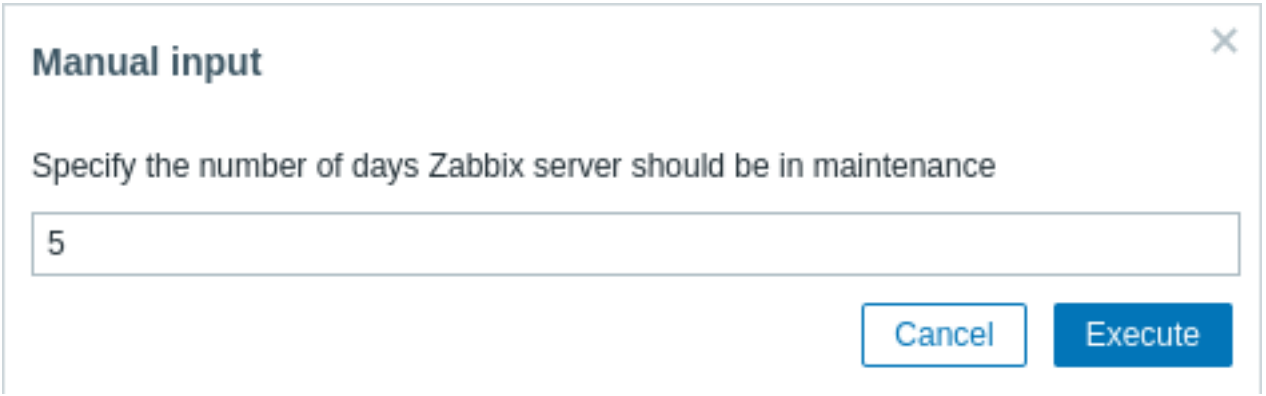
Por exemplo, você pode querer fornecer um número inteiro diferente ou um endereço URL diferente ao script durante a execução.

Para habilitar a entrada manual do usuário:

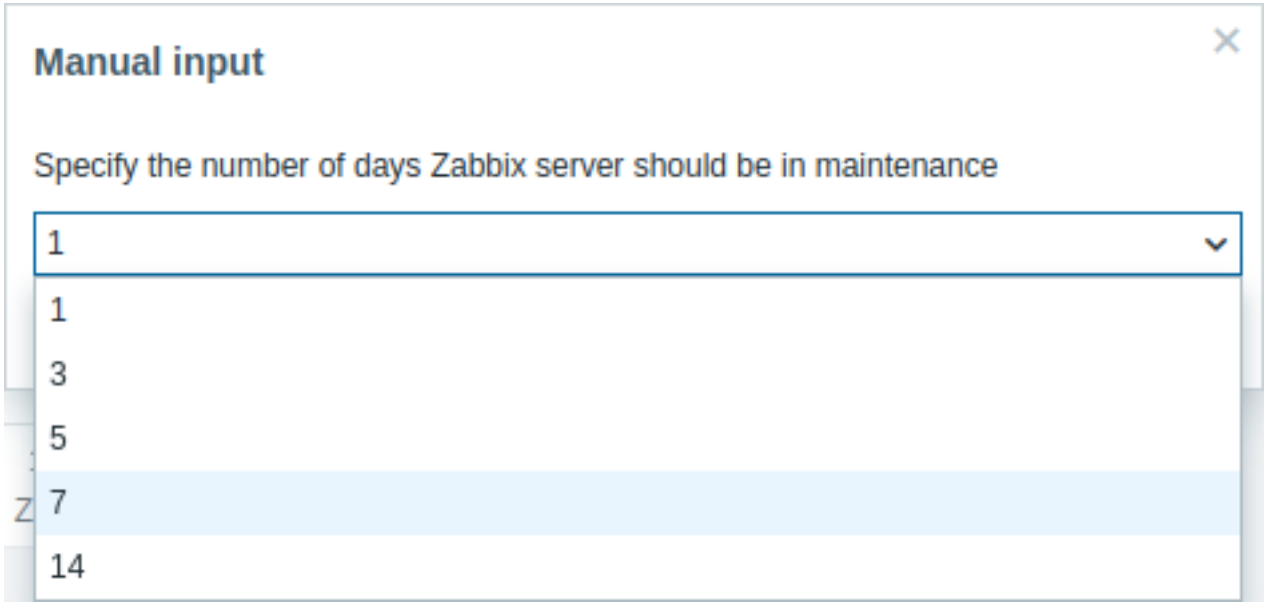
- use a macro {MANUALINPUT} no script (comandos, script, parâmetro do script), onde necessário; ou no campo URL dos scripts de URL;
- em configuração avançada do script, habilite a entrada manual do usuário e configure as opções de entrada.

Com a entrada do usuário habilitada, antes da execução do script, uma janela pop-up de *Entrada manual* será exibida para que o usuário forneça um valor personalizado. O valor fornecido substituirá {MANUALINPUT} no script.

Dependendo da configuração, será solicitado que o usuário insira um valor de string:



Ou selecione o valor em uma lista suspensa de opções predefinidas:



A entrada manual do usuário está disponível somente para scripts cujo escopo seja 'Ação manual do host' ou 'Ação manual do evento'.

Execução do script e resultado

Os scripts executados pelo Zabbix server são executados na ordem descrita na página de execução de comandos.

O resultado do script é exibido em uma janela pop-up que aparece após a execução do script. O valor de retorno do script é uma saída padrão:

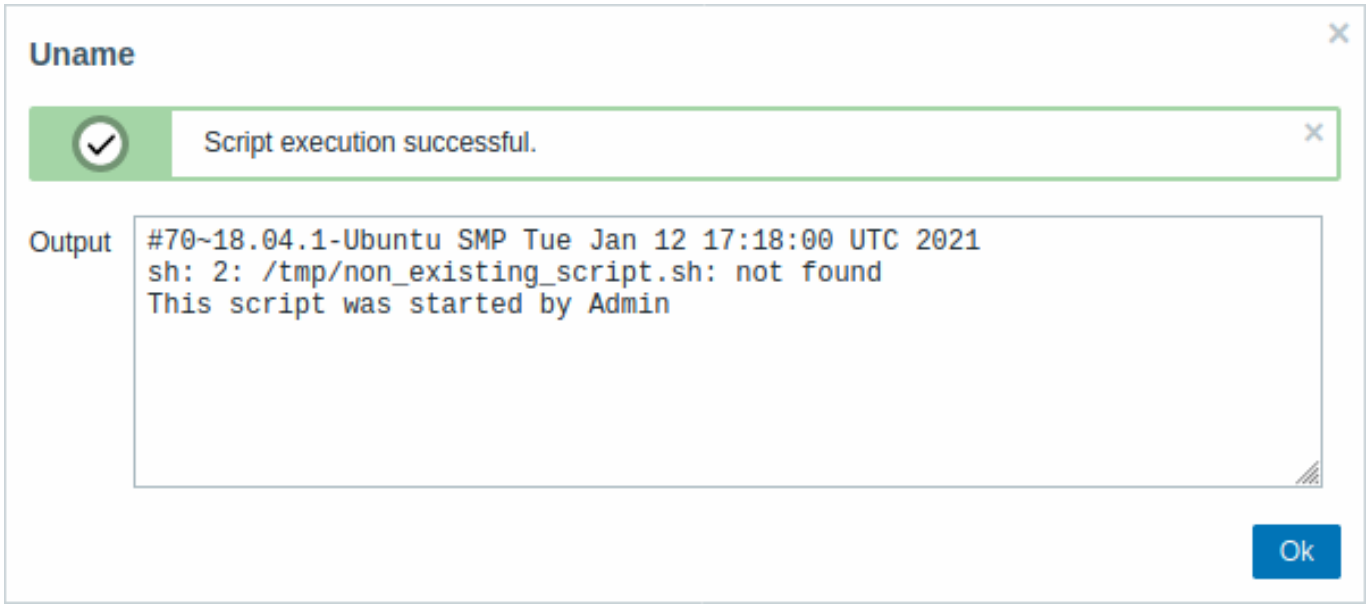
- Se o script for finalizado com sucesso (código de saída 0), o valor de retorno é limitado a 16MB (incluindo espaços em branco à direita, que são truncados).
- Se o script for finalizado com erro (código de saída diferente de zero), o valor de retorno é um erro padrão limitado a 2KB.

O Zabbix não armazena a saída estendida do script por padrão. Para preservar todos os detalhes da saída, você pode implementar o registro de logs dentro do próprio script (por exemplo, redirecionar a saída para um arquivo de log local).

Observe que, para scripts executados no Zabbix server ou no 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}"
```



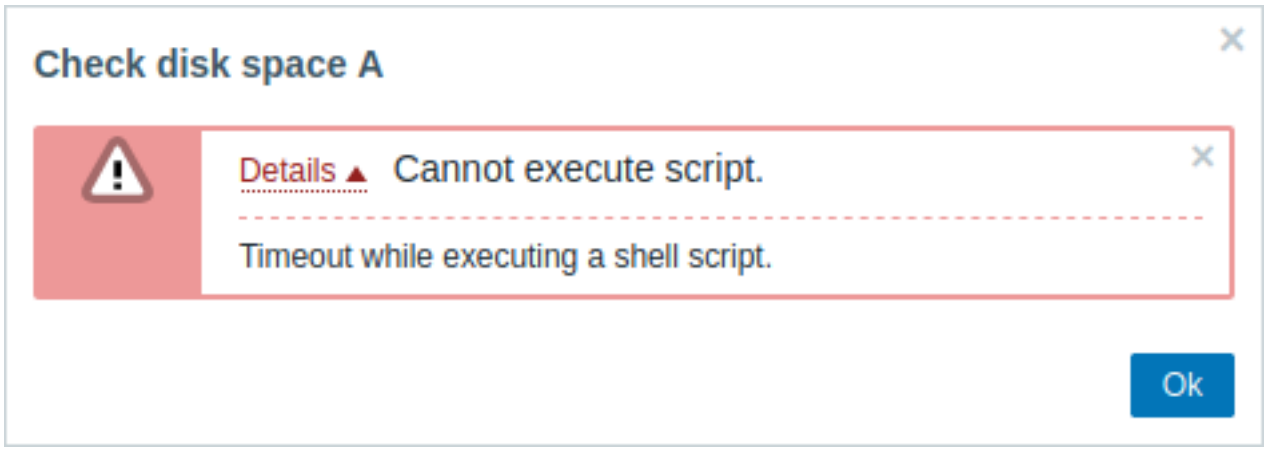
O resultado do script não exibe o próprio script.

Tempo limite do script no Zabbix agent

Você pode se deparar com uma situação em que ocorre um tempo limite durante a execução de um script.

Veja abaixo um exemplo de um script em execução no Zabbix agent e a janela de resultado:

```
sleep 5
df -h
```



A mensagem de erro, nesse caso, é a seguinte:

```
Timeout while executing a shell script.
```

Para evitar essas situações, recomenda-se otimizar o próprio script (no exemplo acima, o valor ”5”) em vez de ajustar o parâmetro `Timeout` na configuração do agent do Zabbix e na configuração do server do Zabbix.

Caso o parâmetro `Timeout` tenha sido alterado na configuração do agent do Zabbix, a seguinte mensagem de erro será exibida:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Isso significa que a modificação foi feita na configuração do agent do Zabbix, mas também é necessário modificar o parâmetro `Timeout` na configuração do server.

Tempo limite de script em agents ativos

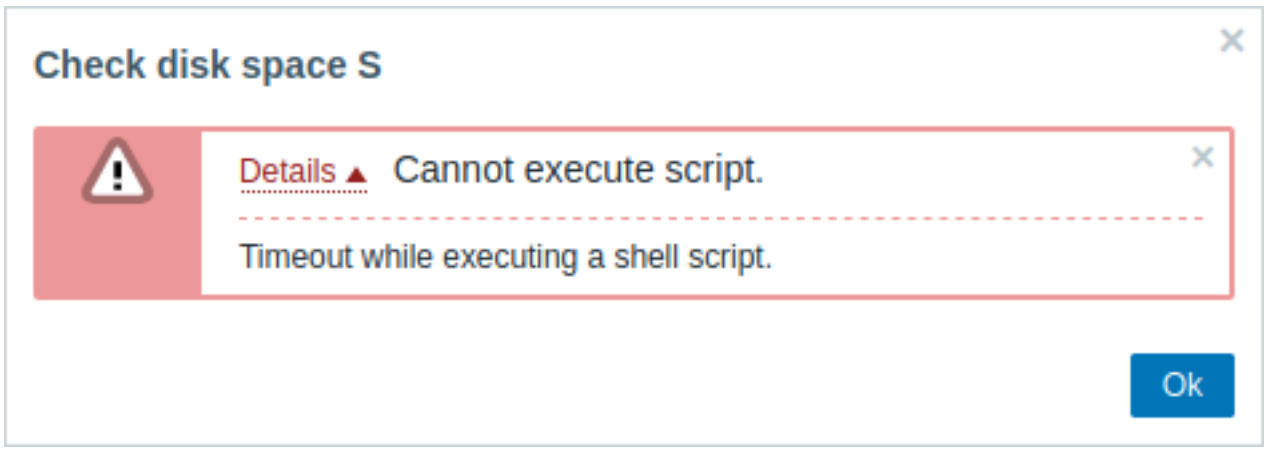
É possível executar scripts em agents que operam apenas no modo ativo. Depois que a execução do script é acionada pela operation da ação ou pela execução manual do script, o comando é incluído na configuração de verificação ativa e executado assim que o agent ativo o recebe.

Scripts manuais são enviados ao agent ativo juntamente com o timeout do server/proxy para execução do script. Aumente o timeout padrão do server/proxy para execução do script. O timeout deve ser maior que a frequência de atualização da verificação ativa; caso contrário, o timeout será excedido antes que o agent ativo receba o script e consiga retornar o resultado.

Tempo limite do script no Zabbix server/proxy

Veja abaixo um exemplo de um script em execução no Zabbix server e a janela de resultados:

```
sleep 11
df -h
```



Também é recomendável otimizar o próprio script (em vez de ajustar o parâmetro `TrapperTimeout` para um valor correspondente (no nosso caso, > 11), modificando a configuração do server do Zabbix).

Usuários

Visão geral

Este menu apresenta seções relacionadas à configuração de usuários no Zabbix. Este menu está disponível apenas para usuários do tipo SuperAdmin.

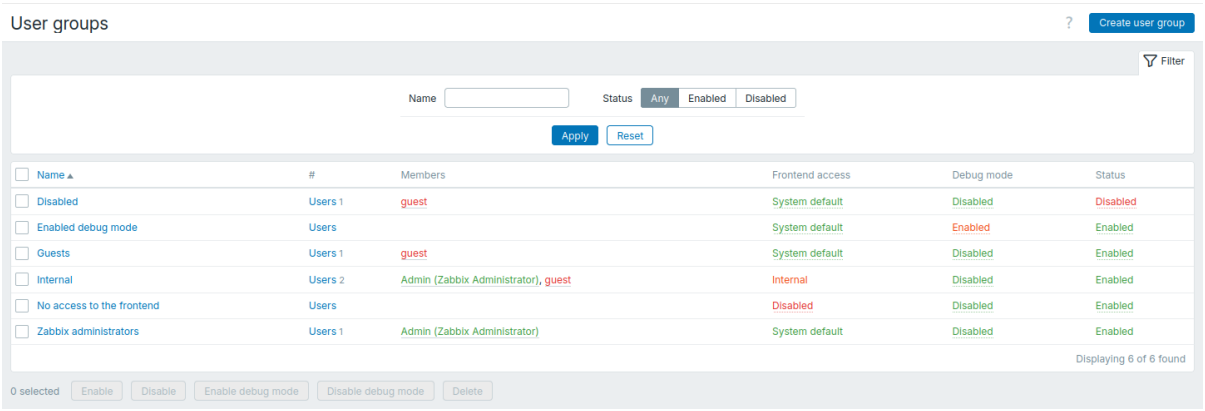
Grupos de usuários

Visão geral

Na seção *Users* > *User groups*, os grupos de usuários do sistema são mantidos.

Grupos de usuários

É exibida uma lista dos grupos de usuários existentes com seus detalhes.



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. Os usuários de grupos desabilitados são exibidos em vermelho.
Acesso ao frontend	O nível de acesso ao frontend é exibido: **Padrão do sistema** - os usuários são autenticados pelo Zabbix, LDAP ou HTTP (dependendo do método de autenticação definido globalmente); **Interno** - os usuários são autenticados pelo Zabbix; ignorado se a autenticação HTTP for o padrão global; **LDAP** - os usuários são autenticados pelo LDAP; ignorado se a autenticação HTTP for o padrão global; **Desabilitado** - o acesso ao frontend do Zabbix é proibido para este grupo. Clicando no nível atual, você pode alterá-lo.
Modo de depuração	O status do modo de depuração é exibido - *Habilitado* ou *Desabilitado*. Clicando no status, você pode alterá-lo.
Status	O status do grupo de usuários é exibido - *Habilitado* ou *Desabilitado*. Clicando no status, você pode alterá-lo.

Para configurar um novo grupo de usuários, clique no botão *Criar grupo de usuários* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

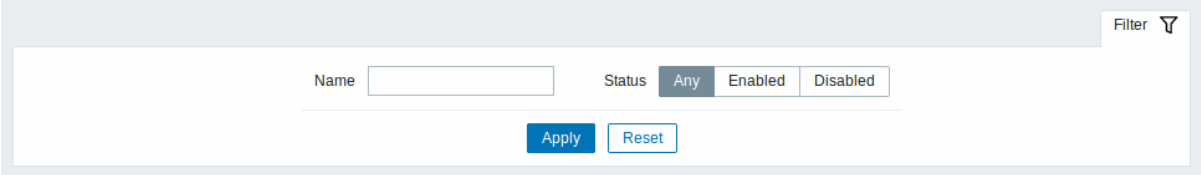
- *Enable* - altera o status do grupo de usuários para *Enabled*
- *Disable* - altera o status do grupo de usuários para *Disabled*
- *Enable debug mode* - ativa o modo de depuração para os grupos de usuários
- *Disable debug mode* - desativa o modo de depuração para os grupos de usuários
- *Delete* - exclui os grupos de usuários

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de usuários e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os grupos de usuários de seu interesse. Para melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filter* está disponível acima da lista de grupos de usuários. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar grupos de usuários por nome e status.



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*, *Administrador*, *Super administrador*).

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 ampliadas além do que está disponível para esse tipo de usuário.

As caixas de seleção das permissões indisponíveis ficam desabilitadas; os usuários não poderão acessar o elemento nem mesmo inserindo diretamente no navegador a URL desse elemento.

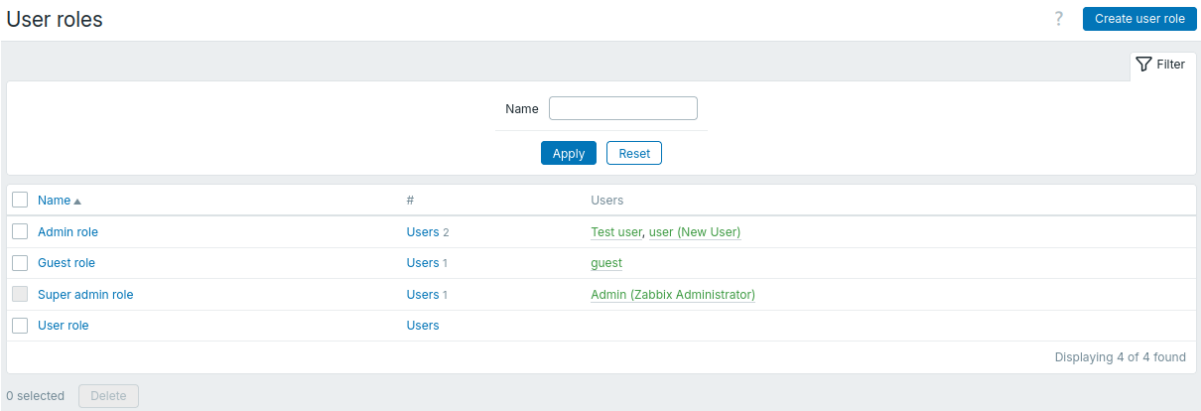
Note:
Restringir o acesso a alguns elementos da interface do usuário impede apenas a abertura dessa página — isso não remove a possibilidade de acessar os dados subjacentes em outras partes da interface.

As funções de usuário podem ser atribuídas a usuários do sistema. Cada usuário pode ter apenas uma função atribuída.

Funções de usuário padrão

Por padrão, o Zabbix é configurado com quatro funções de usuário, que têm um conjunto predefinido de permissões:

- Função Guest
- Função User
- Função Admin
- Função Super admin



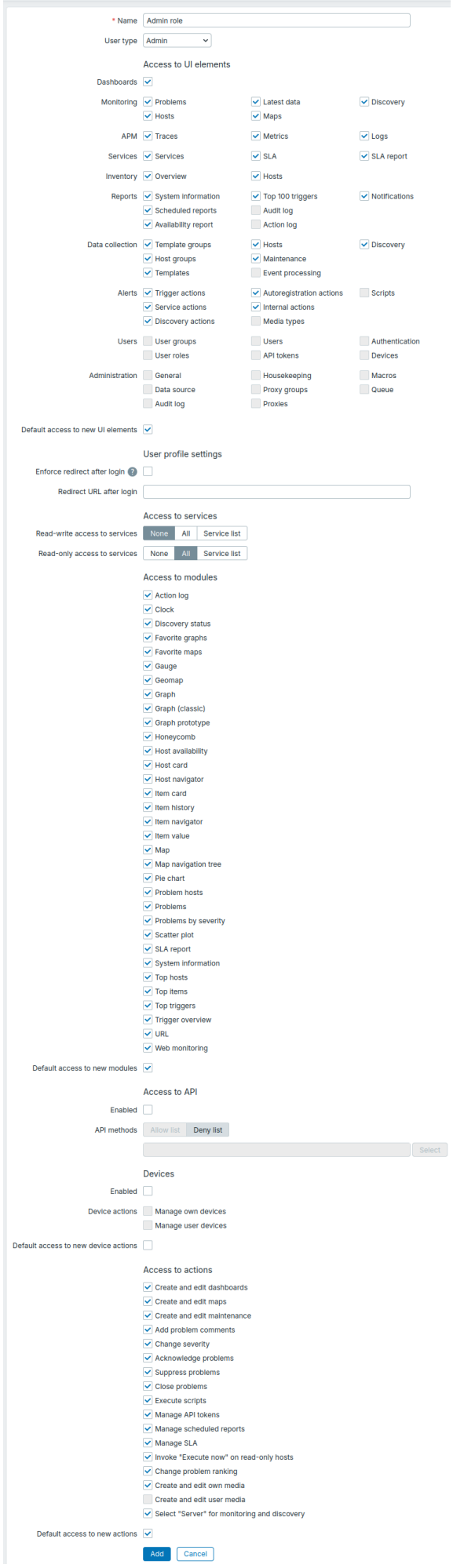
Essas funções são baseadas nos principais tipos de usuário do Zabbix. É exibida a lista de todos os usuários atribuídos à respectiva função. Os usuários incluídos em grupos desabilitados são exibidos em vermelho. A *função Guest* é uma função do tipo de usuário que tem apenas permissões para visualizar algumas seções do frontend.

Note:
A *função Super admin* padrão não pode ser modificada nem excluída, pois deve existir no Zabbix pelo menos um usuário Super admin com privilégios ilimitados. Usuários do tipo *Super admin* podem modificar as configurações da própria função, mas não o tipo de usuário.

Configuração

Para criar uma nova função, clique no botão *Create user role* no canto superior direito. Para atualizar uma função existente, clique no nome da função para abrir o formulário de configuração.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

As permissões disponíveis são exibidas. Para revogar uma determinada permissão, desmarque sua caixa de seleção.

As permissões disponíveis, juntamente com os padrões de 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. Consulte a página Permissões para obter detalhes.

Acesso a outras opções

Algumas das seções descritas na tabela abaixo podem estar ocultas:

- *Acesso aos módulos* fica oculto se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend.
- *Dispositivos* fica oculto se `EnableMobileDevices` estiver desabilitado.

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 existentes do menu dos módulos não são afetadas.	Sim	Sim	Sim	Sim
Acesso aos serviços					
Acesso de leitura e gravação aos serviços	Selecione o acesso de leitura e gravação aos serviços: **Nenhum** - nenhum acesso **Todos** - acesso de leitura e gravação a todos os serviços **Lista de serviços** - selecione os serviços para acesso de leitura e gravação O acesso de leitura e gravação, quando 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 aos serviços com tag	Especifique o nome da tag e, opcionalmente, o valor para conceder adicionalmente acesso de leitura e gravação aos serviços correspondentes à tag. Esta opção está disponível se *Lista de serviços* estiver selecionada no parâmetro *Acesso de leitura e gravação aos serviços*. O acesso de leitura e gravação, quando concedido, tem precedência sobre as configurações de acesso somente leitura e é herdado dinamicamente pelos serviços filhos.				

Acesso somente leitura aos serviços	Selecione o acesso somente leitura aos serviços: **Nenhum** - nenhum acesso **Todos** - acesso somente leitura a todos os serviços **Lista de serviços** - selecione os serviços para acesso somente leitura O acesso somente leitura não tem precedência sobre o acesso de leitura e gravação e é herdado dinamicamente pelos serviços filhos.			Todos	Todos
Acesso somente leitura aos serviços com tag	Especifique o nome da tag e, opcionalmente, o valor para conceder adicionalmente acesso somente leitura aos serviços correspondentes à tag. Esta opção está disponível se *Lista de serviços* estiver selecionada no parâmetro *Acesso somente leitura aos serviços*. O acesso somente leitura não tem precedência sobre o acesso de leitura e gravação e é herdado dinamicamente pelos serviços filhos.				
Acesso aos módulos					
<Nome do módulo>	Permitir ou negar acesso a um módulo específico. Somente os módulos habilitados são exibidos nesta seção. Não é possível conceder ou restringir o acesso a um módulo que esteja desabilitado no momento.	Sim	Sim	Sim	Sim
Acesso padrão a novos módulos	Esta opção especifica como os novos módulos e widgets estarão acessíveis após uma atualização do Zabbix. Ela também se aplica aos módulos e widgets adicionados na seção *Administração > Geral > Módulos*.				
Acesso à API					
Habilitado	Habilitar/desabilitar o acesso à API.	Não	Não	Não	Não
Métodos da API	Selecione *Lista de permissões* para permitir ou *Lista de negações* 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 clicar no botão *Selecionar* e selecionar métodos na lista completa disponível para este tipo de usuário. Observe que, se determinada ação do bloco Acesso às ações estiver desmarcada, os usuários não poderão usar os métodos da API relacionados a essa ação. Caracteres curinga são compatíveis. Exemplos: `dashboard.*` (todos os métodos do serviço de API 'dashboard.') `*` (qualquer método), `*.export` (métodos com o nome '.export' de todos os serviços da API). Se nenhum método tiver sido especificado, a regra de *Lista de permissões/negações* será ignorada.				

Acesso às ações					
Criar e editar painéis	Desmarcar esta caixa de seleção também revogará os direitos de usar os métodos da API `.create`, `.update` e `.delete` para os elementos correspondentes.	Sim	Sim	Sim	Não
Criar e editar mapas					
Criar e editar manutenção				Não	
Adicionar comentários aos problemas	Desmarcar esta caixa de seleção também revogará os direitos de executar a ação correspondente por meio do método da API `event.acknowledge`.			Sim	
Alterar a severidade					
Confirmar problemas					
Suprimir problemas					
Fechar problemas					
Executar scripts	Desmarcar esta caixa de seleção também revogará os direitos de usar o método da API `script.execute`.				
Gerenciar tokens da API	Desmarcar esta caixa de seleção também revogará os direitos de usar todos os métodos da API `token.`.				
Gerenciar relatórios agendados	Desmarcar esta caixa de seleção 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 que os usuários usem a opção "Executar agora" em Dados recentes para itens em hosts somente leitura.			Sim	
Alterar a classificação dos problemas	Permitir que os usuários alterem a classificação do problema de causa para sintoma e vice-versa.				
Criar e editar a própria mídia	Permitir que os usuários criem e editem sua própria mídia.				
Criar e editar a mídia de usuários	Permitir que os usuários criem e editem mídias de outros usuários. Esta opção está disponível somente para usuários superadministradores.		Não	Não	
Selecionar "Server" para monitoramento e descoberta	Desmarcar esta caixa de seleção restringe a criação de hosts monitorados pelo server e de regras de descoberta de rede executadas pelo server, bem como o teste de itens com o server. A opção *Server* será desabilitada na configuração do host para o parâmetro *Monitorado por*, na configuração da regra de descoberta de rede para o parâmetro *Descoberta por* e no teste de itens para o parâmetro *Testar com*.		Sim	Não	
Acesso padrão a novas ações	Esta opção especifica como as novas ações estarão acessíveis após uma atualização do Zabbix.		Sim	Sim	
Dispositivos					
Habilitado	Habilitar/desabilitar o acesso a *Configurações do usuário* > *Dispositivos* (acessível a todos os usuários) e *Usuários* > *Dispositivos* (acessível somente a superadministradores).	Sim	Não	Não	Não
Gerenciar os próprios dispositivos	Permitir que os usuários gerenciem seus próprios dispositivos em *Configurações do usuário* > *Dispositivos*.				
Gerenciar dispositivos de usuários	Permitir que os usuários gerenciem dispositivos pertencentes a outros usuários em *Usuários* > *Dispositivos*.				

Acesso padrão a novas ações	Esta opção especifica como as novas ações estarão acessíveis após uma atualização do Zabbix.

Veja também:

- Configurando um usuário

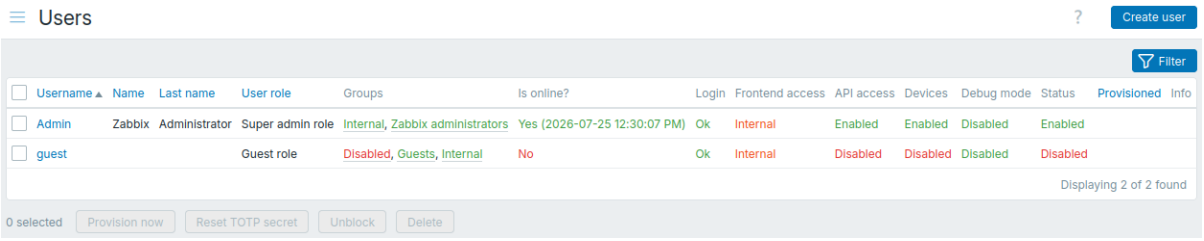
Usuários

Visão geral

Na seção *Usuários* > *Usuários*, os usuários do sistema são gerenciados.

Usuários

Uma lista dos usuários existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome de usuário	Nome de usuário para fazer 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 de usuários. Os 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 bloqueado temporariamente ao exceder o número de tentativas de login malsucedidas definido na seção *Administração > Geral > Outros* (cinco por padrão). Ao clicar 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 da configuração definida para todo o grupo de usuários.
Acesso à API	O status de acesso à API é exibido - *Habilitado* ou *Desabilitado*, dependendo da configuração definida para a função do usuário.
Dispositivos	O status de acesso aos dispositivos é exibido - *Habilitado* ou *Desabilitado*, dependendo da configuração definida para a função do usuário. Esta coluna é exibida se `EnableMobileDevices` estiver habilitado.
Modo de depuração	O status do modo de depuração é exibido - *Habilitado* ou *Desabilitado*, dependendo da configuração definida para todo o grupo de usuários.
Status	O status do usuário é exibido - *Habilitado* ou *Desabilitado*, dependendo da configuração definida para todo o grupo de usuários.
Provisionado	A data em que o usuário foi provisionado pela última vez é exibida. Usado para usuários criados pelo provisionamento JIT a partir de LDAP/SAML.
Informações	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:

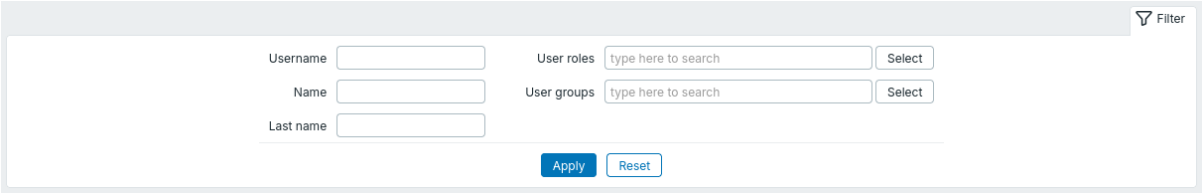
- *Provisionar agora* - atualiza as informações do usuário a partir do LDAP (esta opção só fica habilitada se um usuário LDAP estiver selecionado)
- *Redefinir segredo TOTP* - redefine os segredos TOTP do usuário para todos os métodos TOTP e exclui a sessão do usuário (esta opção só fica habilitada se a MFA estiver habilitada; para usuários sem segredos TOTP, a 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 usuários correspondentes e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os usuários de seu interesse. Para melhorar o desempenho da pesquisa, os dados são pesquisados com as macros não resolvidas.

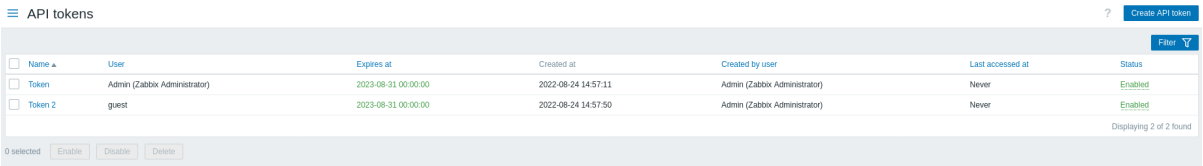
O link *Filtro* está disponível acima da lista de usuários. Ao clicar nele, um filtro ficará disponível, permitindo filtrar os usuários por nome de usuário, nome, sobrenome, função de usuário e grupo de usuários.



Tokens de API

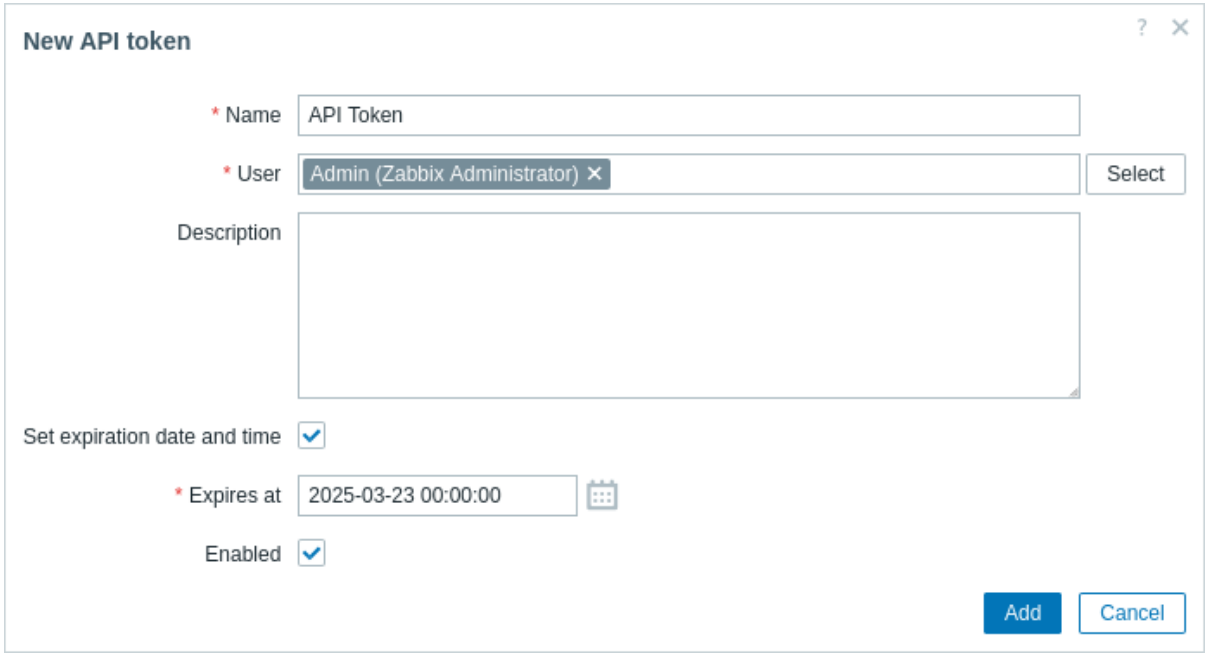
Visão geral

Esta seção permite criar e gerenciar tokens de API.
 Os tokens de API são usados para autenticação na API do Zabbix.
 Cada token de API é atribuído a um único usuário e herda as permissões desse usuário. As solicitações autenticadas com um token de API são autorizadas de acordo com a função e as permissões do usuário associado.



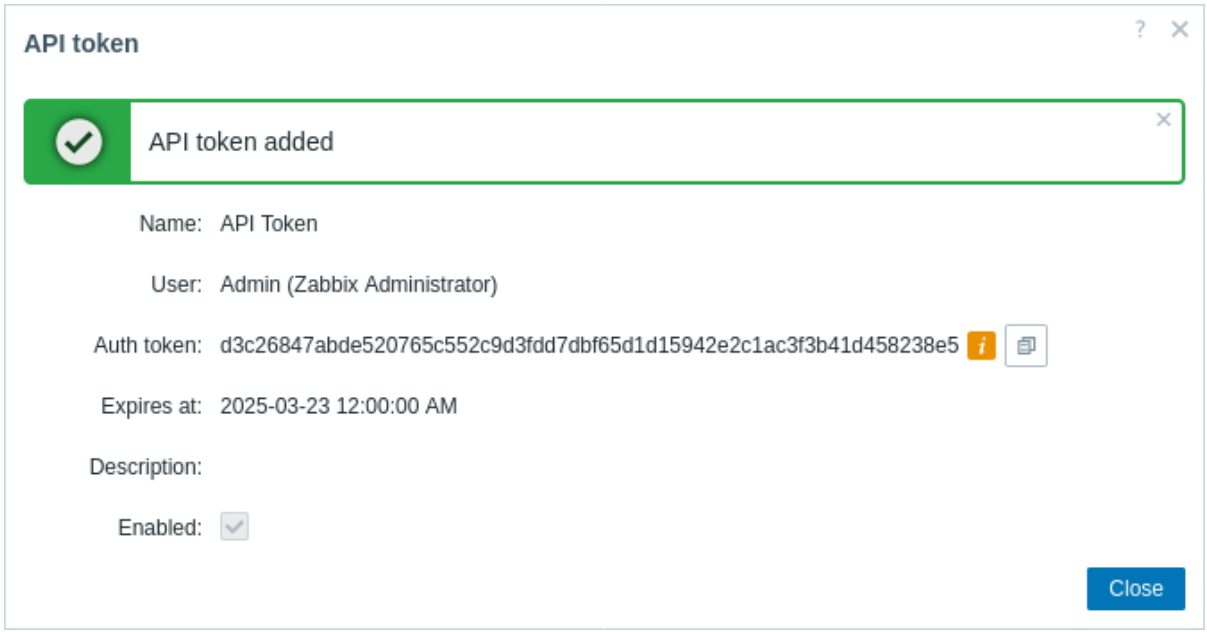
Você pode filtrar os tokens de API por nome, usuários aos quais os tokens estão atribuídos, data de expiração, usuários que criaram os tokens ou status (ativado/desativado). Clique no status do token na lista para ativar/desativar rapidamente um token. Você também pode ativar/desativar vários tokens de uma só vez, selecionando-os na lista e clicando nos botões *Enable/Disable* abaixo da lista.

Para criar um novo token, clique no botão *Create API token* no canto superior direito e preencha os campos obrigatórios na tela de configuração do token:



Parâmetro	Descrição
Name	Nome visível do token.
User	Usuário ao qual o token deve ser atribuído. Para selecionar rapidamente um usuário, comece a digitar o nome de usuário, o nome ou o sobrenome e selecione o usuário desejado na lista de preenchimento automático. Como alternativa, você pode clicar no botão *Select* e selecionar um usuário na lista completa de usuários. Um token pode ser atribuído a apenas um usuário.
Description	Descrição opcional do token.
Set expiration date and time	Desmarque esta caixa de seleção se o token não deve ter uma data de expiração.
Expiry date	Clique no ícone de calendário para selecionar a data de expiração do token ou insira a data manualmente no formato YYYY-MM-DD hh:mm:ss
Enabled	Desmarque esta caixa de seleção se precisar criar um token desativado.

Clique em *Add* para criar um token.



Copie o valor de *Auth token* e salve-o em um local seguro **antes de fechar a página**; em seguida, clique em *Close*. O token aparecerá na lista.

Warning:
O valor de *Auth token* não poderá ser visualizado novamente. Ele fica disponível somente imediatamente após a criação de um token. Se você perder um token salvo, será necessário gerá-lo novamente, e isso criará uma nova string de autorização.

Clique no nome do token para editar o nome, a descrição, as configurações da data de expiração ou o status do token. Observe que não é possível alterar o usuário ao qual o token está atribuído. Clique no botão *Update* para salvar as alterações. Se um token tiver sido perdido ou exposto, você poderá clicar no botão *Regenerate* para gerar um novo valor de token. Uma caixa de diálogo de confirmação será exibida, solicitando que você confirme esta operação, pois, após prosseguir, o token gerado anteriormente se tornará inválido.

Usuários sem acesso à seção de menu *Administration* podem visualizar e modificar os detalhes dos tokens atribuídos a eles na *User profile* > *API tokens* seção somente se *Manage API tokens* estiver permitido nas permissões da sua função de usuário.

Autenticação

Visão geral

A seção *Usuários* > *Autenticação* permite especificar o método de autenticação de usuários do Zabbix e os requisitos de senha interna.

Os métodos de autenticação disponíveis são autenticação interna, HTTP, LDAP, SAML e MFA.

Autenticação padrão

Por padrão, o Zabbix usa a autenticação **interna** do Zabbix para todos os usuários.

É possível alterar o método de autenticação padrão para o sistema LDAP. Para isso, acesse a aba *LDAP* e configure os parâmetros do LDAP. Em seguida, retorne à aba *Authentication* e altere o seletor *Default authentication* 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 poderão ser autenticados pelo Zabbix. Esses grupos devem ter o acesso ao frontend definido como Internal.

Também é possível habilitar a autenticação LDAP somente para grupos de usuários específicos, se a autenticação interna for usada globalmente. Nesse caso, os detalhes da autenticação LDAP podem ser especificados e usados para grupos de usuários específicos, cujo acesso ao frontend deverá ser definido como LDAP. Se um usuário estiver incluído em pelo menos um grupo de usuários com autenticação LDAP, ele 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.

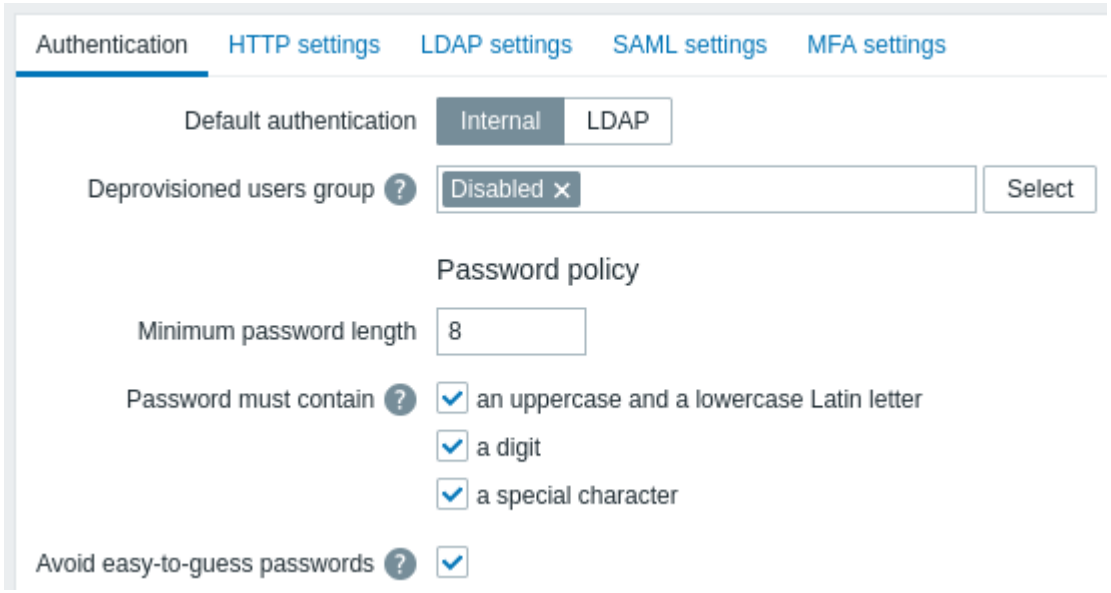
O Zabbix oferece suporte ao provisionamento just-in-time (JIT), que permite criar contas de usuário no Zabbix na primeira vez que um usuário externo é autenticado e provisionar essas contas de usuário. O provisionamento JIT é compatível com LDAP e SAML.

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: *Interna* ou *LDAP*.
Grupo de usuários desprovisionados	Especifique um grupo de usuários para usuários desprovisionados. Essa configuração é necessária somente para o provisionamento JIT, em relação aos usuários que foram criados no Zabbix a partir de sistemas LDAP ou SAML, mas que não precisam mais ser provisionados. É necessário especificar um grupo de usuários desabilitado.
Comprimento mínimo da senha	Por padrão, o comprimento mínimo da senha é definido como 8. Intervalo compatível: 1-70. Observe que as senhas com mais de 72 caracteres serão truncadas.
A senha deve conter	Marque uma ou mais caixas de seleção para exigir o uso dos caracteres especificados em uma senha: - uma letra latina maiúscula e uma minúscula - um dígito - um caractere especial Passe o cursor sobre o ponto de interrogação para ver uma dica com a lista de caracteres de cada opção.
Evitar senhas fáceis de adivinhar	Se marcada, uma senha será verificada de acordo com os 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 "Top 100k passwords" da NCSC, da lista "Top 1M passwords" da SecLists e da lista de senhas específicas do contexto do Zabbix. Usuários internos não poderão definir senhas incluídas nessa lista, pois essas senhas são consideradas fracas devido ao seu uso comum.

As alterações nos requisitos de complexidade de senha não afetarão as senhas de usuários existentes, mas, se um usuário existente optar por alterar uma senha, a nova senha deverá atender aos requisitos atuais. Uma dica com a lista de requisitos será exibida ao lado do campo *Senha* no perfil do usuário e no formulário de configuração do usuário, acessível pelo menu *Usuários* > *Usuários*.

HTTP

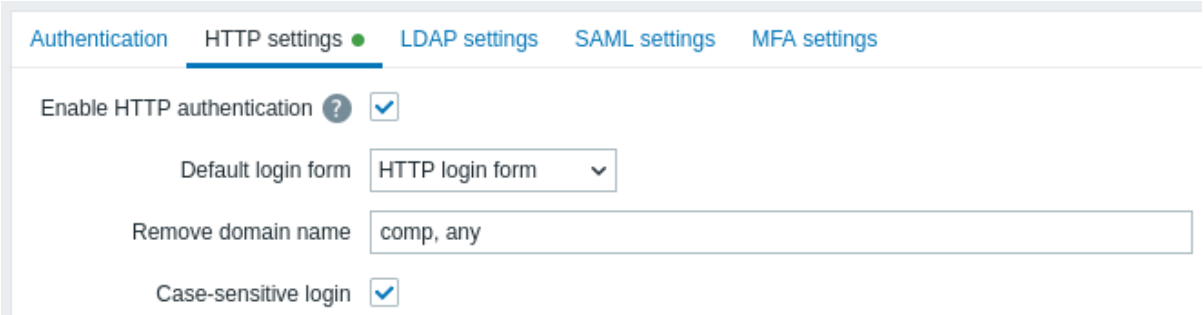
Visão geral

A autenticação baseada em HTTP ou em um 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; no entanto, a senha do Zabbix não será usada.

Attention:
Atenção! Certifique-se de que a autenticação do servidor web esteja configurada e funcionando corretamente antes de ativá-la.

Note:
A configuração da autenticação HTTP pode ser desativada definindo `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` no arquivo de configuração do frontend (`zabbix.conf.php`). Nesse caso, a aba *Configurações HTTP*, com as opções de autenticação HTTP, ficará oculta.

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 informando que, no caso de autenticação pelo web server, todos os usuários (mesmo aqueles com acesso ao frontend definido como LDAP/Interno) serão autenticados pelo web server, e não pelo Zabbix.
Formulário de login padrão	Especifique se os usuários não autenticados devem ser direcionados para: **Formulário de login do Zabbix** - página de login padrão do Zabbix. **Formulário de login HTTP** - página de login HTTP. É recomendável habilitar a autenticação baseada no web server somente para a página `index_http.php`. Se *Formulário de login padrão* estiver definido como 'Página de login HTTP', o usuário será conectado automaticamente se o módulo de autenticação do web server definir um login de usuário válido na variável `$_SERVER`. As chaves `$_SERVER` compatíveis são `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Remover nome de domínio	Uma lista de nomes de domínio, separados por vírgulas, que devem ser removidos do nome de usuário. Por exemplo, `comp,any` - se o nome de usuário for 'Admin@any' ou 'comp\Admin', o usuário será conectado como 'Admin'; se o nome de usuário for 'notacompany\Admin', o login será negado.
Login com distinção entre maiúsculas e minúsculas	Desmarque a caixa de seleção para desabilitar a distinção entre maiúsculas e minúsculas nos nomes de usuário (habilitada por padrão). Desabilitar essa distinção permite, por exemplo, fazer login como "admin" mesmo que o usuário do Zabbix seja "Admin" ou "ADMIN". Observe que, se a distinção entre maiúsculas e minúsculas estiver desabilitada e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login desses usuários será sempre 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, pode ser necessário 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.

LDAP

Visão geral

A autenticação LDAP externa pode ser usada para verificar nomes de usuário e senhas.

A autenticação LDAP do Zabbix funciona pelo menos com Microsoft Active Directory e OpenLDAP.

Se apenas o login LDAP estiver configurado, o usuário também 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 com o atributo de nome de usuário retornado pelo LDAP.

Provisionamento de usuários

É possível configurar o **provisionamento de usuários** JIT (just-in-time) para usuários LDAP. Nesse caso, não é necessário que um usuário já exista no Zabbix. A conta de usuário pode ser criada quando o usuário fizer login no Zabbix pela primeira vez.

Quando um usuário LDAP informa seu login e senha LDAP, o Zabbix verifica no servidor LDAP *padrão* se esse usuário existe. Se o usuário existir e ainda não tiver uma conta no Zabbix, um novo usuário será criado no Zabbix e ele poderá fazer login.

Um usuário criado por meio do provisionamento JIT é associado ao servidor LDAP (diretório) definido como padrão no momento da criação. Alterar o servidor LDAP padrão posteriormente não altera nem atualiza o servidor LDAP vinculado aos usuários que já foram provisionados.

> **Attention:**
> Se o provisionamento JIT estiver habilitado, um grupo de usuários para usuários desprovisionados deverá ser especificado na aba *Autenticação*.

O provisionamento JIT também permite que as contas de usuários provisionados sejam atualizadas com base nas alterações no LDAP. Por exemplo, se um usuário for movido de um grupo LDAP para outro, ele também será movido de um grupo para outro no Zabbix; se um usuário for removido de um grupo LDAP, ele também será removido do grupo no Zabbix e, se não pertencer a nenhum outro grupo, será adicionado ao grupo de usuários para usuários desprovisionados. As contas de usuários provisionados 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 realizado pelo frontend do Zabbix quando o usuário está interagindo com ele ou quando há pelo menos uma página do frontend aberta no navegador. Não há processos dedicados em segundo plano para provisionar usuários.

O LDAP oferece três formas de vinculação ao diretório para autenticação e pesquisas:

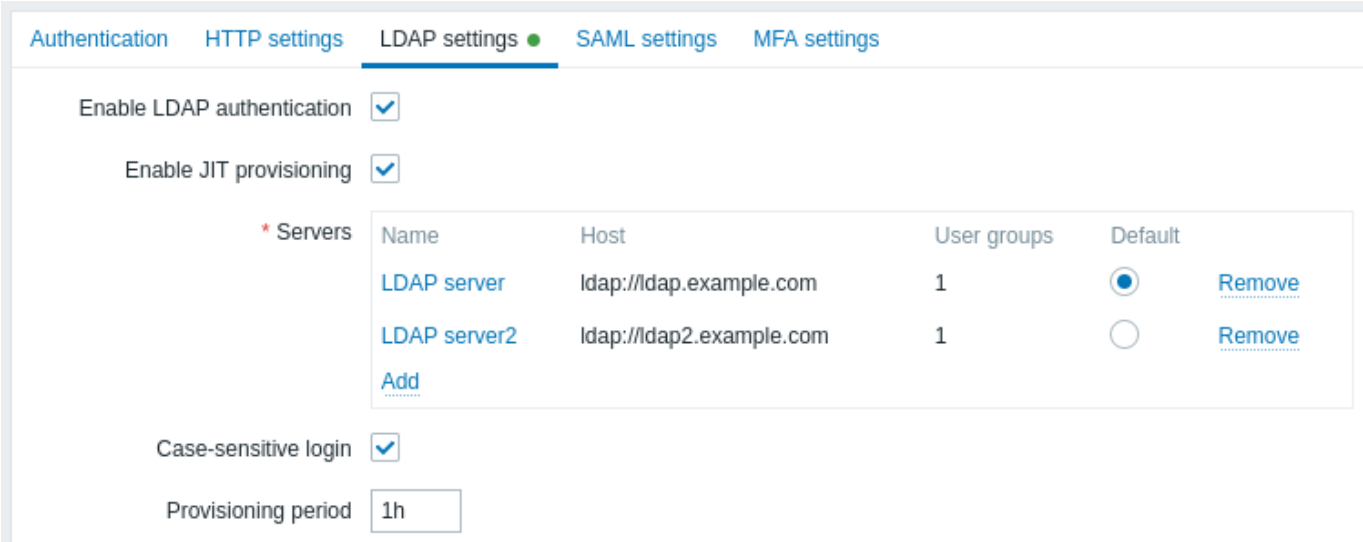
- Vinculação anônima - nenhum *Bind DN* / *Bind password* é fornecido e o servidor LDAP permite consultas anônimas.
- Usuário de vinculação dedicado (conta de serviço) - uma conta LDAP específica é definida em *Bind DN* / *Bind password* e usada pelo Zabbix para pesquisas e provisionamento. Essa é a opção recomendada e mais flexível, pois o Zabbix pode realizar pesquisas e provisionamento em segundo plano sem as credenciais do usuário final.
- Vinculação direta do usuário - o Zabbix realiza a vinculação usando as credenciais informadas pelo usuário no login (nenhum *Bind DN* / *Bind password* configurado); isso é configurado incluindo um placeholder, como `uid=%{user}`, no Base DN. Nesse modo, o Zabbix só tem acesso à senha do usuário durante o login interativo. Como resultado, as ações de provisionamento que exigem autenticação no LDAP fora da sessão de login do usuário (por exemplo, usando o botão *Provision now* ou executando o provisionamento em segundo plano quando o usuário não está conectado ativamente) não podem realizar a autenticação e, portanto, não funcionarão. O provisionamento e as atualizações para a vinculação direta do usuário ocorrem somente no momento em que o usuário faz login.

Vários servidores

Vários servidores LDAP podem ser definidos, se necessário. Por exemplo, um servidor diferente pode ser usado para autenticar um grupo de usuários diferente. Uma vez que os servidores LDAP estejam configurados, na configuração do grupo de usuários torna-se possível selecionar o servidor LDAP necessário para o respectivo grupo de usuários.

Se um usuário estiver em vários grupos de usuários e vários servidores LDAP, o primeiro servidor na lista de servidores LDAP ordenados por nome em ordem crescente será usado para autenticação.

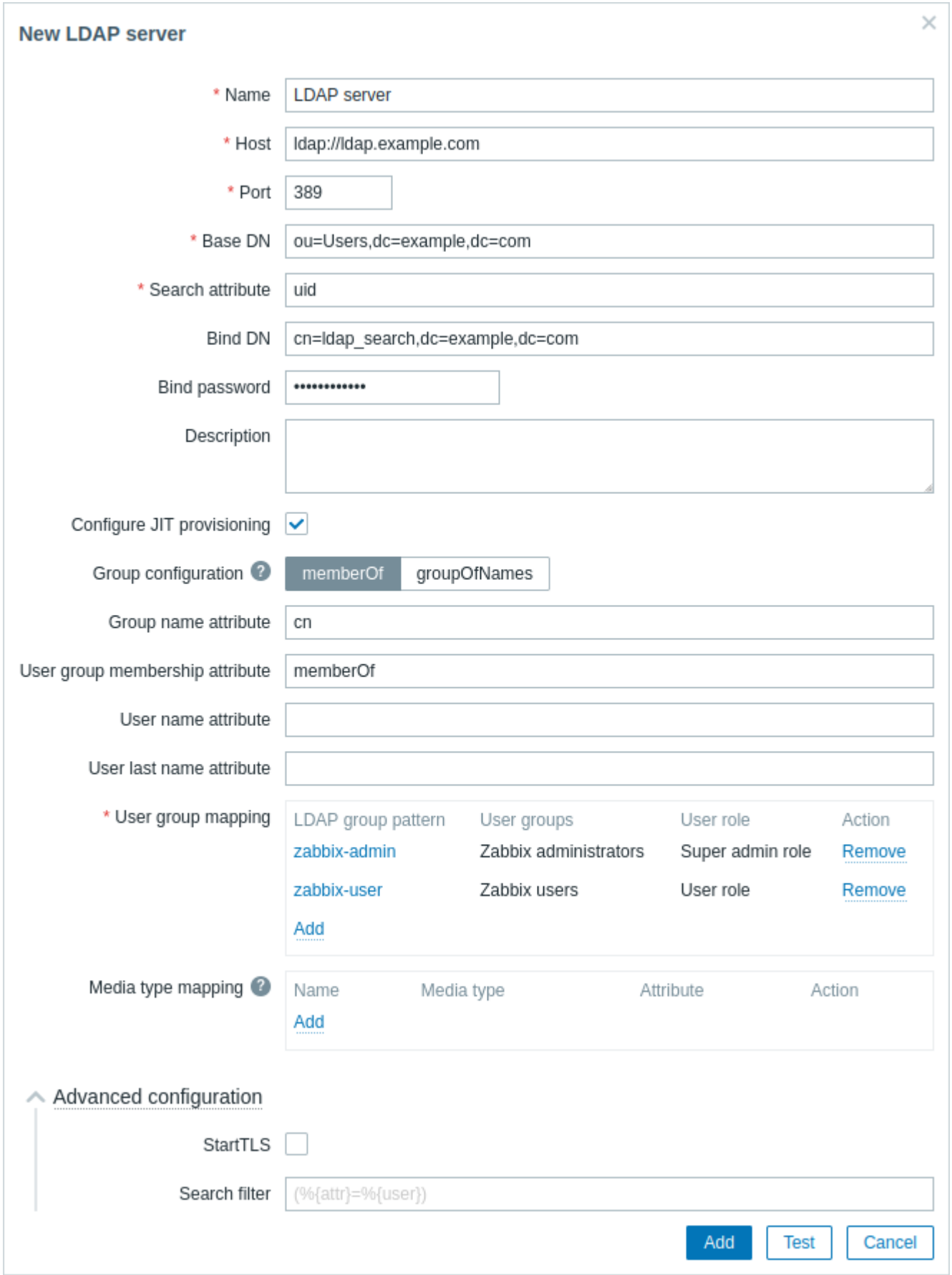
Configuração



Parâmetros de configuração:

Parâmetro	Descrição
Habilitar autenticação LDAP	Marque a caixa de seleção para habilitar a autenticação LDAP.
Habilitar provisionamento JIT	Marque a caixa de seleção para habilitar o provisionamento JIT.
Servidores	Clique em *Adicionar* para configurar um servidor LDAP (consulte Configuração do servidor LDAP abaixo).
Login com distinção entre maiúsculas e minúsculas	Desmarque a caixa de seleção para desabilitar a distinção entre maiúsculas e minúsculas nos nomes de usuário (habilitada por padrão). Desabilitar a distinção entre maiúsculas e minúsculas no login permite, por exemplo, fazer login como ”admin” mesmo que o usuário do Zabbix seja ”Admin” ou ”ADMIN”. Observe que, se a distinção entre maiúsculas e minúsculas no login estiver desabilitada e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login desses usuários será sempre 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 conectado será provisionado enquanto estiver trabalhando com o frontend.

Configuração do server LDAP



Parâmetros de configuração do server LDAP:

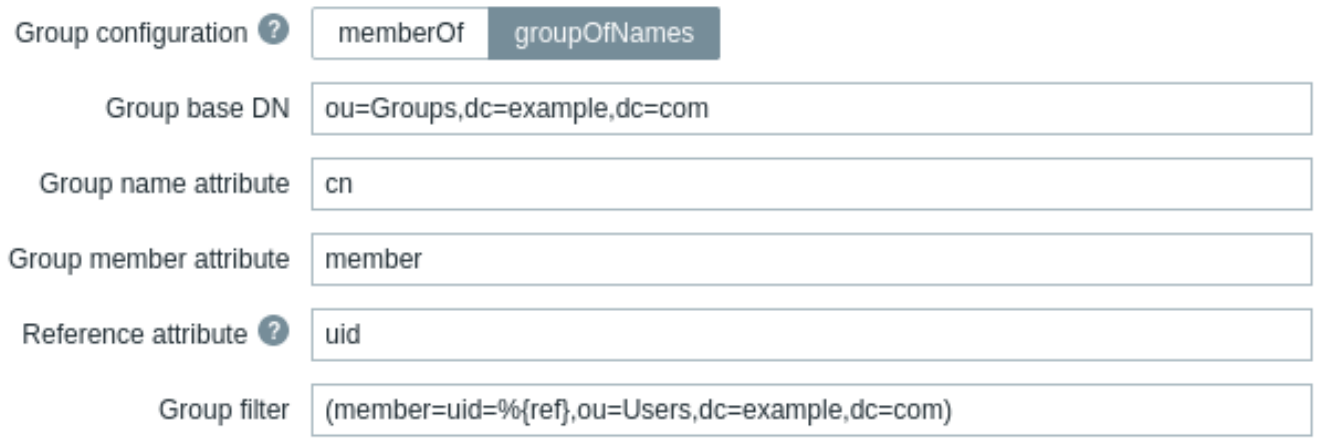
Parameter	Description
Name	Nome do server LDAP na configuração do Zabbix.

Parameter	Description
Host	Hostname, IP ou URI do server LDAP. Exemplos: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Para um server LDAP seguro, use o protocolo *ldaps* e o hostname. Exemplo: ldaps://ldap.example.com Com OpenLDAP 2.x.x e posterior, pode ser usado um LDAP URI completo no formato ldap://hostname:port ou ldaps://hostname:port.
Port	Porta do server LDAP. O padrão é 389. Para conexão LDAP segura, o número da porta normalmente é 636. Não é usado ao utilizar LDAP URIs completos.
Base DN	Caminho base para as contas de usuário no server LDAP: 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, veja uma observação abaixo)
Search attribute	Atributo da conta LDAP usado para pesquisa: uid (para OpenLDAP), sAMAccountName (para Microsoft Active Directory)
Bind DN	Conta LDAP para vinculação e pesquisa no server LDAP, exemplos: uid=ldap_search,ou=system (para OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (para Microsoft Active Directory) A vinculação anônima também é suportada. Observe que a vinculação anônima pode expor a configuração do domínio a usuários não autorizados (informações sobre usuários, computadores, servers, grupos, serviços etc.). Por motivos de segurança, desative vinculações anônimas nos hosts LDAP e use acesso autenticado em vez disso.
Bind password	Senha LDAP da conta para vinculação e pesquisa no server LDAP.
Description	Descrição do server LDAP.
Configure JIT provisioning	Marque esta caixa de seleção para exibir opções relacionadas ao JIT provisioning.
Group configuration	Selecione o método de configuração de grupos: **memberOf** - pesquisando usuários e seu atributo de associação a grupos **groupOfNames** - pesquisando grupos por meio do atributo member Observe que memberOf é preferível porque é mais rápido; use groupOfNames se o seu server LDAP não suportar `memberOf` ou se for necessário filtrar grupos.
Group name attribute	Especifique o atributo para obter o nome do grupo de todos os objetos no atributo `memberOf` (veja o campo *User group membership attribute*) O nome do grupo é necessário para o mapeamento de grupos de usuários.
User group membership attribute	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 esta: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Este campo está disponível apenas para o método memberOf.
User name attribute	Especifique o atributo que contém o primeiro nome do usuário.
User last name attribute	Especifique o atributo que contém o sobrenome do usuário.
User group mapping	Mapeie um padrão de grupo de usuários LDAP para o grupo de usuários e a função de usuário do Zabbix. Isso é necessário para determinar qual grupo/função de usuário o usuário provisionado receberá no Zabbix. Clique em *Add* para adicionar um mapeamento. O campo *LDAP group pattern* 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, ele receberá aquela com o nível de permissão mais alto entre elas.
Media type mapping	Mapeie os atributos LDAP media do usuário (por exemplo, email) para o media do usuário no Zabbix para envio de notificações (o valor do atributo é usado como o campo media *Send to*).
Advanced configuration	Clique no cabeçalho *Advanced configuration* para exibir opções avançadas de configuração (veja abaixo).
StartTLS	Marque a caixa de seleção para usar a operação StartTLS ao conectar-se ao server LDAP. A conexão falhará se o server não suportar StartTLS. StartTLS não pode ser usado com servers que usam o protocolo *ldaps*.

Parameter	Description
Search filter	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 autenticar 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 server LDAP para **vinculação direta de usuário**, acrescente um atributo uid=%{user} ao parâmetro *Base DN* (por exemplo, *uid=%{user},dc=example,dc=com*) e deixe os parâmetros *BindDN* e *Bind password* vazios. Ao autenticar, o placeholder %{user} será substituído pelo nome de usuário informado durante o login.
 Com a vinculação direta de usuário, o Zabbix só tem acesso às credenciais do usuário durante o login interativo. Portanto, tarefas de provisionamento que operam sem um login interativo (por exemplo, o botão *Provision now* ou o método de API `user.provision`) ignorarão usuários que se autenticam usando vinculação direta de usuário, porque o Zabbix não pode se vincular em nome deles. Use vinculação anônima ou um usuário de vinculação dedicado (conta de serviço) se você precisar que o provisionamento e as pesquisas LDAP funcionem no frontend sem exigir que o usuário final esteja autenticado.

Os campos a seguir são específicos de "groupOfNames" como método de *Group configuration*:



Parameter	Description
Group base DN	Caminho base para os grupos no server LDAP.
Group name attribute	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.
Group member attribute	Especifique o atributo que contém informações sobre os membros do grupo no LDAP (por exemplo, `member`).
Reference attribute	Especifique o atributo de referência para o filtro de grupo (veja o campo *Group filter*). Em seguida, use `%{ref}` no filtro de grupo para obter valores para o atributo especificado aqui.
Group filter	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, talvez seja necessário adicionar uma linha `TLS_REQCERT allow` ao arquivo de configuração `/etc/openldap/ldap.conf`. Isso pode reduzir a segurança da conexão com o catálogo LDAP.

Note:
É recomendável criar uma conta LDAP separada (*Bind DN*) para realizar a vinculação e a pesquisa no server LDAP com privilégios mínimos no LDAP, em vez de usar contas de usuário reais (usadas para fazer login no frontend do Zabbix).
 Essa abordagem oferece mais segurança e não exige alterar o *Bind password* quando o usuário altera sua própria senha no server LDAP.
 Na tabela acima, esse 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 previamente com o nome do usuário atual do frontend do Zabbix). Esse nome de usuário deve existir no server 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.

SAML

Visão geral

A autenticação SAML 2.0 pode ser usada para entrar no Zabbix.

Se o login via SAML for configurado sem provisionamento de usuário JIT, o usuário já deve existir no Zabbix; a senha do usuário no Zabbix não é usada para autenticação. Após a autenticação SAML bem-sucedida, o Zabbix compara o atributo de nome de usuário recebido do SAML com um nome de usuário local existente no Zabbix.

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 maneira:

1. Forneça a chave privada e os certificados do SP. Um certificado do IdP deve ser fornecido (cole o conteúdo ou escolha um arquivo) para ativar o SAML por meio do frontend. Um certificado do SP e uma chave privada do SP devem ser fornecidos se as opções *Assinar* ou *Criptografar* estiverem selecionadas.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido no arquivo de configuração do frontend (`zabbix.conf.php`), você poderá colar o texto do certificado ou carregar o arquivo do certificado no frontend durante a configuração do SAML — não são necessários arquivos no sistema de arquivos.

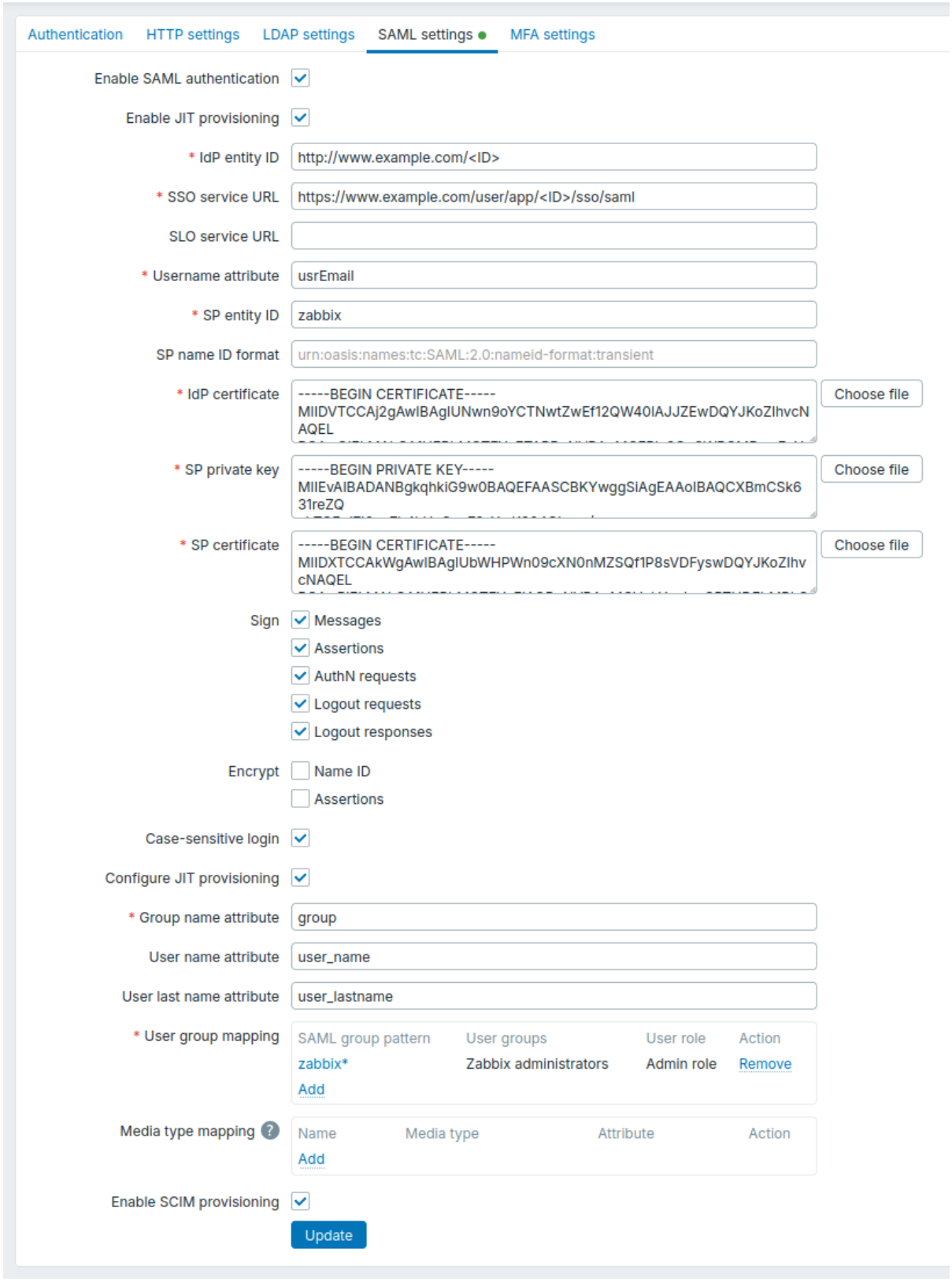
Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no arquivo de configuração, o certificado deverá estar disponível no sistema de arquivos (por padrão, em `ui/conf/certs` ou no caminho configurado no arquivo de configuração), e o frontend não armazenará os certificados no banco de dados. Observe que, se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será considerado.

Por padrão, o Zabbix procurará nos seguintes locais:

- ui/conf/certs/sp.key - arquivo da chave privada do SP
- ui/conf/certs/sp.crt - arquivo do certificado do SP
- ui/conf/certs/idp.crt - arquivo do certificado do IdP

> **Attention:**
> O material dos certificados pode ser importado pelo frontend colando o texto ou carregando arquivos. Embora importações não criptografadas sejam permitidas por conveniência em alguns ambientes, é altamente recomendável usar transporte criptografado (por exemplo, habilitar HTTPS/TLS para o frontend do Zabbix). Quando os certificados/chaves SAML forem armazenados no banco de dados, habilite TLS/SSL para as conexões com o banco de dados, a fim de proteger o tráfego entre a aplicação e o banco de dados.
 Criptografe os 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 definidas 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 usuários JIT.
ID da entidade do IDP	O identificador exclusivo da entidade no 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 ficar vazia, 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 compatíveis 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
ID da entidade do SP	O identificador exclusivo do provedor de serviços (se não corresponder, a operação será rejeitada). É possível especificar uma URL ou qualquer string de dados.
Formato do ID de nome do SP	Solicita um formato específico de ID de nome 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 do IdP	Conteúdo do certificado do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML.
Chave privada do SP	Conteúdo da chave privada do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML. Fornece autenticação segura e troca de dados com o provedor de identidade (IdP).
Certificado do SP	Conteúdo do certificado do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML.
Assinar	Desmarque as caixas de seleção para desabilitar as assinaturas SAML para as entidades correspondentes: *Mensagens* *Asserções* *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: *ID de nome* *Asserções*
Login com distinção entre maiúsculas e minúsculas	Desmarque a caixa de seleção para desabilitar o login com distinção entre maiúsculas e minúsculas para nomes de usuário (habilitado por padrão). Desabilitar essa opção permite, por exemplo, fazer login como ”admin” mesmo que o usuário do Zabbix seja ”Admin” ou ”ADMIN”. Observe que, se o login com distinção entre maiúsculas e minúsculas estiver desabilitado e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login desses 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 exibir opções relacionadas ao provisionamento de usuários JIT.
Atributo de nome do grupo	Especifique o atributo de nome do grupo para o provisionamento de usuários JIT.
Atributo de nome do usuário	Especifique o atributo de nome do usuário para o provisionamento de usuários JIT.
Atributo de sobrenome do usuário	Especifique o atributo de sobrenome do usuário para o provisionamento de usuários JIT.

Parâmetro	Descrição
Mapeamento de grupos de usuários	Mapeie um padrão de grupo de usuários SAML para um grupo de usuários e uma função de usuário do Zabbix. Isso 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* aceita curingas. O nome do grupo deve corresponder a um grupo existente. Se um usuário SAML corresponder a vários grupos de usuários do Zabbix, ele se tornará membro de todos eles. Se um usuário corresponder a várias funções de usuário do Zabbix, ele receberá o nível de permissão mais alto entre elas.
Mapeamento de tipos de mídia	Mapeie os atributos de mídia SAML do usuário (por exemplo, e-mail) para a mídia do usuário do Zabbix, usada para enviar notificações.
Habilitar provisionamento SCIM	Marque esta caixa de seleção para habilitar o provisionamento SCIM 2.0.

Veja exemplos de configuração de provedores de identidade SAML para login e provisionamento de usuários no Zabbix com:

- Microsoft Azure AD
- Okta
- Onelogin

Observações sobre o provisionamento SCIM

Para o provisionamento SCIM, especifique o caminho para o frontend do Zabbix e acrescente `api_scim.php` a ele, no lado do provedor de identidade, ou seja:

`https://<path-to-zabbix-ui>/api_scim.php`

Os atributos de usuário 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`):

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Exemplos de configuração

Exemplo 1: Conexão com uma instância do Zabbix usando um balanceador de carga com TLS/SSL

Se um balanceador de carga for usado para conectar-se a uma instância do Zabbix, e o balanceador de carga usar TLS/SSL enquanto o Zabbix não usar, especifique 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 2: Configuração de algoritmos personalizados de assinatura e resumo SAML

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Exemplo 3: Configuração de SSO com autenticação Kerberos usando ADFS

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Nesse caso, no campo *SP name ID* da configuração SAML, defina:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

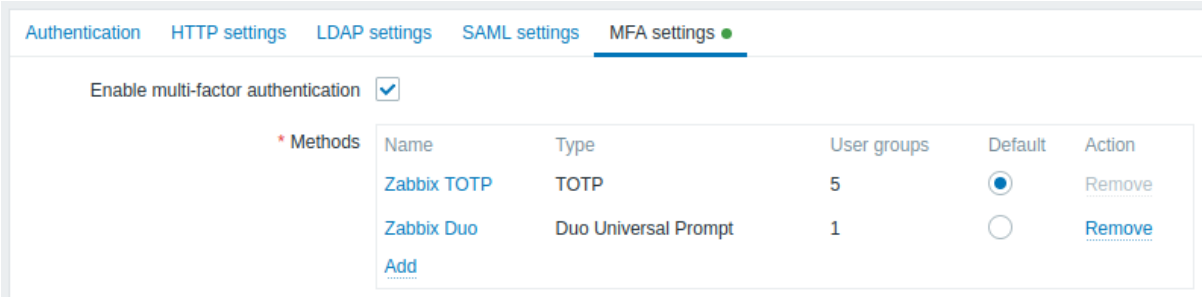
Visão geral

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.

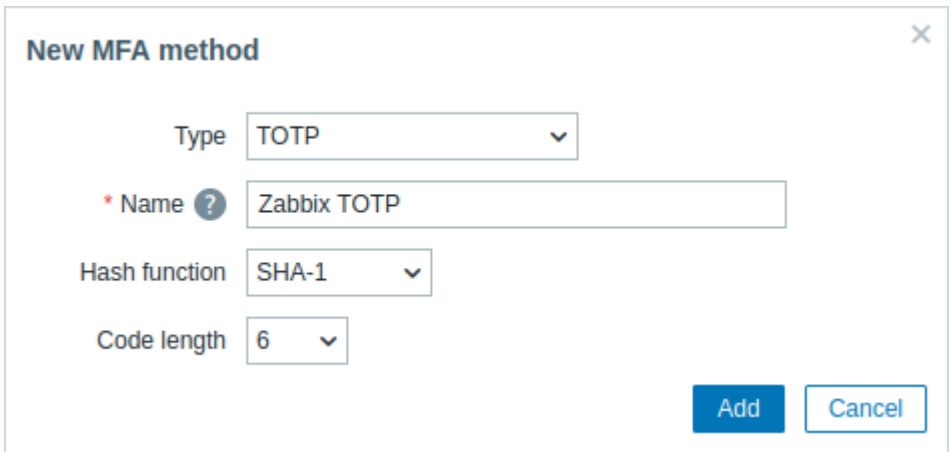
Configuração



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”.
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. Acesse as configurações de MFA no Zabbix em *Users* > *Authentication* e habilite a autenticação multifator.

2. Adicione um novo método de MFA com a seguinte configuração:

   - Type: TOTP
   - Name: Zabbix TOTP
   - Hash function: SHA-1
   - Code length: 6

3. Clique no botão *Add* e depois no botão *Update*.

4. Acesse *Users* > *User groups* e crie um novo grupo de usuários com a seguinte configuração:

   - Group name: TOTP group
   - Users: Admin
   - Multi-factor authentication: Default (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 no 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 gerará um código de verificação que você deve inserir para concluir o processo de login.

7. Para logins subsequentes, obtenha 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 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 da Duo. Além disso, se você tiver habilitado a Política de Segurança de Conteúdo (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 obter uma conta gratuita de administrador da Duo em Duo Signup.

2. Abra o Painel de Administração da Duo, vá para *Applications* > *Protect an Application*, pesquise a aplicação *Web SDK* e clique em *Protect*.

3. Anote as credenciais (Client ID, Client secret, API hostname) necessárias para configurar o método MFA no Zabbix.

4. Acesse as configurações de MFA no Zabbix em *Users* > *Authentication* e habilite a autenticação multifator.

5. Adicione um novo método MFA com a seguinte configuração:

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (use o API hostname da Duo)
- Client ID: (use o Client ID da Duo)
- Client secret: (use o Client secret da Duo)

6. Clique no botão *Add* e depois no botão *Update*.

7. Vá para *Users* > *User groups* e crie um novo grupo de usuários com a seguinte configuração:

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default (ou "Zabbix Duo" se não estiver definido como padrão)

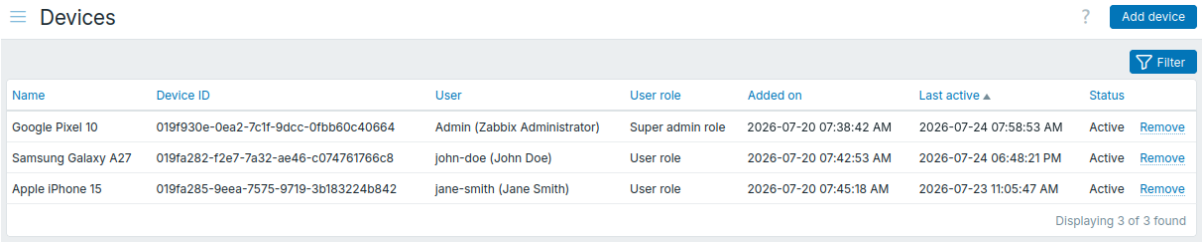
8. Saia do Zabbix e entre novamente usando suas credenciais. Após o login bem-sucedido, você será solicitado a se inscrever no MFA e redirecionado para a Duo. Conclua a configuração da Duo e verifique seu usuário com o aplicativo Duo no seu telefone para fazer login.

9. Para logins subsequentes, use o método MFA apropriado fornecido pelo aplicativo da Duo (como obter um código de verificação, responder a notificações push ou usar chaves de segurança) e insira as informações necessárias durante o login.

Dispositivos

Esta seção permite que Super administradores gerenciem dispositivos móveis que serão usados com o aplicativo móvel do Zabbix.

Esta seção é exibida se `EnableMobileDevices` estiver habilitado.



Para adicionar um dispositivo:

1. Clique em *Adicionar dispositivo*.
2. Selecione um usuário e clique em *Adicionar*. O usuário deve ter acesso a dispositivos habilitado.
3. Um código QR será exibido. O usuário deve escanear esse código no aplicativo móvel do Zabbix.

Note:
Quando você adiciona um dispositivo para um usuário, o Zabbix também adiciona o tipo de mídia de notificação push às mídias do usuário. Quando você remove o dispositivo, essa mídia do usuário também é removida. Se você tiver vários tipos de mídia de notificação push configurados, o Zabbix adicionará aquele que estiver habilitado e aparecer primeiro em ordem alfabética.

Os dispositivos adicionados aparecem em uma lista, que pode ser filtrada.

Você também pode remover dispositivos. Quando você remove um dispositivo, suas credenciais de acesso são revogadas e removidas do Zabbix, após o que ele não poderá mais acessar o Zabbix.

A lista exibe estes detalhes:

Coluna	Descrição
Nome	Nome do dispositivo.
ID do dispositivo	ID do dispositivo.
Usuário	Usuário para o qual o dispositivo foi adicionado ou *Usuário inacessível* se esse usuário tiver sido excluído.
Função do usuário	Função do usuário.
Adicionado em	Data e hora em que o dispositivo foi adicionado.
Última atividade	Data e hora em que o dispositivo esteve ativo pela última vez.
Status	Status do dispositivo: **Novo** - o dispositivo foi adicionado sem escanear o código QR e precisará ser adicionado novamente. **Novo (token expirado)** - o dispositivo foi adicionado sem escanear o código QR, e o token no código QR expirou; o dispositivo precisará ser adicionado novamente. **Ativo** - o dispositivo foi adicionado, o código QR foi escaneado e ele pode usar o aplicativo móvel do Zabbix. **Não atribuído** - o dispositivo foi adicionado, mas seu usuário foi excluído, e ele não pode usar o aplicativo móvel do Zabbix.

Usuários com permissões suficientes também podem gerenciar seus próprios dispositivos em *Configurações do usuário* > *Dispositivos*.

Administração

Visão geral

O menu Administração é destinado às funções administrativas do Zabbix. Este menu está disponível apenas para usuários do tipo SuperAdmin.

Fonte de dados

Visão geral

A seção *Administração* > *Fonte de dados* permite especificar a fonte de dados de telemetria, ou seja, o banco de dados que armazena dados de telemetria (traces, métricas e logs). Os dados de telemetria são usados para o monitoramento do desempenho de aplicações (APM), que ajuda você a visualizar como as solicitações fluem por suas aplicações e serviços, medir o desempenho deles e encontrar a causa de operações lentas ou com falha.

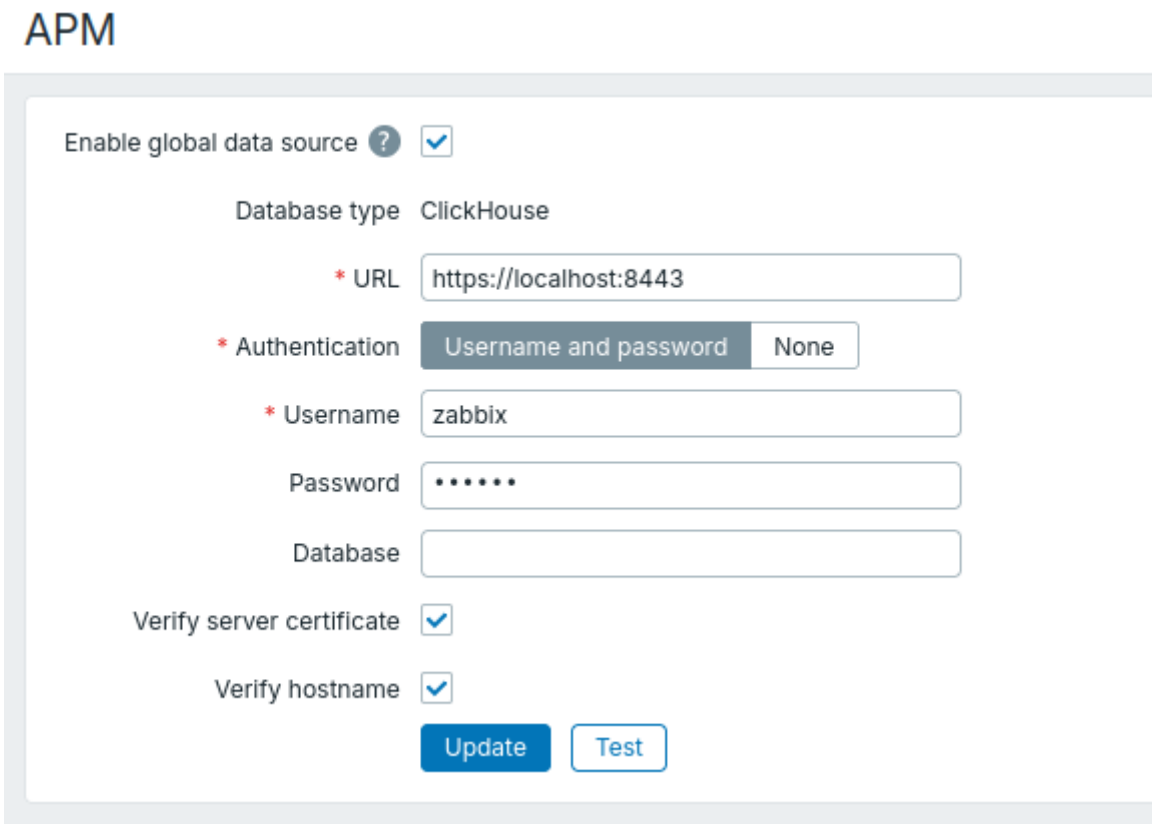
A fonte de dados de telemetria deve oferecer suporte ao OpenTelemetry.

A fonte de dados de telemetria especificada nesta seção é chamada de fonte de dados global, pois é compartilhada pelo Zabbix server, proxy e frontend. No entanto, você pode substituí-la para qualquer um desses componentes especificando uma fonte de dados de telemetria diferente na configuração do server, do proxy ou do frontend.

O item consulta de telemetria consulta essa fonte de dados, a menos que outra seja especificada na configuração do componente.

APM

Para especificar a fonte de dados global, marque a caixa de seleção *Ativar fonte de dados global* e preencha os parâmetros descritos abaixo. Antes de salvar, você pode clicar em *Testar* para verificar a conexão com esta fonte de dados (requer a extensão php-curl).



Parâmetro	Descrição
Tipo de banco de dados	O tipo de banco de dados é definido como *ClickHouse* e não pode ser alterado. Consulte também: Requisitos e Configuração do ClickHouse.
URL	Insira a URL HTTP ou HTTPS usada para conectar-se ao banco de dados.
Autenticação	Selecione o método de autenticação usado para fazer login no banco de dados: *Nome de usuário e senha* ou *Nenhum*.
Nome de usuário	Especifique o nome de usuário usado para fazer login no banco de dados. Este parâmetro está disponível quando *Autenticação* está definida como *Nome de usuário e senha*.
Senha	Especifique a senha usada para fazer login no banco de dados ou deixe o campo vazio se o usuário do banco de dados não tiver uma senha. A senha é redefinida quando você altera a URL. Este parâmetro está disponível quando *Autenticação* está definida como *Nome de usuário e senha*.
Banco de dados	Insira o nome do banco de dados que contém os dados de telemetria. Se deixado vazio, o Zabbix usará o banco de dados padrão configurado no ClickHouse para o usuário.
Verificar certificado do server	Marque a caixa de seleção para verificar o certificado do server. Se marcada, as opções TLS relacionadas deverão ser definidas na configuração do server, do proxy ou do frontend. Este parâmetro está disponível quando a *URL* contém uma URL HTTPS.
Verificar nome do host	Marque a caixa de seleção para verificar se o nome do host na *URL* corresponde ao nome do host no certificado do server. Se marcada, as opções TLS relacionadas deverão ser definidas na configuração do server, do proxy ou do frontend. Este parâmetro está disponível quando a *URL* contém uma URL HTTPS.

Geral

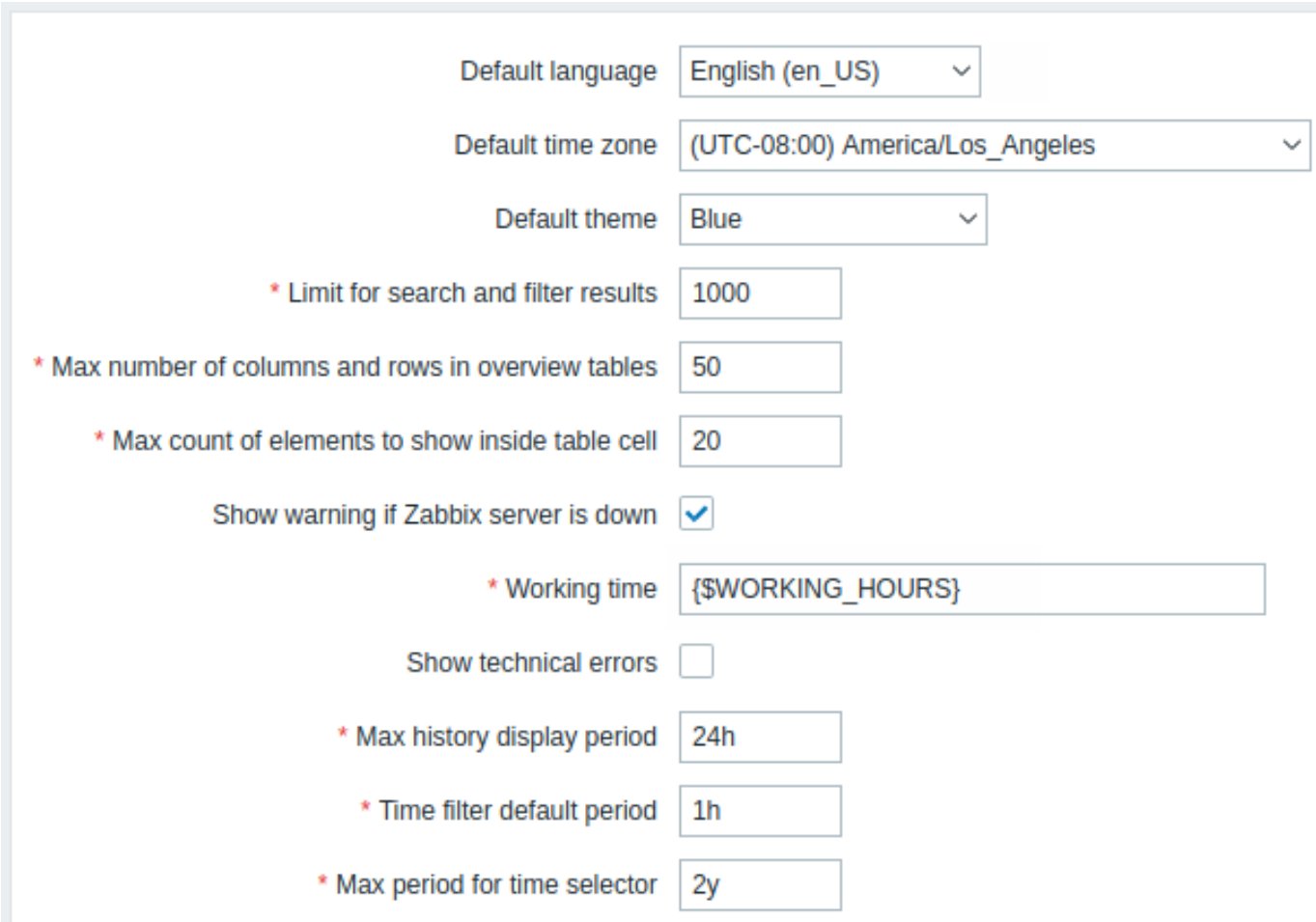
Visão geral

A seção *Administration* > *General* contém várias subseções para definir padrões relacionados ao frontend e personalizar o Zabbix.

A lista de subseções disponíveis aparece ao clicar em *General* na seção de menu *Administration*. Também é possível alternar entre as subseções usando o menu suspenso do título no canto superior esquerdo.

GUI

Esta seção fornece a personalização de vários padrões relacionados ao frontend.



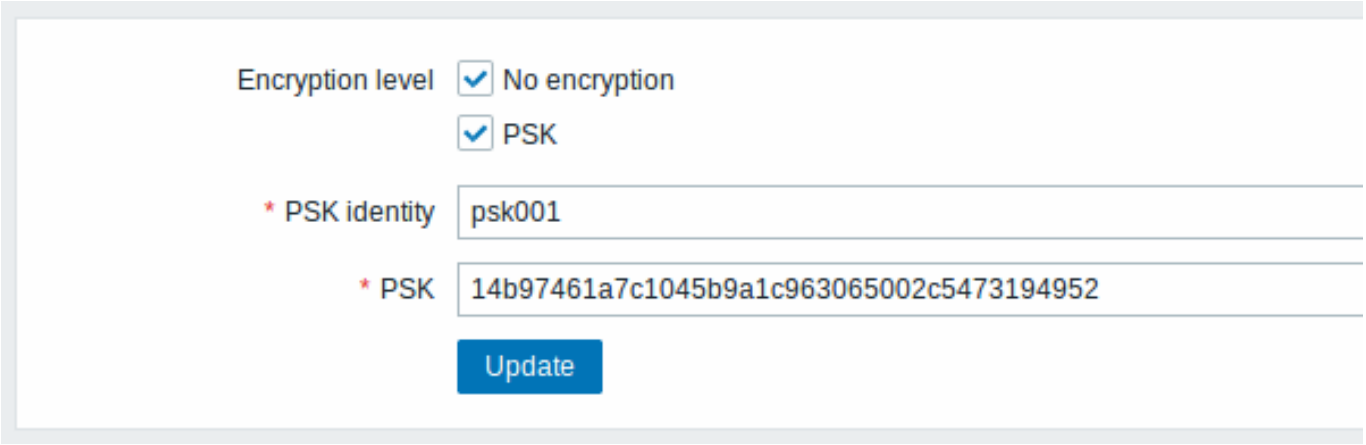
Parâmetros de configuração:

Parameter	Description
Default language	Idioma padrão para usuários que não especificaram um idioma em seus perfis e para usuários convidados. Para mais informações, consulte Instalação de idiomas adicionais do frontend.
Default time zone	Fuso horário padrão para usuários que não especificaram um fuso horário em seus perfis e para usuários convidados.
Default theme	Tema padrão para usuários que não especificaram um tema em seus perfis e para usuários convidados.
Limit for search and filter results	Quantidade máxima de elementos (linhas) que será exibida em uma lista da interface web, por exemplo, em *Data collection* > *Hosts*. *Nota*: se definido, por exemplo, como '50', apenas os primeiros 50 elementos serão exibidos em todas as listas do frontend afetadas. Se alguma lista contiver mais de cinquenta elementos, a indicação disso será o sinal '+' em *"Displaying 1 to 50 of **50+** found"*. Além disso, se a filtragem for usada e ainda houver mais de 50 correspondências, apenas as primeiras 50 serão exibidas. Observe que aumentar o valor deste parâmetro pode levar à redução do desempenho e ao aumento do consumo de memória no lado do servidor web.
Max number of columns and rows in overview tables	Número máximo de colunas e linhas a exibir no widget de dashboard *Trigger overview*. O mesmo limite se aplica tanto a colunas quanto a linhas. Se existirem mais linhas e/ou colunas do que as exibidas, o sistema mostrará um aviso na parte inferior da tabela: "Not all results are displayed. Please provide more specific search criteria."
Max count of elements to show inside table cell	Para entradas exibidas em uma única célula de tabela, será mostrado no máximo o configurado aqui.
Show warning if Zabbix server is down	Este parâmetro habilita a exibição de uma mensagem de aviso em uma janela do navegador se o Zabbix server não puder ser alcançado (possivelmente fora do ar). A mensagem permanece visível mesmo se o usuário rolar a página para baixo. Quando o cursor passa sobre ela, a mensagem é ocultada temporariamente para revelar o conteúdo abaixo.

Parameter	Description
Working time	Este parâmetro global define o horário de trabalho. Em gráficos, o horário de trabalho é exibido como fundo branco e o horário fora do trabalho é exibido em cinza. Consulte a página Time period specification para a descrição do formato de tempo. User macros são suportadas.
Show technical errors	Se marcado, todos os usuários registrados poderão ver erros técnicos (PHP/SQL). Se desmarcado, as informações estarão disponíveis apenas para Zabbix Super Admins e usuários pertencentes aos grupos de usuários com debug mode habilitado.
Max history display period	Período máximo de tempo para exibir dados históricos em *Monitoring* > *Latest data*, informações de web scenario do host em *Monitoring* > *Hosts* e no widget de dashboard *Top items*. Intervalo permitido: 24 horas (padrão) - 1 semana. Sufixos de tempo, por exemplo, 1w (uma semana), 36h (36 horas), são suportados.
Time filter default period	Período de tempo a ser usado por padrão em gráficos e dashboards. 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 é armazenado como preferência do usuário, substituindo o padrão global ou uma seleção anterior do usuário.
Max period for time selector	Período máximo de tempo disponível para gráficos e dashboards. 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 do agent ativo.



Os parâmetros marcados com um asterisco são obrigatórios.

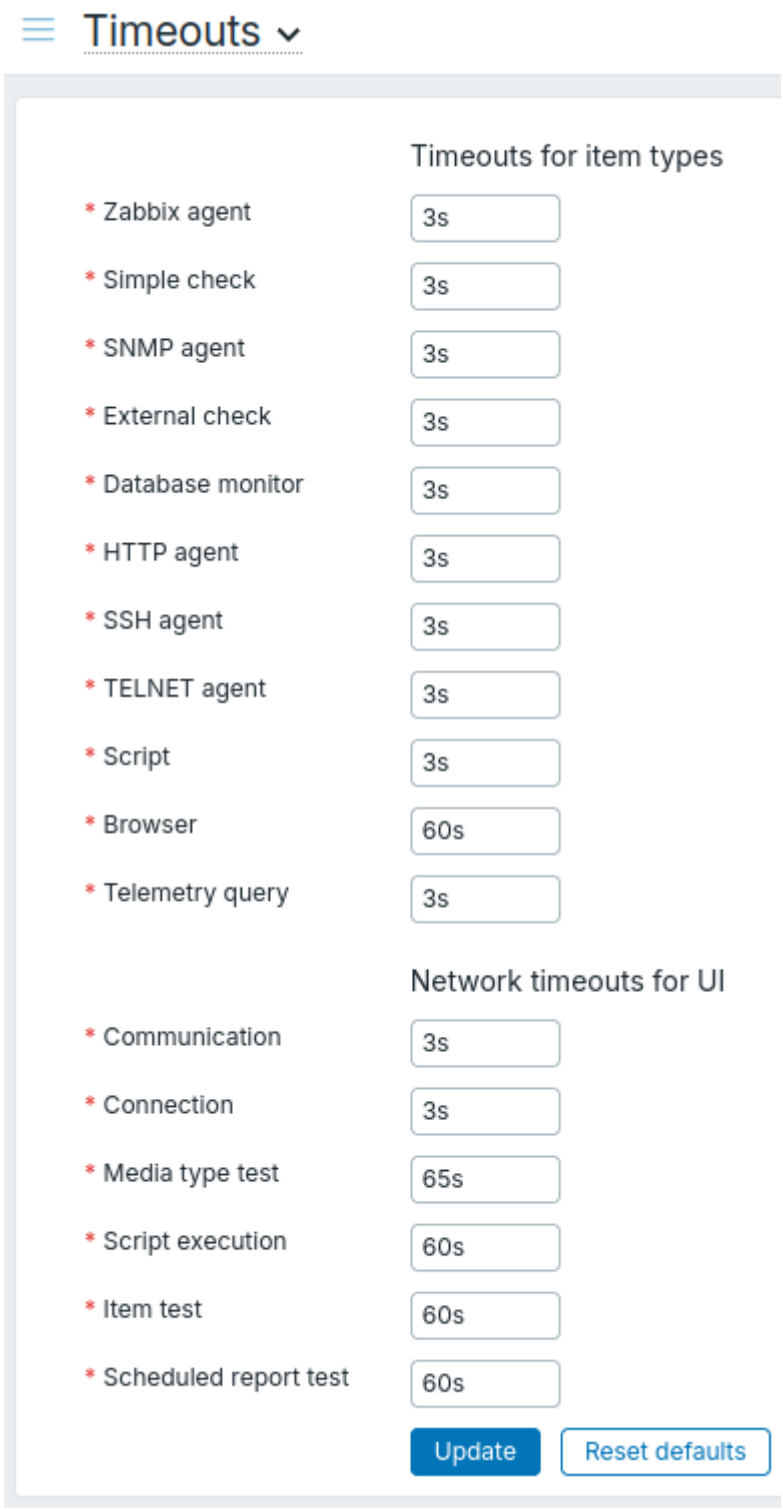
Parâmetros de configuração:

Parameter	Description
Nível de criptografia	Selecione uma ou ambas as opções de nível de criptografia: **Sem criptografia** - conexões sem criptografia são permitidas **PSK** - conexões criptografadas com TLS usando uma chave pré-compartilhada são permitidas
Identidade da PSK	Digite a string de identidade da chave pré-compartilhada. Este campo só está disponível se 'PSK' estiver selecionado como *Nível de criptografia*. Não coloque informações sensíveis na identidade da PSK; ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK usar.
PSK	Digite 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' estiver selecionado como *Nível de criptografia*.

Veja também: Autoregistro seguro

Timeouts

Nesta seção, é possível definir timeouts globais para tipos de item e timeouts de rede. Todos os campos deste formulário são obrigatórios.

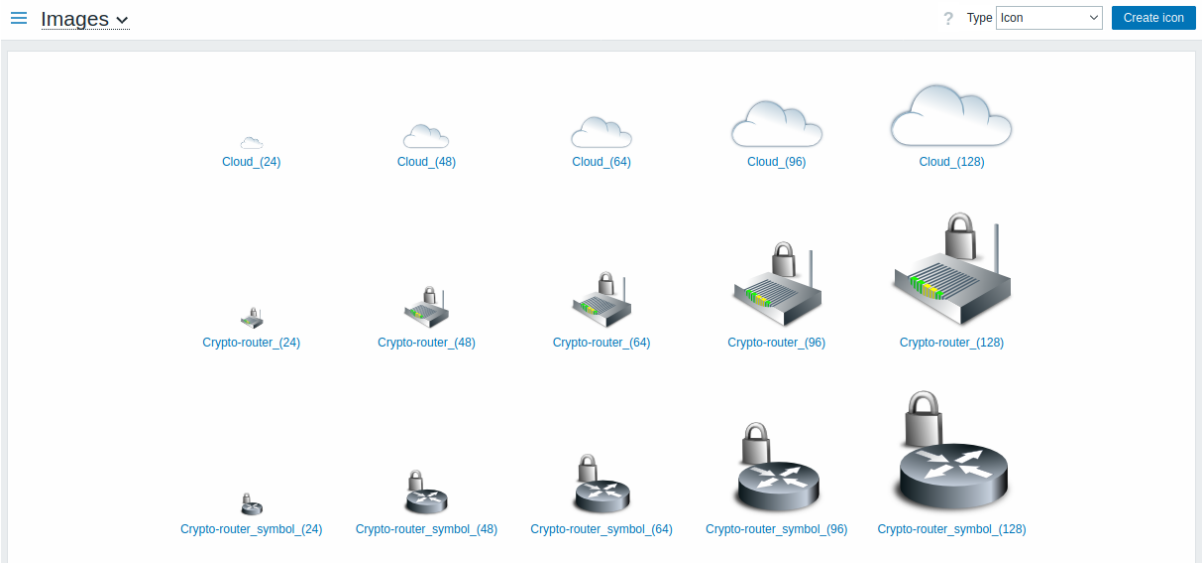


Parameter	Description
Timeouts for item types	Quantos segundos o Zabbix deve aguardar antes de abortar a verificação (com base no tipo). Para detalhes adicionais e limitações, consulte item timeout. Intervalo permitido: 1 - 600s (padrão: 3s; padrão para itens browser: 60s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são suportados. Observe que timeouts no nível do proxy, se configurados, substituem estas definições globais. Além disso, se timeouts para items específicos estiverem configurados, eles substituem tanto as definições globais quanto as do proxy.
Network timeouts for UI	

Parameter		Description
	Communication	Quantos segundos aguardar antes de fechar um socket ocioso (se uma conexão com o Zabbix server tiver sido estabelecida anteriormente, mas o frontend não conseguir concluir a operação de leitura/envio de dados durante esse período, a conexão será encerrada). Intervalo permitido: 1 - 300s (padrão: 3s).
	Connection	Quantos segundos aguardar antes de interromper uma tentativa de conexão com o Zabbix server. Intervalo permitido: 1 - 30s (padrão: 3s).
	Media type test	Quantos segundos aguardar por uma resposta ao testar um tipo de mídia. Intervalo permitido: 1 - 300s (padrão: 65s).
	Script execution	Quantos segundos aguardar por uma resposta ao executar um script. Intervalo permitido: 1 - 300s (padrão: 60s). Este timeout vale para toda a cadeia de execução do script, que pode ter comprimentos variados. Por exemplo, se o script for executado no agent, trata-se de uma ida e volta pelo server (possivelmente também pelo proxy) até o agent e de volta. Para scripts executados no agent, aplica-se o timeout do server ou do proxy. Para scripts executados apenas em um agent ativo, é provável que o timeout padrão do server/proxy precise ser aumentado. O timeout do server/proxy deve ser maior que a frequência de atualização da verificação ativa; caso contrário, o timeout do server/proxy poderá ser excedido antes mesmo de o agent ativo receber o script. Veja também: Script timeout.
	Item test	Quantos segundos aguardar pelos dados retornados ao testar um item. Intervalo permitido: 1 - 600s (padrão: 60s).
	Scheduled report test	Quantos segundos aguardar 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.

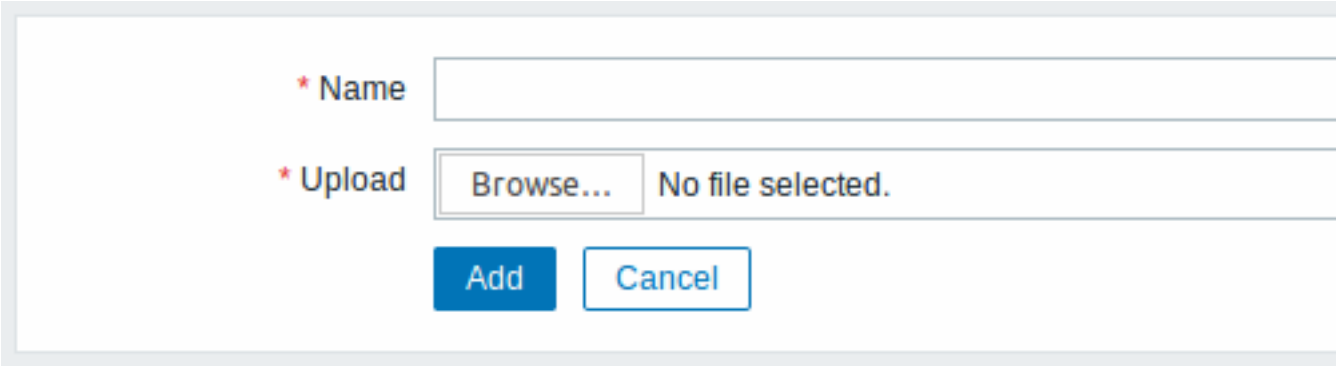


A lista suspensa *Tipo* permite alternar entre imagens de ícone e de plano de fundo:

- Os ícones são usados para exibir elementos do mapa de rede.
- Os planos de fundo são usados como imagens de fundo dos mapas de rede.

Adicionando uma imagem

Você pode adicionar sua própria imagem clicando no botão *Criar ícone* ou *Criar plano de fundo* no canto superior direito.



Atributos da imagem:

Parâmetro	Descrição
Nome	Nome exclusivo de uma imagem.
Upload	Selecione o arquivo (PNG, JPEG, GIF ou WebP) em um sistema local para fazer upload para o Zabbix. *Observação*: pode ser possível fazer upload de outros formatos que serão convertidos para PNG durante o upload. A biblioteca GD é usada para o processamento de imagens; portanto, os formatos compatíveis dependem da versão da biblioteca utilizada (o Zabbix requer a versão 2.0.28 ou superior).

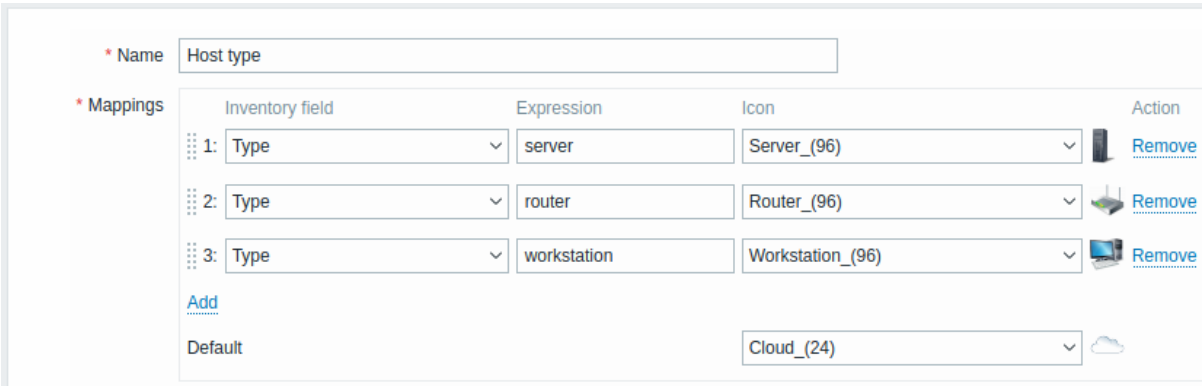
Note:
O tamanho máximo do arquivo de upload é limitado pelo valor de `ZBX_MAX_IMAGE_SIZE`, que é de 1024x1024 bytes ou 1 MB.
O upload de uma imagem poderá falhar se o tamanho da imagem estiver próximo de 1 MB e o parâmetro de configuração `max_allowed_packet` do MySQL estiver com o valor padrão de 1 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 em configuração de mapa de rede para atribuir automaticamente os ícones apropriados aos hosts correspondentes.

Para criar um novo mapa de ícones, clique em *Create icon map* no canto superior direito.



Parâmetros de configuração:

Parameter	Description
Name	Nome exclusivo do mapa de ícones.
Mappings	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.
Inventory field	Campo de inventário do host que será consultado para buscar uma correspondência.
Expression	Expressão regular que descreve a correspondência.
Icon	Ícone a ser usado se uma correspondência para a expressão for encontrada.

Parameter	Description
Default	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 estado do trigger é exibido no frontend e os nomes e cores de severidade do trigger.

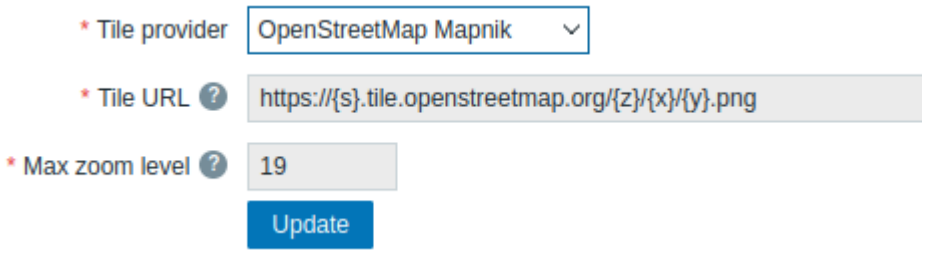


Parameter	Description
Use custom event status colors	Marque este parâmetro para ativar a personalização das cores para problemas reconhecidos/não reconhecidos.
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	Digite um novo código de cor ou clique na cor para selecionar uma nova na paleta fornecida. Se a caixa de seleção *blinking* estiver marcada, os triggers piscarão por algum tempo após a mudança de estado para ficarem mais visíveis.
Display OK triggers for	Período de tempo para exibir triggers OK. Intervalo permitido: 0 - 24 horas. Sufixos de tempo, por exemplo 5m, 2h, 1d, são suportados.
On status change triggers blink for	Duração do piscar do trigger. Intervalo permitido: 0 - 24 horas. Sufixos de tempo, por exemplo 5m, 2h, 1d, são suportados.
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	Nomes e/ou cores de severidade personalizados para exibição em vez dos padrões 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 locais. 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 blocos de mapa geográfico e configurar as definições do provedor de serviço para o widget de dashboard Geomap dashboard widget. Para fornecer visualização usando os mapas geográficos, o Zabbix usa a

biblioteca JavaScript de código aberto Leaflet para mapas interativos. Observe que o Zabbix não tem controle sobre a qualidade das imagens fornecidas por provedores de blocos de terceiros, incluindo os provedores de blocos predefinidos.



Parameter	Description
Tile provider	Selecione um dos provedores de serviço de blocos disponíveis ou selecione *Other* para adicionar outro provedor de blocos ou blocos hospedados localmente (consulte Using a custom tile service provider).
Tile URL	O modelo de URL (até 2048 caracteres) para carregar e exibir a camada de blocos nos mapas geográficos. Este campo é editável somente 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 bloco; *{r}* pode ser usado para adicionar "@2x" à URL e carregar blocos retina. Exemplo: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Texto de atribuição do provedor de blocos a ser exibido em uma pequena caixa de texto no mapa. Este campo fica visível somente se *Tile provider* estiver definido como *Other*.
Max zoom level	Nível máximo de zoom do mapa. Este campo é editável somente se *Tile provider* estiver definido como *Other*.

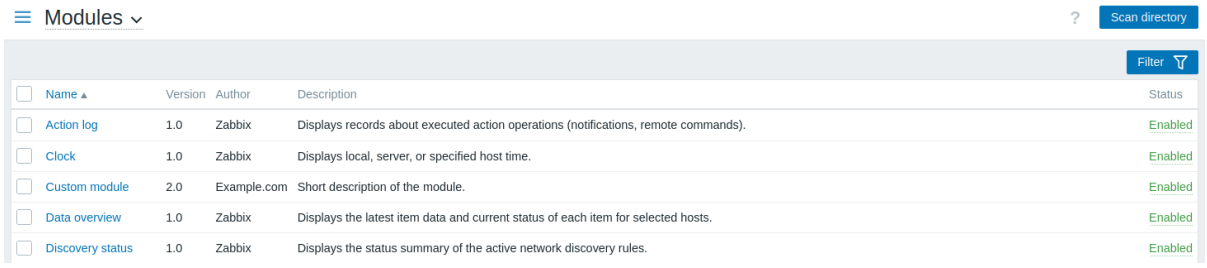
Usando um provedor de serviço de tiles personalizado

O widget Geomap é capaz de carregar imagens de tiles raster de um serviço de provedor de tiles personalizado, hospedado localmente ou de terceiros. Para usar um serviço de provedor de tiles de terceiros personalizado ou uma pasta ou servidor de tiles hospedado localmente, selecione *Other* no campo *Tile provider* e especifique a URL personalizada no campo *Tile URL* usando os placeholders apropriados.

Módulos

Esta seção permite administrar módulos do frontend personalizados e integrados.

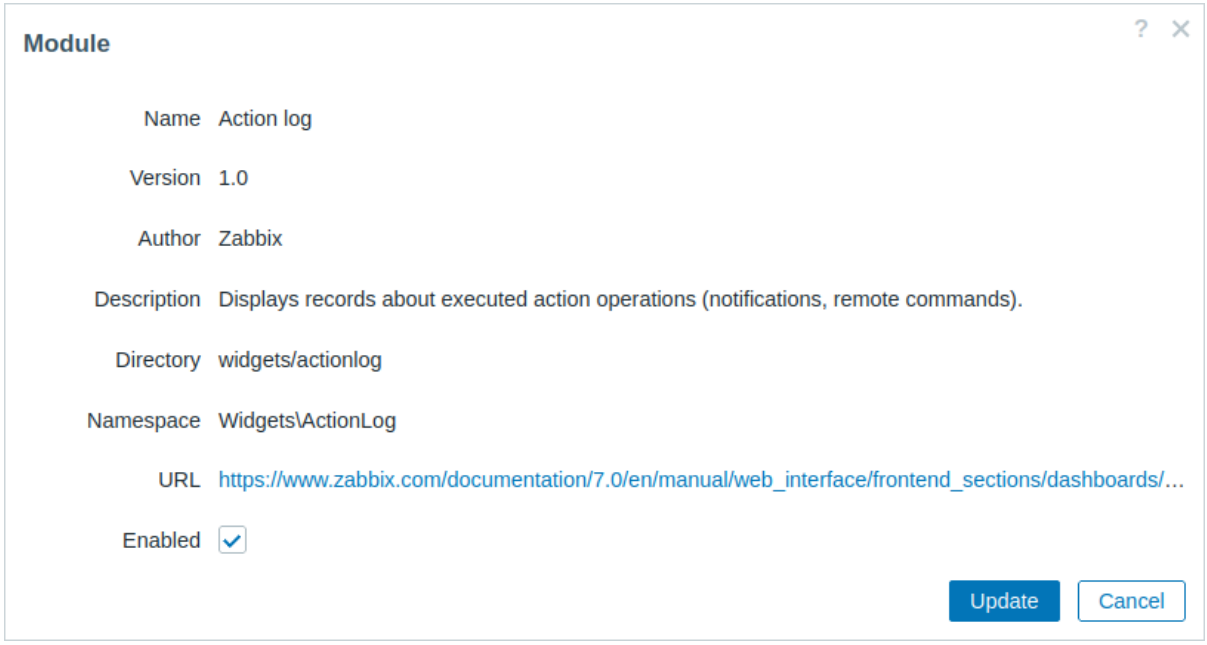
Note:
Se o sinalizador de recurso `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend, o acesso a esta seção será desabilitado.



Clique em *Scan directory* para registrar ou cancelar o registro de módulos personalizados. Os módulos registrados aparecerão na lista; os módulos cujo registro for cancelado serão removidos da lista.

Clique no status do módulo na lista para habilitá-lo ou desabilitá-lo. Você também pode habilitar ou desabilitar vários módulos de uma só vez, selecionando-os na lista e clicando nos botões *Enable/Disable* abaixo da lista.

Clique no nome do módulo na lista para visualizar seus detalhes em uma janela pop-up.

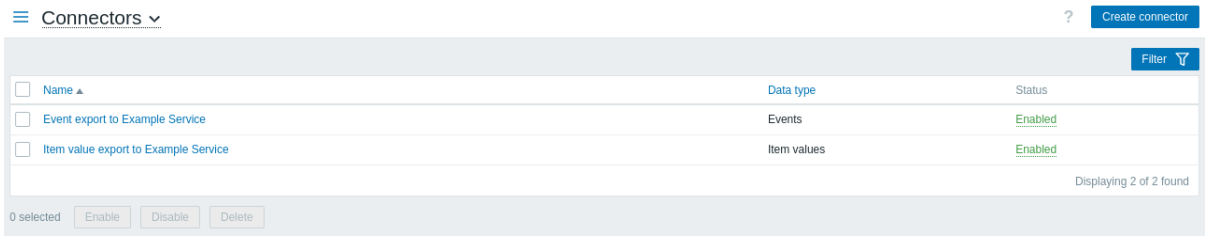


O status do módulo também pode ser atualizado na janela pop-up de detalhes do módulo; para isso, marque ou desmarque a caixa de seleção *Enabled* e clique em *Update*.

Você pode filtrar os módulos por nome ou status (habilitado/desabilitado).

Conectores

Esta seção permite configurar conectores para transmitir dados do Zabbix para sistemas externos por HTTP.



Clique em *Criar conector* para configurar um novo conector.

Você pode filtrar os conectores por nome ou status (habilitado/desabilitado). Clique no status do conector na lista para habilitar/desabilitar um conector. Você também pode habilitar/desabilitar vários conectores em massa selecionando-os na lista e 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 autorregistro de agent serão automaticamente colocados no grupo de hosts selecionado aqui.
Modo padrão de inventário do host	Modo padrão para o inventário do 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/autorregistro do host pela operação *Definir modo de inventário do host*.

Parâmetro	Descrição
Grupo de usuários para mensagem de indisponibilidade do banco de dados	Grupo de usuários para envio da mensagem de alerta de indisponibilidade do banco de dados. O Zabbix server depende da disponibilidade do banco de dados de backend. Ele não pode funcionar sem um banco de dados. Se o banco de dados estiver indisponível, os usuários selecionados poderão ser notificados pelo Zabbix. As notificações serão enviadas ao grupo de usuários definido aqui usando as entradas de mídia de usuário habilitadas. As notificações são transmitidas usando os seguintes tipos de mídia (quando habilitados): e-mail; SMS; scripts de alerta personalizados. Mesmo que uma entrada de mídia webhook esteja configurada e habilitada, ela não acionará notificações. O Zabbix server não será interrompido; ele aguardará até que o banco de dados esteja disponível novamente para continuar o processamento. A notificação consiste no seguinte conteúdo: `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> não é adicionado à mensagem se estiver definido como um valor vazio, e <DB Port> não é adicionado se estiver com 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 continuar indisponível, o gerenciador de alertas repetirá o envio de alertas, mas não com frequência maior que a cada 15 minutos.
Registrar traps SNMP sem correspondência	Registrar trap SNMP se nenhuma interface SNMP correspondente for encontrada.

Autorização

Parâmetro	Descrição
Tentativas de login *Intervalo de bloqueio de login*	Número de tentativas de login sem sucesso 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. Faixa permitida: 30 - 3600 segundos. Sufixos de tempo, por exemplo, 90s, 5m, 1h, são suportados.

Armazenamento de segredos

Parameter	Description
Vault provider	Selecione o software de gerenciamento de segredos para armazenar valores de user macro - *HashiCorp Vault* (padrão) ou *CyberArk Vault*.
Resolve secret vault macros by	Resolver macros do secret vault por: **Zabbix server** - os segredos são recuperados pelo Zabbix server e encaminhados aos 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: Storage of secrets.

Segurança

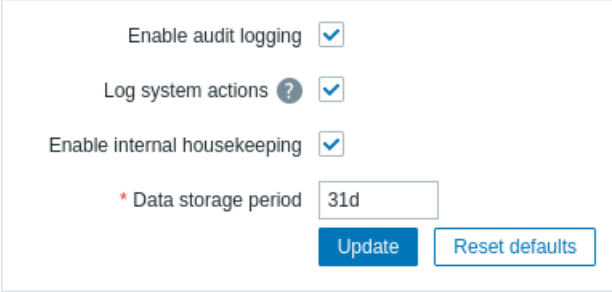
Parâmetro	Descrição
Validar esquemas de URI	Desmarque esta caixa para desabilitar a validação de esquemas de URI (habilitado por padrão). Se marcada, você pode especificar uma lista separada por vírgulas de esquemas de URI permitidos (padrão: http,https,ftp,file,mailto,tel,ssh). Aplica-se a todos os 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`.

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.

Limpeza de dados

Visão geral

A seção *Administration* > *Housekeeping* permite configurar a limpeza interna do Zabbix, que remove automaticamente dados desatualizados ou excluídos pelo usuário do banco de dados. Isso evita que o banco de dados cresça indefinidamente, o que consumiria espaço em disco desnecessário e degradaria o desempenho do banco de dados.

A limpeza pode ser habilitada e configurada para os seguintes tipos de dados:

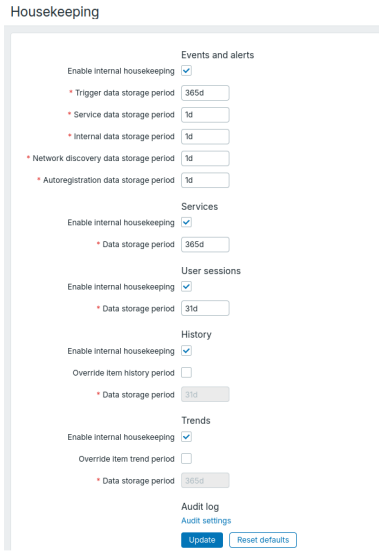
- Eventos e alertas de triggers, serviços, dados internos, descoberta de rede e autoregistro

- Serviços
- Sessões de usuário
- Histórico e tendências para items

Note:
A limpeza do log de auditoria é configurada separadamente.

Configuração

Os seguintes parâmetros estão disponíveis:



Parameter	Description
Enable internal housekeeping	Ative ou desative a limpeza interna (ativada por padrão). Se ativada, o procedimento de housekeeping remove automaticamente do banco de dados os dados que excedem o *Data storage period*.
Data storage period	Especifique por quanto tempo manter os dados antes que eles sejam removidos pelo housekeeper. Obrigatório se a limpeza interna estiver ativada. Intervalo: 1 day (1 hour for history) - 25 years; ou ”0”. Time suffixes (por exemplo, 1d, 1w) são suportados. Para *Events and alerts*, o período de armazenamento de dados é definido separadamente para triggers, services, dados internos, descoberta de rede e autoregistro. O housekeeper remove apenas os eventos que não estão associados a problemas. Por exemplo, um evento de problema/recuperação mais antigo que o *Data storage period* não será removido se estiver associado a um problema não resolvido. Quando o housekeeper remove entidades desatualizadas, ele primeiro remove os problemas e depois os eventos. Para *History* e *Trends*, os períodos de armazenamento de dados também determinam por quanto tempo os dados permanecem visíveis em *Monitoring* > *Latest data*, mesmo que a limpeza interna esteja desativada. Observe que, quando um item ou trigger é excluído, seus problemas e eventos associados também são excluídos.
Override item history period	Se ativada, o período de armazenamento de histórico/tendência especificado na configuração do item é substituído pela configuração *Data storage period* (exceto para items com a opção *Do not store* ativada). Esta opção pode ser usada mesmo quando a limpeza interna está desativada e um housekeeper externo é utilizado.
Override item trend period	

Se estiver usando **ClickHouse**, a seção *ClickHouse data storage period* ficará disponível. Ela mostra o período de armazenamento

de dados para cada tipo de valor de item. Para alterar o período, consulte Configuring ClickHouse data storage period. Observe que o housekeeper não exclui dados do ClickHouse.

Se estiver usando **Elasticsearch**, a seção *Elasticsearch data storage period* ficará disponível. Ela mostra os tipos de valor de item salvos no Elasticsearch, com o período de armazenamento de dados exibido como *Unknown*. Para controlar por quanto tempo os dados são mantidos, consulte Index Lifecycle Management. Observe que o housekeeper não exclui dados do Elasticsearch.

Se estiver usando **TimescaleDB**, a seção *History, trends and audit log compression* ficará disponível.

Note:
Para TimescaleDB, ative *Override item history period*, *Override item trend period* e *Enable internal housekeeping* para history e trends para aproveitar totalmente o particionamento automático. Se essas opções estiverem desativadas, os dados mantidos nas tabelas de history e trends ainda serão particionados, mas o housekeeper não removerá partições desatualizadas, e avisos de configuração serão exibidos. Quando a remoção de partições desatualizadas estiver ativada, o Zabbix server e o frontend não acompanharão mais os items excluídos, e o histórico desses items será limpo quando uma partição desatualizada for excluída.

O housekeeper é executado automaticamente a cada hora por padrão. Você pode alterar a frequência de execução definindo o parâmetro de configuração `HousekeepingFrequency` do Zabbix server.

O botão *Reset defaults* permite reverter quaisquer alterações feitas.

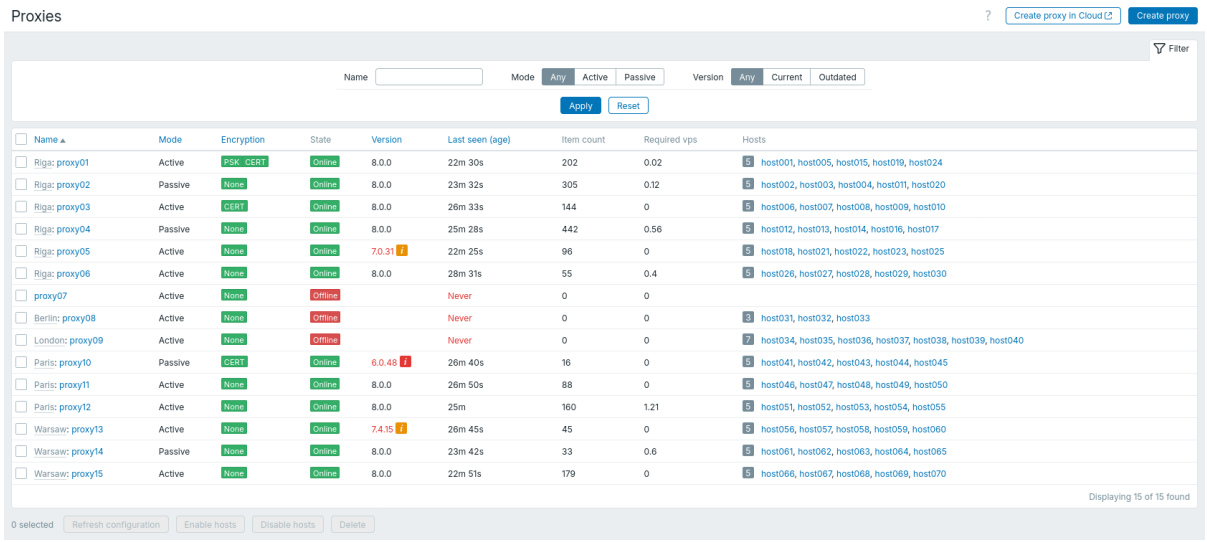
Proxies

Visão geral

Na seção *Administração* > *Proxies*, os proxies para monitoramento distribuído podem ser configurados no frontend do Zabbix.

Proxies

É exibida uma lista dos proxies existentes com seus detalhes.



Dados exibidos:

Coluna	Descrição
Nome	Nome do proxy. Clicar no nome do proxy abre o formulário de configuração do proxy. Se o proxy pertencer a um grupo de proxies, o nome do grupo será 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.

Coluna	Descrição
Estado	Estado do proxy: **Desconhecido** - o proxy foi criado enquanto o Zabbix server estava inativo ou o server ainda não atualizou o estado; **Online** - o proxy se comunicou com o Zabbix server dentro do período de failover (o proxy passivo respondeu à solicitação do server; o proxy ativo enviou uma solicitação); **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 compatível, o número da versão será destacado (em vermelho) e um ícone de status de informação (amarelo ou vermelho) será exibido. Passe o cursor sobre o ícone para ver os detalhes.
Última visualização (idade)	O horário em que o proxy foi visto pela última vez pelo server.
Contagem de itens	O número de itens habilitados em hosts habilitados atribuídos ao proxy.
VPS necessário	Desempenho necessário do proxy (o número de valores que precisam ser coletados por segundo).
Hosts	Quantidade 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. Para implantar um novo proxy no Zabbix Cloud, clique no botão *Criar proxy no Cloud* ao lado dele. Isso abre o Zabbix Cloud em uma nova aba do navegador, onde você pode criar um nó de proxy. Depois que o nó de proxy for criado, configure o proxy usando o botão *Criar proxy*. Para obter mais informações, consulte Configurar um proxy do Zabbix Cloud.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

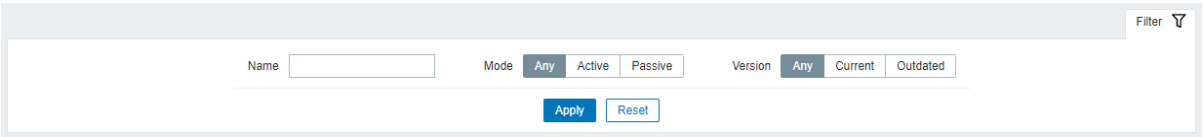
- *Atualizar configuração* - atualiza a configuração dos proxies;
- *Habilitar hosts* - altera o status dos hosts monitorados pelo proxy para *Monitorado*;
- *Desabilitar hosts* - altera o status dos hosts monitorados pelo proxy para *Não monitorado*;
- *Excluir* - exclui os proxies.

Para usar essas opções, marque as caixas de seleção antes dos respectivos proxies e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os proxies nos quais está interessado. Para melhorar o desempenho da pesquisa, os dados são pesquisados com as macros não resolvidas.

O link *Filtro* está disponível acima da lista de proxies. Ao clicar nele, um filtro ficará disponível, permitindo filtrar os proxies por nome, modo e versão. Observe que a opção de filtro *Desatualizado* exibe tanto proxies desatualizados (parcialmente compatíveis) quanto proxies sem suporte.



Grupos de proxy

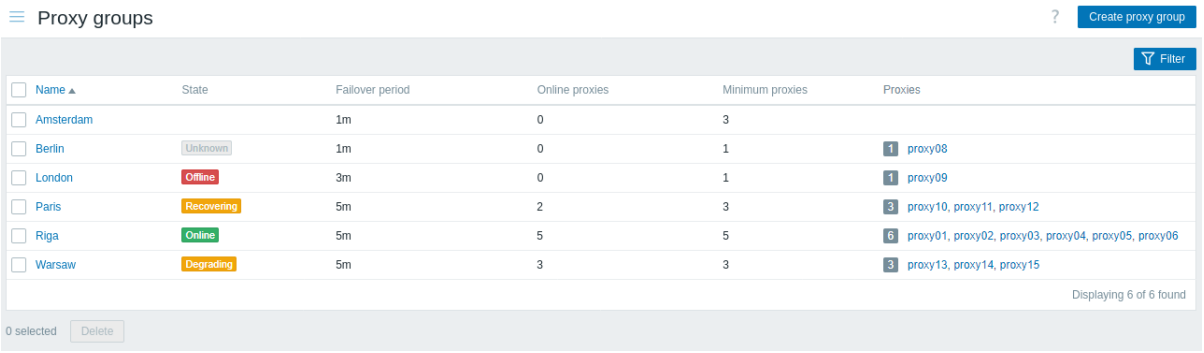
Visão geral

Em *Administration* > *Proxy groups* é possível configurar grupos de proxy.

Os grupos de proxy são usados em proxy load balancing com distribuição automatizada de hosts entre proxies e alta disponibilidade entre proxies.

Grupos de proxy

É exibida uma lista dos grupos de proxy existentes com seus detalhes.



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 inativo ou o server ainda não atualizou o estado; **Online** - o número mínimo de proxies se comunicou com o Zabbix server durante o período de failover; **Degradando** - o grupo ficará offline em breve porque o número de proxies online caiu abaixo do limite de *Mínimo de proxies*; **Offline** - menos que o número mínimo de proxies se comunicou com o Zabbix server durante o período de failover; **Recuperando** - o grupo voltará a ficar online em breve porque o número de proxies online atingiu o limite de *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 necessário para manter o grupo de proxy online.
Proxies	Quantidade de proxies no grupo e uma lista dos proxies no grupo, com links para o formulário de configuração do proxy. O número máximo de proxies listados é limitado pelo valor de *Número máximo 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:

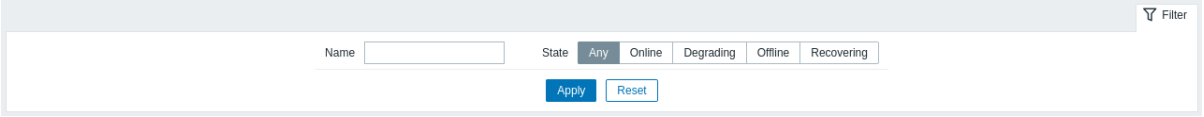
- *Delete* - exclui os grupos de proxy.

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de proxy e, em seguida, clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os grupos de proxy nos quais você está interessado. Para melhor desempenho da pesquisa, os dados são pesquisados com macros não resolvidas.

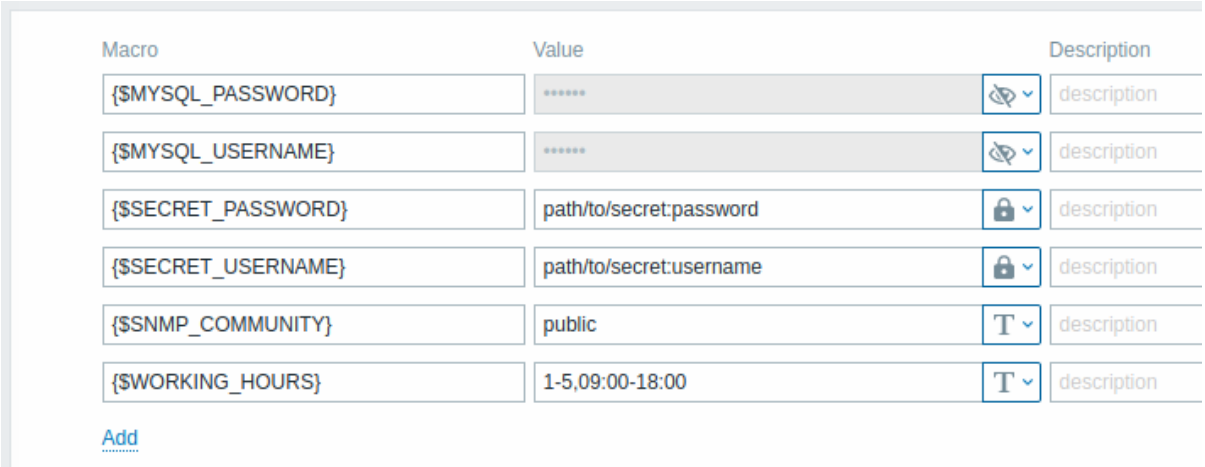
O link *Filter* está disponível acima da lista de grupos de proxy. Se você clicar nele, um filtro ficará disponível, no qual você pode filtrar grupos de proxy por nome e status.



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.



Fila

Visão geral

Na seção *Administração* > *Fila*, são exibidos os itens que aguardam atualização.

Idealmente, ao abrir esta seção, tudo deve estar "verde", o que significa que não há itens na fila. Se todos os itens forem atualizados sem atraso, não haverá itens aguardando. No entanto, devido ao desempenho insuficiente do server, alguns itens podem sofrer atrasos, e essas informações são exibidas nesta seção. Para obter mais detalhes, consulte a seção Fila.

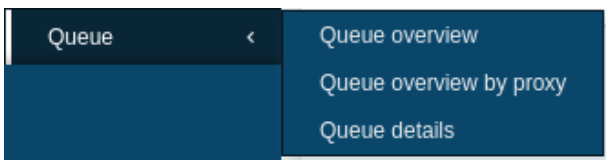
Note:
O proxy do Zabbix não tem conhecimento dos períodos de manutenção; consulte Cálculo das filas durante a manutenção para obter mais detalhes.

Note:
A fila está disponível somente quando o server do Zabbix está em execução. Os itens não são contabilizados na fila se a interface do item ficar indisponível devido a problemas de conexão ou se o agent não estiver funcionando corretamente.

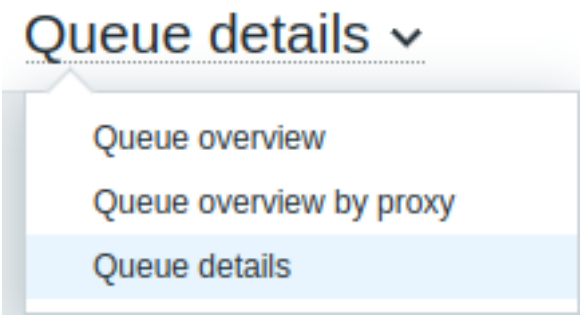
A seção *Administração* > *Fila* contém as seguintes páginas:

- Visão geral da fila - exibe a fila por tipo de item;
- Visão geral da fila por proxy - exibe a fila por proxy;
- Detalhes da fila - exibe uma lista de itens atrasados.

A lista de páginas disponíveis é exibida ao clicar em *Fila* na seção do menu *Administração*. Também é possível alternar entre as páginas usando a lista suspensa de títulos no canto superior esquerdo.



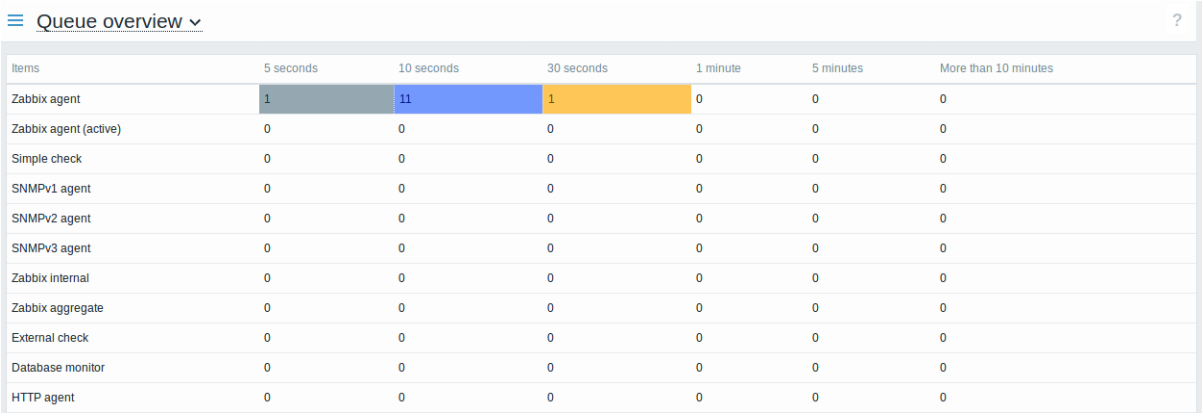
Menu de terceiro nível.



Lista suspensa de títulos.

Visão geral por tipo de item

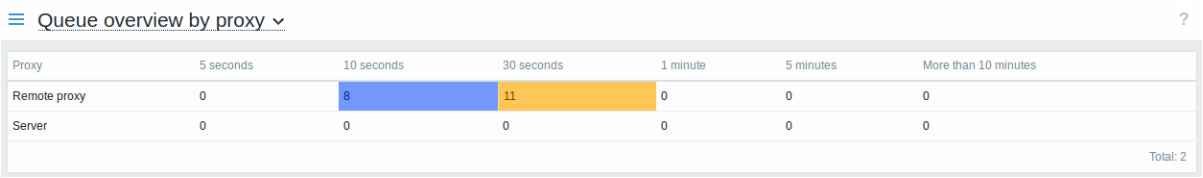
Nesta tela é fácil localizar se o problema está relacionado a um ou vários tipos de item.



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

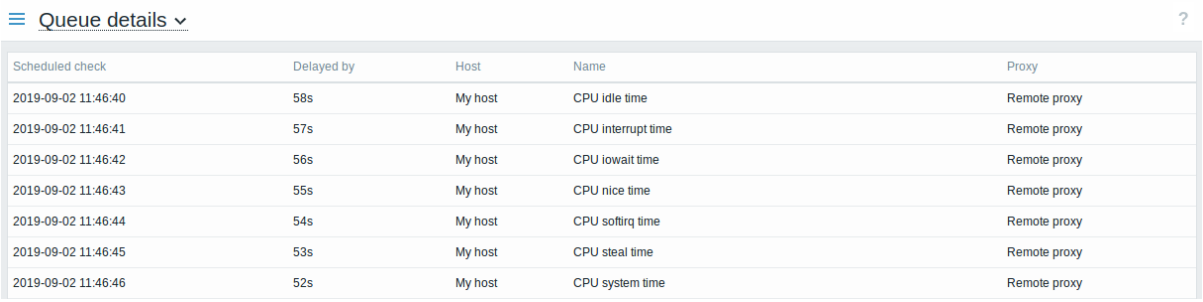
Nesta tela é fácil localizar se o problema está relacionado a um dos proxies ou ao server.



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 se deparar com uma situação em que nenhum dado é exibido e a seguinte mensagem de erro aparece:



A mensagem de erro, nesse caso, é a seguinte:

```
Cannot display item queue. Permission denied
```

Isso acontece quando os parâmetros de configuração do PHP no arquivo de configuração do frontend - `$ZBX_SERVER` ou ambos `$ZBX_SERVER` e `$ZBX_SERVER_PORT` - apontam para um server Zabbix existente que usa um banco de dados diferente.

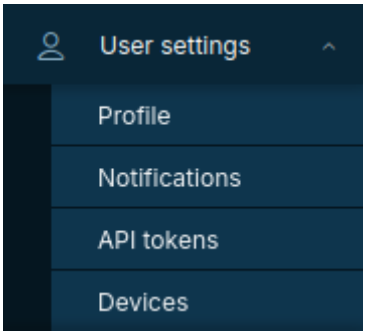
Configurações do usuário

Visão geral

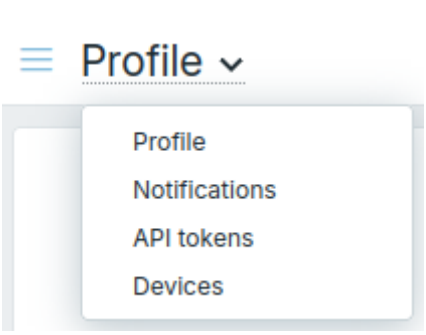
Dependendo da sua função de usuário, a seção *Configurações do usuário* pode conter as seguintes páginas:

- *Perfil* ou *Perfil do usuário* - para personalizar determinados 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;
- *Dispositivos* - para gerenciar os dispositivos móveis pertencentes ao usuário atual.

A lista de páginas disponíveis é exibida 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 do título no canto superior esquerdo.



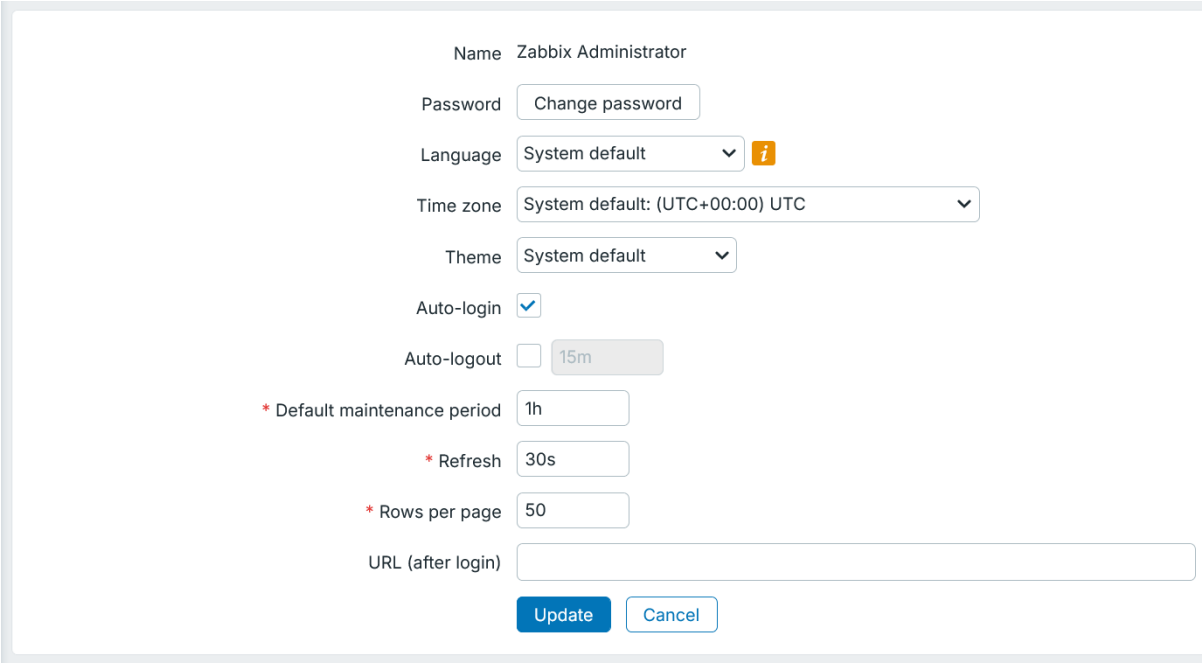
Menu de terceiro nível.



Menu suspenso do título.

Perfil do usuário

A seção *Perfil do usuário* oferece opções para definir um idioma personalizado da interface, um tema de cores, o número de linhas exibidas nas listas etc. As alterações feitas aqui serão aplicadas somente ao usuário atual.



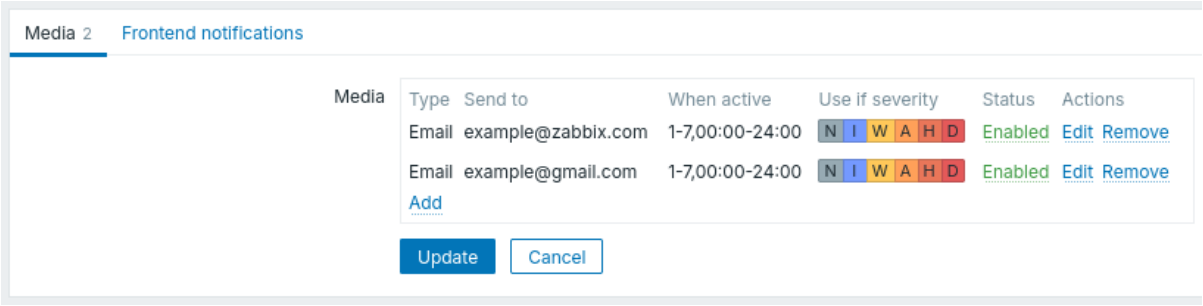
Parâmetro	Descrição
Nome	O nome e o sobrenome do usuário são exibidos. Na ausência de nome e sobrenome, o nome de usuário é exibido.
Senha	Clique no botão *Alterar senha* para abrir três campos: *Senha antiga*, *Nova senha*, *Nova senha (novamente)*. Após uma alteração de senha bem-sucedida, o usuário será desconectado de todas as sessões ativas. Observe que a senha só pode ser alterada para usuários que utilizam a autenticação interna do Zabbix.

Parâmetro	Descrição
Idioma	Selecione o idioma da interface desejado ou selecione **Padrão do sistema** para usar as configurações padrão do sistema. Selecionar *Inglês (en_US)* também ativará o formato de hora/data dos EUA no frontend. Para obter mais informações, consulte Instalação de idiomas adicionais do 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 cores específico para o seu perfil: **Padrão do sistema** - usa as configurações padrão do sistema; **Azul** - tema azul padrão; **Azul (clássico)** - tema azul padrão usando fontes legadas; **Escuro** - tema escuro alternativo; **Azul escuro** - tema azul escuro; **Escuro (clássico)** - tema escuro alternativo usando fontes legadas; **Alto contraste claro** - tema claro com alto contraste; **Alto contraste escuro** - tema escuro com alto contraste.
Login automático	Marque esta caixa de seleção para que o Zabbix se lembre de você e faça seu login automaticamente por 30 dias. Ao fazer login com *Lembrar por 30 dias* ativado: - O temporizador de logout automático é redefinido (a sessão permanece ativa até o fim do período de 30 dias). - O login automático é ativado em 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. Para isso, são usados cookies do navegador.
Logout automático	Com esta caixa de seleção marcada, você será desconectado automaticamente após o número de segundos definido (mínimo de 90 segundos, máximo de 1 dia). Sufixos de tempo são aceitos, por exemplo: 90s, 5m, 2h, 1d. Observe que esta opção não funcionará nos seguintes casos: * Quando as páginas do menu *Monitoramento* realizarem atualizações de informações em segundo plano. Se páginas que atualizam dados em um intervalo específico (painéis, gráficos, dados mais recentes etc.) permanecerem abertas, a duração da sessão será estendida, desativando efetivamente o recurso de logout automático. * Se o login for feito com a opção *Lembrar de mim por 30 dias* marcada; nesse caso, o temporizador de logout automático será substituído, mantendo a sessão ativa durante todo o período. * Quando a autenticação for realizada por meio de um provedor de identidade SSO (IdP), o logout automático encerrará somente a sessão do Zabbix. A sessão SSO com o IdP permanecerá ativa, portanto o usuário poderá conseguir fazer login novamente sem inserir uma senha. *Logout automático* também pode aceitar ”0”, o que significa que o recurso de logout automático será desativado após a atualização das configurações do perfil.
Período de manutenção padrão	Defina o período de manutenção padrão para manutenções ad hoc.
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 aceitos, 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 menores.
URL (após o login)	Defina uma URL específica a ser exibida após o login. Em vez de *Painéis*, que é o 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 os detalhes de mídia do usuário, como os tipos de mídia e endereços a serem usados e quando usá-los para entregar notificações.



Se o tipo de mídia tiver sido desabilitado:

- um ícone de informação amarelo será exibido após o nome;
- "Desabilitado" será exibido na coluna *Status*.

Note:
As permissões do usuário para alterar seus próprios detalhes de mídia podem ser concedidas/revogadas com base em sua função de usuário (consulte a opção *Criar e editar a própria mídia*). As permissões do usuário Super admin para alterar os detalhes de mídia de outros usuários também podem ser concedidas/revogadas com base em sua função de usuário (consulte a opção *Criar e editar mídia de usuário*)

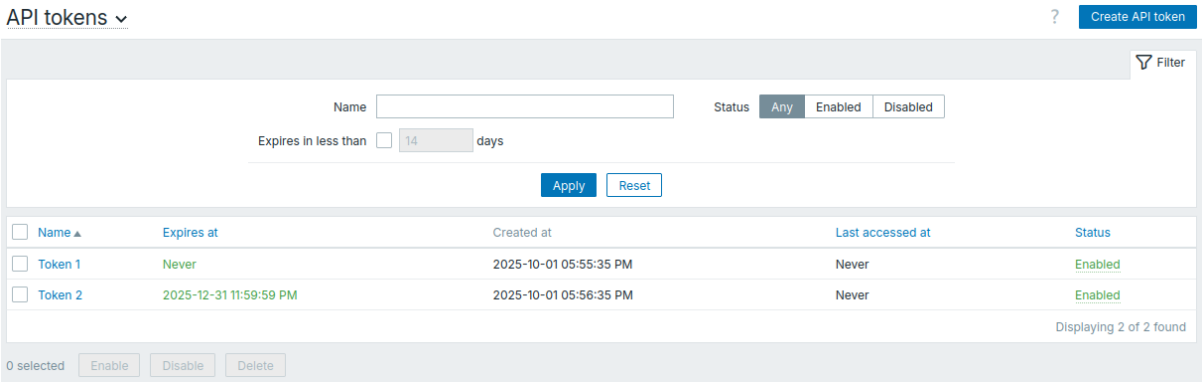
Observe que, para usuários provisionados:

- a mídia de usuário provisionada não pode ser excluída;
- a mídia de usuário provisionada pode ser desabilitada/habilitada;
- campos dessa mídia de usuário provisionada, como *Quando ativo*, *Usar se severidade* e *Habilitado*, podem ser editados manualmente;
- mídia de usuário adicional pode ser adicionada manualmente (por exemplo, endereços de e-mail adicionais) para usuários provisionados;
- a mídia de usuário adicionada manualmente pode ser excluída.

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 somente se a ação *Manage API tokens* estiver □□□□□□ida nas configurações de user role.



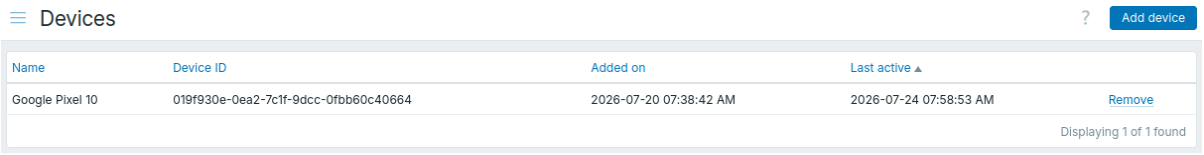
Você pode filtrar os tokens de API por nome, data de expiração ou status (*Enabled/Disabled*). Clique no status do token na lista para ativar/desativar rapidamente um token. Você também pode ativar/desativar vários tokens de uma vez, selecionando-os na lista e clicando nos botões *Enable/Disable* abaixo da lista.

Attention:
Os usuários não podem visualizar o valor do *Auth token* dos tokens atribuídos a eles no Zabbix. O valor do *Auth token* é exibido apenas uma vez - imediatamente após a criação de um token. Se o token tiver sido perdido, ele precisa ser gerado novamente.

Dispositivos

A seção *Dispositivos* permite adicionar ou remover seus dispositivos móveis para uso com o aplicativo móvel do Zabbix.

Esta seção é exibida se `EnableMobileDevices` estiver habilitado.



Para adicionar um dispositivo:

1. Clique em *Adicionar dispositivo*.
2. Leia o código QR no aplicativo móvel do Zabbix.

Note:
Quando você adiciona um dispositivo, o Zabbix também adiciona o tipo de mídia de notificação push às suas mídias de usuário. Quando você remove o dispositivo, essa mídia de usuário também é removida. Se você tiver vários tipos de mídia de notificação push configurados, o Zabbix adicionará aquele que estiver habilitado e aparecer primeiro em ordem alfabética.

Os dispositivos adicionados aparecem em uma lista. Ela mostra o nome e o ID dos dispositivos, além das datas e horários em que foram adicionados e estiveram ativos pela última vez.

Você também pode remover dispositivos. Quando você remove um dispositivo, suas credenciais de acesso são revogadas e removidas do Zabbix, após o que ele não poderá mais acessar o Zabbix.

Superadministradores também podem gerenciar dispositivos de outros usuários em *Usuários* > *Dispositivos*.

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.

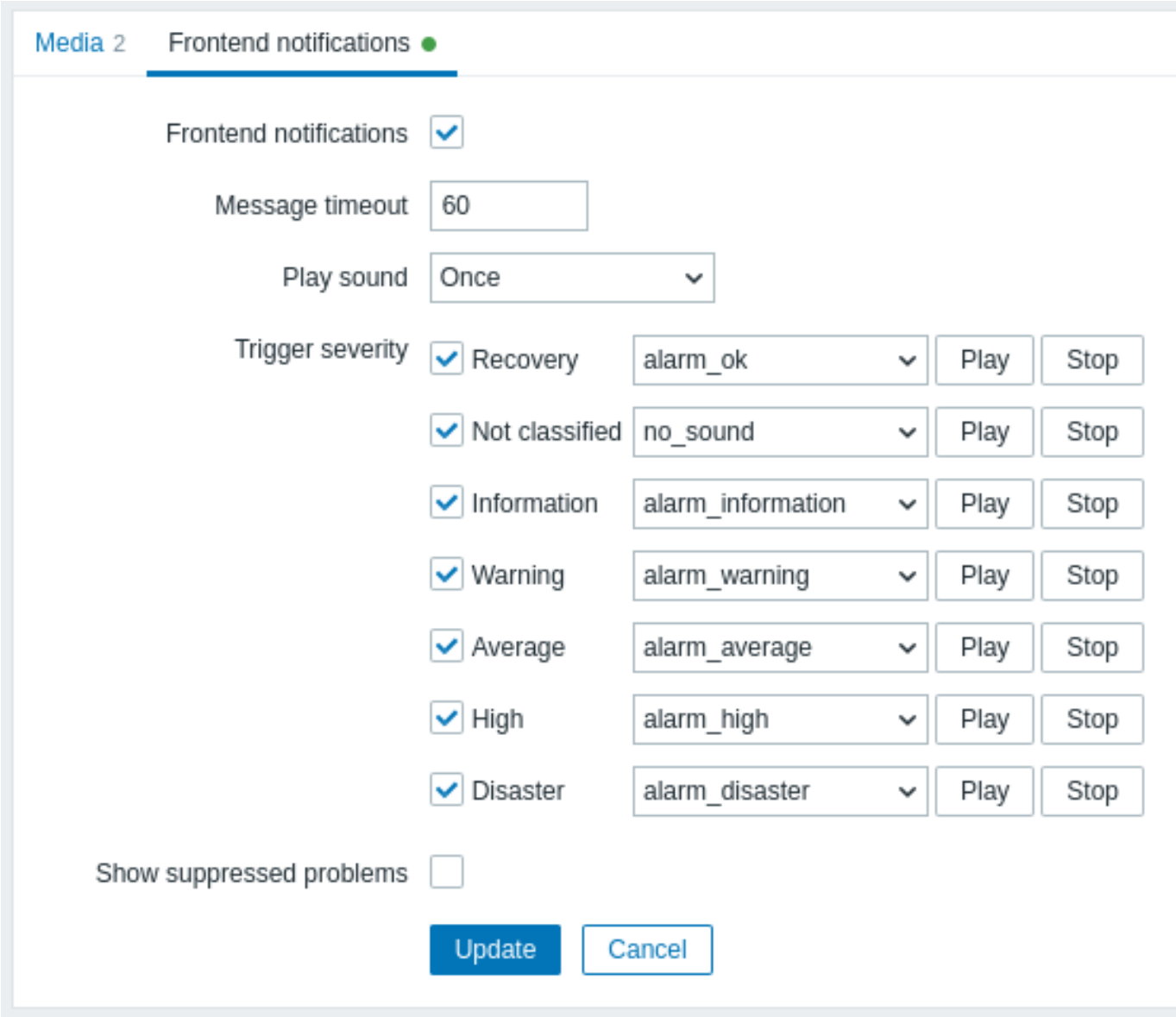
Sem notificações globais, ao trabalhar fora das seções *Problemas* ou *Dashboard*, você não receberia nenhuma informação sobre problemas atuais. As notificações globais garantem que essas informações sejam exibidas, independentemente de sua localização atual dentro do frontend do Zabbix.

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

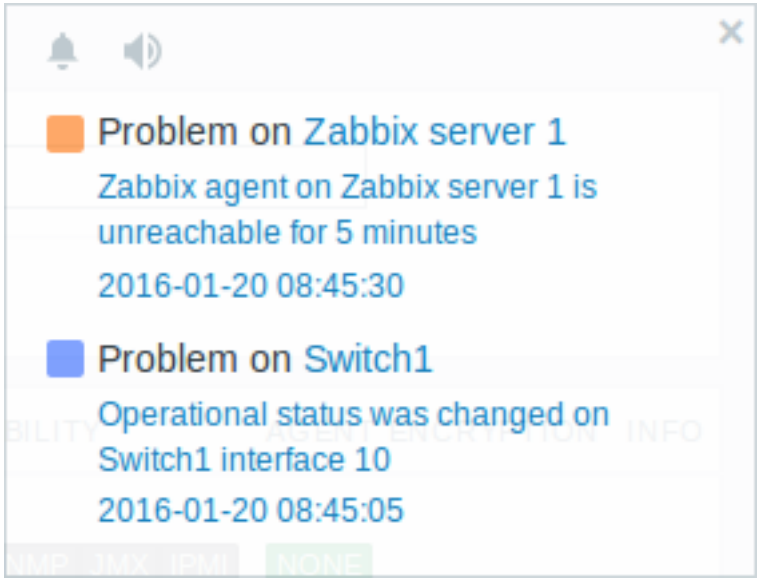
As notificações globais podem ser ativadas por usuário na aba **Notificações do frontend** da seção Notificações.



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.

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.

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 será exibido, 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 qualquer destaque.

Atributos pesquisáveis

Os hosts podem ser pesquisados pelas seguintes propriedades:

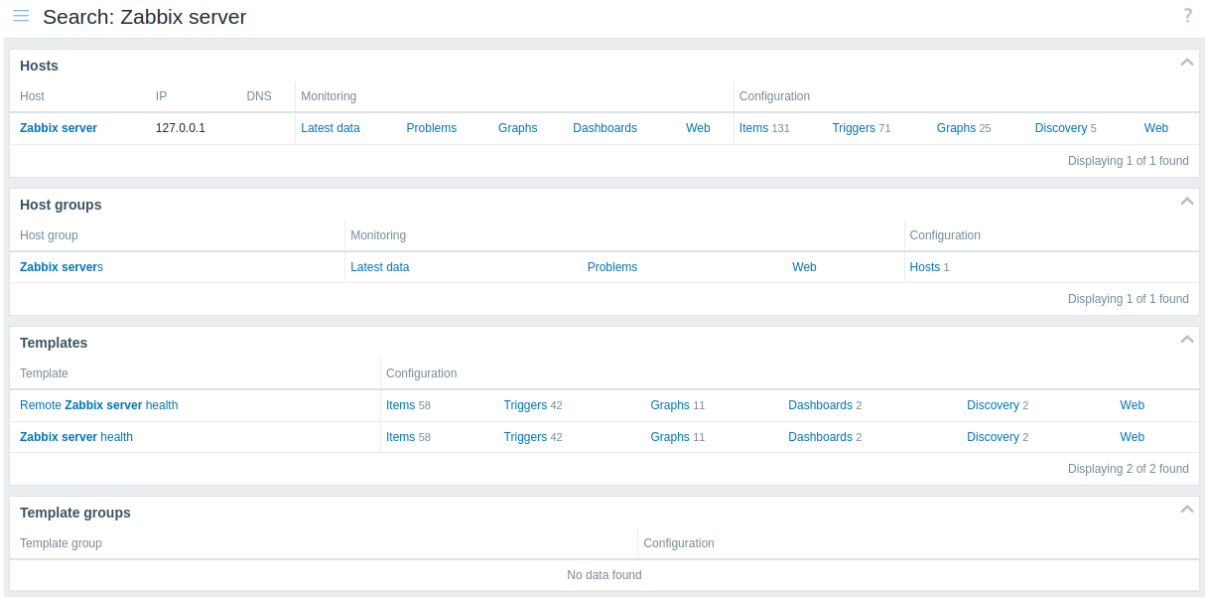
- Nome do host
- Nome visível
- Endereço IP
- Nome DNS

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 será 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

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 ativar o modo de manutenção, abra o arquivo `maintenance.inc.php` (localizado em `/conf` no diretório de documentos HTML do Zabbix no web server) e remova os comentários das seguintes linhas:

```
// Modo de manutenção.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array de endereços IP autorizados a se conectar ao frontend (opcional).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Mensagem exibida na tela de aviso (opcional).
$ZBX_GUI_ACCESS_MESSAGE = 'Estamos atualizando o banco de dados MySQL até as 15:00. Aguarde...';
```

Note:
Na maioria dos casos, o arquivo `maintenance.inc.php` está localizado em `/conf` no diretório de documentos HTML do Zabbix no web server. No entanto, alguns sistemas operacionais e web servers podem usar um local diferente.
Por exemplo, o local para:

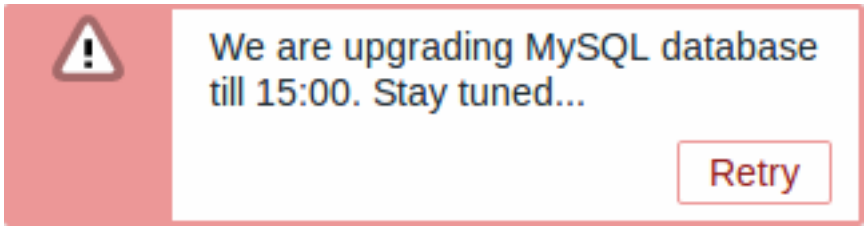
- SUSE e RedHat é `/etc/zabbix/web/maintenance.inc.php`.
- Sistemas baseados em Debian é `/usr/share/zabbix/conf/`.

Consulte também Copiando arquivos PHP.

Parâmetro	Detalhes
ZBX_DENY_GUI_ACCESS	Se definido com qualquer valor, o modo de manutenção será habilitado. Para desabilitar o modo de manutenção, comente ou exclua.
ZBX_GUI_ACCESS_IP_RANGE	Array de endereços IP autorizados a 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, será usada a mensagem padrão *'Zabbix está em manutenção'*.

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.

Parâmetros da página

Visão geral

A maioria das páginas da interface web do Zabbix suporta vários parâmetros HTTP GET que controlam o que será exibido. Eles podem ser passados especificando pares de parâmetro=valor após a URL, separados da URL por um ponto de interrogação (?) e entre si por e comercial (&).

Monitoring > Problems

Os seguintes parâmetros são suportados:

Parameter	Description	Example
show	Opção de filtro *Show*. Valores possíveis: 1 - problemas recentes; 2 - todos; 3 - em estado de problema.	show=1
name	Opção de filtro *Problem*: string livre.	name=Zabbix agent

Parameter	Description	Example
severities	Opção de filtro *Severity*: array de severidades selecionadas no formato `severities[*]=*` (substitua `*` pelo nível de severidade). Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - médio; 4 - alto; 5 - desastre.	severities[3]=3
inventory	Opção de filtro *Host inventory*: array de campos de inventário `[field],[value]`	inventory[0][field]=type&invento
evaltype	Opção de filtro *Tags*: método de avaliação da tag evaluation method. Valores possíveis: 0 - and/or; 2 - or.	evaltype=0
tags	Opção de filtro *Tags*: array de tags definidas `[tag], [operator], [value]` Valores possíveis de operator: 0 - contém; 1 - igual a; 2 - não contém; 3 - não é igual a; 4 - existe; 5 - não existe.	tags[0][tag]=target&tags[0][ope
show_tags	Opção de filtro *Show tags*. Valores possíveis: 0 - nenhum; 1 - um; 2 - dois; 3 - três.	show_tags=3
tag_name_format	Opção de filtro *Tag name*. Valores possíveis: 0 - nome completo; 1 - nome abreviado; 2 - nenhum.	tag_name_format=1
tag_priority	Opção de filtro *Tag display priority*: string separada por vírgulas da prioridade de exibição das tags	tag_priority=customer, target
show_suppressed	Opção de filtro *Show suppressed problems*. Valores possíveis: 1 - mostrar; 2 - não mostrar.	show_suppressed=1
acknowledgement_status	Opção de filtro *Acknowledgement status*. Valores possíveis: 0 - todos; 1 - não reconhecidos; 2 - reconhecidos.	acknowledgement_status=0
acknowledged_by_me	Opção de filtro *By me*; suportado apenas com `acknowledgement_status=2`. Valores possíveis: 0 - desativado; 1 - ativado.	acknowledged_by_me=1
compact_view	Opção de filtro *Compact view*. Valores possíveis: 0 - desativado; 1 - ativado.	compact_view=1

Parameter	Description	Example
highlight_row	Opção de filtro *Highlight whole row*. Valores possíveis: 0 - desativado; 1 - ativado.	highlight_row=1
filter_name	Opção de propriedades do filtro *Name*: string livre	filter_name=Databases
filter_show_counter	Opção de propriedades do filtro *Show number of records*. Valores possíveis: 0 - desativado; 1 - ativado.	filter_show_counter=1
filter_custom_time	Opção de propriedades do filtro *Set custom time period*. Valores possíveis: 0 - desativado; 1 - ativado.	filter_custom_time=1
sort	Coluna pela qual ordenar. Valores possíveis: clock - ordenar pela coluna *Time*; host - ordenar pela coluna *Host*; severity - ordenar pela coluna *Severity*; name - ordenar pela coluna *Problem*.	sort=clock
sortorder	Ordem de classificação dos resultados. Valores possíveis: DESC - decrescente; ASC - crescente.	sortorder=DESC
age_state	Opção de filtro *Age less than*; suportado apenas com `show`=3. Valores possíveis: 0 - desativar o parâmetro `age`; 1 - ativar o parâmetro `age`.	age_state=1
age	Opção de filtro *Age less than*: inteiro, número de dias; suportado apenas com `age_state`=1 e `show`=3.	age=7
groupids	Opção de filtro *Host groups*: 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 *Show timeline*; não suportado com `compact_view`=1. Valores possíveis: 0 - não mostrar; 1 - mostrar.	show_timeline=1
details	Opção de filtro *Show details*. 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 Problems.

Modo quiosque

O modo quiosque nas páginas compatíveis do frontend pode ser ativado usando parâmetros de URL. Por exemplo, nos painéis:

- `/zabbix.php?action=dashboard.view&kiosk=1` - ativar o modo quiosque
- `/zabbix.php?action=dashboard.view&kiosk=0` - ativar o modo normal

Apresentação de slides

É possível ativar uma apresentação de slides no dashboard:

- `/zabbix.php?action=dashboard.view&slideshow=on` - ativar apresentação de slides

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 para os 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 de seleção 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 de TTL, em segundos, usado para invalidar o cache de dados da resposta do Vault. Defina 0 para desativar o cache da 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 exibir.

Padrão: 1000

Idiomas adicionais do frontend

Visão geral Para usar qualquer idioma diferente do inglês na interface web do Zabbix, o locale correspondente deve estar instalado no servidor web. Além disso, a extensão gettext do PHP é necessária para que as traduções funcionem.

Instalando locales 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 locales necessários. Como o Zabbix usa codificação UTF-8, você precisa selecionar locales com charset UTF-8.

Agora execute:

```
locale-gen
```

Reinicie o web server. Observe que, se o PHP-FPM estiver instalado e sendo usado pelo web server, ele também precisa ser reiniciado junto com o web server.

Os locales 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:

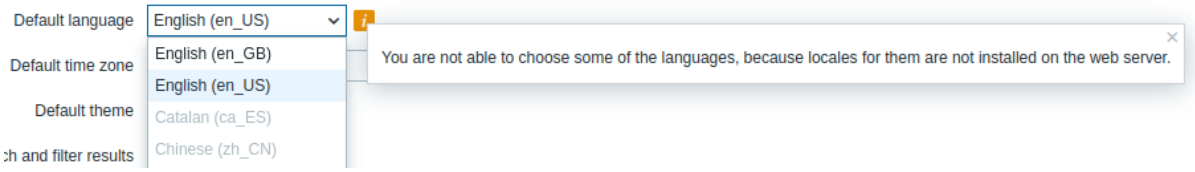
```
make gettext
locale/make_mo.sh
```

Esta etapa não é necessária ao instalar o Zabbix a partir de pacotes ou arquivos `tar.gz` de origem.

Selecionando um idioma Há várias maneiras de selecionar um idioma na interface web do Zabbix:

- Ao instalar a interface web - no assistente de instalação do frontend. O idioma selecionado será definido como padrão do sistema.
- Após a instalação, o idioma padrão do sistema pode ser alterado em *Administration* > *General* > *GUI*.
- O idioma de um usuário específico pode ser alterado no perfil do usuário.

Se um locale para um idioma não estiver instalado na máquina, esse idioma aparecerá esmaecido no seletor de idioma do Zabbix. Um ícone laranja é exibido ao lado do seletor de idioma se pelo menos um locale estiver ausente. Ao clicar nesse ícone, a seguinte mensagem será exibida: "Você não consegue escolher alguns dos idiomas, porque os locales para eles não estão instalados no web server."



Criando seu próprio tema

Visão geral

Por padrão, o Zabbix fornece vários temas predefinidos do frontend. Para criar um tema personalizado, siga o procedimento passo a passo fornecido aqui.

Etapa 1: Crie seu tema

Para criar seu próprio tema:

1. Crie um arquivo CSS do zero ou copie um tema existente (como `blue-theme.css`, `dark-theme.css` ou outro) do diretório `assets/styles/` e modifique-o.
2. Salve seu tema (por exemplo, `custom-theme.css`) no diretório `assets/styles/`.

Etapa 2: Ative seu tema na lista de temas

Para disponibilizar seu tema personalizado no frontend, você deve adicioná-lo à lista de temas retornada pelo método `APP::getThemes()`. Isso é feito substituindo o método `ZBase::getThemes()` dentro da classe `APP`, conforme mostrado a seguir:

1. Abra o arquivo `include/classes/core/APP.php` e localize a classe `APP`:

```
class APP extends ZBase {

}
```

2. Dentro da classe, antes da chave de fechamento final `}`, adicione o método a seguir:

```
public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme')
        ]);
}
```

onde:

- `custom-theme` - o identificador interno do seu tema, que deve corresponder ao nome do seu arquivo CSS sem a extensão `.css`;
- `Custom theme` - o nome de exibição que você verá no frontend.

Para adicionar vários temas, liste-os no array, separando cada entrada com uma vírgula. A vírgula após a última entrada é opcional.

Por exemplo:

```
    public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme'),
            'anothertheme' => _('Another theme'),
            'onemoretheme' => _('One more theme')
        ]);
    }
```

Etapa 3: Altere as cores dos gráficos

Para alterar as cores dos gráficos, um registro do novo tema deve ser adicionado à tabela de banco de dados `graph_theme`.

Exemplo de adição de um registro a um banco de dados MySQL/MariaDB:

```
mysql -u zabbix -p
# Enter password:
```

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

Etapa 4: Usar o novo tema

No frontend do Zabbix, você pode definir este tema como o tema padrão ou selecioná-lo em um perfil de usuário.

Modo de depuração

Visão geral

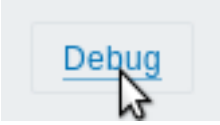
O modo de depuração pode ser usado para diagnosticar problemas de desempenho com as páginas do frontend.

Configuração

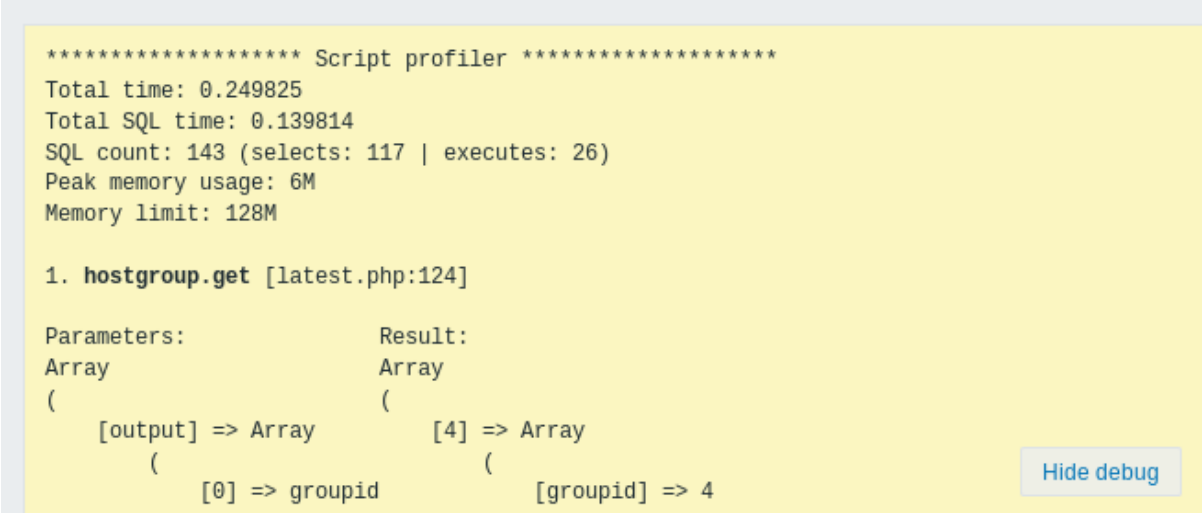
O modo de depuração pode ser ativado para usuários individuais que pertencem a um grupo de usuários:

- ao configurar um grupo de usuários;
- ao visualizar grupos de usuários configurados.

Quando o *Debug mode* está habilitado para um grupo de usuários, seus usuários verão um botão *Debug* no canto inferior direito da janela do navegador:



Clicar no botão *Debug* 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, essa janela pode ser usada para buscar a causa raiz do problema.

Warning:
O *Debug mode* habilitado afeta negativamente o desempenho do frontend.

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)

Nome	Descrição	Valores	Expira/Max-Age	HttpOnly[1]	Secure[2]
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

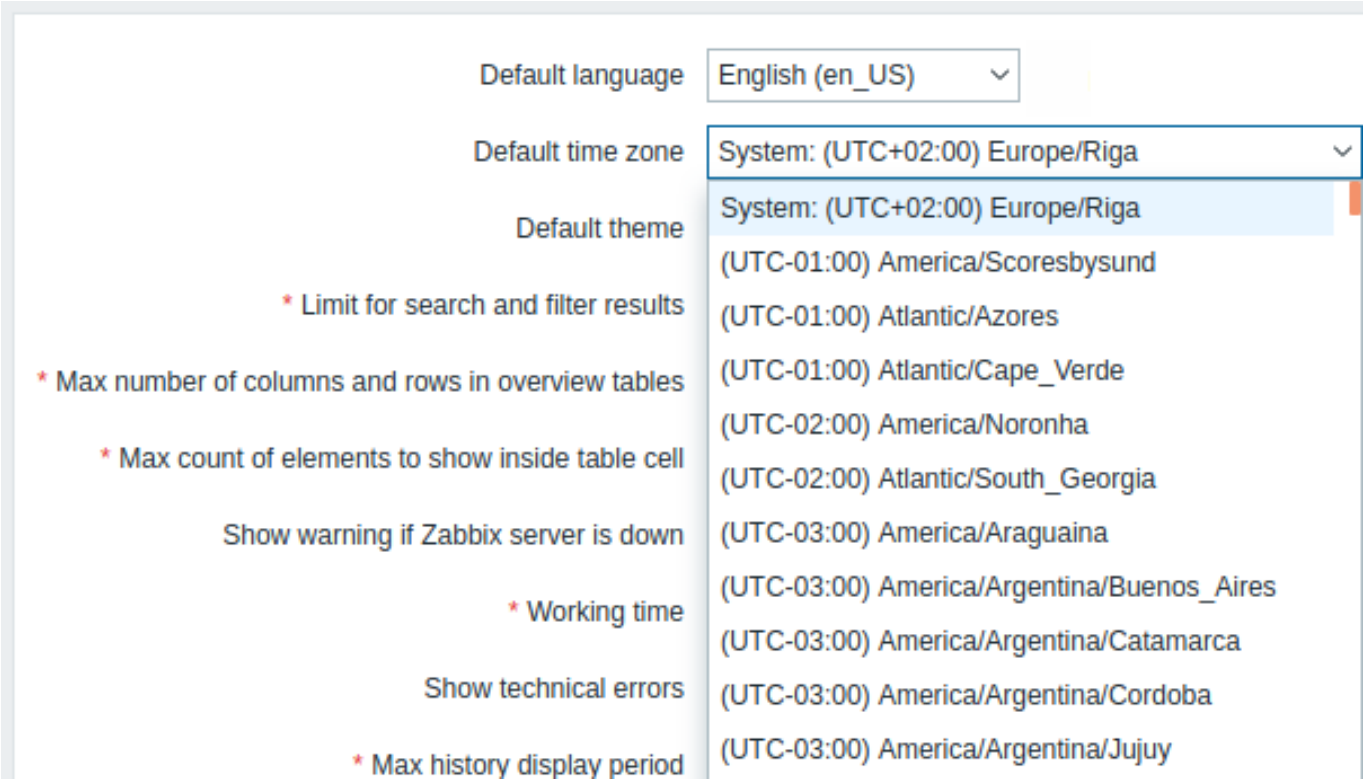
[1] Quando `HttpOnly` é 'true', o cookie será acessível apenas pelo protocolo HTTP. Isso significa que o cookie não estará acessível por linguagens de script, como JavaScript. Essa configuração pode ajudar a reduzir o roubo de identidade por meio de ataques XSS (embora não seja suportada por todos os 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.

Fusos horários

Visão geral

O fuso horário do frontend pode ser definido globalmente no frontend e ajustado para usuários individuais.



Se *Sistema* estiver selecionado, o fuso horário do web server será usado para o frontend (incluindo o valor de `date.timezone` em `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/do usuário especificado somente ao expandir macros nas notificações (por exemplo, `{EVENT.TIME}` pode ser expandida para um fuso horário diferente por usuário) e para o limite de tempo quando as notificações são enviadas (consulte a configuração "Quando ativo" em 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 *Inglês (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 *Administration* > *General* > *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.

Redefinindo a senha

Visão geral Esta seção descreve as etapas para redefinir senhas de usuários no Zabbix.

Etapas Entre em contato com o administrador do Zabbix se você tiver esquecido sua senha do Zabbix e não conseguir fazer login.

Um usuário Super administrador pode alterar as senhas de todos os usuários no formulário de configuração de usuários.

Se um Super administrador tiver esquecido sua senha e não conseguir fazer login, a consulta SQL a seguir deverá ser executada para aplicar a senha padrão ao usuário Super administrador (substitua 'Admin' pelo nome de usuário apropriado do Super administrador):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Após executar essa consulta, a senha do usuário será definida como *zabbix*. Altere a senha padrão no primeiro login.

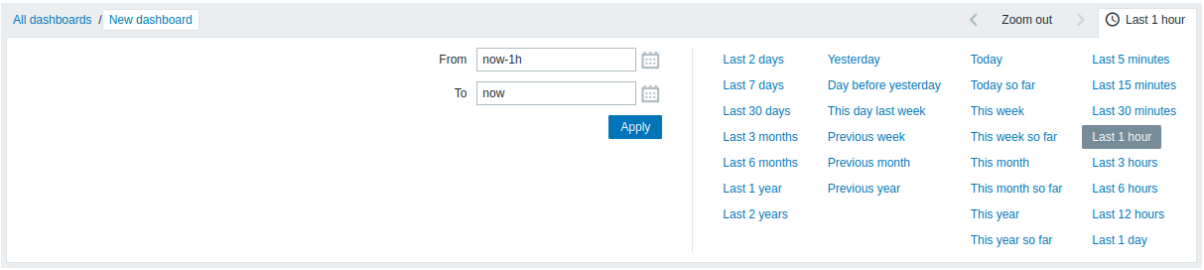
Período de tempo e seletores de host

Visão geral

Esta página descreve os seletores de período 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 único 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 futuros. Opções como *Hoje até o momento*, *Esta semana até o momento* etc. exibem apenas as horas transcorridas.

Depois que um período é selecionado, ele pode ser movido para trás e para frente no tempo clicando nos botões de seta . O botão *Reduzir zoom* permite reduzir o zoom do período em 50% em cada direção.

Note:
Para gráficos (exceto aqueles com o período de tempo *Personalizado*), também é possível selecionar o período de tempo exibido destacando uma área no gráfico com o botão esquerdo do mouse. Depois que você soltar o botão esquerdo do mouse, o gráfico aplicará zoom na área destacada. Também é possível reduzir o zoom clicando duas vezes no gráfico.

Os campos *De/Até* exibem o período selecionado usando a sintaxe de tempo absoluto (no formato `Y-m-d H:i:s`) ou a 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`.

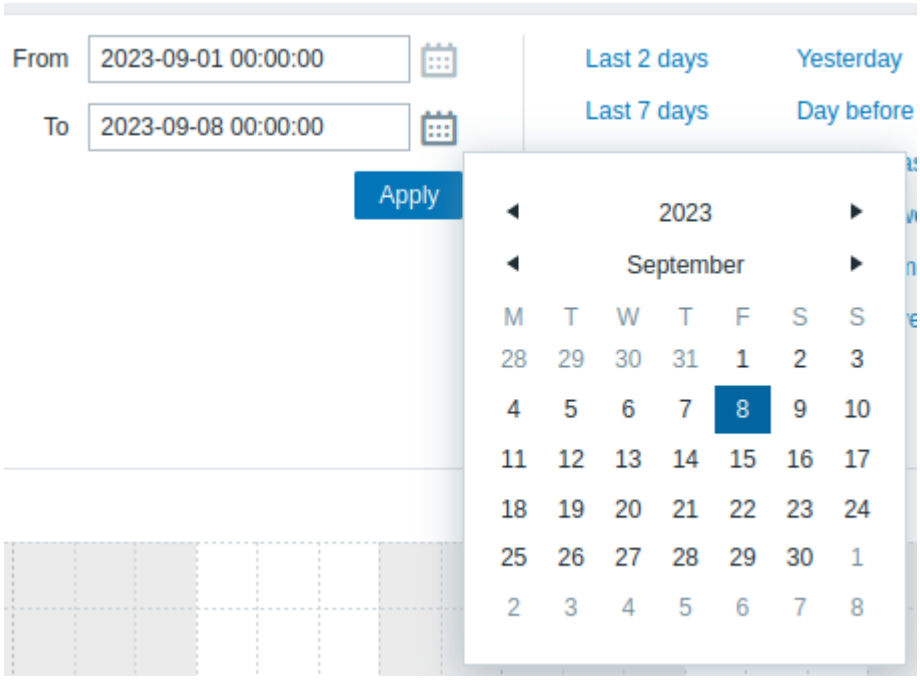
As seguintes abreviações de tempo relativo são compatíveis:

- `now`
- `s` (segundos)
- `m` (minutos)
- `h` (horas)
- `d` (dias)
- `w` (semanas)
- `M` (meses)
- `y` (anos)

A precisão é compatível com o 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:
No seletor de data, você pode navegar entre ano/mês/data usando Tab, Shift+Tab e os botões de seta do teclado. Pressione Enter para confirmar 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 compatíveis do dashboard.

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 uma fonte de dados.



Parâmetros de configuração: frontend do Zabbix

Visão geral Os parâmetros de configuração permitem personalizar o comportamento do Zabbix frontend.

Especifique os valores dos parâmetros no arquivo de configuração do frontend (zabbix.conf.php).

Esta página está estruturada da seguinte forma:

- Visão geral dos parâmetros (para ver todos os detalhes de um parâmetro, clique no nome dele)
- Detalhes dos parâmetros

Parâmetro	Descrição
$DB['TYPE']	Tipo de banco de dados.
$DB['SERVER']	Nome do host ou endereço IP do banco de dados.
$DB['PORT']	Porta do banco de dados.
$DB['DATABASE']	Nome do banco de dados usado pelo Zabbix frontend para se conectar ao banco de dados configurado.
$DB['USER']	Nome de usuário usado pelo Zabbix frontend para se conectar ao banco de dados.
$DB['PASSWORD']	Senha usada pelo Zabbix frontend para se conectar ao banco de dados.
$DB['SCHEMA']	Nome do esquema do banco de dados usado pelo PostgreSQL.
$DB['ENCRYPTION']	Ativa ou desativa a criptografia TLS para conexões entre o Zabbix frontend e o banco de dados.
$DB['KEY_FILE']	Caminho para o arquivo de chave privada do cliente usado na conexão TLS com o banco de dados.
$DB['CERT_FILE']	Caminho para o arquivo de certificado do cliente usado na conexão TLS com o banco de dados.
$DB['CA_FILE']	Caminho para o arquivo de certificado da CA usado na conexão TLS com o banco de dados.
$DB['VERIFY_HOST']	Ativa ou desativa a verificação do nome do host do server do banco de dados no certificado TLS.
$DB['CIPHER_LIST']	Lista de cifras TLS permitidas para a conexão com o banco de dados.
$DB['VAULT']	Especifique o gerenciador de segredos Vault usado para armazenar as credenciais do banco de dados.
$DB['VAULT_URL']	URL do gerenciador de segredos Vault usado para recuperar as credenciais do banco de dados.
$DB['VAULT_PREFIX']	Prefixo do caminho ou da consulta do Vault usado para recuperar as credenciais do banco de dados.
$DB['VAULT_DB_PATH']	Caminho para as credenciais do banco de dados no gerenciador de segredos Vault.
$DB['VAULT_TOKEN']	Token de autenticação usado para acessar o gerenciador de segredos Vault.
$DB['VAULT_APP_ROLE_ID']	ID da função AppRole usado para autenticar no HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	ID secreto do AppRole usado para autenticar no HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Caminho para o arquivo de certificado do cliente usado para autenticar no gerenciador de segredos Vault.
$DB['VAULT_KEY_FILE']	Caminho para o arquivo de chave privada do cliente usado para autenticar no gerenciador de segredos Vault.
$DB['VAULT_CACHE']	Ativa ou desativa o armazenamento em cache das credenciais recuperadas do gerenciador de segredos Vault.
$ZBX_SERVER	Especifique um valor para substituir o nome do host/endereço IP do Zabbix.
$ZBX_SERVER_PORT	Especifique um valor para substituir a porta do Zabbix server.
$ZBX_SERVER_NAME	Nome do Zabbix server.
$IMAGE_FORMAT_DEFAULT	Formato de imagem padrão.
$HISTORY_PROVIDERS[]	Configuração dos provedores de armazenamento de histórico para Elasticsearch ou ClickHouse.
$SSO['SETTINGS']	Configurações para login SAML.
$SSO['CERT_STORAGE']	Especifique onde os certificados SAML são armazenados.
$SSO['SP_KEY']	Caminho para o arquivo de chave privada do provedor de serviços (SP) SAML quando a chave privada está armazenada no sistema de arquivos.
$SSO['SP_CERT']	Caminho para o arquivo de certificado do provedor de serviços (SP) SAML quando o certificado está armazenado no sistema de arquivos.
$SSO['IDP_CERT']	Caminho para o arquivo de certificado do provedor de identidade SAML quando o certificado está armazenado no sistema de arquivos.
$ZBX_FEATURE_FLAGS['banners_enabled']	Ativa ou desativa o suporte a banners.
$ZBX_FEATURE_FLAGS['http_auth_enabled']	Ativa ou desativa a autenticação HTTP de usuários.
$ZBX_FEATURE_FLAGS['modules_configurable']	Ativa ou desativa o acesso aos módulos.
$ZBX_FEATURE_FLAGS['mediatype_predefined']	Desative a edição de tipos de mídia pré-configurados específicos.
$ZBX_SERVER_TLS['ACTIVE']	Especifica se o TLS está ativado para conexões com o Zabbix server.
$ZBX_SERVER_TLS['CA_FILE']	Caminho para o arquivo de certificado da Autoridade de Certificação (CA) usado para verificar o certificado do server.
$ZBX_SERVER_TLS['KEY_FILE']	Caminho para o arquivo de chave privada do cliente correspondente ao certificado do cliente.
$ZBX_SERVER_TLS['CERT_FILE']	Caminho para o arquivo de certificado do cliente quando a autenticação TLS mútua é necessária.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Nome Distinto (DN) do emissor a ser comparado com o certificado do server.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Nome Distinto (DN) do assunto a ser comparado com o certificado do server.
$TELEMETRY_PROVIDERS[]	Configuração de uma fonte de dados de telemetria.

Detalhes dos parâmetros

$DB['TYPE']

Tipo de banco de dados.

Valores: `MYSQL` - MySQL, `POSTGRESQL` - PostgreSQL.

$DB['SERVER']

Nome do host ou endereço IP do banco de dados.

Se `$DB['SERVER']` não estiver definido, o socket de domínio Unix padrão será usado. Para MySQL, definir `$DB['SERVER']` como `localhost` também resulta em uma conexão por meio do socket Unix padrão.
 Para PostgreSQL, `$DB['SERVER']` pode conter várias entradas `host:port` separadas por vírgulas. Se uma porta não estiver definida para um host na lista separada por vírgulas, o valor definido em `$DB['PORT']` será usado como porta padrão. Se um caminho de socket for especificado em `$DB['SERVER']` (por exemplo, `/var/run/pgbouncer`), esse socket de domínio Unix será usado.

Exemplos:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Porta do banco de dados.

Se não for definido, o valor 0 será usado.

$DB['DATABASE']

Nome do banco de dados usado pelo frontend do Zabbix para se conectar ao banco de dados configurado.

Exemplo:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Nome de usuário usado pelo frontend do Zabbix para se conectar ao banco de dados.

$DB['PASSWORD']

Senha usada pelo frontend do Zabbix para se conectar ao banco de dados.

$DB['SCHEMA']

Nome do esquema do banco de dados usado para PostgreSQL.

$DB['ENCRYPTION']

Ative ou desative a criptografia para conexões entre o frontend do Zabbix e o banco de dados.

Valores: `true` - ativada, `false` - desativada.

$DB['KEY_FILE']

Caminho para o arquivo de chave privada do cliente usado na conexão TLS com o banco de dados.

Exemplo:

```
// Para MySQL:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Caminho para o arquivo de certificado do cliente usado na conexão TLS com o banco de dados.

Exemplo:

```
// Para MySQL:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

Caminho para o arquivo de certificado da CA usado na conexão TLS com o banco de dados.

Exemplos:

```
// Para MySQL:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// Para PostgreSQL:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Habilita ou desabilita a verificação do nome do host do banco de dados no certificado TLS.

Valores: `true` - habilitado, `false` - desabilitado.

$DB['CIPHER_LIST']

Lista de cifras TLS permitidas para a conexão com o banco de dados.

Exemplos:

```
// Para MySQL:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Especifique o gerenciador de segredos Vault usado para armazenar credenciais do banco de dados.

Valores: `CyberArk` ou `HashiCorp`.

Exemplos:

```
// Para CyberArk:
$DB['VAULT'] = 'CyberArk';
// Para HashiCorp:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

URL do gerenciador de segredos Vault usado para recuperar as credenciais do banco de dados.

Exemplos:

```
// Para o CyberArk:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// Para o HashiCorp:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Prefixo do caminho ou da consulta do Vault usado para recuperar as credenciais do banco de dados.

Exemplo:

```
// Para o HashiCorp:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Caminho para as credenciais do banco de dados no gerenciador de segredos Vault.

Exemplos:

```
// Para o CyberArk:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
// Para o HashiCorp:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Token de autenticação usado para acessar o gerenciador de segredos Vault.

Exemplo:

```
// Para o HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

ID da função VaultAppSecretID usado para autenticar no HashiCorp Vault.

Exemplo:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID usado para autenticar no HashiCorp Vault.

Exemplo:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Caminho para o arquivo de certificado do cliente usado para autenticar no gerenciador de segredos Vault.

Exemplo:

```
// Para o CyberArk:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Caminho para o arquivo de chave privada do cliente usado para autenticar no gerenciador de segredos Vault.

Exemplo:

```
// Para o CyberArk:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Habilita ou desabilita o armazenamento em cache das credenciais recuperadas do gerenciador de segredos Vault.

Valores: `true` - habilitado, `false` - desabilitado.

$ZBX_SERVER

Especifique um valor para substituir o nome do host/endereço IP do Zabbix.
 Se `$ZBX_SERVER` for especificado sem `$ZBX_SERVER_PORT`, a porta padrão (10051) será usada.

$ZBX_SERVER_PORT

Especifique um valor para substituir a porta do Zabbix server.
 Este parâmetro é usado junto com `$ZBX_SERVER`.

$ZBX_SERVER_NAME

Nome do server do Zabbix.

$IMAGE_FORMAT_DEFAULT

Formato de imagem padrão.

Padrão: `IMAGE_FORMAT_PNG`
 Valores: - `IMAGE_FORMAT_PNG` - formato PNG - `IMAGE_FORMAT_JPEG` - formato JPEG - `IMAGE_FORMAT_GIF` - formato GIF

$HISTORY_PROVIDERS[]

Configuração dos provedores de armazenamento do histórico para Elasticsearch ou ClickHouse.

Parâmetros de configuração compatíveis: - `types` - matriz de tipos de dados a serem armazenados no armazenamento externo; - `provider` - tipo de provedor do histórico: `elasticsearch` ou `clickhouse`; - `url` - URL do provedor do histórico; - `db` - nome do banco de dados (usado pelo ClickHouse); - `username` - usuário do banco de dados (usado pelo ClickHouse); - `password` - senha do banco de dados (usada pelo ClickHouse).

Exemplos:

```
// Para ClickHouse:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Para Elasticsearch:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
```

```
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

Configurações para login SAML. O array `$SSO['SETTINGS']` deve seguir a estrutura esperada pela biblioteca *SAML PHP Toolkit* (fornecida com o Zabbix).

Somente as seguintes opções podem ser especificadas em `$SSO['SETTINGS']`:

- `strict`
- `baseurl`
- `compress`
- `contactPerson`
- `organization`
- `sp` (somente as opções especificadas nesta lista)
  - `attributeConsumingService`
  - `x509certNew`
- `idp` (somente as opções especificadas nesta lista)
  - `singleLogoutService` (somente uma opção)
    * `responseUrl`
  - `certFingerprint`
  - `certFingerprintAlgorithm`
  - `x509certMulti`
- `security` (somente as opções especificadas nesta lista)
  - `signMetadata`
  - `wantNameId`
  - `requestedAuthnContext`
  - `requestedAuthnContextComparison`
  - `wantXMLValidation`
  - `relaxDestinationValidation`
  - `destinationStrictlyMatches`
  - `rejectUnsolicitedResponsesWithInResponseTo`
  - `signatureAlgorithm`
  - `digestAlgorithm`
  - `lowercaseUrlencoding`

Todas as outras opções são obtidas do banco de dados e não podem ser substituídas. A opção `debug` será ignorada.

Se a interface do usuário do Zabbix estiver atrás de um proxy ou balanceador de carga, a opção personalizada `use_proxy_headers` poderá ser usada:

- `false` (padrão) - ignora os cabeçalhos HTTP `X-Forwarded-*`;
- `true` - usa os cabeçalhos HTTP `X-Forwarded-*` para criar a URL base.

Para obter uma descrição completa das opções de configuração disponíveis, consulte a documentação oficial da biblioteca.

Exemplo:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Especifique onde os certificados SAML são armazenados.

Padrão: `file`
 Valores: - `database` - a chave privada e os certificados são armazenados no banco de dados; - `file` - a chave privada e os certificados são armazenados no sistema de arquivos.

Exemplo:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Caminho para o arquivo de chave privada do provedor de serviço SAML (SP) quando a chave privada está armazenada no sistema de arquivos.

Exemplo:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Caminho para o arquivo de certificado do provedor de serviço (SP) SAML quando o certificado está armazenado no sistema de arquivos.

Exemplo:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Caminho para o arquivo de certificado do provedor de identidade SAML quando o certificado está armazenado no sistema de arquivos.

Exemplo:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Ative ou desative o suporte a banners.

Padrão: `true`
 Valores: `true` - ativado, `false` - desativado.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Habilita ou desabilita a autenticação HTTP do usuário.

Padrão: `true`
 Valores: `true` - habilitado, `false` - desabilitado.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Habilita ou desabilita o acesso aos módulos.

Padrão: `true`
 Valores: `true` - habilitado, `false` - desabilitado.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Desabilita a edição de tipos de mídia pré-configurados específicos.

Valores: `email`, `script`, `sms`, `webhook`, `push`.

Exemplo:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Especifica se uma conexão TLS entre o frontend do Zabbix e o server do Zabbix está habilitada.

Padrão: `false`
 Valores: `true` - habilitada, `false` - desabilitada.

$ZBX_SERVER_TLS['CA_FILE']

Caminho para o arquivo de certificado da Autoridade de Certificação (CA) usado para verificar o certificado do server.

Exemplo:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Caminho para o arquivo de chave privada do cliente correspondente ao certificado do cliente usado para autenticação TLS mútua.

Exemplo:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Caminho para o arquivo de certificado do cliente quando a autenticação TLS mútua é necessária.

Exemplo:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Nome distinto (DN) do emissor a ser comparado com o certificado do server.

Exemplo:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Nome distinto (DN) do sujeito a ser comparado com o certificado do server.

Exemplo:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Configuração de uma fonte de dados de telemetria.

Parâmetros de configuração compatíveis com o ClickHouse: - `url` - define a URL HTTP ou HTTPS usada para conectar-se à fonte de dados de telemetria. - `db` - define o nome do banco de dados. - `username` - define o nome de usuário usado para fazer login no banco de dados. - `password` - define a senha usada para fazer login no banco de dados. - `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_ca_file` - define essas opções para configurar uma conexão segura (TLS/SSL) com o banco de dados. - `vault_path` - define o caminho para um segredo do Vault que contém os valores de `username` e `password`. Não use junto com as opções `username` e `password`.

Exemplo:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Visão geral A API do Zabbix permite recuperar e modificar programaticamente a configuração do Zabbix e fornece acesso a dados históricos. Ela é amplamente usada 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 web frontend. Ela usa 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 o JSON, consulte a especificação JSON-RPC 2.0 e a página inicial do formato JSON.

Para mais informações sobre a integração da funcionalidade do Zabbix em seus aplicativos Python, consulte Biblioteca Python para Zabbix.

> **Note:**
> O acesso do usuário no Zabbix, incluindo tanto a configuração quanto os dados históricos, depende do tipo de usuário, da função de usuário atribuída e dos grupos de usuários.

Estrutura A API consiste em vários métodos que, nominalmente, são agrupados em APIs separadas. Cada um dos métodos executa uma tarefa específica. Por exemplo, o método `host.create` pertence à API de *host* e é usado para criar novos hosts. Historicamente, as APIs às vezes são chamadas de "classes".

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.

Fazendo requisições Depois de configurar o frontend, você pode usar requisições HTTP remotas para chamar a API. Para isso, você precisa enviar requisições HTTP POST para o arquivo `api_jsonrpc.php` localizado no diretório do frontend.
 Por exemplo, se o seu frontend do Zabbix estiver instalado em `https://example.com/zabbix`, uma 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 como 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 Zabbix API implementa a versão 2.0 do JSON-RPC);
- `method` - o método da API que está sendo chamado;
- `params` - os parâmetros que serão passados para o método da API;
- `id` - um identificador arbitrário da 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.

Autenticação Para acessar quaisquer dados no Zabbix, você precisa:

- Usar um token de API existente criado no frontend do Zabbix, seção *Users* > *API tokens*, ou criado usando a Token API.
- Usar um token de autenticação obtido com o método `user.login`.

Por exemplo, se você quisesse obter um novo token de autenticação ao fazer login como um usuário padrão *Admin*, uma 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 fornecer as credenciais corretamente, a resposta retornada pela API deverá 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).

Fluxo de trabalho de exemplo A seção a seguir orienta você por vários exemplos de uso com mais detalhes.

Recuperando hosts Agora você tem um token de autenticação de usuário válido (representado como uma variável nos exemplos a seguir) que pode ser usado para acessar os dados no Zabbix. Por exemplo, você pode usar o método `host.get` para recuperar os IDs, nomes dos hosts e interfaces de todos os hosts configurados:

Request:

```
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 motivos 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 que você obteve da solicitação `host.get` anterior. Isso pode ser feito usando o método `item.create`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Uma resposta bem-sucedida conterá o ID do item recém-criado, que pode ser usado para referenciar o item nas solicitações a seguir:

```
{
    "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 vários items com uma única chamada da 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 Ative 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 items com uma única chamada da API.

Atualizando múltiplos triggers Ative múltiplos triggers definindo seus 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
}
```

Tratamento de erros Até o momento, tudo o que você tentou funcionou corretamente. 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 ocorrer um erro, 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 ao usar valores de entrada incorretos, ocorrer timeout de sessão ou tentar acessar objetos inexistentes. Seu aplicativo deve ser capaz de lidar com esses tipos de erro de forma adequada.

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 você está trabalhando. Isso pode ser útil para ajustar sua aplicação para usar recursos específicos de determinada versão.

O Zabbix garante compatibilidade retroativa dos recursos dentro de uma versão principal. Ao fazer alterações incompatíveis com versões anteriores entre lançamentos principais, o Zabbix normalmente deixa os recursos antigos como obsoletos no próximo lançamento e só os remove no lançamento seguinte. Ocasionalmente, o Zabbix pode remover recursos entre lançamentos principais sem fornecer qualquer compatibilidade retroativa. É importante que você nunca dependa de recursos obsoletos e migre para alternativas mais novas o quanto antes.

> **Note:**
> Você pode acompanhar todas as alterações feitas na API no changelog da API.

Leitura adicional Agora, você já tem conhecimento suficiente para começar a trabalhar com a Zabbix API; no entanto, não pare por aqui. Para leitura adicional, recomendamos que você consulte a 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

APM A API do Zabbix permite acessar dados de telemetria coletados para o monitoramento do desempenho de aplicações (APM).

Traces

Recupere os traces e os spans que os compõem.

API de traces de APM | API de spans de APM

Métricas

Recupere pontos de dados de métricas.

API de métricas de APM

Logs

Recupere registros de log.

API de log do APM

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 o monitoramento do nível de serviço e obtenha informações detalhadas de SLA sobre qualquer serviço.

API de serviços

Coleta de dados A API do Zabbix permite gerenciar a configuração do seu sistema de monitoramento.

Hosts e grupos de hosts

Gerencie grupos de hosts, hosts e tudo o que estiver relacionado a eles, incluindo interfaces de host, macros de host e períodos de manutenção.

API de host | API de grupo de hosts | API de macro de usuário | API de mapa de valores | API de manutenção

Items

Defina os items a serem monitorados.

API de item

Triggers

Configure triggers para notificá-lo sobre problemas em seu sistema. Gerencie as dependências dos triggers.

Trigger API

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

Dispositivos

Gerencie dispositivos móveis para uso com o aplicativo móvel do Zabbix.

API de dispositivos

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 do Zabbix Recupere a versão da API do Zabbix ou o horário da última atualização do Zabbix server.

API info API | Server info API

Action

Esta classe foi projetada para trabalhar com ações.

Referências de objetos:

- Action
- Action operation
  - Action operation message
  - Action operation condition
- Action recovery operation
- Action update operation
- Action filter
  - Action filter condition

Métodos disponíveis:

- action.create - criar novas ações
- action.delete - excluir ações
- action.get - recuperar ações
- action.update - atualizar ações

Objeto action

Os seguintes objetos estão diretamente relacionados à API `action`.

Ação

O objeto de ação tem as seguintes propriedades.

Propriedade	Tipo	Descrição
actionid	ID	ID da ação. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
esc_period	string	Duração padrão da etapa da operação. Deve ser de pelo menos 60 segundos. Aceita um número de segundos (por exemplo, 60, 300), um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`) ou uma macro de usuário. Comportamento da propriedade: - *suportada* se `eventsource` estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”

Propriedade	Tipo	Descrição
eventsource	integer	Tipo de eventos que a ação tratará. Consulte a propriedade `source` do evento para obter uma lista dos tipos de eventos compatíveis. Comportamento da propriedade: - *constante* - *obrigatória* para operações de criação
name	string	Nome da ação. Comportamento da propriedade: - *obrigatória* para operações de criação
status	integer	Indica se a ação está habilitada ou desabilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
pause_symptoms	integer	Indica se a escalação deve ser pausada quando o evento for um evento de sintoma. Valores possíveis: 0 - Não pausar a escalação para problemas de sintoma; 1 - *(padrão)* Pausar a escalação para problemas de sintoma. Comportamento da propriedade: - *suportada* se `eventsource` estiver definido como ”evento criado por um trigger”
pause_suppressed	integer	Indica se a escalação deve ser pausada durante períodos de manutenção. Valores possíveis: 0 - Não pausar a escalação; 1 - *(padrão)* Pausar a escalação. Comportamento da propriedade: - *suportada* se `eventsource` estiver definido como ”evento criado por um trigger”
notify_if_canceled	integer	Indica se uma notificação deve ser enviada quando a escalação for cancelada. Valores possíveis: 0 - Não enviar uma notificação quando a escalação for cancelada; 1 - *(padrão)* Enviar uma notificação quando a escalação for cancelada. Comportamento da propriedade: - *suportada* se `eventsource` estiver definido como ”evento criado por um trigger”

Operação da ação

O objeto de operação da ação define uma operação que será realizada quando uma ação for executada. Ele tem 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 o modo de inventário do host; 13 - adicionar tags ao host; 14 - remover tags do 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 uma etapa de escalonamento em segundos. Deve ser superior a 60 segundos. Aceita um número de segundos (por exemplo, 60, 300), um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`) 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: - *compatível* 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	Etapa a partir da qual o escalonamento deve começar. Padrão: 1. Comportamento da propriedade: - *compatível* 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	Etapa na qual o escalonamento deve terminar. Padrão: 1. Comportamento da propriedade: - *compatível* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação das condições da operação. Valores possíveis: 0 - *(padrão)* And/Or; 1 - And; 2 - Or.
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 nos quais os scripts globais serão executados. 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 no qual os scripts globais serão executados. 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 da operação usadas para ações de trigger. O objeto de condição da operação é descrito em detalhes abaixo.
opgroup	array	Grupos de hosts aos quais os hosts serão adicionados. 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 que contém os dados sobre a mensagem enviada pela operação. O objeto de mensagem da operação é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem”
opmessage_grp	array	Grupos de usuários para os quais as mensagens serão enviadas. 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 os quais as mensagens serão enviadas. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_grp` não estiver definido

Propriedade	Tipo	Descrição
optemplate	array	Templates a serem vinculados aos hosts. Os templates devem ter a propriedade `templateid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”vincular template” ou ”desvincular template”
opinventory	object	Modo de inventário a ser definido no host. O inventário deve ter a propriedade `inventory_mode` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”definir o modo de inventário do host”
optag	array	Tags do host a serem adicionadas ou removidas. As tags devem ter a propriedade `tag` definida. A propriedade `value` é opcional. Comportamento da propriedade: - *compatível* se `operationtype` estiver definido como ”adicionar tags ao host” ou ”remover tags do 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. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
conditiontype	integer	Tipo de condição. Valores possíveis: 14 - evento reconhecido. Comportamento da propriedade: - *obrigatório*
value	string	Valor para comparar. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição. Valores possíveis: 0 - *(padrão)* =

Os seguintes operadores e valores são suportados para cada tipo de condição de operação.

Condição	Nome da condição	Operadores suportados	Valor esperado
14	Evento reconhecido	=	Se o evento foi reconhecido. Valores possíveis: 0 - não reconhecido; 1 - reconhecido.

Operação de recuperação de ação

O objeto de operação de recuperação de ação define uma operação que será executada quando um problema for resolvido. As operações de recuperação são possíveis **apenas** para ações de trigger, internas e de serviço. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
operationtype	integer	Tipo de operação. Valores possíveis se o `eventsource` do objeto Action estiver definido como ”evento criado por um trigger” ou ”evento criado na atualização do status do serviço”: 0 - enviar mensagem; 1 - script global; 11 - notificar todos os envolvidos. Valores possíveis se o `eventsource` do objeto Action estiver definido como ”evento interno”: 0 - enviar mensagem; 11 - notificar todos os envolvidos. Comportamento da propriedade: - *obrigatório*
opcommand	object	Script global a ser executado. O script global deve ter a propriedade `scriptid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global”

Propriedade	Tipo	Descrição
opcommand_grp	array	Grupos de hosts para executar scripts globais. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se o `eventsource` do objeto Action estiver definido como "evento criado por um trigger", e `operationtype` estiver definido como "script global", e `opcommand_hst` não estiver definido
opcommand_hst	array	Host para executar scripts globais. Os hosts devem ter a propriedade `hostid` definida. Comportamento da propriedade: - *obrigatório* se o `eventsource` do objeto Action estiver definido como "evento criado por um trigger", e `operationtype` estiver definido como "script global", e `opcommand_grp` não estiver definido
opmessage	object	Objeto contendo os dados sobre a mensagem enviada pela operação de recuperação. O objeto de mensagem de operação é descrito em detalhes acima. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem"
opmessage_grp	array	Grupos de usuários para enviar mensagens. Os grupos de usuários devem ter a propriedade `usrgrpid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem" e `opmessage_usr` não estiver definido
opmessage_usr	array	Usuários para enviar mensagens. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem" e `opmessage_grp` não estiver definido

Operação de atualização de ação

O objeto de operação de atualização de ação define uma operação que será executada quando um problema for atualizado (comentado, reconhecido, severidade alterada ou fechado manualmente). As operações de atualização são possíveis **apenas** para ações de trigger e de serviço. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
operationtype	integer	Tipo de operação. Valores possíveis: 0 - enviar mensagem; 1 - script global; 12 - notificar todos os envolvidos. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
opcommand	object	Script global a ser executado. O script global deve ter a propriedade `scriptid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global”
opcommand_grp	array	Grupos de hosts para executar scripts globais. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, e `operationtype` estiver definido como ”script global”, e `opcommand_hst` não estiver definido
opcommand_hst	array	Host para executar scripts globais. Os hosts devem ter a propriedade `hostid` definida. Comportamento da propriedade: - *obrigatório* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, e `operationtype` estiver definido como ”script global”, e `opcommand_grp` não estiver definido
opmessage	object	Objeto contendo os dados sobre a mensagem enviada pela operação de atualização. O objeto de mensagem de operação é descrito em detalhes acima.
opmessage_grp	array	Grupos de usuários para enviar mensagens. Os grupos de usuários devem ter a propriedade `usrgrpid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_usr` não estiver definido
opmessage_usr	array	Usuários para enviar mensagens. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_grp` não estiver definido

Filtro de ação

O objeto de filtro de ação define um conjunto de condições que devem ser atendidas para executar as operações de ação configuradas. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
conditions	array	Conjunto de condições de filtro a serem usadas para filtrar resultados. As condições serão classificadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação da condição do filtro. Valores possíveis: 0 - E/OU; 1 - E; 2 - OU; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que fazem referência a condições de filtro específicas por seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições de filtros com uma expressão personalizada. A expressão deve conter IDs que fazem referência a condições de filtro específicas por seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição do filtro da ação

O objeto de condição do filtro da ação define uma condição específica que deve ser verificada antes da execução das 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 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 descoberto; 29 - grupo de proxies. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado pelo registro automático de agent ativo”: 20 - proxy; 22 - nome do host; 24 - metadados do host; 29 - grupo de proxies. 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 a ser comparado. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
value2	string	Valor secundário a ser comparado. 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 exclusivo arbitrário usado para referenciar a condição em uma expressão personalizada. Pode conter apenas 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)* é igual a; 1 - não é igual a; 2 - contém; 3 - não contém; 4 - está em; 5 - é maior ou igual a; 6 - é menor ou igual a; 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, consulte os exemplos nas páginas dos métodos action.get e action.create.

Os operadores e valores a seguir são compatíveis com cada tipo de condição.

Condição	Nome da condição	Operadores compatíveis	Valor esperado
0	Grupo de hosts	é igual a, não é igual a	ID do grupo de hosts.
1	Host	é igual a, não é igual a	ID do host.
2	Trigger	é igual a, não é igual a	ID do trigger.
3	Nome do evento	contém, não contém	Nome do evento.
4	Severidade do trigger	é igual a, não é igual a, é maior ou igual a, é menor ou igual a	Severidade do trigger. Consulte a propriedade `severity` do trigger para obter uma lista das severidades de trigger compatíveis.
5	Valor do trigger	é igual a	Valor do trigger. Consulte a propriedade `value` do trigger para obter uma lista dos valores de trigger compatíveis.
6	Período de tempo	está em, não está em	Momento em que o evento foi acionado, como um período de tempo.

Condição	Nome da condição	Operadores compatíveis	Valor esperado
7	IP do host	é igual a, não é igual a	Um ou vários intervalos de IP a serem verificados, separados por vírgulas. Consulte a seção configuração da descoberta de rede para obter mais informações sobre os formatos compatíveis de intervalos de IP.
8	Tipo de serviço descoberto	é igual a, não é igual a	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 obter uma lista dos tipos compatíveis.
9	Porta do serviço descoberto	é igual a, não é igual a	Um ou vários intervalos de portas, separados por vírgulas.
10	Status da descoberta	é igual a	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	é maior ou igual a, é menor ou igual a	Tempo que indica há quantos segundos o objeto descoberto está no status atual.
12	Valores recebidos	é igual a, não é igual a, é maior ou igual a, é menor ou igual a, contém, não contém	Valor retornado ao executar uma verificação de descoberta de Zabbix agent, SNMPv1, SNMPv2 ou SNMPv3.
13	Template de host	é igual a, não é igual a	ID do template vinculado.
16	Problema suprimido	Sim, Não	Nenhum valor é necessário: usar o operador ”Sim” significa que o problema deve ser suprimido; ”Não” significa que não deve ser suprimido.
18	Regra de descoberta	é igual a, não é igual a	ID da regra de descoberta.
19	Verificação de descoberta	é igual a, não é igual a	ID da verificação de descoberta.
20	Proxy	é igual a, não é igual a	ID do proxy.
21	Objeto descoberto	é igual a	Tipo de objeto que acionou 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 compatível com os operadores *corresponde* e *não corresponde* nas condições de registro automático.
23	Tipo de evento	é igual a	Evento interno específico. Valores possíveis: 0 - item no estado ”não suportado”; 1 - item no estado ”normal”; 2 - regra LLD no estado ”não suportado”; 3 - regra LLD no estado ”normal”; 4 - trigger no estado ”desconhecido”; 5 - trigger no estado ”normal”.
24	Metadados do host	contém, não contém, corresponde, não corresponde	Metadados do host registrado automaticamente. O uso de uma expressão regular é compatível com os operadores *corresponde* e *não corresponde*.

Condição	Nome da condição	Operadores compatíveis	Valor esperado
25	Tag	é igual a, não é igual a, contém, não contém	Tag do evento.
26	Valor da tag	é igual a, não é igual a, contém, não contém	Valor da tag do evento.
27	Serviço	é igual a, não é igual a	ID do serviço.
28	Nome do serviço	é igual a, não é igual a	Nome do serviço.
29	Grupo de proxies	é igual a, não é igual a	ID do grupo de proxies.

action.create

Descrição

`object action.create(object/array actions)`

Este método permite criar novas actions.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter 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 da função de usuário. Consulte Funções de usuário para obter mais informações.
 O usuário não poderá excluir uma ação que contenha um proxy ou grupo de proxies inacessível em suas condições de filtro. Consulte configuração do grupo de usuários (*Lista de acesso a proxy*) para obter 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 actions 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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
actionids	ID/array	Retornar apenas ações com os IDs fornecidos.
groupids	ID/array	Retornar apenas ações que usam os grupos de hosts fornecidos nas condições da ação.
hostids	ID/array	Retornar apenas ações que usam os hosts fornecidos nas condições da ação.
triggerids	ID/array	Retornar apenas ações que usam os triggers fornecidos nas condições da ação.
mediatypeids	ID/array	Retornar apenas ações que usam os tipos de mídia fornecidos para enviar mensagens.
usrgrpids	ID/array	Retornar apenas ações que estão configuradas para enviar mensagens para os grupos de usuários fornecidos.
userids	ID/array	Retornar apenas ações que estão configuradas para enviar mensagens para os usuários fornecidos.
scriptids	ID/array	Retornar apenas ações que estão configuradas para executar os scripts fornecidos.
selectFilter	query	Retornar uma propriedade `filter` com o filtro de condição da ação.
selectOperations	query	Retornar uma propriedade `operations` com as operações da ação.
selectRecoveryOperations	query	Retornar uma propriedade `recovery_operations` com as operações de recuperação da ação.
selectUpdateOperations	query	Retornar uma propriedade `update_operations` com as operações de atualização da ação.
sortfield	string/array	Ordenar 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

Recupera todas as ações de trigger configuradas juntamente com as condições e operações da ação.

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 state: {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 actions existentes.

> **Note:**
> Este método está disponível apenas para 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 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.

Parâmetro	Tipo	Descrição
recovery_operations	array	Operações de recuperação de ação para substituir as operações de recuperação existentes. Comportamento do parâmetro: - *suportado* se `eventsource` do Objeto de ação estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”
update_operations	array	Operações de atualização de ação para substituir as operações de atualização existentes. Comportamento do parâmetro: - *suportado* se `eventsource` do Objeto de ação 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 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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 objetos 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 eventos 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 acionaram 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	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parâmetro	Tipo	Descrição
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

Recupere todos os alertas gerados pela ação "3".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

Veja também

- Host
- Tipo de mídia
- Usuário

Fonte

CAlert::get() em *ui/include/classes/api/services/CAlert.php*.

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 tem as seguintes propriedades.

Property	Type	Description
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 `$ZBX_FEATURE_FLAGS['http_auth_enabled']` estiver definido como `true` no arquivo de configuração do frontend.
http_login_form	integer	Formulário de login padrão. Valores possíveis: 0 - *(padrão)* Formulário de login do Zabbix; 1 - Formulário de login HTTP. Comportamento da propriedade: - *suportada* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` estiver definido como `true` no arquivo de configuração do frontend.
http_strip_domains	string	Nome do domínio a remover. Comportamento da propriedade: - *suportada* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` estiver definido como `true` no arquivo de configuração do frontend.
http_case_sensitive	integer	Login HTTP com distinção entre maiúsculas e minúsculas. Valores possíveis: 0 - Desativado; 1 - *(padrão)* Ativado. Comportamento da propriedade: - *suportada* se `$ZBX_FEATURE_FLAGS['http_auth_enabled']` estiver definido como `true` no arquivo de configuração do frontend.
ldap_auth_enabled	integer	Autenticação LDAP. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada.

Property	Type	Description
ldap_case_sensitive	integer	Login LDAP com distinção entre maiúsculas e minúsculas. Valores possíveis: 0 - Desativado; 1 - *(padrão)* Ativado.
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: - *obrigatória* 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 distinção entre maiúsculas e minúsculas. Valores possíveis: 0 - Desativado; 1 - *(padrão)* Ativado.
passwd_min_length	integer	Requisito de comprimento mínimo da senha. O intervalo de valores possíveis é de 1 a 70. Padrão: 8.
passwd_check_rules	integer	Regras de verificação de senha. Valores possíveis do bitmap: 0 - Verificar o 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 usadas com frequência e não contém derivações da palavra ”Zabbix”, do nome, sobrenome ou nome de usuário do usuário. Este é um campo de máscara de bits; qualquer soma dos valores possíveis do bitmap é aceita (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 conectado. Aceita um número de segundos (por exemplo, 3600) ou um valor com um sufixo de tempo (por exemplo, `3600s`, `60m`, `1h`, `1d`, `1M`, `1y`), incluindo suporte para meses e anos. Valor mínimo: `1h`. Padrão: 1h. Disponível somente para provisionamento LDAP.

Property	Type	Description
disabled_usrgrpid	ID	ID do grupo de usuários ao qual atribuir o usuário cujo provisionamento foi desfeito. O grupo de usuários deve estar desabilitado e não poderá ser habilitado ou excluído enquanto estiver configurado. Comportamento da propriedade: - *obrigatória* se `ldap_jit_status` estiver definido como "Habilitado para IdPs LDAP configurados" ou se `saml_jit_status` estiver definido como "Habilitado para IdPs SAML configurados"
mfa_status	integer	Autenticação multifator. Valores possíveis: 0 - Desabilitada (para todos os métodos MFA configurados); 1 - Habilitada (para todos os métodos MFA configurados).
mfaid	ID	Método MFA padrão para grupos de usuários com MFA habilitada. Comportamento da propriedade: - *obrigatória* 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 informados.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(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 de usuário. Consulte User roles para obter 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: - *apenas escrita*
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *apenas escrita*

autoregistration.get

Descrição

`object autoregistration.get(object parameters)`

O método permite recuperar o objeto de autoregistration 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 User roles para obter 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 uma autoregistration 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. Consulte Funções de usuário para obter 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 de 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 tem as seguintes propriedades.

Property	Type	Description
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

Property	Type	Description
authentication	integer	Método de autenticação que será usado pelo cenário web. Valores possíveis: 0 - *(padrão)* nenhum; 1 - autenticação HTTP básica; 2 - autenticação NTLM; 3 - autenticação Kerberos; 4 - autenticação Digest.
delay	string	Intervalo de execução do cenário web. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou uma macro de usuário. Padrão: 1m.
headers	array	Cabeçalhos HTTP que serão enviados ao realizar uma solicitação.
http_password	string	Senha usada para autenticação HTTP básica, NTLM, Kerberos ou Digest.
http_proxy	string	Proxy que será usado pelo cenário web no formato *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Nome de usuário usado para autenticação HTTP básica, NTLM, Kerberos ou Digest.
retries	integer	Número de vezes que um cenário web tentará executar cada etapa antes de falhar. Padrão: 1.
ssl_cert_file	string	Nome do arquivo de certificado SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_file	string	Nome do arquivo de chave privada SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_password	string	Senha da chave privada SSL.
status	integer	Indica 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	Indica se o nome do host da conexão deve ser validado para corresponder ao nome no certificado do host. Valores possíveis: 0 - *(padrão)* ignorar a verificação do host; 1 - verificar o host.
verify_peer	integer	Indica se o certificado do host deve ser validado quanto à autenticidade. Valores possíveis: 0 - *(padrão)* ignorar a verificação do peer; 1 - verificar o peer.
uuid	string	Identificador exclusivo global, usado para vincular cenários web importados aos já existentes. Gerado automaticamente se não for informado. Comportamento da propriedade: - *suportado* se o cenário web pertencer a um template

Etapa do cenário

O objeto de etapa do cenário define uma verificação específica do cenário web. Ele tem 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*
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 solicitação. Os cabeçalhos da etapa do cenário substituirão os cabeçalhos especificados para o cenário web.
posts	string/array	Variáveis HTTP POST como uma string (dados brutos da solicitação) ou como uma matriz de campos HTTP (dados dos campos do 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)* somente o corpo; 1 - somente os cabeçalhos; 2 - cabeçalhos e corpo.
status_codes	string	Intervalos de códigos de status HTTP obrigatórios, separados por vírgulas.
timeout	string	Tempo limite da solicitação em segundos. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 1h) ou uma macro de usuário. Padrão: 15s. Máximo: 1h. Mínimo: 1s.
variables	array	Variáveis da etapa do cenário.
query_fields	array	Campos de consulta - matriz de campos HTTP que serão adicionados à URL ao realizar uma solicitação.

Campo HTTP

O objeto de campo HTTP define o nome e o valor que são usados para especificar as variáveis do cenário web, cabeçalhos HTTP e campos POST ou campos de consulta. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho/variável/campo POST ou GET. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho/variável/campo POST ou GET. Comportamento da propriedade: - *obrigatório*

Tag de cenário web

O objeto de tag de cenário web possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do cenário web. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do cenário web. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template; 1 - host. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

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. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Cenários web para criar.

Além das propriedades padrão do cenário web, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
steps	array	Etapas do cenário. Comportamento do parâmetro: - *obrigatório*
tags	array	Tags do cenário web.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos cenários 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 de usuário. Consulte Funções de usuário para obter 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 informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas cenários web que pertencem aos grupos de hosts 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.
inheritedTags	boolean	Retorna cenários web que possuem as `tags` fornecidas também no template/host/templates vinculados. Valores possíveis: `true` - template/host/templates vinculados também devem ter as tags fornecidas; `false` - *(padrão)* tags de template/host/templates vinculados são ignoradas.
monitored	boolean	Se definido como `true`, retorna apenas cenários web habilitados que pertencem a hosts monitorados.

Parâmetro	Tipo	Descrição
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags que estão no template/host/templates vinculados.
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	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	array	Retorna apenas cenários web com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os cenários web. Valores possíveis para o operador: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
selectHosts	query	Retorna os hosts aos quais o cenário web pertence como um array na propriedade `hosts`.
selectSteps	query	Retorna 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 estão descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando um cenário web

Recupere todos os dados sobre o cenário web "9".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
```

```
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "uuid": "",
            "headers": [],
            "variables": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
                    "no": "2",
                    "url": "http://example.com/about",
```

```
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem os objetos a serem exportados e o formato a ser usado.

Parâmetro	Tipo	Descrição
format	string	Formato no qual os dados devem ser exportados. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - array PHP não processado. Comportamento do parâmetro: - *obrigatório*

Parâmetro	Tipo	Descrição
prettyprint	boolean	Torna a saída mais legível para humanos, adicionando recuo. Valores possíveis: `true` - adiciona recuo; `false` - *(padrão)* não adiciona recuo.
options	object	Objetos a serem exportados. O objeto `options` tem os seguintes parâmetros: `host_groups` - `(array)` IDs dos grupos de hosts a exportar; `hosts` - `(array)` IDs de hosts a exportar; `images` - `(array)` IDs de imagens a exportar; `maps` - `(array)` IDs de mapas a exportar; `mediaTypes` - `(array)` IDs de tipos de mídia a exportar; `template_groups` - `(array)` IDs de grupos de templates a exportar; `templates` - `(array)` IDs de templates a exportar; `dashboards` - `(array)` IDs de dashboards a exportar; `global_regexes` - `(array)` IDs de expressões regulares globais a exportar. Usuários do tipo *Admin* e *User* podem exportar apenas os objetos para os quais têm 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 de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que contêm os dados a serem importados e as regras de como os dados devem ser tratados.

Parameter	Type	Description
format	string	Formato da string serializada. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON. Comportamento do parâmetro: - *required*
source	string	String serializada contendo os dados de configuração. Comportamento do parâmetro: - *required*
rules	object	Regras sobre como novos objetos e objetos existentes devem ser importados. Usuários do tipo *Admin* podem importar apenas os objetos para os quais têm permissão de leitura e gravação, bem como mapas. Por exemplo, um host e suas entidades (items, triggers, graphs, etc.) podem ser importados somente 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: - *required*
returnMissingObjects	boolean	Regras para saber se o objeto de resposta deve conter informações sobre objetos ausentes; padrão: `false`.

Attention:
Se nenhuma regra for informada, a configuração não será atualizada.

O objeto `rules` oferece suporte aos seguintes parâmetros.

Parameter	Type	Description
dashboards	object	Regras sobre como importar dashboards. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos dashboards serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true` e um dashboard importável específico estiver presente no sistema, ele será totalmente atualizado. As páginas e os widgets serão totalmente substituídos pelos novos; padrão: `false`.
discoveryRules	object	Regras sobre como importar regras de LLD. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas regras de LLD serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, regras de LLD existentes serão atualizadas; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, regras de LLD que não estiverem presentes nos dados importados serão excluídas do banco de dados; padrão: `false`.
global_regexes	object	Regras sobre como importar expressões regulares globais. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas expressões regulares globais serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, expressões regulares globais existentes serão atualizadas, com suas subexpressões totalmente substituídas; padrão: `false`.
graphs	object	Regras sobre como importar graphs. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos graphs serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, graphs existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, graphs que não estiverem 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`.
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`.

Parameter	Type	Description
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 que não estiverem presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
images	object	Regras sobre como importar imagens. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas imagens serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, imagens existentes serão atualizadas; padrão: `false`.
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 que não estiverem presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
maps	object	Regras sobre como importar maps. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos maps serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, maps 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`.
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 que não estiverem presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
template_groups	object	Regras sobre como importar grupos de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de template serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de template existentes serão atualizados; padrão: `false`.

Parameter	Type	Description
templateLinkage	object	Regras sobre como importar links de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, templates que não estiverem vinculados ao host ou template sendo importado, mas que estiverem presentes nos dados importados, serão vinculados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, templates que estiverem vinculados ao host ou template sendo importado, mas que não estiverem 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`.
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 que não estiverem presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
valueMaps	object	Regras sobre como importar value maps de host ou template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos value maps serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, value maps existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, value maps que não estiverem 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 esse 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros contendo os possíveis dados a serem importados e as regras de como os dados devem ser tratados.

Parameter	Type	Description
format	string	Formato da string serializada. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	String serializada contendo os dados de configuração. Parameter behavior: - *required*

Parameter	Type	Description
rules	object	Regras sobre como os objetos novos e existentes devem ser comparados. Usuários do tipo *Admin* e *User* podem comparar objetos novos apenas com os existentes para os quais tenham permissão de *read-only* ou *read-write* permission. O parâmetro `rules` é descrito em detalhes na tabela abaixo. Parameter behavior: - *required*
returnMissingObjects	boolean	Regras para indicar se o objeto de resposta deve conter informações sobre objetos ausentes; padrão: `false`.

Attention:
Se nenhuma regra for informada, não haverá nada para atualizar e o resultado ficará 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` oferece suporte aos seguintes parâmetros.

Parameter	Type	Description
dashboards	object	Regras sobre como importar dashboards. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos dashboards serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true` e um dashboard importável específico estiver presente no sistema, ele será totalmente atualizado. As páginas e widgets serão totalmente substituídos pelos novos; padrão: `false`.
discoveryRules	object	Regras sobre como importar regras de LLD. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas regras de LLD serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, regras de LLD existentes serão atualizadas; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, regras de LLD não presentes nos dados importados serão excluídas do banco de dados; padrão: `false`.
global_regexes	object	Regras sobre como importar expressões regulares globais. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas expressões regulares globais serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, expressões regulares globais existentes serão atualizadas, com suas subexpressões totalmente substituídas; padrão: `false`.

Parameter	Type	Description
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`.
httptests	object	Regras sobre como importar cenários web. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos cenários web serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, cenários web existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, cenários web não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
images	object	Regras sobre como importar imagens. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas imagens serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, imagens existentes serão atualizadas; padrão: `false`. Este parâmetro não fará diferença na saída. Ele é permitido apenas por consistência com `configuration.import`.
items	object	Regras sobre como importar items. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos items serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, items existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, items não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
maps	object	Regras sobre como importar mapas. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas existentes serão atualizados; padrão: `false`. Este parâmetro não fará diferença na saída. Ele é permitido apenas por consistência com `configuration.import`.

Parameter	Type	Description
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. Ele é permitido apenas por consistência com `configuration.import`.
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`.
template_groups	object	Regras sobre como importar grupos de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de template serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de template existentes serão atualizados; padrão: `false`.
templateLinkage	object	Regras sobre como importar links de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, templates que não estão vinculados ao host ou template sendo importado, mas que 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`.
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`.

Parameter	Type	Description
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 tem as seguintes propriedades.

Property	Type	Description
default_lang	string	Idioma padrão do sistema. Padrão: `en_US`.
default_timezone	string	Fuso horário padrão do sistema. Padrão: `system` - padrão do sistema. Para obter a lista completa de fusos horários compatíveis, consulte a documentação do PHP.
default_theme	string	Tema padrão. Valores possíveis: `blue-theme` - *(padrão)* Azul; `blue-classic-theme` - Azul (clássico); `dark-theme` - Escuro; `dark-blue-theme` - Azul escuro; `dark-classic-theme` - Escuro (clássico); `hc-light` - Claro de alto contraste; `hc-dark` - Escuro de alto contraste.
search_limit	integer	Limite para resultados de pesquisa e filtragem. Padrão: 1000.
max_overview_table_size	integer	Número máximo de colunas e linhas nos widgets de dashboard Visão geral dos dados e Visão geral dos triggers. Padrão: 50.
max_in_table	integer	Número máximo de elementos a serem exibidos dentro de uma célula da tabela. Padrão: 50.
server_check_interval	integer	Exibir um aviso se o Zabbix server estiver inativo. Valores possíveis: 0 - Não exibir aviso; 10 - *(padrão)* Exibir aviso.
work_period	string	Horário de trabalho. Padrão: 1-5,09:00-18:00.
show_technical_errors	integer	Exibir erros técnicos (PHP/SQL) para usuários que não são Super admin e para usuários que não fazem parte de grupos de usuários com o modo de depuração ativado. Valores possíveis: 0 - *(padrão)* Não exibir erros técnicos; 1 - Exibir erros técnicos.

Property	Type	Description
history_period	string	Período máximo para exibir dados históricos nos widgets de dashboard Últimos dados, Web e Visão geral dos dados. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 24h.
period_default	string	Período padrão do filtro de tempo. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d, 1M, 1y), incluindo suporte para meses e anos. Padrão: 1h.
max_period	string	Período máximo do filtro de tempo. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d, 1M, 1y), incluindo suporte para meses e anos. Padrão: 2y.
severity_color_0	string	Cor da severidade ”Não classificado” como código de cor hexadecimal. Padrão: 97AAB3.
severity_color_1	string	Cor da severidade ”Informação” como código de cor hexadecimal. Padrão: 7499FF.
severity_color_2	string	Cor da severidade ”Aviso” como código de cor hexadecimal. Padrão: FFC859.
severity_color_3	string	Cor da severidade ”Média” como código de cor hexadecimal. Padrão: FFA059.
severity_color_4	string	Cor da severidade ”Alta” como código de cor hexadecimal. Padrão: E97659.
severity_color_5	string	Cor da severidade ”Desastre” como código de cor hexadecimal. Padrão: E45959.
severity_name_0	string	Nome da severidade ”Não classificado”. Padrão: Não classificado.
severity_name_1	string	Nome da severidade ”Informação”. Padrão: Informação.
severity_name_2	string	Nome da severidade ”Aviso”. Padrão: Aviso.
severity_name_3	string	Nome da severidade ”Média”. Padrão: Média.
severity_name_4	string	Nome da severidade ”Alta”. Padrão: Alta.
severity_name_5	string	Nome da severidade ”Desastre”. Padrão: Desastre.
custom_color	integer	Usar cores personalizadas para o status dos eventos. Valores possíveis: 0 - *(padrão)* Não usar cores personalizadas para o status dos eventos; 1 - Usar cores personalizadas para o status dos eventos.

Property	Type	Description
ok_period	string	Exibir o período dos triggers OK. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h). Padrão: 5m.
blink_period	string	Período de intermitência ao alterar o estado do trigger. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h). Padrão: 2m.
problem_unack_color	string	Cor dos eventos PROBLEM não reconhecidos como código de cor hexadecimal. Padrão: CC0000.
problem_ack_color	string	Cor dos eventos PROBLEM reconhecidos como código de cor hexadecimal. Padrão: CC0000.
ok_unack_color	string	Cor dos eventos RESOLVED não reconhecidos como código de cor hexadecimal. Padrão: 009900.
ok_ack_color	string	Cor dos eventos RESOLVED reconhecidos como código de cor hexadecimal. Padrão: 009900.
problem_unack_style	integer	Intermitência para eventos PROBLEM não reconhecidos. Valores possíveis: 0 - Não exibir intermitência; 1 - *(padrão)* Exibir intermitência.
problem_ack_style	integer	Intermitência para eventos PROBLEM reconhecidos. Valores possíveis: 0 - Não exibir intermitência; 1 - *(padrão)* Exibir intermitência.
ok_unack_style	integer	Intermitência para eventos RESOLVED não reconhecidos. Valores possíveis: 0 - Não exibir intermitência; 1 - *(padrão)* Exibir intermitência.
ok_ack_style	integer	Intermitência para eventos RESOLVED reconhecidos. Valores possíveis: 0 - Não exibir intermitência; 1 - *(padrão)* Exibir intermitência.
url	string	URL do frontend.
discovery_groupid	ID	ID do grupo de hosts ao qual os hosts descobertos serão adicionados automaticamente.
default_inventory_mode	integer	Modo padrão do inventário do host. Valores possíveis: -1 - *(padrão)* Desativado; 0 - Manual; 1 - Automático.
alert_usrgrpid	ID	ID do grupo de usuários ao qual será enviada a mensagem de alarme de banco de dados inativo. Se definido como ”0”, a mensagem de alarme não será enviada.

Property	Type	Description
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 malsucedidas após o qual o formulário de login será bloqueado. Padrão: 5.
login_block	string	Intervalo durante o qual o formulário de login será bloqueado se o número de tentativas de login malsucedidas exceder o definido no campo login_attempts. Aceita um número de segundos (por exemplo, 1, 60) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m). 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 sandbox para iframe. Valores possíveis: 0 - Não usar; 1 - *(padrão)* Usar.
iframe_sandboxing_exceptions	string	Exceções do sandbox para iframe.
connect_timeout	string	Tempo limite de conexão com o Zabbix server. O intervalo de valores possíveis é: 1-30s. Padrão: 3s. Comportamento da propriedade: - *obrigatória*
socket_timeout	string	Tempo limite de rede padrão. O intervalo de valores possíveis é: 1-300s. Padrão: 3s. Comportamento da propriedade: - *obrigatória*
media_type_test_timeout	string	Tempo limite de rede para o teste do tipo de mídia. O intervalo de valores possíveis é: 1-300s. Padrão: 65s. Comportamento da propriedade: - *obrigatória*

Property	Type	Description
item_test_timeout	string	Tempo limite de rede para testes de item. O intervalo de valores possíveis é: 1-600s. Padrão: 60s. Comportamento da propriedade: - *obrigatória*
script_timeout	string	Tempo limite de rede para execução de scripts. O intervalo de valores possíveis é: 1-300s. Padrão: 60s. Comportamento da propriedade: - *obrigatória*
report_test_timeout	string	Tempo limite de rede para o teste de relatório agendado. O intervalo de valores possíveis é: 1-300s. Padrão: 60s. Comportamento da propriedade: - *obrigatória*
device_link_timeout	string	Tempo limite para adicionar um dispositivo móvel para uso com o aplicativo móvel do Zabbix; aplica-se à comunicação entre a API do Zabbix e o adaptador de bridge. O intervalo de valores possíveis é: 1-300s. Padrão: 60s. Comportamento da propriedade: - *obrigatória*
auditlog_enabled	integer	Indica se o registro de auditoria deve ser ativado. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
auditlog_mode	integer	Indica se o registro de auditoria de descoberta de baixo nível, descoberta de rede e atividades de registro automático realizadas pelo server (usuário System) deve ser ativado. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
ha_failover_delay	string	Atraso do failover em segundos. Padrão: 1m. Comportamento da propriedade: - *somente leitura*

Property	Type	Description
geomaps_tile_provider	string	Provedor de blocos do mapa geográfico. 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. É compatível com uma string vazia para especificar valores personalizados de `geomaps_tile_url`, `geomaps_max_zoom` e `geomaps_attribution`.
geomaps_tile_url	string	URL dos blocos do mapa geográfico. Comportamento da propriedade: - *compatível* se `geomaps_tile_provider` estiver definido como uma string vazia
geomaps_max_zoom	integer	Nível máximo de zoom do mapa geográfico. Valores possíveis: 0-30. Comportamento da propriedade: - *compatível* se `geomaps_tile_provider` estiver definido como uma string vazia
geomaps_attribution	string	Texto de atribuição do mapa geográfico. Comportamento da propriedade: - *compatível* se `geomaps_tile_provider` estiver definido como uma string vazia
vault_provider	integer	Provedor do Vault. Valores possíveis: 0 - *(padrão)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Resolver valores de macros secretas por meio de: Valores possíveis: 0 - *(padrão)* somente o server; 1 - server e proxies independentemente.
timeout_zabbix_agent	string	Não gastar mais do que `timeout_*` segundos no processamento. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-600s. Padrão: 3s. Padrão para `timeout_browser`: 60s. Comportamento da propriedade: - *obrigatória*
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		

Property	Type	Description
timeout_telemetry_query		
apm_global_db	object	Configuração da fonte de dados global.

Fonte de dados global

O objeto de fonte de dados global tem as seguintes propriedades:

Propriedade	Tipo	Descrição
status	integer	Status da fonte de dados global. Valores possíveis: 0 - *(padrão)* não configurada; 1 - configurada.
url	string	URL usada para conectar-se à fonte de dados. Comportamento da propriedade: - *obrigatória* se o usuário alterar `status` para 1 (configurada).
authentication_type	integer	Tipo de autenticação. Valores possíveis: 0 - *(padrão)* usando um nome de usuário e uma senha; 1 - nenhuma autenticação usada. Comportamento da propriedade: - *obrigatória* se o usuário alterar `status` para 1 (configurada).
username	string	Nome de usuário usado para fazer login na fonte de dados. Comportamento da propriedade: - *suportada* se `status` estiver definido como 1 (configurada) e `authentication_type` estiver definido como 0 (nome de usuário e senha). - *obrigatória* se o usuário alterar `authentication_type` para 0 (nome de usuário e senha).
password	string	Senha usada para fazer login na fonte de dados. O valor deste parâmetro pode estar vazio. Comportamento da propriedade: - *suportada* se `status` estiver definido como 1 (configurada) e `authentication_type` estiver definido como 0 (nome de usuário e senha).
db	string	Nome do banco de dados. O valor deste parâmetro pode estar vazio. Comportamento da propriedade: - *suportada* se `status` estiver definido como 1 (configurada).
ssl_verify_peer	integer	Verificar o certificado do server. Valores possíveis: 0 - *(padrão)* desabilitado; 1 - habilitado. Comportamento da propriedade: - *suportada* se `status` estiver definido como 1 (configurada) e `url` usar o esquema `https://`.

Propriedade	Tipo	Descrição
ssl_verify_host	integer	Verificar o host. Valores possíveis: 0 - desabilitado; 1 - *(padrão)* habilitado. Comportamento da propriedade: - *suportada* se `status` estiver definido como 1 (configurada) e `url` usar o esquema `https://`.

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 de usuário. Consulte Funções de usuário para obter 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

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
```

```
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Não classificado",
        "severity_name_1": "Informação",
        "severity_name_2": "Aviso",
        "severity_name_3": "Médio",
        "severity_name_4": "Alto",
        "severity_name_5": "Desastre",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
```

```
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```


Fonte

CSettings::get() em *ui/include/classes/api/services/CSettings.php*.

settings.update

Descrição

`object settings.update(object settings)`

Este método permite atualizar 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. Consulte User roles para obter 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`.

Connector

O objeto connector tem as seguintes propriedades.

Property	Type	Description
connectorid	ID	ID do connector. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do connector. Comportamento da propriedade: - *obrigatório* para operações de criação
url	string	URL do endpoint, ou seja, URL do receptor. Macros de usuário são compatíveis. 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 dados. Valores possíveis: 0 - *(padrão)* Valores de item; 1 - Eventos.
item_value_type	integer	Soma dos tipos de valores de item a serem enviados. Valores possíveis: 1 - Numérico (ponto flutuante); 2 - Caractere; 4 - Log; 8 - Numérico (sem sinal); 16 - Texto; 32 - Binário; 64 - JSON. Padrão: 31 - Todos os tipos de item (exceto binário e JSON). Comportamento da propriedade: - *compatível* se `data_type` estiver definido como ”Valores de item”.

Property	Type	Description
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 com sinal de 32 bits). Padrão: 0 - Ilimitado.
max_senders	integer	Número de processos de envio a serem executados para este connector. Valores possíveis: 1-100. Padrão: 1.
max_attempts	integer	Número de tentativas. Valores possíveis: 1-5. Padrão: 1.
attempt_interval	string	Intervalo entre as tentativas novamente. Aceita segundos. Valores possíveis: 0s-10s. Padrão: 5s.
timeout	string	Comportamento da propriedade: - *compatível* se `max_attempts` for maior que 1. Tempo limite. Sufixos de tempo são compatíveis (por exemplo, `30s`, `1m`). Macros de usuário são compatíveis. 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 compatíveis.
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 compatíveis. Comportamento da propriedade: - *compatível* se `authtype` estiver definido como ”Básico”, ”NTLM”, ”Kerberos” ou ”Digest”
password	string	Senha. Macros de usuário são compatíveis. Comportamento da propriedade: - *compatível* se `authtype` estiver definido como ”Básico”, ”NTLM”, ”Kerberos” ou ”Digest”

Property	Type	Description
token	string	Token Bearer. Macros de usuário são compatíveis. Comportamento da propriedade: - *obrigatório* se `authtype` estiver definido como ”Bearer”
verify_peer	integer	Se o certificado do host deve ser validado quanto à autenticidade. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
verify_host	integer	Se o nome do host da conexão deve ser validado para verificar se corresponde ao nome no certificado do host. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
ssl_cert_file	string	Caminho do arquivo da chave SSL pública. Macros de usuário são compatíveis.
ssl_key_file	string	Caminho do arquivo da chave SSL privada. Macros de usuário são compatíveis.
ssl_key_password	string	Senha do arquivo da chave SSL. Macros de usuário são compatíveis.
description	text	Descrição do connector.
status	integer	Se o connector está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
tags_evaltype	integer	Método de avaliação das tags. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.

Filtro de tag

O filtro de tag permite exportar apenas os valores de item ou eventos correspondentes. Se não for definido, tudo será exportado.
O objeto de filtro de tag tem as seguintes propriedades.

Property	Type	Description
tag	string	Nome da tag. Comportamento da propriedade: - *required*
operator	integer	Operador da condição. Valores possíveis: 0 - *(default)* Igual a; 1 - Diferente de; 2 - Contém; 3 - Não contém; 12 - Existe; 13 - Não existe.
value	string	Valor da tag.

connector.create

Descrição

`object connector.create(object/array connectors)`

Este método permite criar novos objetos 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Objetos de conector a serem criados.

Além das propriedades padrão do conector, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
tags	array	Filtro de tag do conector.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos conectores criados 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 de usuário. Consulte Funções de usuário para obter 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 connector de acordo com os parâmetros informados.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

(object) Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
connectorids	ID/array	Retorna apenas os connectors com os IDs fornecidos.
selectTags	query	Retorna uma propriedade tags com o filtro de tag do connector. Suporta count.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: connectorid, name, data_type, status.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

(integer/array) Retorna:

- um array 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 de usuário. Consulte Funções de usuário para obter 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*.

Correlation

Esta classe foi projetada para trabalhar com correlações.

Referências de objetos:

- Correlation
- Correlation operation
- Correlation filter
  - Correlation filter condition

Métodos disponíveis:

- correlation.create - cria novas correlações
- correlation.delete - exclui correlações
- correlation.get - recupera correlações
- correlation.update - atualiza 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. 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. Ele possui as seguintes propriedades.

Property	Type	Description
conditions	array	Conjunto de condições de filtro a ser usado para filtrar resultados. As condições serão ordenadas de acordo com sua posição na fórmula. Comportamento da propriedade: - *required*
evaltype	integer	Método de avaliação da condição de filtro. Valores possíveis: 0 - And/Or; 1 - And; 2 - Or; 3 - Expressão personalizada. Comportamento da propriedade: - *required*

Property	Type	Description
eval_formula	string	Expressão gerada que será usada para avaliar as condições de filtro. A expressão contém IDs que fazem referência a condições de filtro específicas por meio de seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *read-only*
formula	string	Expressão definida pelo usuário a ser usada para avaliar condições de filtros com uma expressão personalizada. A expressão deve conter IDs que fazem referência a condições de filtro específicas por meio de seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições de filtro: nenhuma condição pode permanecer sem uso ou ser omitida. Comportamento da propriedade: - *required* se `evaltype` estiver definido como ”expressão personalizada”

Condição de filtro de correlação

O objeto de condição de 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 evento novo; 3 - par de tags de evento; 4 - valor da tag de evento antigo; 5 - valor da tag de evento novo. Comportamento da propriedade: - *obrigatório*
tag	string	Tag do evento (antigo ou novo). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”tag de evento antigo”, ”tag de evento novo”, ”valor da tag de evento antigo” ou ”valor da tag de evento novo”
groupid	ID	ID do grupo de hosts. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”grupo de hosts do evento novo”
oldtag	string	Tag do evento antigo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”par de tags de evento”
newtag	string	Tag do evento novo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”par de tags de evento”
value	string	Valor da tag do evento (antigo ou novo). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”valor da tag de evento antigo” ou ”valor da tag de evento novo”

Propriedade	Tipo	Descrição
formulaid	string	ID único arbitrário que é usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente.
operator	integer	Operador da condição. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como "grupo de hosts do evento novo", "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, veja 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 obter 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 de usuário. Consulte Funções de usuário para obter 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 informados.

> **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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
correlationids	ID/array	Retorna apenas as correlações com os IDs fornecidos.
selectFilter	query	Retorna uma propriedade `filter` com as condições de correlação.
selectOperations	query	Retorna uma propriedade `operations` com as operações de correlação.
sortfield	string/array	Classifica 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. Consulte User roles para obter 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 tem as seguintes propriedades.

Property	Type	Description
dashboardid	ID	ID do dashboard. Property behavior: - *read-only* - *required* para operações de atualização
name	string	Nome do dashboard. Property behavior: - *required* para operações de criação
userid	ID	ID do usuário que é o proprietário do dashboard.

Property	Type	Description
private	integer	Tipo de compartilhamento do dashboard. Valores possíveis: 0 - dashboard público; 1 - *(default)* dashboard privado.
display_period	integer	Intervalo padrão do slideshow (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 - *(default)* iniciar slideshow automaticamente.

Página do dashboard

O objeto da página do dashboard possui as seguintes propriedades.

Property	Type	Description
dashboard_pageid	ID	ID da página do dashboard. Property behavior: - *read-only*
name	string	Nome da página do dashboard. Padrão: string vazia.
display_period	integer	Intervalo da apresentação de slides do dashboard (em segundos). Valores possíveis: 0, 10, 30, 60, 120, 600, 1800, 3600. Padrão: 0 (usará o intervalo padrão da apresentação de slides).
widgets	array	Array dos 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*

Propriedade	Tipo	Descrição
type	string	Tipo do widget do dashboard. Valores possíveis: actionlog - Log de ações; 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 de 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; scatterplot - Gráfico de dispersão; 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 do widget do dashboard. Comportamento da propriedade: - veja widgets individuais em Campos do widget do dashboard

Campo do widget do dashboard

O objeto 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 em 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Dashboards a serem criados.

Além das propriedades padrão do dashboard, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas do dashboard a serem criadas para o dashboard. As páginas do dashboard serão ordenadas na mesma ordem especificada. Comportamento do parâmetro: - *obrigatório*
users	array	Compartilhamentos de usuário do dashboard a serem criados no dashboard.
userGroups	array	Compartilhamentos de grupo 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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dashboardids	ID/array	Retornar apenas dashboards com os IDs fornecidos.
selectPages	query	Retornar uma propriedade `pages` com as páginas do dashboard, corretamente ordenadas.
selectUsers	query	Retornar uma propriedade `users` com os usuários com quem o dashboard é compartilhado.
selectUserGroups	query	Retornar uma propriedade `userGroups` com os grupos de usuários com quem o dashboard é compartilhado.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `dashboardid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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 de usuário. Consulte User roles para obter 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 de cada widget, use a barra lateral para navegar até a página correspondente do widget.

1 Log de ações

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com o widget *Log de ações*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Destinatários	11	userids.0	ID do usuário. Observação: para configurar vários usuários, crie um objeto de campo de widget de dashboard para cada usuário, usando um número incrementado no nome da propriedade.

Parâmetro	tipo	nome	valor
Ações	12	actionids.0	ID da ação. Observação: para configurar várias ações, crie um objeto de campo de widget de dashboard para cada ação, usando um número incrementado no nome da propriedade.
Tipos de mídia	13	mediatypeids.0	ID do tipo de mídia. Observação: para configurar vários tipos de mídia, crie um objeto de campo de widget de dashboard para cada tipo de mídia, usando um número incrementado no nome da propriedade.
Status	0	statuses.0	0 - Em andamento; 1 - Enviado/Executado; 2 - Falhou. Observação: para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor, usando um número incrementado no nome da propriedade.
String de pesquisa	1	message	Qualquer valor de string.
Período	1	time_period._referenceDASHBOARD._timeperiod	`DASHBOARD._timeperiod` - define o seletor de período do dashboard como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período somente 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 relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na sintaxe de tempo relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Ordenar entradas por	0	sort_triggers	3 - Hora (crescente); 4 - *(padrão)* Hora (decrescente); 5 - Tipo (crescente); 6 - Tipo (decrescente); 7 - Status (crescente); 8 - Status (decrescente); 11 - Destinatário (crescente); 12 - Destinatário (decrescente).
Mostrar linhas	0	show_lines	Os valores possíveis variam de 1 a 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Clock* 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 integrados 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 *Clock*, 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).
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 do widget do dashboard permitem configurar o widget *Status da descoberta* 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 do 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 do widget do 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

Estes 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 integrados 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 descrito 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 compatíveis com o widget *Gauge*.

Note:
O número no nome da propriedade *Thresholds* (por exemplo, thresholds.0.color) faz referência à posição do limite em uma lista, classificada em ordem crescente. No entanto, se os limites forem configurados em uma ordem diferente, os valores serão classificados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parâmetro	type	name	value
Ângulo	0	angle	Valores possíveis: 180 *(padrão)* ou 270.
Descrição			
Descrição	1	description	Qualquer valor de string, incluindo macros. Macros compatíveis: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {ITEM.NAME}.
Tamanho	0	desc_size	Os valores possíveis variam de 1 a 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	Os valores possíveis variam de 1 a 10. Padrão: 2.
Tamanho	0	value_size	Os valores possíveis variam de 1 a 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 (caixa de seleção)	0	units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Unidades (valor)	1	units	Qualquer valor de string. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado”
Tamanho	0	units_size	Os valores possíveis variam de 1 a 100. Padrão: 25. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado”
Negrito	0	units_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado”

Parâmetro	type	name	value
Posição	0	units_pos	0 - Antes do valor; 1 - Acima do valor; 2 - *(padrão)* Depois do valor; 3 - Abaixo do valor. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado” Este parâmetro será ignorado se for definido como 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	Os valores possíveis variam de 1 a 100. Padrão: 20.
Ponteiro			
Cor	1	needle_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio). Comportamento do parâmetro: - *compatível* se um objeto de campo do widget do dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou se *Mostrar arco* estiver definido como ”Habilitado”
Escala			
Mostrar unidades	0	scale_show_units	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado” e um objeto de campo do widget do dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou se *Mostrar arco* estiver definido como ”Habilitado”
Tamanho	0	scale_size	Os valores possíveis variam de 1 a 100. Padrão: 15. Comportamento do parâmetro: - *compatível* se um objeto de campo do widget do dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou se *Mostrar arco* estiver definido como ”Habilitado”
Casas decimais	0	scale_decimal_places	Os valores possíveis variam de 1 a 10. Padrão: 0. Comportamento do parâmetro: - *compatível* se um objeto de campo do widget do dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou se *Mostrar arco* estiver definido como ”Habilitado”
Limites			
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Limite	1	thresholds.0.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são compatíveis.

Parâmetro	type	name	value
Mostrar rótulos	0	th_show_labels	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Limites* estiver definido e um objeto de campo do widget do dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou se *Mostrar arco* estiver definido como ”Habilitado”
Mostrar arco	0	th_show_arc	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Limites* estiver definido
Tamanho do arco	0	th_arc_size	Os valores possíveis variam de 1 a 100. Padrão: 5. Comportamento do parâmetro: - *compatível* 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Geomap* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets integrados 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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com 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. Observação: para configurar vários grupos de hosts, crie um objeto de campo de widget do dashboard para cada grupo de hosts, com um número incrementado no nome da propriedade. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com o parâmetro *Reference* definido como "`ABCDE`") como a fonte de dados dos grupos de hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Hosts	3	hostids.0	ID do host. Observação: para configurar vários hosts, crie um objeto de campo de widget do dashboard para cada host, com um número incrementado no nome da propriedade. Para vários 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 compatível ao configurar 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 hosts do dashboard como a fonte de dados dos hosts; `ABCDE._hostids` - defina um widget compatível (com o parâmetro *Reference* definido como "`ABCDE`") como a fonte de dados dos hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Tags			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/OU; 2 - OU. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Operador	0	tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Valor da tag	1	tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Visualização inicial	1	default_view	*latitude*, *longitude* e *nível de zoom* separados por vírgulas (*opcional*, os valores possíveis variam de 0 a 30). Exemplo: `40.6892494,-74.0466891,10`.
Agrupamento			
Modo de agrupamento	0	clustering_mode	0 - *(padrão)* Automático; 1 - Nível de zoom.
Nível de zoom do agrupamento	0	clustering_zoom_level	Os valores possíveis variam de 0 a 30. Padrão: 0. Comportamento do parâmetro: - *compatível* se *Modo de agrupamento* estiver definido como "Nível de zoom"
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Esse valor deve ser exclusivo no 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Graph* 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 integrados 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 *Graph*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Conjunto de dados

Os parâmetros a seguir são compatíveis com a configuração de 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	type	name	value
Tipo de conjunto de dados	0	ds.0.dataset_type	0 - Lista de itens; 1 - Padrões de itens *(padrão)*.
Itens	4	ds.0.itemids.0	ID do item. Ao configurar o widget em um dashboard de template, somente os itens configurados no template devem ser definidos. Observação: para configurar vários itens, crie um objeto de campo de widget do dashboard para cada item, usando um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens" e *Itens (Widget)* não estiver definido
Itens (Widget)	1	ds.0.itemids.0._reference	Em vez do ID do item: `ABCDE._itemid` - defina um widget compatível (com o parâmetro *Referência* definido como "ABCDE") como a fonte de dados dos itens. Observação: para configurar vários widgets, crie um objeto de campo de widget do dashboard para cada widget, usando um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens" e *Itens* não estiver definido
Cor	1	ds.0.color.0	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens"
Padrões de host	1	ds.0.hosts.0	Nome ou padrão de host (por exemplo, "Zabbix*"). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Padrões de itens" Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Padrões de item	1	ds.0.items.0	Nome ou padrão de item (por exemplo, "*: Número de *valores processados por segundo"). Ao configurar o widget em um dashboard de template, somente os padrões dos itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Padrões de itens"

Parâmetro	type	name	value
Cor	1	ds.0.color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *compatível* se *Tipo de conjunto de dados* estiver definido como ”Padrões de itens” e *Paleta de cores* não estiver definida
Paleta de cores	0	ds.0.color_palette	Índice da paleta de cores. Os valores possíveis variam de 0 a 11. Padrão: 0. Comportamento do parâmetro: - *compatível* se *Tipo de conjunto de dados* estiver definido como ”Padrões de itens” e *Cor* não estiver definida
Desenho	0	ds.0.type	0 - Linha *(padrão)*; 1 - Pontos; 2 - Escada; 3 - Barra.
Empilhado	0	ds.0.stacked	0 - Desabilitado *(padrão)*; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Linha”, ”Escada” ou ”Barra”
Largura	0	ds.0.width	Os valores possíveis variam de 1 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Linha” ou ”Escada”
Tamanho do ponto	0	ds.0.pointsize	Os valores possíveis variam de 1 a 10. Padrão: 3. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Pontos”
Transparência	0	ds.0.transparency	Os valores possíveis variam de 1 a 10. Padrão: 5.
Preenchimento	0	ds.0.fill	Os valores possíveis variam de 1 a 10. Padrão: 3. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Linha” ou ”Escada”
Dados ausentes	0	ds.0.missingdatafunc	0 - Nenhum *(padrão)*; 1 - Conectado; 2 - Tratar como 0; 3 - Último conhecido. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Linha” ou ”Escada”
Substituir host	1	ds.0.override_hostid.ABCDE._hostid	`ABCDE._hostid` - defina um widget compatível (com o parâmetro ”Referência” definido como ”ABCDE”) como a fonte de dados dos hosts; `DASHBOARD._hostid` - defina o seletor de host do dashboard como a fonte de dados dos hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Eixo Y	0	ds.0.axisy	0 - Esquerdo *(padrão)*; 1 - Direito.

Parâmetro	type	name	value
Inverter valores	0	ds.0.invert_values	0 - Desabilitado (padrão); 1 - Habilitado.
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 agregação	0	ds.0.aggregate_function	0 - não utilizada *(padrão)*; 1 - mín; 2 - máx; 3 - média; 4 - contagem; 5 - soma; 6 - primeiro; 7 - último.
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.
Agregar	0	ds.0.aggregate_grouping	0 - Cada item *(padrão)*; 1 - Conjunto de dados. Comportamento do parâmetro: - *compatível* se *Função de agregação* estiver definida como ”mín”, ”máx”, ”média”, ”contagem”, ”soma”, ”primeiro” ou ”último”
Aproximação	0	ds.0.approximation	1 - mín; 2 - média *(padrão)*; 4 - máx; 7 - todos.
Rótulo do conjunto de dados	1	ds.0.data_set_label	Qualquer valor de string. Padrão: "" (vazio).
Tags de item			
Tipo de avaliação	0	ds.0.item_tags_evaltype	0 - E/OU *(padrão)*; 2 - OU.
Nome da tag	1	ds.0.item_tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*
Operador	0	ds.0.item_tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*

Parâmetro	type	name	value
Valor da tag	1	ds.0.item_tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*

Opções de exibição

Os seguintes parâmetros são compatíveis com a configuração de *Opções de exibição*.

Parâmetro	type	name	value
Seleção de dados históricos	0	source	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.
Triggers simples	0	simple_triggers	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Período de trabalho	0	working_time	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Nomes dos hosts nos títulos	0	show_hostnames	0 - *(padrão)* Automático; 1 - Mostrar; 2 - Ocultar. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Linha de percentil (esquerda)			
Status	0	percentile_left	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Eixo Y* (na configuração de *Conjunto de dados*) estiver definido como ”Esquerda”
Valor	0	percentile_left_value	Os valores possíveis variam de 1 a 100. Comportamento do parâmetro: - *compatível* se *Eixo Y* (na configuração de *Conjunto de dados*) estiver definido como ”Esquerda”
Linha de percentil (direita)			

Parâmetro	type	name	value
Status	0	percentile_right	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Eixo Y* (na configuração de *Conjunto de dados*) estiver definido como "Direita"
Valor	0	percentile_right_value	Os valores possíveis variam de 1 a 100. Comportamento do parâmetro: - *compatível* se *Eixo Y* (na configuração de *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 em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

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"
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.

Parâmetro	tipo	nome	valor
Mín	1	lefty_min	Qualquer valor numérico. Padrão: "" (vazio).
Máx	1	righty_min lefty_max	Qualquer valor numérico. Padrão: "" (vazio).
Unidades (tipo)	0	righty_max lefty_units	0 - *(padrão)* Automático; 1 - Estático.
Unidades (valor)	1	righty_units lefty_static_units	Qualquer valor de string. Padrão: "" (vazio).
Eixo X	0	righty_static_units xaxis	0 - Desabilitado; 1 - *(padrão)* Habilitado.

Legenda

Os seguintes parâmetros são compatíveis com a configuração de *Legenda*.

Parâmetro	tipo	nome	valor
Mostrar legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Exibir mín-imo/média/máximo	0	legend_statistic	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como ”Habilitado”
Mostrar função de agregação	0	legend_aggregation	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como ”Habilitado”
Linhas	0	legend_lines_mode	0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como ”Habilitado”
Número de linhas/ Número máximo de linhas	0	legend_lines	Os valores possíveis variam de 1 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como ”Habilitado”
Número de colunas	0	legend_columns	Os valores possíveis variam de 1 a 4. Padrão: 4. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como ”Habilitado” e *Exibir mínimo/média/máximo* 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*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Parâmetro	tipo	nome	valor
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Substituições

Os seguintes parâmetros são compatíveis com a configuração de *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 de host (por exemplo, `Zabbix*`). Este parâmetro não é compatível ao configurar o widget em um dashboard de template. Comportamento do parâmetro: - *obrigatório* ao configurar *Substituições*
Padrões de item	1	or.0.items.0	Nome ou padrão de item (por exemplo, `*: Number of processed *values per second`). Ao configurar o widget em um dashboard de template, somente os padrões dos itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* ao configurar *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.
Desenho	0	or.0.type	0 - Linha; 1 - Pontos; 2 - Escada; 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 - Esquerdo; 1 - Direito.
Inverter valores	0	or.0.invert_values	0 - (padrão) Desabilitado; 1 - Habilitado.
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 ”Item list” e consiste em 3 items que são representados por linhas com cores diferentes, mas com a mesma largura, transparência e preenchimento.
- O segundo conjunto de dados é do tipo ”Item patterns”, consiste em 6 items, possui uma agregação configurada e é representado por uma linha com cor, largura, transparência e preenchimento personalizados.
- O segundo conjunto de dados também possui um rótulo de conjunto de dados personalizado.
- Os dados no gráfico são exibidos para um período de tempo das últimas 3 horas.
- Os problemas no gráfico são exibidos apenas para os items configurados.
- O gráfico possui dois eixos Y, dos quais o eixo Y direito exibe valores apenas para o segundo conjunto de dados.
- A legenda do gráfico exibe os items configurados em 4 linhas, bem como os valores mínimo, máximo e médio dos conjuntos de dados.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23269
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "BF00FF"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23257
                            },
                            {
                                "type": 1,
```

```
                "name": "ds.0.color.2",
                "value": "0040FF"
            },
            {
                "type": 0,
                "name": "ds.0.width",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.0.transparency",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.0.fill",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.1.hosts.0",
                "value": "Zabbix server"
            },
            {
                "type": 1,
                "name": "ds.1.items.0",
                "value": "*: Number of processed *values per second"
            },
            {
                "type": 1,
                "name": "ds.1.color",
                "value": "000000"
            },
            {
                "type": 0,
                "name": "ds.1.transparency",
                "value": 0
            },
            {
                "type": 0,
                "name": "ds.1.fill",
                "value": 0
            },
            {
                "type": 0,
                "name": "ds.1.axisy",
                "value": 1
            },
            {
                "type": 0,
                "name": "ds.1.aggregate_function",
                "value": 3
            },
            {
                "type": 1,
                "name": "ds.1.aggregate_interval",
                "value": "1m"
            },
            {
                "type": 0,
                "name": "ds.1.aggregate_grouping",
                "value": 1
```

```
                            },
                            {
                                "type": 1,
                                "name": "ds.1.data_set_label",
                                "value": "Number of processed values per second"
                            },
                            {
                                "type": 0,
                                "name": "graph_time",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 1,
                                "name": "time_period.to",
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_statistic",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "show_problems",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "YZABC"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

9 Gráfico (clássico)

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Graph (classic)* 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 integrados 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 *Graph (classic)*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico (clássico)*.

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	0 - *(padrão)* Gráfico; 1 - Gráfico simples.
Gráfico	6	graphid.0	ID do Gráfico. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico"
Gráfico (Widget)	1	graphid._reference	Em vez do ID do Gráfico: `ABCDE._graphid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para gráficos. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico simples" e *Gráfico* não estiver definido
Item	4	itemid.0	ID do Item. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico simples" e *Item (Widget)* não estiver definido

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 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._referenda	`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 - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Exibir legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Sobrescrever host	1	override_hostid._referenda	`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 exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Graph prototype* 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 integrados 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 *Graph prototype*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com 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	1	time_period._referenc	`DASHBOARD._timeperiod` - define o seletor de período do dashboard como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o parâmetro *Referência* definido como "ABCDE") como fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período somente 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 relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.to` estiver definido

Parâmetro	tipo	nome	valor
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: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Mostrar legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Substituir host	1	override_hostid._reference	`ABCDE._hostid` - define um widget compatível (com o parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para hosts; `DASHBOARD._hostid` - define o seletor de hosts do dashboard como fonte de dados para hosts. Este parâmetro não é compatível 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`). Esse valor deve ser exclusivo no 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

Estes 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 integrados 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			
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.

Parâmetro	tipo	nome	valor
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 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* 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.
Padrões de item	1	items.0	Nome ou padrão do Item. Nota: Para configurar múltiplos padrões de item, crie um objeto de campo de widget de dashboard para cada padrão de item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório*
Tags de item			
Tipo de avaliação	0	evaltype_item	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	item_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*

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 compatíveis com o widget *Honeycomb*.

Note:
O número no nome da propriedade *Thresholds* (por exemplo, thresholds.0.color) faz referência à posição do limite em uma lista, classificada em ordem crescente. No entanto, se os limites forem configurados em uma ordem diferente, os valores serão classificados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Rótulo 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 compatíveis: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {HOST.NAME} Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Texto"
Casas decimais	0	primary_label_decimal_places	Os valores possíveis variam de 0 a 6. Padrão: 2. Comportamento do parâmetro: - *compatível* 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 a 100. Padrão: 20. Comportamento do parâmetro: - *compatível* 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 em theme do objeto Settings e do objeto User: 1F2C33 para "blue-theme", "blue-classic-theme" ou "hc-light"; EEEEEE para "dark-theme", "dark-classic-theme" ou "hc-dark"; E5F1FF para "dark-blue-theme".
Unidades (caixa de seleção)	0	primary_label_units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor"
Unidades (valor)	1	primary_label_units	Qualquer valor de string. "" (vazio) Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor" e *Unidades* (caixa de seleção) estiver definido como "Habilitado"
Posição	0	primary_label_units_pos	0 - Antes do valor; 1 - *(padrão)* Depois do valor. Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor" e *Unidades* (caixa de seleção) estiver definido como "Habilitado" Este parâmetro será ignorado se for definido como uma das seguintes unidades relacionadas ao tempo: unixtime, uptime, s.
Rótulo secundário			
Tipo	0	secondary_label_type	0 - Texto; 1 - *(padrão)* Valor.

Parâmetro	tipo	nome	valor
Texto	1	secondary_label	Qualquer valor de string, incluindo macros. Macros compatíveis: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {{ITEM.LASTVALUE}.fmtnum(2)} Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Texto"
Casas decimais	0	secondary_label_decimals	Os valores possíveis variam de 0 a 6. Padrão: 2. Comportamento do parâmetro: - *compatível* 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 a 100. Padrão: 30. Comportamento do parâmetro: - *compatível* 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 em theme do objeto Settings e do objeto User: 1F2C33 para "blue-theme", "blue-classic-theme" ou "hc-light"; EEEEEE para "dark-theme", "dark-classic-theme" ou "hc-dark"; E5F1FF para "dark-blue-theme".
Unidades (caixa de seleção)	0	secondary_label_units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor"
Unidades (valor)	1	secondary_label_units	Qualquer valor de string. "" (vazio) Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor" e *Unidades* (caixa de seleção) estiver definido como "Habilitado"
Posição	0	secondary_label_position	0 - Antes do valor; 1 - *(padrão)* Depois do valor. Comportamento do parâmetro: - *compatível* se *Tipo* estiver definido como "Valor" e *Unidades* (caixa de seleção) estiver definido como "Habilitado" Este parâmetro será ignorado se for 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 em theme do objeto Settings e do objeto User: D9E7ED para "blue-theme" e "blue-classic-theme"; 3D5059 para "dark-theme" e "dark-classic-theme"; 004FA8 para "dark-blue-theme"; AAD7E9 para "hc-light"; 335463 para "hc-dark".
Limites			

Parâmetro	tipo	nome	valor
Interpolação de cores	0	interpolation	0 - Desabilitada; 1 - *(padrão)* Habilitada.
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Limite	1	thresholds.0.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são compatíveis.

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 do widget do 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 descrito 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.

Parâmetro	tipo	nome	valor
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.
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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Host navigator* 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 integrados 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 navigator*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com 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.

Parâmetro	tipo	nome	valor
Grupos de hosts	2	groupids.0	ID do grupo de hosts. Observação: para configurar vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do grupo de hosts: `ABCDE._hostgroupids` - define um widget compatível (com o parâmetro *Referência* definido como "ABCDE") como a fonte de dados dos grupos de hosts. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Padrões de hosts	1	hosts.0	Nome ou padrão de host. Observação: para configurar vários padrões de hosts, crie um objeto de campo de widget de dashboard para cada padrão de host, usando um número incrementado no nome da propriedade. Para vários padrões de hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual pertençam os hosts correspondentes aos padrões configurados. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Status do host	0	status	-1 - *(padrão)* Qualquer; 0 - Habilitado; 1 - Desabilitado. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Tags de host			
Tipo de avaliação	0	host_tags_evaltype	0 - *(padrão)* E/E ou; 2 - Ou. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Nome da tag	1	host_tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Operador	0	host_tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Valor da tag	1	host_tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Severidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Média; 4 - Alta; 5 - Desastre. Padrão: vazio (todas habilitadas). Observação: para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor, usando um número incrementado no nome da propriedade.
Mostrar hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar problemas	0	show_problems	0 - Todos; 1 - *(padrão)* Não suprimidos; 2 - Nenhum.
Agrupar por			
Atributo	0	group_by.0.attribute	0 - Grupo de hosts; 1 - Valor da tag; 2 - Severidade. Observação: o número no nome da propriedade faz referência à ordem do atributo na lista de atributos de agrupamento. Comportamento do parâmetro: - *obrigatório* ao configurar *Agrupar por*

Parâmetro	tipo	nome	valor
Valor	1	group_by.0.tag_name	Qualquer valor de string. Observação: o número no nome da propriedade faz referência ao conjunto de atributos de agrupamento definido no parâmetro *Atributo*. Comportamento do parâmetro: - *obrigatório* ao configurar *Agrupar por* quando *Atributo* está definido como "Valor da tag"
Limite de hosts	0	show_lines	Os valores possíveis variam de 1 a 9999. Padrão: 100. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Esse valor deve ser exclusivo no 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 dos parâmetros 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
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 compatíveis quando *Show* está definido como "Latest data".

Parameter	type	name	value
Width	0	sparkline.width	Os valores possíveis variam de 0 a 10. Padrão: 1.
Fill	0	sparkline.fill	Os valores possíveis variam de 0 a 10. Padrão: 3.
Color	1	sparkline.color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: 42A5F5.
Time period	1	sparkline.time_period.reference	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 reference igual a ABCDE) como a fonte de dados. Padrão: "" (vazio) Como alternativa, você pode definir o período de tempo somente nos parâmetros *From* e *To*.
From	1	sparkline.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.). Padrão: now-1h. Comportamento do parâmetro: - *compatível* se *Time period* não estiver definido - *obrigatório* se sparkline.time_period.to estiver definido
To	1	sparkline.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.). Padrão: now. Comportamento do parâmetro: - *compatível* se *Time period* não estiver definido - *obrigatório* se time_period.to estiver definido
History data	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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Histórico do item*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com o widget *Histórico de itens*.

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 (consulte abaixo)			
Mostrar linhas	0	show_lines	Os valores possíveis variam de 1 a 100. Padrão: 25.
Substituir host	1	override_hostid._referenceABCDE	`ABCDE._hostid` - define um widget compatível (com o parâmetro *Referência* definido como "ABCDE") como a fonte de dados para hosts; `DASHBOARD._hostid` - define o seletor de hosts do dashboard como a fonte de dados para hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Configuração avançada (consulte abaixo)			
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Esse valor deve ser exclusivo no 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 compatíveis 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.

Parâmetro	type	name	value
Linha única	0	columns.0.max_length	Os valores possíveis variam de 1 a 500. Padrão: 100. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como "Linha única"
Usar fonte monoespaçada	0	columns.0.monospace_font	0 - *(padrão)* Usar a fonte padrão; 1 - Usar fonte monoespaçada.
Exibir hora local	0	columns.0.local_time	0 - *(padrão)* Exibir o timestamp; 1 - Exibir a hora local. Comportamento do parâmetro: - *compatível* se *Item* estiver definido como item do tipo log e *Mostrar 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 de item*.

Parâmetro	tipo	nome	valor
Novos valores	0	sortorder	0 - *(padrão)* Topo; 1 - Fundo.
Mostrar carimbo de data/hora	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._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 *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 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 - *obrigatório* se `time_period.to` estiver definido

Parâmetro	tipo	nome	valor
Até	1	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 - *obrigatório* se `time_period.from` 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 itens* 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 integrados 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 itens*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com o widget *Navegador de itens*.

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. Observação: para configurar vários grupos de hosts, crie um objeto de campo de widget do dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível ao configurar 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 o parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para grupos de hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Hosts	3	hostids.0	ID do host. Observação: para configurar vários hosts, crie um objeto de campo de widget do dashboard para cada host, usando um número incrementado no nome da propriedade. Para vários 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 compatível ao configurar o widget em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do host: `DASHBOARD.hostid` - defina o seletor de hosts do dashboard como fonte de dados para hosts; `ABCDE._hostid` - defina um widget compatível (com o parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Tags de host			

Parâmetro	tipo	nome	valor
Tipo de avaliação	0	host_tags_evaltype	0 - *(padrão)* E/E ou; 2 - Ou. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Nome da tag	1	host_tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Operador	0	host_tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Valor da tag	1	host_tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Padrões de item	1	items.0	Nome ou padrão do item. Observação: para configurar vários padrões de item, crie um objeto de campo de widget do dashboard para cada padrão de item, usando um número incrementado no nome da propriedade.
Tags de item			
Tipo de avaliação	0	item_tags_evaltype	0 - *(padrão)* E/E ou; 2 - Ou.
Nome da tag	1	item_tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*

Parâmetro	tipo	nome	valor
Operador	0	item_tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*
Valor da tag	1	item_tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*
Estado	0	state	-1 - *(padrão)* Todos; 0 - Normal; 1 - Não compatível.
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. Observação: o número no nome da propriedade faz referência à ordem do atributo na lista de atributos de agrupamento. Comportamento do parâmetro: - *obrigatório* ao configurar *Agrupar por*
Valor	1	group_by.0.tag_name	Qualquer valor de string. Observação: o número no nome da propriedade faz referência ao conjunto de atributos de agrupamento definido no parâmetro *Atributo*. Comportamento do parâmetro: - *obrigatório* ao configurar *Agrupar por* e quando *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 a 9999. Padrão: 100.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Esse valor deve ser exclusivo no 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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

Parâmetro	tipo	nome	valor
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	type	name	value
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.
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._reference	`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 `reference` igual a ABCDE) como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *From* e *To*. Comportamento do parâmetro: - *suportado* se *Função de agregação* estiver definida como "min", "max", "avg", "count", "sum", "first", "last"

Parâmetro	type	name	value
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 e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `time_period.from` estiver definido
Dados de histórico	0	history	0 - *(padrão)* Auto; 1 - Histórico; 2 - Tendências.

Descrição

Os seguintes parâmetros de configuração avançada são compatíveis quando *Mostrar* está definido como ”Descrição”.

Parâmetro	tipo	name	value
Descrição	1	description	Qualquer valor de string, incluindo macros. Macros compatíveis: `{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*.
Posição vertical	0	desc_v_pos	0 - Superior; 1 - Meio; 2 - *(padrão)* Inferior. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Tamanho	0	desc_size	Os valores possíveis variam de 1 a 100. Padrão: 15.
Negrito	0	desc_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	desc_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).

Valor

Os seguintes parâmetros de configuração avançada são compatíveis quando *Mostrar* está definido como ”Valor”.

Parâmetro	type	name	value
Casas decimais			
Casas decimais	0	decimal_places	Os valores possíveis variam de 1 a 10. Padrão: 2.
Tamanho	0	decimal_size	Os valores possíveis variam de 1 a 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 a 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 (caixa de seleção)	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)* Depois do valor; 3 - Abaixo do valor.
Tamanho	0	units_size	Os valores possíveis variam de 1 a 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 compatíveis quando *Exibir* está 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*.

Parâmetro	tipo	nome	valor
Posição vertical	0	time_v_pos	0 - *(padrão)* Superior; 1 - Meio; 2 - Inferior. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Tamanho	0	time_size	Os valores possíveis variam de 1 a 100. Padrão: 15.
Negrito	0	time_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	time_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Indicador de alteração

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Indicador de alteração”.

Parâmetro	tipo	nome	valor
Cor do indicador de alteração ↑	1	up_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Cor do indicador de alteração ↓	1	down_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Cor do indicador de alteração ↕	1	updown_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Sparkline

Os seguintes parâmetros de configuração avançada são compatíveis quando *Show* está definido como ”Sparkline”.

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	1	sparkline.time_period.reference	DASHBOARD._timeperiod - define o seletor de período do dashboard como fonte de dados; ABCDE._timeperiod - define um widget compatível (com o parâmetro reference igual a ABCDE) como fonte de dados. Padrão: DASHBOARD._timeperiod Como alternativa, você pode definir o período somente nos parâmetros *De* e *Até*.
De	1	sparkline.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: - *obrigatório* se sparkline.time_period.to estiver definido

Parâmetro	tipo	nome	valor
Até	1	sparkline.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: - *obrigatório* se `sparkline.time_period.from` estiver definido
Dados históricos	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

Estes 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 integrados 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 descrito 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Gráfico de pizza* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets integrados 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 parâmetros a seguir são compatíveis com a configuração de 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	type	name	value
Tipo de conjunto de dados	0	ds.0.dataset_type	0 - Lista de itens; 1 - Padrões de itens *(padrão)*.
Itens	4	ds.0.itemids.0	ID do item. Ao configurar o widget em um dashboard de template, somente os itens configurados no template devem ser definidos. Observação: para configurar vários itens, crie um objeto de campo de widget do dashboard para cada item, usando um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens" e *Itens (Widget)* não estiver definido
Itens (Widget)	1	ds.0.itemids.0._reference	Em vez do ID do item: `ABCDE._itemid` - defina um widget compatível (com o parâmetro *Referência* definido como "ABCDE") como a fonte de dados dos itens. Observação: para configurar vários widgets, crie um objeto de campo de widget do dashboard para cada widget, usando um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens" e *Itens* não estiver definido
Cor	1	ds.0.color.0	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de itens"
Tipo de item	0	ds.0.type.0	0 - Normal *(padrão)*; 1 - Total. O valor "Total" pode ser definido para apenas um item em todo o gráfico. Comportamento do parâmetro: - *compatível* se *Tipo de conjunto de dados* estiver definido como "Lista de itens"
Padrões de host	1	ds.0.hosts.0	Nome ou padrão de host (por exemplo, "Zabbix*"). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Padrões de itens" Este parâmetro não é compatível ao configurar o widget em um dashboard de template.

Parâmetro	type	name	value
Padrões de item	1	ds.0.items.0	Nome ou padrão de item (por exemplo, ”*: Número de *valores processados por segundo”). Ao configurar o widget em um dashboard de template, somente os padrões dos itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como ”Padrões de itens”
Cor	1	ds.0.color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *compatível* se *Tipo de conjunto de dados* estiver definido como ”Padrões de itens” e *Paleta de cores* não estiver definida
Paleta de cores	0	ds.0.color_palette	Índice da paleta de cores. Os valores possíveis variam de 0 a 11. Padrão: 0. Comportamento do parâmetro: - *compatível* se *Tipo de conjunto de dados* estiver definido como ”Padrões de itens” e *Cor* não estiver definida
Função de agregação	0	ds.0.aggregate_function	1 - mínimo; 2 - máximo; 3 - média; 4 - contagem; 5 - soma; 6 - primeiro; 7 - último *(padrão)*.
Agregação do conjunto de dados	0	ds.0.dataset_aggregation	0 - nenhuma *(padrão)*; 1 - mínimo; 2 - máximo; 3 - média; 4 - contagem; 5 - soma. Comportamento do parâmetro: - *compatível* 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).
Tags de item			
Tipo de avaliação	0	ds.0.item_tags_evaltype	0 - E/OU *(padrão)*; 2 - OU.
Nome da tag	1	ds.0.item_tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*

Parâmetro	type	name	value
Operador	0	ds.0.item_tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*
Valor da tag	1	ds.0.item_tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de item*

Exibindo opções

Os parâmetros a seguir são compatíveis com a configuração de *Opções de exibição*.

Parâmetro	tipo	nome	valor
Seleção de dados históricos	0	source	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.
Desenho	0	draw_type	0 - *(padrão)* Pizza; 1 - Rosca.
Largura	0	width	20 - 20% do raio; 30 - 30% do raio; 40 - 40% do raio; 50 - *(padrão)* 50% do raio. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Rosca”
Largura do traço	0	stroke	Os valores possíveis variam de 0 a 10. Padrão: 0. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Rosca”
Mostrar valor total	0	total_show	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Desenho* estiver definido como ”Rosca”
Tamanho	0	value_size_type	0 - *(padrão)* Automático; 1 - Personalizado. Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”

Parâmetro	tipo	nome	valor
Tamanho (valor para tamanho personalizado)	0	value_size	Os valores possíveis variam de 1 a 100. Padrão: 20. Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”
Casas decimais	0	decimal_places	Os valores possíveis variam de 0 a 6. Padrão: 2. Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”
Unidades (caixa de seleção)	0	units_show	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”
Unidades (valor)	1	units	Qualquer valor de string. Comportamento do parâmetro: - *compatível* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado”
Negrito	0	value_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”
Cor	1	value_color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *compatível* se *Mostrar valor total* estiver definido como ”Habilitado”
Espaço entre setores	0	space	Os valores possíveis variam de 0 a 10. Padrão: 1.
Mesclar setores menores que N% (caixa de seleção)	0	merge	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mesclar setores menores que N% (valor)	0	merge_percent	Os valores possíveis variam de 1 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Mesclar setores menores que N%* (caixa de seleção) estiver definido como ”Habilitado”
Mesclar setores menores que N% (cor)	1	merge_color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *compatível* se *Mesclar setores menores que N%* (caixa de seleção) estiver definido como ”Habilitado”

Período de tempo

Os seguintes parâmetros são suportados para configurar o *Período de tempo*.

Parâmetro	tipo	nome	valor
Período de tempo	1	time_period._referenceDASHBOARD	`DASHBOARD._timeperiod` - define o seletor de período de tempo do dashboard como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o 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 - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

Legenda

Os parâmetros a seguir são compatíveis com a configuração de *Legenda*.

Parâmetro	tipo	name	value
Mostrar legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Mostrar valor	0	legend_value	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como "Habilitado"
Mostrar função de agregação	0	legend_aggregation	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como "Habilitado"
Linhas	0	legend_lines_mode	0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como "Habilitado"
Número de linhas/ Número máximo de linhas	0	legend_lines	Os valores possíveis variam de 1 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Mostrar legenda* estiver definido como "Habilitado"

Parâmetro	tipo	name	value
Número de colunas	0	legend_columns	Os valores possíveis variam de 1 a 4. Padrão: 4. Comportamento do parâmetro: - *compatível* 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, todos do tipo ”Normal” e representados por uma cor diferente.
- O segundo conjunto de dados é do tipo ”Padrões de items”, consiste em 6 items, possui uma agregação configurada para cada item e é representado por uma cor personalizada.
- O segundo conjunto de dados também possui um rótulo de conjunto de dados personalizado.
- Os dados no gráfico de pizza são exibidos como um gráfico de rosca com uma largura personalizada e valor total com unidades no centro.
- Os dados no gráfico de pizza são exibidos e agregados para um período de tempo personalizado das últimas 3 horas.
- A legenda do gráfico de pizza exibe os items configurados em 4 linhas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
                        "y": 0,
                        "width": 24,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
```

```
                {
                    "type": 0,
                    "name": "ds.0.type.0",
                    "value": 0
                },
                {
                    "type": 4,
                    "name": "ds.0.itemids.1",
                    "value": 23269
                },
                {
                    "type": 1,
                    "name": "ds.0.color.1",
                    "value": "BF00FF"
                },
                {
                  "type": 0,
                  "name": "ds.0.type.1",
                  "value": 0
                },
                {
                    "type": 4,
                    "name": "ds.0.itemids.2",
                    "value": 23257
                },
                {
                    "type": 1,
                    "name": "ds.0.color.2",
                    "value": "0040FF"
                },
                {
                  "type": 0,
                  "name": "ds.0.type.2",
                  "value": 0
                },
                {
                    "type": 1,
                    "name": "ds.1.hosts.0",
                    "value": "Zabbix server"
                },
                {
                    "type": 1,
                    "name": "ds.1.items.0",
                    "value": "*: Number of processed *values per second"
                },
                {
                    "type": 1,
                    "name": "ds.1.color",
                    "value": "000000"
                },
                {
                    "type": 0,
                    "name": "ds.1.aggregate_function",
                    "value": 3
                },
                {
                    "type": 1,
                    "name": "ds.1.data_set_label",
                    "value": "Number of processed values per second"
                },
                {
                    "type": 0,
```

```
                                "name": "draw_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "width",
                                "value": 30
                            },
                            {
                                "type": 0,
                                "name": "total_show",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "units_show",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "graph_time",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 1,
                                "name": "time_period.to",
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


Resposta:


```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

22 Hosts com problemas

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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.
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.

Parâmetro	tipo	nome	valor
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor 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.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Parâmetro	tipo	nome	valor
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Problems* 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 integrados 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 *Problems*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com o widget *Problemas*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Mostrar	0	show	1 - *(padrão)* Problemas recentes; 2 - Histórico; 3 - Problemas.
Grupos de hosts	2	groupids.0	ID do grupo de hosts. Observação: para configurar vários grupos de hosts, crie um objeto de campo do widget do dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível quando o widget é 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 o parâmetro *Referência* definido como "ABCDE") como fonte de dados para os grupos de hosts. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.
Excluir grupos de hosts	2	exclude_groupids.0	ID do grupo de hosts. Observação: para excluir vários grupos de hosts, crie um objeto de campo do widget do dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível quando o widget é configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Hosts	3	hostids.0	ID do host. Observação: para configurar vários hosts, crie um objeto de campo do widget do dashboard para cada host, usando um número incrementado no nome da propriedade. Para vários 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 compatível quando o widget é configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do host: `DASHBOARD.hostids` - defina o seletor de hosts do dashboard como fonte de dados para os hosts; `ABCDE._hostids` - defina um widget compatível (com o parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para os hosts. Este parâmetro não é compatível quando o widget é 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 - Média; 4 - Alta; 5 - Desastre. Padrão: vazio (todas habilitadas). Observação: para configurar vários valores, crie um objeto de campo do widget do dashboard para cada valor, usando 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. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags do problema*

Parâmetro	tipo	nome	valor
Valor da tag	1	tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *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: - *compatível* se *Mostrar tags* estiver definido como ”1”, ”2” ou ”3”
Prioridade de exibição das tags	1	tag_priority	Lista de tags separada por vírgulas. Comportamento do parâmetro: - *compatível* 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 confirmação	0	acknowledgement_status	0 - *(padrão)* Todos; 1 - Não confirmados; 2 - Confirmados.
Por mim	0	acknowledged_by_me	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Ordenar entradas por	0	sort_triggers	1 - Severidade (decrescente); 2 - Host (crescente); 3 - Hora (crescente); 4 - *(padrão)* Hora (decrescente); 13 - Severidade (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 estar definido como ”Desabilitado”. Os valores ”Host (crescente)” e ”Host (decrescente)” não são compatíveis quando o widget é configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Mostrar linha do tempo	0	show_timeline	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *compatível* se *Ordenar entradas por* estiver definido como ”Hora (decrescente)” ou ”Hora (crescente)”
Destacar linha in-teira	0	highlight_row	0 - *(padrão)* Desabilitado; 1 - Habilitado. Este parâmetro não é compatível com os temas de alto contraste.
Mostrar lin-has	0	show_lines	Os valores possíveis variam de 1 a 100. Padrão: 25.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Esse valor deve ser exclusivo no 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

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Problemas por gravidade* 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 integrados 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 gravidade*, 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.
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).

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.
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".
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.

Parâmetro	tipo	nome	valor
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 Gráfico de dispersão

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Gráfico de dispersão* 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 integrados 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 dispersão*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico de dispersão*.

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 em particular, enquanto o segundo número, se presente, representa o host ou item configurado.

Parâmetro	tipo	nome	valor
Tipo de conjunto de dados ***Conjunto de dados: Lista de items***	0	ds.0.dataset_type	0 - Lista de items; 1 - *(padrão)* Padrões de items.
Cor	1	ds.0.color.0	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items"
Eixo X	4	ds.0.x_axis_itemids.0	ID do Item. Ao configurar o widget em um dashboard de template, apenas os items configurados no template devem ser definidos. Nota: Para configurar múltiplos items, crie um objeto de campo de widget de dashboard para cada item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Eixo X (Widget)* não estiver definido

Parâmetro	tipo	nome	valor
Eixo X (Widget)	1	ds.0.x_axis_itemids.0	Referência do ID do Item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para 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 o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Eixo X* não estiver definido
Eixo Y	4	ds.0.y_axis_itemids.0	ID do Item. Ao configurar o widget em um dashboard de template, apenas os items configurados no template devem ser definidos. Nota: Para configurar múltiplos items, crie um objeto de campo de widget de dashboard para cada item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Eixo Y (Widget)* não estiver definido
Eixo Y (Widget)	1	ds.0.y_axis_itemids.0	Referência do ID do Item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para 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 o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Eixo Y* não estiver definido
Sobrescrever host	1	ds.0.override_hostid.0	`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.
Conjunto de dados: Padrões de items			
Cor	1	ds.0.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *suportado* se o *Tipo de conjunto de dados* estiver definido como "Padrões de items" e *Paleta de cores* não estiver definida

Parâmetro	tipo	nome	valor
Paleta de cores	0	ds.0.color_palette	Índice da paleta de cores. Valores possíveis: 0-11. Padrão: 0. Comportamento do parâmetro: - *suportado* se o *Tipo de conjunto de dados* estiver definido como ”Padrões de items” e *Cor* não estiver definida
Padrões de host	1	ds.0.hosts.0	Nome ou padrão do Host (por exemplo, ”Zabbix*”). Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como ”Padrões de items” Este parâmetro não é suportado ao configurar o widget em um dashboard de template.
Padrões de item do eixo X	1	ds.0.x_axis_items.0	Nome ou padrão do Item (por exemplo, ”*: Número de valores processados por segundo”). Ao configurar o widget em um dashboard de template, apenas os padrões para items configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como ”Padrões de items”
Padrões de item do eixo Y	1	ds.0.y_axis_items.0	Nome ou padrão do Item (por exemplo, ”*: Número de valores processados por segundo”). Ao configurar o widget em um dashboard de template, apenas os padrões para items configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como ”Padrões de items”
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 ao configurar 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 ao configurar o widget em um dashboard de template.
Sobrescrever host	1	ds.0.override_hostid._reference	`ABCDE._hostid` - defina um widget compatível (com seu parâmetro ”Referência” definido como ”ABCDE”) como a fonte de dados para hosts; `DASHBOARD._hostid` - defina o seletor de host do dashboard como a fonte de dados para hosts. Este parâmetro não é suportado ao configurar o widget em um dashboard de template.
Tags de host			

Parâmetro	tipo	nome	valor
Tipo de avaliação	0	evaltype_host	0 - *(padrão)* E/OU; 2 - OU. Este parâmetro não é suportado ao configurar 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 ao configurar 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 ao configurar 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 ao configurar o widget em um dashboard de template.
Marcador	0	ds.0.marker	0 - *(padrão)* Elipse; 1 - Quadrado; 2 - Triângulo; 3 - Losango; 4 - Estrela; 5 - Cruz.
Tamanho do marcador	0	ds.0.marker_size	0 - *(padrão)* Pequeno (6px); 1 - Médio (9px); 2 - Grande (15px).
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).
Intervalo de agregação	1	ds.0.aggregate_interval	String de tempo válida (por exemplo, 3600, 1h, etc.). Você pode usar sufixos de tempo. Padrão: 15m.

Parâmetro	tipo	nome	valor
Função de agregação	0	ds.0.aggregate_function	0 - não usado; 1 - min; 2 - max; 3 - *(padrão)* avg; 4 - count; 5 - sum; 6 - first; 7 - last.

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.
Nomes dos hosts nos rótulos	0	show_hostnames	0 - *(padrão)* Automático; 1 - Mostrar; 2 - Ocultar. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.

Período de tempo

Os seguintes parâmetros são suportados para configurar o *Período de tempo*.

Parâmetro	tipo	nome	valor
Período de tempo	1	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 *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 - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

Eixos

Os seguintes parâmetros são suportados para configurar *Eixos*.

Parâmetro	tipo	nome	valor
Eixo X	0	x_axis	0 - Desabilitado; 1 - *(padrão)* Habilitado.

Parâmetro	tipo	nome	valor
Eixo Y	0	y_axis	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mín	1	x_axis_min y_axis_min	Qualquer valor numérico. Padrão: "" (vazio).
Máx	1	x_axis_max y_axis_max	Qualquer valor numérico. Padrão: "" (vazio).
Unidades (tipo)	0	x_axis_units y_axis_units	0 - *(padrão)* Automático; 1 - Estático.
Unidades (valor)	1	x_axis_static_units y_axis_static_units	Qualquer valor de string. Padrão: "" (vazio).

Legenda

Os seguintes parâmetros são suportados para configurar a *Legenda*.

Parâmetro	tipo	nome	valor
Exibir legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Exibir função de agregação	0	legend_aggregation	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Linhas	0	legend_lines_mode	0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Número de linhas/ Número máximo de linhas	0	legend_lines	Faixa de valores possíveis: 1-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Número de colunas	0	legend_columns	Faixa de valores possíveis: 1-4. Padrão: 4. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"

Limiares

Os seguintes parâmetros são suportados para configurar *Limiares*.

Note:
O número no nome da propriedade *Limiares* (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
Interpolação de cor	0	interpolation	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Eixo X	1	thresholds.0.x_axis_threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados.
Eixo Y	1	thresholds.0.y_axis_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 *Gráfico de dispersão*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Scatter plot*

Configure um widget *Scatter plot* da seguinte forma:

- 5 conjuntos de dados do tipo ”Padrões de item”, cada um com agregação padrão (avg, 15m) e marcadores diferentes.
- A legenda exibe os itens configurados em uma única linha e duas colunas.
- Os thresholds são configurados para colorir um marcador de vermelho quando seu valor atingir 80 em qualquer eixo.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "scatterplot",
                        "name": "Scatter plot",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "ds.0.color",
                                "value": "0C5A87"
                            },
                            {
                                "type": 1,
                                "name": "ds.0.hosts.0",
                                "value": "de-frankfurt*"
                            },
                            {
                                "type": 1,
                                "name": "ds.0.x_axis_items.0",
                                "value": "Memory utilization"
                            },
                            {
                                "type": 1,
                                "name": "ds.0.y_axis_items.0",
                                "value": "CPU utilization"
                            },
                            {
                                "type": 0,
```

```
            "name": "ds.0.marker",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.0.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "66B0D9"
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "fr-paris*"
        },
        {
            "type": 1,
            "name": "ds.1.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.1.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.1.marker",
            "value": 1
        },
        {
            "type": 0,
            "name": "ds.1.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.2.color",
            "value": "0A466A"
        },
        {
            "type": 1,
            "name": "ds.2.hosts.0",
            "value": "lv-riga*"
        },
        {
            "type": 1,
            "name": "ds.2.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.2.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.2.marker",
            "value": 2
```

```
        },
        {
            "type": 0,
            "name": "ds.2.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.3.color",
            "value": "3394C3"
        },
        {
            "type": 1,
            "name": "ds.3.hosts.0",
            "value": "pl-warsaw*"
        },
        {
            "type": 1,
            "name": "ds.3.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.3.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.3.marker",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.3.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.4.color",
            "value": "1492C8"
        },
        {
            "type": 1,
            "name": "ds.4.hosts.0",
            "value": "se-stockholm*"
        },
        {
            "type": 1,
            "name": "ds.4.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.4.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.4.marker",
            "value": 4
        },
        {
```

```
                            "type": 0,
                            "name": "ds.4.marker_size",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "legend_columns",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "interpolation",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.color",
                            "value": "D40000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.x_axis_threshold",
                            "value": "80"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.y_axis_threshold",
                            "value": ""
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "D40000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.x_axis_threshold",
                            "value": ""
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.y_axis_threshold",
                            "value": "80"
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "JQISY"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
```

```
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

26 Relatório de SLA

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são compatíveis com 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*
Service	9	serviceid.0	ID do Service.
Mostrar períodos	0	show_periods	Os valores possíveis variam de 1 a 100. Padrão: 20.
De	1	date_from	String de data válida no formato `YYYY-MM-DD`. Datas relativas com os modificadores `d`, `w`, `M`, `y` (por exemplo, `now`, `now/d`, `now/w-1w` etc.) são compatíveis.

Parâmetro	tipo	nome	valor
Até	1	date_to	String de data válida no formato `YYYY-MM-DD`. Datas relativas com os modificadores `d`, `w`, `M`, `y` (por exemplo, `now`, `now/d`, `now/w-1w` etc.) são compatíveis.

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`

27 Informações do sistema

Descrição

Estes 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 integrados 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 *System Information*.

Parameter	type	name	value
Refresh interval	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.
Show	0	info_type	0 - *(padrão)* Estatísticas do sistema; 1 - Nós de alta disponibilidade.

Parameter	type	name	value
Show software update check details	0	show_software_update_check_details	0 - *(padrão)* Desativado; 1 - Ativado. Parameter behavior: - *supported* se `AllowSoftwareUpdateCheck` na configuração do Zabbix server estiver habilitado e *Show* estiver definido como ”System stats”

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`

28 Principais hosts

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 compatíveis com o widget *Top hosts*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do grupo de hosts. Observação: para configurar vários grupos de hosts, crie um objeto de campo de widget do dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.

Parâmetro	type	name	value
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com o parâmetro *Reference* definido como "`ABCDE`") como fonte de dados para grupos de hosts. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Hosts	3	hostids.0	ID do host. Observação: para configurar vários hosts, crie um objeto de campo de widget do dashboard para cada host, usando um número incrementado no nome da propriedade. Para vários 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 compatível ao configurar 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 hosts do dashboard como fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com o parâmetro *Reference* definido como "`ABCDE`") como fonte de dados para hosts. Este parâmetro não é compatível ao configurar 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 compatível ao configurar o widget em um dashboard de template.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Operador	0	tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar o widget em um dashboard de template.

Parâmetro	type	name	value
Valor da tag	1	tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags de host* Este parâmetro não é compatível ao configurar 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 entre as colunas configuradas.
Ordem	0	order	2 - *(padrão)* N primeiros; 3 - N últimos.
Limite de hosts	0	show_lines	Os valores possíveis variam de 1 a 1000. Padrão: 10. Este parâmetro não é compatível ao configurar 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*
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 compatíveis quando *Dados* está definido como "Valor do item".

Note:
O primeiro número no nome da propriedade *Limites* (por exemplo, columnsthresholds.0.color.0) faz referência à coluna para a qual os limites estão configurados, enquanto o segundo número faz referência à posição do limite em uma lista ordenada em ordem crescente. No entanto, se os limites forem configurados em uma ordem diferente, os valores serão ordenados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Item	1	columns.0.item	Nome de item válido. Ao configurar o widget em um dashboard de template, somente 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.
Exibição	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores; 6 - Sparkline. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico”
Mín.	1	columns.0.min	Qualquer valor numérico. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Barra” ou ”Indicadores”
Máx.	1	columns.0.max	Qualquer valor numérico. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Barra” ou ”Indicadores”
Casas decimais	0	columns.0.decimal_places	Os valores possíveis variam de 0 a 10. Padrão: 2. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico”
Sparkline			
Largura	0	columns.0.sparkline.width	Os valores possíveis variam de 0 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Sparkline”
Preenchimento	0	columns.0.sparkline.fill	Os valores possíveis variam de 0 a 10. Padrão: 3. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Sparkline”

Parâmetro	tipo	nome	valor
Cor	1	columns.0.sparkline.color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `42A5F5`. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibição* estiver definida 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 o parâmetro `reference` igual a `ABCDE`) como fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período de tempo somente nos parâmetros *De* e *Até*. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibição* estiver definida como "Sparkline"
De	1	columns.0.sparkline.time_period.from	String de tempo válida na sintaxe de tempo absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibição* estiver definida como "Sparkline" - *obrigatório* se `columns.0.sparkline.time_period.to` estiver definido
Até	1	columns.0.sparkline.time_period.to	String de tempo válida na sintaxe de tempo absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibição* estiver definida como "Sparkline" - *obrigatório* se `columns.0.sparkline.time_period.from` estiver definido
Dados históricos	0	columns.0.sparkline.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico" e *Exibição* estiver definida como "Sparkline"
Limites			
Cor	1	columnsthresholds.0.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico"
Limite	1	columnsthresholds.0.threshold	Qualquer valor de string. Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Numérico"
Destaques			
Cor	1	columns.0.highlights.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *compatível* se *Exibir valor do item como* estiver definido como "Texto"

Parâmetro	tipo	nome	valor
Padrão	1	columns.0.highlights.0.pattern	Qualquer valor de string. Comportamento do parâmetro: - *compatível* 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: - *compatível* se *Exibir valor do item como* estiver definido como "Binário"
Função de agregação	0	columns.0.aggregate_function	0 - *(padrão)* não utilizada; 1 - mín.; 2 - máx.; 3 - média; 4 - contagem; 5 - soma; 6 - primeiro; 7 - último.
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 o parâmetro reference igual a ABCDE) como fonte de dados. Padrão: DASHBOARD._timeperiod Como alternativa, você pode definir o período de tempo somente nos parâmetros *De* e *Até*. Comportamento do parâmetro: - *compatível* se *Função de agregação* estiver definida como "mín.", "máx.", "média", "contagem", "soma", "primeiro" ou "último"
De	1	columns.0.time_period.from	String de tempo válida na sintaxe de tempo absoluta (YYYY-MM-DD hh:mm:ss) ou relativa (now, now/d, now/w-1w, etc.). Comportamento do parâmetro: - *compatível* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como "mín.", "máx.", "média", "contagem", "soma", "primeiro" ou "último" - *obrigatório* se columns.0.time_period.to estiver definido
Até	1	columns.0.time_period.to	String de tempo válida na sintaxe de tempo absoluta (YYYY-MM-DD hh:mm:ss) ou relativa (now, now/d, now/w-1w, etc.). Comportamento do parâmetro: - *compatível* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como "mín.", "máx.", "média", "contagem", "soma", "primeiro" ou "último" - *obrigatório* se columns.0.time_period.from estiver definido
Dados históricos	0	columns.0.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *compatível* 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). Esse valor deve ser exclusivo no dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Texto

Os seguintes parâmetros são suportados se *Data* estiver definido como "Text".

Parameter	type	name	value
Text	1	columns.0.text	Qualquer valor de string, incluindo macros. Macros suportadas: `{HOST.*}`, `{INVENTORY.*}`. Comportamento do parâmetro: - *obrigatório* se *Data* estiver definido como "Text"

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`

29 Principais items

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Top items* 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 integrados 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 items*, 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.
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.

Parâmetro	type	name	value
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.
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 compatíveis com o widget *Top items*.

Parameter	type	name	value
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”). Observação: para configurar vários padrões de item, crie um objeto de campo do widget do dashboard para cada padrão de item, usando um número incrementado no nome da propriedade. Ao configurar o widget em um dashboard de template, defina somente os padrões dos itens configurados no template. 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 a 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. Observação: para configurar vários padrões de host, crie um objeto de campo do widget do dashboard para cada padrão de host, usando 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 a 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 de item, tag).

Parâmetro	tipo	nome	valor
Padrões de item	1	columns.0.items.0	Nome ou padrão de item (por exemplo, "*: Número de valores processados por segundo"). Observação: para configurar vários padrões de item, crie um objeto de campo de widget do dashboard para cada padrão de item, incrementando o segundo número no nome da propriedade. Ao configurar o widget em um dashboard de template, defina somente os padrões dos itens configurados no template. Comportamento do parâmetro: - *obrigatório*
Tags de 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. Observação: o segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se *Tags de item* estiver sendo configurado
Operador	0	columns.0.item_tags.0.operator	**0** - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se *Tags de item* estiver sendo configurado
Valor da tag	1	columns.0.item_tags.0.value	Qualquer valor de string. Observação: o segundo número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se *Tags de item* estiver sendo configurado
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.
Exibição	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores; 6 - Sparkline.
Mín.	1	columns.0.min	Qualquer valor numérico. Sufixos (por exemplo, "1d", "2w", "4K", "8G") são compatíveis. Comportamento do parâmetro: - *compatível* se *Exibir valor como* estiver definido como "Numérico" e *Exibição* estiver definida como "Barra" ou "Indicadores"

Parâmetro	tipo	nome	valor
Máx.	1	columns.0.max	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são compatíveis. Comportamento do parâmetro: - *compatível* se *Exibir valor como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Barra” ou ”Indicadores”
Sparkline			
Largura	0	columns.0.sparkline.width	Os valores possíveis variam de 0 a 10. Padrão: 1. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como ”Sparkline”
Preenchimento	0	columns.0.sparkline.fill	Os valores possíveis variam de 0 a 10. Padrão: 3. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como ”Sparkline”
Cor	1	columns.0.sparkline.color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `42A5F5`. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como ”Sparkline”
Período	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - define o seletor de período do dashboard como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o parâmetro `reference` igual a `ABCDE`) como fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período somente nos parâmetros *De* e *Até*. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como ”Sparkline”
De	1	columns.0.sparkline.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: - *compatível* se *Exibição* estiver definida como ”Sparkline” - *obrigatório* se `columns.0.sparkline.time_period.to` estiver definido
Até	1	columns.0.sparkline.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: - *compatível* se *Exibição* estiver definida como ”Sparkline” - *obrigatório* se `columns.0.sparkline.time_period.from` estiver definido
Dados históricos	0	columns.0.sparkline.history	0 *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *compatível* se *Exibição* estiver definida como ”Sparkline”
Limiares			

Parâmetro	tipo	nome	valor
Cor	1	columns.0.thresholds.color	Código de cor hexadecimal (por exemplo, `FF0000`). Comportamento do parâmetro: - *compatível* 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 compatíveis. Comportamento do parâmetro: - *compatível* 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: - *compatível* se *Exibir valor como* estiver definido como "Texto"
Limiar	1	columns.0.highlights.pattern	Qualquer expressão regular. Comportamento do parâmetro: - *compatível* se *Exibir valor como* estiver definido como "Texto"
Casas decimais	0	columns.0.decimal_places	Os valores possíveis variam de 0 a 10. Padrão: 2.
Configuração avançada			
Função de agregação	0	columns.0.aggregate_function	Se *Exibir valor como* estiver definido como "Numérico": 0 - *(padrão)* não utilizada; 1 - mín.; 2 - máx.; 3 - média; 4 - contagem; 5 - soma; 6 - primeiro; 7 - último. Se *Exibir valor como* estiver definido como "Texto": 0 - *(padrão)* não utilizada; 4 - contagem; 6 - primeiro; 7 - último.
Período	1	columns.0.time_period.from	`DASHBOARD._timeperiod` - define o seletor de período do dashboard como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o parâmetro `reference` igual a `ABCDE`) como fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período somente nos parâmetros *De* e *Até*. Comportamento do parâmetro: - *compatível* se *Função de agregação* estiver definida como "mín.", "máx.", "média", "contagem", "soma", "primeiro" ou "último"
De	1	columns.0.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: - *compatível* se *Período* não estiver definido e *Função de agregação* estiver definida como "mín.", "máx.", "média", "contagem", "soma", "primeiro" ou "último" - *obrigatório* se `columns.0.time_period.to` estiver definido

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 na sintaxe de tempo relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido e *Função de agregação* estiver definida como ”mín.”, ”máx.”, ”média”, ”contagem”, ”soma”, ”primeiro” ou ”último” - *obrigatório* se `columns.0.time_period.from` estiver definido
Dados históricos	0	columns.0.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências. Comportamento do parâmetro: - *compatível* se *Exibir valor como* estiver definido como ”Numérico”
Agregação	0	columns.0.aggregate_function	0 - *(padrão)* Cada item 1 - Combinado. Comportamento do parâmetro: - *compatível* se *Exibir valor como* estiver definido como ”Numérico” e *Exibição* estiver definida como ”Como está”, ”Barra” ou ”Indicadores”.
Função de agregação combinada	0	columns.0.column_aggregate_function	1 - mín.; 2 - máx.; 3 - média; 4 - contagem; 5 - soma *(padrão)*. Comportamento do parâmetro: - *obrigatório* ao configurar *Agregação*.
Nome da coluna combinada	1	columns.0.combined_column_name	Qualquer valor de string. Comportamento do parâmetro: - *obrigatório* ao configurar *Agregação*.

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`

30 principais triggers

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Top triggers* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets integrados 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 compatíveis com o widget *Principais 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. Observação: para configurar vários grupos de hosts, crie um objeto de campo de widget do dashboard para cada grupo de hosts, usando um número incrementado no nome da propriedade. Este parâmetro não é compatível ao configurar o widget em um dashboard de template.
Hosts	3	hostids.0	ID do host. Observação: para configurar vários hosts, crie um objeto de campo de widget do dashboard para cada host, usando um número incrementado no nome da propriedade. Para vários 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 compatível ao configurar o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
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 - Média; 4 - Alta; 5 - Desastre. Padrão: vazio (todas habilitadas). Observação: para configurar vários valores, crie um objeto de campo de widget do dashboard para cada valor, usando 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. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags do problema*
Valor da tag	1	tags.0.value	Qualquer valor de string. Observação: o número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* ao configurar *Tags do problema*
Período	1	time_period._reference	`DASHBOARD._timeperiod` - define o seletor de período do dashboard como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com o parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Como alternativa, você pode definir o período somente 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 relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na sintaxe de tempo relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *compatível* se *Período* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Limite de trig-gers	0	show_lines	Os valores possíveis variam de 1 a 100. Padrão: 10.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget 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 severidades ”Warning”, ”Average”, ”High” ou ”Disaster”, e problemas que possuem uma tag com o nome ”scope” que contenha os valores ”performance” ou ”availability” ou ”capacity”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
```

```
            "name": "severities.1",
            "value": 3
        },
        {
            "type": 0,
            "name": "severities.2",
            "value": 4
        },
        {
            "type": 0,
            "name": "severities.3",
            "value": 5
        },
        {
            "type": 1,
            "name": "tags.0.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.0.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.0.value",
            "value": "performance"
        },
        {
            "type": 1,
            "name": "tags.1.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.1.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.1.value",
            "value": "availability"
        },
        {
            "type": 1,
            "name": "tags.2.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.2.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.2.value",
            "value": "capacity"
        },
        {
            "type": 0,
            "name": "show_lines",
            "value": 5
```

```
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

31 Visão geral de triggers

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Visão geral de triggers*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Visão geral dos triggers*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Mostrar	0	show	1 - *(padrão)* Problemas recentes; 2 - Qualquer; 3 - Problemas.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
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*

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.
Valor da tag	1	tags.0.value	Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema* Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags.
Mostrar problemas suprimidos	0	show_suppressed	Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema* 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`

32 URL

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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 dos parâmetros 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._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 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`

33 Monitoramento web

Descrição

Estes parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do 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 integrados 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.
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*

Parâmetro	tipo	nome	valor
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de 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 - cria um novo diretório de usuários
- userdirectory.delete - exclui um diretório de usuários
- userdirectory.get - recupera um diretório de usuários
- userdirectory.update - atualiza um diretório de usuários
- userdirectory.test - testa a 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 tem 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 User 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 do protocolo de autenticação usado pelo provedor de identidade para o diretório de usuários. Observe que somente um diretório de usuários do tipo SAML pode existir. 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 Authentication 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 ”O provisionamento SCIM está habilitado”) que contém o nome do usuário, usado como valor da propriedade `name` do objeto User quando o usuário é provisionado. Exemplos: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Atributo do diretório de usuários LDAP/SAML (também atributo SCIM se `scim_status` estiver definido como ”O provisionamento SCIM está habilitado”) que contém o sobrenome do usuário, usado como valor da propriedade `surname` do objeto User 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; além disso, o status do provisionamento LDAP ou SAML (`ldap_jit_status` ou `saml_jit_status` do objeto Authentication) deve estar habilitado).
provision_groups	array	Array de objetos de mapeamentos de grupos de provisionamento para mapear o padrão do grupo de usuários LDAP/SAML para o grupo de usuários e a função de usuário do Zabbix. Comportamento da propriedade: - *obrigatório* se `provision_status` estiver definido como ”Habilitado”
provision_media	array	Array de objetos de mapeamentos de tipos de mídia para mapear os atributos de mídia LDAP/SAML do usuário (por exemplo, e-mail) para a mídia do usuário do Zabbix, usada para enviar notificações.

Propriedade	Tipo	Descrição
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 server LDAP. A URI deve conter o esquema (`ldap://` ou `ldaps://`), o host e a 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 server 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 as 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 do usuário; o espaço reservado "*%{user}*" é obrigatório) Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
search_attribute	string	Atributo do diretório de usuários LDAP usado para identificar a conta de usuário com base nas 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 server LDAP usada para vinculação e pesquisa no server LDAP. Para vinculação direta do 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: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"

Propriedade	Tipo	Descrição
bind_password	string	Senha LDAP da conta usada para vinculação e pesquisa no server LDAP. Para vinculação direta do usuário e vinculação anônima, `bind_password` deve estar vazio. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
description	string	Descrição do diretório de usuários. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
group_basedn	string	Caminho base do diretório de usuários LDAP para os 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: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
group_filter	string	String de filtro para recuperar os grupos do diretório de usuários LDAP dos quais o usuário é membro; usada para configurar uma verificação de associação de usuário no diretório de usuários LDAP. Ignorada ao provisionar um usuário se `group_membership` estiver definido. Espaços reservados `group_filter` compatíveis: *%{attr}* - atributo de pesquisa (substituído pelo valor da propriedade `search_attribute`); *%{groupattr}* - atributo do grupo (substituído pelo valor da propriedade `group_member`); *%{host}* - nome do host, IP ou URI do server 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: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”

Propriedade	Tipo	Descrição
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: - *compatível* 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: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
search_filter	string	String de filtro personalizada usada para localizar e autenticar um usuário em um diretório de usuários LDAP com base nas informações fornecidas na solicitação de login. Espaços reservados `search_filter` compatíveis: *%{attr}* - nome do atributo de pesquisa (por exemplo, *uid*, *sAMAccountName*); *%{user}* - nome de usuário do Zabbix. Padrão: *(%{attr}=%{user})* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
start_tls	integer	Opção de configuração do server LDAP que permite proteger a comunicação com o server LDAP usando Transport Layer Security (TLS). Observe que `start_tls` deve estar definido como "Desabilitado" para hosts que usam o protocolo `ldaps://`. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento da propriedade: - *compatível* 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 inseri-los no lugar do espaço reservado *%{ref}* na string `group_filter`. Exemplos: *cn*, *uid*, *member*, *uniqueMember* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
Propriedades específicas do SAML:		

Propriedade	Tipo	Descrição
idp_entityid	string	URI que identifica o provedor de identidade e é usada para se comunicar com o provedor de identidade nas 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 identifique 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"
username_attribute	string	Atributo do diretório de usuários SAML (também atributo SCIM se `scim_status` estiver definido como "O provisionamento SCIM está habilitado") que contém o nome de usuário, comparado com o valor da propriedade `username` do objeto User durante a autenticação. 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 logoff único SAML do provedor de identidade, para o qual o Zabbix enviará as solicitações de logoff SAML. Exemplo: *https://idp.example.com/idp/slo/saml* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
encrypt_nameid	integer	Indica se o ID de nome SAML deve ser criptografado. Valores possíveis: 0 - *(padrão)* Não criptografar o ID de nome; 1 - Criptografar o ID de nome. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"

Propriedade	Tipo	Descrição
encrypt_assertions	integer	Indica se as asserções SAML devem ser criptografadas. Valores possíveis: 0 - *(padrão)* Não criptografar as asserções; 1 - Criptografar as asserções. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
nameid_format	string	Formato do ID de nome do provedor de serviços do provedor de identidade SAML. Exemplos: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
scim_status	integer	Indica se o provisionamento SCIM para SAML está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* O provisionamento SCIM está desabilitado; 1 - O provisionamento SCIM está habilitado. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
sign_assertions	integer	Indica se as asserções SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - Não assinar as asserções; 1 - *(padrão)* Assinar as asserções. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
sign_authn_requests	integer	Indica se as solicitações AuthN SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - Não assinar as solicitações AuthN; 1 - *(padrão)* Assinar as solicitações AuthN. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
sign_messages	integer	Indica se as mensagens SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - Não assinar as mensagens; 1 - *(padrão)* Assinar as mensagens. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"

Propriedade	Tipo	Descrição
sign_logout_requests	integer	Indica se as solicitações de logoff SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - Não assinar as solicitações de logoff; 1 - *(padrão)* Assinar as solicitações de logoff. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_logout_responses	integer	Indica se as respostas de logoff SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - Não assinar as respostas de logoff; 1 - *(padrão)* Assinar as respostas de logoff. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
idp_certificate	string	Conteúdo do certificado do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML. Exemplos: *-----BEGIN CERTIFICATE----- ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML” e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente gravação*
sp_private_key	string	Conteúdo da chave privada do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML. Fornece autenticação segura e troca de dados com o provedor de identidade (IdP). Exemplos: *-----BEGIN CERTIFICATE----- ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML” e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente gravação*
sp_certificate	string	Conteúdo do certificado do provedor de serviços (SP) para a configuração do serviço de logon único (SSO) SAML. Exemplos: *-----BEGIN CERTIFICATE----- ...mais dados codificados... -----END CERTIFICATE-----* Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML” e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente gravação*

Propriedade	Tipo	Descrição
idp_certificate_hash	string	O hash md5 do valor de idp_certificate. Retornará uma string vazia para um idp_certificate vazio. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente leitura*
sp_private_key_hash	string	O hash md5 do valor de sp_private_key. Retornará uma string vazia para um sp_private_key vazio. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente leitura*
sp_certificate_hash	string	O hash md5 do valor de sp_certificate. Retornará uma string vazia para um sp_certificate vazio. Comportamento da propriedade: - *compatível* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML" e `$SSO['CERT_STORAGE']` estiver definido como `database` no arquivo de configuração do frontend - *somente leitura*

Mapeamentos de tipo de mídia

O objeto de mapeamentos de tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
userdirectory_mediaid	ID	ID do mapeamento de tipo de mídia. Comportamento da propriedade: - *somente leitura*
name	string	Nome visível na lista de mapeamentos de tipo de mídia. Comportamento da propriedade: - *obrigatório*
mediatypeid	ID	ID do tipo de mídia a ser criado; usado como valor para a propriedade `mediatypeid` do objeto Media. Comportamento da propriedade: - *obrigatório*
attribute	string	Atributo do diretório de usuários LDAP/SAML (também atributo SCIM se `scim_status` estiver definido como "Provisionamento SCIM está ativado") que contém a mídia do usuário (por exemplo, *user@example.com*) que é usada como valor para a propriedade `sendto` do objeto Media. Se presente nos dados recebidos do provedor de identidade LDAP/SAML, e o valor não estiver vazio, isso acionará a criação da mídia para o usuário provisionado. Comportamento da propriedade: - *obrigatório*
active	integer	Valor da propriedade `active` da mídia do usuário quando a mídia é criada para o usuário provisionado. Valores possíveis: 0 - *(padrão)* ativado; 1 - desativado.

Propriedade	Tipo	Descrição
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 "SCIM provisioning is enabled"). 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. Um diretório de usuários não pode ser excluído quando é usado diretamente por pelo menos um grupo de usuários.
 O diretório de usuários LDAP padrão não pode ser excluído quando `authentication.ldap_configured` estiver definido como 1 ou quando restarem 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 usuário *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.

Parâmetro	Tipo	Descrição
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`.
search	object	Retorna resultados que correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma pesquisa `LIKE "%...%"`. Propriedades suportadas: `name`, `description`. 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 e mídias de usuário serão atribuídos ao usuário). Para esse tipo de teste, a solicitação da API deve ser feita para um user directory que tenha `provision_status` definido como enabled.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*.

Parâmetros

`(object)` Propriedades do diretório de usuários.

Como a API `userdirectory.get` não retorna o campo `bind_password`, `userdirectoryid` e/ou `bind_password` devem ser fornecidos.
 Além das propriedades padrão do diretório de usuários, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
test_username	string	Nome de usuário para testar no diretório de usuários.
test_password	string	Senha associada ao nome de usuário para testar no diretório de usuários.

Valores de retorno

`(bool)` Retorna verdadeiro em caso de sucesso.

Exemplos

Testar 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 usuário 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*.

Dispositivo

Esta classe foi projetada para funcionar com dispositivos móveis usados com o aplicativo móvel do Zabbix.

Referências de objetos:

- Device

Métodos disponíveis:

- device.get - recuperar dispositivos
- device.offboard - remover dispositivos (retirá-los do gerenciamento)

Objeto dispositivo

O objeto dispositivo representa um dispositivo móvel para uso com o aplicativo móvel do Zabbix.

Propriedade	Tipo	Descrição
deviceid	ID	ID do dispositivo (interno, por exemplo, 1, 2). Comportamento da propriedade: - Somente leitura
userid	ID	ID do usuário do dispositivo. Comportamento da propriedade: - Constante
uuid	string	Identificador universalmente exclusivo do dispositivo (UUIDv7) (externo, por exemplo, `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Comportamento da propriedade: - Somente leitura
name	string	Nome do dispositivo. Comportamento da propriedade: - Somente leitura
status	integer	Status do dispositivo. Valores possíveis: 0 - Novo 1 - Ativo 2 - Não atribuído Comportamento da propriedade: - Somente leitura
activated_at	timestamp	Data e hora em que o dispositivo foi adicionado ao aplicativo móvel, ou 0 se ainda não tiver sido adicionado. Comportamento da propriedade: - Somente leitura
lastaccess	timestamp	Data e hora em que o dispositivo foi autenticado pela última vez no aplicativo móvel, ou 0 se ainda não tiver sido autenticado. Comportamento da propriedade: - Somente leitura

device.get

Este método permite recuperar dispositivos móveis para uso com o aplicativo móvel do Zabbix.

O método está disponível para usuários *User*, *Admin* e *Super admin* se `EnableMobileDevices` estiver habilitado e o usuário tiver acesso aos dispositivos. O acesso aos métodos da API pode ser revogado na configuração da função de usuário.

Usuários *User* e *Admin* só podem recuperar seus próprios dispositivos.

Assinatura do método: `integer/array device.get(object parameters)`

Origem do método: `CDevice::get()` em `ui/include/classes/api/services/CDevice.php`

Parâmetros

Parâmetro	Tipo	Descrição
deviceids	ID/array	Retorna somente os dispositivos com os IDs de dispositivo fornecidos.
userids	ID/array	Retorna somente os dispositivos pertencentes aos usuários com os IDs de usuário fornecidos.
roleids	ID/array	Retorna somente os dispositivos pertencentes aos usuários com os IDs de função de usuário fornecidos.
usrgrpids	ID/array	Retorna somente os dispositivos pertencentes aos usuários com os IDs de grupo de usuários fornecidos.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `deviceid`, `name`.
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

Retorna uma matriz de objetos ou a quantidade de objetos recuperados (se `countOutput` for usado).

Exemplos

Recuperar dispositivos

Recupere os dispositivos pertencentes aos usuários com estes IDs de usuário: 1, 2, 3.
 Observe que o usuário 2 (convidado) não possui dispositivos, e o dispositivo 3 ainda não foi adicionado nem autenticado no aplicativo móvel do Zabbix.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
```

```
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```


device.offboard

Este método permite remover dispositivos móveis para que deixem de ser usados com o aplicativo móvel do Zabbix.

O método está disponível para usuários *User*, *Admin* e *Super admin* se `EnableMobileDevices` estiver habilitado e o usuário tiver acesso aos dispositivos. O acesso aos métodos da API pode ser revogado na configuração da função de usuário.

Usuários *User* e *Admin* só podem remover seus próprios dispositivos.

Assinatura do método: `object device.offboard(object parameters)`

Origem do método: `CDevice::offboard()` em `ui/include/classes/api/services/CDevice.php`

Parâmetros

Parâmetro	Tipo	Descrição
uuid	string	Identificador universalmente exclusivo do dispositivo (UUIDv7). Corresponde a `deviceid`.
force	boolean	Defina como `true` para ignorar o erro "Dispositivo não encontrado" e remover o dispositivo à força. Padrão: `false`

Valores de retorno

Retorna um objeto contendo o UUID do dispositivo removido.

Exemplos

Remover dispositivo

Remove um dispositivo com o UUID especificado, ignorando o erro "Dispositivo não encontrado".

Solicitação:


```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

Event

Esta classe foi projetada para trabalhar com eventos.

Referências de objetos:

- Event
- Event tag
- Media type URL

Métodos disponíveis:

- event.get - recuperar eventos
- event.acknowledge - reconhecer eventos

Objeto event

Os seguintes objetos estão diretamente relacionados à API `event`.

Evento

> **Note:**
> Os eventos são criados pelo Zabbix server e não podem ser modificados via API.

O objeto evento possui as seguintes propriedades.

Propriedade	Tipo	Descrição
eventid	ID	ID do evento.
source	integer	Tipo do evento. Valores possíveis: 0 - evento criado por um trigger; 1 - evento criado por uma regra de descoberta; 2 - evento criado por autoregistro de agent ativo; 3 - evento interno; 4 - evento criado na atualização do status do serviço.

Propriedade	Tipo	Descrição
object	integer	Tipo de objeto relacionado ao evento. Valores possíveis se `source` estiver definido como ”evento criado por um trigger”: 0 - trigger. Valores possíveis se `source` estiver definido como ”evento criado por uma regra de descoberta”: 1 - host descoberto; 2 - serviço descoberto. Valores possíveis se `source` estiver definido como ”evento criado por autoregistro de agent ativo”: 3 - host autoregistrado. Valores possíveis se `source` estiver definido como ”evento interno”: 0 - trigger; 4 - item; 5 - regra LLD. Valores possíveis se `source` estiver definido como ”evento criado na atualização do status do serviço”: 6 - serviço.
objectid	ID	ID do objeto relacionado.
acknowledged	integer	Se o evento foi reconhecido.
clock	timestamp	Hora em que o evento foi criado.
ns	integer	Nanosegundos em que o evento foi criado.
name	string	Nome do evento resolvido.
value	integer	Estado do objeto relacionado. Valores possíveis se `source` estiver definido como ”evento criado por um trigger” ou ”evento criado na atualização do status do serviço”: 0 - OK; 1 - problema. Valores possíveis se `source` estiver definido como ”evento criado por uma regra de descoberta”: 0 - host ou serviço ativo; 1 - host ou serviço inativo; 2 - host ou serviço descoberto; 3 - host ou serviço perdido. Valores possíveis se `source` estiver definido como ”evento interno”: 0 - estado ”normal”; 1 - estado ”desconhecido” ou ”não suportado”. Comportamento da propriedade: - *suportado* se `source` estiver definido como ”evento criado por um trigger”, ”evento criado por uma regra de descoberta”, ”evento interno” ou ”evento criado na atualização do status do serviço”
severity	integer	Gravidade atual do evento. Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 5 - desastre.
r_eventid	ID	ID do evento de recuperação.

Propriedade	Tipo	Descrição
c_eventid	ID	ID do evento que foi usado para sobrescrever (fechar) o evento atual sob a regra de correlação global. Veja `correlationid` para identificar a regra de correlação exata. Este parâmetro só é definido quando o evento é fechado por regra de correlação global.
cause_eventid	ID	ID do evento de causa.
correlationid	ID	ID da regra de correlação que gerou o fechamento do problema. Este parâmetro só é definido quando o evento é fechado por regra de correlação global.
userid	ID	ID do usuário que fechou o evento (se o evento foi fechado manualmente).
suppressed	integer	Se o evento está suprimido. Valores possíveis: 0 - evento está em estado normal; 1 - evento está suprimido.
opdata	string	Dados operacionais com macros expandidas.
urls	array	URLs de tipo de mídia ativas.

Tag de evento

O objeto de tag de evento possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de evento.
value	string	Valor da tag de evento.

URL do tipo de mídia

O objeto de URL do tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da URL definida 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 executadas:

- Fechar evento. Se o evento já estiver resolvido, esta ação será ignorada.
- Reconhecer evento. Se o evento já estiver reconhecido, esta ação será ignorada.
- Desreconhecer 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.
- Dessuprimir evento. Se o evento não estiver suprimido, esta ação será ignorada.
- Alterar a classificação do evento para causa. Se o evento já estiver classificado como causa, esta ação será ignorada.
- Alterar a classificação do evento para sintoma. Se o evento já estiver classificado como sintoma, esta ação será ignorada.

Attention:
Somente eventos de trigger podem ser atualizados.
 Somente eventos de problema podem ser atualizados.
 São necessários direitos de leitura/gravação para o trigger para fechar o evento ou alterar a severidade do evento.
 Para fechar um evento, o fechamento manual deve ser permitido no 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 executadas.

Parâmetro	Tipo	Descrição
eventids	ID/array	IDs dos eventos a serem reconhecidos. Comportamento do parâmetro: - *obrigatório*
action	integer	Ação(ões) de atualização do evento. Valores possíveis do bitmap: 1 - fechar o problema; 2 - reconhecer o evento; 4 - adicionar mensagem; 8 - alterar a severidade; 16 - remover o reconhecimento do evento; 32 - suprimir o evento; 64 - remover a supressão do evento; 128 - alterar a classificação do evento para causa; 256 - alterar a classificação do evento para sintoma. Este é um campo de máscara de bits; qualquer soma dos valores possíveis do bitmap é aceita (por exemplo, 6 para reconhecer um evento e deixar uma mensagem, ou 34 para reconhecer e suprimir um evento). Comportamento do parâmetro: - *obrigatório*
cause_eventid	ID	ID do evento de causa. Comportamento do parâmetro: - *obrigatório* se `action` contiver o bit ”alterar a 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 dos 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 a severidade”

Parâmetro	Tipo	Descrição
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 o 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
}
```

Suprimindo um evento indefinidamente

Suprima um único evento sem um tempo de expiração.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Dessuprimir um evento

Dessuprima um único evento previamente suprimido.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Reconhecer e suprimir um evento

Reconheça um ou mais eventos, adicione uma mensagem e suprima-os na mesma solicitação.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Suppressing during planned maintenance.",
        "suppress_until": 0
    },
    "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 informados.

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. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parameter	Type	Description
eventids	ID/array	Retorna apenas eventos com os IDs informados.
groupids	ID/array	Retorna apenas eventos criados por objetos que pertencem aos grupos de host informados.
hostids	ID/array	Retorna apenas eventos criados por objetos que pertencem aos hosts informados.
objectids	ID/array	Retorna apenas eventos criados pelos objetos informados.

Parameter	Type	Description
source	integer	Retorna apenas eventos com o tipo informado. Consulte a página do objeto event para obter uma lista dos tipos de evento suportados. Padrão: 0 - eventos de trigger.
object	integer	Retorna apenas eventos criados por objetos do tipo informado. Consulte a página do objeto event para obter uma lista dos tipos de objeto 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 informadas foram executadas. Para várias ações, use a soma de quaisquer valores de bitmap aceitáveis como bitmask (por exemplo, 34 para reconhecer e suprimir o evento).
action_userids	ID/array	Retorna apenas eventos com os IDs informados dos usuários que executaram as ações de atualização do evento.
suppressed	boolean	Se definido como `true`, retorna apenas eventos suprimidos.
symptom	boolean	Se definido como `true`, retorna apenas eventos de sintoma.
severities	integer/array	Retorna apenas eventos com as severidades informadas. Aplica-se somente se object for trigger.
trigger_severities	integer/array	Retorna apenas eventos com as severidades de trigger informadas. Aplica-se somente se object for trigger.
evaltype	integer	Método de avaliação de tags. Valores possíveis: 0 - *(padrão)* And/Or; 2 - Or.
tags	array	Retorna apenas eventos com as tags informadas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os eventos. Valores possíveis de operator: 0 - *(padrão)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
eventid_from	string	Retorna apenas eventos com IDs maiores ou iguais ao ID informado.
eventid_till	string	Retorna apenas eventos com IDs menores ou iguais ao ID informado.
time_from	timestamp	Retorna apenas eventos criados no momento informado ou depois dele.
time_till	timestamp	Retorna apenas eventos criados no momento informado ou antes dele.
problem_time_from	timestamp	Retorna apenas eventos que estiveram no estado de problema a partir de `problem_time_from`, independentemente do status atual. Aplica-se somente quando a source é um evento de trigger e o object é um trigger. Este parâmetro é obrigatório se `problem_time_till` for especificado.
problem_time_till	timestamp	Retorna apenas eventos que estiveram no estado de problema até `problem_time_till`, independentemente do status atual. Aplica-se somente quando a source é um evento de trigger e o object é um trigger. Este parâmetro é obrigatório se `problem_time_from` for especificado.
value	integer/array	Retorna apenas eventos com os valores informados.

Parameter	Type	Description
selectAcknowledges	query	Retorna uma propriedade `acknowledges` com as atualizações do evento. As atualizações do evento são ordenadas em ordem cronológica inversa. O objeto de atualização do evento tem as seguintes propriedades: `acknowledgeid` - `(ID)` ID do reconhecimento; `userid` - `(ID)` ID do usuário que atualizou o evento; `clock` - `(timestamp)` momento em que o evento foi atualizado; `message` - `(string)` texto da mensagem; `action` - `(integer)` ação de atualização executada, 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)` momento até o qual o evento será suprimido; `taskid` - `(ID)` ID da tarefa se o evento atual estiver passando por uma alteração de classificação; `maintenanceid` - `(ID)` ID da maintenance; `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 alerts gerados pelo evento. Os alerts são ordenados em ordem cronológica inversa.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts que contêm o objeto que criou o evento. Suportado apenas para eventos gerados por triggers, items ou regras LLD.
selectRelatedObject	query	Retorna uma propriedade `relatedObject` com o objeto que criou o evento. O tipo de objeto retornado depende do tipo de evento.
selectSuppressionData	query	Retorna uma propriedade `suppression_data` com a lista de manutenções ativas e supressões manuais: `maintenanceid` - `(ID)` ID da maintenance; `userid` - `(ID)` ID do usuário que suprimiu o evento; `suppress_until` - `(integer)` tempo até o qual o evento está suprimido.
selectTags	query	Retorna uma propriedade `tags` com as tags do evento.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro informado. Aceita um objeto, em que as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparação. Não suporta propriedades do tipo de dados `text` data type.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. Valores possíveis: `eventid`, `objectid`, `clock`. Valores possíveis quando usado em conjunto com `groupBy`: `objectid`. Valores possíveis quando usado em conjunto com `countOutput` e `groupBy`: `objectid`, `rowscount`.
groupBy	string/array	Agrupa os resultados pelas propriedades informadas. As propriedades especificadas serão retornadas nos resultados. Valores possíveis: `objectid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.

Parameter	Type	Description
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 dos seguintes:

- uma matriz de objetos
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado, mas o parâmetro `groupBy` não tiver sido usado
- uma matriz de objetos com resultados de agregação, se o parâmetro `groupBy` tiver sido usado

Exemplos

Recuperando eventos de trigger

Recupere os eventos mais recentes do trigger ”22395”.

Request:

```
{
    "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",
            "suppressed": "0",
            "cause_eventid": "0",
```

```
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
```

```
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

Recuperando eventos por período de tempo

Recupere todos os eventos que foram criados entre 17 e 18 de outubro de 2012, em ordem cronológica inversa.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
```

```
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Less than 25% free in the history cache",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


Recuperando eventos reconhecidos por um usuário específico

Recuperando eventos reconhecidos pelo usuário com ID=10

Request:

```
{
    "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",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Recuperando os principais triggers com contagem de eventos de problema

Recupere os 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 tem as seguintes propriedades.

Property	Type	Description
regexpid	ID	ID da expressão regular. Property behavior: - *read-only* - *required* for update operations.
name	string	Nome da expressão regular. Property behavior: - *required* for create operations.
description	text	Descrição da expressão regular.
test_string	string	String de teste.

Expressões

O objeto expressions tem as seguintes propriedades.

Property	Type	Description
expression	string	Expressão regular. Property behavior: - *required*
expression_type	integer	Tipo de expressão regular. Valores possíveis: 0 - Contém a string; 1 - Contém qualquer substring da lista; 2 - Não contém a string; 3 - Corresponde à expressão regular; 4 - Não corresponde à expressão regular. Property behavior: - *required*
exp_delimiter	string	Delimitador da expressão. Valor padrão: `","`. Valores possíveis: `","` ou `"."`, ou `"/"`. Property behavior: - *supported* se `expression_type` estiver definido como *Contém qualquer substring da lista*.
case_sensitive	integer	Sensibilidade a maiúsculas e minúsculas. Valor padrão: 0. Valores possíveis: 0 - Sem distinção entre maiúsculas e minúsculas; 1 - Com distinção entre maiúsculas e 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 obter 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.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Global regular expression used to discover and identify memory resources on the mo
        "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": [
            "9"
        ]
    },
    "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. Consulte Funções de usuário para obter 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(object parameters)`

O método permite recuperar expressões regulares globais de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para *Super Admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
regexpids	ID/array	Retorna apenas as expressões regulares com os IDs fornecidos.
selectExpressions	query	Retorna uma propriedade `expressions`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `regexpid`, `name`.
countOutput	boolean	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.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
```

```
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```


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 obter mais informações.

Parâmetros

`(object/array)` Propriedades da expressão regular a serem atualizadas.

A propriedade `regexpid` deve ser definida para cada objeto, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
expressions	array	Opções de expressões.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das expressões regulares atualizadas na propriedade `regexpids`.

Exemplos

Atualizando a expressão regular global para descoberta de sistemas de arquivos.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": "File systems for discovery",
        "description": "Expressão regular global usada para descobrir e identificar sistemas de arquivos n
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "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*
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 de template pai. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
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".
ymin_type	integer	Método de cálculo do valor mínimo para o eixo Y. Valores possíveis: 0 - *(padrão)* calculado; 1 - fixo; 2 - item.
uuid	string	Identificador único universal, usado para vincular gráficos importados a já existentes. Gerado automaticamente, se não for fornecido. Comportamento da propriedade: - *suportado* se o gráfico pertencer a um template

graph.create

Descrição

`object graph.create(object/array graphs)`

Este método permite criar novos graphs.

> **Note:**
> Este método está disponível apenas para 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)` 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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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.

Parâmetro	Tipo	Descrição
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.
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	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	
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 graphs existentes.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object/array)` Propriedades do gráfico a serem atualizadas.

A propriedade `graphid` deve ser definida para cada gráfico, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do gráfico, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
gitems	array	Itens do gráfico para substituir os itens do gráfico existentes. Se um item do gráfico tiver a propriedade `gitemid` definida, ele será atualizado; caso contrário, um novo item do gráfico será criado.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos gráficos atualizados 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 - cria novos grupos de hosts
- hostgroup.delete - exclui grupos de hosts
- hostgroup.get - recupera grupos de hosts
- hostgroup.massadd - adiciona objetos relacionados aos grupos de hosts
- hostgroup.massremove - remove objetos relacionados dos grupos de hosts
- hostgroup.propagate - propaga permissões e filtros de tags para subgrupos dos grupos de hosts
- hostgroup.update - atualiza 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 simples; 4 - um grupo de hosts descoberto. Comportamento da propriedade: - *somente leitura*
uuid	string	Identificador universal único, usado para vincular grupos de hosts importados aos já existentes. Gerado automaticamente, se não fornecido.

hostgroup.create

Descrição

`object hostgroup.create(object/array hostGroups)`

Este método permite criar novos grupos de hosts.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Grupos de hosts a serem criados.

O método aceita grupos de hosts com as propriedades padrão do grupo de hosts.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs 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:

- ele contiver hosts que pertençam somente a esse grupo;
- ele estiver marcado como interno;
- ele for usado por um protótipo de host;
- ele for usado em um script global;
- ele for usado em uma condição de correlação.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs dos grupos de hosts a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs 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 informados.

> **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 obter 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 fornecidos.
groupids	ID/array	Retorna apenas grupos de hosts com os IDs de grupo de host fornecidos.
hostids	ID/array	Retorna apenas grupos de hosts que contenham os hosts fornecidos.
maintenanceids	ID/array	Retorna apenas grupos de hosts que são afetados pelas manutenções fornecidas.
triggerids	ID/array	Retorna apenas grupos de hosts que contenham hosts com os triggers fornecidos.
with_graphs	boolean	Retorna apenas grupos de hosts que contenham hosts com gráficos.
with_graph_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de gráficos.
with_hosts	boolean	Retorna apenas grupos de hosts que contenham hosts.
with_httptests	boolean	Retorna apenas grupos de hosts que contenham hosts com verificações web. Sobrescreve o parâmetro `with_monitored_httptests`.
with_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items. Sobrescreve os parâmetros `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.

Parâmetro	Tipo	Descrição
with_simple_graph_item_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de item, que estão habilitados para criação e possuem tipo de informação numérica.
with_monitored_httptests	boolean	Retorna apenas grupos de hosts que contenham hosts com verificações web habilitadas.
with_monitored_hosts	boolean	Retorna apenas grupos de hosts que contenham hosts monitorados.
with_monitored_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items habilitados. Sobrescreve o parâmetro `with_simple_graph_items`.
with_monitored_triggers	boolean	Retorna apenas grupos de hosts que contenham hosts com triggers habilitados. Todos os items usados no trigger também devem estar habilitados.
with_simple_graph_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items numéricos.
with_triggers	boolean	Retorna apenas grupos de hosts que contenham hosts com triggers. Sobrescreve o parâmetro `with_monitored_triggers`.
selectDiscoveryRules	query	Retorna uma propriedade `discoveryRules` com as regras LLD que descobriram o grupo de hosts.
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)` hora em que o grupo de hosts que não é mais descoberto será excluído; `status` - `(int)` status da descoberta do grupo de hosts: 0 - *(padrão)* grupo de hosts é descoberto, 1 - grupo de hosts não é mais descoberto.
selectHostPrototypes	query	Retorna uma propriedade `hostPrototypes` com os protótipos de host que descobriram este grupo de hosts.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts que pertencem ao grupo de hosts. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - os resultados serão classificados por `host`.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `groupid`, `name`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Parâmetro	Tipo	Descrição
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 por nome

Recupere todos os dados sobre dois grupos de hosts chamados ”Zabbix servers” e ”Linux servers”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

Veja também

- Host

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 host groups informados.

> **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)` 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 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)` Parâmetros contendo os IDs dos grupos de hosts a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
groupids	ID/array	IDs dos grupos de hosts a serem atualizados. Comportamento do parâmetro: - *obrigatório*
hostids	ID/array	IDs dos hosts a serem removidos de todos os grupos de hosts.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs 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.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 obter 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 tag. 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 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 grupo de hosts a serem atualizadas.

A propriedade `groupid` deve ser definida para cada grupo de hosts, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs 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
}
```

Renomeando vários grupos de hosts

Renomeie dois grupos de hosts para incluir informações de região em seus nomes.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Fonte

CHostGroup::update() em *ui/include/classes/api/services/CHostGroup.php*.

Grupo de proxy

Esta classe foi projetada para trabalhar com grupos de proxy.

Referências de objetos:

- Grupo de proxy

Métodos disponíveis:

- proxygroup.create - criar novos grupos de proxy
- proxygroup.delete - excluir grupos de proxy
- proxygroup.get - recuperar grupos de proxy
- proxygroup.update - atualizar grupos de proxy

Objeto de grupo de proxy

Os seguintes objetos estão diretamente relacionados à API `proxygroup`.

Grupo de proxy

O objeto de grupo de proxy tem as seguintes propriedades.

Property	Type	Description
proxy_groupid	ID	ID do grupo de proxy. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
name	string	Nome do grupo de proxy. Comportamento da propriedade: - *obrigatória* 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 compatíveis (por exemplo, `30s`, `1m`). Macros de usuário são compatíveis. Valores possíveis: 10s-15m. Padrão: 1m.

Property	Type	Description
min_online	string	Número mínimo de proxies online necessário para manter o grupo de proxy online. Macros de usuário são compatíveis. Intervalo de valores possíveis: 1-1000. Padrão: 1.
state	integer	Estado do grupo de proxy. Valores possíveis: 0 - Desconhecido; 1 - Offline; 2 - Em recuperação; 3 - Online; 4 - Em degradação. 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. Consulte Funções de usuário para obter 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 obter 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 da função de usuário. Consulte Funções de usuário para obter mais informações. Somente os grupos de proxy permitidos pela *Lista de acesso de proxy* dos grupos de usuários do usuário são retornados.

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. Consulte Funções de usuário para obter 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 template

Esta classe foi projetada para trabalhar com grupos de template.

Referências de objeto:

- Grupo de template

Métodos disponíveis:

- templategroup.create - criar novos grupos de template
- templategroup.delete - excluir grupos de template
- templategroup.get - recuperar grupos de template
- templategroup.massadd - adicionar objetos relacionados a grupos de template
- templategroup.massremove - remover objetos relacionados de grupos de template
- templategroup.propagate - propagar permissões para subgrupos de grupos de template
- templategroup.update - atualizar grupos de template

Objeto de grupo de template

Os seguintes objetos estão diretamente relacionados à API `templategroup`.

Grupo de template

O objeto de grupo de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de template. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do grupo de template. Comportamento da propriedade: - *obrigatório* para operações de criação
uuid	string	Identificador 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 de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object/array)` Grupos de templates a serem criados. O método aceita grupos de templates com as propriedades padrão do grupo de templates.

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 somente a esse grupo.

> **Note:**
> Este método está disponível apenas para 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 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 que você recupere 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
graphids	ID/array	Retorna apenas grupos de templates que contenham templates com os gráficos informados.
groupids	ID/array	Retorna apenas grupos de templates com os IDs de grupo de template informados.
templateids	ID/array	Retorna apenas grupos de templates que contenham os templates informados.
triggerids	ID/array	Retorna apenas grupos de templates que contenham templates com os triggers informados.
with_graphs	boolean	Retorna apenas grupos de templates que contenham templates com gráficos.
with_graph_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de gráficos.
with_httptests	boolean	Retorna apenas grupos de templates que contenham templates com verificações web.
with_items	boolean	Retorna apenas grupos de templates que contenham templates com items. Sobrescreve os parâmetros `with_simple_graph_items`.
with_item_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de item, que estejam habilitados para criação e tenham tipo de informação numérica.
with_simple_graph_items	boolean	Retorna apenas grupos de templates que contenham templates com items numéricos.
with_templates	boolean	Retorna apenas grupos de templates que contenham templates.
with_triggers	boolean	Retorna apenas grupos de templates que contenham templates com triggers.
selectTemplates	query	Retorna uma propriedade `templates` com os templates que pertencem ao grupo de template. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectTemplates` - os resultados serão classificados por `template`.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. Valores possíveis: `groupid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	

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

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 vários objetos relacionados a todos os grupos de template informados.

> **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)` Parâmetros contendo os IDs dos grupos de templates a serem atualizados e os objetos a serem adicionados a todos os grupos de templates.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de templates a serem atualizados. Os grupos de templates devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
templates	object/array	Templates a serem adicionados a todos os grupos de templates. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template atualizados na propriedade `groupids`.

Exemplos

Adicionando templates a grupos de templates

Adicione dois templates a grupos de templates com IDs 12 e 13.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
```

```
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Veja também

- Template

Fonte

CTemplateGroup::massAdd() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Descrição

`object templategroup.massremove(object parameters)`

Este método permite remover objetos relacionados de vários grupos de templates.

Note:
Este método está disponível apenas para 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)` 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.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 User roles para obter 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 templates existentes.

> **Note:**
> Este método está disponível apenas para 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 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 tem as seguintes propriedades.

Property	Type	Description
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	Indica se o modo de depuração está ativado ou desativado. Valores possíveis: 0 - *(padrão)* desativado; 1 - ativado.
gui_access	integer	Método de autenticação do 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 - desativar o acesso ao frontend.
mfa_status	integer	Indica se a MFA está ativada ou desativada para os usuários do grupo. Valores possíveis: 0 - desativada (para todos os métodos de MFA configurados); 1 - ativada (para todos os métodos de MFA configurados).
mfaid	ID	Método de MFA usado pelos usuários do grupo. Comportamento da propriedade: - *compatível* com usuários do tipo *Super admin* - *compatível* se `mfa_status` do objeto Authentication estiver definido como ”Enabled”
proxy_mode	integer	Modo de gerenciamento do acesso a proxies que não estão incluídos em grupos de proxies. Valores possíveis: 0 - Lista de negação; 1 - *(padrão)* Lista de permissão. Comportamento da propriedade: - *compatível* com usuários do tipo *Super admin*.
proxy_group_mode	integer	Modo de gerenciamento do acesso a grupos de proxies (e seus proxies). Valores possíveis: 0 - Lista de negação; 1 - *(padrão)* Lista de permissão. Comportamento da propriedade: - *compatível* com usuários do tipo *Super admin*.

Property	Type	Description
users_status	integer	Indica se o grupo de usuários está ativado ou desativado. Para usuários desprovisionados, o grupo de usuários não pode ser ativado. Valores possíveis: 0 - *(padrão)* ativado; 1 - desativado.
userdirectoryid	ID	ID do diretório de usuários usado para autenticação. Comportamento da propriedade: - *compatível* com usuários do tipo *Super admin* - *compatível* 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. Consulte User roles para obter mais informações.

Parâmetros

(object/array) Grupos de usuários a serem criados.

Além das propriedades padrão dos grupos de usuários, o método aceita os seguintes parâmetros.

Parameter	Type	Description
hostgroup_rights	object/array	Permissões de grupos de hosts a serem atribuídas ao grupo de usuários.
proxies	object/array	Proxies a serem vinculados ao grupo de usuários. Os proxies devem ter apenas a propriedade `proxyid` definida.
proxy_groups	object/array	Grupos de proxies a serem vinculados ao grupo de usuários. Os grupos de proxies devem ter apenas a propriedade `proxy_groupid` definida.
templategroup_rights	object/array	Permissões de grupos de templates 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 Deprovisioned (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 User roles para obter 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 informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
mfaids	ID/array	Retorna apenas os grupos de usuários com os métodos de MFA especificados. Comportamento do parâmetro: - *compatível* com usuários do tipo *Super admin*
mfa_status	integer	Retorna apenas os grupos de usuários com o status de MFA especificado. Consulte a página de grupos de usuários para obter uma lista dos status compatíveis.
proxyids	ID/array	Retorna apenas os grupos de usuários que contêm os proxies especificados. Comportamento do parâmetro: - *compatível* com usuários do tipo *Super admin*.
proxy_groupids	ID/array	Retorna apenas os grupos de usuários que contêm os grupos de proxies especificados. Comportamento do parâmetro: - *compatível* com usuários do tipo *Super admin*.
selectProxies	ID/array	Retorna uma propriedade `proxies` com os proxies do grupo de usuários. Comportamento do parâmetro: - *compatível* com usuários do tipo *Super admin*.
selectProxyGroups	ID/array	Retorna uma propriedade `proxy_groups` com os grupos de proxies do grupo de usuários. Comportamento do parâmetro: - *compatível* com usuários do tipo *Super admin*.
status	integer	Retorna apenas os grupos de usuários com o status especificado. Consulte a página de grupos de usuários para obter uma lista dos status compatíveis.
userids	ID/array	Retorna apenas os grupos de usuários que contêm os usuários especificados.
usrgrpids	ID/array	Retorna apenas os grupos de usuários com os IDs especificados.
selectTagFilters	query	Retorna as permissões baseadas em tags do grupo de usuários na propriedade `tag_filters`.
selectUsers	query	Retorna os usuários do grupo de usuários na propriedade `users`. Consulte `user.get` para obter informações sobre as restrições baseadas no tipo de usuário.
selectHostGroupRights	query	Retorna as permissões do grupo de usuários para grupos de hosts na propriedade `hostgroup_rights`. Consulte a página de grupos de usuários para obter uma lista dos níveis de acesso aos grupos de hosts.

Parâmetro	Tipo	Descrição
selectTemplateGroupRights	query	Retorna as permissões do grupo de usuários para grupos de templates na propriedade `templategroup_rights`. Consulte a página de grupos de usuários para obter uma lista dos níveis de acesso aos grupos de templates.
limitSelects	integer	Limita o número de registros retornados por subconsultas.
output	query	Propriedades do objeto de 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 especificadas. Valores possíveis: `usrgrpid`, `name`.
countOutput	boolean	Esses parâmetros estão 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. Consulte Funções de usuário para obter 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.

Parameter	Type	Description
hostgroup_rights	object/array	Permissões de grupos de hosts que substituirão as permissões atuais atribuídas ao grupo de usuários.
proxies	object/array	Proxies a serem vinculados ao grupo de usuários. Os proxies devem ter apenas a propriedade `proxyid` definida.
proxy_groups	object/array	Grupos de proxies a serem vinculados ao grupo de usuários. Os grupos de proxies devem ter apenas a propriedade `proxy_groupid` definida.
templategroup_rights	object/array	Permissões de grupos de templates que substituirão as permissões atuais atribuídas ao grupo de usuários.
tag_filters	array	Permissões baseadas em tags que substituirão as permissões atuais atribuídas ao grupo de usuários.
users	object/array	Usuários que substituirão 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*.

History

Esta classe foi projetada para trabalhar com dados de histórico.

Referências de objetos:

- Histórico de float
- Histórico de inteiro
- Histórico de string
- Histórico de texto
- Histórico de log

Métodos disponíveis:

- history.clear - limpa os dados de histórico
- history.get - recupera os dados de histórico
- history.push - envia dados de histórico para o Zabbix server

Objeto de histórico

Os seguintes objetos estão diretamente relacionados à API `history`.

Note:
Os objetos de histórico diferem dependendo do tipo de informação do item. Eles são criados pelo Zabbix server e não podem ser modificados via API.

Histórico de ponto flutuante

O objeto de histórico de ponto flutuante possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	float	Valor recebido.

Histórico de inteiros

O objeto de histórico de inteiros possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	integer	Valor recebido.

Histórico de string

O objeto de histórico de string possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	string	Valor recebido.

Histórico de texto

O objeto de histórico de texto possui as seguintes propriedades.

Propriedade	Tipo	Descrição
id	ID	ID da entrada de histórico.
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	text	Valor recebido.

Histórico de log

O objeto de histórico de log possui as seguintes propriedades.

Propriedade	Tipo	Descrição
id	ID	ID da entrada do histórico.
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
logeventid	integer	ID da entrada do log de eventos do Windows.
ns	integer	Nanosegundos em que o valor foi recebido.
severity	integer	Nível da entrada do log de eventos do Windows.
source	string	Fonte da entrada do log de eventos do Windows.
timestamp	timestamp	Hora da entrada do log de eventos do Windows.
value	text	Valor recebido.

Histórico binário

O objeto de histórico binário possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	text	Valor recebido.

Histórico JSON

O objeto de histórico JSON possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	text	Valor recebido.

history.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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parameter	Type	Description
history	integer	Tipos de objeto de histórico a retornar. Valores possíveis: 0 - float numérico; 1 - caractere; 2 - log; 3 - *(padrão)* numérico sem sinal; 4 - texto; 5 - binário; 6 - JSON.
hostids	ID/array	Retorna apenas o histórico dos hosts informados.
itemids	ID/array	Retorna apenas o histórico dos items informados.
maxValueSize	integer	Número máximo de bytes a retornar na propriedade `value`. Intervalo de valores possíveis: 1-134217728 (1 byte a 128MiB). Se definido como `null`, nenhum limite é aplicado. Padrão: 65536 (64KiB). Comportamento do parâmetro: - *supported* se `history` estiver definido como ”binary” ou ”JSON”
time_from	timestamp	Retorna apenas valores que foram recebidos após ou no horário informado.
time_till	timestamp	Retorna apenas valores que foram recebidos antes ou no horário informado.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. Valores possíveis: `itemid`, `clock`, `ns`. Não oferece suporte à classificação por `ns` se os dados de histórico estiverem armazenados no ClickHouse.
search	object	Retorna resultados que correspondem ao padrão informado (sem diferenciar maiúsculas de minúsculas). Aceita um objeto, em que as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for informada, isso executará uma pesquisa `LIKE "%…%"`. Não oferece suporte à pesquisa por `value` se `history` estiver definido como 6 (JSON).
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	

Parameter	Type	Description
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

Parâmetro	Tipo	Descrição
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.update - atualizar hosts

Objeto host

Os seguintes objetos estão diretamente relacionados à API de host.

Host

O objeto host tem as seguintes propriedades.

Property	Type	Description
hostid	ID	ID do host. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
host	string	Nome técnico do host. Comportamento da propriedade: - *obrigatória* para operações de criação
description	text	Descrição do host.

Property	Type	Description
flags	integer	Origem do host. Valores possíveis: 0 - um host comum; 4 - um host convertido 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 - direto; 5 - OEM; 6 - RMCP+.
ipmi_password	string	Senha do IPMI.
ipmi_privilege	integer	Nível de privilégio do IPMI. Valores possíveis: 1 - callback; 2 - *(padrão)* usuário; 3 - operador; 4 - administrador; 5 - OEM.
ipmi_username	string	Nome de usuário do IPMI.
maintenance_from	timestamp	Hora de início da manutenção efetiva. Comportamento da propriedade: - *somente leitura*
maintenance_status	integer	Status da manutenção efetiva. Valores possíveis: 0 - *(padrão)* sem manutenção; 1 - manutenção em vigor. Comportamento da propriedade: - *somente leitura*
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 atualmente em vigor no host. Comportamento da propriedade: - *somente leitura*
name	string	Nome visível do host. Padrão: valor da propriedade `host`.

Property	Type	Description
monitored_by	integer	Origem usada para monitorar o host. Valores possíveis: 0 - *(padrão)* Zabbix server; 1 - proxy; 2 - grupo de proxies. Comportamento da propriedade: - *obrigatória* se `select_server_for_monitoring` estiver definido como ”Desabilitado”.
proxyid	ID	ID do proxy usado para monitorar o host. Comportamento da propriedade: - *obrigatória* se `monitored_by` estiver definido como ”Proxy”
proxy_groupid	ID	ID do grupo de proxies usado para monitorar o host. Comportamento da propriedade: - *obrigatória* se `monitored_by` estiver definido como ”Grupo de proxies”
status	integer	Status e função do host. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
tls_connect	integer	Conexões com 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 da PSK; deve ser associada a apenas uma PSK (entre autoregistration, hosts e proxies). Não inclua informações confidenciais na identidade da PSK, pois ela é enviada sem criptografia pela rede para informar ao receptor qual PSK deve ser usada. Comportamento da propriedade: - *somente escrita* - *obrigatória* se `tls_connect` estiver definido como ”PSK” ou se `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* - *obrigatória* se `tls_connect` estiver definido como ”PSK” ou se `tls_accept` contiver o bit ”PSK”

Property	Type	Description
active_available	integer	Status de disponibilidade da interface ativa do host. Valores possíveis: 0 - o status da interface é desconhecido; 1 - a interface está disponível; 2 - a 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 proxies. 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 ativo.	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
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

ID	Propriedade	Tipo	Descrição	Comprimento máximo
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	Fornecedor.	64 caracteres

Tag de host

O objeto de tag de host possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do host. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do host. Comportamento da propriedade: - *somente leitura*.

Propriedade	Tipo	Descrição
automatic	integer	Tipo de tag de host. Valores possíveis: 0 - *(padrão)* manual (tag criada pelo usuário); 1 - automática (tag criada por descoberta de baixo nível) Comportamento da propriedade: - *somente leitura*
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Hosts a serem criados.

Além das propriedades padrão do host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts aos quais o host será adicionado. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
interfaces	object/array	Interfaces a serem criadas para o host.
tags	object/array	Tags de 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 a criptografia PSK configurada. Observe que o host precisa estar pré-configurado para usar PSK.

Request:

```
{
    "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
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Criando um host monitorado por um proxy

Crie um host que será monitorado pelo proxy com ID ”1”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Criando um host monitorado por um grupo de proxy

Crie um host que será monitorado por um grupo de proxy com ID ”1”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Veja também

- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Inventário de host
- Tag de host
- Proxy
- Grupo de proxy

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 de usuário. Consulte Funções de usuário para obter 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 informados.

> **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 obter 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 fornecidos.
dserviceids	ID/array	Retorna apenas hosts que estão relacionados aos serviços descobertos fornecidos.
graphids	ID/array	Retorna apenas hosts que possuem os gráficos fornecidos.
hostids	ID/array	Retorna apenas hosts com os IDs de host fornecidos.
httptestids	ID/array	Retorna apenas hosts que possuem as verificações web fornecidas.
interfaceids	ID/array	Retorna apenas hosts que usam as interfaces fornecidas.
itemids	ID/array	Retorna apenas hosts que possuem os items fornecidos.
maintenanceids	ID/array	Retorna apenas hosts que são afetados pelas manutenções fornecidas.
monitored_hosts	flag	Retorna apenas hosts monitorados.
proxyids	ID/array	Retorna apenas hosts que são monitorados pelos proxies fornecidos.
proxy_groupids	ID/array	Retorna apenas hosts que são monitorados pelos grupos de proxy fornecidos.

Parâmetro	Tipo	Descrição
templated_hosts	flag	Retorna tanto hosts quanto templates.
templateids	ID/array	Retorna apenas hosts que estão vinculados aos templates fornecidos.
triggerids	ID/array	Retorna apenas hosts que possuem os triggers fornecidos.
with_items	flag	Retorna apenas hosts que possuem items. Sobrescreve os parâmetros `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	flag	Retorna apenas hosts que possuem protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Retorna apenas hosts que possuem protótipos de item, que estão habilitados para criação e possuem tipo de informação numérica.
with_graphs	flag	Retorna apenas hosts que possuem gráficos.
with_graph_prototypes	flag	Retorna apenas hosts que possuem protótipos de gráfico.
with_httptests	flag	Retorna apenas hosts que possuem verificações web. Sobrescreve o parâmetro `with_monitored_httptests`.
with_monitored_httptests	flag	Retorna apenas hosts que possuem verificações web habilitadas.
with_monitored_items	flag	Retorna apenas hosts que possuem items habilitados. Sobrescreve o parâmetro `with_simple_graph_items`.
with_monitored_triggers	flag	Retorna apenas hosts que possuem triggers habilitados. Todos os items usados no trigger também devem estar habilitados.
with_simple_graph_items	flag	Retorna apenas hosts que possuem items com tipo de informação numérica.
with_triggers	flag	Retorna apenas hosts que possuem triggers. Sobrescreve o parâmetro `with_monitored_triggers`.
withProblemsSuppressed	boolean	Se definido como `true`, retorna apenas hosts com problemas suprimidos.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
severities	integer/array	Retorna hosts que possuem apenas problemas com as severidades fornecidas. Aplica-se apenas se o objeto de problema for trigger.
tags	object/array	Retorna apenas hosts com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os hosts. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
inheritedTags	boolean	Retorna hosts que possuem as `tags` fornecidas também em todos os seus templates vinculados. Valores possíveis: `true` - os templates vinculados também devem possuir as `tags` fornecidas; `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 a partir do qual o host foi criado; `status` - `(int)` status da descoberta do host: 0 - *(padrão)* host descoberto, 1 - host não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um host que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um host que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador se o host foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
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`.

Parâmetro	Tipo	Descrição
selectValueMaps	query	Retorna uma propriedade `valuemaps` com os mapas de valor do host.
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: propriedades de interface de host.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectParentTemplates` - os resultados serão classificados por `host`; `selectInterfaces`; `selectItems` - classificados por `name`; `selectDiscoveryRules` - classificados por `name`; `selectTriggers` - classificados por `description`; `selectGraphs` - classificados por `name`; `selectDashboards` - classificados por `name`.
search	object	Retorna resultados que correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma pesquisa `LIKE "%…%"`. 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 correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma pesquisa `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Estes parâmetros são 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, use `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, 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 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
}
```

Pesquisa por tags de host

Recuperar hosts que possuem a tag ”host-name” igual a ”linux-server”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperar hosts que possuem essas tags não apenas no nível do host, mas também em seus templates pai vinculados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
```

```
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Recuperar hosts excluindo aqueles com a tag ”component”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "tags": [
                {
                    "tag": "class",
                    "value": "os",
                    "automatic": "0"
                },
                {
```

```
                    "tag": "target",
                    "value": "linux",
                    "automatic": "0"
                }
            ]
        },
        {
            "hostid": "10629",
            "tags": [
                {
                    "tag": "class",
                    "value": "application",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "browser",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Pesquisando host com tags e tags de template

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
}
```

Recuperando items do host

Recupere os items do host "Zabbix server". A solicitação é limitada às propriedades do item `itemid`, `name` e `status`. Devido ao grande tamanho da resposta, apenas um subconjunto de items é mostrado no exemplo.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximum number of open file descriptors",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximum number of processes",
          "status": "0"
        },
        {
          "itemid": "69869",
```

```
            "name": "Interface enp0s3: Inbound packets discarded",
            "status": "0"
          },
          {
            "itemid": "69870",
            "name": "Interface enp0s3: Inbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69871",
            "name": "Interface enp0s3: Bits received",
            "status": "0"
          },
          {
            "itemid": "69872",
            "name": "Interface enp0s3: Outbound packets discarded",
            "status": "0"
          },
          {
            "itemid": "69873",
            "name": "Interface enp0s3: Outbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69874",
            "name": "Interface enp0s3: Bits sent",
            "status": "0"
          },
          {
            "itemid": "42253",
            "name": "Number of processes",
            "status": "0"
          },
          {
            "itemid": "42254",
            "name": "Number of running processes",
            "status": "0"
          },
          {
            "itemid": "42255",
            "name": "System boot time",
            "status": "0"
          }
        ]
      }
    ],
    "id": 1
}
```


Veja também

- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Proxy
- Grupo de proxies

Fonte

CHost::get() em *ui/include/classes/api/services/CHost.php*.

host.massadd

Descrição

`object host.massadd(object parameters)`

Este método permite adicionar simultaneamente vários objetos relacionados a todos os hosts informados.

> **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)` 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 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)` Parâmetros contendo os IDs dos hosts a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
hostids	ID/array	IDs dos hosts a serem atualizados. Comportamento do parâmetro: - *obrigatório*
groupids	ID/array	IDs dos grupos de hosts dos quais os hosts fornecidos devem ser removidos.
interfaces	object/array	Interfaces de host a serem removidas dos hosts fornecidos. O objeto de interface de host deve ter apenas as propriedades `ip`, `dns` e `port` definidas.
macros	string/array	Macros de usuário a serem excluídas dos hosts fornecidos.
templateids	ID/array	IDs dos templates a serem desvinculados dos hosts fornecidos.
templateids_clear	ID/array	IDs dos templates a serem desvinculados e limpos dos hosts fornecidos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados 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.update

Descrição

`object host.update(object/array hosts)`

Este método permite atualizar hosts existentes.

> **Note:**
> Este método está disponível apenas para 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 host a serem atualizadas.

A propriedade `hostid` deve ser definida para cada host, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Observe, no entanto, que atualizar o nome técnico do host também atualizará o nome visível do host (se não for fornecido separadamente) pelo valor do nome técnico do host.

Além das propriedades padrão do host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts para substituir os grupos de hosts atuais aos quais o host pertence. Todos os grupos de hosts que não estiverem listados na solicitação serão desvinculados. Os grupos de hosts devem ter apenas a propriedade `groupid` definida.
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 do host.
macros	object/array	Macros de usuário para substituir as macros de usuário atuais. Todas as macros que não estiverem listadas na solicitação serão removidas.
templates	object/array	Templates para substituir os templates atualmente vinculados. Todos os templates que não estiverem listados na solicitação serão apenas desvinculados. Os templates devem ter apenas a propriedade `templateid` definida.
templates_clear	object/array	Templates para desvincular e limpar do host. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados 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
}
```

Habilitando múltiplos hosts

Habilite o monitoramento de dois hosts, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "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 criptografia PSK somente para conexões do host para o server Zabbix e altere a identidade PSK e a chave PSK. Observe que o host precisa estar pré-configurado para usar PSK.

Request:

```
{
    "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
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Veja também

- host.massadd
- host.massremove
- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Inventário de host
- Tag de host
- Proxy
- Grupo de proxies

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 - recuperar 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.

Propriedade	Tipo	Descrição
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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dhostids	ID/array	Retorna apenas hosts descobertos com os IDs fornecidos.
druleids	ID/array	Retorna apenas hosts descobertos que foram criados pelas regras de descoberta fornecidas.
dserviceids	ID/array	Retorna apenas hosts descobertos que estão executando os serviços fornecidos.
selectDRules	query	Retorna uma propriedade `drules` com um array das regras de descoberta que detectaram o host.
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`.

Limpeza

O objeto de configurações tem as seguintes propriedades.

Propriedade	Tipo	Descrição
hk_events_mode	integer	Ativar a limpeza interna de eventos e alertas. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_events_trigger	string	Período de armazenamento dos dados de trigger. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 365d.
hk_events_service	string	Período de armazenamento dos dados de serviço. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 1d.
hk_events_internal	string	Período de armazenamento dos dados internos. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 1d.
hk_events_discovery	string	Período de armazenamento dos dados de descoberta de rede. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 1d.
hk_events_autoreg	string	Período de armazenamento dos dados de registro automático. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 1d.
hk_services_mode	integer	Ativar a limpeza interna de serviços. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_services	string	Período de armazenamento dos dados de serviços. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 365d.

Propriedade	Tipo	Descrição
hk_audit_mode	integer	Ativar a limpeza interna da auditoria. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_audit	string	Período de armazenamento dos dados de auditoria. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 31d.
hk_sessions_mode	integer	Ativar a limpeza interna de sessões. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_sessions	string	Período de armazenamento dos dados de sessões. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 31d.
hk_history_mode	integer	Ativar a limpeza interna do histórico. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_history_global	integer	Substituir o período do histórico do item. Valores possíveis: 0 - Não substituir; 1 - *(padrão)* Substituir.
hk_history	string	Período de armazenamento dos dados do histórico. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 31d.
hk_trends_mode	integer	Ativar a limpeza interna das tendências. Valores possíveis: 0 - Desativar; 1 - *(padrão)* Ativar.
hk_trends_global	integer	Substituir o período de tendências do item. Valores possíveis: 0 - Não substituir; 1 - *(padrão)* Substituir.
hk_trends	string	Período de armazenamento dos dados de tendências. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 365d.
db_extension	string	Sinalizador de configuração da extensão do banco de dados. Se esse sinalizador for definido como ”timescaledb”, o server alterará seu comportamento em relação à limpeza e à exclusão de itens. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
compression_availability	integer	Indica se a compactação de dados é compatível com o banco de dados (ou com sua extensão). Valores possíveis: 0 - Indisponível; 1 - Disponível. Comportamento da propriedade: - *somente leitura*
compression_status	integer	Ativar a compactação do TimescaleDB para o histórico e as tendências. Valores possíveis: 0 - *(padrão)* Desativada; 1 - Ativada.
compress_older	string	Compactar os registros de histórico e tendências mais antigos que o período especificado. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Padrão: 7d.

housekeeping.get

Descrição

`object housekeeping.get(object parameters)`

O método permite recuperar um objeto de 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 de usuário. Consulte User roles para obter 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 de usuário. Consulte Funções de usuário para obter 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*.

Icon map

Esta classe foi projetada para trabalhar com mapas de ícones.

Referências de objetos:

- Mapa de ícones
- Mapeamento de ícones

Métodos disponíveis:

- iconmap.create - criar novos mapas de ícones
- iconmap.delete - excluir mapas de ícones
- iconmap.get - recuperar mapas de ícones
- iconmap.update - atualizar mapas de ícones

Objeto de mapa de ícones

Os seguintes objetos estão diretamente relacionados à API `iconmap`.

Mapa de ícones

O objeto mapa de ícones possui as seguintes propriedades.

Propriedade	Tipo	Descrição
iconmapid	ID	ID do mapa de ícones. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
default_iconid	ID	ID do ícone padrão. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do mapa de ícones. Comportamento da propriedade: - *obrigatório* para operações de criação

Mapeamento de ícones

O objeto de mapeamento de ícones define um ícone específico a ser usado para hosts com um determinado valor de campo de inventário. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
iconid	ID	ID do ícone usado pelo mapeamento de ícones. Comportamento da propriedade: - *obrigatório*
expression	string	Expressão para comparar com o campo de inventário. Comportamento da propriedade: - *obrigatório*
inventory_link	integer	ID do campo de inventário do host. Consulte o objeto de inventário do host para obter uma lista de campos de inventário suportados. Comportamento da propriedade: - *obrigatório*
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 obter 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 obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
iconmapids	ID/array	Retornar apenas mapas de ícones com os IDs fornecidos.
sysmapids	ID/array	Retornar apenas mapas de ícones que são usados nos mapas fornecidos.
selectMappings	query	Retornar uma propriedade `mappings` com os mapeamentos de ícones usados.
sortfield	string/array	Classificar 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 obter 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*.

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
imagetype	integer	Tipo de imagem. Valores possíveis: 1 - *(padrão)* ícone; 2 - imagem de fundo. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
image	string	Imagem codificada em Base64. O tamanho máximo da imagem codificada é de 1 MB. O tamanho máximo pode ser ajustado alterando o valor da constante `ZBX_MAX_IMAGE_SIZE`. Formatos de imagem suportados: PNG, JPEG, GIF e WebP. Comportamento da propriedade: - *obrigatório* para operações de criação

image.create

Descrição

`object image.create(object/array images)`

Este método permite criar novas imagens.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter 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 de ícone

Crie um ícone de nuvem.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```


Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

Criar uma imagem de fundo

Crie uma pequena imagem de fundo para ser usada como padrão.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```


Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Fonte

CImage::create() em *ui/include/classes/api/services/CImage.php*.

image.delete

Descrição

`object image.delete(array imageIds)`

Este método permite excluir imagens.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter 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 informados.

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 obter 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.

Parâmetro	Tipo	Descrição
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 de usuário. Consulte User roles para obter 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 que você recupere a versão da Zabbix API.

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*.

Informações do server

Esta classe foi projetada para recuperar informações sobre o server do Zabbix.

Métodos disponíveis:

- serverinfo.get - recuperar o horário da última atualização do server do Zabbix

serverinfo.get

Este método permite recuperar o horário da última atualização do Zabbix server especificado.

O método está disponível para *User*, *Admin*, *Super admin* e usuários não autenticados.

Assinatura do método: `object serverinfo.get(object parameters)`

Origem do método: `CServerInfo::get()` em `ui/include/classes/api/services/CServerInfo.php`

Parâmetros

Parameter	Type	Description
serverid	ID	ID do server do Zabbix. Comportamento do parâmetro: - Obrigatório
output	query	Este parâmetro é descrito nos comentários de referência.

Valores de retorno

Property	Type	Description
serverid	ID	ID do Zabbix server.
lastaccess	timestamp	Hora da última atualização do Zabbix server. Para um cluster de alta disponibilidade (HA), isso mostra o valor de `lastaccess` do nó ativo.

Exemplos

Recuperar informações do server

Recupere informações sobre o server Zabbix especificado.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

Interface do host

Esta classe fornece acesso às interfaces do host.

Referências de objetos:

- Interface do host
  - Detalhes

Método disponível:

- hostinterface.get - recuperar interfaces do host

As interfaces do host são criadas, atualizadas e excluídas usando métodos do host.

Objeto de interface de host

Os seguintes objetos estão diretamente relacionados à API `hostinterface`.

Interface do host

O objeto de interface do host tem as seguintes propriedades.

Propriedade	Tipo	Descrição
interfaceid	ID	ID da interface. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização

Propriedade	Tipo	Descrição
available	integer	Disponibilidade da interface do host. Valores possíveis: 0 - *(padrão)* desconhecida; 1 - disponível; 2 - indisponível. Comportamento da propriedade: - *somente leitura*
hostid	ID	ID do host ao qual a interface pertence. Comportamento da propriedade: - *constante* - *obrigatória* para operações de criação
type	integer	Tipo de interface. Valores possíveis: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Comportamento da propriedade: - *obrigatória* para operações de criação
ip	string	Endereço IP usado pela interface. Pode estar vazio se a conexão for feita via DNS. Comportamento da propriedade: - *obrigatória* para operações de criação
dns	string	Nome DNS usado pela interface. Pode estar vazio se a conexão for feita via IP. Comportamento da propriedade: - *obrigatória* para operações de criação
port	string	Número da porta usado pela interface. Pode conter macros de usuário. Comportamento da propriedade: - *obrigatória* para operações de criação
useip	integer	Indica 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: - *obrigatória* para operações de criação
main	integer	Indica se a interface é usada como padrão no host. Somente uma interface de cada tipo pode ser definida como padrão em um host. Valores possíveis: 0 - não é padrão; 1 - padrão. Comportamento da propriedade: - *obrigatória* para operações de criação
details	object	Objeto details adicional para a interface. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP”

Propriedade	Tipo	Descrição
disable_until	timestamp	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 solicitações SNMP bulk. Valores possíveis: 0 - não usar solicitações bulk; 1 - (padrão) - usar solicitações bulk.
community	string	Comunidade SNMP. Usado apenas por interfaces SNMPv1 e SNMPv2. Comportamento da propriedade: - *obrigatório* se `version` estiver definido como "SNMPv1" ou "SNMPv2c"
max_repetitions	integer	Valor máximo de repetição para 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. Usado apenas por interfaces SNMPv3.
securitylevel	integer	Nível de segurança SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	Senha de autenticação SNMPv3. Usado apenas por interfaces SNMPv3.
privpassphrase	string	Senha de privacidade SNMPv3. Usado apenas por interfaces SNMPv3.

Propriedade	Tipo	Descrição
authprotocol	integer	Protocolo de autenticação SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	Protocolo de privacidade SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	Nome do contexto SNMPv3. Usado apenas por interfaces SNMPv3.

hostinterface.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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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 pelos itens nos triggers fornecidos.
selectItems	query	Retorna uma propriedade `items` com os itens que usam a interface. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts que usam a interface.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectItems`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	

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

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*.

Item

Esta classe foi projetada para trabalhar com items.

Referências de objetos:

- Item
  - HTTP header
  - HTTP query field
- Item tag
- Item preprocessing

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 por meio da API do Zabbix.

O objeto item tem 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 um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário. Se usada, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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 estiver 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), ”Telemetry query” (24), ou se type estiver 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 a um host e type estiver definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” ou ”SNMP agent” - *com suporte* se o item pertencer a um host e type estiver 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: - *obrigatória* para operações de criação - *somente leitura* para objetos herdados ou descobertos
name	string	Nome do item. Oferece suporte a macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
name_resolved	string	Nome do item com as 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; 24 - Telemetry query. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
url	string	String de URL. Oferece suporte a macros de usuário, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
value_type	integer	Tipo de informação do item. Valores possíveis: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
allow_traps	integer	Permite que o valor seja preenchido de forma semelhante ao item trapper. 0 - *(padrão)* Não aceitar dados recebidos; 1 - Aceitar dados recebidos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos descobertos

Propriedade	Tipo	Descrição
authtype	integer	Método de autenticação. Valores possíveis se `type` estiver definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` estiver definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”) ou objetos descobertos
description	string	Descrição do item. Comportamento da propriedade: - *somente leitura* para objetos descobertos
error	string	Texto do erro caso haja problemas ao atualizar o valor do item. Comportamento da propriedade: - *somente leitura*
flags	integer	Origem do item. Valores possíveis: 0 - um item comum; 4 - um item convertido de um protótipo. Comportamento da propriedade: - *somente leitura*
follow_redirects	integer	Seguir redirecionamentos da resposta durante a coleta de dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
headers	array	Matriz de cabeçalhos que será enviada ao executar uma solicitação HTTP. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
history	string	Unidade de tempo durante a qual 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: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
inventory_link	integer	ID do campo de inventário do host preenchido pelo item. Consulte a página de inventário do host para obter uma lista dos campos de inventário do host compatíveis e seus IDs. Padrão: 0. Comportamento da propriedade: - *com suporte* se `value_type` estiver 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` estiver definido como ”IPMI agent” e `key_` não estiver definido como ”ipmi.get” - *com suporte* se `type` estiver definido como ”IPMI agent” - *somente leitura* para objetos herdados ou descobertos
jmx_endpoint	string	String de conexão personalizada do JMX agent. Valor padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”JMX agent” - *somente leitura* para objetos descobertos
lastclock	timestamp	Momento em que o valor do item foi atualizado pela última vez. Por padrão, somente os valores das últimas 24 horas são exibidos. É possível estender esse período alterando o valor do parâmetro *Max history display period* na seção de menu *Administration > General*. Comportamento da propriedade: - *somente leitura*
lastns	integer	Nanossegundos do momento em que o valor do item foi atualizado pela última vez. Por padrão, somente os valores das últimas 24 horas são exibidos. É possível estender esse período alterando o valor do parâmetro *Max history display period* na seção de menu *Administration > General*. Comportamento da propriedade: - *somente leitura*
lastvalue	string	Último valor do item. Por padrão, somente os valores das últimas 24 horas são exibidos. É possível estender esse período alterando o valor do parâmetro *Max history display period* na seção de menu *Administration > General*. Comportamento da propriedade: - *somente leitura*
logtimefmt	string	Formato da hora nas entradas de log. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”log” - *somente leitura* para objetos herdados ou descobertos
master_itemid	ID	ID do item mestre. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Dependent item” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
output_format	integer	Define se a resposta deve ser convertida para JSON. 0 - *(padrão)* Armazenar como bruta; 1 - Converter para JSON. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
params	string	Parâmetros adicionais dependendo do tipo do item: - script executado para itens SSH agent e TELNET agent; - consulta SQL para itens Database monitor; - fórmula para itens Calculated; - script para itens Script e Browser. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` estiver definido como ”Script” ou ”Browser”) ou objetos descobertos
parameters	object/array	Parâmetros adicionais se `type` estiver definido como ”Script” ou ”Browser”. Matriz de objetos com as propriedades `name` e `value`, em que `name` deve ser exclusivo. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados ou descobertos
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”JMX agent” e `username` estiver definido - *com suporte* se `type` estiver definido como ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”) ou objetos descobertos
post_type	integer	Tipo do corpo dos dados de postagem armazenado na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos; 2 - Dados JSON; 3 - Dados XML. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
posts	string	Dados do corpo da solicitação HTTP(S). Comportamento da propriedade: - *obrigatórios* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”JSON data” ou ”XML data” - *com suporte* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”Raw data” - *somente leitura* para objetos herdados ou descobertos
prevvalue	string	Valor anterior do item. Por padrão, somente os valores das últimas 24 horas são exibidos. É possível estender esse período alterando o valor do parâmetro *Max history display period* na seção de menu *Administration > General*. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key” - *somente leitura* para objetos descobertos
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key” - *somente leitura* para objetos descobertos
query_fields	array	Matriz de campos de consulta que será enviada ao executar uma solicitação HTTP. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
request_method	integer	Tipo do método de solicitação. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
retrieve_mode	integer	Parte da resposta que deve ser armazenada. Valores possíveis se `request_method` estiver definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - O corpo e os cabeçalhos serão armazenados. Valores possíveis se `request_method` estiver definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP agent” - *somente leitura* para objetos herdados ou descobertos
ssl_cert_file	string	Caminho do arquivo de chave SSL pública. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
ssl_key_file	string	Caminho do arquivo de chave SSL privada. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
ssl_key_password	string	Senha do arquivo de chave SSL. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
state	integer	Estado do item. Valores possíveis: 0 - *(padrão)* normal; 1 - sem suporte. 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 oferece suporte a macros de usuário como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
templateid	ID	ID do item do template pai. *Dica*: use a propriedade `hostid` para especificar o template ao qual o item pertence. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da solicitação de coleta de dados do item. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, `300s`, `1m`). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-600s. Padrão: ”” - usar as configurações do proxy/globais. Comportamento da propriedade: - *com suporte* se `type` estiver 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, - se o valor padrão for aplicado e o item pertencer a um host monitorado por um proxy ou grupo de proxies inacessível, será usado o tempo limite global.

Propriedade	Tipo	Descrição
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *somente leitura* para objetos descobertos - *com suporte* se `type` estiver definido como ”Zabbix trapper”, ou se `type` estiver definido como ”HTTP agent” e `allow_traps` estiver definido como ”Allow incoming data”. Se a propriedade for omitida, ela será definida automaticamente como a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}` (valor padrão). Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Se a propriedade for definida como uma string vazia, as conexões recebidas serão negadas de todos os hosts.
trends	string	Unidade de tempo durante a qual os dados de tendências devem ser armazenados. Também aceita macro de usuário. Padrão: 365d. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”numeric float” ou ”numeric unsigned” - *somente leitura* para objetos descobertos
units	string	Unidades do valor. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”numeric float” ou ”numeric unsigned” - *somente leitura* para objetos herdados ou descobertos
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent”, ”TELNET agent”, ou se `type` estiver definido como ”JMX agent” e `password` estiver definido - *com suporte* se `type` estiver definido como ”Simple check”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”) ou objetos descobertos
uuid	string	Identificador universal exclusivo, usado para vincular o item importado aos itens já existentes. Gerado automaticamente se não for fornecido. Comportamento da propriedade: - *com suporte* se o item pertencer a um template
valuemapid	ID	ID do mapa de valores associado. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”numeric float”, ”character” ou ”numeric unsigned” - *somente leitura* para objetos herdados ou descobertos
verify_host	integer	Define se deve ser validado que o nome do host da conexão corresponde ao nome no certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
verify_peer	integer	Define se deve ser validado que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
query	object	Configuração da consulta de telemetria. Define o que o item consulta na fonte de dados de telemetria, como filtra os dados e como agrega o resultado. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Telemetry query” - *somente leitura* para objetos herdados ou descobertos
time_shift	string	Quanto para trás (antes do momento atual) termina o intervalo de tempo consultado pelo item. Aceita um número de segundos (por exemplo, `1`, `60`, `300`) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`). Também aceita macros de usuário. O intervalo de valores possíveis é: 0-86400s. Padrão: 15s. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”
lookback_limit	string	Quanto para trás (antes do momento atual) o item consulta dados quando não há uma consulta anterior da qual continuar (por exemplo, na primeira execução do item), ou quando a consulta anterior é mais antiga que `delay` (por exemplo, porque o Zabbix server estava inativo). Aceita um número de segundos (por exemplo, `1`, `60`, `300`) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-259200s. Padrão: 10m. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”
granularity	string	Duração de cada parte em que o intervalo de tempo completo da consulta é dividido. Aceita um número de segundos (por exemplo, `1`, `60`, `300`) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-86400s. Padrão: 15s. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”

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*

Consulta de telemetria

O objeto de consulta de telemetria tem as seguintes propriedades.

Property	Type	Description
signal_type	integer	Tipo de sinal consultado pelo item. Valores possíveis: 0 - Traces; 1 - Métricas; 2 - Logs. Comportamento da propriedade: - *obrigatório*
metric_point_type	integer	Tipo de ponto de métrica consultado pelo item. Valores possíveis: 0 - Soma; 1 - Gauge; 2 - Histograma; 3 - Histograma exponencial. Comportamento da propriedade: - *obrigatório* se `signal_type` estiver definido como ”Metrics”

Property	Type	Description
columns	array	Array de colunas de dados de telemetria consultadas pelo item. Pode estar vazio. Comportamento da propriedade: - *obrigatório*
aggregated_columns	array	Array de colunas agregadas de dados de telemetria consultadas pelo item e de como ele calcula um valor a partir dos registros das colunas. Comportamento da propriedade: - *obrigatório*
filter	object	Filtro de consulta de telemetria para os registros de colunas na consulta. Comportamento da propriedade: - *obrigatório*

Colunas de dados de telemetria

O objeto de colunas de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria consultada pelo item. Consulte colunas de dados de telemetria compatíveis. Para várias colunas de valor único, cada valor de `column` deve ser exclusivo. Para várias colunas que contêm pares de chave-valor, cada par de valores `column` e `attribute_key` deve ser exclusivo.
attribute_key	string	Nome da chave consultada pelo item na coluna que contém pares de chave-valor. Para várias colunas que contêm pares de chave-valor, cada par de valores `column` e `attribute_key` deve ser exclusivo. Comportamento da propriedade: - *obrigatório* se `column` estiver definido como uma coluna que contém pares de chave-valor

Colunas agregadas de dados de telemetria

O objeto de colunas agregadas de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria consultada pelo item. Consulte colunas de dados de telemetria compatíveis. Para várias colunas, cada valor de `column` deve ser exclusivo. Esta propriedade não é compatível quando `function` está definido como ”Count”. Comportamento da propriedade: - *obrigatória* se `function` estiver definido como ”Min”, ”Max”, ”Average”, ”Sum” ou ”Percentile”

Property	Type	Description
function	integer	Função que o item aplica aos registros da coluna. Valores possíveis: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Comportamento da propriedade: - *obrigatória*
parameters	array	Percentil que o item calcula a partir dos registros da coluna. Valor possível: um número de 0 a 100, fornecido como uma string. Comportamento da propriedade: - *obrigatória* se `function` estiver definido como "Percentile"
alias	string	Nome exclusivo do valor calculado, usado para identificá-lo no valor do item. Cada valor de `alias` deve ser exclusivo. Comportamento da propriedade: - *obrigatória*

Filtro de dados de telemetria

O objeto de filtro de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
conditions	array	Condições do filtro de dados de telemetria que um registro em uma coluna de dados de telemetria deve atender. Comportamento da propriedade: - *obrigatório*
evaltype	integer	Método de avaliação das condições do filtro. Valores possíveis: 0 - And/Or; 1 - And; 2 - Or; 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 específicas do filtro por meio de `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*

Property	Type	Description
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 façam referência a condições específicas do filtro por meio de `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer sem uso ou ser omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”Expressão personalizada”

Condições de filtro de dados de telemetria

O objeto de condições de filtro de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria à qual a condição de filtro se aplica. Consulte colunas de dados de telemetria compatíveis. Comportamento da propriedade: - *obrigatório*
attribute_key	string	Nome da chave à qual a condição se aplica, dentro da coluna que contém pares de chave-valor. Comportamento da propriedade: - *obrigatório* se `column` estiver definida como uma coluna que contém pares de chave-valor
operator	integer	Operador da condição de filtro. Valores possíveis para colunas de valor único: 0 - É igual a; 1 - Não é igual a; 2 - Contém; 3 - Não contém. Valores possíveis para colunas que contêm pares de chave-valor: 0 - É igual a; 1 - Não é igual a; 12 - Existe. Comportamento da propriedade: - *obrigatório*
value	string	Valor a ser comparado. Comportamento da propriedade: - *obrigatório* se `operator` estiver definido como ”É igual a”, ”Não é igual a”, ”Contém” ou ”Não contém”
formulaid	string	ID exclusivo arbitrário usado para referenciar a condição em uma expressão personalizada. Pode conter apenas letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições de filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto Filtro de consulta de telemetria estiver definido como ”Expressão personalizada”

Tag de item

O objeto de tag de 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. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template; 1 - host. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

Pré-processamento de item

O objeto de pré-processamento de item tem as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo de opção de pré-processamento. Valores possíveis: 1 - Multiplicador personalizado; 2 - Remover espaços à direita; 3 - Remover espaços à esquerda; 4 - Remover espaços; 5 - Expressão regular; 6 - Booleano para decimal; 7 - Octal para decimal; 8 - Hexadecimal para decimal; 9 - Alteração simples; 10 - Alteração por segundo; 11 - XPath XML; 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 se inalterado; 20 - Descartar se inalterado com heartbeat; 21 - JavaScript; 22 - Padrão do Prometheus; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 26 - Verificar não suportado; 27 - XML para JSON; 28 - Valor de SNMP walk; 29 - SNMP walk para JSON; 30 - Valor de SNMP get. Comportamento da propriedade: - *obrigatória*
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Vários 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>]`, em que *pattern* é uma expressão regular e *scope* é um dos seguintes: -1 - corresponde a qualquer erro; 0 - verifica se a mensagem de erro corresponde a *pattern*; 1 - verifica se a mensagem de erro não corresponde a *pattern*. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Remover espaços à direita” (2), ”Remover espaços à esquerda” (3), ”Remover espaços” (4), ”Expressão regular” (5), ”XPath XML” (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 se inalterado com heartbeat” (20), ”JavaScript” (21), ”Padrão do Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), Verificar não suportado (26), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor de SNMP get” (30)

Propriedade	Tipo	Descrição
error_handler	integer	Tipo de ação usado em caso de falha da etapa de pré-processamento. Valores possíveis: 0 - A mensagem de erro é 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: - *obrigatória* 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), ”XPath XML” (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 do Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Verificar não suportado” (26), ”XML para JSON” (27), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor de SNMP get” (30)
error_handler_params	string	Parâmetros do manipulador de erros. Comportamento da propriedade: - *obrigatória* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os seguintes parâmetros e manipuladores de erros são compatíveis com cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
1	Multiplicador personalizado	número			0, 1, 2, 3
2	Remover espaços à direita	lista de caracteres[2]			
3	Remover espaços à esquerda	lista de caracteres[2]			
4	Remover espaços	lista de caracteres[2]			
5	Expressão regular	padrão[3]	saída[2]		0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
6	Booleano para decimal				0, 1, 2, 3
7	Octal para decimal				0, 1, 2, 3
8	Hexadecimal para decimal				0, 1, 2, 3
9	Alteração simples				0, 1, 2, 3
10	Alteração por segundo				0, 1, 2, 3
11	XPath XML	caminho[4]			0, 1, 2, 3
12	JSONPath	caminho[4]			0, 1, 2, 3
13	No intervalo	mín.^1, 6^	máx.^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 se inalterado				

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
20	Descartar se inalterado com heartbeat	segundos^[5], 6^			
21	JavaScript	script[2]			
22	Padrão do Prometheus	padrão^[6, 7]^	`value`, `label`, `function`	saída^[8, 9]^	0, 1, 2, 3
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 pesquisa[2]	substituição[2]		
26	Verificar não suportado	escopo[1]	padrão^[3, 6]^		1, 2, 3
27	XML para JSON				0, 1, 2, 3
28	Valor de 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 do 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 de SNMP get	Formato: 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS			0, 1, 2, 3

[1] número de ponto flutuante (os inteiros são convertidos implicitamente em valores de ponto flutuante)
 [2] string
 [3] expressão regular
 [4] JSONPath ou XPath XML
 [5] inteiro positivo (com suporte a sufixos de tempo, por exemplo, `30s`, `1m`, `2h`, `1d`)
 [6] macro de usuário
 [7] padrão do Prometheus seguindo a sintaxe: `<metric name>{<label name>="<label value>", ...} == <value>`. Cada componente do padrão do Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser uma macro de usuário.
 [8] saída do Prometheus seguindo a sintaxe: `<label name>` (pode ser uma macro de usuário) se `label` estiver selecionado como segundo parâmetro.
 [9] Uma das funções de agregação: `sum`, `min`, `max`, `avg`, `count` se `function` estiver selecionado como segundo parâmetro.
 [10] Compatível com vários registros "Nome do campo,Prefixo do 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 a Zabbix API.

> **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)` 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": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
```

```
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Criando um item dependente

Crie um item dependente para o item mestre com ID 24759. 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 web não podem ser excluídos via Zabbix API.

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 items a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos items excluídos na propriedade `itemids`.

Exemplos

Excluindo vários 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 informados.

> **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 User roles para obter 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.
inheritedTags	boolean	Retorna items que possuem as `tags` fornecidas também em host/template/templates vinculados. Valores possíveis: `true` - template/host/templates vinculados também devem ter as tags fornecidas; `false` - *(padrão)* tags de template/host/templates vinculados são ignoradas.
templated	boolean	Se definido como `true`, retorna apenas os items que pertencem a templates.
monitored	boolean	Se definido como `true`, retorna apenas os items habilitados que pertencem a hosts monitorados.
group	string	Retorna apenas os items que pertencem a um grupo com o nome fornecido.
host	string	Retorna apenas os items que pertencem a um host com o nome fornecido.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.

Parâmetro	Tipo	Descrição
tags	array	Retorna apenas os items com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os items. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
with_triggers	boolean	Se definido como `true`, retorna apenas os items que são usados em triggers.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts aos quais o item pertence.
selectInterfaces	query	Retorna uma propriedade `interfaces` com um array de interfaces de host usadas pelo item.
selectTriggers	query	Retorna uma propriedade `triggers` com os triggers nos quais o item é usado. Suporta `count`.
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.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags que estão em template/host/templates vinculados.
selectPreprocessing	query	Retorna uma propriedade `preprocessing` com as opções de pré-processamento do item.
selectTags	query	Retorna as tags do item na propriedade `tags`.
selectValueMap	query	Retorna uma propriedade `valuemap` com o mapa de valores do item.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o item pertence.

Parâmetro	Tipo	Descrição
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `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`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
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 usado.

Exemplos

Encontrando items pela chave

Recupere todos os items usados em triggers para um ID de host específico que tenham a palavra ”system.cpu” na chave do item e ordene os resultados pelo nome.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "CPU utilization",
            "key_": "system.cpu.util",
            "delay": "0",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "%",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42267",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "CPU utilization in %.",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "42264",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "CPU utilization",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
```

```
        },
        {
            "itemid": "42259",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (15m avg)",
            "key_": "system.cpu.load[all,avg15]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42219",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (15m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
```

```
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42249",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (1m avg)",
            "key_": "system.cpu.load[all,avg1]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42209",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (1m avg)",
            "parameters": [],
```

```
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (5m avg)",
            "key_": "system.cpu.load[all,avg5]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
```

```
            "name_resolved": "Load average (5m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "key_": "system.cpu.num",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
```

```
            "state": "0",
            "error": "",
            "name_resolved": "Number of CPUs",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```

Encontrando items dependentes por chave

Recupere todos os items dependentes do host com ID ”10116” que tenham 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": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
```

```
        "flags": "0",
        "interfaceid": "0",
        "description": "",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "25545",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "state": "0",
        "error": "",
        "name_resolved": "Days",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "25555",
        "type": "18",
        "snmp_oid": "",
        "hostid": "10116",
        "name": "Hours",
        "key_": "apache.status.uptime.hours",
        "delay": "0",
        "history": "90d",
        "trends": "365d",
        "status": "0",
        "value_type": "3",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
```

```
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Hours",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Encontrar item HTTP agent

Encontrar 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": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
```

```
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "template item",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": ""
        }
    ],
    "id": 1
}
```

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 Zabbix API.

> **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 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 foram usados pelo script anterior.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItem::update() em *ui/include/classes/api/services/CItem.php*.

Item de gráfico

Esta classe foi projetada para trabalhar com itens de gráfico.

Referências de objetos:

- Item de gráfico

Métodos disponíveis:

- graphitem.get - recupera itens de gráfico

Objeto de item de gráfico

Os seguintes objetos estão diretamente relacionados à API `graphitem`.

Item do gráfico

> **Note:**
> Os itens do gráfico só podem ser modificados via a API `graph`.

O objeto item do gráfico possui as seguintes propriedades.

Propriedade	Tipo	Descrição
gitemid	ID	ID do item do gráfico. Comportamento da propriedade: - *somente leitura*
color	string	Cor de desenho do item do gráfico como um código de cor hexadecimal. Comportamento da propriedade: - *obrigatório* para operações de criação
itemid	ID	ID do item. Comportamento da propriedade: - *obrigatório* para operações de criação
calc_fnc	integer	Valor do item que será exibido. Valores possíveis: 1 - valor mínimo; 2 - *(padrão)* valor médio; 4 - valor máximo; 7 - todos os valores; 9 - último valor, usado apenas por gráficos de pizza e gráficos explodidos.
drawtype	integer	Estilo de desenho do item do 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 do gráfico pertence.
sortorder	integer	Posição do item no gráfico. Padrão: começa com ”0” e aumenta em um a cada entrada.
type	integer	Tipo de item do gráfico. Valores possíveis: 0 - *(padrão)* simples; 2 - soma do gráfico, usada apenas por gráficos de pizza e gráficos explodidos.
yaxisside	integer	Lado do gráfico onde a escala Y do item do 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 gráfico 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 obter 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 uma lista dos tipos de itens de gráfico suportados.
selectGraphs	query	Retorna uma propriedade `graphs` com um array de gráficos aos quais o item pertence.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `gitemid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valores de retorno

`(integer/array)` Retorna:

- 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 APM

Esta classe foi projetada para trabalhar com logs coletados para monitoramento do desempenho de aplicações (APM).

Referências de objetos:

- Log de APM

Métodos disponíveis:

- apmlog.get - recuperar registros de log

Objeto de log do APM

Os objetos a seguir estão diretamente relacionados à API `apmlog`.

> **Note:**
> Cada propriedade corresponde à coluna com o mesmo nome na tabela `otel_logs` da fonte de dados de telemetria (por exemplo, a propriedade `severity_text` corresponde à coluna `SeverityText`). Para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis.

Log do APM

O objeto de log do APM representa um registro de log (um registro de texto de um evento, como um pagamento com falha ou um timeout de banco de dados). Um registro de log contém informações sobre um evento que ocorreu em um momento específico, incluindo sua gravidade, mensagem e atributos. Se um registro de log foi produzido durante uma operação que faz parte de um trace, ele também contém os IDs desse trace e span.

O objeto de log do APM tem as seguintes propriedades:

Propriedade	Tipo	Descrição
timestamp	integer	Hora Unix (em nanossegundos) em que o evento ocorreu. Comportamento da propriedade: - *somente leitura*
traceid	string	ID, se houver, do trace durante o qual o registro de log foi produzido. Comportamento da propriedade: - *somente leitura*
spanid	string	ID, se houver, do span durante o qual o registro de log foi produzido. Comportamento da propriedade: - *somente leitura*
trace_flags	integer	Flags de trace do contexto do trace do registro de log, como uma máscara de bits. Se o bit menos significativo estiver definido (valor 1), o trace será amostrado. Comportamento da propriedade: - *somente leitura*
severity_text	string	Nível de gravidade do registro de log, conforme fornecido pela aplicação (por exemplo, `INFO`, `WARN` ou `ERROR`). Comportamento da propriedade: - *somente leitura*
severity_number	integer	Nível de gravidade do registro de log, como um número. Valores possíveis: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL. Comportamento da propriedade: - *somente leitura*
service_name	string	Nome do serviço que produziu o registro de log (por exemplo, `zabbix-frontend`), obtido do atributo de recurso `service.name` do OpenTelemetry. Comportamento da propriedade: - *somente leitura*
body	string	Mensagem do registro de log. Comportamento da propriedade: - *somente leitura*
resource_schema_url	string	URL, se houver, do schema do OpenTelemetry dos atributos de recurso. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
resource_attributes	object	Atributos de recurso de chave-valor do registro de log (por exemplo, `host.name: eabf8c2a5ddc`). Comportamento da propriedade: - *somente leitura*
scope_schema_url	string	URL, se houver, do schema do OpenTelemetry do escopo de instrumentação e de seus atributos. Comportamento da propriedade: - *somente leitura*
scope_name	string	Nome do escopo de instrumentação que produziu o registro de log. Comportamento da propriedade: - *somente leitura*
scope_version	string	Versão, se houver, do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
scope_attributes	object	Atributos de chave-valor do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
log_attributes	object	Atributos de log de chave-valor do registro de log (por exemplo, `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Comportamento da propriedade: - *somente leitura*
event_name	string	Nome, se houver, do tipo de evento que o registro de log representa. Comportamento da propriedade: - *somente leitura*

apmlog.get

Descrição

`integer/array apmlog.get(object parameters)`

O método permite recuperar registros de log 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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parameter	Type	Description
time_from	timestamp	Retorna somente os registros de log de eventos que ocorreram no horário especificado ou posteriormente. Comportamento do parâmetro: - *obrigatório*
time_till	timestamp	Retorna somente os registros de log de eventos que ocorreram antes do horário especificado. Comportamento do parâmetro: - *obrigatório*

Parameter	Type	Description
traceids	string/array	Retorna somente os registros de log produzidos durante os traces especificados.
spanids	string/array	Retorna somente os registros de log produzidos durante os spans especificados.
with_flags_on	integer	Retorna somente os registros de log que têm todos os trace flags especificados definidos. Para especificar vários flags, use a soma de seus valores.
with_flags_off	integer	Retorna somente os registros de log que não têm nenhum dos trace flags especificados definido. Para especificar vários flags, use a soma de seus valores.
resource_attributes	object/array	Retorna somente os registros de log com os atributos de recurso especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os registros de log. Valores possíveis de operator: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
resource_attributes_evaltype	integer	Método de avaliação das condições em `resource_attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
scope_attributes	object/array	Retorna somente os registros de log com os atributos de escopo especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os registros de log. Valores possíveis de operator: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
scope_attributes_evaltype	integer	Método de avaliação das condições em `scope_attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
log_attributes	object/array	Retorna somente os registros de log com os atributos de log especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os registros de log. Valores possíveis de operator: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.

Parameter	Type	Description
log_attributes_evaltype	integer	Método de avaliação das condições em `log_attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
filter	object	Retorna somente os resultados que correspondem exatamente ao filtro especificado. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são um único valor ou uma matriz de valores a serem comparados. Propriedades compatíveis: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
search	object	Retorna os resultados que correspondem ao padrão especificado (sem distinção entre maiúsculas e minúsculas). Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são strings a serem pesquisadas. Se nenhuma opção adicional for especificada, será realizada uma pesquisa `LIKE "%...%"`. Propriedades compatíveis: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Número de registros a ignorar antes de retornar o resultado.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a quantidade de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando logs de erro de um trace

Recupere os registros de log com a severidade `ERROR` (números de severidade 17-20) associados ao trace "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
```

```
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

Pesquisando mensagens de log

Recupere o horário, o serviço e a mensagem dos 10 registros de log mais recentes cuja mensagem contenha ”timeout”.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```

Veja também

- Trace de APM
- Span de APM
- Métrica de APM

Origem

CApmLog::get() em *ui/include/classes/api/services/CApmLog.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. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
auditid	ID	ID da entrada do log de auditoria. Gerado usando o algoritmo CUID.
userid	ID	ID do usuário autor da entrada do log de auditoria.
username	string	Nome de usuário do autor da entrada do log de auditoria.
clock	timestamp	Timestamp de criação da entrada do log de auditoria.
ip	string	Endereço IP do autor da entrada do log de auditoria.

Propriedade	Tipo	Descrição
action	integer	Ação da entrada do log de auditoria. Valores possíveis: 0 - Adicionar; 1 - Atualizar; 2 - Excluir; 4 - Logout; 7 - Executar; 8 - Login; 9 - Falha no login; 10 - Limpar histórico; 11 - Atualizar configuração; 12 - Push.

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.
resource_cuid	ID	Identificador exclusivo do recurso da entrada do log de auditoria, gerado usando o algoritmo CUID.
resourcename	string	Nome legível do recurso da entrada do log de auditoria.
recordsetid	ID	ID do recordset da entrada do log de auditoria. Os registros de log de auditoria criados durante a mesma operação terão o mesmo ID de recordset. Gerado usando o algoritmo CUID.

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 informados.

> **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 User roles para obter 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` tiver sido usado.

Exemplos

Recuperar o log de auditoria

Recupere os dois registros mais recentes do log de auditoria.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resource_cuid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resource_cuid": "0",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

Veja também

- Objeto de log de auditoria

Fonte

CAuditLog::get() em *ui/include/classes/api/services/CAuditLog.php*.

Macro de usuário

Esta classe foi projetada para trabalhar com macros de usuário em nível de host e globais.

Referências de 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 gravação* 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 é gerenciada pelo usuário; 1 - Macro é 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 - 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 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)` 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”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "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. Consulte Funções de usuário para obter 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 o valor "public".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "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 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 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. Consulte Funções de usuário para obter 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 informados.

> **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 obter 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 os grupos de hosts aos quais a macro de host pertence na propriedade `hostgroups`. Usado apenas ao recuperar macros de host.
selectHosts	query	Retorna os hosts aos quais a macro de host pertence na propriedade `hosts`. Usado apenas ao recuperar macros de host.
selectTemplateGroups	query	Retorna os grupos de templates aos quais a macro de template pertence na propriedade `templategroups`. Usado apenas ao recuperar macros de template.
selectTemplates	query	Retorna os templates aos quais a macro de host pertence na propriedade `templates`. Usado apenas ao recuperar macros de host.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `macro`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

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 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 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. Consulte User roles para obter 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 maintenance possui as seguintes propriedades.

Property	Type	Description
maintenanceid	ID	ID da maintenance. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da maintenance. Comportamento da propriedade: - *obrigatório* para operações de criação
active_since	timestamp	Momento em que a maintenance se torna ativa (inclusive). O valor informado será arredondado para baixo para minutos. Comportamento da propriedade: - *obrigatório* para operações de criação
active_till	timestamp	Momento em que a maintenance deixa de estar ativa (exclusive). O valor informado será arredondado para baixo para minutos. Comportamento da propriedade: - *obrigatório* para operações de criação
description	string	Descrição da maintenance.
maintenance_type	integer	Tipo de maintenance. Valores possíveis: 0 - *(padrão)* com coleta de dados; 1 - sem coleta de dados.
tags_evaltype	integer	Método de avaliação da tag de problema. Valores possíveis: 0 - *(padrão)* And/Or; 2 - Or.

Período de tempo

O objeto de período de tempo é usado para definir períodos em que a manutenção deve entrar em vigor. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
period	integer	Duração do período de manutenção em segundos. O valor fornecido será arredondado para baixo para minutos. Valores possíveis: 300 - 86399940. Padrão: 3600.

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”

Nome do evento

O objeto de nome do evento é usado para definir quais problemas devem ser suprimidos quando a manutenção entrar em vigor. Os nomes dos eventos só podem ser especificados se `maintenance_type` do objeto Maintenance estiver definido como ”com coleta de dados”. Ele tem as seguintes propriedades.

Property	Type	Description
operator	integer	Operador da condição. Valores possíveis: 2 - *(padrão)* Contém; 3 - Não contém.

Property	Type	Description
value	string	Valor do nome do evento. Comportamento da propriedade: - *obrigatório*

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 `maintenance_type` do objeto Maintenance estiver definido como "com coleta de dados". Ele tem as seguintes propriedades.

Property	Type	Description
tag	string	Nome da tag de problema. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição. Valores possíveis: 0 - Igual a; 1 - Diferente de; 2 - *(padrão)* Contém; 3 - Não contém.
value	string	Valor da tag de problema.

maintenance.create

Descrição

`object maintenance.create(object/array maintenances)`

Este método permite criar novas manutenções.

Note:
Este método está disponível apenas para 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)` 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 a serem colocados em 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 a serem colocados em manutenção. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório* se `groups` não estiver definido

Parâmetro	Tipo	Descrição
triggers	object/array	Triggers a serem colocados em manutenção. Os triggers devem ter apenas a propriedade `triggerid` definida. Comportamento do parâmetro: - *obrigatório* se `groups` e `hosts` não estiverem definidos - *suportado* se maintenance_type=0 (com coleta de dados)
event_names	object/array	Nomes de eventos para definir os problemas a serem incluídos na manutenção. Comportamento do parâmetro: - *suportado* se maintenance_type=0 (com coleta de dados)
tags	object/array	Tags de problemas para definir os problemas a serem incluídos na manutenção. Comportamento do parâmetro: - *suportado* se `maintenance_type` do objeto Maintenance estiver definido como "com coleta de dados"
timeperiods	object/array	Períodos de tempo da manutenção. Comportamento do parâmetro: - *obrigatório*

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 "20" e com as tags de problema **service:mysqld** e **error**.
Ela deve estar ativa de 17.03.2026 a 17.03.2027, entrar em vigor todos os domingos às 18:00 e durar uma hora.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
```

```
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Criando uma manutenção ad hoc

Crie uma manutenção ad hoc de uma hora com coleta de dados para o grupo de hosts com ID ”4”, o trigger com ID ”12345” e o nome do evento **Printer: ink low**.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
```

```
            "4"
        ]
    },
    "id": 1
}
```

Veja também

- Host
- Grupo de hosts
- Trigger
- Nome do evento
- Tag do problema
- 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 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 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas as manutenções atribuídas aos grupos de hosts especificados.
hostids	ID/array	Retorna apenas as manutenções atribuídas aos hosts especificados.
triggerids	ID/array	Retorna apenas as manutenções atribuídas aos triggers especificados.
maintenanceids	ID/array	Retorna apenas as manutenções com os IDs especificados.
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.
selectTriggers	query	Retorna uma propriedade `triggers` com os triggers atribuídos à manutenção.
selectEventNames	query	Retorna uma propriedade `event_names` com os nomes de eventos 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 especificadas. Valores possíveis: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Esses parâmetros estão 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 manutenções

Recupere todas as manutenções configuradas, os dados sobre o escopo de manutenção atribuído (grupo de hosts, host, trigger, nome do evento, tags) e os períodos definidos.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Manutenção regular do banco de dados",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "20",
          "name": "Bancos de dados",
          "flags": "0",
          "uuid": "748ad4d098d447d492bb935c907f652f"
        }
      ],
      "triggers": [],
      "event_names": [],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "4",
          "every": "1",
          "month": "4095",
          "dayofweek": "0",
          "day": "1",
          "start_time": "64800",
          "period": "3600",
          "start_date": "0"
        }
      ]
    },
    {
      "maintenanceid": "2",
      "name": "Ad hoc: Impressora: tinta ESGOTADA",
      "maintenance_type": "0",
```

```
      "description": "",
      "active_since": "1792057740",
      "active_till": "1792061340",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "4",
          "name": "Servidores Zabbix",
          "flags": "0",
          "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
      ],
      "triggers": [],
      "event_names": [
        {
          "operator": "2",
          "value": "Impressora: tinta ESGOTADA"
        }
      ],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "0",
          "every": "1",
          "month": "0",
          "dayofweek": "0",
          "day": "0",
          "start_time": "0",
          "period": "3600",
          "start_date": "1792057740"
        }
      ]
    }
  ],
  "id": 1
}
```


Veja também

- Host
- Grupo de hosts
- Trigger
- Nome do evento
- Tag do problema
- 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 maintenances existentes.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter 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 demais permanecerão inalteradas.

Além das propriedades padrão de manutenção, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts que substituirão 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 que substituirão os hosts atuais. Os hosts devem ter apenas a propriedade hostid definida. Comportamento do parâmetro: - *obrigatório* se groups não estiver definido
triggers	object/array	Triggers que substituirão os triggers atuais. Os triggers devem ter apenas a propriedade triggerid definida. Comportamento do parâmetro: - *obrigatório* se groups e hosts não estiverem definidos - *suportado* se maintenance_type=0 (com coleta de dados)
event_names	object/array	Nomes de eventos que substituirão os nomes de eventos atuais. Comportamento do parâmetro: - *suportado* se maintenance_type=0 (com coleta de dados)
tags	object/array	Tags de problema que substituirão as tags atuais. Comportamento do parâmetro: - *suportado* se maintenance_type do objeto Maintenance estiver definido como "com coleta de dados"
timeperiods	object/array	Períodos de tempo de manutenção que substituirão os períodos atuais.

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
- Map element
  - Map element Host
  - Map element Host group
  - Map element Map
  - Map element Trigger
  - Map element Tag
  - Map element URL
- Map link
  - Map link trigger
  - Map link indicators
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

Métodos disponíveis:

- map.create - criar novos mapas
- map.delete - excluir mapas
- map.get - recuperar mapas
- map.update - atualizar mapas

Objeto de mapa

Os seguintes objetos estão diretamente relacionados à API de `map`.

Map

O objeto map 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

Propriedade	Tipo	Descrição
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.
background_scale	integer	Se a escala proporcional da imagem de fundo está habilitada. Valores possíveis: 0 - desabilitar escala; 1 - *(padrão)* habilitar escala da imagem.
expand_macros	integer	Se macros em rótulos devem ser expandidas 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 o alinhamento à grade está habilitado. Valores possíveis: 0 - desabilitar alinhamento à grade; 1 - *(padrão)* habilitar alinhamento à grade.
grid_show	integer	Se a grade deve ser exibida no mapa. Valores possíveis: 0 - não exibir a grade; 1 - *(padrão)* exibir a grade.
grid_size	integer	Tamanho da grade do mapa em pixels. Valores suportados: 20, 40, 50, 75 e 100. Padrão: 50.
highlight	integer	Se o destaque de ícones está habilitado. Valores possíveis: 0 - destaque desabilitado; 1 - *(padrão)* destaque habilitado.
iconmapid	ID	ID do mapa de ícones usado no mapa.
label_format	integer	Se rótulos avançados estão habilitados. Valores possíveis: 0 - *(padrão)* desabilitar rótulos avançados; 1 - habilitar rótulos avançados.
label_location	integer	Localização do rótulo do elemento do mapa. Valores possíveis: 0 - *(padrão)* inferior; 1 - esquerda; 2 - direita; 3 - superior.

Propriedade	Tipo	Descrição
label_string_host	string	Rótulo personalizado para elementos de host. Comportamento da propriedade: - *obrigatório* se `label_type_host` estiver definido como ”custom”
label_string_hostgroup	string	Rótulo personalizado para elementos de grupo de hosts. Comportamento da propriedade: - *obrigatório* se `label_type_hostgroup` estiver definido como ”custom”
label_string_image	string	Rótulo personalizado para elementos de imagem. Comportamento da propriedade: - *obrigatório* se `label_type_image` estiver definido como ”custom”
label_string_map	string	Rótulo personalizado para elementos de mapa. Comportamento da propriedade: - *obrigatório* se `label_type_map` estiver definido como ”custom”
label_string_trigger	string	Rótulo personalizado para elementos de trigger. Comportamento da propriedade: - *obrigatório* se `label_type_trigger` estiver definido como ”custom”
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.

Propriedade	Tipo	Descrição
label_type_trigger	integer	Tipo de rótulo para elementos de trigger. Valores possíveis: 0 - rótulo; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada; 5 - personalizado.
markelements	integer	Se os elementos do mapa que mudaram recentemente de status devem ser destacados. Valores possíveis: 0 - *(padrão)* não destacar elementos; 1 - destacar elementos.
severity_min	integer	Severidade mínima dos triggers que serão exibidos no mapa. Consulte a propriedade `severity` do trigger para uma lista de severidades suportadas.
show_element_label	integer	Como exibir rótulos de elementos por padrão. Valores possíveis: 0 - sempre exibir; 1 - *(padrão)* ocultar automaticamente.
show_link_label	integer	Como exibir rótulos de 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*

Propriedade	Tipo	Descrição
elements	array	Objeto de dados Elemento. Comportamento da propriedade: - *obrigatório* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
elementtype	integer	Tipo de elemento do mapa. Valores possíveis: 0 - host; 1 - mapa; 2 - trigger; 3 - grupo de hosts; 4 - imagem. Comportamento da propriedade: - *obrigatório*
iconid_off	ID	ID da imagem usada para exibir o elemento no estado padrão. Comportamento da propriedade: - *obrigatório*
areatype	integer	Como os hosts do grupo de hosts devem ser exibidos. Valores possíveis: 0 - *(padrão)* o elemento do grupo de hosts ocupará todo o mapa; 1 - o elemento do grupo de hosts terá um tamanho fixo.
elementsubtype	integer	Como um elemento de grupo de hosts deve ser exibido em um mapa. Valores possíveis: 0 - *(padrão)* exibir o grupo de hosts como um único elemento; 1 - exibir cada host do grupo separadamente.
evaltype	integer	Condição de filtragem de tag do elemento do mapa método de avaliação. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
height	integer	Altura do elemento de grupo de hosts de tamanho fixo em pixels. Padrão: 200.
iconid_disabled	ID	ID da imagem usada para exibir elementos de mapa desabilitados. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
iconid_maintenance	ID	ID da imagem usada para exibir elementos de mapa em manutenção. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
iconid_on	ID	ID da imagem usada para exibir elementos de mapa com problemas. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
label	string	Rótulo do elemento.

Propriedade	Tipo	Descrição
label_location	integer	Localização do rótulo do elemento do mapa. Valores possíveis: -1 - *(padrão)* localização padrão; 0 - inferior; 1 - esquerda; 2 - direita; 3 - superior.
permission	integer	Tipo de nível de permissão. Valores possíveis: -1 - nenhum; 2 - somente leitura; 3 - leitura-escrita.
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.

Objeto Map do elemento do mapa

O objeto Map do elemento do mapa define um elemento de mapa.

Propriedade	Tipo	Descrição
sysmapid	ID	ID do mapa.

Elemento de mapa Trigger

O objeto de elemento de mapa Trigger define um ou mais elementos de trigger.

Propriedade	Tipo	Descrição
triggerid	ID	ID do trigger.

Tag do elemento do mapa

O objeto Tag do elemento do mapa possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do elemento do mapa. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição da tag do elemento do mapa. Valores possíveis: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
value	string	Valor da tag do elemento do mapa.

URL do elemento do mapa

O objeto de URL do elemento do mapa define um link clicável que estará disponível para um elemento de mapa específico. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmapelementurlid	ID	ID da URL do elemento do mapa. Comportamento da propriedade: - *somente leitura*
name	string	Legenda do link. Comportamento da propriedade: - *obrigatório*
url	string	URL do link. Comportamento da propriedade: - *obrigatório*
selementid	ID	ID do elemento do mapa ao qual a URL pertence.

Link de mapa

O objeto de link de mapa define um link entre dois elementos do mapa. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
linkid	ID	ID do link do mapa. Comportamento da propriedade: - *somente leitura*
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 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 link de mapa

O objeto de indicadores de link 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-escrita. 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 - centro; 1 - esquerda; 2 - 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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte Funções de usuário para obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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 correspondentes do mapa e expande macros em todas as URLs dos elementos do mapa.
selectIconMap	query	Retorna uma propriedade `iconmap` com o mapa de ícones usado no mapa.
selectLinks	query	Retorna uma propriedade `links` com os links do mapa entre os elementos.
selectSelements	query	Retorna uma propriedade `selements` com os elementos do mapa.
selectUrls	query	Retorna uma propriedade `urls` com as URLs do mapa.
selectUsers	query	Retorna uma propriedade `users` com os usuários com os quais o mapa é compartilhado.
selectUserGroups	query	Retorna uma propriedade `userGroups` com os grupos de usuários com os quais o mapa é compartilhado.
selectShapes	query	Retorna uma propriedade `shapes` com as formas do mapa.
selectLines	query	Retorna uma propriedade `lines` com as linhas do mapa.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `name`, `width`, `height`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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 de usuário. Consulte Funções de usuário para obter 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, 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 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 valores

O objeto mapa de valores possui as seguintes propriedades.

Propriedade	Tipo	Descrição
valuemapid	ID	ID do mapa de valores. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
hostid	ID	ID do host ou template ao qual o mapa de valores pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
name	string	Nome do mapa de valores. Comportamento da propriedade: - *obrigatório* para operações de criação
mappings	array	Mapeamentos de valores para o mapa de valores atual. O objeto de mapeamento é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* para operações de criação
uuid	string	Identificador único universal, usado para vincular mapas de valores importados a mapas de valores já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o mapa de valores pertencer a um template

Mapeamentos de valores

O objeto de mapeamentos de valores define os mapeamentos de valores do mapa de valores. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo de correspondência do mapeamento. Valores possíveis: 0 - *(padrão)* o mapeamento será aplicado se o valor for igual; 1 - o mapeamento será aplicado se o valor for maior ou igual[1]; 2 - o mapeamento será aplicado se o valor for menor ou igual[1]; 3 - o mapeamento será aplicado se o valor estiver em um intervalo (os intervalos são inclusivos; vários intervalos, separados por vírgula, podem ser definidos)[1]; 4 - o mapeamento será aplicado se o valor corresponder a uma expressão regular[2]; 5 - se nenhuma correspondência for encontrada, o mapeamento não será aplicado e o valor padrão será usado. Se `type` for definido como ”0”, ”1”, ”2”, ”3”, ”4”, então `value` não pode estar vazio. Se `type` for definido como ”5”, então `value` deve estar vazio.
value	string	Valor original. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”1”, ”2”, ”3”, ”4” - *suportado* se `type` for definido como ”5”
newvalue	string	Valor para o qual o valor original é mapeado. Comportamento da propriedade: - *obrigatório*

[1] suportado apenas para items com tipo de valor ”numérico sem sinal”, ”numérico flutuante”.
 [2] suportado apenas para items com tipo de valor ”caractere”.

valuemap.create

Descrição

`object valuemap.create(object/array valuemaps)`

Este método permite criar novos mapeamentos de valor.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(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 de usuário. Consulte User roles para obter 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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 value maps 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 User roles para obter 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*.

MFA

Esta classe foi projetada para trabalhar com métodos MFA (Autenticação Multifator).

Referências de objetos:

- MFA

Métodos disponíveis:

- mfa.create - criar novos métodos MFA
- mfa.delete - excluir métodos MFA
- mfa.get - recuperar métodos MFA
- mfa.update - atualizar métodos MFA

Objeto MFA

Os seguintes objetos estão diretamente relacionados à API `mfa`.

MFA

O objeto MFA (Autenticação Multifator) possui as seguintes propriedades.

Propriedade	Tipo	Descrição
mfaid	ID	ID do método MFA. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
type	integer	Tipo do método MFA. Valores possíveis: 1 - TOTP (Senhas de uso único baseadas em tempo); 2 - Duo Universal Prompt.
name	string	Nome exclusivo do método MFA. Comportamento da propriedade: - *obrigatório* para operações de criação
hash_function	integer	Tipo de função hash para geração de códigos TOTP. Valores possíveis: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”TOTP”
code_length	integer	Comprimento do código de verificação. Valores possíveis: 6 - 6 dígitos; 8 - 8 dígitos. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”TOTP”
api_hostname	string	Nome do host da API fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”
clientid	string	ID do cliente fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”

Propriedade	Tipo	Descrição
client_secret	string	Segredo do cliente fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”

mfa.create

Descrição

`object mfa.create(object/array MFA methods)`

Este método permite criar novos métodos de 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 User roles para obter 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 de 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 User roles para obter 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 de MFA de acordo com os parâmetros informados.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(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 os 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 as propriedades: `mfaid` - ID do método MFA; `type` - Tipo do método MFA.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `name`.
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 "%…%"`. 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 MFA methods)`

Este método permite atualizar métodos de 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 de usuário. Consulte Funções de usuário para obter 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*.

Métrica de APM

Esta classe foi projetada para trabalhar com métricas coletadas para o monitoramento do desempenho de aplicações (APM).

Referências de objetos:

- Métrica de APM
- Exemplar

Métodos disponíveis:

- apmmetric.get - recuperar pontos de dados da métrica

Objeto de métrica APM

Os objetos a seguir estão diretamente relacionados à API `apmmetric`.

Note:
Cada propriedade, exceto `type`, corresponde a uma coluna nas tabelas de métricas (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` e `otel_metrics_exponential_histogram`) da fonte de dados de telemetria. A maioria das propriedades corresponde à coluna com o mesmo nome (por exemplo, a propriedade `metric_name` corresponde à coluna `MetricName`). Quando os nomes são diferentes, as colunas da fonte de dados são especificadas nas tabelas abaixo. Para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis.

Métrica de APM

O objeto de métrica de APM representa um ponto de dados de uma métrica (o valor ou os valores agregados de uma medição em um momento específico).

O objeto de métrica de APM tem as seguintes propriedades:

Propriedade	Tipo	Descrição
type	integer	Tipo do ponto de dados. Valores possíveis: 0 - Gauge; 1 - Sum; 2 - Histogram; 3 - Histograma exponencial. O tipo é determinado pela tabela que contém o ponto de dados: `otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` ou `otel_metrics_exponential_histogram`. Comportamento da propriedade: - *somente leitura*
resource_attributes	object	Atributos do recurso de chave-valor do ponto de dados (por exemplo, `host.name: eabf8c2a5ddc`). Comportamento da propriedade: - *somente leitura*
resource_schema_url	string	URL, se houver, do schema do OpenTelemetry dos atributos do recurso. Comportamento da propriedade: - *somente leitura*
scope_name	string	Nome do escopo de instrumentação que produziu o ponto de dados. Comportamento da propriedade: - *somente leitura*
scope_version	string	Versão, se houver, do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
scope_attributes	object	Atributos de chave-valor do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
scope_schema_url	string	URL, se houver, do schema do OpenTelemetry do escopo de instrumentação e de seus atributos. Comportamento da propriedade: - *somente leitura*
service_name	string	Nome do serviço que produziu o ponto de dados (por exemplo, `zabbix-frontend`), obtido do atributo de recurso `service.name` do OpenTelemetry. Comportamento da propriedade: - *somente leitura*
metric_name	string	Nome da métrica à qual o ponto de dados pertence (por exemplo, `http.server.request.duration`). Comportamento da propriedade: - *somente leitura*
metric_description	string	Descrição, se houver, da métrica. Comportamento da propriedade: - *somente leitura*
metric_unit	string	Unidade, se houver, dos valores da métrica (por exemplo, `s` ou `By`; consulte UCUM). Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
attributes	object	Atributos do ponto de dados de chave-valor (por exemplo, `http.request.method: POST`). Comportamento da propriedade: - *somente leitura*
start_time_unix	integer	Hora Unix (em nanossegundos) em que o período de medição do ponto de dados começou. Comportamento da propriedade: - *somente leitura*
time_unix	integer	Hora Unix (em nanossegundos) do ponto de dados. Para pontos de dados *Sum*, *Histogram* e *Histograma exponencial*, corresponde ao fim do período de medição. Comportamento da propriedade: - *somente leitura*
value	float	Valor do ponto de dados. Aplica-se a pontos de dados *Gauge* e *Sum*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
flags	integer	Máscara de bits dos sinalizadores do ponto de dados. Se o bit menos significativo estiver definido (valor 1), nenhum valor foi registrado para o ponto de dados. Comportamento da propriedade: - *somente leitura*
exemplars	array	Matriz de objetos exemplar, ou seja, medições individuais mantidas como exemplos, com o trace e o span durante os quais foram registradas. Comportamento da propriedade: - *somente leitura*
aggregation_temporality	integer	Como os valores do ponto de dados são agregados ao longo do tempo. Aplica-se a pontos de dados *Sum*, *Histogram* e *Histograma exponencial*. Valores possíveis: 0 - *(padrão)* Não especificado; 1 - Delta (agregado ao longo do período desde o ponto de dados anterior); 2 - Cumulativo (agregado ao longo do período desde `start_time_unix`). Comportamento da propriedade: - *somente leitura*
is_monotonic	boolean	Indica se a soma pode apenas aumentar ou permanecer igual ao longo do tempo. Aplica-se a pontos de dados *Sum*. Padrão: `false`. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
count	integer	Número de medições no ponto de dados. Aplica-se a pontos de dados *Histogram* e *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
sum	float	Soma de todas as medições no ponto de dados. Aplica-se a pontos de dados *Histogram* e *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
bucket_counts	array	Matriz de inteiros com o número de medições em cada bucket do histograma. Os buckets são definidos por `explicit_bounds`. Aplica-se a pontos de dados *Histogram*. Padrão: `[]`. Comportamento da propriedade: - *somente leitura*
explicit_bounds	array	Matriz de números de ponto flutuante com os limites entre os buckets do histograma. Aplica-se a pontos de dados *Histogram*. Padrão: `[]`. Comportamento da propriedade: - *somente leitura*
min	float	Menor valor entre as medições no ponto de dados. Aplica-se a pontos de dados *Histogram* e *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
max	float	Maior valor entre as medições no ponto de dados. Aplica-se a pontos de dados *Histogram* e *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
scale	integer	Escala do histograma exponencial, que determina os limites e a resolução dos buckets. Uma escala maior fornece buckets mais estreitos e uma resolução mais precisa; uma escala menor fornece buckets mais largos e uma resolução mais grosseira. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
zero_count	integer	Número de medições no bucket zero do histograma exponencial. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
positive_bucket_counts	array	Matriz de inteiros com o número de medições positivas em cada bucket do histograma exponencial. Os limites dos buckets são determinados por `scale`; o índice do primeiro bucket é definido em `positive_offset`. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: `[]`. Comportamento da propriedade: - *somente leitura*
positive_offset	integer	Índice do primeiro bucket em `positive_bucket_counts`. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*
negative_bucket_counts	array	Matriz de inteiros com o número de medições negativas em cada bucket do histograma exponencial. Os limites dos buckets são determinados por `scale`; o índice do primeiro bucket é definido em `negative_offset`. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: `[]`. Comportamento da propriedade: - *somente leitura*
negative_offset	integer	Índice do primeiro bucket em `negative_bucket_counts`. Aplica-se a pontos de dados *Histograma exponencial*. Padrão: 0. Comportamento da propriedade: - *somente leitura*

Exemplar

O objeto exemplar tem as seguintes propriedades:

Propriedade	Tipo	Descrição
traceid	string	ID, se houver, do trace durante o qual a medição foi registrada. Coluna da fonte de dados: `Exemplars.TraceId`.
spanid	string	ID, se houver, do span durante o qual a medição foi registrada. Coluna da fonte de dados: `Exemplars.SpanId`.
value	float	Valor da medição. Coluna da fonte de dados: `Exemplars.Value`.
time_unix	integer	Horário Unix (em nanossegundos) em que a medição foi registrada. Coluna da fonte de dados: `Exemplars.TimeUnix`.
filtered_attributes	object	Atributos de chave-valor da medição que não estão incluídos nos `attributes` do ponto de dados. Coluna da fonte de dados: `Exemplars.FilteredAttributes`.

apmmetric.get

Descrição

`integer/array apmmetric.get(object parameters)`

O método permite recuperar pontos de dados de métricas 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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
time_from	timestamp	Retorna somente os pontos de dados cujo horário seja igual ou posterior ao horário especificado. Comportamento do parâmetro: - *obrigatório*
time_till	timestamp	Retorna somente os pontos de dados cujo horário seja anterior ao horário especificado. Comportamento do parâmetro: - *obrigatório*
types	integer/array	Retorna somente os pontos de dados dos tipos especificados. Consulte a página do objeto de métrica APM para obter uma lista dos tipos de métricas compatíveis.
with_flags_on	integer	Retorna somente os pontos de dados que tenham todos os sinalizadores especificados definidos. Para especificar vários sinalizadores, use a soma de seus valores.
with_flags_off	integer	Retorna somente os pontos de dados que não tenham nenhum dos sinalizadores especificados definido. Para especificar vários sinalizadores, use a soma de seus valores.
resource_attributes	object/array	Retorna somente os pontos de dados com os atributos de recurso especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os pontos de dados. Valores possíveis de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
resource_attributes_evaltype	integer	Método de avaliação das condições em `resource_attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.

Parâmetro	Tipo	Descrição
scope_attributes	object/array	Retorna somente os pontos de dados com os atributos de escopo especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os pontos de dados. Valores possíveis de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
scope_attributes_evaltype	integer	Método de avaliação das condições em `scope_attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
attributes	object/array	Retorna somente os pontos de dados com os atributos especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os pontos de dados. Valores possíveis de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
attributes_evaltype	integer	Método de avaliação das condições em `attributes`. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
filter	object	Retorna somente os resultados que correspondam exatamente ao filtro especificado. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são um único valor ou uma matriz de valores a serem comparados. Propriedades compatíveis: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.
search	object	Retorna os resultados que correspondam ao padrão especificado (sem distinção entre maiúsculas e minúsculas). Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são strings a serem pesquisadas. Se nenhuma opção adicional for especificada, será realizada uma pesquisa `LIKE "%…%"`. Propriedades compatíveis: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.

Parâmetro	Tipo	Descrição
offset	integer	Número de registros a ignorar antes de retornar o resultado.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna um dos seguintes:

- um array de objetos;
- a quantidade de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando pontos de dados do histograma

Recupere o ponto de dados mais recente da métrica de histograma `http.server.request.duration` do serviço `zabbix-frontend`.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
```

```
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```

Veja também

- Rastreamento APM
- Span APM
- Log APM

Fonte

CApmMetric::get() em *ui/include/classes/api/services/CApmMetric.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`.

Module

O objeto module possui as seguintes propriedades.

Property	Type	Description
moduleid	ID	ID do module conforme armazenado no banco de dados. Property behavior: - *read-only* - *required* para operações de atualização
id	string	ID exclusivo do module, conforme definido por um desenvolvedor no arquivo `manifest.json` do module. Valores possíveis para modules integrados: consulte a descrição da propriedade "type" em Dashboard widget. Property behavior: - *required* para operações de criação
relative_path	string	Caminho para o diretório do module relativo ao diretório do Zabbix frontend. Valores possíveis: `widgets/*` - para modules de widget integrados; `modules/*` - para modules de terceiros. Property behavior: - *required* para operações de criação

Property	Type	Description
status	integer	Indica se o module está habilitado ou desabilitado. Valores possíveis: 0 - *(default)* Desabilitado; 1 - Habilitado.
config	object	Configuração do module.

module.create

Descrição

`object module.create(object/array modules)`

Este método permite instalar novos módulos do 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 da função de usuário. Consulte Funções de usuário para obter mais informações.
 Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend, o usuário não poderá chamar este método.

Attention:
Os arquivos dos módulos devem ser descompactados manualmente nos subdiretórios corretos, correspondentes 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 funções de usuário. Consulte Funções de usuário para obter mais informações.
Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend, o usuário não poderá chamar este método.

Attention:
Os arquivos dos módulos 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 da função de usuário. Consulte Funções de usuário para obter mais informações.
> Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend, o usuário não poderá chamar este método.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
moduleids	ID/array	Retorna apenas os módulos com os IDs fornecidos.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `moduleid`, `relative_path`.
countOutput	boolean	Estes parâmetros são descritos em Comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando um módulo pelo ID

Recupere todos os dados sobre os módulos "1", "2" e "25".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

Veja também

- Módulo
- Widget de dashboard
- Módulos do frontend

Fonte

CModule::get() em *ui/include/classes/api/services/CModule.php*.

module.update

Descrição

`object module.update(object/array modules)`

Este método permite atualizar módulos existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de funções de usuário. Consulte Funções de usuário para obter mais informações.
> Se `$ZBX_FEATURE_FLAGS['modules_config_enabled']` estiver definido como `false` no arquivo de configuração do frontend, o usuário não poderá chamar este método.

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 objetos:

- Nó de alta disponibilidade

Métodos disponíveis:

- hanode.get - recuperar nós

Objeto de nó de alta disponibilidade

O objeto a seguir está relacionado à operação de um cluster de alta disponibilidade de servidores Zabbix.

Nó de alta disponibilidade

> **Note:**
> Os nós são criados pelo server Zabbix e não podem ser modificados via API.

O objeto Nó de alta disponibilidade possui as seguintes propriedades.

Property	Type	Description
ha_nodeid	ID	ID do nó.

Property	Type	Description
name	string	Nome atribuído ao nó, usando a entrada de configuração HANodeName de `zabbix_server.conf`. Vazio para um server em execução no modo standalone.
address	string	Endereço IP ou nome DNS de onde o nó se conecta.
port	integer	Porta na qual o nó está em execução.
lastaccess	integer	Tempo do heartbeat, ou seja, o horário da última atualização recebida 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 do cluster de Alta disponibilidade de acordo com os parâmetros informados.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. Consulte User roles para obter 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 foi projetada para trabalhar com problemas.

Referências de objetos:

- Problem
  - Media type URL
- Problem tag

Métodos disponíveis:

- problem.get - recuperar problemas

Objeto problem

Os seguintes objetos estão diretamente relacionados à API `problem`.

Problema

> **Note:**
> 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 de problema.
source	integer	Tipo do evento de 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 de problema. Valores possíveis se `source` estiver definido como "evento criado por um trigger": 0 - trigger. 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.
clock	timestamp	Hora em que o evento de problema foi criado.
ns	integer	Nanosegundos em que o evento de 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 - médio; 4 - alto; 5 - desastre.

Propriedade	Tipo	Descrição
suppressed	integer	Se o problema está suprimido. Valores possíveis: 0 - problema está em estado normal; 1 - problema está suprimido.
opdata	string	Dados operacionais com macros expandidas.
urls	array	URLs de tipos de mídia ativas.

URL do tipo de mídia

O objeto URL do tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da URL definida 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. 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 é usado para recuperar problemas não resolvidos. Também é possível, se especificado, recuperar adicionalmente problemas resolvidos recentemente. O período que determina quão antigo é "recentemente" é definido em *Administration* > *General*. Problemas que foram resolvidos antes desse período não são mantidos na tabela de problemas. Para recuperar problemas que foram resolvidos mais no passado, 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 de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parameter	Type	Description
eventids	ID/array	Retorna apenas problemas com os IDs informados.
groupids	ID/array	Retorna apenas problemas criados por objetos que pertencem aos grupos de hosts informados.

Parameter	Type	Description
hostids	ID/array	Retorna apenas problemas criados por objetos que pertencem aos hosts informados.
objectids	ID/array	Retorna apenas problemas criados pelos objetos informados.
source	integer	Retorna apenas problemas com o tipo informado. Consulte a página do objeto de evento de problema para obter uma lista dos tipos de evento suportados. Padrão: 0 - problema criado por um trigger.
object	integer	Retorna apenas problemas criados por objetos do tipo informado. Consulte a página do objeto de evento de problema para obter uma lista dos tipos de objeto suportados. Padrão: 0 - trigger.
acknowledged	boolean	Se definido como `true`, retorna apenas problemas reconhecidos.
action	integer	Retorna apenas problemas para os quais as ações de atualização de evento informadas foram executadas. Para várias ações, use a soma de quaisquer valores de bitmap aceitáveis como máscara de bits (por exemplo, 34 para reconhecer e suprimir o evento).
action_userids	ID/array	Retorna apenas problemas com os IDs informados dos usuários que executaram as ações de atualização do evento de problema.
suppressed	boolean	Se definido como `true`, retorna apenas problemas suprimidos.
symptom	boolean	Se definido como `true`, retorna apenas eventos de problema sintoma.
severities	integer/array	Retorna apenas problemas com as severidades de evento informadas. Aplica-se somente se object for trigger.
evaltype	integer	Método de avaliação de tags. Valores possíveis: 0 - *(padrão)* And/Or; 2 - Or.
tags	array	Retorna apenas problemas com as tags informadas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os problemas. Valores possíveis de operator: 0 - *(padrão)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
recent	boolean	Se definido como `true`, retorna problemas recentes, incluindo aqueles resolvidos dentro do `ok_period` e quaisquer problemas não resolvidos. Se definido como `false`, retorna apenas problemas não resolvidos.
eventid_from	string	Retorna apenas problemas com IDs maiores ou iguais ao ID informado.
eventid_till	string	Retorna apenas problemas com IDs menores ou iguais ao ID informado.
time_from	timestamp	Retorna apenas problemas que foram criados após ou no horário informado.
time_till	timestamp	Retorna apenas problemas que foram criados antes ou no horário informado.

Parameter	Type	Description
selectAcknowledges	query	Retorna uma propriedade `acknowledges` com as atualizações do problema. As atualizações do problema são ordenadas em ordem cronológica inversa. O objeto de atualização do problema tem 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; `maintenanceid` - `(ID)` ID da maintenance. 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 maintenances ativas e supressões manuais: `maintenanceid` - `(ID)` ID da maintenance; `userid` - `(ID)` ID do usuário que suprimiu o problema; `suppress_until` - `(integer)` tempo até o qual o problema está suprimido.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro informado. Aceita um objeto, em que as chaves são nomes de propriedades e os valores são um único valor ou um array de valores com os quais comparar. Não oferece suporte a propriedades do tipo de dados `text` data type.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. 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 dos seguintes:

- uma matriz de objetos;
- a contagem dos objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando eventos de problema de trigger

Recupere eventos recentes do trigger ”15112.”

Request:

```
{
    "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
}
```

Response:

```
{
    "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",
            "suppressed": "1",
            "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",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
```

```
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

Recuperando problemas reconhecidos por um usuário especificado

Recuperando problemas reconhecidos pelo usuário com ID=10

Request:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
```

```
        }
    ],
    "id": 1
}
```

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 - 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 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 tempo 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 aos já existentes. Gerado automaticamente, se não for 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 gráfico.

> **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)` 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 gráfico. 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 gráfico.

> **Note:**
> Este método está disponível apenas para 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 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 gráfico de acordo com os parâmetros informados.

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 obter 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 correspondência. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o protótipo de gráfico pertence; `hostid` - ID do host ao qual o protótipo de gráfico pertence.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

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 prototypes de gráfico 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 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 objetos:

- 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

Vinculação de grupo

O objeto de vinculação 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. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

Interface personalizada

Interfaces personalizadas são suportadas se `custom_interfaces` do objeto de protótipo de host estiver definido como ”usar interfaces personalizadas de 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*
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”
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”

Propriedade	Tipo	Descrição
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 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)` 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 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 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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.
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 foi descoberto. Possui as seguintes propriedades: `host` - `(string)` ID do host; `parent_hostid` - `(string)` ID do protótipo de host do qual o protótipo de host foi criado; `status` - `(int)` status de 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 de 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 da regra LLD pai à 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.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com as tags que estão nos templates vinculados.
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	

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

Recuperando protótipos de host de uma regra LLD

Recupere todos os protótipos de host, seus links de grupo, protótipos de grupo e tags de uma regra LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "discover": "0",
            "custom_interfaces": "1",
            "uuid": "051a1469d4d045cbbf818fcc843a352e",
            "flags": "2",
            "inventory_mode": "-1",
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "groupLinks": [
```

```
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "available": "0",
                    "error": "",
                    "errors_from": "0",
                    "disable_until": "0",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```


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 host prototypes existentes.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object/array)` Propriedades do protótipo de host a serem atualizadas.

A propriedade `hostid` deve ser definida para cada protótipo de host, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do protótipo de host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupLinks	array	Vínculos de grupo para substituir os vínculos de grupo atuais no protótipo de host. Comportamento do parâmetro: - *somente leitura* para objetos herdados
groupPrototypes	array	Protótipos de grupo para substituir os protótipos de grupo existentes no protótipo de host. Todos os protótipos de grupo que não estiverem listados na solicitação serão removidos. Comportamento do parâmetro: - *somente leitura* para objetos herdados
macros	object/array	Macros de usuário para substituir as macros de usuário atuais. Todas as macros que não estiverem listadas na solicitação serão removidas.
tags	object/array	Tags do protótipo de host para substituir as tags atuais. Todas as tags que não estiverem listadas na solicitação serão removidas. Comportamento do parâmetro: - *somente leitura* para objetos herdados
interfaces	object/array	Interfaces personalizadas do protótipo de host para substituir as interfaces atuais. O objeto de interface personalizada deve conter todos os seus parâmetros. Todas as interfaces que não estiverem listadas na solicitação serão removidas. Comportamento do parâmetro: - *suportado* se `custom_interfaces` do objeto protótipo de host estiver definido como "usar interfaces personalizadas 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 de protótipo de item tem as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do protótipo de item. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
delay	string	Intervalo de atualização do protótipo de item. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário e macros LLD. Quando usado, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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 estiver 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), ”Telemetry query” (24), ou se type estiver 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 a um host e type estiver definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” ou ”SNMP agent” - *com suporte* se o protótipo de item pertencer a um host e type estiver 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. Oferece suporte a 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; 24 - Telemetry query. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
url	string	String de URL. Oferece suporte a 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` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
value_type	integer	Tipo de informação do protótipo de item. Valores possíveis: 0 - ponto flutuante numérico; 1 - caractere; 2 - log; 3 - inteiro sem sinal numérico; 4 - texto; 5 - binário; 6 - JSON. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
allow_traps	integer	Permite que o valor seja preenchido de forma semelhante ao item trapper. 0 - *(padrão)* Não aceitar dados recebidos; 1 - Aceitar dados recebidos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent”

Propriedade	Tipo	Descrição
authtype	integer	Método de autenticação. Valores possíveis se `type` estiver definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` estiver definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver 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 da resposta durante a coleta de dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
headers	array	Matriz de cabeçalhos que serão enviados ao executar uma solicitação HTTP. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
history	string	Unidade de tempo durante a qual os dados históricos 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: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”IPMI agent” e `key_` não estiver definido como ”ipmi.get” - *com suporte* se `type` estiver definido como ”IPMI agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
jmx_endpoint	string	String de conexão personalizada do JMX agent. Padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”JMX agent”
logtimefmt	string	Formato da hora nas entradas de log. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”log” - *somente leitura* para objetos herdados
master_itemid	ID	ID do item mestre. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Dependent item” - *somente leitura* para objetos herdados
output_format	integer	Define se a resposta deve ser convertida para JSON. Valores possíveis: 0 - *(padrão)* Armazenar como bruta; 1 - Converter para JSON. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
params	string	Parâmetros adicionais dependendo do tipo de 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; - script para protótipos de item Script e Browser. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` estiver definido como ”Script” ou ”Browser”)
parameters	object/array	Parâmetros adicionais se `type` estiver definido como ”Script” ou ”Browser”. Matriz de objetos com as propriedades `name` e `value`, em que `name` deve ser exclusivo. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”JMX agent” e `username` estiver definido - *com suporte* se `type` estiver definido como ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)

Propriedade	Tipo	Descrição
post_type	integer	Tipo do corpo dos dados de postagem armazenado na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos. 2 - Dados JSON. 3 - Dados XML. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
posts	string	Dados do corpo da solicitação HTTP(S). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”Dados JSON” ou ”Dados XML” - *com suporte* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”Dados brutos” - *somente leitura* para objetos herdados
privatekey	string	Nome do arquivo da chave privada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”chave pública”
publickey	string	Nome do arquivo da chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”chave pública”
query_fields	array	Matriz de campos de consulta que serão enviados ao executar uma solicitação HTTP. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo do método de solicitação. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
retrieve_mode	integer	Parte da resposta que deve ser armazenada. Valores possíveis se `request_method` estiver definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - O corpo e os cabeçalhos serão armazenados. Valores possíveis se `request_method` estiver definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo da chave SSL pública. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_file	string	Caminho do arquivo da chave SSL privada. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_password	string	Senha do arquivo da chave SSL. Comportamento da propriedade: - *com suporte* se `type` estiver 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 sem suporte.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Também oferece suporte a macros de usuário ou macros LLD como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
templateid	ID	ID do protótipo de item do template pai. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da solicitação de coleta de dados do item. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m). Também aceita macros de usuário e macros LLD. O intervalo de valores possíveis é: 1-600s. Padrão: ”” - usar as configurações do proxy/globais. Comportamento da propriedade: - *com suporte* se `type` estiver 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

Propriedade	Tipo	Descrição
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Zabbix trapper”, ou se `type` estiver definido como ”HTTP agent” e `allow_traps` estiver definido como ”Permitir dados recebidos”. Se a propriedade for omitida, ela será definida automaticamente como a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}` (valor padrão). Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Se a propriedade for definida como uma string vazia, as conexões recebidas serão negadas de todos os hosts.
trends	string	Unidade de tempo durante a qual os dados de tendências devem ser armazenados. Também aceita macro de usuário e macro LLD. Padrão: 365d. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”ponto flutuante numérico” ou ”inteiro sem sinal numérico”
units	string	Unidades do valor. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”ponto flutuante numérico” ou ”inteiro sem sinal numérico” - *somente leitura* para objetos herdados
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” ou ”TELNET agent”, ou se `type` estiver definido como ”JMX agent” e `password` estiver definido - *com suporte* se `type` estiver definido como ”Simple check”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)
uuid	string	Identificador universal exclusivo, usado para vincular protótipos de item importados aos já existentes. Gerado automaticamente se não for fornecido. Comportamento da propriedade: - *com suporte* se o protótipo de item pertencer a um template
valuemapid	ID	ID do mapa de valores associado. Comportamento da propriedade: - *com suporte* se `value_type` estiver definido como ”ponto flutuante numérico”, ”caractere” ou ”inteiro sem sinal numérico” - *somente leitura* para objetos herdados
verify_host	integer	Define se deve ser validado que o nome do host da conexão corresponde ao nome no certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
verify_peer	integer	Define se deve ser validado que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
discover	integer	Status da descoberta do protótipo de item. Valores possíveis: 0 - *(padrão)* novos itens serão descobertos; 1 - novos itens não serão descobertos e os itens existentes serão marcados como perdidos.
query	object	Configuração da consulta de telemetria. Define o que o item consulta na fonte de dados de telemetria, como filtra os dados e como agrega o resultado. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Telemetry query” - *somente leitura* para objetos herdados ou descobertos
time_shift	string	Quanto para trás, antes do horário atual, termina o intervalo de tempo consultado pelo item. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30`s`, 1`m`, 2`h`, 1`d`). Também aceita macros de usuário e macros LLD. O intervalo de valores possíveis é: 0-86400s. Padrão: 15s. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”
lookback_limit	string	Quanto para trás, antes do horário atual, o item consulta dados quando não há uma consulta anterior da qual continuar (por exemplo, na primeira execução do item) ou quando a consulta anterior é mais antiga que `delay` (por exemplo, porque o Zabbix server estava indisponível). Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30`s`, 1`m`, 2`h`, 1`d`). Também aceita macros de usuário e macros LLD. O intervalo de valores possíveis é: 1-259200s. Padrão: 10m. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”

Propriedade	Tipo	Descrição
granularity	string	Duração de um segmento no qual o intervalo de tempo completo da consulta é dividido. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d). Também aceita macros de usuário e macros LLD. O intervalo de valores possíveis é: 1-86400s. Padrão: 15s. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Telemetry query”

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*

Consulta de telemetria

O objeto de consulta de telemetria tem as seguintes propriedades.

Property	Type	Description
signal_type	integer	Tipo de sinal consultado pelo item. Valores possíveis: 0 - Traces; 1 - Métricas; 2 - Logs. Comportamento da propriedade: - *obrigatório*
metric_point_type	integer	Tipo de ponto de métrica consultado pelo item. Valores possíveis: 0 - Soma; 1 - Gauge; 2 - Histograma; 3 - Histograma exponencial. Comportamento da propriedade: - *obrigatório* se `signal_type` estiver definido como ”Metrics”
columns	array	Array de colunas de dados de telemetria consultadas pelo item. Pode estar vazio. Comportamento da propriedade: - *obrigatório*
aggregated_columns	array	Array de colunas agregadas de dados de telemetria consultadas pelo item e de como ele calcula um valor a partir dos registros das colunas. Comportamento da propriedade: - *obrigatório*
filter	object	Filtro de consulta de telemetria para os registros de colunas na consulta. Comportamento da propriedade: - *obrigatório*

Colunas de dados de telemetria

O objeto de colunas de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria consultada pelo item. Consulte colunas de dados de telemetria compatíveis. Para várias colunas de valor único, cada valor de `column` deve ser exclusivo. Para várias colunas que contêm pares de chave-valor, cada par de `column` e `attribute_key` deve ser exclusivo.
attribute_key	string	Nome da chave consultada pelo item na coluna que contém pares de chave-valor. Para várias colunas que contêm pares de chave-valor, cada par de `column` e `attribute_key` deve ser exclusivo. Comportamento da propriedade: - *obrigatório* se `column` estiver definido como uma coluna que contém pares de chave-valor

Colunas agregadas de dados de telemetria

O objeto de colunas agregadas de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria consultada pelo item. Consulte colunas de dados de telemetria compatíveis. Para várias colunas, cada valor de `column` deve ser exclusivo. Esta propriedade não é compatível quando `function` está definido como ”Count”. Comportamento da propriedade: - *obrigatória* se `function` estiver definido como ”Min”, ”Max”, ”Average”, ”Sum” ou ”Percentile”
function	integer	Função que o item aplica aos registros da coluna. Valores possíveis: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Comportamento da propriedade: - *obrigatória*
parameters	array	Percentil que o item calcula a partir dos registros da coluna. Valor possível: um número de 0 a 100, fornecido como uma string. Comportamento da propriedade: - *obrigatória* se `function` estiver definido como ”Percentile”
alias	string	Nome exclusivo do valor calculado, usado para identificá-lo no valor do item. Cada valor de `alias` deve ser exclusivo. Comportamento da propriedade: - *obrigatória*

Filtro de dados de telemetria

O objeto de filtro de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
conditions	array	Condições do filtro de dados de telemetria que um registro em uma coluna de dados de telemetria deve atender. Comportamento da propriedade: - *obrigatório*
evaltype	integer	Método de avaliação das condições do filtro. Valores possíveis: 0 - And/Or; 1 - And; 2 - Or; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*

Property	Type	Description
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que fazem referência a condições específicas do filtro por meio de 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 façam referência a condições específicas do filtro por meio de seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer sem uso ou ser omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”Expressão personalizada”

Condições de filtro de dados de telemetria

O objeto de condições de filtro de dados de telemetria tem as seguintes propriedades.

Property	Type	Description
column	string	Nome da coluna de dados de telemetria à qual a condição de filtro se aplica. Consulte colunas de dados de telemetria compatíveis. Comportamento da propriedade: - *obrigatório*
attribute_key	string	Nome da chave à qual a condição se aplica, dentro da coluna que contém pares de chave-valor. Comportamento da propriedade: - *obrigatório* se `column` estiver definida como uma coluna que contém pares de chave-valor
operator	integer	Operador da condição de filtro. Valores possíveis para colunas de valor único: 0 - É igual a; 1 - Não é igual a; 2 - Contém; 3 - Não contém. Valores possíveis para colunas que contêm pares de chave-valor: 0 - É igual a; 1 - Não é igual a; 12 - Existe. Comportamento da propriedade: - *obrigatório*
value	string	Valor a ser comparado. Comportamento da propriedade: - *obrigatório* se `operator` estiver definido como ”É igual a”, ”Não é igual a”, ”Contém” ou ”Não contém”

Property	Type	Description
formulaid	string	ID exclusivo arbitrário usado para referenciar a condição em uma expressão personalizada. Pode conter apenas letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições de filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto Filtro de consulta de telemetria estiver definido como ”Expressão personalizada”

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. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template; 1 - host. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

Pré-processamento do protótipo de item

O objeto de pré-processamento do protótipo de item tem as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo da opção de pré-processamento. Valores possíveis: 1 - Multiplicador personalizado; 2 - Remover espaços à direita; 3 - Remover espaços à esquerda; 4 - Remover espaços; 5 - Expressão regular; 6 - Booleano para decimal; 7 - Octal para decimal; 8 - Hexadecimal para decimal; 9 - Alteração simples; 10 - Alteração por segundo; 11 - XPath XML; 12 - JSONPath; 13 - No intervalo; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar erro no JSON; 17 - Verificar erro no XML; 18 - Verificar erro usando expressão regular; 19 - Descartar se inalterado; 20 - Descartar se inalterado com heartbeat; 21 - JavaScript; 22 - Padrão do Prometheus; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 26 - Verificar não suportado; 27 - XML para JSON; 28 - Valor de SNMP walk; 29 - SNMP walk para JSON; 30 - Obter valor via SNMP. Comportamento da propriedade: - *obrigatória*
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Vários 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>]`, em que *pattern* é uma expressão regular e *scope* é um dos seguintes: -1 - corresponde a qualquer erro; 0 - verifica se a mensagem de erro corresponde a *pattern*; 1 - verifica se a mensagem de erro não corresponde a *pattern*. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Remover espaços à direita” (2), ”Remover espaços à esquerda” (3), ”Remover espaços” (4), ”Expressão regular” (5), ”XPath XML” (11), ”JSONPath” (12), ”No intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro no JSON” (16), ”Verificar erro no XML” (17), ”Verificar erro usando expressão regular” (18), ”Descartar se inalterado com heartbeat” (20), ”JavaScript” (21), ”Padrão do Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Verificar não suportado” (26), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Obter valor via SNMP” (30)

Propriedade	Tipo	Descrição
error_handler	integer	Tipo de ação usado em caso de falha da etapa de pré-processamento. Valores possíveis: 0 - A mensagem de erro é 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), ”XPath XML” (11), ”JSONPath” (12), ”No intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro no JSON” (16), ”Verificar erro no XML” (17), ”Verificar erro usando expressão regular” (18), ”Padrão do Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Verificar não suportado” (26), ”XML para JSON” (27), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Obter valor via SNMP” (30)
error_handler_params	string	Parâmetros do manipulador de erros. Comportamento da propriedade: - *obrigatório* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os parâmetros e manipuladores de erros a seguir são compatíveis com cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
1	Multiplicador personalizado	número			0, 1, 2, 3
2	Remover espaços à direita	lista de caracteres[2]			
3	Remover espaços à esquerda	lista de caracteres[2]			
4	Remover espaços	lista de caracteres[2]			
5	Expressão regular	padrão[3]	saída[2]		0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
6	Booleano para decimal				0, 1, 2, 3
7	Octal para decimal				0, 1, 2, 3
8	Hexadecimal para decimal				0, 1, 2, 3
9	Alteração simples				0, 1, 2, 3
10	Alteração por segundo				0, 1, 2, 3
11	XPath XML	caminho[4]			0, 1, 2, 3
12	JSONPath	caminho[4]			0, 1, 2, 3
13	No intervalo	mín.^1, 6^	máx.^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 no JSON	caminho[4]			0, 1, 2, 3
17	Verificar erro no XML	caminho[4]			0, 1, 2, 3
18	Verificar erro usando expressão regular	padrão[3]	saída[2]		0, 1, 2, 3
19	Descartar se inalterado				

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
20	Descartar se inalterado com heartbeat	segundos^5, 6^			
21	JavaScript	script[2]			
22	Padrão do Prometheus	padrão^6, 7^	`value`, `label`, `function`	saída^8, 9^	0, 1, 2, 3
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 pesquisa[2]	substituição[2]		
26	Verificar não suportado	escopo[1]	padrão^3, 6^		1, 2, 3
27	XML para JSON				0, 1, 2, 3
28	Valor de SNMP walk	OID[2]	Formato: 0 - Inalterado 1 - UTF-8 a partir de Hex-STRING 2 - MAC a partir de Hex-STRING 3 - Inteiro a partir de BITS		0, 1, 2, 3
29	SNMP walk para JSON[10]	Nome do campo[2]	Prefixo do OID[2]	Formato: 0 - Inalterado 1 - UTF-8 a partir de Hex-STRING 2 - MAC a partir de Hex-STRING 3 - Inteiro a partir de BITS	0, 1, 2, 3
30	Obter valor via SNMP	Formato: 1 - UTF-8 a partir de Hex-STRING 2 - MAC a partir de Hex-STRING 3 - Inteiro a partir de BITS			0, 1, 2, 3

[1] número inteiro ou de ponto flutuante
 [2] string
 [3] expressão regular
 [4] JSONPath ou XPath XML
 [5] número inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [6] macro de usuário, macro de LLD
 [7] padrão do Prometheus seguindo a sintaxe: `<metric name>{<label name>="<label value>", ...} == <value>`. Cada componente do padrão do Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser uma macro de usuário ou macro de LLD.
 [8] saída do Prometheus seguindo a sintaxe: `<label name>` (pode ser uma macro de usuário ou macro de LLD) se `label` estiver selecionado como segundo parâmetro.
 [9] Uma das funções de agregação: `sum`, `min`, `max`, `avg`,

`count` se `function` estiver selecionado como segundo parâmetro.
 [10] Compatível com vários registros no formato "Nome do campo,Prefixo do 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. Consulte User roles para obter 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 de usuário. Consulte Funções de usuário para obter 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 o protótipo de item for excluído.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::delete() 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 informados.

> **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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
discoveryids	ID/array	Retorna apenas os protótipos de item que pertencem às regras LLD fornecidas.
graphids	ID/array	Retorna apenas os protótipos de item que são usados nos protótipos de gráfico fornecidos.
hostids	ID/array	Retorna apenas os protótipos de item que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas os protótipos de item herdados de um template.
itemids	ID/array	Retorna apenas os protótipos de item com os IDs fornecidos.
monitored	boolean	Se definido como `true`, retorna apenas os protótipos de item habilitados que pertencem a hosts monitorados.

Parâmetro	Tipo	Descrição
templated	boolean	Se definido como `true`, retorna apenas os protótipos de item que pertencem a templates.
templateids	ID/array	Retorna apenas os protótipos de item que pertencem aos templates fornecidos.
triggerids	ID/array	Retorna apenas os protótipos de item que são usados nos protótipos de trigger fornecidos.
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)* o protótipo de item é descoberto, 1 - o 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 da regra LLD pai à qual o protótipo de item pertence.
selectGraphs	query	Retorna uma propriedade `graphs` com os protótipos de gráfico nos quais 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.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags que estão no template/host/templates vinculados.
selectTags	query	Retorna as tags do protótipo de item na propriedade `tags`.
selectTriggers	query	Retorna uma propriedade `triggers` com os protótipos de trigger nos quais 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.

Parâmetro	Tipo	Descrição
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.
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",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        },
        {
            "itemid": "10010",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10001",
            "name": "Processor load (1 min average per core)",
            "key_": "system.cpu.load[percpu,avg1]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
```

```
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Encontrando um item dependente

Encontre um item dependente para um ID de item específico.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.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": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
```

```
    ],
    "id": 1
}
```

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": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
```

```
            "status_codes": "",
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


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 item prototypes existentes.

> **Note:**
> Este método está disponível apenas para 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 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 de LLD

Esta classe foi projetada para trabalhar com protótipos de regras de descoberta de baixo nível.

Referências de objetos:

- Protótipo de regra de LLD
  - Cabeçalho HTTP
  - Campo de consulta HTTP
- Filtro de protótipo de regra de LLD
  - Condição de filtro de protótipo de regra de LLD
- Caminho de macro de protótipo de LLD
- Pré-processamento de protótipo de regra de LLD
- Substituições de protótipo de regra de LLD
  - Filtro de substituição de protótipo de regra de LLD
    - Condição de filtro de substituição de protótipo de regra de LLD
  - Operação de substituição de protótipo de regra de LLD
    - Status da operação de substituição de protótipo de regra de LLD
    - Descoberta de operação de substituição de protótipo de regra de LLD
    - Período de operação de substituição de protótipo de regra de LLD
    - Histórico de operação de substituição de protótipo de regra de LLD
    - Tendências de operação de substituição de protótipo de regra de LLD
    - Gravidade de operação de substituição de protótipo de regra de LLD
    - Tag de operação de substituição de protótipo de regra de LLD
    - Template de operação de substituição de protótipo de regra de LLD
    - Inventário de operação de substituição de protótipo de regra de LLD

Métodos disponíveis:

- discoveryruleprototype.create - cria novos protótipos de regras de LLD
- discoveryruleprototype.delete - exclui protótipos de regras de LLD
- discoveryruleprototype.get - recupera protótipos de regras de LLD
- discoveryruleprototype.update - atualiza protótipos de regras de LLD

Objeto de protótipo de regra de LLD

Os seguintes objetos estão diretamente relacionados à API `discoveryruleprototype`.

Protótipo de regra de LLD

O objeto de protótipo de regra de descoberta de baixo nível tem as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do protótipo de regra de LLD. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
ruleid	ID	ID da regra de LLD/protótipo de regra de LLD pai. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de criação
delay	string	Intervalo de atualização do protótipo de regra de LLD. Aceita um número de segundos (por exemplo, `1`, `60`, `300`), um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário. Se usada, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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` estiver 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` estiver definido como ”Zabbix agent (active)” (7) e `key_` não contiver ”mqtt.get”
hostid	ID	ID do host ao qual pertence o protótipo de regra de LLD. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
flags	integer	Origem do protótipo de regra de 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 de LLD. Comportamento da propriedade: - *obrigatório* se o protótipo de regra de LLD pertencer a um host e `type` estiver definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent” ou ”SNMP agent” - *com suporte* se o protótipo de regra de LLD pertencer a um host e `type` estiver definido como ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent”
key_	string	Chave do protótipo de regra de LLD. É necessária pelo menos uma macro de LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
name	string	Nome do protótipo de regra de LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
type	integer	Tipo do protótipo de regra de LLD. Valores possíveis: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 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` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
allow_traps	integer	Permite que o valor seja preenchido de forma semelhante ao item trapper. Valores possíveis: 0 - *(padrão)* Não aceitar dados recebidos; 1 - Aceitar dados recebidos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent”
authtype	integer	Método de autenticação. Valores possíveis se `type` estiver definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` estiver definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)

Propriedade	Tipo	Descrição
description	string	Descrição do protótipo de regra de LLD.
error	string	Texto do erro caso haja problemas ao atualizar o valor do protótipo de regra de LLD. Comportamento da propriedade: - *somente leitura*
follow_redirects	integer	Seguir redirecionamentos da resposta durante a coleta de dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *com suporte* se `type` estiver 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: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
http_proxy	string	String de conexão do proxy HTTP(S). Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”IPMI agent” e `key_` não estiver definido como ”ipmi.get” - *com suporte* se `type` estiver definido como ”IPMI agent” - *somente leitura* para objetos herdados
jmx_endpoint	string	String de conexão personalizada do JMX agent. Padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”JMX agent”
lifetime	string	Período após o qual os itens que não forem mais descobertos serão excluídos. Aceita um número de segundos (por exemplo, `1`, `60`, `300`), um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`) ou uma macro de usuário. Padrão: `7d`.
lifetime_type	integer	Cenário para excluir recursos de LLD perdidos. Valores possíveis: 0 - *(padrão)* Excluir após atingir o limite de tempo de vida; 1 - Não excluir; 2 - Excluir imediatamente.
enabled_lifetime	string	Período após o qual os itens que não forem mais descobertos serão desabilitados. Aceita um número de segundos (por exemplo, `1`, `60`, `300`), um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `2h`, `1d`) ou uma macro de usuário. Padrão: `0`.

Propriedade	Tipo	Descrição
enabled_lifetime_type	integer	Cenário para desabilitar recursos de LLD perdidos. Valores possíveis: 0 - Desabilitar após atingir o limite de tempo de vida; 1 - Não desabilitar; 2 - *(padrão)* Desabilitar imediatamente.
master_itemid	ID	ID do item mestre. A regra de descoberta não pode ser um item mestre de outra regra de descoberta. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Dependent item” - *somente leitura* para objetos herdados
output_format	integer	Define se a resposta deve ser convertida em JSON. Valores possíveis: 0 - *(padrão)* Armazenar como bruta; 1 - Converter para JSON. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
params	string	Parâmetros adicionais dependendo do tipo do protótipo de regra de LLD: - script executado para regras de LLD SSH e Telnet; - consulta SQL para regras de LLD de monitoramento de banco de dados; - fórmula para regras de LLD calculadas; - script para regras de LLD de script e navegador. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` estiver definido como ”Script” ou ”Browser”)
parameters	object/array	Parâmetros adicionais se `type` estiver definido como ”Script” ou ”Browser”. Array de objetos com as propriedades `name` e `value`, em que `name` deve ser exclusivo. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”JMX agent” e `username` estiver definido - *com suporte* se `type` estiver definido como ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)

Propriedade	Tipo	Descrição
post_type	integer	Tipo do corpo dos dados de postagem armazenado na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos; 2 - Dados JSON; 3 - Dados XML. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
posts	string	Dados do corpo da solicitação HTTP(S). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”JSON data” ou ”XML data” - *com suporte* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”Raw data” - *somente leitura* para objetos herdados
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key”
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key”
query_fields	array	Array de campos de consulta que serão enviados ao executar uma solicitação HTTP. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo do método de solicitação. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
retrieve_mode	integer	Parte da resposta que deve ser armazenada. Valores possíveis se `request_method` estiver definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - O corpo e os cabeçalhos serão armazenados. Valores possíveis se `request_method` estiver definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo da chave SSL pública. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_file	string	Caminho do arquivo da chave SSL privada. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_password	string	Senha do arquivo de chave SSL. Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
state	integer	Estado do protótipo de regra de LLD. Valores possíveis: 0 - *(padrão)* normal; 1 - sem suporte. Comportamento da propriedade: - *somente leitura*
status	integer	Status do protótipo de regra de LLD. Valores possíveis: 0 - *(padrão)* protótipo de regra de LLD habilitado; 1 - protótipo de regra de LLD desabilitado.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Também aceita macros de usuário como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *com suporte* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
templateid	ID	ID do protótipo de regra de LLD pai do template. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
timeout	string	Tempo limite da solicitação de coleta de dados do item. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-600s. Padrão: "" - usar configurações do proxy/globais. Comportamento da propriedade: - *com suporte* se `type` estiver 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: - *com suporte* se `type` estiver definido como "Zabbix trapper" ou se `type` estiver definido como "HTTP agent" e `allow_traps` estiver definido como "Allow incoming data". Se a propriedade for omitida, ela será definida automaticamente como a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}` (valor padrão). Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Se a propriedade for definida como uma string vazia, as conexões recebidas serão negadas de todos os hosts.
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como "SSH agent", "TELNET agent" ou se `type` estiver definido como "JMX agent" e `password` estiver definido - *com suporte* se `type` estiver definido como "Simple check", "Database monitor" ou "HTTP agent" - *somente leitura* para objetos herdados (se `type` estiver definido como "HTTP agent")
uuid	string	Identificador universal exclusivo, usado para vincular protótipos de regra de LLD importados aos já existentes. Gerado automaticamente se não for fornecido. Comportamento da propriedade: - *com suporte* se o protótipo de regra de LLD pertencer a um template
verify_host	integer	Define se o nome do host da conexão deve ser validado para corresponder ao nome no certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver definido como "HTTP agent" - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
verify_peer	integer	Define se o certificado do host deve ser validado quanto à autenticidade. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *com suporte* se `type` estiver 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. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Filtro de 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 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 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*
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 protótipo de regra LLD estiver definido como ”expressão personalizada”

Propriedade	Tipo	Descrição
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, consulte 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 do protótipo de regra LLD

O objeto de pré-processamento do protótipo de regra LLD tem as seguintes propriedades.

Property	Type	Description
type	integer	O tipo da opção de pré-processamento. Valores possíveis: 5 - Expressão regular; 11 - XPath XML; 12 - JSONPath; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar se há erro no JSON; 17 - Verificar se há erro no XML; 20 - Descartar valor inalterado com heartbeat; 21 - JavaScript; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 27 - XML para JSON; 28 - Valor de SNMP walk; 29 - SNMP walk para JSON; 30 - Obter valor via SNMP. Comportamento da propriedade: - *required*

Property	Type	Description
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Vários parâmetros são separados pelo caractere de nova linha (\n). Comportamento da propriedade: - *required* se `type` estiver definido como ”Expressão regular” (5), ”XPath XML” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar se há erro no JSON” (16), ”Verificar se há erro no XML” (17), ”Descartar valor inalterado com heartbeat” (20), ”JavaScript” (21), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Obter valor via SNMP” (30)
error_handler	integer	Tipo de ação usado em caso de falha da etapa de pré-processamento. Valores possíveis: 0 - A mensagem de erro é definida pelo Zabbix server; 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Comportamento da propriedade: - *required* se `type` estiver definido como ”Expressão regular” (5), ”XPath XML” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar se há erro no JSON” (16), ”Verificar se há erro no XML” (17), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”XML para JSON” (27), ”Valor de SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Obter valor via SNMP” (30)
error_handler_params	string	Parâmetros do manipulador de erros. Comportamento da propriedade: - *required* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os parâmetros e manipuladores de erros a seguir são compatíveis com cada tipo de pré-processamento.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
5	Expressão regular	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Corresponde à expressão regular	pattern[1]			0, 1, 2, 3
15	Não corresponde à expressão regular	pattern[1]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
16	Verificar se há erro no JSON	path[3]			0, 1, 2, 3
17	Verificar se há erro no XML	path[3]			0, 1, 2, 3
20	Descartar valor inalterado com heartbeat	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus para JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV para JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Substituir	search string[2]	replacement[2]		
27	XML para JSON				0, 1, 2, 3
28	Valor de 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[7]	Field name[2]	OID prefix[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS	0, 1, 2, 3
30	Obter valor via SNMP	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 XPath XML
 [4] inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [5] macro de usuário
 [6] padrão do Prometheus seguindo a sintaxe: `<metric name>{<label name>="<label value>", ...} == <value>`. Cada componente do padrão do Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser uma macro de usuário.
 [7] Compatível com vários registros "Field name,OID prefix,Format" delimitados por um caractere de nova linha.

Substituições de protótipo de regra de LLD

O objeto de substituições de protótipo de regra de LLD define um conjunto de regras (filtros, condições e operações) que são usadas para substituir propriedades de diferentes objetos de protótipo. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome único da substituição. Comportamento da propriedade: - *obrigatório*
step	integer	Número de ordem único da substituição. Comportamento da propriedade: - *obrigatório*
stop	integer	Parar o processamento das próximas substituições se corresponder. Valores possíveis: 0 - *(padrão)* não para o processamento das substituições; 1 - para o processamento das substituições se o filtro corresponder.
filter	object	Filtro de substituição.
operations	object/array	Operações de substituição.

Filtro de substituição 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 será aplicada. Ele 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 de 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 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*

Propriedade	Tipo	Descrição
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 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 substituição de protótipo de regra LLD

O objeto de condição de filtro de substituição 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 exclusivo arbitrário usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto de filtro de substituição de protótipo de 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.

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 - Protótipo de item; 1 - Protótipo de trigger; 2 - Protótipo de gráfico; 3 - Protótipo de host. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição de 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 de operação de sobrescrição para objetos de protótipo de item, trigger e host.
opdiscover	object	Objeto de status de descoberta de operação de sobrescrição (todos os tipos de objeto).
opperiod	object	Objeto de período de operação de sobrescrição (intervalo de atualização) para objeto de protótipo de item.
ophistory	object	Objeto de histórico de operação de sobrescrição para objeto de protótipo de item.
optrends	object	Objeto de tendências de operação de sobrescrição para objeto de protótipo de item.
opseverity	object	Objeto de severidade de operação de sobrescrição para objeto de protótipo de trigger.
optag	object/array	Objeto de tag de operação de sobrescrição para objetos de protótipo de trigger e host.
optemplate	object/array	Objeto de template de operação de sobrescrição para objeto de protótipo de host.
opinventory	object	Objeto de inventário de operação de sobrescrição para objeto de 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 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 de LLD

O período de operação de substituição do protótipo de regra de LLD é um valor de intervalo de atualização definido para o item descoberto. Ele tem as seguintes propriedades:

Property	Type	Description
delay	string	Substitui o intervalo de atualização do protótipo de item. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário ou macros de LLD. Quando usado, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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	Substitui o histórico do protótipo de item, que é uma unidade de tempo de quanto tempo os dados do histórico devem ser armazenados. Também aceita macro de usuário e macro LLD. Comportamento da propriedade: - *obrigatório*

Tendências de operação de substituição de 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 - médio; 4 - alto; 5 - desastre. Comportamento da propriedade: - *obrigatório*

Tag de operação de substituição de protótipo de regra de LLD

O objeto de tag de operação de substituição de protótipo de regra de LLD contém o nome e o valor da tag que são definidos para o objeto descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
tag	string	Novo nome da tag. Comportamento da propriedade: - *obrigatório*
value	string	Novo valor da tag.

Template de operação de 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	Substitui o template dos templates vinculados ao protótipo de host. Comportamento da propriedade: - *obrigatório*

Inventário de operação de substituição de protótipo de regra LLD

Valor do modo de inventário de operação de substituição de protótipo de 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 novos protótipos de regra de LLD.

Note:
Este método está disponível apenas para 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)` Protótipos de regras LLD a serem criados.

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 a regra LLD.
preprocessing	object/array	Opções de pré-processamento do protótipo de regra LLD.
lld_macro_paths	object/array	Opções de lld_macro_path do protótipo de regra LLD.
overrides	object/array	Opções de substituições do protótipo de regra 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 regra 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. Consulte Funções de usuário para obter 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 de LLD de acordo com os parâmetros informados.

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 obter 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`.
selectFilter	query	Retorna uma propriedade `filter` com os 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`.

Parâmetro	Tipo	Descrição
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 substituição, 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 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	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 de 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 de 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. Consulte Funções de usuário para obter 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
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 tem as seguintes propriedades.

Property	Type	Description
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 proxies. 0 - o proxy não está atribuído a nenhum grupo de proxies.
local_address	string	Endereço para agents ativos. Endereço IP ou nome DNS ao qual se conectar. Comportamento da propriedade: - *obrigatório* se `proxy_groupid` não for 0
local_port	string	Número da porta local do proxy à qual se conectar. Macros de usuário são compatíveis. Padrão: 10051. Comportamento da propriedade: - *compatível* 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	Momento em que o proxy se conectou pela última vez ao server. Comportamento da propriedade: - *somente leitura*
address	string	Endereço IP ou nome DNS ao qual se conectar. Macros de usuário são compatíveis. Comportamento da propriedade: - *obrigatório* se o modo de operação do proxy Zabbix for passivo
port	string	Número da porta à qual se conectar. Macros de usuário são compatíveis. Padrão: 10051. Comportamento da propriedade: - *compatível* se o modo de operação do proxy Zabbix for passivo
allowed_addresses	string	Endereços IP ou nomes DNS do proxy Zabbix ativo, delimitados por vírgulas.
tls_connect	integer	Conexões com o proxy. Valores possíveis: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado.

Property	Type	Description
tls_accept	integer	Conexões do proxy. Valores possíveis da máscara de bits: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado. Este é um campo de máscara de bits; qualquer soma dos valores possíveis da máscara de bits é aceita (por exemplo, 6 para PSK e certificado).
tls_issuer	string	Emissor do certificado.
tls_subject	string	Assunto do certificado.
tls_psk_identity	string	Identidade da PSK; deve ser associada a apenas uma PSK (entre autoregistration, hosts e proxies). Não inclua informações confidenciais na identidade da PSK, pois ela é enviada sem criptografia pela rede para informar ao receptor qual PSK deve ser usada. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `tls_connect` estiver definido como ”PSK” ou se `tls_accept` contiver o bit ”PSK”
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `tls_connect` estiver definido como ”PSK” ou se `tls_accept` contiver o bit ”PSK”
custom_timeouts	integer	Define se os tempos limite globais dos itens devem ser substituídos no nível do proxy. Valores possíveis: 0 - *(padrão)* usar configurações globais; 1 - substituir tempos limite.
timeout_zabbix_agent	string	Não gastar mais do que `timeout_*` segundos no processamento. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`). Também aceita macros de usuário. Usado somente se `custom_timeouts` estiver definido como ”substituir tempos limite”; caso contrário, o valor será ignorado. O intervalo de valores possíveis é: 1-600s. Comportamento da propriedade: - *obrigatório* se `custom_timeouts` estiver definido como ”substituir tempos limite”
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		

Property	Type	Description
version	integer	Versão do proxy. Número de versão do Zabbix com três partes, em que 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 desconhecida do proxy. Comportamento da propriedade: - *somente leitura*
compatibility	integer	Versão do proxy em comparação com a versão do server Zabbix. Valores possíveis: 0 - Indefinida; 1 - Versão atual (o proxy e o server têm a mesma versão principal); 2 - Versão desatualizada (a versão do proxy é anterior à versão do server, mas tem suporte parcial); 3 - Versão sem suporte (a versão do proxy é anterior à versão da versão LTS anterior do server ou a versão principal do server é anterior à 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*

Monitoramento de desempenho de aplicações (APM)

O objeto APM tem as seguintes propriedades.

Property	Type	Description
data_collection_status	integer	Indica se a coleta de dados de telemetria está habilitada no proxy. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada.
max_messages_per_second	integer	Número máximo de mensagens por segundo que o proxy aceita. O intervalo de valores possíveis é: 0 (ilimitado) – 2147483647. Padrão: 0. Comportamento da propriedade: - *supported* se `data_collection_status` estiver definido como ”Enabled”
additional_resource_attributes	array	Matriz de atributos de recurso a serem adicionados aos dados de telemetria recebidos. Comportamento da propriedade: - *supported* se `data_collection_status` estiver definido como ”Enabled”

Atributos do recurso

O objeto de atributos do recurso tem as seguintes propriedades.

Property	Type	Description
key	string	Nome do atributo do recurso. Deve ser exclusivo para o `signal_type` especificado. Comportamento da propriedade: - *obrigatório*
value	string	Valor do atributo do recurso. Pode estar vazio.
signal_type	integer	Tipo de dados de telemetria ao qual o atributo se aplica. Valores possíveis: 0 - Traces; 1 - Métricas; 2 - Logs. Comportamento da propriedade: - *obrigatório*

proxy.create

Descrição

`object proxy.create(object/array proxies)`

Este método permite que você crie 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. Consulte Funções de usuário para obter 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 outro proxy, ele será reatribuído ao proxy atual. Os hosts devem ter apenas a propriedade `hostid` definida.
apm	object	Configurações de monitoramento de desempenho de aplicações (APM) do proxy.

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
}
```

Criando um proxy com criptografia baseada em certificado

Crie um proxy ativo com criptografia baseada em certificado. O proxy está configurado para monitorar o host "10785".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

Criando um proxy com APM habilitado

Crie um proxy ativo "Active proxy" com o APM habilitado. O proxy aceita até 1000 mensagens por segundo, adiciona o atributo de recurso `deployment.environment` aos traces e adiciona o atributo de recurso `cloud.region` às métricas.

Solicitação:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "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 de usuário. Consulte Funções de usuário para obter 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 da função de usuário. Consulte Funções de usuário para obter mais informações.
 Somente os proxies permitidos pela *Lista de acesso do proxy* dos grupos de usuários do usuário são retornados.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método é compatível com os seguintes parâmetros.

Parâmetro	Tipo	Descrição
proxyids	ID/array	Retorna somente os proxies com os IDs especificados.
proxy_groupids	ID/array	Retorna somente os proxies que pertencem aos grupos de proxies especificados.
selectApm	query	Retorna uma propriedade `apm` com as configurações de monitoramento de desempenho de aplicações (APM) do proxy.
selectAssignedHosts	query	Retorna uma propriedade `assignedHosts` com os hosts atribuídos ao proxy. É compatível com `count`.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts monitorados pelo proxy. É compatível com `count`.
selectProxyGroup	query	Retorna uma propriedade `proxyGroup` com o objeto do grupo de proxies.

Parâmetro	Tipo	Descrição
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `proxyid`, `name`, `operating_mode`.
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 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 User roles para obter 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.

Parameter	Type	Description
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.
apm	object	Configurações de monitoramento de desempenho de aplicações (APM) do proxy.

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 de LLD.

Referências de objetos:

- Regra de descoberta

Métodos disponíveis:

- drule.create - criar novas regras de descoberta de rede
- drule.delete - excluir regras de descoberta de rede
- drule.get - recuperar regras de descoberta de rede
- drule.update - atualizar regras de descoberta de rede

Objeto de regra de descoberta

Os seguintes objetos estão diretamente relacionados à API `drule`.

Regra de descoberta

O objeto de regra de descoberta define uma regra de descoberta de rede. Ele tem as seguintes propriedades.

Property	Type	Description
druleid	ID	ID da regra de descoberta. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
iprange	string	Um ou vários intervalos de IP a serem verificados, separados por vírgulas. Consulte a seção configuração da descoberta de rede para obter mais informações sobre os formatos de intervalos de IP compatíveis. Comportamento da propriedade: - *obrigatória* para operações de criação
name	string	Nome da regra de descoberta. Comportamento da propriedade: - *obrigatória* para operações de criação
delay	string	Intervalo de execução da regra de descoberta. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou uma macro de usuário. Padrão: 1h.
proxyid	ID	ID do proxy usado para a descoberta.
status	integer	Indica se a regra de descoberta está habilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
concurrency_max	integer	Número máximo de verificações simultâneas por regra de descoberta. Valores possíveis: 0 - *(padrão)* número ilimitado de verificações; 1 - uma verificação; 2-999 - número personalizado de verificações.
error	string	Texto do erro caso tenha ocorrido algum problema durante a execução da 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. Consulte Funções de usuário para obter 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

- Discovery check

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. Consulte Funções de usuário para obter 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 informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dhostids	ID/array	Retorna apenas regras de descoberta que criaram os hosts descobertos fornecidos.
druleids	ID/array	Retorna apenas regras de descoberta com os IDs fornecidos.
dserviceids	ID/array	Retorna apenas regras de descoberta que criaram os serviços descobertos fornecidos.
selectDChecks	query	Retorna uma propriedade `dchecks` com as verificações de descoberta usadas pela regra de descoberta. Suporta `count`.
selectDHosts	query	Retorna uma propriedade `dhosts` com os hosts descobertos criados pela regra de descoberta. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectDChecks` - os resultados serão ordenados por `dcheckid`; `selectDHosts` - os resultados serão ordenados por `dhostsid`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `druleid`, `name`.
countOutput	boolean	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 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 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

- Discovery check

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 de macro de LLD
- Pré-processamento de regra de LLD
- Substituições de regra de LLD
  - Filtro de substituição de regra de LLD
    - Condição de filtro de substituição de regra de LLD
  - Operação de substituição de regra de LLD
    - Status da operação de substituição de regra de LLD
    - Descoberta de operação de substituição de regra de LLD
    - Período de operação de substituição de regra de LLD
    - Histórico de operação de substituição de regra de LLD
    - Tendências de operação de substituição de regra de LLD
    - Gravidade da operação de substituição de regra de LLD
    - Tag de operação de substituição de regra de LLD
    - Template de operação de substituição de regra de LLD
    - Inventário de operação de substituição de regra de LLD

Métodos disponíveis:

- discoveryrule.create - criar novas regras de LLD
- discoveryrule.delete - excluir regras de LLD
- discoveryrule.get - recuperar regras de LLD
- discoveryrule.update - atualizar regras de LLD

Objeto de regra de LLD

Os seguintes objetos estão diretamente relacionados à API `discoveryrule`.

Regra de LLD

O objeto de regra de descoberta de baixo nível tem as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID da regra de LLD. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
delay	string	Intervalo de atualização da regra de LLD. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário. Se usada, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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: - *obrigatória* se `type` estiver 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` estiver definido como ”Zabbix agent (active)” (7) e `key_` não contiver ”mqtt.get”
hostid	ID	ID do host ao qual a regra de LLD pertence. Comportamento da propriedade: - *constante* - *obrigatória* 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 um protótipo. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
interfaceid	ID	ID da interface do host da regra de LLD. Comportamento da propriedade: - *obrigatória* se a regra de LLD pertencer a um host e `type` estiver definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent” ou ”SNMP agent” - *compatível* se a regra de LLD pertencer a um host e `type` estiver definido como ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent”
key_	string	Chave da regra de LLD. Comportamento da propriedade: - *obrigatória* para operações de criação - *somente leitura* para objetos herdados
name	string	Nome da regra de LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
type	integer	Tipo da regra de LLD. Valores possíveis: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Esse tipo é permitido somente se o pai da regra for um template (que se presume estar vinculado a um protótipo de host na descoberta) ou um host descoberto (que tenha uma regra de 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: - *obrigatória* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
allow_traps	integer	Permite que o valor seja preenchido de forma semelhante ao item trapper. Valores possíveis: 0 - *(padrão)* Não aceitar dados recebidos; 1 - Aceitar dados recebidos. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent”

Propriedade	Tipo	Descrição
authtype	integer	Método de autenticação. Valores possíveis se `type` estiver definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` estiver definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)
description	string	Descrição da regra de LLD.
error	string	Texto do erro caso haja problemas ao atualizar o valor da regra de LLD. Comportamento da propriedade: - *somente leitura*
follow_redirects	integer	Seguir redirecionamentos da resposta durante a coleta de dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *compatível* se `type` estiver 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: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
http_proxy	string	String de conexão do proxy HTTP(S). Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”IPMI agent” e `key_` não estiver definido como ”ipmi.get” - *compatível* se `type` estiver definido como ”IPMI agent” - *somente leitura* para objetos herdados
jmx_endpoint	string	String de conexão personalizada do JMX agent. Padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *compatível* se `type` estiver definido como ”JMX agent”

Propriedade	Tipo	Descrição
lifetime	string	Período após o qual os itens que não forem mais descobertos serão excluídos. Aceita um número de segundos (por exemplo, 3600), um valor com um sufixo de tempo (por exemplo, 3600s, 180m, 24h, 1d) ou uma macro de usuário. Padrão: 7d.
lifetime_type	integer	Cenário para excluir recursos de LLD perdidos. Valores possíveis: 0 - *(padrão)* Excluir após atingir o limite de tempo de vida; 1 - Não excluir; 2 - Excluir imediatamente.
enabled_lifetime	string	Período após o qual os itens que não forem mais descobertos serão desabilitados. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou uma macro de usuário. Padrão: 0.
enabled_lifetime_type	integer	Cenário para desabilitar recursos de LLD perdidos. Valores possíveis: 0 - Desabilitar após atingir o limite de tempo de vida; 1 - Não desabilitar; 2 - *(padrão)* Desabilitar imediatamente.
master_itemid	ID	ID do item mestre. A regra de descoberta não pode ser item mestre de outra regra de descoberta. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Dependent item” - *somente leitura* para objetos herdados
output_format	integer	Define se a resposta deve ser convertida para JSON. Valores possíveis: 0 - *(padrão)* Armazenar como bruta; 1 - Converter para JSON. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
params	string	Parâmetros adicionais dependendo do tipo da regra de LLD: - script executado para regras de LLD SSH e Telnet; - consulta SQL para regras de LLD de monitoramento de banco de dados; - fórmula para regras de LLD calculadas; - script para regras de LLD de script e navegador. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` estiver definido como ”Script” ou ”Browser”)
parameters	object/array	Parâmetros adicionais se `type` estiver definido como ”Script” ou ”Browser”. Array de objetos com as propriedades `name` e `value`, em que `name` deve ser exclusivo. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”JMX agent” e `username` estiver definido - *compatível* se `type` estiver definido como ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` estiver definido como ”HTTP agent”)
post_type	integer	Tipo do corpo dos dados de postagem armazenado na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos; 2 - Dados JSON; 3 - Dados XML. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
posts	string	Dados do corpo da solicitação HTTP(S). Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”JSON data” ou ”XML data” - *compatível* se `type` estiver definido como ”HTTP agent” e `post_type` estiver definido como ”Raw data” - *somente leitura* para objetos herdados
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key”
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH agent” e `authtype` estiver definido como ”public key”
query_fields	array	Array de campos de consulta que serão enviados ao executar uma solicitação HTTP. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo do método de solicitação. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
retrieve_mode	integer	Parte da resposta que deve ser armazenada. Valores possíveis se `request_method` estiver definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - O corpo e os cabeçalhos serão armazenados. Valores possíveis se `request_method` estiver definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
snmp_oid	string	OID SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo de chave SSL pública. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_file	string	Caminho do arquivo de chave SSL privada. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_password	string	Senha do arquivo de chave SSL. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
state	integer	Estado da regra de LLD. Valores possíveis: 0 - *(padrão)* normal; 1 - não compatível. Comportamento da propriedade: - *somente leitura*
status	integer	Status da regra de LLD. Valores possíveis: 0 - *(padrão)* regra de LLD habilitada; 1 - regra de LLD desabilitada.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Também aceita macros de usuário como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados
templateid	ID	ID da regra de LLD do template pai. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
timeout	string	Tempo limite da solicitação de coleta de dados do item. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m). Também aceita macros de usuário. O intervalo de valores possíveis é: 1-600s. Padrão: "" - usar as configurações do proxy/globais. Comportamento da propriedade: - *compatível* se `type` estiver 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, - se o valor padrão for aplicado e o item pertencer a um host monitorado por um proxy ou grupo de proxies inacessível, o tempo limite global será usado.
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *compatível* se `type` estiver definido como "Zabbix trapper" ou se `type` estiver definido como "HTTP agent" e `allow_traps` estiver definido como "Allow incoming data". Se a propriedade for omitida, ela será definida automaticamente como a macro de usuário `{$TRAPPER.ALLOWED_HOSTS}` (valor padrão). Por padrão, `{$TRAPPER.ALLOWED_HOSTS}` é `127.0.0.1,::1`. Se a propriedade for definida como uma string vazia, as conexões recebidas serão negadas de todos os hosts.
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como "SSH agent", "TELNET agent" ou se `type` estiver definido como "JMX agent" e `password` estiver definido - *compatível* se `type` estiver definido como "Simple check", "Database monitor" ou "HTTP agent" - *somente leitura* para objetos herdados (se `type` estiver definido como "HTTP agent")
uuid	string	Identificador universal exclusivo, usado para vincular regras de LLD importadas às regras já existentes. Gerado automaticamente se não for fornecido. Comportamento da propriedade: - *compatível* se a regra de LLD pertencer a um template
verify_host	integer	Define se deve ser validado que o nome do host da conexão corresponde ao nome no certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *compatível* se `type` estiver definido como "HTTP agent" - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
verify_peer	integer	Define se deve ser validado que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”HTTP agent” - *somente leitura* para objetos herdados

Cabeçalho HTTP

O objeto header possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho HTTP. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho. Comportamento da propriedade: - *obrigatório*

Campo de consulta HTTP

O objeto de campo de consulta define um nome e valor que é usado para especificar um parâmetro de URL. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Filtro de regra LLD

O objeto de filtro de regra LLD define um conjunto de condições que podem ser usadas para filtrar objetos descobertos. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
conditions	object/array	Conjunto de condições de filtro a serem usadas para filtrar resultados. As condições serão ordenadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação da condição de filtro. Valores possíveis: 0 - E/Ou; 1 - E; 2 - Ou; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que referenciam condições de filtro específicas pelo seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições de filtros com uma expressão personalizada. A expressão deve conter IDs que referenciem condições de filtro específicas pelo seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição de filtro da regra LLD

O objeto de condição de filtro da regra LLD define uma verificação separada a ser realizada no valor de uma macro LLD. 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 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, veja 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 da regra LLD

O objeto de pré-processamento da regra LLD tem as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo da opção de pré-processamento. Valores possíveis: 5 - Expressão regular; 11 - XPath XML; 12 - JSONPath; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar se há erro no JSON; 17 - Verificar se há erro no XML; 20 - Descartar se não houver alteração com heartbeat; 21 - JavaScript; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 27 - XML para JSON; 28 - Valor da varredura SNMP; 29 - Varredura SNMP para JSON; 30 - Obter valor via SNMP. 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` estiver definido como ”Expressão regular” (5), ”XPath XML” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar se há erro no JSON” (16), ”Verificar se há erro no XML” (17), ”Descartar se não houver alteração com heartbeat” (20), ”JavaScript” (21), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Valor da varredura SNMP” (28), ”Varredura SNMP para JSON” (29) ou ”Obter valor via SNMP” (30)

Propriedade	Tipo	Descrição
error_handler	integer	Tipo de ação usado em caso de falha da etapa de pré-processamento. Valores possíveis: 0 - A mensagem de erro é definida pelo Zabbix server; 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Expressão regular” (5), ”XPath XML” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar se há erro no JSON” (16), ”Verificar se há erro no XML” (17), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”XML para JSON” (27), ”Valor da varredura SNMP” (28), ”Varredura SNMP para JSON” (29) ou ”Obter valor via SNMP” (30)
error_handler_params	string	Parâmetros do manipulador de erros. Comportamento da propriedade: - *obrigatório* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os parâmetros e manipuladores de erros a seguir são compatíveis com cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
5	Expressão regular	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Corresponde à expressão regular	pattern[1]			0, 1, 2, 3
15	Não corresponde à expressão regular	pattern[1]			0, 1, 2, 3
16	Verificar se há erro no JSON	path[3]			0, 1, 2, 3
17	Verificar se há erro no XML	path[3]			0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erros compatíveis
20	Descartar se não houver alteração com heartbeat	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus para JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV para JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Substituir	search string[2]	replacement[2]		
27	XML para JSON				0, 1, 2, 3
28	Valor da varredura SNMP	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	Varredura SNMP para JSON[7]	Field name[2]	OID prefix[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS	0, 1, 2, 3
30	Obter valor via SNMP	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 XPath XML
 [4] inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [5] macro de usuário
 [6] padrão do Prometheus seguindo a sintaxe: `<metric name>{<label name>="<label value>", ...} == <value>`. Cada componente do padrão do Prometheus (métrica, nome do label, valor do label e valor da métrica) pode ser uma macro de usuário.
 [7] Compatível com vários registros "Nome do campo,Prefixo do OID,Registros de formato", delimitados por um caractere de nova linha.

Substituições de regras de LLD

O objeto de substituições de regras de LLD define um conjunto de regras (filtros, condições e operações) que são usadas para substituir propriedades de diferentes objetos de protótipo. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome único da substituição. Comportamento da propriedade: - *obrigatório*
step	integer	Número de ordem único da substituição. Comportamento da propriedade: - *obrigatório*
stop	integer	Parar o processamento das próximas substituições se corresponder. Valores possíveis: 0 - *(padrão)* não parar o processamento das substituições; 1 - parar 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 substituição de regra LLD

O objeto de condição de filtro de substituição de regra LLD define uma verificação separada a ser realizada no valor de uma macro LLD. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
macro	string	Macro LLD para realizar a verificação. Comportamento da propriedade: - *obrigatório*
value	string	Valor para comparar. Comportamento da propriedade: - *obrigatório* se `operator` estiver definido como ”corresponde à expressão regular” ou ”não corresponde à expressão regular”
formulaid	string	ID exclusivo arbitrário usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto de filtro de substituição de regra LLD estiver definido como ”expressão personalizada”
operator	integer	Operador de condição. Valores possíveis: 8 - *(padrão)* corresponde à expressão regular; 9 - não corresponde à expressão regular; 12 - existe; 13 - não existe.

Operação de substituição de regra LLD

A operação de substituição de regra LLD é uma combinação de condições e ações a serem realizadas no objeto protótipo. Ela 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 substituição. Valores possíveis: 0 - *(padrão)* igual; 1 - diferente; 2 - contém; 3 - não contém; 8 - corresponde; 9 - não corresponde.
value	string	Padrão para corresponder ao nome do protótipo de item, trigger, gráfico ou host, dependendo do objeto selecionado.
opstatus	object	Objeto de status da operação de substituição para objetos protótipo de item, trigger e host.
opdiscover	object	Objeto de status de descoberta da operação de substituição (todos os tipos de objeto).

Propriedade	Tipo	Descrição
opperiod	object	Objeto de período (intervalo de atualização) da operação de substituição para objeto protótipo de item.
ophistory	object	Objeto de histórico da operação de substituição para objeto protótipo de item.
optrends	object	Objeto de tendências da operação de substituição para objeto protótipo de item.
opseverity	object	Objeto de severidade da operação de substituição para objeto protótipo de trigger.
optag	object/array	Objeto de tag da operação de substituição para objetos protótipo de trigger e host.
optemplate	object/array	Objeto de template da operação de substituição para objeto protótipo de host.
opinventory	object	Objeto de inventário da operação de substituição para objeto protótipo de host.

Status da operação de substituição da regra LLD

Status da operação de substituição da regra LLD que é definida para o objeto descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
status	integer	Substitui o status para o objeto selecionado. Valores possíveis: 0 - Criar habilitado; 1 - Criar desabilitado. Comportamento da propriedade: - *obrigatório*

Operação de descoberta de substituição de regra LLD

Status de descoberta da operação de substituição de regra LLD que é definido para o objeto descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
discover	integer	Substitui o status de descoberta para o objeto selecionado. Valores possíveis: 0 - Sim, continue descobrindo os objetos; 1 - Não, novos objetos não serão descobertos e os existentes serão marcados como perdidos. Comportamento da propriedade: - *obrigatório*

Período da operação de substituição da regra LLD

O período da operação de substituição da regra LLD é um valor de intervalo de atualização definido para o item descoberto. Ele tem as seguintes propriedades:

Property	Type	Description
delay	string	Substitui o intervalo de atualização do protótipo de item. Aceita um número de segundos (por exemplo, 1, 60, 300), um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h, 1d) ou, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma combinação de intervalos flexíveis e de agendamento. Aceita macros de usuário ou macros LLD. Quando usado, o valor deve ser uma única macro. Não há suporte para várias macros ou macros combinadas com texto. Os 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*

Severidade da operação de substituição da regra LLD

Valor de severidade 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 severidade do protótipo de trigger. Valores possíveis: 0 - *(padrão)* não classificado; 1 - informação; 2 - aviso; 3 - médio; 4 - alto; 5 - desastre. Comportamento da propriedade: - *obrigatório*

Tag da operação de substituição da regra LLD

O objeto tag da operação de substituição da regra LLD contém o nome e o valor da tag que são definidos para o objeto descoberto. Ele possui as seguintes propriedades:

Property	Type	Description
tag	string	Novo nome da tag. Também aceita macro de usuário e macro LLD. Property behavior: - *required*
value	string	Novo valor da tag. Também aceita macro de usuário e macro LLD.

Modelo de operação de substituição de regra LLD

Objeto de modelo de operação de substituição 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 sobrescrição de regra de LLD

Valor do modo de inventário da operação de sobrescrição de regra de LLD que é definido para o host descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
inventory_mode	integer	Sobrescreve 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 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)` 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 de LLD
- Caminhos de macro de LLD
- Pré-processamento de regra de 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 de LLD.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações da função de usuário. Consulte Funções de usuário para obter mais informações.
 O usuário não poderá excluir uma regra de LLD que descubra entidades usando um proxy inacessível. Consulte configuração do grupo de usuários (*Lista de acesso a proxy*) para obter mais informações.

Parâmetros

`(array)` IDs das regras LLD a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das regras LLD excluídas na propriedade `ruleids`.

Exemplos

Excluindo várias regras de LLD

Exclua duas regras de LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Fonte

CDiscoveryRule::delete() 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 informados.

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 obter 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 ela foi descoberta.

Parâmetro	Tipo	Descrição
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 os dados do filtro usado pela regra LLD.
selectGraphs	query	Retorna uma propriedade `graphs` com protótipos de gráficos que pertencem à regra LLD. Suporta `count`.
selectHostPrototypes	query	Retorna uma propriedade `hostPrototypes` com protótipos de host que pertencem à regra LLD. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts aos quais 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 substituição, condições e operações que são realizadas em objetos protótipos.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual a regra LLD pertence.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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 de LLD existentes.

> **Note:**
> Este método está disponível apenas para 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 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

Parâmetro	Tipo	Descrição
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*.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "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 de relatório tem as seguintes propriedades:

Property	Type	Description
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 exclusivo do relatório. Comportamento da propriedade: - *obrigatório* para operações de criação
dashboardid	ID	ID do dashboard no qual o relatório se baseia. 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.

Property	Type	Description
start_time	integer	Horário do dia, em segundos, em que o relatório será preparado para envio. Padrão: 0.
weekdays	integer	Dias da semana para o envio do relatório. Valores possíveis do 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 do bitmap é aceita (por exemplo, 21 para segunda, quarta e sexta-feira). Padrão: 0. Comportamento da propriedade: - *obrigatório* se `cycle` estiver definido como ”weekly”.
active_since	string	Data de início. Valores possíveis: string vazia - *(padrão)* não especificada (armazenada como 0); data específica no formato YYYY-MM-DD (armazenada como um timestamp do início do dia (00:00:00)).
active_till	string	Data de término. Valores possíveis: string vazia - *(padrão)* não especificada (armazenada como 0); data específica no formato YYYY-MM-DD (armazenada como um timestamp do final do dia (23:59:59)).
subject	string	Assunto da mensagem do relatório.
message	string	Texto da mensagem do relatório.
status	integer	Indica se o relatório está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
description	text	Descrição do relatório.
state	integer	Estado do relatório. Valores possíveis: 0 - *(padrão)* o relatório ainda não foi processado; 1 - o relatório foi gerado e enviado com sucesso a todos os destinatários; 2 - falha na geração do relatório; ”info” contém informações sobre o erro; 3 - o relatório foi gerado, mas houve falha no envio para alguns ou todos os destinatários; ”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*

Property	Type	Description
info	string	Descrição do erro ou informações adicionais. Comportamento da propriedade: - *somente leitura*

Usuários

O objeto users possui as seguintes propriedades:

Propriedade	Tipo	Descrição
userid	ID	ID do usuário para o qual o relatório será enviado. Comportamento da propriedade: - *obrigatório*
access_userid	ID	ID do usuário em nome do qual o relatório será gerado. 0 - *(padrão)* Gerar relatório pelo destinatário.
exclude	integer	Se deve excluir o usuário da lista de 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 de quem 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 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)` Relatórios agendados a serem criados.

Além das propriedades padrão de relatórios agendados, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
users	object/array	Usuários para enviar o relatório. Comportamento do parâmetro: - *obrigatório* se `user_groups` não estiver definido

Parâmetro	Tipo	Descrição
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 o tipo 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 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. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
reportids	ID/array	Retornar apenas relatórios agendados com os IDs de relatório fornecidos.
expired	boolean	Se definido como `true`, retorna apenas relatórios agendados expirados.
selectUsers	query	Retorna uma propriedade `users` com os usuários para os quais o relatório está configurado para ser enviado.
selectUserGroups	query	Retorna uma propriedade `user_groups` com os grupos de usuários para os quais o relatório está configurado para ser enviado.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `reportid`, `name`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperando dados do relatório

Solicitação:

```
{
    "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": "Relatório semanal",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Relatório semanal",
            "message": "Texto complementar do relatório",
            "status": "1",
            "description": "Descrição do relatório",
            "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": "Relatório mensal",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Relatório mensal",
            "message": "Texto complementar do relatório",
            "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 o tipo 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 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
- Role rules
  - UI element
  - Service
  - Service tag
  - Module
  - Action

Métodos disponíveis:

- role.create - criar novas funções de usuário
- role.delete - excluir funções de usuário
- role.get - recuperar funções de usuário
- role.update - atualizar funções de usuário

Objeto role

Os seguintes objetos estão diretamente relacionados à API `role`.

Role

O objeto role possui as seguintes propriedades:

Propriedade	Tipo	Descrição
roleid	ID	ID da role. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da role. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
type	integer	Tipo de usuário. Valores possíveis: 1 - *(padrão)* Usuário; 2 - Admin; 3 - Super admin. Comportamento da propriedade: - *obrigatório* para operações de criação
readonly	integer	Se a role é somente leitura. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *somente leitura*

Regras de função

O objeto de regras de função tem as seguintes propriedades:

Property	Type	Description
ui	array	Array de objetos de elemento da UI.
ui.default_access	integer	Indica se o acesso a novos elementos da UI está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
services.read.mode	integer	Acesso somente leitura aos serviços. Valores possíveis: 0 - Acesso somente leitura aos serviços especificados pelas propriedades `services.read.list` ou correspondentes à propriedade `services.read.tag`; 1 - *(padrão)* Acesso somente leitura a todos os serviços.
services.read.list	array	Array de objetos Service. Será concedido à função de usuário acesso somente leitura aos serviços especificados, incluindo os serviços filhos. O acesso somente leitura não substituirá o acesso de leitura e gravação aos serviços. Comportamento da propriedade: - *suportada* se `services.read.mode` estiver definido como "0"
services.read.tag	object	Array de objetos Service tag. Será concedido à função de usuário acesso somente leitura aos serviços correspondentes à tag, incluindo os serviços filhos. O acesso somente leitura não substituirá o acesso de leitura e gravação aos serviços. Comportamento da propriedade: - *suportada* se `services.read.mode` estiver definido como "0"
services.write.mode	integer	Acesso de leitura e gravação aos serviços. Valores possíveis: 0 - *(padrão)* Acesso de leitura e gravação aos serviços especificados pelas propriedades `services.write.list` ou correspondentes à propriedade `services.write.tag`; 1 - Acesso de leitura e gravação a todos os serviços.

Property	Type	Description
services.write.list	array	Array de objetos Service. Será concedido à função de usuário acesso de leitura e gravação aos serviços especificados, incluindo os serviços filhos. O acesso de leitura e gravação substituirá o acesso somente leitura aos serviços. Comportamento da propriedade: - *suportada* se `services.write.mode` estiver definido como ”0”
services.write.tag	object	Array de objetos Service tag. Será concedido à função de usuário acesso de leitura e gravação aos serviços correspondentes à tag, incluindo os serviços filhos. O acesso de leitura e gravação substituirá o acesso somente leitura aos serviços. Comportamento da propriedade: - *suportada* se `services.write.mode` estiver definido como ”0”
modules	array	Array de objetos module.
modules.default_access	integer	Indica se o acesso a novos módulos está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
api.access	integer	Indica se o acesso à API está habilitado. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado.
api.mode	integer	Modo de tratamento dos métodos da API listados na propriedade `api`. Valores possíveis: 0 - *(padrão)* Lista de negação; 1 - Lista de permissão.
api	array	Array de métodos da API.
actions	array	Array de objetos action.
actions.default_access	integer	Indica se o acesso a novas ações está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
devices.access	integer	Indica se o acesso aos próprios dispositivos móveis do usuário para uso com o aplicativo móvel do Zabbix está habilitado. Valores possíveis: 0 - (*padrão para User e Admin*) Desabilitado; 1 - (*padrão para Super admin*) Habilitado. Comportamento da propriedade: - *suportada* se `EnableMobileDevices` estiver habilitado
devices.actions	array	Array de objetos Device action. Comportamento da propriedade: - *suportada* se `devices.access` estiver definido como ”1”
devices.actions.default_access	integer	Indica se o acesso a novas ações de dispositivo está habilitado. Valores possíveis: 0 - (*padrão para User e Admin*) Desabilitado; 1 - (*padrão para Super admin*) Habilitado. Comportamento da propriedade: - *suportada* se `EnableMobileDevices` estiver habilitado

Elemento da interface do usuário

O objeto de elemento da interface do usuário tem as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do elemento da interface do usuário. Valores possíveis se `type` do objeto Role estiver definido como ”User”, ”Admin” ou ”Super admin”: `monitoring.dashboard` - *Dashboards*; `monitoring.problems` - *Monitoring > Problems*; `monitoring.hosts` - *Monitoring > Hosts*; `monitoring.latest_data` - *Monitoring > Latest data*; `monitoring.maps` - *Monitoring > Maps*; `services.services` - *Services > Services*; `services.sla_report` - *Services > SLA report*; `inventory.overview` - *Inventory > Overview*; `inventory.hosts` - *Inventory > Hosts*; `reports.availability_report` - *Reports > Availability report*; `reports.top_triggers` - *Reports > Triggers top 100*. Valores possíveis se `type` do objeto Role estiver definido como ”Admin” ou ”Super admin”: `monitoring.discovery` - *Monitoring > Discovery*; `services.sla` - *Services > SLA*; `reports.scheduled_reports` - *Reports > Scheduled reports*; `reports.notifications` - *Reports > Notifications*; `configuration.template_groups` - *Data collection > Template groups*; `configuration.host_groups` - *Data collection > Host groups*; `configuration.templates` - *Data collection > Templates*; `configuration.hosts` - *Data collection > Hosts*; `configuration.maintenance` - *Data collection > Maintenance*; `configuration.discovery` - *Data collection > Discovery*; `configuration.trigger_actions` - *Alerts > Actions > Trigger actions*; `configuration.service_actions` - *Alerts > Actions > Service actions*; `configuration.discovery_actions` - *Alerts > Actions > Discovery actions*; `configuration.autoregistration_actions` - *Alerts > Actions > Autoregistration actions*; `configuration.internal_actions` - *Alerts > Actions > Internal actions*. Valores possíveis se `type` do objeto Role estiver definido como ”Super admin”: `reports.system_info` - *Reports > System information*; `reports.audit` - *Reports > Audit log*; `reports.action_log` - *Reports > Action log*; `configuration.event_correlation` - *Data collection > Event correlation*; `administration.media_types` - *Alerts > Media types*; `administration.scripts` - *Alerts > Scripts*; `administration.user_groups` - *Users > User groups*; `administration.user_roles` - *Users > User roles*; `administration.users` - *Users > Users*; `administration.api_tokens` - *Users > API tokens*; `administration.authentication` - *Users > Authentication*; `administration.linked_devices` - *Users > Devices*; `administration.general` - *Administration > General*; `administration.audit_log` - *Administration > Audit log*; `administration.housekeeping` - *Administration > Housekeeping*; `administration.proxy_groups` - *Administration > Proxy groups*; `administration.proxies` - *Administration > Proxies*; `administration.macros` - *Administration > Macros*; `administration.queue` - *Administration > Queue*. Comportamento da propriedade: *- required*

Propriedade	Tipo	Descrição
status	integer	Indica se o acesso ao elemento da interface do usuário está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Serviço

Propriedade	Tipo	Descrição
serviceid	ID	ID do Serviço. Comportamento da propriedade: - *obrigatório*

Tag de serviço

Propriedade	Tipo	Descrição
tag	string	Nome da tag. Se uma string vazia for especificada, a tag de serviço não será usada para correspondência de serviço. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag. Se nenhum valor ou uma string vazia for especificada, apenas o nome da tag será usado para correspondência de serviço.

Módulo

O objeto módulo possui as seguintes propriedades:

Propriedade	Tipo	Descrição
moduleid	ID	ID do módulo. Comportamento da propriedade: - *obrigatório*
status	integer	Se o acesso ao módulo está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Ação

O objeto de ação tem as seguintes propriedades:

Property	Type	Description
name	string	Nome da ação. Valores possíveis se `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 aos problemas; `change_severity` - Alterar a severidade do problema; `acknowledge_problems` - Confirmar problemas; `suppress_problems` - Suprimir problemas; `close_problems` - Fechar problemas; `execute_scripts` - Executar scripts; `manage_api_tokens` - Gerenciar tokens de API; `change_problem_ranking` - Alterar a classificação do problema de causa para sintoma e vice-versa; `edit_own_media` - Permitir que os usuários criem e editem seus próprios meios de comunicação. Valores possíveis se `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, `select_server_for_monitoring` - Permitir que os usuários criem hosts monitorados pelo server e pelas regras de descoberta de rede executadas pelo server. Valores possíveis se `type` do objeto Role estiver definido como ”User” ou ”Admin”: `invoke_execute_now` - Permitir que os usuários com apenas permissões de leitura em um host executem verificações de item. Valores possíveis se `type` do objeto Role estiver definido como ”Super admin”: `edit_user_media` - Permitir que os usuários criem e editem meios de comunicação para outros usuários. Comportamento da propriedade: - *required*
status	integer	Indica se o acesso para executar a ação está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Ação de dispositivo

O objeto de ação de dispositivo tem as seguintes propriedades:

Property	Type	Description
name	string	Nome da ação de dispositivo. Valores possíveis se `type` do objeto Role estiver definido como ”User”, ”Admin” ou ”Super admin”: `manage_own` - Permitir que os usuários adicionem ou removam seus próprios dispositivos móveis. Valores possíveis se `type` do objeto Role estiver definido como ”Super admin”: `manage_user` - Permitir que os usuários adicionem ou removam dispositivos móveis de todos os usuários. Comportamento da propriedade: - *required*
status	integer	Indica 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 novos roles.

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 user role. Consulte User roles para obter 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 roles.

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 role de usuário. Consulte User roles para obter 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 roles de acordo com os parâmetros informados.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de role do usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
roleids	ID/array	Retorna apenas as roles com os IDs especificados.
current	boolean	Se definido como `true`, retorna apenas a role do usuário atual.
selectRules	query	Retorna uma propriedade `rules` com as regras da role.
selectUsers	query	Retorna uma propriedade `users` com os usuários aos quais a role está atribuída. Consulte `user.get` para ver as restrições com base no tipo de usuário.
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `roleid`, `name`.
countOutput	boolean	Esses parâmetros estão descritos na documentação 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": [
                {
                    "moduleid": 1,
                    "status": "1"
                },
                {
                    "moduleid": 2,
                    "status": "1"
                },
                {
                    "moduleid": 3,
                    "status": "1"
                },
                {
                    "moduleid": 4,
                    "status": "1"
                },
                {
                    "moduleid": 5,
                    "status": "1"
                },
                {
                    "moduleid": 6,
                    "status": "1"
                },
```

```
                {
                    "moduleid": 7,
                    "status": "1"
                },
                {
                    "moduleid": 8,
                    "status": "1"
                },
                {
                    "moduleid": 9,
                    "status": "1"
                },
                {
                    "moduleid": 10,
                    "status": "1"
                },
                {
                    "moduleid": 11,
                    "status": "1"
                },
                {
                    "moduleid": 12,
                    "status": "1"
                },
                {
                    "moduleid": 13,
                    "status": "1"
                },
                {
                    "moduleid": 14,
                    "status": "1"
                },
                {
                    "moduleid": 15,
                    "status": "1"
                },
                {
                    "moduleid": 16,
                    "status": "1"
                },
                {
                    "moduleid": 17,
                    "status": "1"
                },
                {
                    "moduleid": 18,
                    "status": "1"
                },
                {
                    "moduleid": 19,
                    "status": "1"
                },
                {
                    "moduleid": 20,
                    "status": "1"
                },
                {
                    "moduleid": 21,
                    "status": "1"
                },
                {
                    "moduleid": 22,
```

```
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
            {
                "name": "acknowledge_problems",
```

```
                        "status": "1"
                    },
                    {
                        "name": "suppress_problems",
                        "status": "1"
                    },
                    {
                        "name": "close_problems",
                        "status": "1"
                    },
                    {
                        "name": "change_severity",
                        "status": "1"
                    },
                    {
                        "name": "add_problem_comments",
                        "status": "1"
                    },
                    {
                        "name": "execute_scripts",
                        "status": "1"
                    },
                    {
                        "name": "manage_api_tokens",
                        "status": "1"
                    },
                    {
                        "name": "edit_maintenance",
                        "status": "1"
                    },
                    {
                        "name": "manage_scheduled_reports",
                        "status": "1"
                    },
                    {
                        "name": "manage_sla",
                        "status": "1"
                    },
                    {
                        "name": "invoke_execute_now",
                        "status": "1"
                    },
                    {
                        "name": "change_problem_ranking",
                        "status": "1"
                    },
                    {
                        "name": "edit_own_media",
                        "status": "1"
                    },
                    {
                        "name": "edit_user_media",
                        "status": "1"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```


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 roles 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 role de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object/array)` Propriedades da role a serem atualizadas.

A propriedade `roleid` deve ser definida para cada role, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da role, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
rules	array	Regras de acesso a serem atualizadas para a role.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos papéis atualizados na propriedade `roleids`.

Exemplos

Desabilitando a capacidade de executar scripts

Atualize a função com 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 tem as seguintes propriedades.

Property	Type	Description
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 ”action operation”: 0 - Script; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Valores possíveis se `scope` estiver definido como ”manual host action” ou ”manual event action”: 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 - action operation; 2 - manual host action; 4 - manual event action. Comportamento da propriedade: - *obrigatório* para operações de criação
execute_on	integer	Local onde o script será executado. Valores possíveis: 0 - executar no Zabbix agent; 1 - executar no Zabbix server. É *suportado* somente 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 semelhante a um menu no frontend quando clicadas em um host ou evento. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”manual host action” ou ”manual event action”

Property	Type	Description
authtype	integer	Método de autenticação usado para scripts do tipo 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 ”password”, ou se `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 ”public key”
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 ”public key”
port	string	Número da porta usado para scripts SSH e TELNET. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH” ou ”TELNET”
groupid	ID	ID do grupo de hosts no qual o script pode ser executado. Se definido como ”0”, o script estará disponível em todos os grupos de hosts. Padrão: 0.
usrgrpid	ID	ID do grupo de usuários que terá permissão para executar o script. Se definido como ”0”, o script estará disponível para todos os grupos de usuários. Padrão: 0. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”manual host action” ou ”manual event action”
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 ”manual host action” ou ”manual event action”

Property	Type	Description
confirmation	string	Texto da janela pop-up de confirmação. A janela pop-up será exibida ao tentar executar o script no frontend do Zabbix. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”manual host action” ou ”manual event action”
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 a 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 uma entrada fornecida pelo usuário. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado; Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”manual host action” ou ”manual event action”
manualinput_prompt	string	Texto do prompt de entrada manual. Comportamento da propriedade: - *obrigatório* se `manualinput` estiver definido como ”Enabled”
manualinput_validator	string	Campo de texto usado para validar a entrada fornecida pelo usuário. A string consiste em uma expressão regular ou em um conjunto de valores separados por vírgulas. Comportamento da propriedade: - *obrigatório* se `manualinput` estiver definido como ”Enabled”
manualinput_validator_type	integer	Determina o tipo de entrada do usuário esperado. 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 separada por vírgulas de possíveis valores de entrada. Comportamento da propriedade: - *suportado* se `manualinput` estiver definido como ”Enabled”

Property	Type	Description
manualinput_default_value	string	Valor padrão para preencher automaticamente a entrada do usuário. Comportamento da propriedade: - *suportado* se `manualinput_validator_type` estiver definido como "String"

Parâmetros do webhook

Os parâmetros passados para o script do webhook quando ele é chamado têm 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 de usuário. Consulte Funções de usuário para obter 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

Crie um script de webhook

Crie um script de webhook que envie uma solicitação HTTP para um serviço externo.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Criar um script SSH

Crie um script SSH com autenticação por chave pública que pode ser executado em um host e possui um menu de contexto.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Criar um script personalizado

Crie um script personalizado que irá reiniciar um server. O script irá requerer acesso de escrita ao host e irá solicitar ao usuário uma entrada manual. Após o envio bem-sucedido da entrada, o script exibirá uma mensagem de confirmação no frontend.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
        "manualinput_default_value": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Criar um script do tipo URL

Crie um script do tipo URL para escopo de host que permanece na mesma janela e possui texto de confirmação.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
```

```
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Crie um script do tipo URL com entrada manual

Crie um script do tipo URL para escopo de evento que abre em uma nova janela e possui entrada manual.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "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. Consulte Funções de usuário para obter 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; esses 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 de usuário. Consulte Funções de usuário para obter 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

Parâmetro	Tipo	Descrição
eventid	ID	ID do evento no qual executar o script.
manualinput	string	Comportamento do parâmetro: - *obrigatório* se `hostid` não estiver definido 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 de 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 informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retornar apenas scripts que podem ser executados nos grupos de hosts fornecidos.
hostids	ID/array	Retornar apenas scripts que podem ser executados nos hosts fornecidos.
scriptids	ID/array	Retornar apenas scripts com os IDs fornecidos.
usrgrpids	ID/array	Retornar apenas scripts que podem ser executados por usuários nos grupos de usuários fornecidos.
selectHostGroups	query	Retornar uma propriedade `hostgroups` com os grupos de hosts nos quais o script pode ser executado.
selectHosts	query	Retornar uma propriedade `hosts` com os hosts nos quais o script pode ser executado.
selectActions	query	Retornar uma propriedade `actions` com as ações associadas ao script.
sortfield	string/array	Classificar o resultado pelas propriedades fornecidas. Valores possíveis: `scriptid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

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 no evento informado 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 de usuário. Consulte Funções de usuário para obter 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 do macro {MANUALINPUT} fornecido pelo usuário.

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 informado 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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` O método aceita um objeto ou array de objetos com os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostid	ID	ID do host para retornar scripts. Deve ser único. Comportamento do parâmetro: - *obrigatório*
scriptid	ID	ID do script a ser retornado.
manualinput	string	Valor do macro {MANUALINPUT} fornecido pelo usuário.

Valores de retorno

`(object)` Retorna um objeto com os IDs dos hosts como propriedades e arrays de scripts disponíveis como valores. Se o ID do script for fornecido, o valor associado 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 de usuário. Consulte Funções de usuário para obter 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 service possui as seguintes propriedades.

Property	Type	Description
serviceid	ID	ID do service. Property behavior: - *read-only* - *required* for update operations
algorithm	integer	Regra de cálculo do status. Aplicável somente se existirem serviços filhos. Valores possíveis: 0 - define o status como OK; 1 - mais crítico se todos os filhos tiverem problemas; 2 - mais crítico entre os serviços filhos. Property behavior: - *required* for create operations
name	string	Nome do service. Property behavior: - *required* for create operations
sortorder	integer	Posição do service usada para ordenação. Valores possíveis: 0-999. Property behavior: - *required* for create operations
weight	integer	Peso do service. Valores possíveis: 0-1000000. Padrão: 0.

Property	Type	Description
propagation_rule	integer	Regra de propagação do status. Valores possíveis: 0 - *(default)* propaga o status do service como está - sem alterações; 1 - aumenta o status propagado por um determinado `propagation_value` (em 1 a 5 severidades); 2 - diminui o status propagado por um determinado `propagation_value` (em 1 a 5 severidades); 3 - ignora este service - o status não é propagado ao service pai; 4 - define um status fixo do service usando um determinado `propagation_value`. Property behavior: - *required* se `propagation_value` estiver definido
propagation_value	integer	Valor de propagação do status. Valores possíveis se `propagation_rule` estiver definido como "0" ou "3": 0 - Não classificado. Valores possíveis se `propagation_rule` estiver definido como "1" ou "2": 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Valores possíveis se `propagation_rule` estiver definido como "4": -1 - OK; 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Property behavior: - *required* se `propagation_rule` estiver definido
status	integer	Indica se o service está em estado OK ou com problema. Se o service estiver em estado de problema, `status` será igual à severidade do problema mais crítico ou ao status mais alto de um service filho em estado de problema. Se o service estiver em estado OK, `status` será igual a: -1. Property behavior: - *read-only*
description	string	Descrição do service.
uuid	string	Identificador universal exclusivo, usado para vincular services importados a services já existentes. Gerado automaticamente, se não for informado.
created_at	integer	Timestamp Unix em que o service foi criado.

Property	Type	Description
readonly	integer	Acesso ao service. Valores possíveis: 0 - leitura e gravação; 1 - somente leitura. Property behavior: - *read-only*

Regra de status

O objeto de regra de status possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Condição para definir o status (Novo status). Valores possíveis: 0 - se pelo menos (N) serviços filhos tiverem status (Status) ou superior; 1 - se pelo menos (N%) dos serviços filhos tiverem status (Status) ou superior; 2 - se menos de (N) serviços filhos tiverem status (Status) ou inferior; 3 - se menos de (N%) dos serviços filhos tiverem status (Status) ou inferior; 4 - se o peso dos serviços filhos com status (Status) ou superior for pelo menos (W); 5 - se o peso dos serviços filhos com status (Status) ou superior for pelo menos (N%); 6 - se o peso dos serviços filhos com status (Status) ou inferior for menor que (W); 7 - se o peso dos serviços filhos com status (Status) ou inferior for menor que (N%). Onde: - N (W) é `limit_value`; - (Status) é `limit_status`; - (Novo status) é `new_status`. Comportamento da propriedade: - *obrigatório*
limit_value	integer	Valor limite. Valores possíveis: - para N e W: 1-100000; - para N%: 1-100. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
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 de um serviço. Eles têm as seguintes propriedades.

Property	Type	Description
clock	timestamp	Momento em que a mudança de estado do serviço ocorreu.
value	integer	Status do serviço. Consulte a propriedade `status` do service 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 tem as seguintes propriedades.

Property	Type	Description
tag	string	Nome da tag de problema. Property behavior: - *required*
operator	integer	Operador da condição de mapeamento operator. Valores possíveis: 0 - *(default)* Igual a; 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 de usuário. Consulte User roles para obter 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 de usuário. Consulte Funções de usuário para obter 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 que você recupere services 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 User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
serviceids	ID/array	Retorna apenas os serviços com os IDs fornecidos.
parentids	ID/array	Retorna apenas os serviços que estão vinculados aos serviços pai fornecidos.
deep_parentids	flag	Retorna todos os serviços filhos diretos e indiretos. Usado junto com `parentids`.
childids	ID/array	Retorna apenas os serviços que estão vinculados aos serviços filhos fornecidos.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	object/array	Retorna apenas os serviços com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
problem_tags	object/array	Retorna apenas os serviços com as tags de problema fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
without_problem_tags	flag	Retorna apenas os serviços sem tags de problema.
slaids	ID/array	Retorna apenas os serviços que estão vinculados ao(s) SLA(s) específico(s).
selectChildren	query	Retorna uma propriedade `children` com os serviços filhos. Suporta `count`.
selectParents	query	Retorna uma propriedade `parents` com os serviços pai. Suporta `count`.

Parâmetro	Tipo	Descrição
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 resolvido do evento. Suporta `count`.
selectProblemTags	query	Retorna uma propriedade `problem_tags` com as tags de problema. Suporta `count`.
selectStatusRules	query	Retorna uma propriedade `status_rules` com as regras de status. Suporta `count`.
selectStatusTimeline	object/array	Retorna uma propriedade `status_timeline` contendo as mudanças de estado do serviço para os períodos fornecidos. Formato `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` sendo a data de início (inclusiva; timestamp inteiro) e `period_to` sendo a data de término (exclusiva; timestamp inteiro) para o período de interesse. Retorna um array de entradas contendo uma propriedade `start_value` e um array `alarms` para as mudanças de estado dentro dos períodos especificados.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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": ["serviceid", "name"]
    },
    "id": 1
}
```

Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
```

```
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


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 de 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 informados.

> **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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dserviceids	ID/array	Retorna apenas os serviços descobertos com os IDs fornecidos.
dhostids	ID/array	Retorna apenas os serviços descobertos que pertencem aos hosts descobertos fornecidos.
dcheckids	ID/array	Retorna apenas os serviços descobertos que foram detectados pelas verificações de descoberta fornecidas.
druleids	ID/array	Retorna apenas os serviços descobertos que foram detectados pelas regras de descoberta fornecidas.
selectDRules	query	Retorna uma propriedade `drules` com um array das regras de descoberta que detectaram o serviço.
selectDHosts	query	Retorna uma propriedade `dhosts` com um array dos hosts descobertos aos quais o serviço pertence.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts com o mesmo endereço IP e proxy que o serviço. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - o resultado será ordenado por `hostid`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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
- Agendamento de SLA
- Tempo de inatividade excluído do SLA

- Tag de serviço do SLA

Métodos disponíveis:

- sla.create - criar novos SLAs
- sla.delete - excluir SLAs
- sla.get - recuperar SLAs
- sla.getsli - recuperar dados do Indicador de Nível de Serviço (SLI) para SLAs
- sla.update - atualizar SLAs

Objeto SLA

Os seguintes objetos estão diretamente relacionados à API `sla` (Acordo de Nível de Serviço).

SLA

O objeto SLA possui as seguintes propriedades.

Propriedade	Tipo	Descrição
slaid	ID	ID do SLA. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do SLA. Comportamento da propriedade: - *obrigatório* para operações de criação
period	integer	Período de relatório do SLA. Valores possíveis: 0 - diário; 1 - semanal; 2 - mensal; 3 - trimestral; 4 - anual. Comportamento da propriedade: - *obrigatório* para operações de criação
slo	float	Objetivo de Nível de Serviço mínimo aceitável expresso em percentual. Se o Indicador de Nível de Serviço (SLI) cair abaixo, o SLA é considerado em estado de problema/não cumprido. Valores possíveis: 0-100 (até 4 casas decimais). Comportamento da propriedade: - *obrigatório* para operações de criação
effective_date	integer	Data de início de vigência do SLA. Valores possíveis: timestamp da data em UTC.
timezone	string	Fuso horário do relatório, por exemplo: `Europe/London`, `UTC`. Para a lista completa de fusos horários suportados, consulte a documentação do PHP. Comportamento da propriedade: - *obrigatório* para operações de criação
status	integer	Status do SLA. Valores possíveis: 0 - *(padrão)* SLA desabilitado; 1 - SLA habilitado.
description	string	Descrição do SLA.

Agenda de SLA

O objeto de agenda de SLA define períodos em que o(s) serviço(s) conectado(s) estão programados para estar em funcionamento. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
period_from	integer	Hora de início do período semanal recorrente (inclusive). Valores possíveis: número de segundos (contando a partir de domingo). Comportamento da propriedade: - *obrigatório*
period_to	integer	Hora de término do período semanal recorrente (exclusivo). Valores possíveis: número de segundos (contando a partir de domingo). Comportamento da propriedade: - *obrigatório*

SLA - tempo de inatividade excluído

O objeto de tempo de inatividade excluído define períodos em que o(s) serviço(s) conectado(s) estão programados para ficar fora de operação, sem afetar o SLI, por exemplo, durante uma manutenção planejada. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do tempo de inatividade excluído. Comportamento da propriedade: - *obrigatório*
period_from	integer	Hora de início do tempo de inatividade excluído (inclusivo). Valores possíveis: timestamp. Comportamento da propriedade: - *obrigatório*
period_to	integer	Hora de término do tempo de inatividade excluído (exclusivo). Valores possíveis: timestamp. Comportamento da propriedade: - *obrigatório*

Tag de serviço de SLA

O objeto de tag de serviço de SLA vincula os serviços a serem incluídos nos cálculos do SLA. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do serviço SLA. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da tag do serviço SLA. Valores possíveis: 0 - *(padrão)* Igual; 2 - Contém.
value	string	Valor da tag do serviço SLA.

sla.create

Descrição

`object sla.create(object/array SLAs)`

Este método permite criar novos objetos 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 de usuário. Consulte User roles para obter 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 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 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 SLA de acordo com os parâmetros informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
slaids	ID/array	Retorna apenas SLAs com os IDs fornecidos.
serviceids	ID/array	Retorna apenas SLAs correspondentes aos serviços específicos.
selectSchedule	query	Retorna uma propriedade `schedule` com os agendamentos do SLA. Suporta `count`.
selectExcludedDowntimes	query	Retorna uma propriedade `excluded_downtimes` com os períodos de indisponibilidade excluídos do SLA. Suporta `count`.
selectServiceTags	query	Retorna uma propriedade `service_tags` com as tags de serviço do SLA. Suporta `count`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Parâmetro	Tipo	Descrição
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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros contendo o ID do SLA, períodos de relatório e, opcionalmente, os IDs dos serviços - para calcular o SLI.

Parâmetro	Tipo	Descrição
slaid	ID	ID do SLA para retornar informações de disponibilidade. Comportamento do parâmetro: - *obrigatório*
period_from	timestamp	Timestamp inicial (inclusivo) para relatar o SLI. Valores possíveis: Unix timestamp.
period_to	timestamp	Timestamp final (inclusivo) para relatar o SLI. Valores possíveis: Unix timestamp.
periods	integer	Número de períodos a serem relatados. Valores possíveis: 1-100
serviceids	ID/array	IDs dos serviços para retornar o SLI.

Particionamento de períodos

A tabela a seguir demonstra o arranjo dos períodos retornados com base nas combinações de parâmetros.

Note:
Os períodos retornados não precedem o primeiro período disponível (com base na data efetiva do SLA) e não excedem o período atual.

Parâmetros			Períodos retornados
period_from	**period_to**	**periods**	
-	-	-	Últimos 20 períodos, incluindo o atual.

Parâmetros			Períodos retornados
-	-	N	Últimos N períodos.
-	especificado	-	Últimos 20 períodos antes de `period_to`.
-	especificado	N	Últimos N períodos antes de `period_to`.
especificado	-	-	Primeiros 20 períodos a partir de `period_from`.
especificado	-	N	Primeiros N períodos a partir de `period_from`.
especificado	especificado	-	Até 100 períodos no intervalo especificado.
especificado	especificado	N	N períodos no intervalo especificado.

Valores de retorno

`(object)` Retorna os resultados do cálculo.

Propriedade	Tipo	Descrição
periods	array	Lista dos períodos reportados. Cada período reportado é representado como um objeto consistindo de: - `period_from` - `(timestamp)` Timestamp inicial (inclusivo) do período reportado. - `period_to` - `(timestamp)` Timestamp final (exclusivo) do período reportado. Os períodos são ordenados por `period_from`, com os períodos mais antigos aparecendo primeiro.
serviceids	array	Lista de IDs de serviços nos períodos reportados. A ordem de classificação da lista não está definida. Mesmo que o parâmetro `serviceids` tenha sido passado para o método `sla.getsli`.
sli	array	Dados SLI (como um **array bidimensional**) para cada período 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 SLI

Os dados SLI retornados para cada período relatado e serviço consistem em:

Propriedade	Tipo	Descrição
uptime	integer	Quantidade de tempo que o serviço permaneceu em estado *OK* durante o tempo de atividade agendado, menos os períodos de inatividade excluídos.
downtime	integer	Quantidade de tempo que o serviço permaneceu em estado *não OK* durante o tempo de atividade agendado, menos os períodos de inatividade excluídos.
sli	float	SLI (porcentagem do tempo total de atividade), com base no uptime e downtime.
error_budget	integer	Orçamento de erro (em segundos), com base no SLI e no SLO.

Propriedade	Tipo	Descrição
excluded_downtimes	array	Array de períodos de inatividade excluídos neste período de relatório. Cada objeto conterá os seguintes parâmetros: - `name` - (`string`) Nome do período de inatividade excluído. - `period_from` - (`timestamp`) Timestamp inicial (inclusivo) do período de inatividade excluído. - `period_to` - (`timestamp`) Timestamp final (exclusivo) do período de inatividade excluído. Os períodos de inatividade excluídos são classificados por `period_from`, com os períodos mais antigos aparecendo primeiro.

Exemplos

Calculando SLI para um SLA diário

Recupere os dados de SLI para os serviços com IDs ”1” e ”4” que estão vinculados ao SLA com ID ”1”. Recupere os dados para um único período até ”1761861599” (30 de outubro de 2025 23:59:59 GMT+0200). Como o período de relatório do SLA é diário, os dados de SLI são recuperados de ”1761775200” (30 de outubro de 2025 00:00:00 GMT+0200) a ”1761861600” (31 de outubro de 2025 00:00:00 GMT+0200).

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599,
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
```

```
                            "period_from": 1761825600,
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```

Calculando SLI para um SLA mensal

Recupere os dados de SLI dos serviços com IDs ”50”, ”60” e ”70” que estão vinculados ao SLA com ID ”5”. Recupere os dados para três períodos a partir de ”1635724800” (01 de novembro de 2021 00:00:00 UTC). Como o período de relatório do SLA é mensal, os dados de SLI são recuperados para os três meses seguintes:

- De ”1635724800” (01 de novembro de 2021 00:00:00 UTC) a ”1638316800” (01 de dezembro de 2021 00:00:00 UTC)
- De ”1638316800” (01 de dezembro de 2021 00:00:00 UTC) a ”1640995200” (01 de janeiro de 2022 00:00:00 UTC)
- De ”1640995200” (01 de janeiro de 2022 00:00:00 UTC) a ”1643673600” (01 de fevereiro de 2022 00:00:00 UTC)

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
```

```
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
```

```
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


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 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 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*.

Span de APM

Esta classe foi projetada para trabalhar com spans coletados para monitoramento do desempenho de aplicações (APM).

Referências de objetos:

- Span de APM

Métodos disponíveis:

- apmspan.get - recuperar spans

Objeto de span do APM

Os objetos a seguir estão diretamente relacionados à API `apmspan`.

Note:
Cada propriedade corresponde a uma coluna na tabela `otel_traces` da fonte de dados de telemetria. Para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis.

Span de APM

O objeto de span de APM representa um span (uma única operação dentro de um trace, como uma solicitação HTTP, uma consulta ao banco de dados ou uma chamada de função).

Os spans formam uma hierarquia, começando pelo span raiz, que representa a primeira operação da solicitação. Todos os outros spans são spans filhos do span durante o qual a operação correspondente foi iniciada.

O objeto de span de APM tem as seguintes propriedades:

Propriedade	Tipo	Descrição
timestamp	integer	Hora Unix (em nanossegundos) em que o span foi iniciado. Comportamento da propriedade: - *somente leitura*
traceid	string	ID do trace ao qual o span pertence. Comportamento da propriedade: - *somente leitura*
spanid	string	ID do span. Comportamento da propriedade: - *somente leitura*
parent_spanid	string	ID do span pai. Vazio para o span raiz. Comportamento da propriedade: - *somente leitura*
trace_state	string	Dados adicionais, se houver, anexados ao trace pelas ferramentas de tracing (consulte W3C Trace Context). Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
span_name	string	Nome da operação representada pelo span (por exemplo, `mysqli_query`). Comportamento da propriedade: - *somente leitura*
span_kind	string	Função do span. Valores possíveis: `Unspecified` - o tipo do span não foi especificado; `Internal` - operação executada dentro de um aplicativo; `Server` - tratamento de uma solicitação recebida; `Client` - solicitação de saída; `Producer` - criação de uma operação assíncrona (por exemplo, envio de uma mensagem); `Consumer` - processamento de uma operação assíncrona. Comportamento da propriedade: - *somente leitura*
service_name	string	Nome do serviço que produziu o span (por exemplo, `zabbix-frontend`), obtido do atributo de recurso `service.name` do OpenTelemetry. Comportamento da propriedade: - *somente leitura*
resource_attributes	object	Atributos de recurso de chave-valor do span (por exemplo, `host.name: eabf8c2a5ddc`). Comportamento da propriedade: - *somente leitura*
scope_name	string	Nome do escopo de instrumentação que produziu o span. Comportamento da propriedade: - *somente leitura*
scope_version	string	Versão do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
span_attributes	object	Atributos do span de chave-valor do span (por exemplo, `db.system: mysql`). Comportamento da propriedade: - *somente leitura*
duration	integer	Duração do span em nanossegundos. Comportamento da propriedade: - *somente leitura*
status_code	string	Status da operação representada pelo span. Valores possíveis: `Unset` - nenhum status explícito foi atribuído ao span; `Ok` - a operação foi concluída com sucesso; `Error` - a operação foi concluída com um erro. Comportamento da propriedade: - *somente leitura*
status_message	string	Informações adicionais, se houver, sobre o status do span. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
events	array	Array de objetos event do span, ou seja, pontos relevantes no tempo durante a operação (por exemplo, exceções). Comportamento da propriedade: - *somente leitura*
links	array	Array de objetos link do span, que apontam para spans relacionados no mesmo trace ou em outro trace. Comportamento da propriedade: - *somente leitura*

apmspan.get

Descrição

`integer/array apmspan.get(object parameters)`

O método permite recuperar spans 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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
time_from	timestamp	Retorna somente spans iniciados no horário especificado ou após ele. Comportamento do parâmetro: - *obrigatório*
time_till	timestamp	Retorna somente spans iniciados antes do horário especificado. Comportamento do parâmetro: - *obrigatório*
traceids	string/array	Retorna somente spans pertencentes aos traces especificados.
spanids	string/array	Retorna somente spans com os IDs especificados.
parent_spanids	string/array	Retorna somente spans com os parent spans especificados.
resource_attributes	object/array	Retorna somente spans com os atributos de recurso especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os spans. Possíveis valores de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
resource_attributes_evaltype	integer	Método de avaliação das condições em `resource_attributes`. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.

Parâmetro	Tipo	Descrição
span_attributes	object/array	Retorna somente spans com os atributos de span especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os spans. Possíveis valores de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
span_attributes_evaltype	integer	Método de avaliação das condições em `span_attributes`. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
min_duration	integer	Retorna somente spans cuja duração seja maior ou igual ao valor especificado, em nanossegundos.
max_duration	integer	Retorna somente spans cuja duração seja menor que o valor especificado, em nanossegundos.
filter	object	Retorna somente os resultados que correspondem exatamente ao filtro especificado. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são um único valor ou uma matriz de valores a serem comparados. Propriedades compatíveis: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Retorna resultados que correspondem ao padrão especificado, sem diferenciar maiúsculas de minúsculas. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são strings a serem pesquisadas. Se nenhuma opção adicional for especificada, será realizada uma pesquisa `LIKE "%…%"`. Propriedades compatíveis: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Número de registros a ignorar antes de retornar o resultado.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna um dos seguintes:

- um array de objetos;
- a quantidade de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando a hierarquia de spans de um trace

Recupere os quatro primeiros spans do trace "dee625f33cb0994e86c5cd4640e3b09c", ordenados pelo horário de início. A propriedade `parent_spanid` mostra como os spans estão relacionados: o span raiz não tem um pai, e cada outro span faz referência ao span que o iniciou.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```

Veja também

- Rastreamento APM
- Log APM
- Métrica APM

Origem

CApmSpan::get() em *ui/include/classes/api/services/CApmSpan.php*.

Task

Esta classe foi projetada para trabalhar com tarefas (como verificar items ou regras de descoberta de baixo nível sem recarregar a configuração).

Referências de objetos:

- Task
  - Objeto de solicitação 'Executar agora'
  - Objeto de solicitação 'Atualizar configuração do proxy'
  - Objeto de solicitação 'Informações de diagnóstico'
    - Objeto de solicitação de estatística
    - Objeto de resultado de estatística

Métodos disponíveis:

- task.create - criar novas tarefas
- task.get - recuperar tarefas

Objeto Task

Os seguintes objetos estão diretamente relacionados à API `task`.

Task

O objeto task possui as seguintes propriedades:

Propriedade	Tipo	Descrição
taskid	ID	ID da task. Comportamento da propriedade: - *somente leitura*
type	integer	Tipo da task. Valores possíveis: 1 - Informações de diagnóstico; 2 - Atualizar configuração do proxy; 6 - Executar agora. Usuários do tipo *Admin* e *User* podem criar apenas tasks 'Executar agora'. Observe que `task.get` sempre retorna "7" (Resumo da execução da task). Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
status	integer	Status da task. Valores possíveis: 1 - nova task; 2 - task em andamento; 3 - task concluída; 4 - task expirada. Comportamento da propriedade: - *somente leitura*
clock	timestamp	Hora em que a task foi criada. Comportamento da propriedade: - *somente leitura*
ttl	integer	O 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”
request	object	Objeto de solicitação da task de acordo com o tipo de 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 das 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 e gravação* permission, ou permissão de *leitura* e a ação `invoke_execute_now` action habilitada para sua função. O mesmo se aplica a usuários do tipo *Admin* para regras de descoberta de baixo nível (LLD).

Objeto de solicitação ’Atualizar configuração do proxy’

O objeto de solicitação da tarefa ’Atualizar configuração do proxy’ possui as seguintes propriedades.

Propriedade	Tipo	Descrição
proxyids	array	IDs do proxy.

Objeto de solicitação de 'Informações de diagnóstico'

O objeto de solicitação de tarefa de informações de diagnóstico possui as seguintes propriedades. O objeto de solicitação de estatísticas para todos os tipos de propriedades é descrito em detalhes abaixo.

Propriedade	Tipo	Descrição
historycache	object	Solicitação de estatísticas do cache de histórico. Disponível no server e proxy.
valuecache	object	Solicitação de estatísticas do cache de items. Disponível no server.
preprocessing	object	Solicitação de estatísticas do gerenciador de pré-processamento. Disponível no server e proxy.
alerting	object	Solicitação de estatísticas do gerenciador de alertas. Disponível no server.
lld	object	Solicitação de estatísticas do gerenciador 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 ordenar e limitar as informações solicitadas.

Tipo de diagnóstico	Campos disponíveis	Tipo
historycache	values	integer

Tipo de diagnóstico	Campos disponíveis	Tipo
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 se ocorreu um erro durante a execução da tarefa.

task.create

Descrição

`object task.create(object/array tasks)`

Este método permite que você crie tasks.

> **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 User roles para obter 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 que você recupere tasks 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 User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
taskids	ID/array	Retornar apenas tarefas com os IDs fornecidos.
output	query	Esses parâmetros são descritos no comentário de referência.
preservekeys	boolean	

Valores de retorno

`(integer/array)` Retorna um array de objetos.

Exemplos

Recuperar tarefa por ID

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.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
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 do template. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do template. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template. Comportamento da propriedade: - *somente leitura*.

Propriedade	Tipo	Descrição
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

template.create

Descrição

`object template.create(object/array templates)`

Este método permite criar novos templates.

> **Note:**
> Este método está disponível apenas para 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)` 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.

Excluir um template também remove todas as suas entidades (items, triggers, gráficos etc.) de quaisquer hosts ou templates aos quais ele esteja vinculado. Para manter essas entidades, primeiro desvincule o template dos hosts ou templates relevantes usando os métodos `host.update`, `host.massremove`, `template.update` ou `template.massremove`.

> **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 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 informados.

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)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
templateids	ID/array	Retorna apenas templates com os IDs de template fornecidos.
groupids	ID/array	Retorna apenas templates que pertencem aos grupos de templates fornecidos.
parentTemplateids	ID/array	Retorna apenas templates aos quais o template fornecido está vinculado.
hostids	ID/array	Retorna apenas templates que estão vinculados aos hosts/templates fornecidos.
graphids	ID/array	Retorna apenas templates que contêm os gráficos fornecidos.
itemids	ID/array	Retorna apenas templates que contêm os items fornecidos.
triggerids	ID/array	Retorna apenas templates que contêm os triggers fornecidos.
with_items	flag	Retorna apenas templates que possuem items.
with_triggers	flag	Retorna apenas templates que possuem triggers.
with_graphs	flag	Retorna apenas templates que possuem gráficos.
with_httptests	flag	Retorna apenas templates que possuem cenários web.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.

Parâmetro	Tipo	Descrição
tags	object/array	Retorna apenas templates com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<valor>", "operator": "<operador>"}, ...]`. Um array vazio retorna todos os templates. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
inheritedTags	boolean	Retorna templates que possuem as tags fornecidas também em templates vinculados. Valores possíveis: `true` - templates vinculados também devem possuir as tags fornecidas; `false` - *(padrão)* tags de templates vinculados são ignoradas.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags que estão em templates vinculados.
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`.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o template pertence na propriedade `templategroups`.
selectTemplates	query	Retorna templates aos quais o template fornecido está vinculado na propriedade `templates`. Suporta `count`.
selectParentTemplates	query	Retorna 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 items do template na propriedade `items`. Suporta `count`.
selectTriggers	query	Retorna triggers do template na propriedade `triggers`. Suporta `count`.
selectGraphs	query	Retorna gráficos do template na propriedade `graphs`. Suporta `count`.
selectMacros	query	Retorna as macros do template na propriedade `macros`.
selectDashboards	query	Retorna dashboards do template na propriedade `dashboards`. Suporta `count`.
selectValueMaps	query	Retorna uma propriedade `valuemaps` com mapas de valores do template.

Parâmetro	Tipo	Descrição
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	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	
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 by Zabbix agent” e ”Windows by Zabbix agent”.

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": "This is an official Linux template. It requires Zabbix agent 8.0 or newer.\r\n
            "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": "## Overview\r\n\r\nThis is an official Linux template. It requires Zabbix agent 8.0
        },
        {
            "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": "This is an official Windows template. It requires Zabbix agent 8.0 or newer.\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": "## Overview\r\n\r\nThis is an official Windows template. It requires Zabbix agent &
        }
    ],
    "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

Recuperar templates que possuem a tag ”host-name” igual a ”{HOST.NAME}”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "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 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)` Parâmetros contendo os IDs dos templates a serem atualizados e os objetos a serem adicionados aos templates.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
templates	object/array	Templates a serem atualizados. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*
groups	object/array	Grupos de template para adicionar os templates fornecidos. Os grupos de template devem ter apenas a propriedade `groupid` definida.
macros	object/array	Macros de usuário a serem criadas para os templates fornecidos.
templates_link	object/array	Templates para vincular aos templates fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.

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 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)` 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.update

Descrição

`object template.update(object/array templates)`

Este método permite atualizar templates existentes.

> **Note:**
> Este método está disponível apenas para 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 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 vários grupos de templates

Substitua todos os grupos de templates para vários templates por outros diferentes.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10086",
            "groups": [
                {
                    "groupid": "24"
                }
            ]
        },
        {
            "templateid": "10087",
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        },
        {
            "templateid": "10088",
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "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 múltiplas macros de template

Substitua todas as macros de usuário pela macro de usuário fornecida em vários templates.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout after which agent is considered unavailable. Works only fo
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout after which agent is considered unavailable. Works only fo
                    }
                ]
            }
        ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "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": "10001"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Limpando vários templates vinculados a um template

Desvincula e limpa um template vinculado específico dos templates fornecidos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "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 objetos:

- Dashboard de template
- Página de dashboard de template
  - Widget de dashboard de template
    - Campo de widget de 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.

Property	Type	Description
dashboardid	ID	ID do dashboard de template. Property behavior: - *read-only* - *required* for update operations
name	string	Nome do dashboard de template. Property behavior: - *required* for create operations
templateid	ID	ID do template ao qual o dashboard pertence. Property behavior: - *constant* - *required* for create operations
display_period	integer	Intervalo padrão da apresentação de slides (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 a apresentação de slides automaticamente; 1 - *(padrão)* iniciar a apresentação de slides automaticamente.
uuid	string	Identificador único universal, usado para vincular dashboards de template importados a outros já existentes. Gerado automaticamente, se não for informado.

Página de dashboard do template

O objeto da página de dashboard do template possui as seguintes propriedades.

Property	Type	Description
dashboard_pageid	ID	ID da página de dashboard. Property behavior: - *read-only*
name	string	Nome da página de dashboard. Padrão: string vazia.
display_period	integer	Intervalo da apresentação de slides do dashboard (em segundos). Valores possíveis: 0, 10, 30, 60, 120, 600, 1800, 3600. Padrão: 0 (usará o intervalo padrão da apresentação de slides).
widgets	array	Array dos objetos template dashboard widget.

Widget de dashboard de template

O objeto widget de dashboard de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
widgetid	ID	ID do widget de dashboard. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
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; scatterplot - Gráfico de dispersão; 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 do widget do dashboard do template

O objeto de campo do widget do dashboard do 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 do widget do dashboard. Observe que alguns parâmetros relacionados ao host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade do host*, etc.) não estão disponíveis ao configurar o widget em um dashboard de template. Isso ocorre porque dashboards de template exibem dados apenas do host ao qual o template está vinculado. Comportamento da propriedade: - *obrigatório*
value	mixed	Valor do campo do widget dependendo do tipo. Valores possíveis: veja Campos do widget do dashboard. Observe que alguns parâmetros relacionados ao host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade do host*, etc.) não estão disponíveis ao configurar o widget em um dashboard de template. Isso ocorre porque dashboards de template exibem dados apenas do host ao qual o template está vinculado. Comportamento da propriedade: - *obrigatório*

templatedashboard.create

Descrição

`object templatedashboard.create(object/array templateDashboards)`

Este método permite criar novos dashboards de template.

> **Note:**
> Este método está disponível apenas para 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)` 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 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 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 informados.

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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dashboardids	ID/array	Retorna apenas os dashboards de template com os IDs fornecidos.
templateids	ID/array	Retorna apenas os 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 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 de dashboard é obrigatório 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 foi projetada para trabalhar com tipos de mídia.

Referências de objetos:

- Tipo de mídia
  - Parâmetros do webhook
  - Parâmetros do script
- Modelo de mensagem

Métodos disponíveis:

- mediatype.create - criar novos tipos de mídia
- mediatype.delete - excluir tipos de mídia
- mediatype.get - recuperar tipos de mídia
- mediatype.update - atualizar tipos de mídia

Objeto media type

Os seguintes objetos estão diretamente relacionados à API `mediatype`.

Tipo de mídia

O objeto de tipo de mídia tem as seguintes propriedades.

Property	Type	Description
mediatypeid	ID	ID do tipo de mídia. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
name	string	Nome do tipo de mídia. Comportamento da propriedade: - *obrigatória* para operações de criação
type	integer	Transporte usado pelo tipo de mídia. Valores possíveis: 0 - E-mail; 1 - Script; 2 - SMS; 4 - Webhook; 5 - Push. Comportamento da propriedade: - *obrigatória* para operações de criação
exec_path	string	Nome do arquivo de script (por exemplo, notification.sh) localizado no diretório especificado no parâmetro de configuração do server AlertScriptsPath. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Script”
gsm_modem	string	Nome do dispositivo serial do modem GSM. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”SMS”
passwd	string	Senha de autenticação. Comportamento da propriedade: - *compatível* se `smtp_authentication` estiver definido como ”Normal password”
provider	integer	Provedor de e-mail. Valores possíveis: 0 - *(padrão)* SMTP genérico; 1 - Gmail; 2 - Relay do Gmail; 3 - Office365; 4 - Relay do Office365.
smtp_email	string	Endereço de e-mail a partir do qual as notificações serão enviadas. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Email”
smtp_helo	string	HELO SMTP. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Email”
smtp_server	string	server SMTP. Comportamento da propriedade: - *obrigatória* se `type` estiver definido como ”Email”

Property	Type	Description
smtp_port	integer	Porta do server SMTP à qual se conectar. Padrão: 25. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Email”
smtp_security	integer	Nível de segurança da conexão SMTP a ser usado. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - STARTTLS; 2 - SSL/TLS. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Email”
smtp_verify_host	integer	Verificação SSL do host para SMTP. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *compatível* se `smtp_security` estiver definido como ”STARTTLS” ou ”SSL/TLS”
smtp_verify_peer	integer	Verificação SSL do peer para SMTP. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *compatível* se `smtp_security` estiver definido como ”STARTTLS” ou ”SSL/TLS”
smtp_authentication	integer	Método de autenticação SMTP a ser usado. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - Senha normal; 2 - Token OAuth. A autenticação OAuth não é permitida para o provedor de e-mail *Office365 relay*. Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Email”
redirection_url	string	URL do frontend do Zabbix para redirecionar de volta à autorização OAuth. Padrão: Valor da propriedade `url` das configurações da API com a parte `zabbix.php?action=oauth.authorize` Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como ”OAuth token”
client_id	string	Identificador do cliente registrado no server de autorização OAuth. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como ”OAuth token”

Property	Type	Description
client_secret	string	Segredo do cliente registrado no server de autorização OAuth. Acessível somente a usuários do tipo Super Admin. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como "OAuth token"
authorization_url	string	URL OAuth, com parâmetros, para obter tokens de acesso e de atualização. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como "OAuth token"
token_url	string	URL OAuth para trocar o token de autorização por tokens de acesso e de atualização. Essa URL também é usada pelo server para atualizar um token de acesso inválido. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como "OAuth token"
tokens_status	integer	Máscara de bits do status dos tokens. Valores possíveis: 0 - (padrão) Ambos os tokens contêm um valor inválido 1 - O token de acesso contém um valor válido 2 - O token de atualização contém um valor válido 3 - Ambos os tokens contêm valores válidos. Comportamento da propriedade: - *compatível* se `smtp_authentication` estiver definido como "OAuth token"
access_token	string	Valor do token de acesso OAuth. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como "OAuth token"
access_token_updated	timestamp	Timestamp da última modificação de `access_token` feita pelo server durante a atualização com `refresh_token` ou pela API quando os tokens são alterados. Comportamento da propriedade: - *compatível* se `smtp_authentication` estiver definido como "OAuth token"
access_expires_in	integer	Tempo, em segundos, até que `access_token` fique desatualizado e seja necessário fazer uma solicitação para `refresh_url`. É definido pelo server do Zabbix durante a atualização de `access_token` ou pela API quando os tokens são alterados. O timestamp é calculado adicionando o valor de `access_token_updated`. Comportamento da propriedade: - *compatível* se `smtp_authentication` estiver definido como "OAuth token"
refresh_token	string	Valor do token de atualização OAuth. Comportamento da propriedade: - *obrigatória* se `smtp_authentication` estiver definido como "OAuth token"

Property	Type	Description
status	integer	Indica se o tipo de mídia está habilitado. Valores possíveis: 0 - *(padrão)* Habilitado; 1 - Desabilitado.
username	string	Nome de usuário. Comportamento da propriedade: - *compatível* se `smtp_authentication` estiver definido como "Normal password"
maxsessions	integer	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 "Email", "Script" ou "Webhook": 0-100. Padrão: 1.
maxattempts	integer	Número máximo de tentativas para enviar um alerta. Valores possíveis: 1-100. Padrão: 3.
attempt_interval	string	Intervalo entre as tentativas de repetição. Aceita um número de segundos (por exemplo, 1, 30, 60) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`, `1h`). Valores possíveis: 0-1h. Padrão: 10s.
message_format	integer	Formato da mensagem. Valores possíveis: 0 - Texto simples; 1 - *(padrão)* HTML. Comportamento da propriedade: - *compatível* se `type` estiver definido como "Email"
script	text	Corpo do script do Webhook (JavaScript). Comportamento da propriedade: - *obrigatória* se `type` estiver definido como "Webhook"
timeout	string	Tempo limite do script do Webhook. Aceita um número de segundos (por exemplo, 1, 60) ou um valor com um sufixo de tempo (por exemplo, `30s`, `1m`) Valores possíveis: 1-60s. Padrão: 30s. Comportamento da propriedade: - *compatível* se `type` estiver definido como "Webhook"
process_tags	integer	Processa os valores das propriedades JSON na resposta do script do Webhook como tags. Essas tags são adicionadas a quaisquer tags de problema existentes. Valores possíveis: 0 - *(padrão)* Ignorar a resposta do script do Webhook; 1 - Processar a resposta do script do Webhook como tags. Comportamento da propriedade: - *compatível* se `type` estiver definido como "Webhook"

Property	Type	Description
show_event_menu	integer	Inclui uma entrada no menu de eventos que direciona para uma URL personalizada. Também adiciona a propriedade `urls` à saída de problem.get e event.get. Valores possíveis: 0 - *(padrão)* Não incluir a entrada no menu de eventos nem a propriedade `urls`; 1 - Incluir a entrada no menu de eventos e a propriedade `urls`.
event_menu_url	string	Comportamento da propriedade: - *compatível* se `type` estiver definido como ”Webhook” URL usada na entrada do menu de eventos e na propriedade `urls` retornada por problem.get e event.get.
event_menu_name	string	Comportamento da propriedade: - *compatível* se `show_event_menu` estiver definido como ”Include event menu entry and `urls` property” Nome usado na entrada do menu de eventos e na propriedade `urls` retornada por problem.get e event.get.
parameters	array	Comportamento da propriedade: - *compatível* se `show_event_menu` estiver definido como ”Include event menu entry and `urls` property” Parâmetros de Webhook ou de script.
description	text	Comportamento da propriedade: - *compatível* 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 somente para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações da função de usuário. Consulte Funções de usuário para obter mais informações.
 Se os métodos de entrega forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend, o usuário não poderá chamar o método para os tipos de mídia correspondentes.

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 somente para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações da função de usuário. Consulte Funções de usuário para obter mais informações.
 Se os métodos de entrega forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend, o usuário não poderá chamar o método para os tipos de mídia correspondentes.

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 da função de usuário. Consulte Funções de usuário para obter mais informações.
 Se os métodos de entrega forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend, o usuário não poderá chamar o método para os tipos de mídia correspondentes.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

> **Note:**
> Ao solicitar informações relacionadas ao usuário dos tipos de mídia, usuários do tipo *Admin* 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	Retornar apenas os tipos de mídia com os IDs fornecidos.
mediaids	ID/array	Retornar apenas os tipos de mídia usados pela mídia fornecida.
userids	ID/array	Retornar apenas os tipos de mídia usados pelos usuários fornecidos.
selectActions	query	Retornar uma propriedade `actions` com as ações que usam o tipo de mídia.
selectMessageTemplates	query	Retornar 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	Retornar 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	Classificar o resultado pelas propriedades fornecidas. Valores possíveis: `mediatypeid`.
filter	object	Retornar apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type.
output	query	Possíveis propriedades do objeto Media type para usuários do tipo *Admin* e *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`. 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*. Padrão: `extend`.

Parâmetro	Tipo	Descrição
search	object	Retornar resultados que correspondam ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings para pesquisar. 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`.
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 media types 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 da função de usuário. Consulte Funções de usuário para obter mais informações.
 Se os métodos de entrega forem especificados no sinalizador de recurso `$ZBX_FEATURE_FLAGS['media_type_denylist']` no arquivo de configuração do frontend, o usuário não poderá chamar o método para os media types correspondentes.

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.

Propriedade	Tipo	Descrição
userid	ID	ID do usuário ao qual o token foi atribuído. Padrão: *usuário atual*. Comportamento da propriedade: - *constante*
lastaccess	timestamp	Data e hora mais recente em que o token foi autenticado. ”0” se o token nunca foi autenticado. Comportamento da propriedade: - *somente leitura*
status	integer	Status do token. Valores possíveis: 0 - *(padrão)* token habilitado; 1 - token desabilitado.
expires_at	timestamp	Data e hora de expiração do token. ”0” para tokens que nunca expiram.
created_at	timestamp	Data e hora de criação do token. Comportamento da propriedade: - *somente leitura*
creator_userid	ID	ID do usuário que criou o token. Comportamento da propriedade: - *somente leitura*

token.create

Descrição

`object token.create(object/array tokens)`

Este método permite criar novos tokens.

> **Note:**
> A permissão *Gerenciar tokens da API* é necessária para que a função de usuário gerencie tokens de outros usuários.

> **Attention:**
> Um token criado por este método também precisa ser gerado antes de poder ser usado.

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 *Manage API tokens* 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 *Manage API tokens* é necessária para que a função de usuário gerencie tokens de outros usuários.

Attention:
Um token pode ser gerado por este método somente se 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 informados.

> **Note:**
> Somente usuários do tipo *Super admin* têm permissão para 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	Retornar apenas tokens com os IDs fornecidos.
userids	ID/array	Retornar apenas tokens criados para os usuários fornecidos.
token	string	Retornar apenas tokens criados para o *Auth token* fornecido.
valid_at	timestamp	Retornar apenas tokens que são válidos (não expirados) na data e hora fornecidas.
expired_at	timestamp	Retornar apenas tokens que estão expirados (não válidos) na data e hora fornecidas.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	

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` 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 *Manage API tokens* é necessária para que a função de usuário gerencie tokens de outros usuários.

Parâmetros

`(object/array)` Propriedades do token a serem atualizadas.

A propriedade `tokenid` deve ser definida para cada token, todas as outras propriedades são opcionais. 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*.

Trace de APM

Esta classe foi projetada para trabalhar com traces coletados para monitoramento do desempenho de aplicações (APM).

Referências de objetos:

- Trace de APM
- Evento
- Link

Métodos disponíveis:

- apmtrace.get - recuperar traces

Objeto de trace APM

Os objetos a seguir estão diretamente relacionados à API `apmtrace`.

Note:
Cada propriedade, exceto `span_count` e `error_count`, corresponde a uma coluna na tabela `otel_traces` da fonte de dados de telemetria. A maioria das propriedades corresponde à coluna com o mesmo nome (por exemplo, a propriedade `span_name` corresponde à coluna `SpanName`). Quando os nomes são diferentes, as colunas da fonte de dados são especificadas nas tabelas abaixo. Para obter a lista de colunas, consulte Colunas de dados de telemetria compatíveis.

Trace de APM

O objeto de trace de APM representa um trace (o caminho percorrido por uma solicitação através de suas aplicações e serviços). Um trace consiste em um ou mais spans; cada span representa uma única operação.

O objeto de trace de APM baseia-se no span raiz do trace (o span que inicia o trace e não tem um span pai). Todas as propriedades contêm dados do span raiz, exceto `span_count` e `error_count`, que são calculadas a partir de todos os spans do trace.

O objeto de trace de APM tem as seguintes propriedades:

Property	Type	Description
timestamp	integer	Horário Unix (em nanossegundos) em que o span raiz foi iniciado. Comportamento da propriedade: - *somente leitura*
traceid	string	ID do trace, compartilhado por todos os spans que pertencem ao trace. Comportamento da propriedade: - *somente leitura*
spanid	string	ID do span raiz. Comportamento da propriedade: - *somente leitura*
trace_state	string	Dados adicionais, se houver, anexados ao trace pelas ferramentas de tracing (consulte W3C Trace Context). Comportamento da propriedade: - *somente leitura*
span_name	string	Nome da operação representada pelo span raiz (por exemplo, `POST /zabbix.php`). Comportamento da propriedade: - *somente leitura*
span_kind	string	Função do span raiz. Valores possíveis: `Unspecified` - o tipo do span não foi especificado; `Internal` - operação executada dentro de uma aplicação; `Server` - tratamento de uma solicitação recebida; `Client` - solicitação de saída; `Producer` - criação de uma operação assíncrona (por exemplo, envio de uma mensagem); `Consumer` - processamento de uma operação assíncrona. Comportamento da propriedade: - *somente leitura*
service_name	string	Nome do serviço que produziu o span raiz (por exemplo, `zabbix-nginx`), obtido do atributo de recurso `service.name` do OpenTelemetry. Comportamento da propriedade: - *somente leitura*
resource_attributes	object	Atributos de recurso de chave-valor do span raiz (por exemplo, `host.name: eabf8c2a5ddc`). Comportamento da propriedade: - *somente leitura*

Property	Type	Description
scope_name	string	Nome do escopo de instrumentação que produziu o span raiz. Comportamento da propriedade: - *somente leitura*
scope_version	string	Versão do escopo de instrumentação. Comportamento da propriedade: - *somente leitura*
span_attributes	object	Atributos do span de chave-valor do span raiz (por exemplo, `http.method: POST`). Comportamento da propriedade: - *somente leitura*
duration	integer	Duração do trace em nanossegundos. Comportamento da propriedade: - *somente leitura*
status_code	string	Status da operação representada pelo span raiz. Valores possíveis: `Unset` - nenhum status explícito foi atribuído ao span raiz; `Ok` - a operação foi concluída com sucesso; `Error` - a operação foi concluída com um erro. Comportamento da propriedade: - *somente leitura*
status_message	string	Informações adicionais, se houver, sobre o status do span raiz. Comportamento da propriedade: - *somente leitura*
events	array	Array de objetos event do span raiz, ou seja, pontos notáveis no tempo durante a operação (por exemplo, exceções). Comportamento da propriedade: - *somente leitura*
links	array	Array de objetos link do span raiz, que apontam para spans relacionados no mesmo trace ou em outro trace. Comportamento da propriedade: - *somente leitura*
span_count	integer	Número total de spans no trace. Comportamento da propriedade: - *somente leitura*
error_count	integer	Número de spans no trace com `status_code` definido como `Error`. Comportamento da propriedade: - *somente leitura*

Evento

O objeto de evento tem as seguintes propriedades.

Property	Type	Description
timestamp	integer	Hora Unix (em nanossegundos) em que o evento ocorreu. Coluna da fonte de dados: `Events.Timestamp`
name	string	Nome do evento. Coluna da fonte de dados: `Events.Name`

Property	Type	Description
attributes	object	Atributos de chave-valor com informações adicionais, se houver, sobre o evento. Coluna da fonte de dados: `Events.Attributes`

Link

O objeto link tem as seguintes propriedades.

Property	Type	Description
traceid	string	ID do trace que contém o span vinculado. Coluna da fonte de dados: `Links.TraceId`
spanid	string	ID do span vinculado. Juntamente com `traceid`, identifica exclusivamente o span vinculado. Coluna da fonte de dados: `Links.SpanId`
trace_state	string	Estado do trace, se houver, do span vinculado (consulte W3C Trace Context). Coluna da fonte de dados: `Links.TraceState`
attributes	object	Atributos de chave-valor com informações adicionais sobre o link. Coluna da fonte de dados: `Links.Attributes`

apmtrace.get

Descrição

`integer/array apmtrace.get(object parameters)`

O método permite recuperar traces 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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
time_from	timestamp	Retorna somente os traces cujo span raiz foi iniciado no horário especificado ou posteriormente. Comportamento do parâmetro: - *obrigatório*
time_till	timestamp	Retorna somente os traces cujo span raiz foi iniciado antes do horário especificado. Comportamento do parâmetro: - *obrigatório*
traceids	string/array	Retorna somente os traces com os IDs especificados.

Parâmetro	Tipo	Descrição
resource_attributes	object/array	Retorna somente os traces cujo span raiz tenha os atributos de recurso especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os traces. Valores possíveis de operator: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
resource_attributes_evaltype	integer	Método de avaliação das condições em `resource_attributes`. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
span_attributes	object/array	Retorna somente os traces cujo span raiz tenha os atributos de span especificados. Formato: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os traces. Valores possíveis de operator: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 4 - Existe; 5 - Não existe.
span_attributes_evaltype	integer	Método de avaliação das condições em `span_attributes`. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
min_duration	integer	Retorna somente os traces cujo `duration` do span raiz seja maior ou igual ao valor especificado, em nanossegundos.
max_duration	integer	Retorna somente os traces cujo `duration` do span raiz seja menor que o valor especificado, em nanossegundos.
filter	object	Retorna somente os resultados que correspondam exatamente ao filtro especificado. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são um único valor ou uma matriz de valores a serem comparados. Propriedades compatíveis: `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Retorna os resultados que correspondam ao padrão especificado (sem distinção entre maiúsculas e minúsculas). Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são strings a serem pesquisadas. Se nenhuma opção adicional for especificada, será realizada uma pesquisa `LIKE "%…%"`. Propriedades compatíveis: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.

Parâmetro	Tipo	Descrição
sortfield	string/array	Ordena o resultado pelas propriedades especificadas. Valores possíveis: `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Número de registros a ignorar antes de retornar o resultado.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna um dos seguintes:

- um array de objetos;
- a quantidade de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando traces lentos

Recupere os dois traces mais lentos do serviço `zabbix-nginx`. Pesquise traces das 10:00 às 11:00 de 1º de outubro de 2026 (UTC) e inclua apenas aqueles que levaram 1 segundo ou mais.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
```

```
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```


Veja também

- APM span
- APM metric
- APM log

Origem

CApmTrace::get() em *ui/include/classes/api/services/CApmTrace.php*.

Trend

Esta classe foi projetada para trabalhar com dados de tendência.

Referências de objetos:

- Tendência de ponto flutuante
- Tendência de inteiro

Métodos disponíveis:

- trend.get - recuperar tendências

Objeto de tendência

Os seguintes objetos estão diretamente relacionados à API `trend`.

Note:
Os objetos de tendência diferem dependendo do tipo de informação do item. Eles são criados pelo Zabbix server e não podem ser modificados via API.

Tendência de ponto flutuante

O objeto de tendência de ponto flutuante possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Timestamp de uma hora para a qual o valor foi calculado. Por exemplo, o timestamp ”04:00:00” significa valores calculados para o período ”04:00:00-04:59:59”.
itemid	ID	ID do item relacionado.
num	integer	Número de valores que estavam disponíveis para a hora.
value_min	float	Valor mínimo por hora.
value_avg	float	Valor médio por hora.
value_max	float	Valor máximo por hora.

Tendência de inteiro

O objeto de tendência de inteiro possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Timestamp de uma hora para a qual o valor foi calculado. Por exemplo, o timestamp ”04:00:00” significa valores calculados para o período ”04:00:00-04:59:59”.
itemid	ID	ID do item relacionado.
num	integer	Número de valores que estavam disponíveis para a hora.
value_min	integer	Valor mínimo por hora.
value_avg	integer	Valor médio por hora.
value_max	integer	Valor máximo por hora.

trend.get

Descrição

`integer/array trend.get(object parameters)`

O método permite recuperar dados de tendência de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo.
As permissões para chamar o método podem ser revogadas nas configurações de função de usuário.
Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
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

Propriedade	Tipo	Descrição
expression	string	Expressão do trigger. Itens com `value_type` definido como 5 (binário) ou 6 (JSON) não podem ser usados em expressões de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
event_name	string	Nome do evento gerado pelo trigger.
opdata	string	Dados operacionais.
comments	string	Descrição adicional do trigger.
error	string	Texto do erro caso tenha havido algum problema ao atualizar o estado do trigger. Comportamento da propriedade: - *somente leitura*
flags	integer	Origem do trigger. Valores possíveis: 0 - *(padrão)* um trigger simples; 4 - um trigger convertido de 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 da expressão do trigger. Valores possíveis: 0 - *(padrão)* o estado do trigger está atualizado; 1 - o estado atual do trigger é desconhecido. Comportamento da propriedade: - *somente leitura*
status	integer	Se o trigger está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
templateid	ID	ID do trigger do template pai. Comportamento da propriedade: - *somente leitura*
type	integer	Se o trigger pode gerar múltiplos eventos de problema. Valores possíveis: 0 - *(padrão)* não gerar múltiplos eventos; 1 - gerar múltiplos eventos.
url	string	URL associada ao trigger.
url_name	string	Rótulo para a URL associada ao trigger.

Propriedade	Tipo	Descrição
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 de recuperação do trigger. Itens com `value_type` definido como 5 (binário) ou 6 (JSON) não podem ser usados em expressões de recuperação de trigger.
correlation_mode	integer	Fechamento de eventos OK. Valores possíveis: 0 - *(padrão)* Todos os problemas; 1 - Todos os problemas se os valores das tags coincidirem.
correlation_tag	string	Tag para correspondência.
manual_close	integer	Permitir fechamento manual. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim.
uuid	string	Identificador único universal, usado para vincular triggers importados a triggers 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. Comportamento da propriedade: - *somente leitura*.
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template; 1 - host; 3 - item. Comportamento da propriedade: - *somente leitura*.

Propriedade	Tipo	Descrição
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.
automatic	integer	Tipo de tag de trigger: Valores possíveis: 0 - *(padrão)* tag criada pelo usuário; 1 - tag criada por LLD;Comportamento da propriedade: - *somente leitura*.

trigger.create

Descrição

`object trigger.create(object/array triggers)`

Este método permite criar novos triggers.

> **Note:**
> Este método está disponível apenas para 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)` 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 de trigger.

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 dois triggers, cada um dependendo de outro trigger.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Processor load is too high on {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Service has stopped",
```

```
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
                {
                    "triggerid": "17368"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::create() 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 de usuário. Consulte Funções de usuário para obter 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 informados.

> **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 User roles para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
triggerids	ID/array	Retorna somente os triggers com os IDs fornecidos.
groupids	ID/array	Retorna somente os triggers pertencentes a hosts ou templates dos grupos de hosts ou grupos de templates fornecidos.
templateids	ID/array	Retorna somente os triggers pertencentes aos templates fornecidos.
hostids	ID/array	Retorna somente os triggers pertencentes aos hosts fornecidos.
itemids	ID/array	Retorna somente os triggers que contêm os items fornecidos.
maintenanceids	ID/array	Retorna somente os triggers afetados pelas manutenções fornecidas.
functions	string/array	Retorna somente os triggers que usam as funções fornecidas. Consulte a página de funções compatíveis para obter uma lista das funções compatíveis.
group	string	Retorna somente os triggers pertencentes a hosts ou templates do grupo de hosts ou grupo de templates com o nome fornecido.
host	string	Retorna somente os triggers pertencentes ao host com o nome técnico fornecido.
inherited	boolean	Se definido como `true`, retorna somente os triggers herdados de um template.
templated	boolean	Se definido como `true`, retorna somente os triggers pertencentes a templates.

Parâmetro	Tipo	Descrição
dependent	boolean	Se definido como `true`, retorna somente os triggers que têm dependências. Se definido como `false`, retorna somente os triggers que não têm dependências.
monitored	flag	Retorna somente os triggers habilitados que pertencem a hosts monitorados e contêm apenas items habilitados.
active	flag	Retorna somente os triggers habilitados que pertencem a hosts monitorados.
maintenance	boolean	Se definido como `true`, retorna somente os triggers habilitados que pertencem a hosts em manutenção.
withUnacknowledgedEvents	flag	Retorna somente os triggers que têm eventos não reconhecidos.
withAcknowledgedEvents	flag	Retorna somente os triggers com todos os eventos reconhecidos.
withLastEventUnacknowledged	flag	Retorna somente os triggers cujo último evento não foi reconhecido.
skipDependent	flag	Ignora triggers em estado de problema que dependem de outros triggers. Observe que os outros triggers são ignorados se estiverem desabilitados, tiverem items desabilitados ou pertencerem a hosts com items desabilitados.
lastChangeSince	timestamp	Retorna somente os triggers que alteraram seu estado após o horário fornecido.
lastChangeTill	timestamp	Retorna somente os triggers que alteraram seu estado antes do horário fornecido.
only_true	flag	Retorna somente os triggers que estiveram recentemente em estado de problema (dentro de `ok_period`).
min_severity	integer	Retorna somente os triggers com severidade maior ou igual à severidade fornecida.
evaltype	integer	Método de avaliação das tags. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
tags	array	Retorna somente os triggers com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Uma matriz vazia retorna todos os triggers. Valores possíveis de operador: 0 - *(padrão)* Contém; 1 - É igual a; 2 - Não contém; 3 - Não é igual a; 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 as funções usadas no trigger na propriedade `functions`. Os objetos de função representam as funções usadas na expressão do trigger e têm 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 os triggers dos quais o trigger depende na propriedade `dependencies`.

Parâmetro	Tipo	Descrição
selectDiscoveryData	query	Retorna uma propriedade `discoveryData` com os dados do objeto de descoberta do trigger. O objeto de descoberta do trigger vincula um trigger descoberto a um protótipo de trigger a partir do qual ele foi descoberto. Ele tem as seguintes propriedades: `parent_triggerid` - `(ID)` ID do protótipo de trigger a partir do qual o trigger foi criado; `status` - `(int)` status de descoberta do trigger: 0 - *(padrão)* o trigger foi descoberto, 1 - o trigger não é mais descoberto; `ts_delete` - `(timestamp)` horário em que um trigger que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` horário em que um trigger que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador de que o trigger foi desabilitado por uma regra de LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra de LLD.
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`.
inheritedTags	boolean	Retorna triggers que também têm as `tags` fornecidas no template/host/templates vinculados. Valores possíveis: `true` - o template/host/templates vinculados também devem ter as tags fornecidas; `false` - *(padrão)* as tags do template/host/templates vinculados são ignoradas.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags herdadas de templates, templates vinculados e hosts, e de items referenciados nas expressões do trigger ou de recuperação.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o trigger pertence na propriedade `templategroups`.
filter	object	Retorna somente os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto cujas chaves são nomes de propriedades e cujos valores são um único valor ou uma matriz de valores a serem comparados. Não oferece suporte a propriedades do tipo de dados `text`. Oferece suporte às seguintes 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 subseleções. Aplica-se às seguintes subseleções: `selectHosts` - os resultados serão ordenados por `host`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Esses parâmetros estão descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	

Parâmetro	Tipo	Descrição
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Retorna o objeto de descoberta do trigger na propriedade `triggerDiscovery`. Os objetos de descoberta do trigger vinculam o trigger a um protótipo de trigger a partir do qual ele foi criado. Esta consulta está **obsoleta**; 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",
                    "triggerid": "14062",
                    "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\")}",
                    "automatic": "1"
                },
                {
                    "tag": "error",
                    "value": "",
                    "automatic": "1"
                }
            ]
        }
    ],
    "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 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 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 das quais o trigger depende. Os triggers devem ter apenas a propriedade triggerid definida.
tags	array	Tags do trigger.

Valores de retorno

(object) Retorna um objeto contendo os IDs 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 - cria novos protótipos de trigger

- triggerprototype.delete - exclui protótipos de trigger
- triggerprototype.get - recupera protótipos de trigger
- triggerprototype.update - atualiza protótipos de trigger

Objeto de protótipo de trigger

Os seguintes objetos estão diretamente relacionados à API `triggerprototype`.

Protótipo de trigger

O objeto protótipo de trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
triggerid	ID	ID do protótipo de trigger. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
description	string	Nome do protótipo de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
expression	string	Expressão do trigger. Deve conter pelo menos um protótipo de item. Protótipos de item com `value_type` definido como 5 (binário) ou 6 (JSON) não podem ser usados em expressões de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
event_name	string	Nome do evento gerado pelo trigger.
opdata	string	Dados operacionais.
comments	string	Comentários adicionais ao protótipo de trigger.
priority	integer	Severidade do protótipo de trigger. Valores possíveis: 0 - *(padrão)* não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 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*

Propriedade	Tipo	Descrição
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 de recuperação do trigger. Deve conter pelo menos um protótipo de item. Protótipos de item com `value_type` definido como 5 (binário) ou 6 (JSON) não podem ser usados em expressões de recuperação de trigger.
correlation_mode	integer	Evento OK fecha. Valores possíveis: 0 - *(padrão)* Todos os problemas; 1 - Todos os problemas se os valores das tags coincidirem.
correlation_tag	string	Tag para correspondência.
manual_close	integer	Permitir fechamento manual. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim.
discover	integer	Status de descoberta do protótipo de trigger. Valores possíveis: 0 - *(padrão)* novos triggers serão descobertos; 1 - novos triggers não serão descobertos e triggers existentes serão marcados como perdidos.
uuid	string	Identificador único universal, usado para vincular protótipos de trigger importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de trigger pertencer a um template

Tag de protótipo de trigger

O objeto de tag de protótipo de trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do protótipo de trigger. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do protótipo de trigger. Comportamento da propriedade: - *somente leitura*.

Propriedade	Tipo	Descrição
object	integer	Tipo de objeto do qual a tag foi herdada. Valores possíveis: 0 - template; 1 - host; 3 - item; 4 - protótipo de item. Comportamento da propriedade: - *somente leitura*.
objectid	ID	ID do objeto do qual a tag foi herdada. Comportamento da propriedade: - *somente leitura*.

triggerprototype.create

Descrição

`object triggerprototype.create(object/array triggerPrototypes)`

Este método permite criar novos protótipos de trigger.

Note:
Este método está disponível apenas para tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte User roles para obter mais informações.

Parâmetros

`(object/array)` Protótipos de trigger a serem criados.

Além das propriedades padrão do protótipo de trigger, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers e protótipos de trigger dos quais o protótipo de trigger depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags de protótipo de trigger.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de trigger criados na propriedade `triggerids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de 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 prototypes de trigger.

> **Note:**
> Este método está disponível apenas para 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 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 informados.

> **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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
active	flag	Retorna apenas protótipos de trigger habilitados que pertencem a hosts monitorados.
discoveryids	ID/array	Retorna apenas protótipos de trigger que pertencem às regras LLD fornecidas.
functions	string/array	Retorna apenas triggers que usam as funções fornecidas. Consulte a página Funções suportadas para obter uma lista de funções suportadas.
group	string	Retorna apenas protótipos de trigger que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates com o nome fornecido.
groupids	ID/array	Retorna apenas protótipos de trigger que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates fornecidos.
host	string	Retorna apenas protótipos de trigger que pertencem a hosts com o nome fornecido.
hostids	ID/array	Retorna apenas protótipos de trigger que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas protótipos de trigger herdados de um template.
maintenance	boolean	Se definido como `true`, retorna apenas protótipos de trigger habilitados que pertencem a hosts em manutenção.
min_severity	integer	Retorna apenas protótipos de trigger com severidade maior ou igual à severidade fornecida.
monitored	flag	Retorna apenas protótipos de trigger habilitados que pertencem a hosts monitorados e contêm apenas items habilitados.
templated	boolean	Se definido como `true`, retorna apenas protótipos de trigger que pertencem a templates.
templateids	ID/array	Retorna apenas protótipos de trigger que pertencem aos templates fornecidos.
triggerids	ID/array	Retorna apenas protótipos de trigger com os IDs fornecidos.
expandExpression	flag	Expande funções e macros na expressão do trigger.
selectDependencies	query	Retorna protótipos de trigger e triggers dos quais o protótipo de trigger depende na propriedade `dependencies`.

Parâmetro	Tipo	Descrição
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 a partir 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 de 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`.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com tags herdadas de templates, templates vinculados e hosts, e de protótipos de item referenciados em expressões de trigger ou de recuperação.
selectItems	query	Retorna items e protótipos de item usados no protótipo de trigger na propriedade `items`.
selectTags	query	Retorna as tags do protótipo de trigger na propriedade `tags`.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o protótipo de trigger pertence na propriedade `templategroups`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para corresponder. 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`.

Parâmetro	Tipo	Descrição
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

Recupere todos os protótipos de trigger e suas funções de uma regra LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
```

```
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


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 trigger prototypes existentes.

> **Note:**
> Este método está disponível apenas para 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 trigger prototype a serem atualizadas.

A propriedade `triggerid` deve ser definida para cada trigger prototype, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do trigger prototype, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers e trigger prototypes dos quais o trigger prototype depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags do trigger prototype.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de 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 de usuário tem as seguintes propriedades.

Property	Type	Description
userid	ID	ID do usuário. Comportamento da propriedade: - *somente leitura* - *obrigatória* para operações de atualização
username	string	Nome do usuário. Comportamento da propriedade: - *obrigatória* para operações de criação - *somente leitura* para usuários provisionados se o usuário estiver vinculado a um diretório de usuários (`userdirectoryid` estiver definido com um valor válido diferente de ”0”), o provisionamento do diretório de usuários estiver habilitado (`provision_status` do objeto User directory estiver definido como ”1”) e o status de autenticação de todo o provisionamento LDAP ou SAML estiver habilitado (`ldap_jit_status` do objeto Authentication estiver definido como ”Enabled for configured LDAP IdPs” ou `saml_jit_status` do objeto Authentication estiver definido como ”Enabled for configured SAML IdPs”)

Property	Type	Description
passwd	string	Senha do usuário. O valor deste parâmetro pode ser uma string vazia se o usuário estiver vinculado a um diretório de usuários. Comportamento da propriedade: - *somente gravação*
roleid	ID	ID da função do usuário. Observe que usuários sem uma função podem fazer login no Zabbix somente usando a autenticação LDAP ou SAML, desde que as informações LDAP/SAML correspondam aos mapeamentos de grupos de usuários configurados no Zabbix.
attempt_clock	timestamp	Hora da última tentativa de login malsucedida. Comportamento da propriedade: - *somente leitura*
attempt_failed	integer	Contagem recente de tentativas de login malsucedidas. Comportamento da propriedade: - *somente leitura*
attempt_ip	string	Endereço IP de onde veio a última tentativa de login malsucedida. Comportamento da propriedade: - *somente leitura*
autologin	integer	Indica se o login automático deve ser habilitado. Valores possíveis: 0 - *(padrão)* login automático desabilitado; 1 - login automático habilitado.
autologout	string	Tempo de vida da sessão do usuário. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 2h). Se definido como 0s, a sessão nunca expirará. Padrão: 15m.
default_maintenance_period	string	Período de manutenção padrão para uma manutenção criada por meio de um atalho. Aceita segundos ou uma unidade de tempo com sufixo (300s, 5m, 1h, 1d, 1w). O valor fornecido será arredondado para baixo para minutos inteiros. Padrão: 1h.
lang	string	Código do idioma do usuário, por exemplo, `en_US`. Padrão: `default` - padrão do sistema.
name	string	Nome do usuário.
provisioned	integer	Indica se o usuário foi provisionado. Valores possíveis: 0 - não provisionado; 1 - provisionado. Comportamento da propriedade: - *somente leitura*
refresh	string	Período de atualização automática. Aceita um número de segundos (por exemplo, 1, 60, 300) ou um valor com um sufixo de tempo (por exemplo, 30s, 1m, 1h). Padrão: 30s.
rows_per_page	integer	Quantidade de linhas de objetos a serem exibidas por página. Padrão: 50.

Property	Type	Description
surname	string	Sobrenome do usuário.
theme	string	Tema do usuário. Valores possíveis: `default` - *(padrão)* padrão do sistema; `blue-theme` - Azul; `blue-classic-theme` - Azul (clássico); `dark-theme` - Escuro; `dark-blue-theme` - Azul escuro; `dark-classic-theme` - Escuro (clássico); `hc-light` - Claro de alto contraste; `hc-dark` - Escuro de alto contraste.
ts_provisioned	timestamp	Hora em que a operação de provisionamento mais recente foi realizada. Comportamento da propriedade: - *somente leitura* - *compatível* com usuários do tipo *Super admin*
url	string	URL da página para a qual o usuário será redirecionado após fazer 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* - *compatível* com usuários do tipo *Super admin*
timezone	string	Fuso horário do usuário, por exemplo, `Europe/London`, `UTC`. Padrão: `default` - padrão do sistema. Para obter a lista completa de fusos horários compatíveis, consulte a documentação do PHP.

Mídia

O objeto de mídia tem as seguintes propriedades.

Propriedade	Tipo	Descrição
mediaid	ID	ID da mídia do usuário. Comportamento da propriedade: - *somente leitura*
mediatypeid	ID	ID do tipo de mídia usado pela mídia do usuário. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
sendto	string/array	Endereço, nome de usuário ou outro identificador do destinatário. Se `type` de Tipo de mídia estiver definido como ”Email” ou ”Push”, os valores serão representados como uma matriz. Para ”Push”, a matriz pode conter um curinga (*) ou um ou mais IDs de dispositivos móveis. Se `type` de Tipo de mídia estiver definido como ”Script”, ”SMS” ou ”Webhook”, o valor será representado como uma string. Comportamento da propriedade: - *obrigatório*
active	integer	Indica se a mídia está habilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
severity	integer	Severidades de trigger sobre as quais enviar notificações. Valores de bitmap possíveis: 1 - Não classificado; 2 - Informação; 4 - Aviso; 8 - Média; 16 - Alta; 32 - Desastre. Este é um campo de máscara de bits; qualquer soma dos valores de bitmap possíveis é aceita (por exemplo, 48 para Média, Alta 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	Indica se o usuário foi provisionado. Valores possíveis: 0 - não provisionado; 1 - provisionado.
userdirectory_mediaid	ID	ID do mapeamento de mídia do diretório de usuários para a 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.

Parameter	Type	Description
extend	boolean	Se deseja 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: - *supported* se `sessionid` estiver definido
sessionid	string	Token de autenticação do usuário. Comportamento do parâmetro: - *required* se `token` não estiver definido
token	string	Token de API do usuário. Comportamento do parâmetro: - *required* 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",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "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 da função de usuário. Consulte Funções de usuário para obter mais informações.

Note:
A força da senha do usuário é validada de acordo com as regras da política de senhas definidas pela API de autenticação. Consulte objeto de autenticação para obter mais informações.

Parâmetros

`(object/array)` Usuários a serem criados.

Além das propriedades padrão do usuário, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
usrgrps	array	Grupos de usuários aos quais o usuário será adicionado. Os grupos de usuários devem ter apenas a propriedade `usrgrpid` definida.
medias	array	Mídias do usuário a serem criadas.

Valores de retorno

(object) Retorna um objeto contendo os IDs dos usuários criados na propriedade userids. A ordem dos IDs retornados corresponde à ordem dos usuários passados.

Exemplos

Criando um usuário

Crie um novo usuário, adicione-o a um grupo de usuários e crie uma nova mídia para ele.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

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 de usuário. Consulte User roles para obter 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 informados.

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 User roles para obter 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. Possíveis propriedades do objeto User 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 User 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 pesquisa `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Possíveis propriedades do objeto User 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	Estes 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.

Solicitação:

```
{
    "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",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
```

```
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

Recuperando usuários como *Admin*

Como usuário do tipo *Admin*, recupere dados detalhados sobre seu próprio usuário e dados limitados sobre os usuários do seu grupo de usuários.

Solicitação:

```
{
    "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",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
```

```
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "3",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "0",
    "usrgrps": [
      {
        "usrgrpid": "7",
        "name": "Administradores do Zabbix",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "13",
        "name": "Interno",
        "gui_access": "1",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Função de superadministrador",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
```

```
      "roleid": "4",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "1",
      "usrgrps": [
        {
          "usrgrpid": "8",
          "name": "Convidados",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Desabilitados",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Interno",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Função de convidado",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Recuperando dados do usuário

Recupere os dados de um usuário com o ID ”2”.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "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 que você faça login na API e gere 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 sessão abertos.

Attention:
Este método está disponível apenas para usuários não autenticados que não pertencem a nenhum user group com autenticação multifator habilitada.

Parâmetros

`(object)` Parâmetros contendo o nome de usuário e a senha.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
password	string	Senha do usuário. Comportamento do parâmetro: - *obrigatório*
username	string	Nome do usuário. Comportamento do parâmetro: - *obrigatório*
userData	flag	Retorna informações sobre o usuário autenticado.

Valores de retorno

`(string/object)` Se o parâmetro `userData` for usado, retorna um objeto contendo informações sobre o usuário autenticado.

Além das propriedades padrão do usuário, as seguintes informações são retornadas:

Propriedade	Tipo	Descrição
auth_type	integer	Autenticação padrão para o usuário. Consulte a propriedade `authentication_type` do objeto Authentication para uma lista de valores possíveis.
debug_mode	integer	Se o modo de depuração está ativado ou desativado para o usuário. Consulte a propriedade `debug_mode` do objeto User group para uma lista de valores possíveis.
deprovisioned	boolean	Se o usuário pertence a um grupo de usuários desprovisionados.
gui_access	string	Método de autenticação do usuário no frontend. Consulte a propriedade `gui_access` do objeto User group para uma lista de valores possíveis.
mfaid	integer	ID do método MFA a ser usado para o usuário durante o login. Retorna "0" se o MFA estiver desativado globalmente ou para todos os grupos de usuários aos quais o usuário pertence.
secret	string	String aleatória de 32 caracteres. É gerada no login do usuário.
sessionid	string	Token de autenticação, que deve ser usado nas próximas requisições da API.
type	integer	Tipo de usuário. Consulte a propriedade `type` do objeto Role para uma lista de valores possíveis.
userip	string	Endereço IP do usuário.

Note:
Se um usuário foi autenticado com sucesso após uma ou mais tentativas falhas, o método retornará os valores atuais das propriedades `attempt_clock`, `attempt_failed` e `attempt_ip` e então 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 que você saia 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. Consulte Funções de usuário para obter 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 User roles para obter mais informações.

Parâmetros

(array) IDs dos usuários a serem provisionados.

Valores de retorno

(object) Retorna um objeto contendo os IDs dos usuários provisionados na propriedade userids.

Exemplos

Provisionando múltiplos usuários

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 de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

(array) IDs dos usuários para os quais redefinir os segredos TOTP.

> **Note:**
> As sessões de usuário para os usuários especificados também serão excluídas (exceto para o usuário que 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. Consulte Funções de usuário para obter 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 de usuário. Consulte User roles para obter mais informações.

Note:
A força da senha do usuário é validada de acordo com as regras de política de senha definidas pela Authentication API. Consulte Authentication API para obter 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 foi projetada para trabalhar com verificações de descoberta.

Referências de objetos:

- Verificação de descoberta

Métodos disponíveis:

- dcheck.get - recuperar verificações de descoberta

Objeto de verificação de descoberta

Os seguintes objetos estão diretamente relacionados à API `dcheck`.

Verificação de descoberta

O objeto de verificação de descoberta define uma verificação específica realizada por uma regra de descoberta de rede. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dcheckid	ID	ID da verificação de descoberta.
druleid	ID	ID da regra de descoberta à qual a verificação pertence.
key_	string	Chave do item (se `type` estiver definido como ”Zabbix agent”) ou OID SNMP (se `type` estiver definido como ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”
ports	string	Um ou vários intervalos de portas a serem verificados, separados por vírgulas. Padrão: 0. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH” (0), ”LDAP” (1), ”SMTP” (2), ”FTP” (3), ”HTTP” (4), ”POP” (5), ”NNTP” (6), ”IMAP” (7), ”TCP” (8), ”Zabbix agent” (9), ”SNMPv1 agent” (10), ”SNMPv2 agent” (11), ”SNMPv3 agent” (13), ”HTTPS” (14) ou ”Telnet” (15)
snmp_community	string	Comunidade SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMPv1 agent” ou ”SNMPv2 agent”
snmpv3_authpassphrase	string	Senha de autenticação. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authNoPriv” ou ”authPriv”
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”

Propriedade	Tipo	Descrição
snmpv3_contextname	string	Nome do contexto SNMPv3. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”
snmpv3_privpassphrase	string	Senha de privacidade. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authPriv”
snmpv3_privprotocol	integer	Protocolo de privacidade. Valores possíveis: 0 - *(padrão)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authPriv”
snmpv3_securitylevel	string	Nível de segurança. Valores possíveis: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”
snmpv3_securityname	string	Nome de segurança. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”
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*

Propriedade	Tipo	Descrição
uniq	integer	Se deve usar esta verificação como critério de exclusividade do dispositivo. Apenas uma verificação exclusiva pode ser configurada para uma regra de descoberta. Valores possíveis: 0 - *(padrão)* não usar esta verificação como critério de exclusividade; 1 - usar esta verificação como critério de exclusividade. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”
host_source	integer	Fonte para o nome do host. Valores possíveis: 1 - *(padrão)* DNS; 2 - IP; 3 - valor de descoberta desta verificação.
name_source	integer	Fonte para o nome visível. Valores possíveis: 0 - *(padrão)* não especificado; 1 - DNS; 2 - IP; 3 - valor de descoberta desta verificação.
allow_redirect	integer	Permitir situação em que o alvo do ICMP ping responde de um endereço IP diferente. Valores possíveis: 0 - *(padrão)* tratar respostas redirecionadas como se o host de destino estivesse inativo (falha); 1 - tratar respostas redirecionadas como se o host de destino estivesse ativo (sucesso). Comportamento da propriedade: - *suportado* se `type` estiver definido como ”ICMP ping”

dcheck.get

Descrição

`integer/array dcheck.get(object parameters)`

O método permite recuperar verificações de descoberta de acordo com os parâmetros informados.

> **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 obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dcheckids	ID/array	Retorna apenas verificações de descoberta com os IDs fornecidos.
druleids	ID/array	Retorna apenas verificações de descoberta que pertencem às regras de descoberta fornecidas.
dserviceids	ID/array	Retorna apenas verificações de descoberta que detectaram os serviços descobertos fornecidos.

Parâmetro	Tipo	Descrição
selectDRules	query	Retorna regras de descoberta relacionadas às verificações de descoberta.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dcheckid`, `druleid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- 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 Assinatura do método

Cada página de método da API contém a assinatura do método.

Exemplo: `object action.create(object/array actions)`

Assinatura	Descrição
`object`	O tipo de dados retornado pelo método.
`action.create`	O nome completo do método.
`object/array`	O tipo de dados aceito pelo método.
`actions`	Um breve resumo do que o método aceita.

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.
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.
editable	boolean	Se definido como `true`, retorna apenas objetos para os quais o usuário tem permissões de escrita. Padrão: `false`.
excludeSearch	boolean	Retorna resultados que não correspondem aos critérios fornecidos no parâmetro `search`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades (por exemplo, propriedades do objeto Host em `host.get`, propriedades do objeto Item em `item.get`, etc.), e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Observe que alguns métodos possuem 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. Observe que o ID do objeto (ou seja, `hostid`, `itemid`, etc.) é sempre incluído na resposta, mesmo que não seja especificado no parâmetro `output`. Padrão: `extend`.
preservekeys	boolean	Usa os IDs como chaves no array resultante.

Parâmetro	Tipo	Descrição
search	object	Retorna resultados que correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades (por exemplo, propriedades do objeto Host em `host.get`, propriedades do objeto Item em `item.get`, etc.), e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma busca `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Observe que alguns métodos possuem funcionalidades específicas para este parâmetro, que são descritas na página do método (por exemplo, o parâmetro `search` em host.get também suporta propriedades da interface do Host).
searchByAny	boolean	Se definido como `true`, retorna resultados que correspondem a qualquer um dos critérios fornecidos nos parâmetros `filter` ou `search` em vez de todos eles. Padrão: `false`.
searchWildcardsEnabled	boolean	Se definido como `true`, habilita o uso de "*" como caractere curinga no parâmetro `search`. Padrão: `false`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Consulte a descrição de um método get específico da API para obter uma lista de propriedades que podem ser usadas para ordenação. Macros não são expandidas antes da ordenação. Se nenhum valor for especificado, os dados serão retornados sem ordenação.
sortorder	string/array	Ordem de ordenação. Se um array for passado, cada valor será associado à propriedade correspondente fornecida no parâmetro `sortfield`. Valores possíveis: `ASC` - *(padrão)* ascendente; `DESC` - descendente.
startSearch	boolean	O parâmetro `search` irá comparar o início dos campos, ou seja, realizará uma busca `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 histórico

ZBXNEXT-10069 Adicionado o novo parâmetro `maxValueSize` ao método history.get. Se você estiver solicitando dados para items com um valor binário (`history=5`), defina `maxValueSize` como `null`; caso contrário, `history.get` retornará o valor do item truncado para 64 KiB.

host

ZBXNEXT-9185 Adicionada a descrição do comportamento da propriedade `monitored_by` do objeto host. ZBXNEXT-10387 `host.massupdate`: Método removido. Use `host.update` em seu lugar.

hostgroup

ZBXNEXT-10387 `hostgroup.massupdate`: Método removido. Use `hostgroup.update`, `hostgroup.massadd` ou `hostgroup.massremov` em seu lugar.

hostinterface

ZBXNEXT-10814 Os métodos `hostinterface.create`, `hostinterface.update`, `hostinterface.delete`, `hostinterface.massadd` e `hostinterface.massremove` foram removidos da API `hostinterface`. Use os métodos do objeto host.

função

ZBXNEXT-9185 Adicionado o novo valor `select_server_for_monitoring` para a propriedade `name` do objeto action da função.

template

ZBXNEXT-10387 `template.massupdate`: Método removido. Use `template.update`, `template.massadd` ou `template_massremove` em vez disso.

grupotemplate

ZBXNEXT-10387 `templategroup.massupdate`: Método removido. Use `templategroup.update`, `templategroup.massadd` ou `templategroup.massremove` em vez disso.

usergroup

ZBXNEXT-9185 Foram adicionados os novos parâmetros `proxies` e `proxy_groups` aos métodos `usergroup.create` e `usergroup.update`.
 Foram adicionados os novos parâmetros `proxyids`, `proxy_groupids`, `selectProxies` e `selectProxyGroups` ao método `usergroup.get`.
 Foram adicionadas as novas propriedades `proxy_mode` e `proxy_group_mode` ao objeto grupo de usuários.

Outras alterações e correções de bugs ação

ZBXNEXT-9399 Adicionado o novo valor 29 (grupo de proxy) ao parâmetro de filtro `conditiontype` action se a origem do evento for descoberta/registro automático.

apmlog

ZBXNEXT-10083 Adicionada uma nova API de log do APM.

apmmetric

ZBXNEXT-10083 Adicionada uma nova API de métrica de APM.

apmspan

ZBXNEXT-10083 Adicionada uma nova API de APM span.

apmtrace

ZBXNEXT-10083 Adicionada uma nova API de rastreamento APM.

configuração

ZBXNEXT-5419 Adicionado um novo parâmetro `returnMissingObjects` ao método configuration.importcompare.
 ZBXNEXT-5419 Adicionado um novo parâmetro `dashboards` ao objeto rules (método `configuration.importcompare`).
 ZBXNEXT-5419 Adicionado um novo parâmetro `returnMissingObjects` ao método configuration.import.
 ZBXNEXT-5419 Adicionado um novo parâmetro `dashboards` ao objeto rules (método `configuration.import`). ZBXNEXT-10390 Adicionado um novo parâmetro `global_regexes` ao objeto `options` do método `configuration.export`. ZBXNEXT-10390 Adicionado um novo parâmetro `global_regexes` ao objeto `rules` dos métodos `configuration.importcompare` e `configuration.import`.

connector

ZBXNEXT-10069 Adicionado novo valor `item_value_type` (64 - JSON) ao objeto connector.

dashboard

ZBXNEXT-4769 Adicionado o novo campo de widget de dashboard `show_hostnames` ao widget graph.
 ZBXNEXT-9546 Adicionados os novos campos de widget de dashboard `ds.0.item_tags_evaltype`, `ds.0.item_tags.0.tag`, `ds.0.item_tags.0.operator` e `ds.0.item_tags.0.value` aos widgets svggraph e piechart.
 ZBXNEXT-10158 Adicionados os novos campos de widget de dashboard `columns.0.aggregate_columns`, `columns.0.combined_column_name` e `columns.0.column_aggregate_function` ao widget topitems.
 ZBXNEXT-9424, ZBXNEXT-4769 Adicionado o novo tipo de widget de dashboard scatterplot.
 ZBXNEXT-8160 Adicionados os novos campos de widget de dashboard `clustering_mode` e `clustering_zoom_level` ao widget geomap.
 ZBXNEXT-943 Adicionado o novo campo de widget de dashboard `ds.0.invert_values` ao widget graph.
 ZBXNEXT-10657 Adicionados valores padrão para o novo tema "dark-blue-theme" aos parâmetros `Color` e `Background color` do widget honeycomb.

drule

ZBXNEXT-9185 Comportamento atualizado da propriedade `timeout` do objeto regra de descoberta.

dispositivo

ZBXNEXT-10357 Adicionada uma nova Device API com os métodos `device.get` e `device.offboard`.

evento

ZBXNEXT-1150 Adicionada a propriedade `maintenanceid` ao parâmetro `selectAcknowledges` do método event.get.

history

ZBXNEXT-10069 Adicionado novo valor `history` (6 - JSON) ao método history.get.
 ZBXNEXT-10069 Adicionado novo objeto `JSON history`.

hostprototype

ZBXNEXT-9546 Adicionado novo parâmetro `selectInheritedTags` ao método hostprototype.get.

httptest

ZBXNEXT-9546 Adicionados novos parâmetros `inheritedTags` e `selectInheritedTags` ao método httptest.get.

item

ZBXNEXT-10249 Adicionado um novo valor de `type` (24 - Consulta de telemetria) ao objeto item.
 ZBXNEXT-10249 Adicionadas as novas propriedades `query`, `time_shift`, `lookback_limit` e `granularity` ao objeto item.
 ZBXNEXT-9185 Atualizado o comportamento da propriedade `timeout` do objeto item.
 ZBXNEXT-10069 Adicionado um novo valor de `value_type` (6 - JSON) ao objeto item.
 ZBXNEXT-9546 Adicionados os novos parâmetros `inheritedTags` e `selectInheritedTags` ao método item.get.

itemprototype

ZBXNEXT-10249 Adicionado um novo valor de `type` (24 - Consulta de telemetria) ao objeto item prototype.
 ZBXNEXT-10249 Adicionadas as novas propriedades `query`, `time_shift`, `lookback_limit` e `granularity` ao objeto item prototype.
 ZBXNEXT-10069 Adicionado um novo valor de `value_type` (6 - JSON) ao objeto itemprototype.
 ZBXNEXT-9546 Adicionado um novo parâmetro `selectInheritedTags` ao método itemprototype.get.

manutenção

ZBXNEXT-8339 Foram adicionadas as propriedades `triggers` e `event_names` aos métodos maintenance.create e maintenance.update. A propriedade `problem_tags` passou a incluir os operadores "Não é igual a" e "Não contém".
 ZBXNEXT-8339 Foram adicionados os parâmetros `triggerids`, `selectTriggers` e `selectEventNames` ao método maintenance.get.

mediatype

ZBXNEXT-10357 Adicionado um novo valor de `type` (5 - Push) ao objeto media type.

problema

ZBXNEXT-1150 Adicionada a propriedade `maintenanceid` ao parâmetro `selectAcknowledges` do método problem.get.

proxy

ZBXNEXT-10084 Adicionado o novo parâmetro `selectApm` ao método proxy.get.
 ZBXNEXT-10084 Adicionado o novo parâmetro `apm` aos métodos proxy.create e proxy.update.
 ZBXNEXT-10084 Adicionado o novo objeto Application performance monitoring (APM).
 ZBXNEXT-10249 Adicionada a nova propriedade `timeout_telemetry_query` ao objeto proxy.

regexp

ZBXNEXT-10390 Adicionada uma nova propriedade `description` ao objeto expressão regular.

função

ZBXNEXT-10469 Alterado o valor padrão da propriedade `api.access` no objeto de regras de função. O valor padrão agora é 0 (desabilitado).

serverinfo

ZBXNEXT-10357 Adicionada uma nova API de informações do server com o método `serverinfo.get`.

configurações

ZBXNEXT-10249 Adicionada a nova propriedade `timeout_telemetry_query` ao objeto settings.
 ZBXNEXT-10680 Adicionada a nova propriedade `apm_global_db` ao objeto settings.
 ZBXNEXT-10357 Adicionada a nova propriedade `device_link_timeout` ao objeto settings.
 ZBXNEXT-10657 Adicionado o valor `dark-blue-theme` à propriedade `default_theme` do objeto settings.

template

ZBXNEXT-9546 Adicionados novos parâmetros `inheritedTags` e `selectInheritedTags` ao método template.get.

templatedashboard

ZBXNEXT-9424 Adicionado novo tipo de widget de dashboard de template scatterplot.

trigger

ZBXNEXT-9546 Adicionados os novos parâmetros `inheritedTags` e `selectInheritedTags` ao método trigger.get.
 ZBXNEXT-8339 Adicionada a nova propriedade `maintenanceids` ao método trigger.get.

triggerprototype

ZBXNEXT-9546 Adicionado novo parâmetro `selectInheritedTags` ao método triggerprototype.get.

usuário

ZBXNEXT-10657 Adicionado o valor `dark-blue-theme` à propriedade `theme` do objeto user.
 ZBXNEXT-8339 Adicionada a nova propriedade `default_maintenance_period` aos métodos user.create, user.get e user.update.

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.
 DEV-5022 Alterado o valor padrão das propriedades `sign_assertions`, `sign_authn_requests`, `sign_messages`, `sign_logout_requests` e `sign_logout_responses` no objeto user directory. O valor padrão agora é 1 (habilitado).

8.0.1 Esta versão ainda não foi lançada.

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 server ou proxy do Zabbix, por exemplo, dados periódicos de disponibilidade e desempenho no caso de scripts de usuário de longa execução. O envio de dados para o server ou proxy do Zabbix é possível usando a ferramenta Zabbix sender ou o protocolo do Zabbix sender. O envio de dados para o server do 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.

Itens do Zabbix agent `system.run[]`

`system.run[]` é um item do Zabbix agent 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 Zabbix agent está em execução.

> **Note:**
> Os itens `system.run[]` são desabilitados por padrão e, se forem usados, devem ser habilitados (permitidos) e definidos no arquivo de configuração do Zabbix agent ou do Zabbix agent 2 (parâmetro de configuração `AllowKey` ou `AllowKeyRegexp`).

> **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 do HTTP agent

O item do HTTP agent é um item para executar solicitações de dados por HTTP/HTTPS. Os itens do HTTP agent são úteis para enviar solicitações a endpoints HTTP para recuperar dados de serviços como *Elasticsearch* e *OpenWeatherMap*, verificar o status da API do Zabbix ou o status do web server Apache ou NGINX etc. Os itens do HTTP agent (com o 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 estendem as capacidades de monitoramento do Zabbix agent 2. Escritos na linguagem de programação Go, eles fornecem uma alternativa aos módulos carregáveis (escritos em C).

É possível desenvolver seu próprio plugin (coletor de dados independente) para o Zabbix agent 2. Veja Developer center.

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.

Módulos carregáveis

Visão geral

Módulos carregáveis oferecem uma opção orientada ao desempenho para estender a funcionalidade do Zabbix.

Você pode estender a funcionalidade do Zabbix de várias maneiras, por exemplo, com parâmetros de usuário, verificações externas e itens do agent do Zabbix system.run. Eles funcionam muito bem, mas têm uma grande desvantagem, que é o fork(). O Zabbix precisa criar um novo processo toda vez que lida com uma métrica de usuário, o que não é bom para o desempenho. Normalmente não é um grande problema, no entanto, pode ser uma questão séria ao monitorar sistemas embarcados, ter um grande número de parâmetros monitorados ou scripts pesados com lógica complexa ou longo tempo de inicialização.

O suporte a módulos carregáveis oferece maneiras de estender o agent, server e proxy do Zabbix sem sacrificar o desempenho.

Um módulo carregável é basicamente uma biblioteca compartilhada usada pelo daemon do Zabbix e carregada na inicialização. A biblioteca deve conter certas funções, para que um processo do Zabbix possa detectar que o arquivo é realmente um módulo que pode carregar e trabalhar.

Módulos carregáveis têm vários benefícios. Ótimo desempenho e capacidade de implementar qualquer lógica são muito importantes, mas talvez a vantagem mais importante seja a capacidade de desenvolver, usar e compartilhar módulos do Zabbix. Isso contribui para uma manutenção sem problemas e ajuda a entregar novas funcionalidades de forma mais fácil e independente do código base principal do Zabbix.

O licenciamento do módulo e a distribuição em forma binária são regidos pela licença AGPL-3.0 (os módulos são vinculados ao Zabbix em tempo de execução e usam os cabeçalhos do Zabbix; todo o código do Zabbix é licenciado sob AGPL-3.0 desde o Zabbix 7.0). A compatibilidade binária não é garantida pelo Zabbix.

A estabilidade da API do módulo é garantida durante um ciclo de release LTS (Long Term Support) do Zabbix. A estabilidade da API do Zabbix não é garantida (tecnicamente é possível chamar funções internas do Zabbix a partir de um módulo, mas não há garantia de que tais módulos funcionarão).

API do módulo

Para que uma biblioteca compartilhada seja tratada como um módulo Zabbix, ela deve implementar e exportar várias funções. Atualmente, existem seis funções na API do módulo Zabbix, das quais apenas uma é obrigatória e as outras cinco são opcionais.

Interface obrigatória

A única função obrigatória é `zbx_module_api_version()`:

```
int zbx_module_api_version(void);
```

Essa função deve retornar a versão da API implementada por este módulo e, para que o módulo seja carregado, essa 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, essa função deve retornar essa constante. A constante antiga `ZBX_MODULE_API_VERSION_ONE`, usada para esse propósito, agora está definida como igual a `ZBX_MODULE_API_VERSION` para preservar a compatibilidade com o 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_wr` e `zbx_module_uninit()`:

```
int zbx_module_init(void);
```

Esta função deve executar 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 encerrada 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 as funções de callback que o Zabbix server 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 tiver interesse no histórico de um determinado tipo.

```
int zbx_module_uninit(void);
```

Esta função deve executar a desinicialização necessária (se houver), como liberar recursos alocados, fechar descritores de arquivo 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 encerramento 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 de o item aceitar parâmetros ou não), **function** é uma função C que implementa o item (por exemplo, `zbx_module_dummy_random`), e **test_param** é a lista de parâmetros a ser usada quando o agent Zabbix é iniciado com 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 ponteiro, 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 tiver sido obtido com sucesso. Caso contrário, devem retornar SYSINFO_RET_FAIL. Consulte o exemplo do módulo "dummy" abaixo para obter detalhes sobre como obter informações de AGENT_REQUEST e como definir informações em AGENT_RESULT.

Fornecendo callbacks de exportação de histórico

Attention:
A exportação de histórico via módulo não é mais suportada pelo Zabbix proxy.

O módulo pode especificar funções para exportar dados de histórico por tipo: Numeric (float), Numeric (unsigned), Character, Text 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 com history_num elementos como argumento. 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 cache de valores.

> **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

Os módulos atualmente devem ser compilados dentro da árvore de código-fonte do Zabbix, porque a API do módulo depende de algumas estruturas de dados que são definidas nos cabeçalhos do Zabbix.

O cabeçalho mais importante para módulos carregáveis é `include/module.h`, que define essas estruturas de dados. Outros cabeçalhos 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 que o módulo seja 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 detalhes.

Outro cabeçalho útil é `include/zbxcommon.h`, que define a função `zabbix_log()`, que pode ser usada para fins de registro e depuração.

Parâmetros de configuração

O agent, o server e o proxy do Zabbix oferecem suporte a dois parâmetros para trabalhar 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 /), `LoadModulePath` será ignorado. É permitido usar vários parâmetros `LoadModule`.

Por exemplo, para estender o agent do Zabbix, podemos 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
```

Na inicialização do agent, ele carregará os módulos mariadb.so, apache.so e kernel.so do diretório `/usr/local/lib/zabbix/agent`, enquanto dummy.so será carregado de `/usr/local/lib/zabbix`. O agent não será iniciado se um módulo estiver ausente, em

caso de permissões incorretas ou se uma biblioteca compartilhada não for um módulo do Zabbix.

Configuração do frontend

Módulos carregáveis são suportados pelo Zabbix agent, server e proxy. Portanto, o tipo de item no frontend do Zabbix depende de onde o módulo é carregado. Se o módulo for carregado no agent, então o tipo de item deve ser "Zabbix agent" ou "Zabbix agent (active)". Se o módulo for carregado no server ou proxy, então o tipo de item deve ser "Simple check".

A exportação de histórico por meio de módulos do Zabbix não requer nenhuma configuração do 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 não NULL, então a exportação de histórico será habilitada automaticamente.

Módulo dummy

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 template para seus próprios módulos.

Depois que `./configure` tiver sido executado na raiz da árvore de código-fonte do Zabbix, conforme descrito acima, basta executar `make` para compilar `dummy.so`.

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* a variável mantém a configuração de timeout para o processamento de item */
static int  item_timeout = 0;

/* o módulo DEVE definir funções internas como static e usar um padrão de nomenclatura diferente dos símbo
/* do Zabbix (zbx_*) e das funções da API de módulos carregáveis (zbx_module_*) para evitar conflitos
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
```

```
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: retorna o número da versão da interface do módulo                 *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - versão de module.h com a 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;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: define o valor de timeout para o processamento de items           *
 *                                                                            *
 * Parameters: timeout - timeout em segundos, 0 - nenhum timeout definido     *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: retorna a lista de chaves de item suportadas pelo módulo          *
 *                                                                            *
 * Return value: 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("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: ponto de entrada principal para o processamento de um item        *
 *                                                                            *
 * Parameters: 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 da chave vazio) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (string entre aspas ou sem aspas) *
 *                                                                            *
 *             result - estrutura que conterá o resultado                     *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - a função falhou, o item será marcado      *
 *                                  como não suportado pelo zabbix            *
 *               SYSINFO_RET_OK - sucesso                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) pode ser usado para obter um ponteiro    *
 *          para o N-ésimo parâmetro, começando em 0 (primeiro parâmetro).    *
 *          Certifique-se de que ele exista verificando o valor de request->nparam.
 *          Da mesma forma, get_rparam_type(request, N-1) pode ser usado para  *
 *          obter o tipo de um 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("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* não há validação rigorosa de parâmetros e tipos por simplicidade */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
```

```
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: a função é chamada na inicialização do agent                      *
 *          Ela deve ser usada para chamar quaisquer rotinas de inicialização *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - sucesso                                      *
 *               ZBX_MODULE_FAIL - inicialização do módulo falhou             *
 *                                                                            *
 * Comment: 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;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: a função é chamada no encerramento do agent                       *
 *          Ela deve ser usada para limpar recursos utilizados, se houver     *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - sucesso                                      *
 *               ZBX_MODULE_FAIL - função falhou                              *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: funções de callback para armazenar dados históricos dos tipos     *
 *          float, integer, string, text e log, respectivamente, em           *
 *          armazenamento externo                                             *
 *                                                                            *
 * Parameters: 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)
{
```

```c
    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_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, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: retorna um conjunto de funções do módulo que o Zabbix chamará    *
 *          para exportar diferentes tipos de dados históricos                *
 *                                                                            *
 * Return value: estrutura com ponteiros para funções de callback (pode ser   *
 *               NULL se o módulo não tiver interesse em determinados tipos   *
 *               de dados)                                                    *
 *                                                                            *
 ******************************************************************************/
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 exatamente como foi recebido; por exemplo, `dummy.echo[ABC]` retornará ABC
- `dummy.random[param1, param2]` - retorna um número aleatório dentro do intervalo de param1-param2; por exemplo, `dummy.random[1,1000000]`

Limitações

O suporte a módulos carregáveis é implementado apenas para a plataforma Unix. Isso significa que ele não funciona para agents do Windows.

Em alguns casos, um módulo pode precisar ler parâmetros de configuração relacionados ao módulo a partir de `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.

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 causar falha no frontend. Se isso acontecer, basta remover o código do módulo do frontend. Assim que você recarregar o frontend do Zabbix, verá uma observação informando que alguns módulos estão ausentes. Acesse Administração de módulos (em *Administração* > *Geral* > *Módulos*) e clique em *Scan directory* novamente para remover do banco de dados os módulos inexistentes.

Instalação

Leia sempre o manual de instalação de um módulo específico. Recomenda-se instalar novos módulos um por um para identificar falhas com facilidade.

Logo antes de instalar um módulo:

- Certifique-se de ter baixado 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 única versão pode ser habilitada por vez.

Etapas para instalar um módulo:

- Descompacte o módulo em sua própria pasta no diretório `modules` da instalação do frontend do Zabbix (por exemplo, `zabbix/ui/modules`).
- Verifique se a pasta do módulo contém pelo menos o arquivo `manifest.json`.
- Acesse Administração de módulos e clique no botão *Scan directory*.
- 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, verifique se `manifest.json` existe em `modules/seu-modulo/`. Se não existir, talvez você tenha 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 permissão de leitura e busca (`r-x`) para a pasta do módulo e todas as subpastas, e permissão de leitura (`r--`) para todos os arquivos dentro dela.
O frontend travou	O código do módulo não é compatível com a versão atual do Zabbix ou com a configuração do server. Exclua os arquivos do módulo e recarregue o frontend. Você verá um aviso de que alguns módulos estão ausentes. Acesse Administração de módulos e clique em *Scan directory* novamente para remover do banco de dados os módulos inexistentes.
Aparece uma mensagem de erro sobre namespace, ID ou actions idênticos	O novo módulo tentou registrar um namespace, ID ou actions que já estão registrados por outros módulos habilitados. Desabilite o módulo em conflito (mencionado na mensagem de erro) antes de habilitar o novo.
Aparecem mensagens de erro técnicas	Reporte os erros ao desenvolvedor do módulo.

Desenvolvendo módulos

Para obter informações sobre o desenvolvimento de módulos personalizados, consulte o Centro do desenvolvedor.

Apêndices

Visão geral Esta seção contém materiais de referência, parâmetros de configuração, especificações técnicas e guias de procedimentos que complementam o manual principal.

Referência essencial

- Instalação e configuração - criação do banco de dados, instalação e configuração do server e do agent.
- Configuração de processos - parâmetros de configuração para Zabbix server, proxy, agent e mais.
- Items - referência de chaves de item e tipos de dados suportados.
- Macros - sintaxe e uso de macros internas e de usuário.

Instalação e configuração

Esta seção aborda a criação do banco de dados, a configuração de segurança, os procedimentos de configuração específicos da plataforma e a configuração de recursos.

Use a barra lateral para encontrar o procedimento relevante para o seu ambiente.

Criação do banco de dados

Visão geral

Um banco de dados do Zabbix deve ser criado durante a instalação do server ou proxy do Zabbix.

Esta seção fornece instruções para criar um banco de dados do 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 do 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. É conhecido por funcionar sem falhas de segurança. Os usuários devem estar cientes de que há problemas de segurança conhecidos ao usar algumas das outras codificações. Para mudar para UTF-8, consulte Reparando o conjunto de caracteres e a collation do banco de dados do 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

O conjunto de caracteres utf8mb4 com a ordenação utf8mb4_bin é compatível com o server/proxy do Zabbix para funcionar corretamente com bancos de dados MySQL/MariaDB.
 Para evitar problemas futuros, é altamente recomendável usar o conjunto de caracteres utf8mb4 com a ordenação utf8mb4_bin tanto em instalações novas quanto existentes. Para converter bancos de dados que usam o conjunto de caracteres utf8mb3 e a ordenação utf8mb3_bin para o conjunto de caracteres utf8mb4 e a ordenação utf8mb4_bin, siga as instruções em Reparando o conjunto de caracteres e a ordenação do banco de dados do Zabbix.

Os triggers determinísticos precisam ser criados durante a importação do schema. No MySQL e no MariaDB, isso exige que `GLOBAL log_bin_trust_function_creators = 1` esteja definido se o registro binário estiver habilitado, 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 o Zabbix a partir de pacotes ou containers, consulte as instruções correspondentes.

Se você estiver instalando o Zabbix a partir do código-fonte:

- Crie e configure um banco de dados e um usuário. Por motivos de segurança, use uma senha forte e exclusiva para o usuário do banco de dados do Zabbix.

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- Importe os dados para o banco de dados e defina o conjunto de caracteres utf8mb4 como padrão (supondo que você esteja no diretório raiz do código-fonte do Zabbix). Para um banco de dados de proxy do Zabbix, somente `schema.sql` deve ser importado (não importe images.sql nem data.sql).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### pare aqui se estiver criando um banco de dados para o proxy do Zabbix
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` pode ser desabilitado depois que o schema tiver sido importado com sucesso:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Você precisa ter um usuário de banco de dados com permissões para criar objetos de banco de dados.

Se você estiver instalando o Zabbix a partir de pacotes ou containers, consulte as instruções correspondentes.

Se você estiver instalando o Zabbix a partir do código-fonte:

- Crie um usuário de banco de dados.

O seguinte comando de shell criará o usuário `zabbix`. Especifique uma senha quando solicitado e repita a senha (observe que talvez você seja solicitado primeiro a informar a senha do `sudo`):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Crie um banco de dados.

O seguinte comando de shell criará o banco de dados `zabbix` (último parâmetro) com o usuário criado anteriormente como proprietário (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importe o schema e os dados iniciais (assumindo que você esteja no diretório raiz dos fontes do Zabbix). Para um banco de dados de proxy do Zabbix, somente `schema.sql` deve ser importado (não `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.

Elasticsearch e ClickHouse

O Zabbix também pode armazenar dados de histórico em Elasticsearch ou ClickHouse. Para instruções de configuração, consulte as páginas vinculadas.

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.

Reparando o conjunto de caracteres e a collation do banco de dados do Zabbix

MySQL/MariaDB

Historicamente, MySQL e seus derivados usavam 'utf8' como um alias para utf8mb3 - a implementação própria de 3 bytes do padrão UTF8, que é de 4 bytes. A partir do MySQL 8.0.28 e do 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 a 'utf8mb4'. Desde o Zabbix 6.0, 'utf8mb4' é suportado. Para evitar problemas futuros, é altamente recomendável usar 'utf8mb4'. Outra vantagem de mudar para 'utf8mb4' é o suporte a caracteres Unicode suplementares.

Warning:
Como as versões anteriores ao Zabbix 6.0 não conhecem utf8mb4, certifique-se de primeiro atualizar o server Zabbix e o esquema do banco de dados para 6.0.x ou posterior antes de executar a conversão para utf8mb4.

1. Verifique o conjunto de caracteres e a collation do banco de dados.

Por exemplo:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2_general_ci    |
+--------------------------+----------------------+
```

Ou:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Como vemos, o conjunto de caracteres aqui não é 'utf8mb4' e a collation não é 'utf8mb4_bin', então precisamos corrigi-los.

2. Pare o Zabbix.

3. Crie uma cópia de backup do banco de dados!

4. Corrija o conjunto de caracteres e a collation no nível do banco de dados:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Valores corrigidos:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Carregue o script para corrigir o conjunto de caracteres e a collation no nível de tabela e coluna:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Execute o script:

```
               SET @ZABBIX_DATABASE = '<your DB name>';
If MariaDB →   set innodb_strict_mode = OFF;
               CALL zbx_convert_utf8();
If MariaDB →   set innodb_strict_mode = ON;
               drop procedure zbx_convert_utf8;
```

Observe que 'utf8mb4' deve consumir um pouco mais de espaço em disco.

7. Se não houver erros, talvez você queira criar uma cópia de backup do banco de dados com o banco de dados corrigido.

8. Inicie o Zabbix.

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 consultas e economizar espaço. Isso também melhora o gerenciamento e a sincronização de dados em configurações em cluster, ajudando na escalabilidade e garantindo que o sistema permaneça confiável mesmo se alguns servers falharem.

> **Attention:**
> As instruções fornecidas nesta página foram elaboradas para usuários avançados e podem precisar ser ajustadas para sua configuração específica. A atualização para chaves primárias pode ser demorada e consumir muitos recursos. Certifique-se de que haja espaço livre em disco suficiente; dependendo do tamanho do banco de dados e dos dados armazenados, o processo pode exigir até 2,5 vezes o espaço atualmente usado pelas tabelas de histórico.

As chaves primárias são usadas em 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** atualizar o Zabbix server para 6.0 ou mais recente.

Attention:
Desde o Zabbix 7.0, a atualização de tabelas para chaves primárias também atualiza as tabelas para usar tipos de dados de dupla precisão.

 Se você tiver o Zabbix 7.0 (ou posterior), as tabelas já usam dupla precisão. No entanto, as instruções desta página ainda podem ser seguidas para atualizar as tabelas para chaves primárias sem afetar as tabelas que já usam dupla precisão.

 Se você tiver o Zabbix 6.4 (ou anterior), considere atualizar as tabelas para dupla precisão primeiro. Para mais informações, consulte Atualização para valores numéricos de faixa estendida na documentação do Zabbix 7.0.

As instruções estão disponíveis para:

- MariaDB/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.

MariaDB/MySQL

A exportação e a importação devem ser realizadas no 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 server Zabbix em execução, mas é recomendável parar o server durante o processo de atualização. O MySQL Shell (*mysqlsh*) deve estar instalado e conseguir se conectar ao DB.

- Faça login no console do MySQL como root (recomendado) ou como qualquer usuário com privilégios 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<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Exporte e importe os dados.

Conecte-se via mysqlsh. Se estiver usando uma conexão por socket, pode ser necessário especificar o caminho.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Alterne 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 tem suporte a JS. Nesse caso, instale o pacote oficial MySQL Shell package da Oracle (ou compile-o 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 upgrade leva mais tempo e deve ser usado somente se um upgrade com *mysqlsh* não for possível.

Upgrade das tabelas

- Faça login no console do MySQL como root (recomendado) ou como qualquer usuário com privilégios FILE.
- Se estiver realizando a migração com um server Zabbix em execução, inicie o MySQL com a variável local_infile habilitada.
- Renomeie as tabelas antigas e crie as novas executando `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migração com o server parado

`max_execution_time` (no MySQL) ou `max_statement_time` (no MariaDB) deve ser desabilitado antes da migração dos 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, ajuste os caminhos dos arquivos nas consultas de acordo ou defina o valor de *secure_file_priv* como uma string vazia durante o upgrade.

Se o valor de *secure_file_priv* estiver vazio, a exportação/importação poderá ser realizada de qualquer local.

Se o valor de *secure_file_priv* for NULL, defina-o para o caminho que contém os dados exportados da tabela ('/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 da exportação dos 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 a importação devem ser realizadas no tmux/screen para garantir que a sessão não seja interrompida. Para instalações com TimescaleDB, pule esta seção e prossiga para PostgreSQL + TimescaleDB.

Veja também: Notas importantes

Atualização de tabelas

- Renomeie as tabelas usando `history_upgrade_prepare.sql`:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migração com o server parado

- Exporte o histórico atual, importe-o para a tabela temporária e, em seguida, insira os dados nas novas tabelas enquanto ignora 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 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 a importação devem ser realizadas em tmux/screen para garantir que a sessão não seja encerrada. O server do Zabbix deve estar desligado durante a atualização.

Veja também: Notas importantes

- Renomeie as tabelas usando `history_upgrade_prepare.sql`.
    - Se a compactação estiver habilitada (na instalação padrão), execute o script de `/usr/share/zabbix/sql-scripts/postgres`

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
```

- Se a compactação estiver desabilitada, execute o script de /usr/share/zabbix/sql-scripts/postgresql/timescaledb/

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
```

- Execute os scripts de migração de hypertable do TimescaleDB com base nas configurações de compactação:
  - Se a compactação estiver habilitada (na instalação padrão), execute os scripts de /usr/share/zabbix/sql-scripts/postgre

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
```

  - Se a compactação estiver desabilitada, execute os scripts de /usr/share/zabbix/sql-scripts/postgresql/timescaledb

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
```

Veja também: Dicas para melhorar o desempenho 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

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 da 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 adicional em disco quanto o próprio tamanho da tabela. 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 em disco disponível for uma preocupação, siga as instruções em Alterando a chave primária sem tempo de inatividade. Outras opções também estão disponíveis:

- Aumentar o parâmetro MySQL sort_buffer_size pode ajudar a reduzir o uso de espaço em disco ao reconstruir índices manualmente. No entanto, modificar essa 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 tempo de inatividade

1. Remova a chave primária atual da tabela auditlog e adicione a nova chave primária.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Reconstrua os índices (opcional, mas altamente recomendado; veja Notas importantes sobre a reconstrução de índices).

```
OPTIMIZE TABLE auditlog;
```

Alterando a chave primária sem downtime

O método manual de alteração da chave primária é descrito aqui. Como alternativa, você pode usar o toolkit pt-online-schema-change da Percona. Esse toolkit executa automaticamente as ações a seguir, ao mesmo tempo em que minimiza o espaço usado para alterar a tabela auditlog.

1. Crie uma nova tabela com a nova chave primária e crie 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 blocos (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 da chave primária durante a operação `ALTER TABLE`. No entanto, é altamente recomendável também reconstruir manualmente os índices com a instrução `REINDEX TABLE CONCURRENTLY` para garantir o desempenho ideal do banco de dados.

A reconstrução de índices pode exigir temporariamente até três vezes o espaço em disco atualmente usado pelos índices. Para obter o tamanho atual dos índices, você pode executar a seguinte consulta:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Se o espaço em disco disponível for uma preocupação, siga as instruções em Alterando a chave primária sem tempo de inatividade. Outras opções também estão disponíveis:

- Aumentar o parâmetro `maintenance_work_mem` do PostgreSQL pode ajudar a reduzir o uso de espaço em disco ao reconstruir índices manualmente. No entanto, modificar essa 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 `temp_tablespaces` do PostgreSQL para especificar um tablespace diferente para objetos temporários.
- 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 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. Recrie os índices (opcional, mas altamente recomendado; veja Notas importantes sobre a recriação de índices).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Alterando a chave primária sem downtime

O método manual para alterar a 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 blocos (várias instruções `INSERT INTO` com cláusulas `WHERE clock` conforme necessário) para evitar uso excessivo de recursos.

4. Remova a tabela antiga.

```
DROP TABLE auditlog_old;
```

Veja também

- Atualização do banco de dados para chaves primárias

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 server de banco de dados. Para instruções de instalação, consulte a documentação do TimescaleDB.

> **Warning:**
> Antes de instalar o TimescaleDB, instale uma versão suportada do PostgreSQL a partir do repositório oficial do PostgreSQL.

Habilite a extensão TimescaleDB para o banco de dados específico executando:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

A execução deste comando requer privilégios de administrador do banco de dados.

> **Note:**
> Se você usar um esquema 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 esquema/dados iniciais (consulte criação do banco de dados).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Ignore as mensagens de aviso informando que as melhores práticas não estão sendo seguidas ao executar o script `schema.sql` na versão 2.9.0 e superiores do TimescaleDB. Independentemente desse aviso, a configuração será concluída com sucesso.

A migração de dados existentes de histórico, tendências e log de auditoria pode levar bastante tempo. O server e o frontend do Zabbix devem ficar inativos durante o período de migração.

O script `schema.sql` define os seguintes parâmetros de housekeeping:

- Substituir período de histórico do item
- Substituir período de tendências do item

Para usar housekeeping particionado para histórico e tendências, ambas as opções devem estar 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 com mais de 7 dias

Para remover com sucesso os dados compactados pelo housekeeper, as opções *Substituir o período de histórico do item* e *Substituir o período de tendências do item* devem estar ativadas. Se a substituição estiver desativada e as tabelas tiverem blocos compactados, o housekeeper não removerá os dados dessas tabelas e avisos sobre configuração incorreta serão exibidos nas seções *Housekeeping* e *Informações do sistema*.

Todos esses parâmetros podem ser alterados em *Administração* > *Limpeza* após a instalação.

Note:
Você pode querer executar a ferramenta timescaledb-tune fornecida pelo TimescaleDB para otimizar os parâmetros de configuração do PostgreSQL em seu `postgresql.conf`.

Atualizando o schema do TimescaleDB

Ao atualizar o Zabbix para uma versão que contém novos hypertables do TimescaleDB, o Zabbix server não configura automaticamente esses hypertables (por exemplo, ao atualizar do Zabbix 6.4 para o 8.0, pois as versões 7.0.0 e 7.0.2 introduziram novos hypertables).

Para configurar novos hypertables do TimescaleDB, siga estas etapas:

1. Inicie o Zabbix server; isso atualizará o banco de dados existente.
2. Verifique no arquivo de log do server se a atualização do banco de dados foi concluída; quando concluída, pare o Zabbix server. Observe que o server registra um aviso se tentar habilitar a compactação para uma tabela que não é um hypertable.
3. Execute o script `postgresql/timescaledb/schema.sql`; isso configura os novos hypertables do TimescaleDB. Observe que, desde o Zabbix 7.0.0, a localização e o nome do script foram alterados de `postgresql/timescaledb.sql` para `postgresql/timescaledb/schema.sql`.

Attention:
Ignore as mensagens de aviso informando que as melhores práticas não estão sendo seguidas ao executar o script `schema.sql` no TimescaleDB versão 2.9.0 ou superior. Independentemente desse aviso, a configuração será concluída com sucesso.

Compressão 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 tabelas grandes pode levar bastante 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 gravada no log do server do Zabbix 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 usá-la.

Observe que há certas limitações impostas pela compressão, especificamente:

- Modificações em chunks compactados (inserts, deletes, updates) não são permitidas
- Alterações de schema em tabelas compactadas não são permitidas.

As configurações de compressão podem ser alteradas no bloco *Compressão de histórico, tendências e log de auditoria* na seção *Administração* > *Housekeeping* do frontend do Zabbix.

Parameter	Default	Comments
Enable compression	Enabled	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 `HousekeepingFrequency` horas (definido em `zabbix_server.conf`) Após desabilitar a compressão, novos chunks que entrarem no período de compressão não serão compactados. No entanto, todos os dados já compactados permanecerão compactados. Para descompactar chunks previamente compactados, siga as instruções na documentação do TimescaleDB.
Compress records older than	7d	Ao atualizar a partir de versões mais antigas do Zabbix com suporte ao TimescaleDB, a compressão não será habilitada por padrão. Este parâmetro não pode ser menor que 7 dias. Devido à imutabilidade dos chunks compactados, 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 trends, talvez seja interessante reduzir o "chunk_time_interval" das tabelas `trends` e `trends_uint` de 30 dias para 7 dias ou menos, dependendo de quantos items usam trends. O objetivo dessa configuração é seguir as melhores práticas do TimescaleDB e garantir que o tamanho do chunk permaneça dentro dos recursos disponíveis do sistema.

Configuração do Elasticsearch

O Zabbix pode armazenar dados de histórico em Elasticsearch como uma alternativa a um banco de dados relacional.

Attention:
O suporte ao Elasticsearch atualmente é experimental.

Este guia aborda a configuração das versões suportadas do Elasticsearch. Se você estiver usando uma versão diferente, algumas funcionalidades podem não funcionar como esperado.

O Elasticsearch pode armazenar os seguintes tipos de valor:

Item value type	Database table	Elasticsearch type
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text
Binary	history_bin	not supported by Zabbix
JSON	history_json	json

Note:
O Elasticsearch não aceita arrays JSON. Um valor JSON deve ser um único objeto ou um conjunto de objetos.

Notas importantes

- O Elasticsearch requer libcurl. Consulte os requisitos para obter detalhes.
- O housekeeper não exclui dados do Elasticsearch. Para controlar por quanto tempo os dados são mantidos, consulte Index Lifecycle Management.
- O Zabbix **não** calcula nem armazena trends no Elasticsearch. Considere estender o período de armazenamento do histórico para preservar dados mais antigos.
- 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.
- O Elasticsearch não é suportado para proxies do Zabbix.

Se o Elasticsearch ainda não estiver instalado, consulte o guia oficial de instalação antes de prosseguir.

Configurando o Elasticsearch

Para armazenar dados históricos no Elasticsearch, você precisa:

- Criar um índice para cada tipo de valor que deseja armazenar—é aqui que o Elasticsearch armazena os dados, semelhante a uma tabela em um banco de dados relacional.
- Definir um mapeamento para cada índice—isso define a estrutura dos dados, semelhante ao esquema de uma tabela.
- Configurar um pipeline de ingestão para processar valores antes do armazenamento (necessário para valores JSON e índices baseados em data).

O Elasticsearch pode armazenar dados em um único índice por tipo de valor ou em vários índices baseados em data. Ambas as abordagens são descritas abaixo.

Armazenando histórico em um único índice

Nesta abordagem, todos os dados de histórico para um determinado tipo de valor são gravados em um único índice (por exemplo, `uint` ou `text`).

Para criar um índice para o tipo de valor *Numérico (sem sinal)*, envie a seguinte solicitação (com `/uint` na URL) para sua instância do Elasticsearch:

```
curl -X PUT \
 http://localhost:9200/uint \
 -H 'content-type:application/json' \
 -d '{
     "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": { "type": "long" }
      }
   }
}'
```

O Elasticsearch responderá com uma confirmação de que o índice foi criado:

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Solicitações semelhantes devem ser enviadas para cada tipo de valor adicional que você deseja armazenar no Elasticsearch.

> **Note:**
> Os mapeamentos para todos os tipos de valor estão disponíveis no repositório de código-fonte do Zabbix.

Por exemplo, para criar um índice para o tipo de valor *Texto*:

```
curl -X PUT \
 http://localhost:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
```

```
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Tipo de valor JSON

Ao contrário de outros tipos de valor, os valores JSON requerem processamento adicional antes do armazenamento.

O índice abaixo utiliza campos separados para valores analisados e brutos, portanto, um ingest pipeline é necessário para analisar cada valor como JSON e armazená-lo no campo correto.

Para criar um índice para o tipo de valor *JSON*, envie a seguinte solicitação (com `/json` na URL) para sua instância do Elasticsearch.

```
curl -X PUT \
 http://localhost:9200/json \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
    }
  }
}'
```

Em seguida, crie o ingest pipeline:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json \
 -H 'content-type:application/json' \
 -d '{
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "set": {
            "if": "ctx.value_parsed == null",
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ],
   "on_failure": [
      {
         "set": {
            "field": "value_raw",
            "value": "{{{ value }}}"
```

```
         }
      }
   ]
}'
```

O Elasticsearch responderá com uma confirmação de que o ingest pipeline foi criado:

```
{"acknowledged": true}
```

Armazenando o histórico em índices baseados em data

Em vez de gravar todos os dados de histórico em um único índice (por exemplo, `uint`), o Elasticsearch pode distribuir esses dados entre vários índices baseados em data (por exemplo, `uint-2026-01-01`, `uint-2026-01-02`). Isso facilita o gerenciamento do volume de dados e da retenção ao longo do tempo.

Para habilitar isso, você precisa:

- Criar um index template para cada tipo de valor que deseja armazenar — isso informa ao Elasticsearch qual mapeamento aplicar quando ele criar automaticamente um novo índice baseado em data.
- Criar um ingest pipeline para cada tipo de valor — ele processa cada valor recebido e o encaminha para o índice baseado em data correto.

Indexar templates

Para criar um template para o índice `text`, envie uma solicitação com os seguintes detalhes:

- Use `_template/text_template` na URL da sua instância do Elasticsearch.
- Use `"text*"` no campo `"index_patterns"` para corresponder ao nome do índice.
- Use um mapeamento para o tipo de valor `text` (veja os mapeamentos no repositório fonte do Zabbix).

```
curl -X PUT \
 http://localhost:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "text*" ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Template para o índice `json`:

```
curl -X PUT \
 http://localhost:9200/_template/json_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "json*" ],
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
```

```
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
      }
   }
}'
```

Pipelines de ingestão

Para criar um pipeline de ingestão para o índice `text`:

- Use `_ingest/pipeline/text-pipeline` na URL da sua instância do Elasticsearch.
- Inclua um processador `date_index_name` para direcionar cada valor para o índice baseado em data correto com base em seu timestamp.

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/text-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily text index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": ["UNIX"],
            "index_name_prefix": "text-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Para o índice `json`, o pipeline também precisa analisar o valor JSON antes de direcioná-lo para o índice correto:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily json index naming"
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "script": {
            "source": "if (ctx.value_parsed == null || !(ctx.value_parsed instanceof Map)) { ctx.value_raw
         }
      },
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [ "UNIX" ],
            "index_name_prefix": "json-",
            "date_rounding": "d"
         }
      }
   ]
```

```
}'
```

Configurando o Zabbix server

No arquivo de configuração do seu Zabbix server (`zabbix_server.conf`), defina o parâmetro `HistoryProvider`.

Por exemplo, para armazenar valores dos tipos *Character*, *Log*, *Text* e *JSON* no Elasticsearch (mantendo os valores *Numeric* em um banco de dados):

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Se você estiver usando índices baseados em data, adicione `date_index=1` ao parâmetro:

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

Após fazer as alterações, reinicie o Zabbix server:

```
systemctl restart zabbix-server
```

Configurando o frontend do Zabbix

No seu arquivo de configuração do frontend do Zabbix (`zabbix.conf.php`), defina a variável `$HISTORY_PROVIDERS` para corresponder à configuração do server:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Solução de problemas

As etapas a seguir podem ajudar você a solucionar problemas com sua configuração do Elasticsearch:

1. Verifique os logs do Elasticsearch ou do server do Zabbix em busca de erros.
2. Para identificar consultas lentas, use a opção `log_slow_queries` no parâmetro de configuração do server do Zabbix `HistoryProvider`.
3. Verifique se o Elasticsearch permite acesso a partir do server do Zabbix e do frontend do Zabbix.
4. Verifique se `auto_create_index` está habilitado:

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Para habilitá-lo, envie a seguinte solicitação:

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. Verifique se os mappings, templates e ingest pipelines estão corretos enviando solicitações `GET` para suas respectivas URLs:

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

Você pode comparar as respostas recebidas com as respostas esperadas na documentação da API do Elasticsearch.

6. Verifique se há algum shard em estado de falha; reiniciar o Elasticsearch pode resolver isso.
7. Consulte o Elasticsearch para ver se os dados coletados pelo Zabbix estão armazenados, por exemplo:

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Se você precisar redefinir sua configuração do Elasticsearch e começar do zero, poderá excluir todos os índices, templates e ingest pipelines:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

Configuração do ClickHouse

O Zabbix pode armazenar dados históricos no ClickHouse como alternativa a um banco de dados relacional.

> **Note:**
> O ClickHouse também pode ser usado como uma fonte de dados de telemetria para monitoramento do desempenho de aplicações (APM); para obter detalhes, consulte *Administração* > *Fonte de dados*.

Este guia aborda a configuração das versões compatíveis do ClickHouse. Se você estiver usando uma versão diferente, algumas funcionalidades poderão não funcionar conforme o esperado.

O ClickHouse pode armazenar os seguintes tipos de valor:

Tipo de valor do item	Tabela do banco de dados	Tipo do ClickHouse
Numérico (sem sinal)	history_uint	uint
Numérico (ponto flutuante)	history	dbl
Caractere	history_str	str
Log	history_log	log
Texto	history_text	text
Binário	history_bin	não compatível com o Zabbix
JSON	history_json	json

> **Note:**
> O ClickHouse não aceita arrays JSON. Um valor JSON deve ser um único objeto ou um conjunto de objetos. Além disso, o ClickHouse trata chaves JSON com NULL da mesma forma que chaves ausentes.

Observações importantes

- O housekeeper não exclui dados do ClickHouse. Para controlar por quanto tempo os dados são mantidos, configure o período de armazenamento de dados do ClickHouse no ClickHouse.
- O Zabbix **não** calcula nem armazena tendências no ClickHouse. Considere estender o período de armazenamento do histórico para preservar dados mais antigos.
- Se quiser migrar os dados do histórico de um banco de dados existente do Zabbix (MySQL ou PostgreSQL) para o ClickHouse, consulte os scripts de migração do esquema e do histórico do ClickHouse.
- O ClickHouse não é compatível como armazenamento de histórico para proxies do Zabbix, mas pode ser usado como uma fonte de dados de telemetria para um proxy do Zabbix com APM habilitado.
- Se você tiver um cluster do ClickHouse, consulte Configurando clusters do ClickHouse.

Configurando o ClickHouse

Você precisa criar e configurar um banco de dados e um usuário do Zabbix, além de importar o esquema do banco de dados.

Este guia fornece instruções para instalações do ClickHouse via Docker ou pacote.

Docker

1. Crie e configure o banco de dados e o usuário do Zabbix ao executar o container do ClickHouse:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Confirme que o ClickHouse está funcionando e que você consegue se conectar a ele:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Importe o esquema do banco de dados usando o script history_all.sh do seu diretório do Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
O período de armazenamento dos dados (Time-To-Live, ou TTL) para o ClickHouse é, por padrão, de 31 dias. Para alterá-lo, use a opção `--ttl` ao importar o esquema do banco de dados (por exemplo, `--ttl 604800` para 7 dias) ou configure isso depois.

4. Verifique se as tabelas foram criadas:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Pacotes

1. Inicie o ClickHouse:

```
sudo service clickhouse-server start
```

2. Crie e configure o banco de dados e o usuário do Zabbix:

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Importe o schema do banco de dados usando o script history_all.sh do seu diretório do Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
O período de armazenamento dos dados (Time-To-Live, ou TTL) para o ClickHouse é, por padrão, de 31 dias. Para alterá-lo, use a opção `--ttl` ao importar o schema do banco de dados (por exemplo, `--ttl 604800` para 7 dias) ou configure isso posteriormente.

4. Verifique se as tabelas foram criadas:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Configurando o Zabbix server

No arquivo de configuração do Zabbix server (`zabbix_server.conf`), defina o parâmetro `HistoryProvider`.

Por exemplo, para armazenar todos os tipos de valores suportados no ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Após fazer as alterações, reinicie o Zabbix server:

```
systemctl restart zabbix-server
```

Configurando o frontend do Zabbix

No seu arquivo de configuração do frontend do Zabbix (`zabbix.conf.php`), defina a variável `$HISTORY_PROVIDERS` para corresponder à configuração do server:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Configuração adicional

As etapas abaixo são opcionais. Você não precisa delas para uma configuração básica.

Configurando o período de armazenamento de dados do ClickHouse

O período de armazenamento de dados (Time-To-Live, ou TTL) do ClickHouse é, por padrão, de 31 dias. Para alterá-lo, execute os comandos abaixo.

Os exemplos abaixo usam Docker. Se você instalou o ClickHouse usando pacotes, execute as consultas diretamente no cliente do ClickHouse.

1. Altere a tabela (substitua `history_json` e 3600 pelos valores necessários):

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Aplique as alterações imediatamente:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Verifique se o período de armazenamento de dados foi alterado:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Reinicie o Zabbix server para atualizar o período de armazenamento de dados em *Administration* > *Housekeeping*:

```
systemctl restart zabbix-server
```

Configurando clusters do ClickHouse

Esta seção fornece etapas de configuração para clusters do ClickHouse com replicação e sharding.

As etapas abaixo usam Docker e o ClickHouse `cluster_2S_2R` da documentação do ClickHouse (executando ClickHouse 26.4.4.38 com ClickHouse Keeper).

Antes de começar, é recomendável colocar um balanceador de carga na frente do seu cluster do ClickHouse para o Zabbix. A configuração do server e do frontend do Zabbix suporta apenas uma única URL do ClickHouse, então ela só pode apontar para um nó. Se esse nó ficar indisponível, um balanceador de carga redirecionará o Zabbix para um nó saudável, usando a mesma URL.

1. Crie o banco de dados do Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

Attention:
A cláusula `ON CLUSTER` alcança apenas os nós que fazem parte do cluster no momento em que você executa o comando. Se você adicionar nós ao cluster mais tarde, execute este comando novamente.

2. Crie e configure o usuário do Zabbix:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
```

```
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Importe o esquema do banco de dados usando o script `history_all.sh` do seu diretório do Zabbix:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
O período de retenção dos dados (Time-To-Live, ou TTL) para o ClickHouse é, por padrão, de 31 dias. Para alterá-lo, use a opção `--ttl` ao importar o esquema do banco de dados (por exemplo, `--ttl 604800` para 7 dias) ou configure isso mais tarde.

Note:
Se o seu cluster do ClickHouse exigir caminhos explícitos para réplicas (por exemplo, se a configuração do seu server do ClickHouse não definir caminhos padrão para tabelas replicadas), execute os scripts de esquema (`history_schema.sh`, `history_uint_schema.sh`, etc.) individualmente e defina um caminho diferente para cada tabela:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Para mais detalhes sobre replicação no ClickHouse, consulte Replicating data e Replicated* table engines na documentação do ClickHouse.

4. Configure o sharding para cada tipo de tabela. Renomeie as tabelas e crie uma tabela `Distributed` com o nome original, apontando para a tabela renomeada. Essa tabela `Distributed` é a partir da qual o Zabbix realmente lê e grava; a tabela divide os dados automaticamente entre os shards. Por exemplo, para a tabela `history`:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Para mais detalhes sobre sharding no ClickHouse, consulte Distributed table engine na documentação do ClickHouse.

5. Configure o server do Zabbix.

6. Configure o frontend do Zabbix.

7. Verifique se os dados do Zabbix estão distribuídos entre os shards. A contagem de linhas em `zabbix.history_local` pode mudar entre verificações repetidas, já que o server do Zabbix insere novos dados continuamente. Execute este comando em cada nó:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Verifique se a replicação está sincronizada (`active_replicas` deve corresponder a `total_replicas`):

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Solução de problemas

As etapas a seguir podem ajudar você a solucionar problemas com sua configuração do ClickHouse:

1. Verifique os logs do ClickHouse ou do server do Zabbix em busca de erros.
2. Para identificar consultas lentas, use a opção `log_slow_queries` no parâmetro de configuração do server do Zabbix `HistoryProvider`.
3. Verifique se o ClickHouse permite acesso a partir do server do Zabbix e do frontend do Zabbix.
4. Consulte o ClickHouse para ver se os dados coletados pelo Zabbix estão armazenados, por exemplo:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Observações específicas da distribuição sobre a configuração do NGINX para o Zabbix

RHEL

No RHEL 8, 9 e 10, o NGINX está incluído no AppStream — não são necessários repositórios adicionais. Basta habilitar e instalar com:

```
dnf module list nginx
dnf -y install nginx
```

No RHEL 7, o NGINX pode ser instalado a partir do repositório CentOS 7 Extras (se você estiver usando o CentOS) por meio de:

```
yum -y install nginx
```

Se preferir as versões upstream mais recentes, configure o repositório oficial do NGINX em `/etc/yum.repos.d/nginx.repo`:

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

No SUSE Linux Enterprise Server 15, você precisa configurar o `php-fpm` (o caminho para o arquivo de configuração pode variar 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
```

Configuração de SAML com Microsoft Entra ID

Visão geral

Esta seção fornece diretrizes para configurar o login único e o provisionamento de usuários no Zabbix a partir do Microsoft Entra ID (anteriormente Microsoft Azure Active Directory) usando autenticação SAML 2.0.

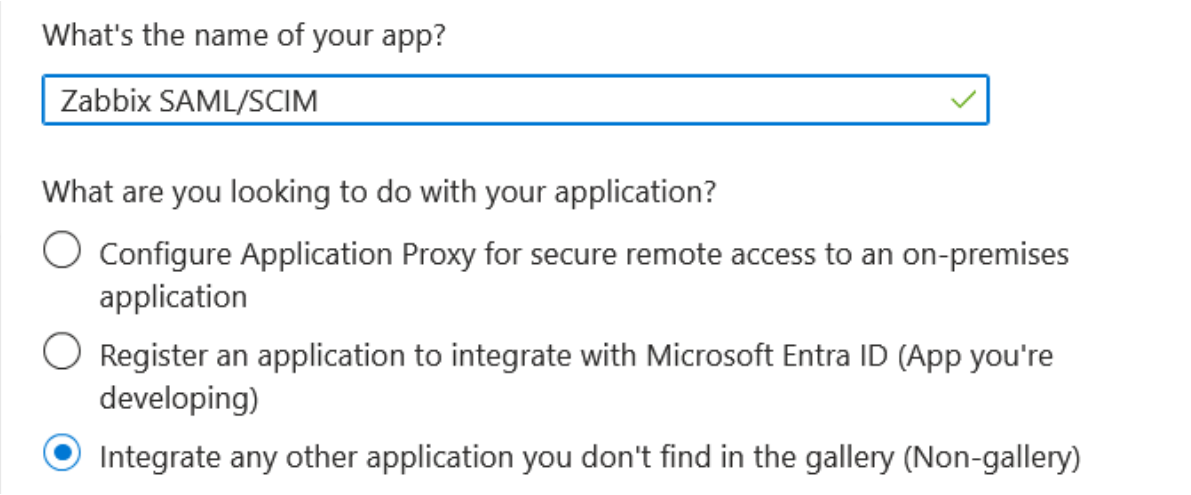
Configuração do Microsoft Entra ID

Criando uma aplicação

1. Faça login no Microsoft Entra admin center em Microsoft Entra ID. Para fins de teste, você pode criar uma conta de avaliação gratuita no Microsoft Entra ID.

2. No Microsoft Entra admin center, selecione *Applications* > *Enterprise applications* > *New application* > *Create your own application*.

3. Adicione o nome do seu app e selecione a opção *Integrate any other application...*. Depois disso, clique em *Create*.

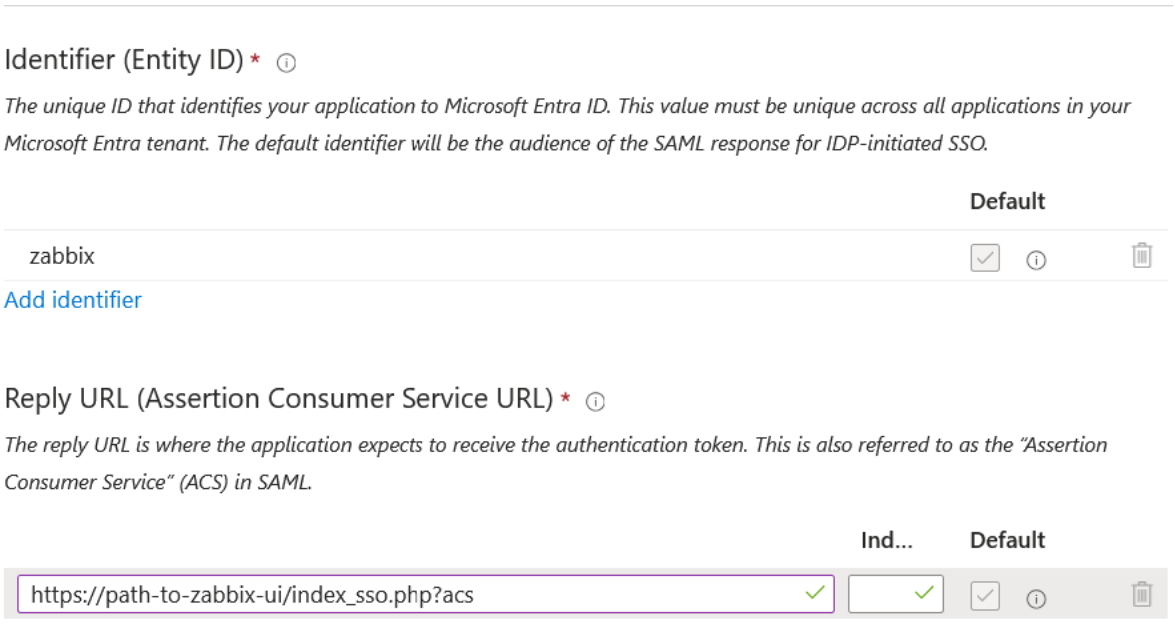


Configurando o logon único

1. Na página do seu aplicativo, acesse *Configurar logon único* e clique em *Começar*. Em seguida, selecione *SAML*.

2. Edite *Configuração básica do SAML*:

- Em *Identificador (ID da entidade)*, defina um nome exclusivo para identificar seu aplicativo no Microsoft Entra ID, por exemplo, `zabbix`;
- Em *URL de resposta (URL do Serviço de Consumidor de Asserção)*, defina o endpoint de logon único do Zabbix: `https://<path-to-zabbix-ui>/index_sso.php?acs`:



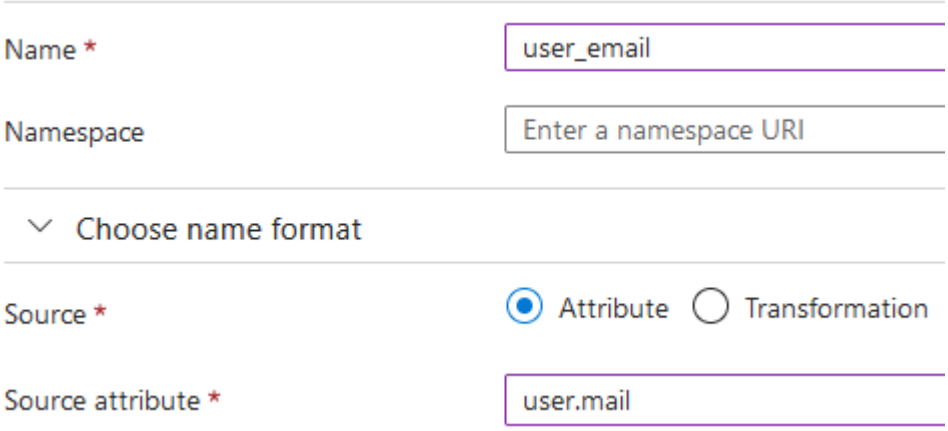
Observe que `https` é obrigatório. Para usar isso com o Zabbix, adicione a seguinte linha ao arquivo de configuração do frontend:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

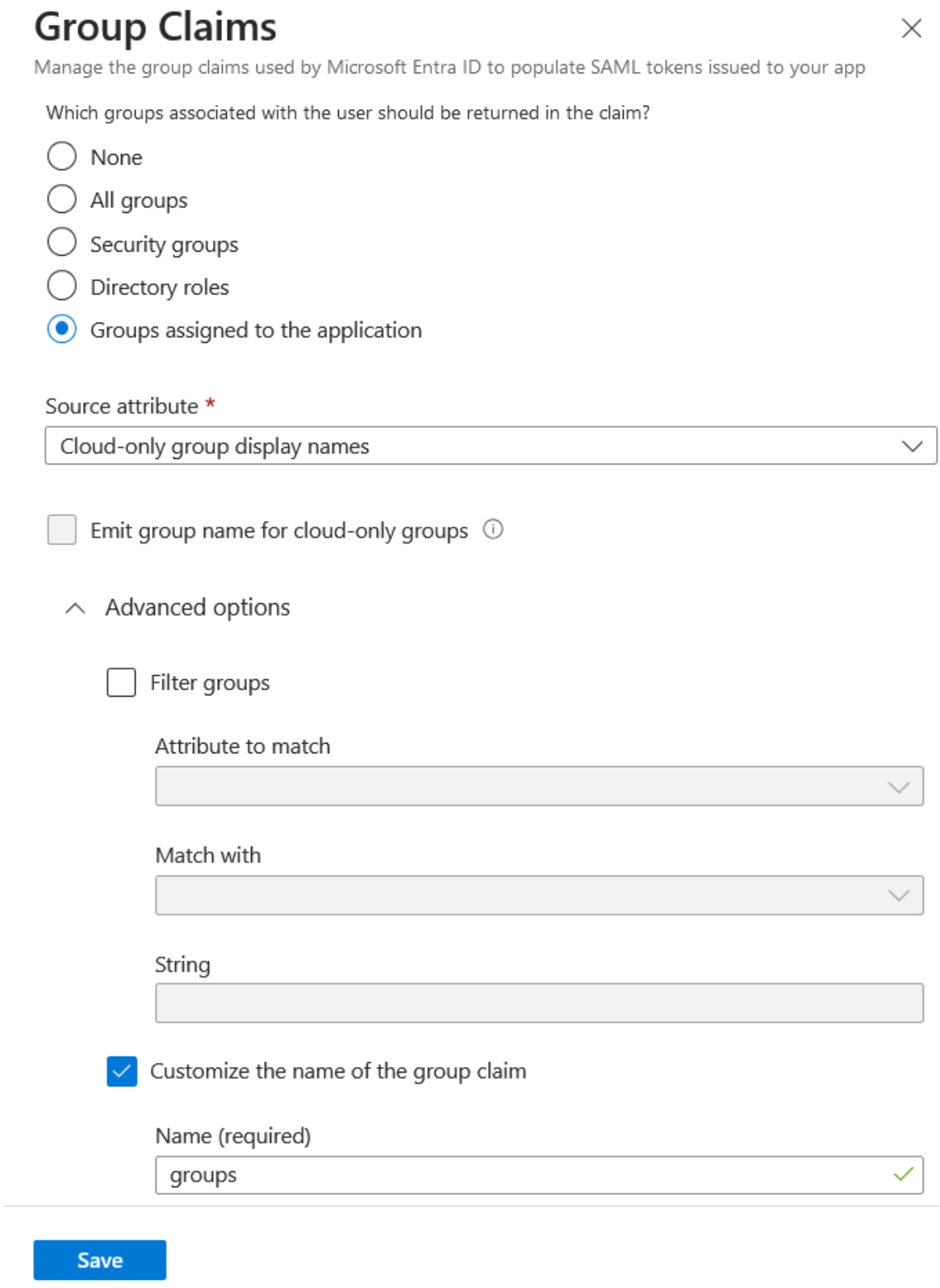
3. Edite *Atributos e declarações*. Você deve adicionar todos os atributos que deseja passar para o Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Os nomes dos atributos são arbitrários. Podem ser usados nomes de atributos diferentes; no entanto, é necessário que eles correspondam ao respectivo valor do campo 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 ao Zabbix:



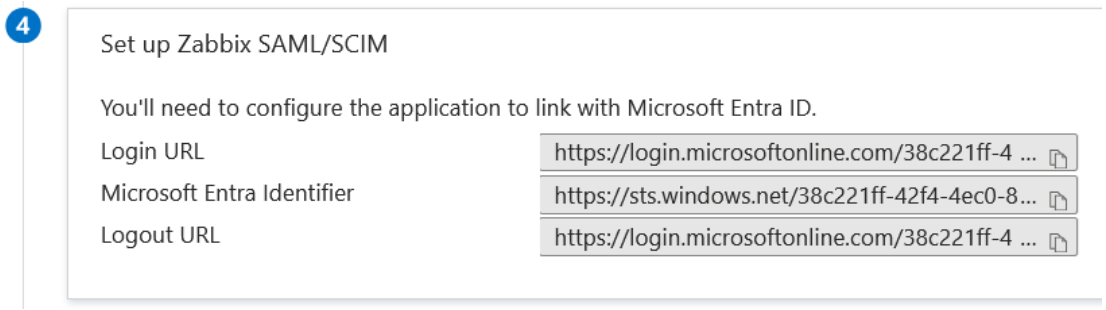
É importante que, nessa declaração, os nomes dos grupos (em vez dos IDs dos grupos) sejam passados ao Zabbix pelo *Atributo de origem* selecionado. Caso contrário, o provisionamento de usuários JIT não funcionará corretamente.

4. Em *Certificados SAML*, baixe o certificado Base64 fornecido pelo Entra ID e coloque-o em `conf/certs` da instalação do frontend do Zabbix (o local pode variar de acordo com o sistema operacional ou a origem da instalação do Zabbix).

Defina as permissões como 644 executando:

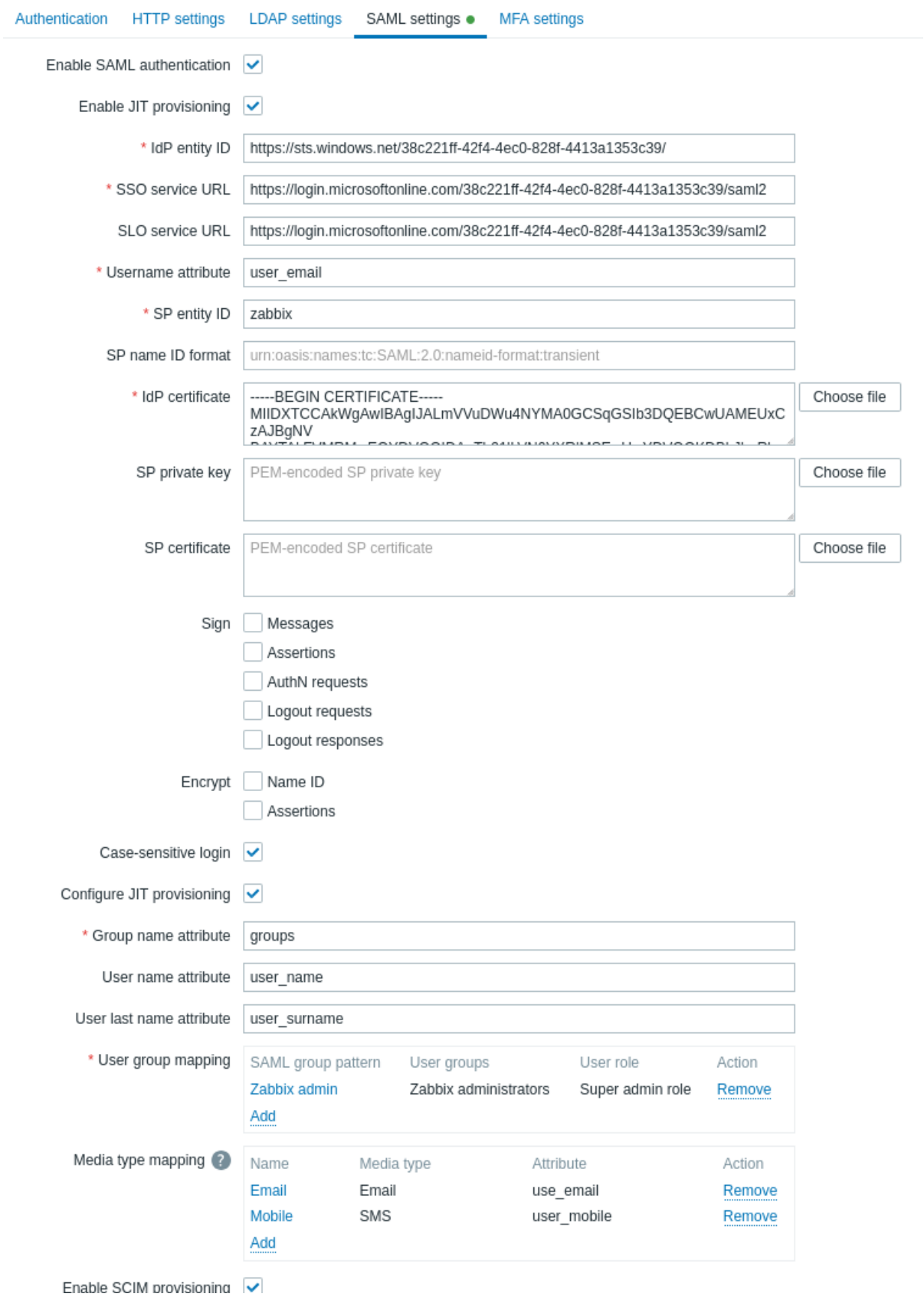
```
chmod 644 entra.cer
```

5. Use os valores de *Configurar <your app name>* no Entra ID para configurar a autenticação SAML do Zabbix (consulte a próxima seção):



Configuração do Zabbix

1. No Zabbix, acesse as configurações do 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
ID da entidade do IdP	Identificador do Microsoft Entra	
URL do serviço SSO	URL de login	
URL do serviço SLO	URL de logout	
ID da entidade do SP	Identificador (ID da entidade)	
Atributo de nome de usuário	Atributo personalizado (claim)	`user_email`
Atributo de nome do grupo	Atributo personalizado (claim)	`groups`
Atributo de nome do usuário	Atributo personalizado (claim)	`user_name`

Campo do Zabbix	Campo de configuração no Entra ID	Valor de exemplo
Atributo de sobrenome do usuário	Atributo personalizado (claim)	`user_lastname`

2. Adicione o certificado Base64 fornecido pelo Entra ID.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido no arquivo de configuração do frontend (`zabbix.conf.php`), você poderá colar o texto do certificado ou carregar o arquivo do certificado no frontend durante a configuração do SAML — não são necessários arquivos no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no arquivo de configuração, o certificado deverá estar disponível no sistema de arquivos (por padrão, em `ui/conf/certs` ou no caminho configurado no arquivo de configuração), e o frontend não armazenará os certificados no banco de dados. Observe que, se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será considerado, e os certificados serão lidos de `ui/conf/certs`.

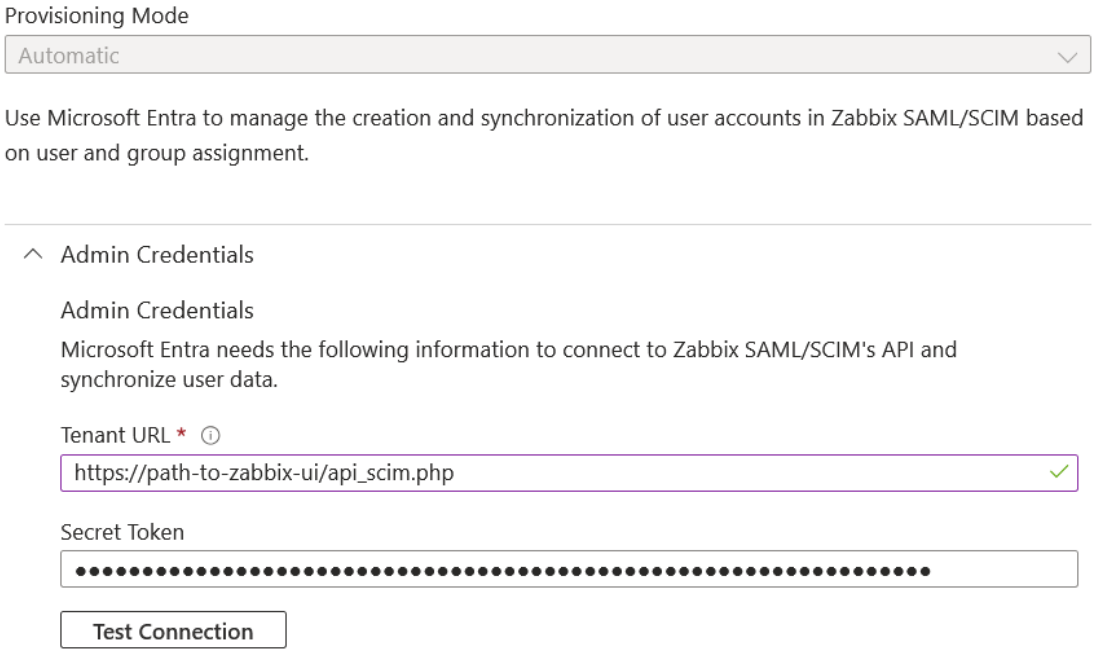
Também é necessário configurar o mapeamento de grupos de usuários. O mapeamento de mídias é opcional.

3. Clique no botão *Atualizar* para salvar essas configurações.

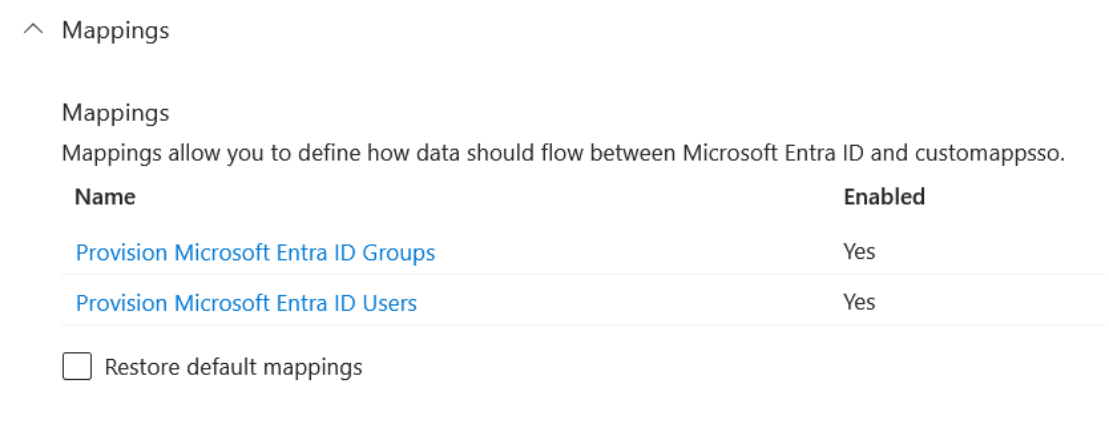
Provisionamento de usuários SCIM

1. Na página do aplicativo do Entra ID, no menu principal, abra a página Provisioning. Clique em *Get started* e, em seguida, selecione o modo de provisionamento automático:

- Em *Tenant URL*, defina o seguinte valor: `https://<path-to-zabbix-ui>/api_scim.php`
- Em *Secret token*, insira um token da API do Zabbix com permissões de Super admin.
- Clique em *Test connection* para verificar se a conexão foi estabelecida.

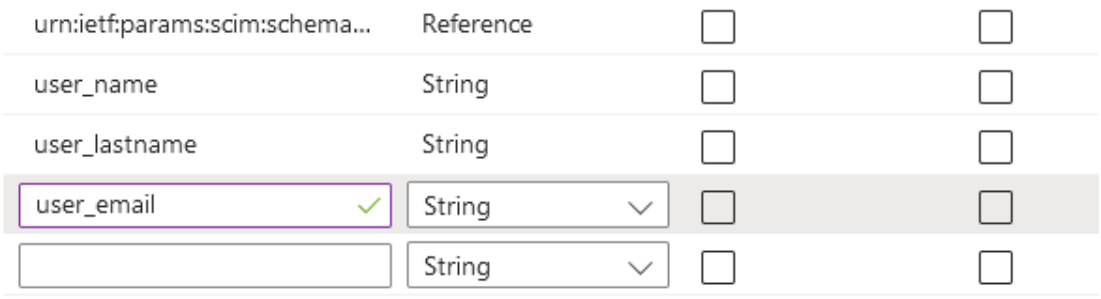


2. Agora você pode adicionar todos os atributos que serão enviados via SCIM para o Zabbix. Para isso, clique em *Mappings* e depois em *Provision Microsoft Entra ID Users*.



Na parte inferior da lista Attribute Mapping, habilite *Show advanced options* e, em seguida, clique em *Edit attribute list for custom-appsso*.

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 Attribute Mapping, clique em *Add New Mapping* 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, vá para *Microsoft Entra admin center* e adicione usuários/grupos nas respectivas páginas Users e Groups.

5. Quando os usuários e grupos tiverem sido criados no Entra ID, você pode ir ao menu *Users and groups* do seu aplicativo e adicioná-los ao app.

6. Vá ao menu *Provisioning* do seu app e clique em *Start provisioning* para que os usuários sejam provisionados no Zabbix.

Observe que a solicitação PATCH de Users no Entra ID não oferece suporte a alterações em mídia.

Assinatura de solicitações de autenticação

É possível configurar o Entra ID para validar a assinatura de solicitações de autenticação assinadas.

Para que isso funcione, crie as chaves pública e privada:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Atribua as 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

Podem ocorrer problemas de autenticação com navegadores MicroSoft Edge quando um usuário que tenta fazer login no Zabbix via SAML já está conectado com o perfil do MicroSoft Edge. Como sinal desse problema, o usuário pode conseguir fazer login no Zabbix usando o MicroSoft Edge no modo privado.

Para evitar problemas de autenticação nesse caso, pode ser necessário definir `requestedAuthnContext` como `false` no arquivo de configuração do frontend do Zabbix.

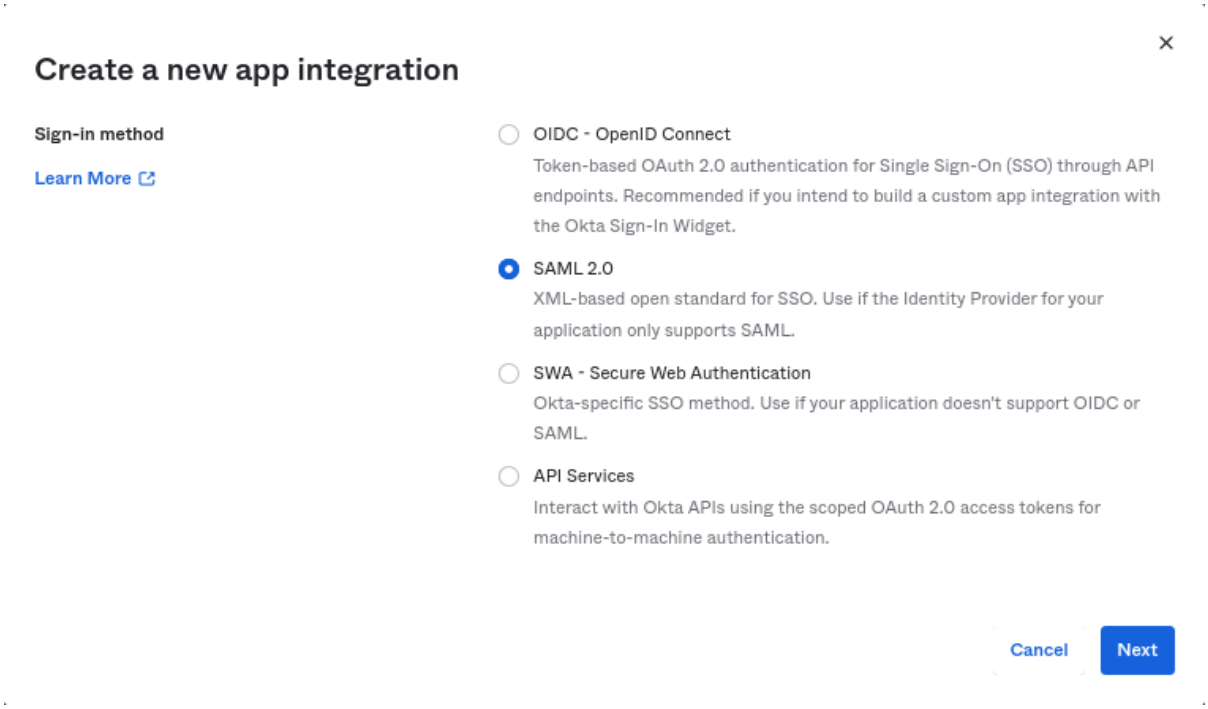
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

Configuração de SAML com Okta

Esta seção fornece orientações para configurar Okta para habilitar a autenticação SAML 2.0 e o provisionamento de usuários para o Zabbix.

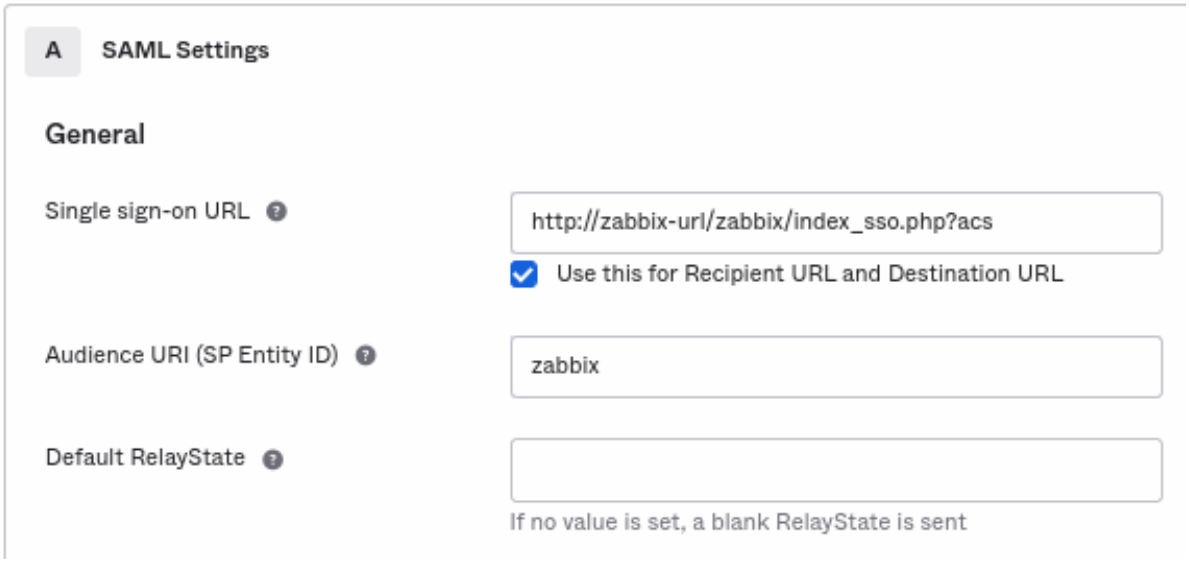
Configuração do Okta

1. Acesse https://developer.okta.com/signup/ e registre-se/entre na sua conta.
2. Na interface web do Okta, navegue até *Applications* > *Applications*.
3. Clique em *Create App Integration*.

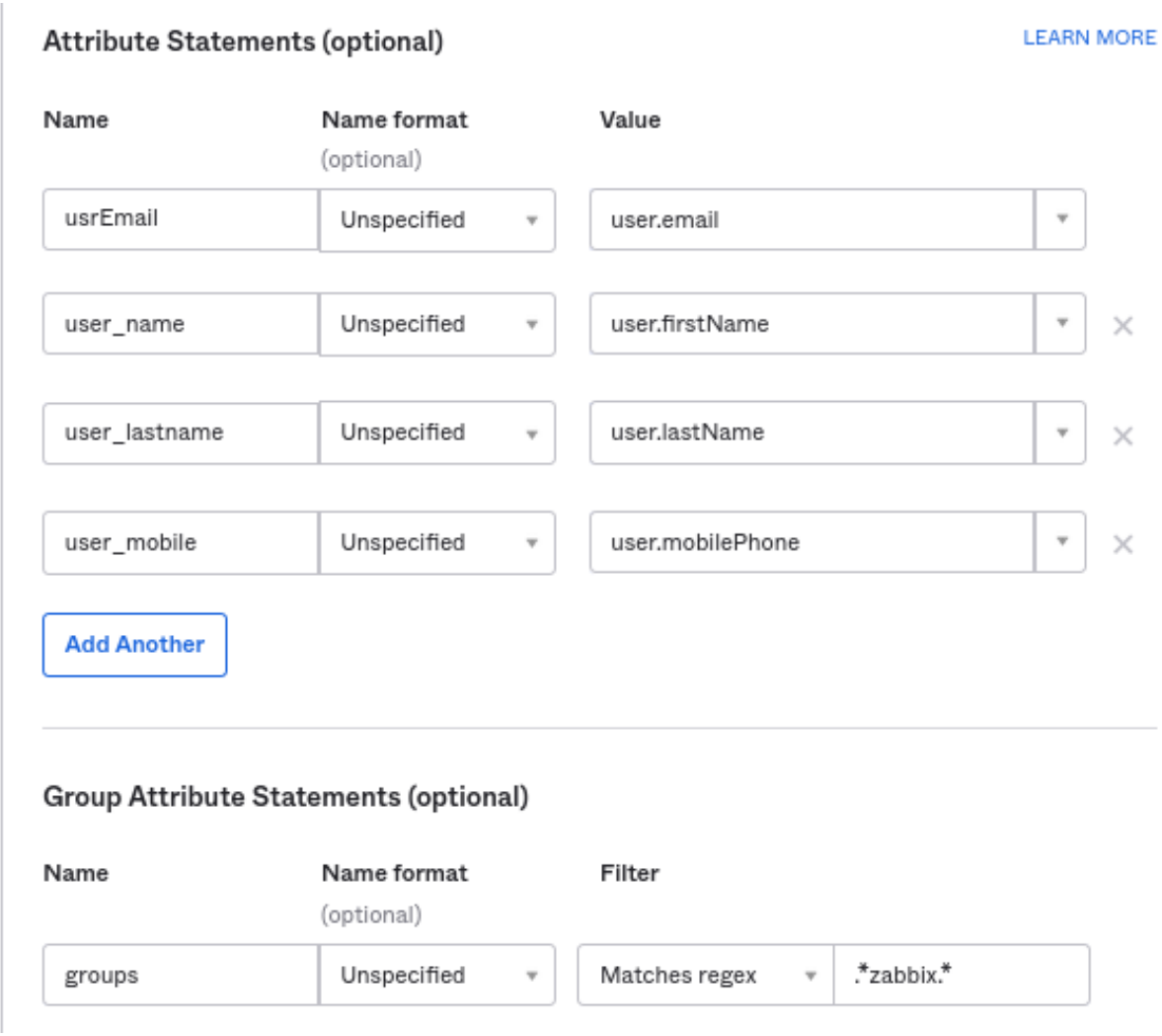


Selecione ”SAML 2.0” como método de login e clique em *Next*.

4. Nas configurações gerais, preencha o nome do app e clique em *Next*.
5. Na configuração SAML, insira os valores fornecidos abaixo e, em seguida, clique em *Next*.



- Em **General**, adicione:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Observe o uso de "http", e não "https", para que o parâmetro `acs` não seja removido na requisição. A caixa de seleção *Use this for Recipient URL and Destination URL* também deve estar marcada.
  - *Audience URI (SP Entity ID)*: `zabbix<br>` Observe que esse valor será usado dentro da asserção SAML como um identificador exclusivo do provedor de serviços (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 for necessário um redirecionamento personalizado, ele pode ser adicionado no Zabbix em *Users* > *Users*.
  - Preencha os outros campos de acordo com suas preferências.
- Em **Attribute Statements/Group Attribute Statements**, adicione:



Essas declarações de atributo são inseridas nas asserções SAML compartilhadas com o Zabbix.

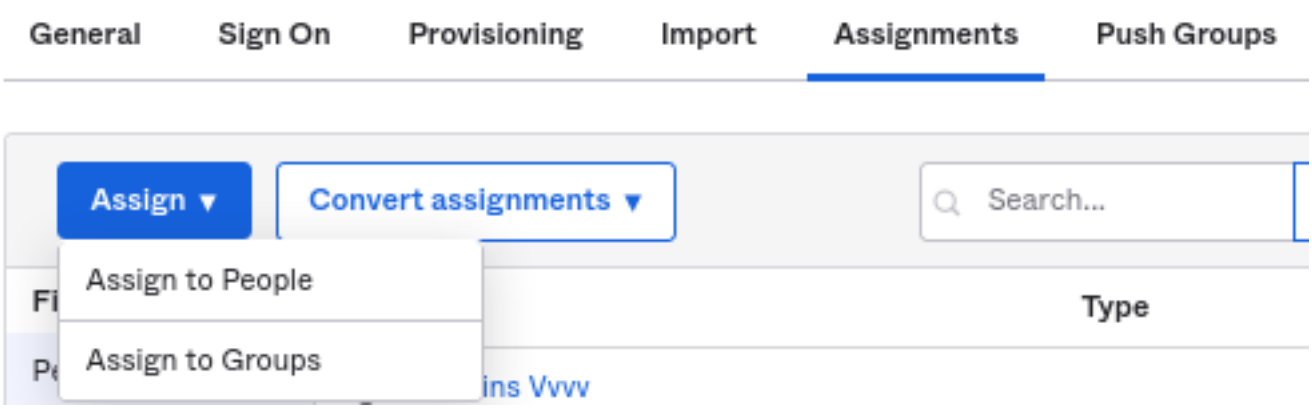
Os nomes de atributo usados aqui são exemplos arbitrários. Você pode usar nomes de atributo diferentes; no entanto, é necessário que eles correspondam ao valor do respectivo campo nas configurações SAML do Zabbix.

Se você quiser configurar o login SAML no Zabbix *sem* provisionamento de usuários JIT, então apenas o atributo de e-mail é necessário.

Note:
Se estiver planejando usar uma conexão criptografada, gere os certificados de criptografia privado e público e, em seguida, envie o certificado público para o Okta. O formulário de envio do certificado aparece quando *Assertion Encryption* está definido como "Encrypted" (clique em *Show Advanced Settings* para encontrar esse 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" da aplicação recém-criada e clique no botão *Assign*, depois selecione "Assign to People" no menu suspenso.



8. Na janela pop-up que aparece, atribua o app às pessoas que usarão SAML 2.0 para autenticar no Zabbix e, em seguida, clique em *Save and go back*.

9. Baixe o certificado do IdP. Defina permissões 644 para ele executando:

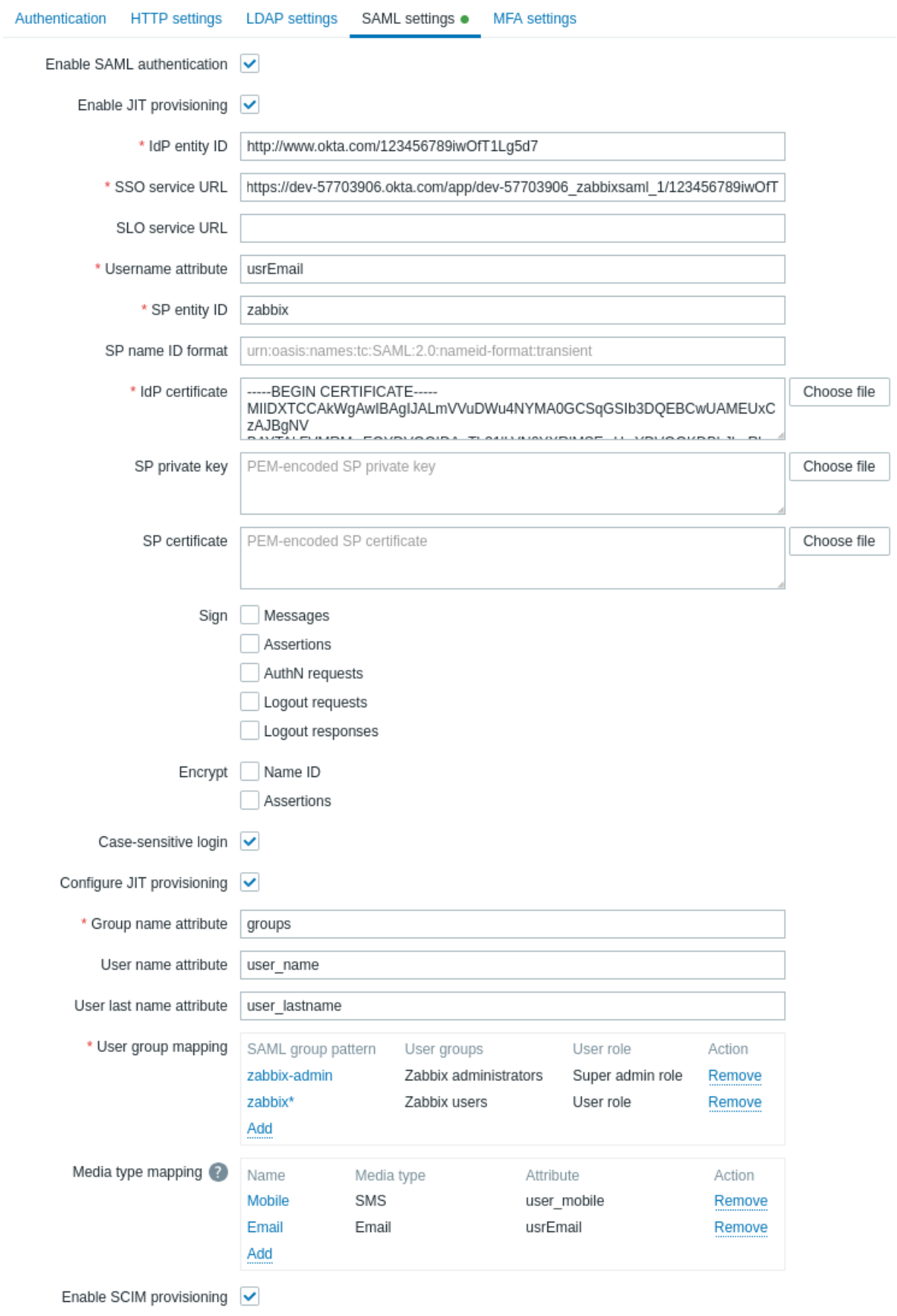
```
chmod 644 idp.crt
```

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 essa 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
ID da entidade do IdP	Emissor do provedor de identidade	
URL do serviço SSO	URL de logon único do provedor de identidade	
Atributo de nome de usuário	Nome do atributo	`usrEmail`
ID da entidade do SP	URI do público-alvo	`zabbix`
Atributo de nome do grupo	Nome do atributo	`groups`

Campo do Zabbix	Campo de configuração no Okta	Valor de exemplo
Atributo de nome do usuário	Nome do atributo	`user_name`
Atributo do sobrenome do usuário	Nome do atributo	`user_lastname`

Também é necessário configurar o mapeamento de grupos de usuários e mídias.

2. Adicione o certificado Base64 fornecido nas instruções de configuração SAML do Okta.

Se `$SSO['CERT_STORAGE'] = 'database'` estiver definido no arquivo de configuração do frontend (`zabbix.conf.php`), você poderá colar o texto do certificado ou carregar o arquivo do certificado no frontend durante a configuração do SAML — não são necessários arquivos no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no arquivo de configuração, o certificado deverá estar disponível no sistema de arquivos (por padrão, em `ui/conf/certs` ou no caminho configurado no arquivo de configuração), e o frontend não armazenará os certificados no banco de dados.
Observe que, se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será considerado, e os certificados serão lidos de `ui/conf/certs`.

3. Se *Criptografia da asserção* estiver definida como "Criptografada" no Okta, a caixa de seleção "Assertions" do parâmetro *Criptografar* também deverá ser marcada no Zabbix.

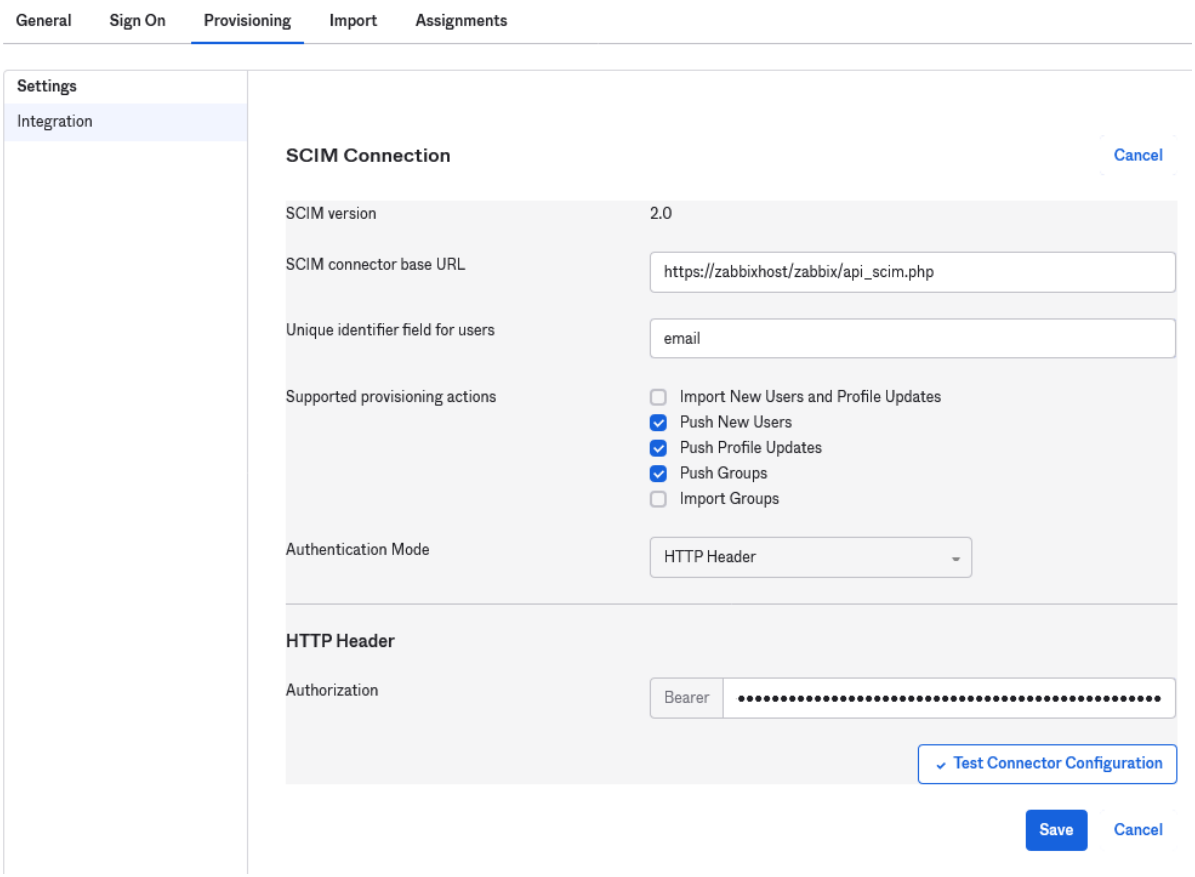
4. Clique no botão *Atualizar* para salvar essas configurações.

Provisionamento SCIM

1. Para ativar o provisionamento SCIM, acesse *General* > *App Settings* do aplicativo no Okta.

Marque a caixa de seleção *Enable SCIM provisioning*. Como resultado, uma nova aba *Provisioning* será exibida.

2. Acesse a aba "Provisioning" para configurar uma conexão SCIM: - Em *SCIM connector base URL*, especifique o caminho para o frontend do Zabbix e acrescente `api_scim.php`, 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, consulte Authorization header forwarding.

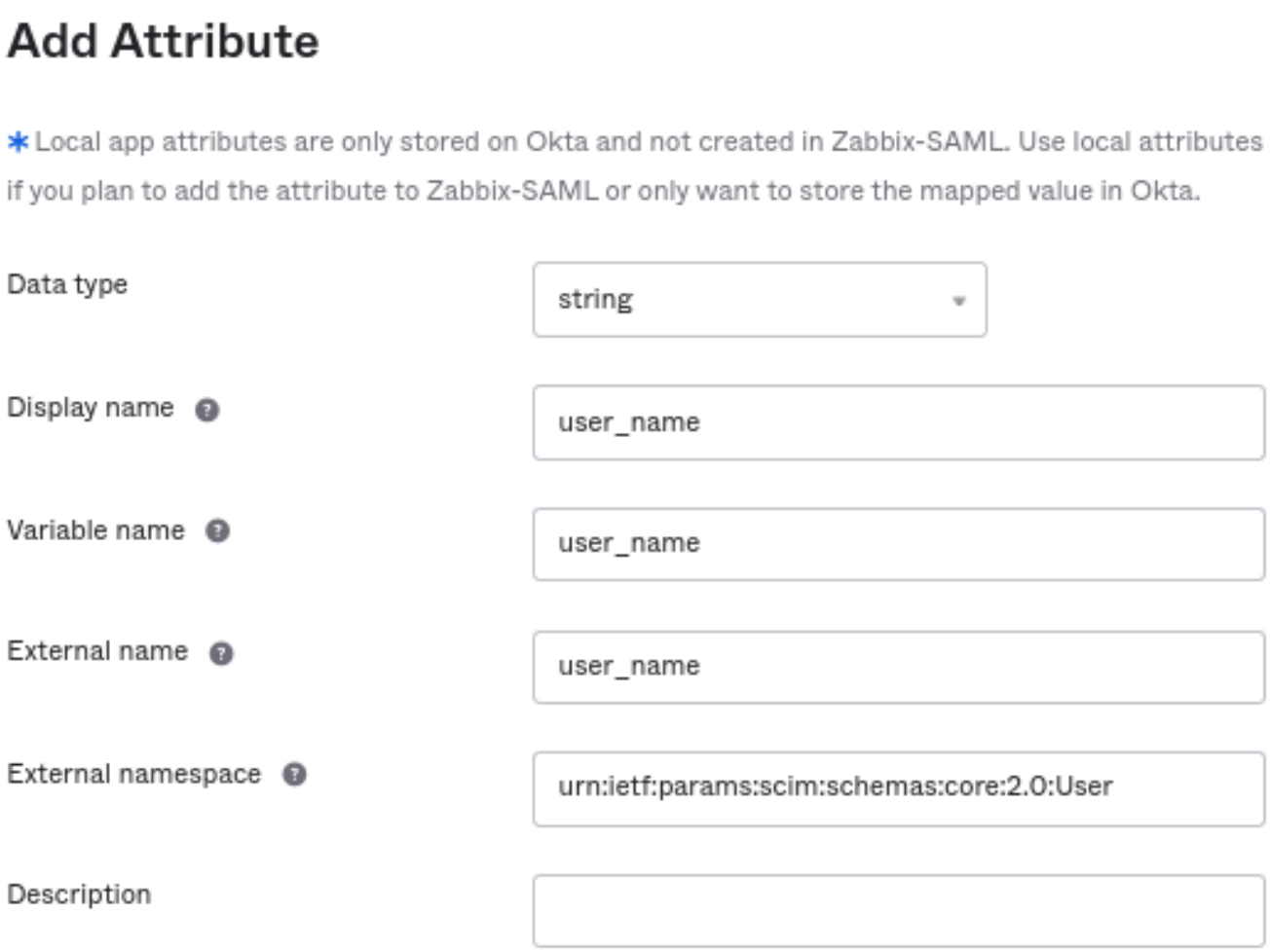
3. Clique em *Test Connector Configuration* para testar a conexão. Se tudo estiver correto, uma mensagem de sucesso será exibida.

4. Em *Provisioning* > *To App*, certifique-se de marcar as seguintes caixas de seleção: - Create Users - Update User Attributes - Deactivate Users

Isso garantirá que esses tipos de solicitação sejam enviados ao Zabbix.

5. Certifique-se de que todos os atributos definidos no SAML também estejam definidos no SCIM. Você pode acessar o editor de perfil do aplicativo em *Provisioning* > *To App*, clicando em *Go to Profile Editor*.

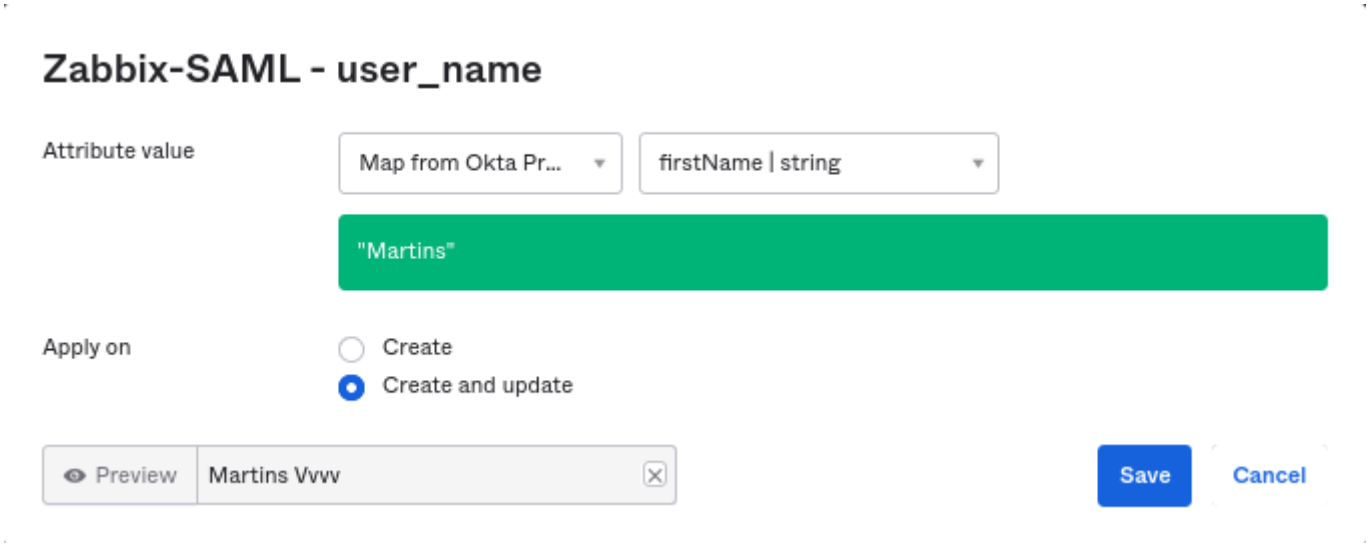
Clique em *Add Attribute*. Preencha os valores de *Display name*, *Variable name* e *External name* com o nome do atributo SAML, por exemplo, `user_name`.



External namespace deve ser igual ao esquema de usuário: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Acesse *Provisioning* > *To App* > *Attribute Mappings* do seu aplicativo. Clique em *Show Unmapped Attributes* na parte inferior. Os atributos recém-adicionados serão exibidos.

7. Mapeie cada atributo adicionado.



8. Adicione usuários na aba ”Assignments”. Primeiro, os usuários precisam ser adicionados em *Directory* > *People*. Todas essas atribuições serão enviadas como solicitações ao 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 sempre que alguma alteração é feita.

Configuração de 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 application

1. Faça login na sua conta no OneLogin. Para fins de teste, você pode criar uma conta gratuita de desenvolvedor no OneLogin.

2. Na interface web do OneLogin, navegue até *Applications* > *Applications*.

3. Clique em ”Add App” e pesquise o app apropriado. As diretrizes nesta página são baseadas no exemplo de app *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

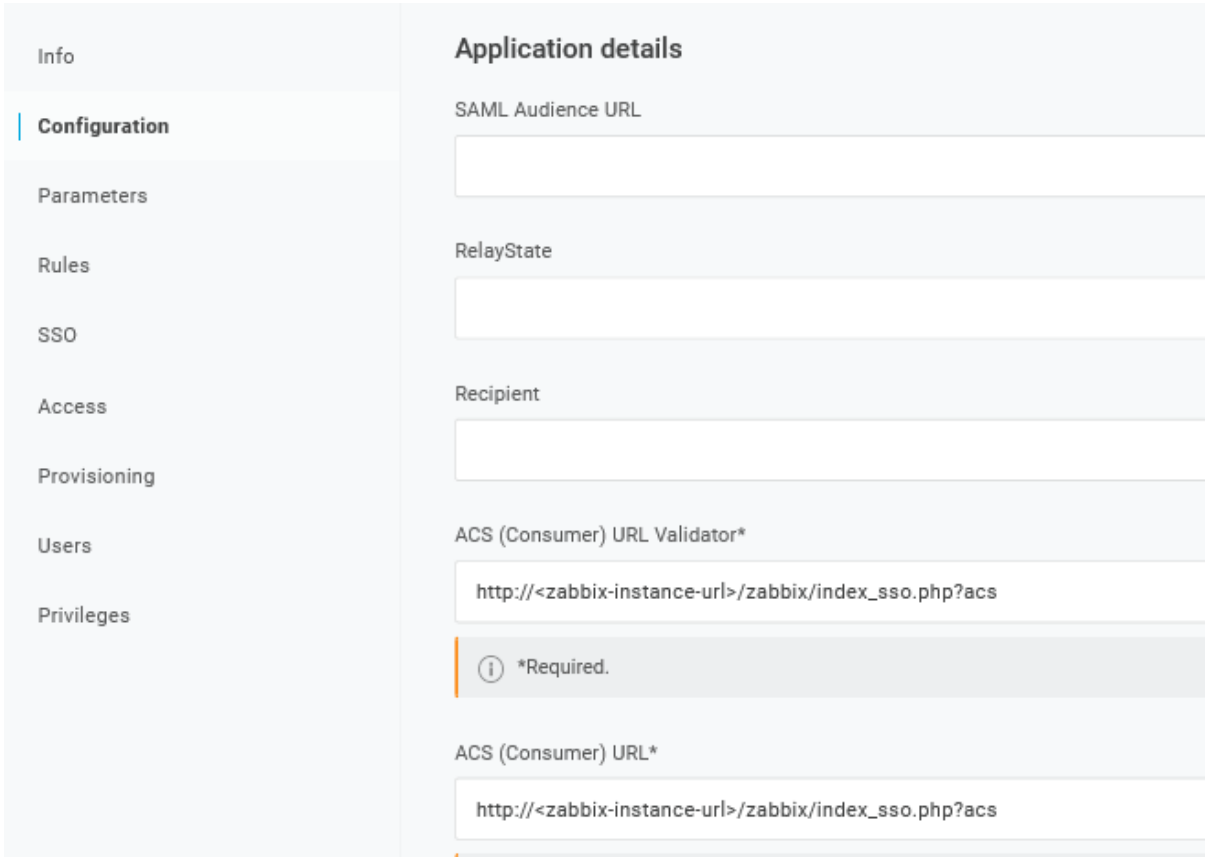
4. Para começar, talvez você queira personalizar o nome de exibição do seu app. Você também pode querer adicionar o ícone e os detalhes do app. Depois disso, clique em *Save*.

Configurando o provisionamento SSO/SCIM

1. Em *Configuration* > *Application details*, defina o endpoint de logon único do Zabbix `http://<zabbix-instance-url>/zabbix/index_s` como valor dos seguintes campos:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Observe o uso de ”http”, e não ”https”, para que o parâmetro `acs` não seja removido da solicitação.



Também é possível usar "https". Para usar essa opção com o Zabbix, adicione a seguinte linha ao arquivo de configuração do frontend (`zabbix.conf.php`):

```
$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 enviar ao Zabbix via SCIM, como `user_name`, `user_lastname`, `user_email` e `user_mobile`:

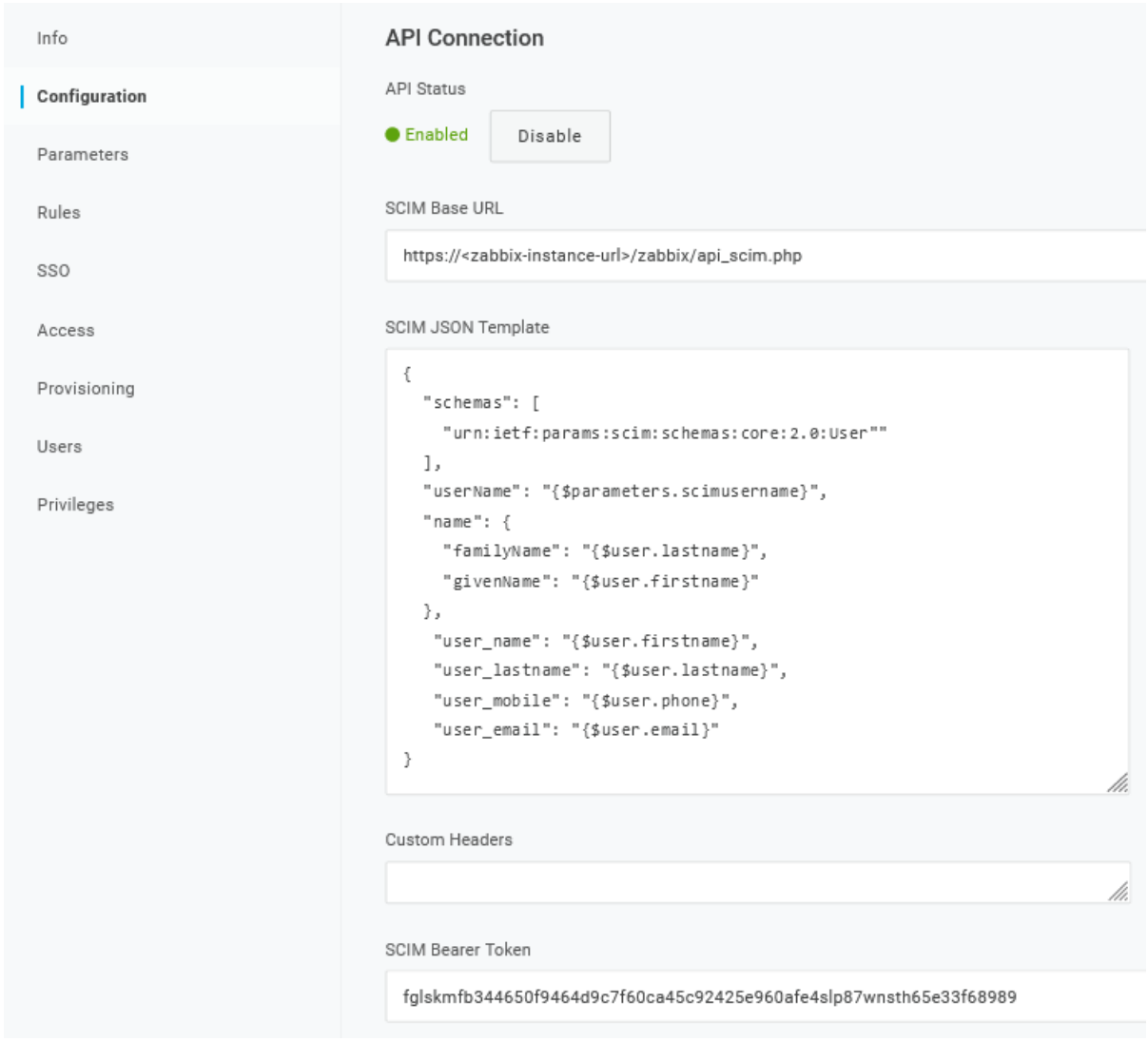
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Os nomes dos atributos são arbitrários. É possível usar nomes de atributos diferentes; no entanto, é necessário que eles correspondam ao respectivo valor de campo nas configurações SAML do Zabbix.

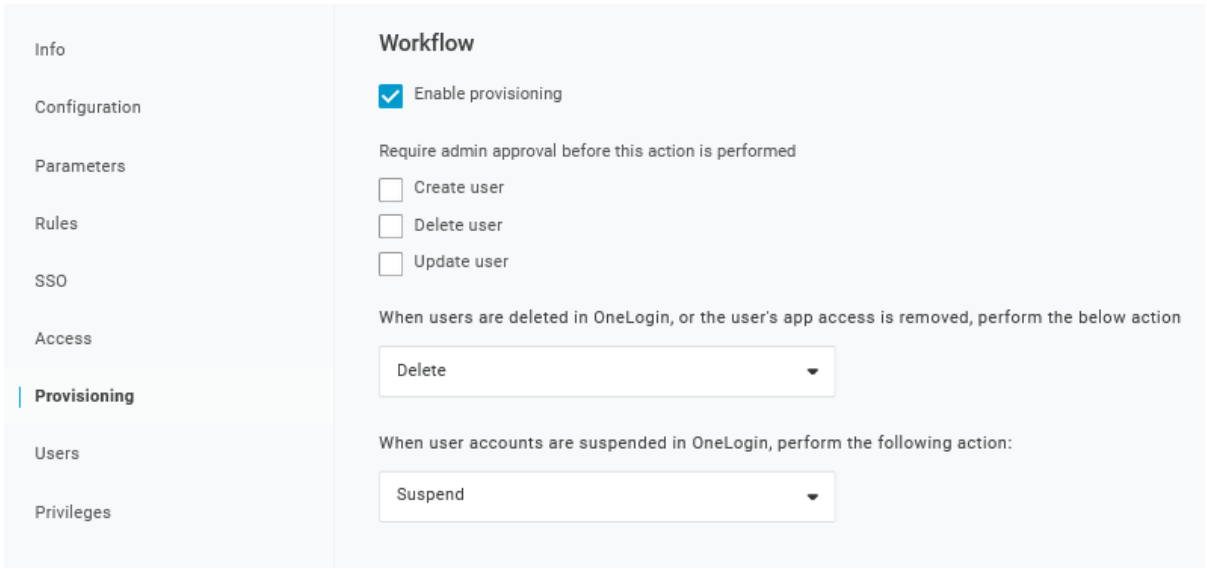
Observe que, para que o provisionamento de usuários funcione, o OneLogin precisa receber na resposta um atributo 'name' com 'givenName' e 'familyName', mesmo que isso não tenha sido exigido pelo provedor de serviços. Portanto, é necessário especificar isso no schema, na seção de configuração da aplicação.

- *SCIM Bearer Token*: insira um token de API do Zabbix com permissões de Super admin.

Clique em *Enable* para ativar a conexão.

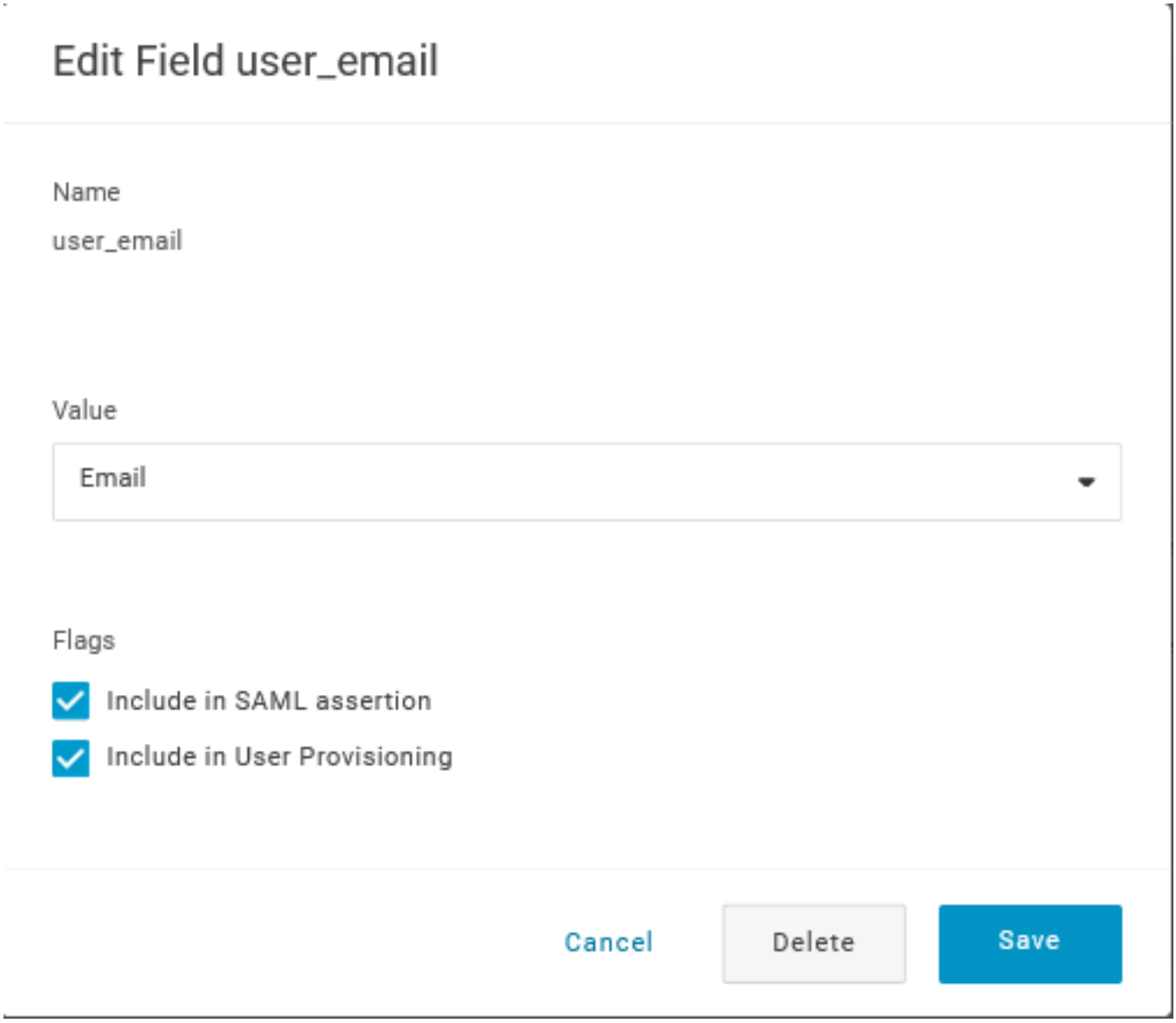


3. Na página *Provisioning*, ative a opção Provisioning:



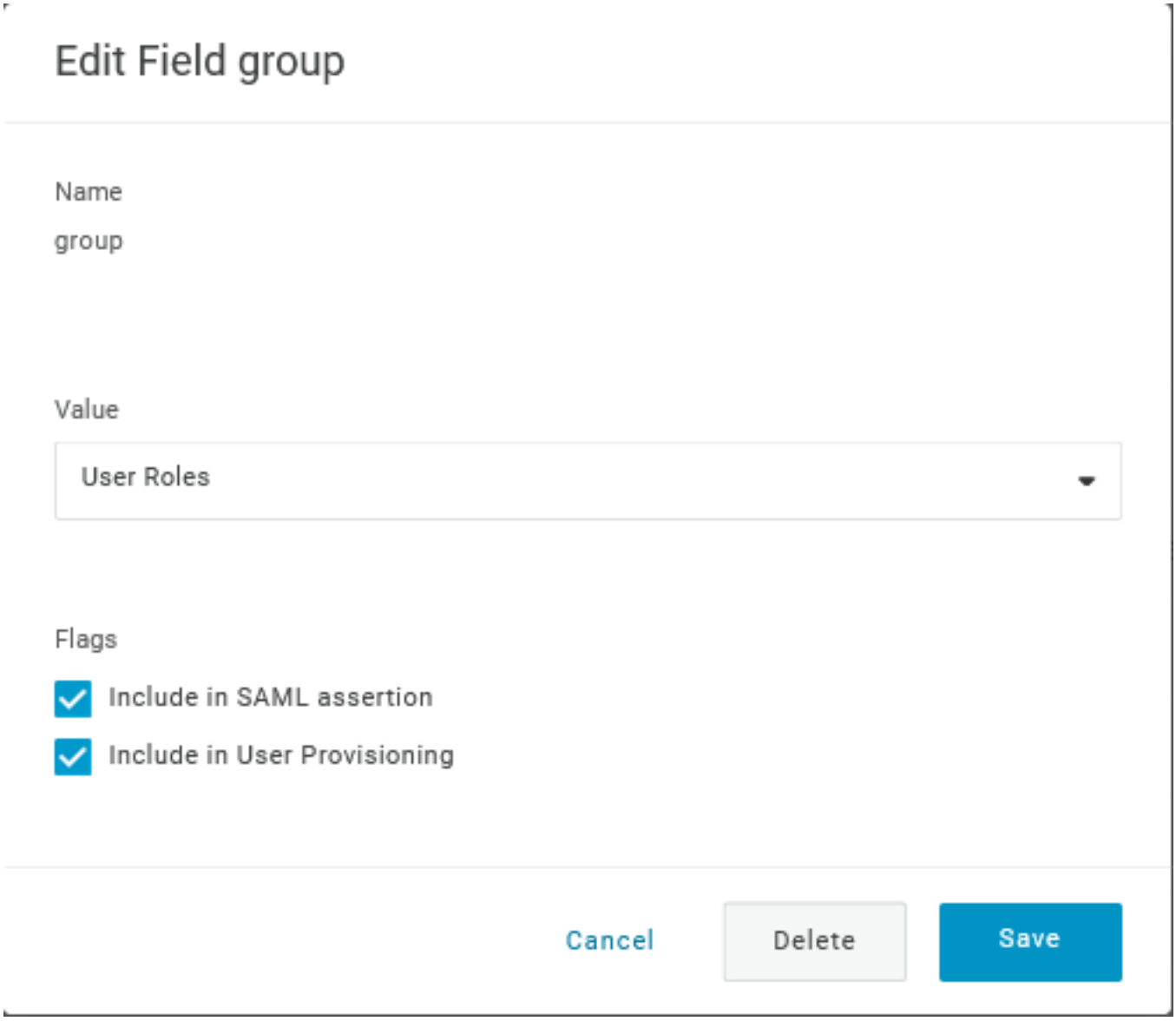
4. A página *Parameters* contém uma lista de parâmetros padrão:

- Certifique-se de que 'scimusername' corresponda ao valor de login do usuário no OneLogin (por exemplo, o 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`:

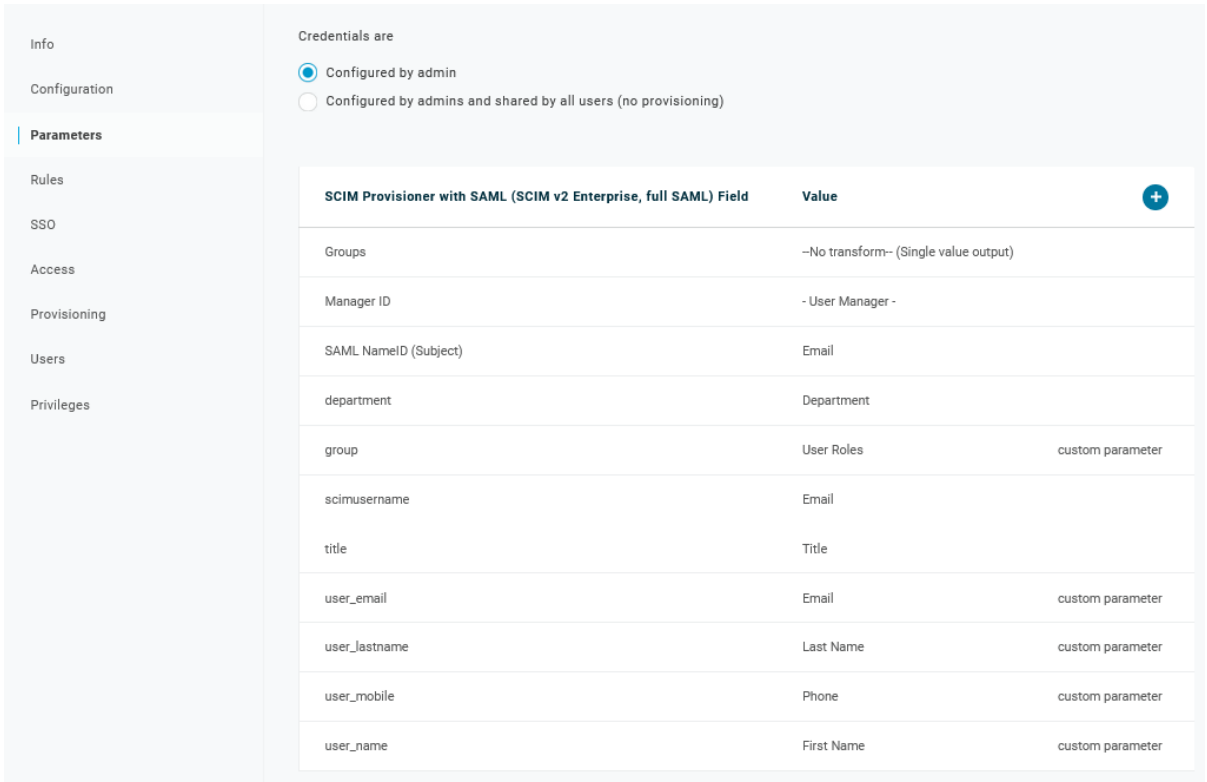


Ao adicionar um parâmetro, certifique-se de marcar as opções *Include in SAML assertion* e *Include in User Provisioning*.

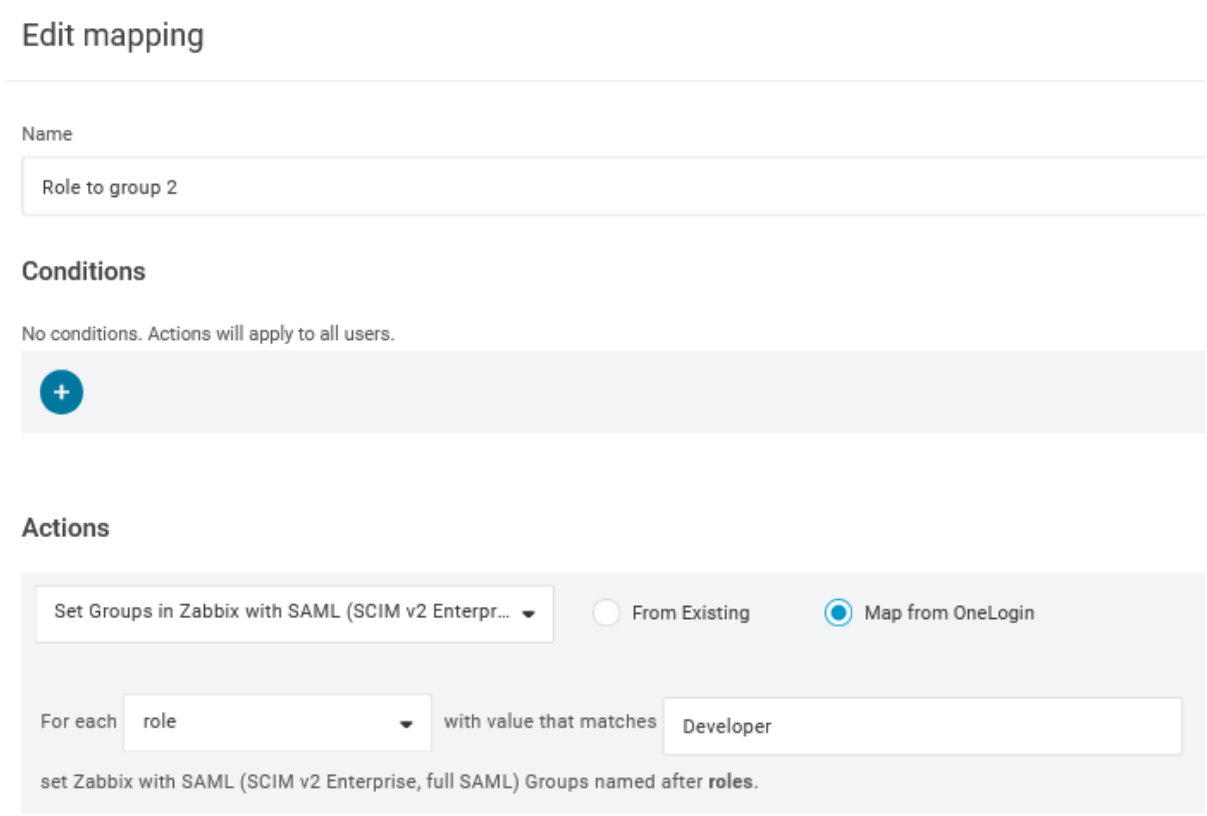
- Adicione um parâmetro 'group' que corresponda às funções de usuário no OneLogin. As funções de usuário serão enviadas como uma string, separadas por ponto e vírgula `;`. As funções de usuário do OneLogin serão usadas para criar grupos de usuários no Zabbix:



Verifique a lista de parâmetros:



5. Na página *Rules*, crie mapeamentos de funções de usuário para o parâmetro Groups padrão.



Você pode usar uma expressão regular para enviar funções específicas como grupos. Os nomes das funções não devem conter `;`, pois o OneLogin o utiliza como separador ao enviar um atributo com várias funções.

6. Baixe o certificado do IdP. Defina as permissões como 644 executando:

```
chmod 644 idp.crt
```

Configuração do Zabbix

1. No Zabbix, acesse as configurações do SAML e preencha as opções de configuração com base na configuração do OneLogin:



Campo do Zabbix	Campo de configuração no OneLogin	Valor de exemplo
ID da entidade do IdP	URL do emissor (consulte a aba SSO da sua aplicação no OneLogin)	

Campo do Zabbix	Campo de configuração no OneLogin	Valor de exemplo
URL do serviço SSO	Endpoint SAML 2.0 (HTTP) (consulte a aba SSO da sua aplicação no OneLogin)	
URL do serviço SLO	Endpoint SLO (HTTP) (consulte a aba SSO da sua aplicação no OneLogin)	
Atributo de nome de usuário	Parâmetro personalizado	`user_email`
Atributo de nome do grupo	Parâmetro personalizado	`group`
Atributo de nome	Parâmetro personalizado	`user_name`
Atributo de sobrenome	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 no arquivo de configuração do frontend (`zabbix.conf.php`), você poderá colar o texto do certificado ou carregar o arquivo do certificado no frontend durante a configuração do SAML — não são necessários arquivos no sistema de arquivos.

Se `$SSO['CERT_STORAGE'] = 'file'` estiver definido no arquivo de configuração, o certificado deverá estar disponível no sistema de arquivos (por padrão, em `ui/conf/certs` ou no caminho configurado no arquivo de configuração), e o frontend não armazenará os certificados no banco de dados.
Observe que, se `$SSO['CERT_STORAGE']` não estiver definido ou estiver comentado, o armazenamento em arquivo será considerado padrão, e os certificados serão lidos de `ui/conf/certs`.

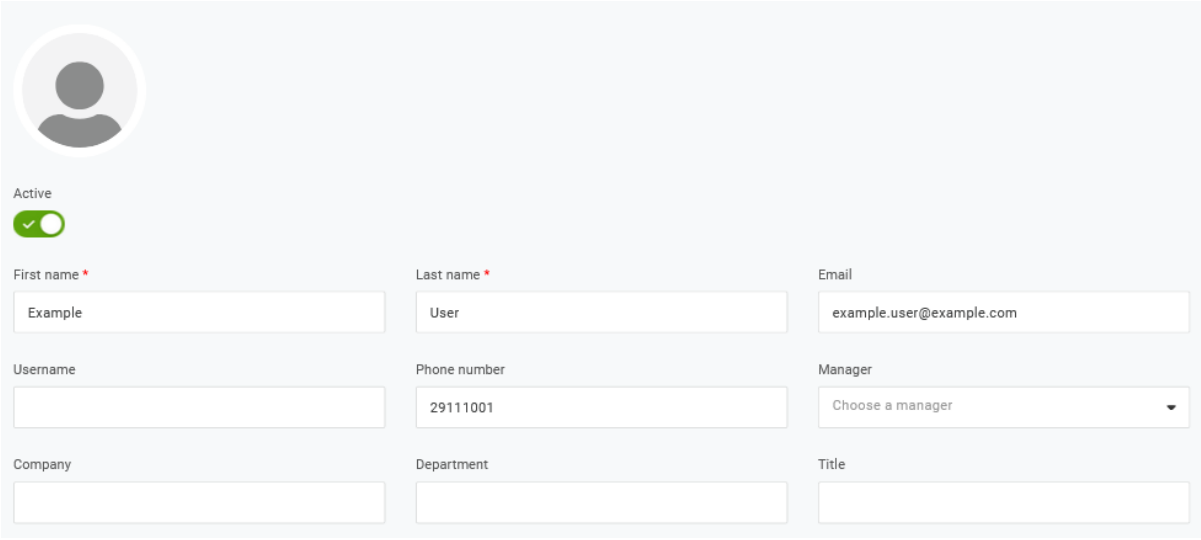
No OneLogin, você pode acessar o download do certificado em *Applications* > *SSO* > clique em *View details* no certificado atual.

3. Clique no botão *Atualizar* para salvar essas configurações.

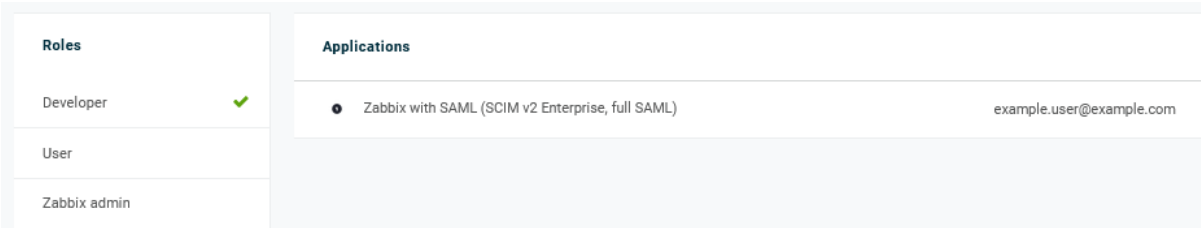
Provisionamento de usuários SCIM

Com o provisionamento de usuários habilitado, agora é possível adicionar/atualizar usuários e suas funções no OneLogin e tê-los provisionados imediatamente no Zabbix.

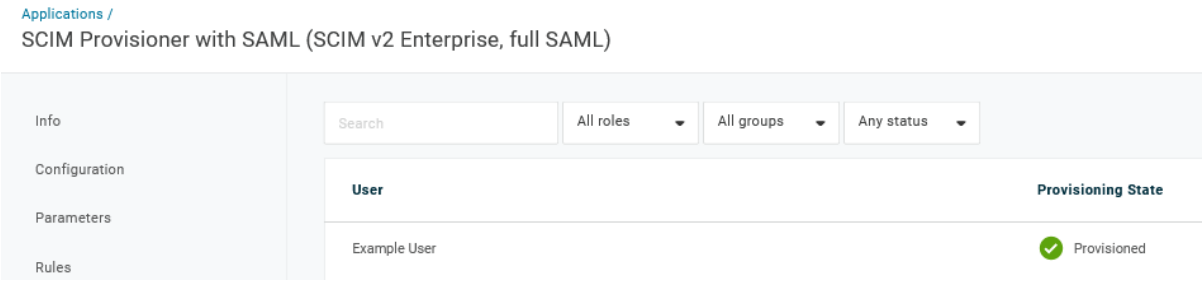
Por exemplo, você pode criar um novo usuário:



Adicione-o a uma função de usuário e à aplicação que fará o provisionamento do usuário:



Ao salvar o usuário, ele será provisionado no Zabbix. Em *Application* > *Users*, você pode verificar o status de provisionamento dos usuários atuais da aplicação:



Se o provisionamento for bem-sucedido, 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

Configurando relatórios agendados

Visão geral

Esta seção fornece instruções sobre como instalar o serviço web do Zabbix e configurar o Zabbix para permitir a geração de relatórios agendados.

Instalação

Um novo processo do serviço web do Zabbix e o navegador Google Chrome devem ser instalados para permitir a geração de relatórios agendados. O serviço web pode ser instalado na mesma máquina onde o 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 Zabbix web service a partir do código-fonte, consulte Installing Zabbix web service.

Após a instalação, execute `zabbix_web_service` na máquina onde o web service está instalado:

```
zabbix_web_service
```

Configuração

Para garantir a comunicação adequada entre todos os elementos envolvidos, certifique-se de que os parâmetros de configuração do server e configuração do frontend estejam configurados corretamente.

Zabbix server

Os seguintes parâmetros no arquivo de configuração do Zabbix server precisam ser atualizados: `WebServiceURL` e `StartReportWriters`.

WebServiceURL

Este parâmetro é necessário para habilitar a comunicação com o serviço web. A URL deve estar no formato `http[s]://host:port/report`.

- Por padrão, o serviço web escuta na porta 10053. Uma porta diferente pode ser especificada no arquivo de configuração do serviço web.
- Especificar o caminho `/report` é obrigatório (o caminho é fixo e não pode ser alterado).

Exemplo:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
É altamente recomendável configurar a criptografia entre o Zabbix server e o serviço web do Zabbix usando certificados. Por padrão, os dados transmitidos entre o Zabbix server e o serviço web do Zabbix 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 estiver definido ou for igual a 0, a geração de relatórios será desabilitada. Com base no número e na frequência dos relatórios necessários, é possível habilitar de 1 a 100 processos de geração de relatórios.

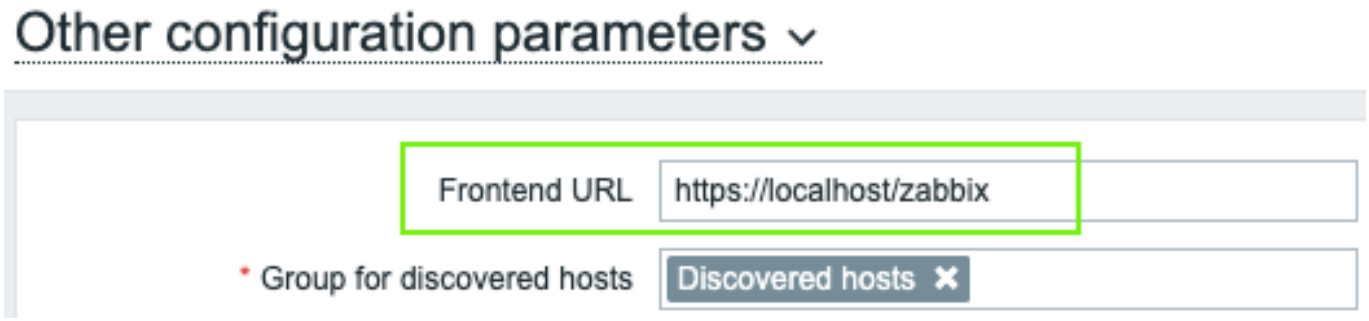
Exemplo:

```
StartReportWriters=3
```

Frontend do Zabbix

O parâmetro *Frontend URL* deve ser definido para habilitar a comunicação entre o frontend do Zabbix e o serviço web do Zabbix:

- Acesse a seção de menu do frontend *Administração* > *Geral* > *Outros*
- Especifique a URL completa da interface web do Zabbix no parâmetro *Frontend URL*



Note:
Após a conclusão do procedimento de configuração, você pode configurar e enviar um relatório de teste para garantir que tudo está funcionando corretamente.

Itens

Esta seção contém informações de referência e notas técnicas sobre items, incluindo detalhes de parâmetros para chaves de item específicas, detalhes específicos de plataforma e observações sobre configurações especiais de item.

Use a barra lateral para encontrar informações relevantes para o seu caso de uso.

Parâmetros de 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 relida a partir dele)
- **available** - memória disponível, calculada de forma diferente dependendo da plataforma (veja a tabela abaixo)
- **buffers** - cache para coisas como metadados do sistema de arquivos
- **cached** - cache para várias coisas
- **exec** - código executável, normalmente de um arquivo (programa)
- **file** - cache para conteúdos de arquivos acessados recentemente
- **free** - memória que está prontamente disponível para qualquer entidade que solicitar memória
- **inactive** - memória marcada como não utilizada
- **pavailable** - memória 'disponível' como porcentagem do 'total' (calculada como `available/total*100`)
- **pinned** - igual a 'wired'
- **pused** - memória 'usada' como porcentagem do 'total' (calculada como `used/total*100`)
- **shared** - memória que pode ser acessada simultaneamente por vários processos
- **slab** - quantidade total de memória usada pelo kernel para armazenar em cache 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 é 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, cached são consideradas usadas, porque armazenam alguma informação útil.
 * Ao mesmo tempo, memórias inativas, cached, free são consideradas disponíveis, porque esses tipos de memórias podem ser dadas instantaneamente a processos que solicitam mais memória.
Portanto, a memória inativa é usada e disponível simultaneamente. Por causa disso, o item *vm.memory.size[used]* é projetado apenas para fins informativos, enquanto *vm.memory.size[available]* é projetado para ser usado em triggers.

Veja também

1. Detalhes adicionais sobre o cálculo de memória em diferentes SO

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 de agent suportados no Windows

Item key	Grupo de usuários	
	Recomendado	Mínimo tecnicamente aceitável (a funcionalidade é limitada)
agent.hostname	Convidados	Convidados
agent.ping	Convidados	Convidados

Item key	Grupo de usuários	
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
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

Item key	User group	
	Recomendado	Mínimo tecnicamente aceitável (a funcionalidade é limitada)
eventlog	Leitores do 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

Item key	User group	
service.discovery	Convidados	Convidados
service.info	Convidados	Convidados
services	Convidados	Convidados
wmi.get	Administradores	Convidados
vm.vmemory.size	Convidados	Convidados

Codificação dos valores retornados

O server Zabbix 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. Nesses casos, quase todas as chaves do **zabbix** possíveis contêm um parâmetro adicional da chave do item - **<encoding>**. Esse 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, há 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 não UTF8, esse caractere e os seguintes serão descartados quando o banco de dados armazenar esse valor. Nenhuma mensagem de aviso será gravada no `zabbix_server.log`.
 Aplicável, no mínimo, à versão 5.1.61 do MySQL.

PostgreSQL

Se um valor contiver um caractere não ASCII em uma codificação não UTF8, isso levará a uma consulta SQL com falha (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) e os dados não serão armazenados. Uma mensagem de aviso apropriada será gravada em `zabbix_server.log`.
 Aplicável, no mínimo, à versão 9.1.3 do PostgreSQL.

Suporte a arquivos grandes

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 de um sistema no momento da compilação do Zabbix, mas é completamente desativado em um Solaris de 32 bits devido à sua incompatibilidade com procfs e swapctl.

Sensor

Visão geral

Cada chip de sensor recebe seu próprio diretório na árvore sysfs `/sys/devices`. Para encontrar todos os chips de sensores, é mais fácil seguir os links simbólicos de 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 do 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” (tensão), ”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 tensões que começam em 0). Se os arquivos não se referem a um elemento específico, eles têm um nome simples sem número.

As informações sobre os sensores disponíveis no host podem ser adquiridas usando as ferramentas **sensors-detect** e **sensors** (pacote lm-sensors). **Sensors-detect** ajuda a determinar quais módulos são necessários para os sensores disponíveis. Quando os módulos são carregados, o programa **sensors** pode ser usado para mostrar as leituras de todos os chips de sensores. A rotulagem das leituras dos sensores, usada por este programa, pode ser diferente do esquema de nomenclatura comum (`<type><number>_<item>`):

- se houver um arquivo chamado <type><number>_label, então o rótulo dentro deste arquivo será usado em vez do nome <type><number><item>;
- se não houver arquivo <type><number>_label, então o programa procura dentro do /etc/sensors.conf (pode ser também /etc/sensors3.conf, ou diferente) pela substituição do nome.

Essa rotulagem permite ao usuário determinar que tipo de hardware está sendo usado. Se não houver arquivo <type><number>_label nem rótulo dentro do arquivo de configuração, o tipo de hardware pode ser determinado pelo atributo name (hwmon*/device/name). Os nomes reais dos sensores, que o zabbix_agent aceita, podem ser obtidos executando o programa **sensors** com o parâmetro -u (sensors -u).

No programa **sensors**, os sensores disponíveis são separados pelo tipo de barramento (adaptador ISA, adaptador PCI, adaptador SPI, dispositivo virtual, interface ACPI, adaptador HID).

Obtendo identificadores de sensores

Os rótulos dos sensores, conforme impressos pelo comando sensors, nem sempre podem ser usados diretamente porque a rotulagem varia entre os fornecedores de chips. Por exemplo, a saída do sensors pode conter as seguintes linhas:

```
sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

Desses, apenas um nome de exibição corresponde diretamente a um identificador interno:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Tentar usar outros rótulos de exibição (como Vcore ou +12V) não funcionará:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Para encontrar o identificador interno do sensor que o Zabbix aceita, execute sensors -u. Na saída, o seguinte pode ser observado:

```
sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

Portanto, o rótulo Vcore corresponde ao identificador in1 e +12V corresponde a in4. De acordo com a especificação do kernel hwmon sysfs, essas são tensões nos pinos do chip e, de modo geral, podem precisar de escala.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Não apenas a voltagem (in), mas também leituras de corrente (curr), temperatura (temp) e velocidade do ventilador (fan) podem ser recuperadas pelo Zabbix.

Compatibilidade

Distribuições legadas

As leituras dos sensores são obtidas do diretório /proc/sys/dev/sensors:

- **device** - nome do dispositivo (se <mode> for usado, é uma expressão regular)
- **sensor** - nome do sensor (se <mode> for usado, é uma expressão regular)
- **mode** - valores possíveis: avg, max, min (se este parâmetro for omitido, device e sensor são tratados literalmente)

Exemplo de chave: `sensor[w83781d-i2c-0-2d,temp1]`.

Distribuições atuais

As leituras dos sensores são obtidas do diretório `/sys/class/hwmon`:

- **device** - nome do dispositivo (não é uma expressão regular). O nome do dispositivo pode ser o nome real do dispositivo (por exemplo, `0000:00:18.3`) ou o nome adquirido usando o programa de sensores (por exemplo, `k8temp-pci-00c3`). Cabe ao usuário escolher qual nome usar.
- **sensor** - nome do sensor (não é uma expressão regular).
- **mode** - valores possíveis: `avg`, `max`, `min` (se este parâmetro for omitido, o dispositivo e o sensor são tratados literalmente).

Exemplos de chaves:

`sensor[k8temp-pci-00c3,temp,max]` ou `sensor[0000:00:18.3,temp1]`.

`sensor[smsc47b397-isa-0880,in,avg]` ou `sensor[smsc47b397.2176,in1]`.

Notas sobre o parâmetro memtype em itens proc.mem

Visão geral

O parâmetro **memtype** é compatível com as plataformas Linux, AIX, FreeBSD e Solaris.

Três valores comuns de `memtype` são compatíveis com todas essas plataformas: `pmem`, `rss` e `vsize`. Além disso, alguns valores específicos de `memtype` são compatíveis com determinadas plataformas.

AIX

Consulte na tabela os valores compatíveis com o parâmetro `memtype` no AIX.

Valor compatível	Descrição	Origem na estrutura procentry64	Tenta ser compatível com
`vsize` [1]	Tamanho da memória virtual	`pi_size`	
`pmem`	Porcentagem da 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 de `ps gvw`
`dsize`	Tamanho dos dados	`pi_dsize`	
`tsize`	Tamanho do texto (código)	`pi_tsize`	Coluna TSIZ de `ps gvw`
`sdsize`	Tamanho dos dados da biblioteca compartilhada	`pi_sdsize`	
`drss`	Tamanho do conjunto de dados residente	`pi_drss`	
`trss`	Tamanho do conjunto de texto residente	`pi_trss`	

Notas para AIX:

1. Ao escolher os parâmetros para a chave de item `proc.mem[]` no AIX, tente especificar critérios restritos de seleção de processos. Caso contrário, existe o risco de que processos indesejados sejam contabilizados no resultado de `proc.mem[]`.

Exemplo:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Este exemplo mostra que especificar apenas o parâmetro da linha de comando (expressão regular a ser correspondida) faz com que o agent do Zabbix contabilize a si próprio, o que provavelmente não é o que você deseja.

2. Não use `ps -ef` para listar processos — ele mostra apenas processos que não são do kernel. Use `ps -Af` para ver todos os processos que serão vistos pelo Zabbix agent.
3. Vamos analisar o exemplo de `topasrec` para entender como o Zabbix agent `proc.mem[]` seleciona os processos.

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

`proc.mem[]` tem os seguintes argumentos:

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

onde: - O primeiro critério é um nome de processo (argumento <name>). No nosso exemplo, o Zabbix agent o verá como topasrec. Para corresponder, você precisa especificar topasrec ou deixá-lo vazio. - O segundo critério é um nome de usuário (argumento <user>). Para corresponder, você precisa especificar root ou deixá-lo vazio. - O terceiro critério usado na seleção do processo é um argumento <cmdline>. O Zabbix agent 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 esta string ou deixá-la vazia.

Os argumentos <mode> e <memtype> são aplicados após o uso dos três critérios mencionados acima.

FreeBSD

Consulte na tabela os valores compatíveis com o parâmetro memtype no FreeBSD.

Valor compatível	Descrição	Origem 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 da memória real	calculado a partir de rss	ps -o pmem
rss	Tamanho do conjunto residente	kp_eproc.e_vm.vm_rssize ou ki_rssize	ps -o rss
size [1]	Tamanho do processo (código + dados + pilha)	tsize + dsize + ssize	
tsize	Tamanho do texto (código)	kp_eproc.e_vm.vm_tsize ou ki_tsize	ps -o tsiz
dsize	Tamanho dos dados	kp_eproc.e_vm.vm_dsize ou ki_dsize	ps -o dsiz
ssize	Tamanho da pilha	kp_eproc.e_vm.vm_ssize ou ki_ssize	ps -o ssiz

Linux

Consulte na tabela os valores compatíveis com o parâmetro memtype no Linux.

Valor compatível	Descrição	Origem no arquivo /proc/<pid>/status
vsize [1]	Tamanho da memória virtual	VmSize
pmem	Porcentagem da 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	Tamanho máximo do conjunto residente	VmHWM
lck	Tamanho da memória bloqueada	VmLck
lib	Tamanho das bibliotecas compartilhadas	VmLib
peak	Tamanho máximo da memória virtual	VmPeak
pin	Tamanho das páginas fixadas	VmPin
pte	Tamanho das entradas da tabela de páginas	VmPTE
size	Tamanho dos segmentos de código + dados + pilha do processo	VmExe + VmData + VmStk
stk	Tamanho do segmento de pilha	VmStk
swap	Tamanho do espaço de swap utilizado	VmSwap

Observações para Linux:

1. Nem todos os valores de memtype são compatíveis com versões mais antigas do kernel do Linux. Por exemplo, os kernels Linux 2.4 não são compatíveis com os valores hwm, pin, peak, pte e swap.
2. Observamos que o monitoramento próprio do processo de verificação ativa do Zabbix agent com proc.mem[...,...,...,...,data] mostra um valor 4 kB maior que o informado pela linha VmData no arquivo /proc/<pid>/status do agent. No momento da medição própria, o segmento de dados do agent aumenta em 4 kB e depois retorna ao tamanho anterior.

Solaris

Consulte na tabela os valores compatíveis com o parâmetro memtype no Solaris.

Valor compatível	Descrição	Origem na estrutura psinfo	Tenta ser compatível com
vsize [1]	Tamanho da imagem do processo	pr_size	ps -o vsz
pmem	Porcentagem de memória real	pr_pctmem	ps -o pmem
rss	Tamanho do conjunto residente Pode ser subestimado (consulte a descrição de rss em man ps).	pr_rssize	ps -o rss

Notas de rodapé

[1] Valor padrão.

Notas sobre a seleção de processos em itens proc.mem e proc.num

Processos que modificam sua linha de comando

Alguns programas usam a modificação de sua linha de comando como um método para exibir sua atividade atual. Um usuário pode ver a atividade executando os comandos ps e top. Exemplos de tais programas incluem *PostgreSQL*, *Sendmail*, *Zabbix*.

Vamos ver um exemplo no Linux. Suponha que queremos monitorar o número de processos do agent Zabbix.

O comando ps mostra os processos de interesse como

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

Selecionar processos por nome e usuário resolve o problema:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Agora vamos renomear o executável zabbix_agentd para zabbix_agentd_30 e reiniciá-lo.

O ps agora mostra

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

Agora, selecionar processos por nome e usuário produz um resultado incorreto:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Por que uma simples renomeação do executável para um nome mais longo levou a um resultado tão diferente?

O agent Zabbix começa verificando o nome do processo. O arquivo /proc/<pid>/status é aberto e a linha Name é verificada. No nosso caso, as linhas Name são:

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
```

```
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

O nome do processo no arquivo `status` é truncado para 15 caracteres.

Um resultado semelhante pode ser visto com o comando `ps`:

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

Obviamente, isso não é igual ao valor do parâmetro `name` do nosso `proc.num[]`, `zabbix_agentd_30`. Não conseguindo corresponder ao nome do processo do arquivo `status`, o agent Zabbix recorre ao arquivo `/proc/<pid>/cmdline`.

Como o agent vê o arquivo "cmdline" pode ser ilustrado executando um comando

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

Os arquivos `/proc/<pid>/cmdline` em nosso caso contêm bytes nulos invisíveis e não imprimíveis, usados para terminar strings na linguagem *C*. Os bytes nulos são mostrados como "<NUL>" neste exemplo.

O agent Zabbix verifica o "cmdline" para o processo principal e pega um `zabbix_agentd_30`, que corresponde ao valor do nosso parâmetro `name` `zabbix_agentd_30`. Portanto, o processo principal é contado pelo item `proc.num[zabbix_agentd_30,zabbix]`.

Ao verificar o próximo processo, o agent pega `zabbix_agentd_30: collector [idle 1 sec]` do arquivo `cmdline` e ele não corresponde ao nosso parâmetro `name` `zabbix_agentd_30`. Assim, apenas o processo principal que não modifica sua linha de comando é contado. Outros processos do agent modificam sua linha de comando e são ignorados.

Este exemplo mostra que o parâmetro `name` não pode ser usado em `proc.mem[]` e `proc.num[]` para selecionar processos neste caso.

Note:
Para o item `proc.get[]`, quando o agent Zabbix verifica o "cmdline" para o nome do processo, ele usará apenas parte do nome começando a partir da última barra 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.

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.

Parâmetros de 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 do processo pai	ppid: PID do processo 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 da 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 bloqueada

mode=process	mode=thread	mode=summary
pmem: porcentagem da memória real	cputime_user: tempo total de CPU gasto no modo de usuário (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)	pin: tamanho das páginas fixadas
rss: tamanho do conjunto residente	cputime_system: tempo total de CPU gasto no modo de sistema (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)	pte: tamanho das entradas da tabela de páginas
data: tamanho do segmento de dados	state: estado da thread	size: tamanho dos segmentos de código + dados + pilha do processo
exe: tamanho do segmento de código	ctx_switches: número de trocas de contexto	stk: tamanho do segmento de pilha
hwm: tamanho máximo do conjunto residente	page_faults: número de falhas de página	swap: tamanho do espaço de swap utilizado
lck: tamanho da memória bloqueada		cputime_user: tempo total de CPU gasto no modo de usuário (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)
lib: tamanho das bibliotecas compartilhadas		cputime_system: tempo total de CPU gasto no modo de sistema (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)
peak: tamanho máximo da memória virtual		ctx_switches: número de trocas de contexto
pin: tamanho das páginas fixadas		threads: número de threads
pte: tamanho das entradas da tabela de páginas		page_faults: número de falhas de página
size: tamanho dos segmentos de código + dados + pilha do processo		pss: memória do conjunto proporcional
stk: tamanho do segmento de pilha		
swap: tamanho do espaço de swap utilizado		
cputime_user: tempo total de CPU gasto no modo de usuário (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)		
cputime_system: tempo total de CPU gasto no modo de sistema (o valor é expresso em ticks de relógio — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter os segundos)		
state: estado do processo (obtido de forma transparente do procfs, formato longo)		
ctx_switches: número de trocas de contexto		
threads: número de threads		
page_faults: número de falhas de página		

mode=process	mode=thread	mode=summary
pss: memória do conjunto proporcional		

Sistemas operacionais baseados em BSD

Os seguintes parâmetros de processo são retornados no FreeBSD, OpenBSD e NetBSD para cada modo:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome do processo
ppid: PID do processo pai	ppid: PID do processo pai	processes: número de processos
jid: ID da jail (apenas FreeBSD)	jid: ID da jail (apenas FreeBSD)	vsize: tamanho da memória virtual
jname: nome da jail (apenas FreeBSD)	jname: nome da jail (apenas FreeBSD)	pmem: porcentagem de memória real (apenas FreeBSD)
name: nome do processo	name: nome do processo	rss: tamanho do conjunto residente
cmdline: comando com argumentos	user: usuário (real) sob o qual o processo é executado	size: tamanho do processo (código + dados + pilha)
user: usuário (real) sob o qual o processo é executado	group: grupo (real) sob o qual o processo é executado	tsize: tamanho do texto (código)
group: grupo (real) sob o qual o processo é executado	uid: ID do usuário	dsize: tamanho dos dados
uid: ID do usuário	gid: ID do grupo sob o qual o processo é executado	ssize: tamanho da pilha
gid: ID do grupo sob o qual o processo é executado	tid: ID da thread	cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)
vsize: tamanho da memória virtual	tname: nome da thread	cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)
pmem: porcentagem de memória real (apenas FreeBSD)	cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)	ctx_switches: número de trocas de contexto
rss: tamanho do conjunto residente	cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)	threads: número de threads (não suportado para NetBSD)
size: tamanho do processo (código + dados + pilha)	state: estado da thread	stk: tamanho do segmento de pilha
tsize: tamanho do texto (código)	ctx_switches: número de trocas de contexto	page_faults: número de falhas de página
dsize: tamanho dos dados	io_read_op: número de vezes que o sistema teve que realizar entrada	fds: número de descritores de arquivos (apenas OpenBSD)
ssize: tamanho da pilha	io_write_op: número de vezes que o sistema teve que realizar saída	swap: tamanho do espaço de swap usado
cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)		io_read_op: número de vezes que o sistema teve que realizar entrada
cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)		io_write_op: número de vezes que o sistema teve que realizar saída
state: estado do processo (espera de disco/executando/esperando/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)		

mode=process	mode=thread	mode=summary
swap: tamanho do espaço de swap usado io_read_op: número de vezes que o sistema teve que realizar entrada io_write_op: número de vezes que o sistema teve que realizar saída		

Windows

Os seguintes parâmetros de processo são retornados no Windows para cada modo:

mode=process	mode=thread	mode=summary
pid: PID	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: tempo total de CPU gasto no modo usuário (relatado em milissegundos)
vmsize: tamanho da memória virtual	tid: ID da thread	cputime_system: tempo total de CPU gasto no modo sistema (relatado em milissegundos)
wkset: tamanho do conjunto de trabalho do processo		threads: número de threads
cputime_user: tempo total de CPU gasto no modo usuário (relatado em milissegundos)		page_faults: número de falhas de página
cputime_system: tempo total de CPU gasto no modo sistema (relatado em milissegundos)		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
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		

Configurações de interface de host inacessível/indisponível

Visão geral

Vários parâmetros de configuração definem como o server Zabbix deve se comportar quando uma verificação de agent (Zabbix, SNMP, IPMI, JMX) falha e uma interface do host se torna inacessível.

Interface inacessível

Uma interface de host é tratada como inacessível após uma verificação com falha (erro de rede, timeout) por agentes Zabbix, SNMP, IPMI ou JMX. Desde o Zabbix 6.2.0, verificações ativas do Zabbix agent também afetam a disponibilidade da interface.

Quando as verificações ativas ficam indisponíveis, elas contribuem para o status geral de disponibilidade da interface do agent.

A partir do momento em que uma interface se torna inacessível, **UnreachableDelay** define com que frequência ela é verificada novamente usando um dos items (incluindo regras LLD). Essas novas verificações são executadas por pollers de inacessibilidade (ou pollers IPMI para verificações IPMI). Por padrão, o intervalo entre verificações consecutivas de conectividade é de 15 segundos.

> **Attention:**
> Verificações executadas por pollers assíncronos não são movidas para pollers de inacessibilidade.

No log do server Zabbix, a inacessibilidade é indicada por mensagens como:

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

As mensagens de log especificam o item exato que falhou e seu tipo (Zabbix agent).

> **Note:**
> O parâmetro *Timeout* também afeta o quão cedo uma interface é verificada novamente durante a inacessibilidade. Se Timeout estiver definido como 20 segundos e UnreachableDelay como 30 segundos, a próxima verificação ocorrerá 50 segundos após a primeira tentativa.

O parâmetro **UnreachablePeriod** define a duração total do período de inacessibilidade. Por padrão, UnreachablePeriod é 45 segundos. Esse valor deve ser várias vezes maior que UnreachableDelay para garantir que uma interface seja verificada novamente várias vezes antes de ser marcada como indisponível.

Um item interno, zabbix[host,active_agent,available], permite monitorar a disponibilidade de verificações ativas em cenários de inacessibilidade.

Alternando a interface de volta para disponível

Quando o período de inacessibilidade termina, a interface é 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.

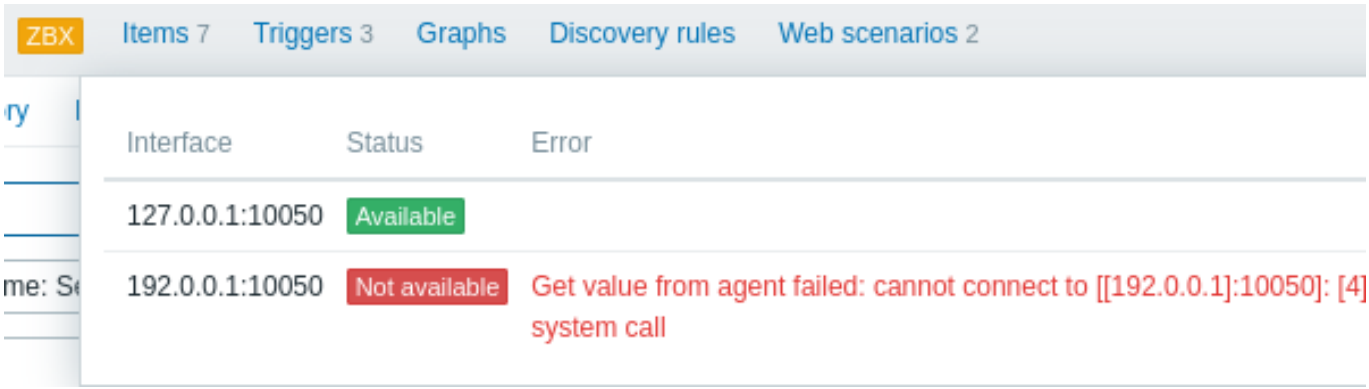
Interface indisponível

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
```

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 provedores de suporte/serviços possam monitorar remotamente os Zabbix servers/proxies de seus clientes ou, em organizações em que 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 hierárquico.

As estatísticas internas do Zabbix são expostas a um conjunto configurável de endereços listados no novo parâmetro ’StatsAllowedIP’ server/proxy. As solicitações serão aceitas somente desses endereços.

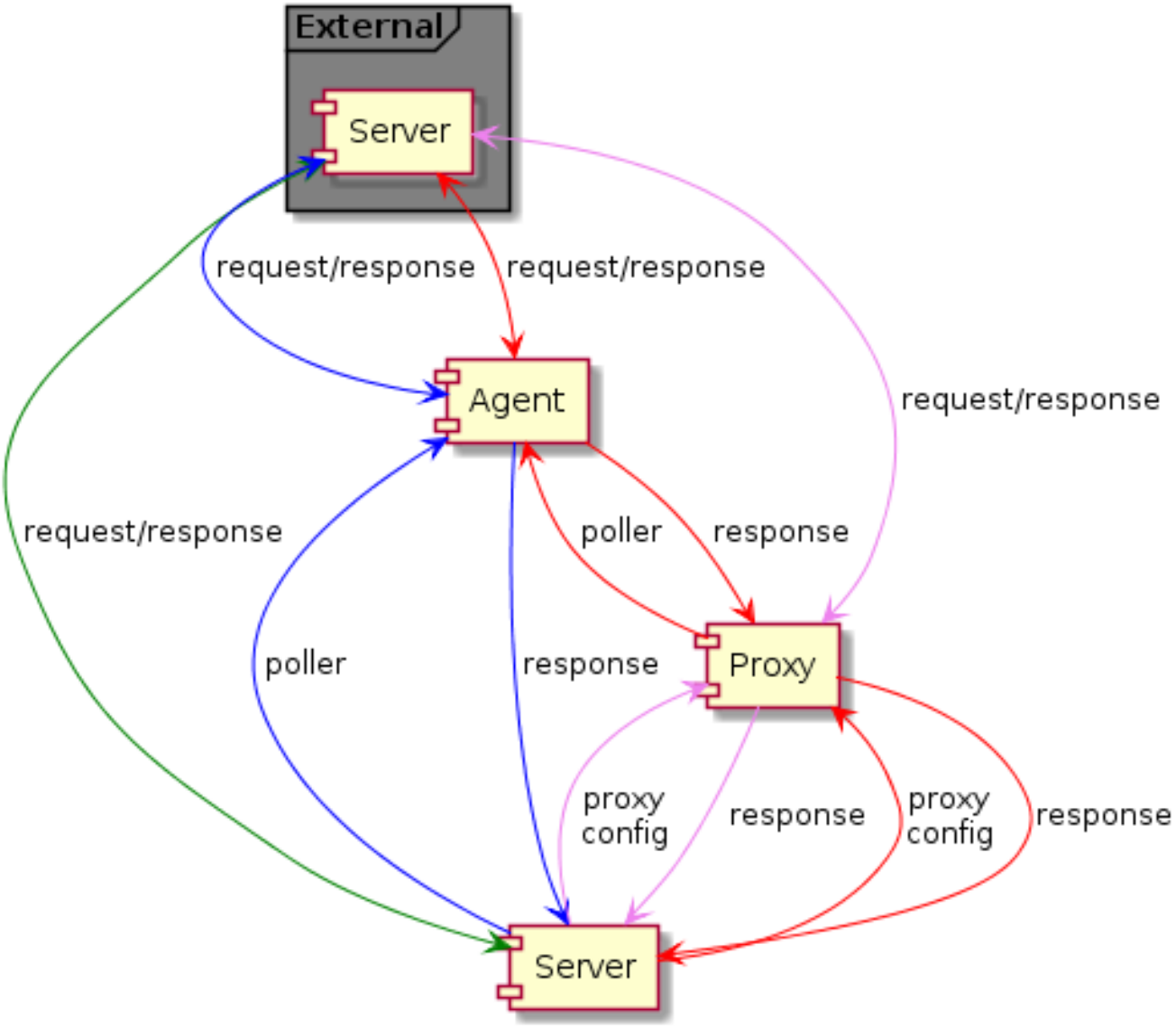
Items

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

Há também dois outros itens que permitem consultar remotamente, especificamente, as estatísticas da fila interna em outra

instância do Zabbix:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` item interno - para consultas diretas à fila interna do Zabbix server/proxy remoto
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` item do agent - para consultas à fila interna baseadas em agent no Zabbix server/proxy remoto

Consulte também: Itens internos, Itens do Zabbix agent

Configurando Kerberos com Zabbix

Visão geral

A autenticação Kerberos pode ser usada no monitoramento da web e em items HTTP no Zabbix.

Esta página descreve um exemplo de configuração do Kerberos para o servidor Zabbix realizar o monitoramento da web de `www.example.com` com um principal Kerberos para o processo Zabbix no Debian/Ubuntu.

Configuração

1. Instale o KDC e os utilitários do cliente:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Durante a configuração do pacote, responda aos prompts, por exemplo:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. Mapeie um nome de host amigável (opcional, para testes locais).

Edite /etc/hosts e adicione uma entrada para o DC e o webserver caso você não tenha DNS:

```
sudo vi /etc/hosts
```

Exemplo de linha que você pode adicionar:

```
192.168.1.100  dc01.example.com dc01
```

3. Configure o cliente Kerberos e o realm do KDC:

```
sudo vi /etc/krb5.conf
```

Exemplo de configurações:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

Se você planeja usar `.localdomain` ou outros nomes não públicos, adicione mapeamentos explícitos de domínio→realm para que o mapeamento de nome de host→realm funcione. Incompatibilidades aqui causam erros `Server not found in Kerberos database`.

4. Inicialize o banco de dados do Kerberos (uma única vez, no host do KDC). Defina uma senha mestre segura quando solicitado:

```
sudo krb5_newrealm
```

5. Crie o principal `HTTP/host.fqdn@REALM` usando exatamente o nome de host que os clientes usarão; prefira letras minúsculas (por exemplo, `HTTP/web.example.com@EXAMPLE.COM`). Uma incompatibilidade de maiúsculas/minúsculas ou de nome causa `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Dentro de `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM     # principal administrativo
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Mova o keytab para o host web (ou mantenha-o local se for a mesma máquina) e defina permissões que possam ser usadas pelo Apache:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verificar
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Instale e habilite o módulo GSSAPI do Apache:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Nem todas as versões do mod_auth_gssapi são compatíveis com todas as diretivas `Gssapi*`. Se o Apache falhar com `Invalid command 'GssapiCredStore'`, remova a diretiva não compatível ou atualize o módulo.

7. Configure um VirtualHost (ajuste `DocumentRoot` / o caminho para a interface do Zabbix):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Dentro de `zabbix.conf`:

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Reinicie o Apache:

```
sudo systemctl restart apache2
```

8. Habilite/inicie os serviços do KDC e verifique as portas em escuta (host do KDC):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # ou: sudo netstat -tnlp | grep :80
```

9. Obtenha um TGT para teste (execute como o usuário que usará o ticket).

Você deverá ver `krbtgt/EXAMPLE.COM@EXAMPLE.COM` na lista de tickets. Execute `kinit` como o mesmo usuário do sistema

operacional que precisa do ticket (por exemplo, zabbix para verificações web ou www-data/Apache para testes de SSO do navegador interativo). Os tickets emitidos para um usuário diferente do sistema operacional não ficarão visíveis, a menos que KRB5CCNAME e as permissões sejam ajustados.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Teste a troca SPNEGO com curl (a partir de um cliente com um TGT válido). Um 200 OK (ou um redirecionamento para o aplicativo) indica que o SPNEGO foi bem-sucedido:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Opcionalmente, se a interface do Zabbix deverá aceitar logins autenticados por HTTP, habilite a autenticação HTTP no frontend do Zabbix.

No arquivo de configuração do frontend (zabbix.conf.php), defina:

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Em seguida, configure a autenticação HTTP na interface web.

12. Configuração do navegador (o Firefox é usado como exemplo): defina network.negotiate-auth.trusted-uris como o(s) host(s) que executam Negotiate (dc01.example.com) para que o navegador envie tokens Kerberos automaticamente.

Dentro de about:config:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Agora, acessar http://dc01.example.com deverá fazer login diretamente no Zabbix, sem exibir o formulário.

13. Mantenha as chaves/tickets atualizados. A duração padrão de um ticket Kerberos é de aproximadamente 10 horas. Adicione um cron/timer do systemd para evitar expirações:

```
####para o serviço web
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####para o usuário de monitoramento
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Verificações de manutenção:

- klist -k /etc/apache2/http.keytab - verifique se o principal do serviço está presente no keytab.
- sudo tail -f /var/log/apache2/error.log - monitore os erros GSSAPI (gss_acquire_cred[_from]() failed to get server creds significa um problema no keytab/permissões ou um principal ausente).
- O retorno 401/403 de curl --negotiate geralmente significa principal incorreto, ausência de ticket, incompatibilidade no cabeçalho do host ou problema de permissão no sistema de arquivos; verifique os logs e os mapeamentos de domínio em /etc/krb5.conf.

Observações sobre segurança e permissões de arquivos

Os arquivos keytab devem ser legíveis somente pela conta que precisa deles. Exemplos de permissões: 0400, pertencente a zabbix:zabbix, para um keytab de usuário zabbix, ou 0440 e root:www-data para um keytab do Apache.

Evite armazenar senhas em texto simples de longa duração no host. Use keytabs ou principais de máquina associados ao domínio sempre que possível.

Ao executar testes ou scripts que definem KRB5CCNAME ou copiam keytabs, verifique novamente o proprietário e as permissões após a operação — quando um servidor web rejeita credenciais, geralmente o problema está nas permissões do arquivo.

Parâmetros de 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
type	Tipo de dado: para Read Coil e Read Discrete Input - *bit* para Read Holding Registers e Read Input Registers: *int8* - 8bit *uint8* - 8bit (sem sinal) *int16* - 16bit *uint16* - 16bit (sem sinal) *int32* - 32bit *uint32* - 32bit (sem sinal) *float* - 32bit *uint64* - 64bit (sem sinal) *double* - 64bit	bit uint16	uint64

Parâmetro	Descrição	Padrão	Exemplo
endianness	Tipo de endianness: *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Limitações: para 1 bit - be para 8 bits - be,le para 16 bits - be,le	be	le
offset	Número de registradores, a partir de 'address', cujo resultado será descartado. O tamanho de cada registrador é 16bit (necessário para suportar equipamentos que não suportam acesso de leitura aleatória).	0	4

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]`, em que:

- `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*

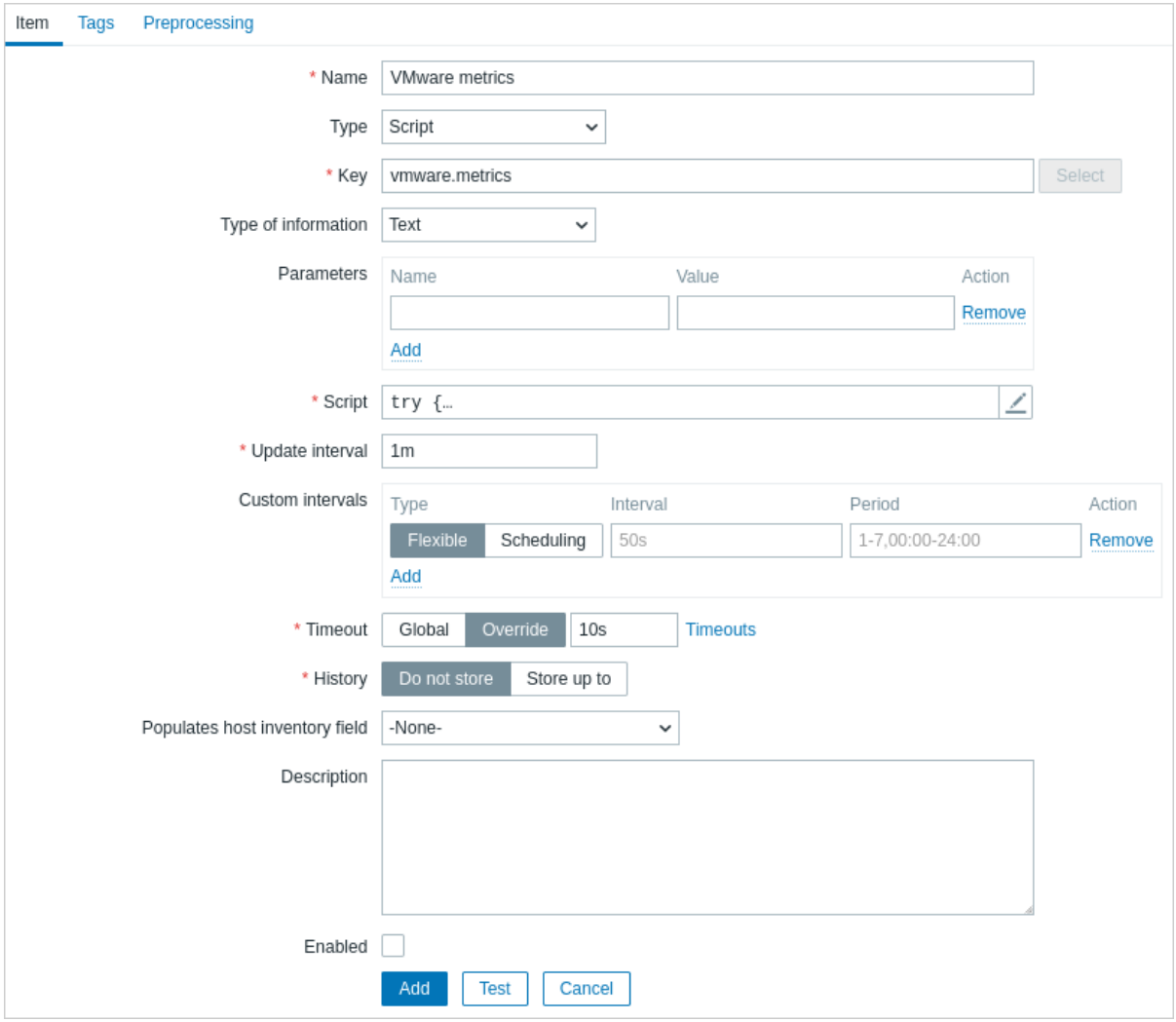
Assim, o exemplo acima resultaria no seguinte caminho do contador: `cpu/usagemhz[average]`

As descrições dos grupos de contadores de desempenho, os nomes dos contadores e os tipos de rollup podem ser encontrados na documentação da VMware.

É possível obter nomes internos e criar nomes personalizados de contadores de desempenho usando um item do tipo script no Zabbix.

Configuração

1. Crie um item Script desabilitado no host principal do VMware (onde o item **eventlog[]** está presente) com os seguintes parâmetros:



- *Nome*: VMware metrics
- *Tipo*: Script
- *Chave*: vmware.metrics
- *Tipo de informação*: Text
- *Script*: copie e cole o script fornecido abaixo
- *Timeout*: 10
- *Histórico*: Não armazenar
- *Habilitado*: desmarcado

Script

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

Depois que o item estiver configurado, clique no botão *Test*, depois em *Get value*.



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. No exemplo acima, o XPath será:

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

O caminho resultante do contador de desempenho neste caso é: `cpu/usagemhz[average]`

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 propriedades:

- **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 estiver disponível, defina como 0)
- **arch** - arquitetura do pacote
- **buildtime** - um objeto com 2 entradas:
  - **timestamp** - timestamp UNIX de quando o pacote foi compilado (se não estiver disponível, defina como 0)
  - **value** - data e hora em formato legível de quando o pacote foi compilado (se não estiver disponível, defina como string vazia)
- **installtime** - um objeto com 2 entradas:
  - **timestamp** - timestamp UNIX de quando o pacote foi instalado (se não estiver disponível, defina como 0)
  - **value** - data e hora em formato legível de quando o pacote foi instalado (se não estiver disponível, defina 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": ""
        }
    }
]
```

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 que contém informações de registros 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 DNS server, o item `net.dns.get[192.0.2.1,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[192.0.2.1,example.com]` pode retornar um erro:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```


Os seguintes tipos de códigos de erro são possíveis:

Scenario	”zbx_error_code”	”zbx_error_msg”
Nenhum erro e a resposta DNS foi recebida e analisada.	0	
DNS está indisponível.	-1	”Communication error”
Ocorre um erro durante a análise de JSON	-2	”Received unexpected response”

Com parâmetros adicionais, o item `net.dns.get[192.0.2.1,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": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Veja também

Para mais informações sobre registros DNS, consulte:

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

Valores de retorno para net.if.get

Visão geral

Esta seção fornece detalhes sobre os valores de retorno do item `net.if.get`. O item é suportado no Linux pelo Zabbix agent e pelo Zabbix agent 2.

Detalhes

Este item retorna informações detalhadas sobre interfaces de rede cujos nomes correspondem à expressão regular fornecida. A saída é estruturada como um objeto JSON contendo arrays `config` e `values`.

A saída contém as seguintes propriedades:

- **name** - nome da interface, lido de `/proc/net/dev`;

- **ifalias** - alias da interface, lido de `/sys/class/net/<name>/ifalias`;
- **mac** - endereço MAC da interface, lido de `/sys/class/net/<name>/address`;
- **type** - tipo da interface (`physical`, `virtual`, `loopback`);
- **speed** - velocidade do link em Mbits/s, específica de Ethernet, lida de `/sys/class/net/<name>/speed`;
- **duplex** - valor de duplex mais recente ou atual da interface (`half`, `full`), específico de Ethernet, lido de `/sys/class/net/<name>/du`
- **administrative_state** - estado configurado (`down`, `up`), lido de `/sys/class/net/<name>/flags`;
- **operational_state** - estado operacional RFC2863 da interface (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), lido de `sys/class/net/<name>/operstate`;
- **carrier** - presença de link físico (0 - o link físico está down, 1 - o link físico está up), lido de `/sys/class/net/<name>/carrier`;
- **carrier_changes** - número total de vezes que o estado do link mudou de up para down e vice-versa, lido de `/sys/class/net/<name>/carrier_changes`;
- **carrier_up_count** - número de vezes que o link ficou up, lido de `/sys/class/net/<name>/carrier_up_count`;
- **carrier_down_count** - número de vezes que o link ficou down, lido de `/sys/class/net/<name>/carrier_down_count`;
- **in** - tráfego de entrada, lido de `/proc/net/dev`. Contém contadores da interface:
  - **bytes** - número de bytes válidos;
  - **packets** - número de pacotes válidos;
  - **errors** - número total de pacotes inválidos;
  - **dropped** - número de pacotes recebidos, mas não processados + contagem de pacotes perdidos pelo host;
  - **overruns** - número de eventos de overflow;
  - **frame** - número de pacotes com erros de enquadramento;
  - **compressed** - número de pacotes compactados recebidos corretamente;
  - **multicast** - pacotes multicast recebidos.
- **out** - tráfego de saída, lido de `/proc/net/dev`. Contém contadores da interface:
  - **bytes** - número de bytes válidos;
  - **packets** - número de pacotes válidos;
  - **errors** - número total de pacotes inválidos;
  - **dropped** - número de pacotes recebidos, mas não processados + contagem de pacotes perdidos pelo host;
  - **overruns** - número de eventos de overflow;
  - **collisions** - número de colisões durante transmissões de pacotes;
  - **carrier** - número de erros de transmissão de quadros devido à perda de carrier durante a transmissão;
  - **compressed** - número de pacotes compactados transmitidos.

Métricas indisponíveis são excluídas da saída JSON. No exemplo a seguir, nenhum alias está configurado para a interface "lo", portanto a propriedade `ifalias` é excluída.

Exemplo:


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
```

```
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
```

```
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


Veja também

Para mais informações sobre a ABI sysfs para dispositivos de rede, consulte:

- Documentação do kernel Linux

Valores de retorno para vfs.dev.get

Visão geral

Esta seção fornece detalhes sobre o valor de retorno do item `vfs.dev.get`. O item é suportado no Linux pelo Zabbix agent e pelo Zabbix agent 2, e no AIX pelo Zabbix agent.

Detalhes

Este item retorna informações detalhadas sobre dispositivos de bloco cujos nomes correspondem à expressão regular fornecida.

A saída é estruturada como um objeto JSON contendo `config` e, dependendo do parâmetro <mode>, arrays `values`.

O array `config` contém as propriedades disponíveis para o modo selecionado. O array `values` é retornado apenas para modos que fornecem estatísticas de dispositivos de bloco.

As métricas indisponíveis são excluídas da saída JSON.

As seções a seguir descrevem as propriedades retornadas para cada modo.

modo disks

`vfs.dev.get[disks,.*]` retorna dados de configuração de dispositivos de bloco.

Propriedades:

- **name** - nome do dispositivo de bloco;
- **devid** - identificador do dispositivo de bloco, por exemplo, um ID de dispositivo baseado em WWN ou gerado pelo kernel;
- **type** - tipo do dispositivo de bloco (`disk`, `partition`, `rom`);
- **path** - caminho para o nó do dispositivo de bloco;
- **model** - modelo do dispositivo;
- **serial** - número de série do dispositivo;

- **wwn** - World Wide Name (WWN) do dispositivo;
- **size_bytes** - tamanho do dispositivo em bytes;
- **logical_block_size** - tamanho do bloco lógico em bytes;
- **physical_block_size** - tamanho do bloco físico em bytes.

Exemplo para Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


Exemplo para AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
```

```
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```


modo devices

`vfs.dev.get[devices,.*]` retorna dados de configuração de dispositivos de bloco, incluindo partições.

Propriedades:

- **name** - nome do dispositivo de bloco.
- **devid** - identificador do dispositivo de bloco.
- **type** - tipo do dispositivo de bloco (`disk`, `partition`, `rom`).
- **partitions** - partições no dispositivo. Contém os nomes das partições e seus tamanhos em bytes.

Exemplo para Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```


modo disk_stats

`vfs.dev.get[disk_stats,.*]` retorna dados de configuração de dispositivos de bloco e estatísticas de disco.

Propriedades:

- **name** - nome do dispositivo de bloco.
- **devid** - identificador do dispositivo de bloco.
- **type** - tipo do dispositivo de bloco (`disk`, `partition`, `rom`).

- **size_bytes** - tamanho do dispositivo em bytes.
- **logical_block_size** - tamanho lógico do bloco em bytes.
- **stats** - estatísticas de E/S do dispositivo de bloco. Contém:
  - **reads_completed** - número de operações de leitura concluídas;
  - **writes_completed** - número de operações de gravação concluídas;
  - **bytes_read** - número de bytes lidos;
  - **bytes_written** - número de bytes gravados;
  - **io_time_ms** - tempo total gasto realizando E/S, em milissegundos.

Exemplo para Linux


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    }
  ]
```

```
}
```

Exemplo para AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
  ]
}
```

modo device_stats

`vfs.dev.get[device_stats,.*]` retorna dados de configuração de dispositivos de bloco e estatísticas de dispositivos.

Propriedades:

- **name** - nome do dispositivo de bloco.
- **devid** - identificador do dispositivo de bloco.
- **type** - tipo do dispositivo de bloco (`disk`, `partition`, `rom`).
- **size_bytes** - tamanho do dispositivo em bytes.
- **stats** - estatísticas de I/O do dispositivo de bloco. Contém:
  - **reads_completed** - número de operações de leitura concluídas;
  - **writes_completed** - número de operações de gravação concluídas;
  - **bytes_read** - número de bytes lidos;
  - **bytes_written** - número de bytes gravados;
  - **io_time_ms** - tempo total gasto em I/O, em milissegundos.

Exemplo para Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
```

```
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb",
```

```
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "sdb1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    },
    {
      "name": "nvme0n1p1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
```

```
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```

Notas sobre itens system.cpu.util no Windows

O item system.cpu.util fornece a porcentagem de utilização da CPU.

Quando um processo coletor é iniciado no agent Zabbix para Windows, um buffer para N items para N processadores lógicos (threads) é inicializado. Os valores são periodicamente atualizados no buffer usando contadores de desempenho do Windows. Esses valores são 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
```

Suporte a valores JSON grandes

Para garantir o tratamento confiável de valores grandes com o tipo de dado JSON data type (1 MiB ou mais), revise e aplique as seguintes alterações de configuração conforme necessário:

1. Se você estiver usando MySQL ou MariaDB, aumente o tamanho máximo permitido do pacote em `/etc/mysql/my.cnf` (em alguns sistemas, o padrão pode ser 1 MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Ajuste outras server system variables do MySQL/MariaDB e InnoDB system variables conforme necessário. Por exemplo:

```
innodb_io_capacity=1900
connect_timeout=600
wait_timeout=57600
interactive_timeout=57600
net_read_timeout=7200
net_write_timeout=7200
net-buffer-length=32704
```

3. Aumente `HistoryCacheSize` para 2G no arquivo de configuração do Zabbix server.

4. Se você estiver usando Zabbix proxy, aumente `HistoryCacheSize` para 2G e `ProxyMemoryBufferSize` para 2G no arquivo de configuração do proxy.

5. Se você estiver usando Zabbix agent, aumente os timeouts de comunicação entre o Zabbix server/proxy e o agent (por exemplo, item timeouts ou o valor do parâmetro `Timeout` nos arquivos de configuração do Zabbix server/proxy e agent). Caso contrário, as mensagens entre eles podem não ser concluídas, e você poderá ver erros como *"message length does not match expected length"*. Além disso, revise outros timeouts conforme necessário, como o timeout de script timeout ou de item test.

Funções suportadas

Veja Expressão de trigger e funções.

Macros

É possível usar Macros suportadas prontas para uso e Macros de usuário suportadas por localização.

Macros suportadas por localização

Visão geral

A tabela macros suportadas por local contém a referência de todas as macros (integradas, de expressão e definíveis pelo usuário) que são suportadas no local específico.

Se todas as macros de um grupo relacionado forem suportadas, então apenas o nome do grupo é fornecido. Clique em cada nome de macro ou nome de grupo para ver os detalhes completos 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 funções de macro.

Macros suportadas por localização

Location	Supported macros	Comments
Action		
Time period condition, Default operation step duration, Step duration	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados
Notifications and commands		
Trigger-based	macros Action macros Date/time {ESC.HISTORY} macros Event, macros Cause/symptom event 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 Target host (somente comandos) macros Host inventory macros Item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) macros 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 definíveis pelo usuário	Veja também: Using macros in messages

Location	Supported macros	Comments
Problem update	macros Action macros Date/time {ESC.HISTORY} macros Event, macros Cause/symptom event, macros Recovery event macros Event update: todos (exceto {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro de expressão: {?EXPRESSION} macros Function Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Target host (somente comandos) macros Host inventory macros Item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) macros 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 definíveis pelo usuário	
Problem recovery	macros Recovery event macros Event {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Discovery	macros Action macros Date/time macros Discovery Macros de evento: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} macros Target host (somente comandos) macros Proxy	
Autoregistration	macros Action macros Date/time 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 Target host (somente comandos) macros Proxy	
Service-based	macros Action macros Date/time {ESC.HISTORY} Macros de evento: todos exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA} macros Service macros definíveis pelo usuário	
Service recovery	macros Event (exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macros Recovery event	

Location	Supported macros	Comments
Service update	macros Action macros Date/time {ESC.HISTORY} macros Event (exceto {EVENT.ACK.STATUS}, {EVENT.OPDATA}) macros de atualização de evento: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} macros Service macros definíveis pelo usuário	
Internal notifications	macros Action macros Date/time {ESC.HISTORY} macros Event (exceto {EVENT.NSEVERITY}, {EVENT.SEVERITY}), macros Recovery event Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory Macros de item: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} macros Proxy	
Item-based	Macros de item: {ITEM.STATE}, {ITEM.STATE.ERROR}	
LLD rule-based	macros Low-level discovery notification	
Trigger-based	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 definíveis pelo usuário	
Alert script parameters	macros Alert macros definíveis pelo usuário (somente global)	
Connector		
URL, Username, Password, Bearer token, HTTP proxy, SSL certificate, SSL key file, SSL key password field	macros definíveis pelo usuário (somente global)	
Timeout field	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.

Location	Supported macros	Comments
Graph name	Macro de expressão: {?EXPRESSION}	Somente uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (podem ser usados sufixos de tempo) é suportada neste macro nesta localização. O macro {HOST.HOST<1-9>} pode ser usado como host dentro do macro (veja também indexed macros). 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, host prototype		
Interface IP/DNS	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} macros definíveis pelo usuário [1]	[1] Para endereço IP, um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados
Interface port	macros definíveis pelo usuário	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
SNMP community *SNMPv3 Context name, Security name, Authentication passphrase, Privacy passphrase* *IPMI username, password*	macros definíveis pelo usuário	
Item, item prototype, LLD rule		
Name	macros definíveis pelo usuário	Não suportado em nomes de regras LLD.
Description	macros definíveis pelo usuário	

Location	Supported macros	Comments
Key parameters	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	Os macros {HOST.*} suportados serão resolvidos para a interface selecionada para o item. Quando usados em items sem interfaces, eles serão resolvidos para a interface do Zabbix agent, SNMP, JMX ou IPMI do host, nessa ordem de prioridade, ou para 'UNKNOWN' se o host não tiver nenhuma interface. Macros não são suportados na chave do item de items Zabbix trapper.
Update interval, *Custom intervals*, *Timeout (available for supported item types)*, *Store up to (for history/trends)* fields	macros definíveis pelo usuário	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Delete lost resources, *Disable lost resources* fields	macros definíveis pelo usuário	Esses campos são suportados apenas para regras LLD. Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Preprocessing step parameters (including custom scripts), *custom error-handling parameters* (*Set value to* and *Set error to* fields)	macros definíveis pelo usuário	
Filter regular expressions	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	Esses campos são suportados apenas para regras LLD.
Override filter regular expressions	macros definíveis pelo usuário	Esses campos são suportados apenas para regras LLD.
Override operations: *Update interval*, *History storage period*, *Trend storage period* (for item prototypes)	macros definíveis pelo usuário	Esses campos são suportados apenas para regras LLD. Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Item types		

Location	Supported macros	Comments
Browser item *parameter names and values*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macros definíveis pelo usuário	
Script item *parameter names and values*		
Calculated/aggregate item formula (*expression constants*, *function parameters*, *item key parameters*, (aggregate item only) *filter conditions* (host group name and tag name))	macros definíveis pelo usuário	
Database monitor item *Username, Password, SQL query* fields	macros definíveis pelo usuário	
HTTP agent item *URL, Query fields, Request body, Headers (names and values), SSL certificate file, SSL key file* fields	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 definíveis pelo usuário	URLs que contenham um macro secreto não funcionarão, pois o macro nelas será resolvido como "******"
HTTP agent item *HTTP Proxy* field	Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macros definíveis pelo usuário	
HTTP agent item *Required status codes, HTTP authentication username/password, SSL key password* fields	macros definíveis pelo usuário	
HTTP agent item *Allowed hosts* field	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} macros definíveis pelo usuário	
JMX agent item *Endpoint* field	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} macros definíveis pelo usuário	
Trapper item *Allowed hosts* field	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} macros definíveis pelo usuário	
SNMP agent item *SNMP OID* field	macros definíveis pelo usuário	
SSH and Telnet item *script*	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	
SSH agent item *Username, Public key file, Private key file, Password* fields	macros definíveis pelo usuário	
Telnet agent item *Username, Password* fields	macros definíveis pelo usuário	
Map		

Location	Supported macros	Comments
URL, URL name	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} macros Host inventory macros Map {TRIGGER.ID}	
Element label	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 Host inventory macros Map Macros de trigger: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] Somente uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (podem ser usados sufixos de tempo) é suportada neste macro nesta localização. O macro {HOST.HOST<1-9>} pode ser usado como host dentro do 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)}
Link label	Macro de expressão: {?EXPRESSION} [1]	
Shape label	Macro de expressão: {?EXPRESSION} [1]	
Text field in map shapes	{MAP.NAME}	
Network discovery		
Update interval field	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
SNMP community, SNMP OID fields *SNMPv3 Context name, Security name, Authentication passphrase, Privacy passphrase* fields	macros definíveis pelo usuário (somente global)	
Proxy		
Address for active agents > Port (if the proxy belongs to a group) For passive proxy mode: interface *Address* and *Port* fields *Timeouts for item types* fields in override mode	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Proxy group		

Location	Supported macros	Comments
Failover period, Minimum number of proxies fields	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Scripts		
Commands, confirmation text (Type: script, manual **host action**)	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory [2] {MANUALINPUT} macros Username macros definíveis pelo usuário	[1] Somente texto de confirmação [2] Suportado para Zabbix server e Zabbix proxy
Commands (Type: script, manual **event action**)	macros Date/time macros Event, macros Cause/symptom event, macros Recovery event (se a recuperação tiver ocorrido) 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 Host inventory macros Item (exceto {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} macros 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 Username macros definíveis pelo usuário	
Confirmation text (Type: script, manual **event action**)	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 Host inventory {MANUALINPUT} macros Username macros definíveis pelo usuário	
URL, confirmation text (Type: URL, manual **host/event action**)		
Tag name and value	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory Macros de item: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de trigger: {TRIGGER.ID} (somente valor da tag de trigger) macros definíveis pelo usuário	Macros de nome e valor de tag são resolvidos somente durante o processo de geração do evento.
Trigger		

Location	Supported macros	Comments
Name	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 posicionais/referências: $1...$9 macros definíveis pelo usuário	
Event name	Macros de date/time: {TIME}, {TIMESTAMP} Macro de expressão: {?EXPRESSION} macros Function 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}	
Operational data	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 definíveis pelo usuário	
Expression	{TRIGGER.VALUE} macros definíveis pelo usuário [1]	[1] Somente em constantes e parâmetros de função; macros secretos não são suportados
Tag for matching, *Menu entry name*, *Menu entry URL* fields	macros definíveis pelo usuário	URLs que contenham um macro secreto não funcionarão, pois o macro nelas será resolvido como "******".
Description	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 definíveis pelo usuário	
URL	{EVENT.ID} 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}	
User parameter commands	Macros posicionais/referências: $1...$9	
Web scenario		
Name, *Agent*, *HTTP proxy* fields	macros definíveis pelo usuário	
Update interval field	macros definíveis pelo usuário	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Variables, *Headers*, *SSL certificate*, *SSL key file* fields	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	

Location	Supported macros	Comments
User, Password, SSL key password fields	macros definíveis pelo usuário	
Web scenario step *Name, Variables (values only)* fields	macros definíveis pelo usuário	
Web scenario step *Timeout, Required status codes* fields	macros definíveis pelo usuário	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Web scenario step *URL, Headers (names and values), Required string* fields	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	
Web scenario step *Post* field	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Webhook media type		
Parameter values	macros Alert Todos os macros suportados em notificações de problema baseadas em trigger (veja acima)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
JavaScript script, *JavaScript* script parameter name	macros definíveis pelo usuário (somente global)	
Menu entry name, Menu entry URL	{EVENT.TAGS.<tag name>}	
Widgets		
Gauge widget *Description* field	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory Macros de item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} macros definíveis pelo usuário (somente global)	
Honeycomb widget *Primary/secondary label* field		
Item value widget *Description* field		
Top hosts widget *Text data* column	Macros de host: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros Host inventory macros definíveis pelo usuário (somente global)	
URL (Dynamic) widget *URL* field	Macros de host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} macros definíveis pelo usuário	
***Custom text* column**		
Monitoring > Problems	macros Host (exceto {HOST.METADATA}) macros Host inventory macros Item (exceto {ITEM.STATE.ERROR}) Macros de trigger: {TRIGGER.ID} Macros de evento: {EVENT.ID} macros definíveis pelo usuário	

Location	Supported macros	Comments
Monitoring > Hosts	macros Host (exceto {HOST.METADATA}) macros Host inventory macros definíveis pelo usuário	
Monitoring > Latest data	macros Host (exceto {HOST.METADATA}) macros Host inventory macros Item (exceto {ITEM.STATE.ERROR}) macros definíveis pelo usuário	
Data collection > Templates	Macros de host: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME}	
Data collection > Hosts	macros Host (exceto {HOST.METADATA}) macros Host inventory macros definíveis pelo usuário	
Other locations		
Working time field	macros definíveis pelo usuário (somente global)	Um único macro precisa preencher todo o campo; múltiplos macros/combinação com texto não são suportados.
Timeouts for item types field (global setting)		
User media *When active* field	macros definíveis pelo usuário (somente global)	
Email media type *Username, Password* fields	macros definíveis pelo usuário (somente global)	Macros secretos são recomendados.
Scheduled reports *Subject, Message* fields	Macros de date/time: {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 no formato `aaaa.mm.dd`.
 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}

Uma matriz JSON contendo objetos de tags de evento.
Expandido para uma matriz vazia se não existirem tags.

{EVENT.TAGS.<tag name>}

O valor da tag do evento referenciado pelo nome da tag.
 Um nome de tag que contenha caracteres não alfanuméricos (incluindo caracteres multibyte UTF não ingleses) deve estar 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 no formato `hh:mm:ss`.
 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 no formato `aaaa.mm.dd`.
 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}

Uma matriz JSON contendo objetos de tag de evento de causa objects.
Expandido para uma matriz vazia se não houver 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 que contenha caracteres não alfanuméricos (incluindo caracteres UTF multibyte não ingleses) deve estar 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 causador no formato `hh:mm:ss`.
 Usado no contexto de um evento 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 do serviço e valores.
 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 no formato `aaaa.mm.dd`.

{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 do evento de recuperação. Expandida para uma string vazia se não existirem tags.

{EVENT.RECOVERY.TAGSJSON}

Um array JSON contendo objetos de tag do evento de recuperação. Expandido para um array vazio se não existirem tags.

{EVENT.RECOVERY.TIME}

A hora do evento de recuperação no formato `hh:mm:ss`.

{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 da(s) ação(ões) executada(s) durante a atualização do problema.
 Resolve para os seguintes valores: *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* e *closed* (dependendo de quantas ações são executadas em uma única atualização).

{EVENT.UPDATE.ACTIONJSON}

Uma matriz JSON contendo detalhes da(s) ação(ões) executada(s) durante a atualização do problema.
 Possíveis valores de propriedades JSON:
- true (para as propriedades `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` e `symptom`);
- <message string> (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 severidade `old` e `new`).

 Por exemplo: `{"acknowledge":true,"message":"Manutenção mensal.","severity":{"old":2,"new":1},"suppress_until":1730851199,"timestamp":1730822048}`.

{EVENT.UPDATE.DATE}

A data da atualização do evento (reconhecimento, etc) no formato `yyyy.mm.dd`.

{EVENT.UPDATE.HISTORY}

O log de atualizações de problemas (reconhecimentos, etc).

{EVENT.UPDATE.MESSAGE}

A mensagem de atualização do problema.

{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) no formato `hh:mm:ss`.

{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.
 Somente funções com `/host/key` como primeiro parâmetro são contadas.

 Esta macro pode ser usada com um índice numérico, por exemplo, {FUNCTION.VALUE<1-9>}, para apontar para a primeira, segunda, terceira etc. função em uma expressão de trigger. Veja indexed macros.

{FUNCTION.RECOVERY.VALUE}

O valor da N-ésima função baseada em item na expressão de recuperação no momento do evento.*
 Somente funções com `/host/keys` 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 indexed macros.

Macros de host

{HOST.CONN}

O endereço IP do host ou o nome DNS, dependendo das configurações do host.
 Em scripts globais, campos de IP/DNS da 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 para '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. Consulte 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. Consulte indexed macros.

{HOST.DNS}

O nome DNS do host.
 Em scripts globais, campos de IP/DNS da interface e cenários web, a macro será resolvida para a interface agent principal. Se uma interface agent não estiver definida, será usada a interface SNMP principal. Se uma interface SNMP também não estiver definida, será usada a interface JMX principal. Se uma interface JMX também não estiver definida, será usada a interface IPMI principal. Se o host não tiver nenhuma interface, a macro será resolvida como 'UNKNOWN'.

 Essa 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. Consulte indexed macros.

{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}).`

 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 indexed macros.

{HOST.IP}

O endereço IP do host.
 Em scripts globais, campos de IP/DNS da interface e cenários web, a macro será resolvida para a interface agent principal. Se uma interface agent não estiver definida, será usada a interface SNMP principal. Se uma interface SNMP também não estiver definida, será usada a interface JMX principal. Se uma interface JMX também não estiver definida, será usada a interface IPMI principal. Se o host não tiver nenhuma interface, a macro será resolvida como 'UNKNOWN'.

 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 indexed macros.

{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. Consulte indexed macros.

{HOST.PORT}

A porta do host (agent).
 Em scripts globais, campos de IP/DNS da 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 não estiver definida, a interface principal 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.PORT<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

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 indexed macros.

{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 indexed macros.

{INVENTORY.CHASSIS}

O campo *Chassis* 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 indexed macros.

{INVENTORY.CONTACT}

O campo *Contact* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.CONTACT<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Consulte indexed macros.

{INVENTORY.CONTRACT.NUMBER}

O campo *Contract number* 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 indexed macros.

{INVENTORY.DEPLOYMENT.STATUS}

O campo *Deployment status* 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 indexed macros.

{INVENTORY.HARDWARE}

O campo *Hardware* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HARDWARE<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{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 indexed macros.

{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 indexed macros.

{INVENTORY.HOST.NETWORKS}

O campo *Host networks* 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. Consulte indexed macros.

{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 indexed macros.

{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. Consulte indexed macros.

{INVENTORY.HW.DATE.DECOMM}

O campo *Date hardware decommissioned* 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 indexed macros.

{INVENTORY.HW.DATE.EXPIRY}

O campo *Date hardware maintenance expires* 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 indexed macros.

{INVENTORY.HW.DATE.INSTALL}

O campo *Date hardware installed* 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 indexed macros.

{INVENTORY.HW.DATE.PURCHASE}

O campo *Date hardware purchased* 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. Consulte indexed macros.

{INVENTORY.INSTALLER.NAME}

O campo *Installer name* 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 indexed macros.

{INVENTORY.LOCATION}

O campo *Location* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.LOCATION<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{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 indexed macros.

{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 indexed macros.

{INVENTORY.MACADDRESS.A}

O campo *MAC address A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.MACADDRESS.A<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{INVENTORY.MACADDRESS.B}

O campo *MAC address 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 indexed macros.

{INVENTORY.MODEL}

O campo *Model* 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 indexed macros.

{INVENTORY.NAME}

O campo *Name* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.NAME<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{INVENTORY.NOTES}

O campo *Notes* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.NOTES<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{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 indexed macros.

{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. Consulte indexed macros.

{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 indexed macros.

{INVENTORY.OS}

O campo *OS* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OS<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{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 *OS (Short)* 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 indexed macros.

{INVENTORY.POC.PRIMARY.CELL}

O campo *Primary POC cell* 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 indexed macros.

{INVENTORY.POC.PRIMARY.EMAIL}

O campo *E-mail principal do POC* 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 indexed macros.

{INVENTORY.POC.PRIMARY.NAME}

O campo *Nome do POC principal* 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 indexed macros.

{INVENTORY.POC.PRIMARY.NOTES}

O campo *Primary POC notes* 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 indexed macros.

{INVENTORY.POC.PRIMARY.PHONE.A}

O campo *Primary POC phone A* no inventário do host.

 Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.PHONE.A<1-9>} para apontar para o primeiro, segundo, terceiro etc. host em uma expressão de trigger. Veja indexed macros.

{INVENTORY.POC.PRIMARY.PHONE.B}

O campo *Primary POC phone B* 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. Consulte indexed macros.

{INVENTORY.POC.PRIMARY.SCREEN}

O campo *Primary POC screen name* 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 indexed macros.

{INVENTORY.POC.SECONDARY.CELL}

O campo *Secondary POC cell* 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 indexed macros.

{INVENTORY.POC.SECONDARY.EMAIL}

O campo *Secondary POC email* 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 indexed macros.

{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 indexed macros.

{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 *Secondary POC phone 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. Consulte indexed macros.

{INVENTORY.POC.SECONDARY.PHONE.B}

O campo *Secondary POC phone B* 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 indexed macros.

{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 indexed macros.

{INVENTORY.SERIALNO.A}

O campo *Serial number A* no inventário do host.

 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 indexed macros.

{INVENTORY.SERIALNO.B}

O campo *Serial number 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 indexed macros.

{INVENTORY.SITE.ADDRESS.A}

O campo *Site address 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 indexed macros.

{INVENTORY.SITE.ADDRESS.B}

O campo *Site address 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 indexed macros.

{INVENTORY.SITE.ADDRESS.C}

O campo *Site address 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 indexed macros.

{INVENTORY.SITE.CITY}

O campo *Site city* 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 indexed macros.

{INVENTORY.SITE.COUNTRY}

O campo *Site country* 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 indexed macros.

{INVENTORY.SITE.NOTES}

O campo *Site notes* 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. Consulte indexed macros.

{INVENTORY.SITE.RACK}

O campo *Localização do rack do site* 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 indexed macros.

{INVENTORY.SITE.STATE}

O campo *Site state/province* 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 indexed macros.

{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 indexed macros.

{INVENTORY.SOFTWARE}

O campo *Software* no inventário do host.

 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 indexed macros.

{INVENTORY.SOFTWARE.APP.A}

O campo *Software application 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 indexed macros.

{INVENTORY.SOFTWARE.APP.B}

O campo *Software application 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 indexed macros.

{INVENTORY.SOFTWARE.APP.C}

O campo *Software application 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 indexed macros.

{INVENTORY.SOFTWARE.APP.D}

O campo *Software application 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 indexed macros.

{INVENTORY.SOFTWARE.APP.E}

O campo *Software application 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 indexed macros.

{INVENTORY.SOFTWARE.FULL}

O campo *Software (Full details)* 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 indexed macros.

{INVENTORY.TAG}

O campo *Tag* no inventário do host.

 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 indexed macros.

{INVENTORY.TYPE}

O campo *Type* no inventário do host.

 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 indexed macros.

{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. Consulte 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 indexed macros.

{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 indexed macros.

{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. Consulte indexed macros.

{INVENTORY.VENDOR}

O campo *Vendor* 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 indexed macros.

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 indexed macros.

{ITEM.DESCRIPTION.ORIG}

A descrição original (com macros não resolvidos) 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 indexed macros.

{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 indexed macros.

{ITEM.KEY}

Chave do N-ésimo item na expressão de trigger que causou uma notificação.
 Funções de macro não são suportadas para esta macro se ela for usada como um espaço reservado no primeiro parâmetro de uma função de histórico, por exemplo, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Esta macro pode ser usada com um índice numérico como {ITEM.KEY<1-9>} para apontar para o primeiro, segundo, terceiro etc. host na expressão de trigger. Veja macros indexadas.

{ITEM.KEY.ORIG}

A chave original (com macros não resolvidos) 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 indexed macros.

{ITEM.LASTVALUE}

O valor mais recente do N-ésimo item na expressão de trigger que causou uma notificação.
 Ele será resolvido como *UNKNOWN* no frontend se o valor mais recente do histórico tiver sido coletado há mais tempo do que o período *Max history display period* (definido na seção de menu *Administration > General*).
 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, ela manterá o valor do item do momento em que o problema ocorreu.
Quando usada em notificações, em alguns casos a macro pode não ser resolvida para o valor mais recente do item no momento em que o trigger disparou. Por exemplo, se um item receber rapidamente dois valores, ”A” e ”B”, e o trigger disparar para ”A”, as notificações podem mostrar ”B” como o valor mais recente devido a um pequeno atraso de processamento - o valor mais recente do item mudou entre o momento em que o trigger disparou e quando a notificação foi criada. Para evitar isso, você pode usar a macro {ITEM.VALUE}, que é resolvida para o valor no momento em que o trigger dispara, garantindo que o valor correto seja usado na notificação.
 Ela é um alias de `last(/{HOST.HOST}/{ITEM.KEY})`.
 O valor resolvido para itens de texto/log é truncado para 20 caracteres pelo frontend nos seguintes locais:
- Dados operacionais;
- Descrição do trigger;
- URLs do trigger;
- Rótulos de URL do trigger;
- Descrição do widget de valor do item.
Para resolver para um valor completo, você pode usar funções de macro, pois nenhum valor é truncado pelo server. Por exemplo: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Esta macro pode ser usada com um índice numérico, por exemplo {ITEM.LASTVALUE<1-9>} para apontar para o primeiro, segundo, terceiro etc. item em uma expressão de trigger. Veja macros indexadas.

{ITEM.LASTVALUE.AGE}

O tempo decorrido entre a coleta do valor mais recente 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.LASTVALUE.AGE<1-9>} para apontar para o primeiro, segundo, terceiro etc. item em uma expressão de trigger. Veja indexed macros.

{ITEM.LASTVALUE.DATE}

A data em que o valor mais recente do item foi coletado.
 Exibida no formato `YYYYMMDD`.
 O fuso horário do server é usado em nomes de trigger (na lista *Monitoring* > *Problems*), nomes de eventos e nomes e valores de tags. Em todos os outros casos, é usado o fuso horário do usuário.

 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. Consulte macros indexadas.

{ITEM.LASTVALUE.TIME}

O horário em que o valor mais recente do item foi coletado.
 Exibido no formato `HHMMSS`.
 O fuso horário do server é usado nos nomes de trigger (na lista *Monitoring* > *Problems*), nomes de eventos e nomes e valores de tags. Em todos os outros casos, é usado o fuso horário do usuário.

 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 indexed macros.

{ITEM.LASTVALUE.TIMESTAMP}

O timestamp UNIX em que o valor mais recente 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 indexed macros.

{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 indexed macros.

{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 indexed macros.

{ITEM.LOG.EVENTID}

O ID do evento no log de eventos.
 Somente para monitoramento do 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 indexed macros.

{ITEM.LOG.NSEVERITY}

A severidade numérica do evento no log de eventos.
 Somente para monitoramento do 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 indexed macros.

{ITEM.LOG.SEVERITY}

A severidade verbal do evento no log de eventos.
 Somente para monitoramento do log de eventos do Windows.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.SEVERITY<1-9>} para apontar para o primeiro, segundo, terceiro etc. item em uma expressão de trigger. Veja indexed macros.

{ITEM.LOG.SOURCE}

A origem do evento no log de eventos.
 Somente para monitoramento do 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 indexed macros.

{ITEM.LOG.TIME}

O momento em que a entrada de log foi gravada no log.

 Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.TIME<1-9>} para apontar para o primeiro, segundo, terceiro etc. item em uma expressão de trigger. Veja indexed macros.

{ITEM.LOG.TIMESTAMP}

O timestamp UNIX em que a entrada de 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 indexed macros.

{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 indexed macros.

{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 indexed macros.

{ITEM.STATE}

O estado mais recente do N-ésimo item na expressão do trigger que causou uma notificação.
 Valores possíveis: *Not supported*, *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 indexed macros.

{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 um dos seguintes:

- O valor histórico (no momento do evento) do N-ésimo item na expressão do trigger, se usado no contexto de alteração de estado do trigger, por exemplo, ao exibir eventos ou enviar notificações;
- O valor mais recente do N-ésimo item na expressão do trigger, se usado sem o contexto de alteração de estado do trigger, por exemplo, ao exibir uma lista de triggers em uma janela pop-up de seleção. Nesse caso, funciona da mesma forma que {ITEM.LASTVALUE}.

No primeiro caso, ele será resolvido como *UNKNOWN* se o valor do histórico já tiver sido excluído ou nunca tiver sido armazenado.
No segundo caso, e somente no frontend, ele será resolvido como *UNKNOWN* se o valor histórico mais recente tiver sido coletado há mais tempo do que o período *Max history display period* (definido na seção de menu *Administration > General*).
O valor resolvido para itens de texto/log é truncado para 20 caracteres pelo frontend nos seguintes locais:
- Operational data;
- Trigger description;
- Trigger URLs;
- Trigger URL labels;
- Description of the item value widget.
Para resolver para um valor completo, você pode usar macro functions, 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 do trigger. Veja indexed macros.

{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 indexed macros.

{ITEM.VALUE.DATE}

A data em que o valor do item foi coletado.
 Exibida no formato `YYYYMMDD`.
 O fuso horário do server é usado em nomes de trigger (na lista *Monitoring* > *Problems*), nomes de eventos e nomes e valores de tags. Em todos os outros casos, é usado o fuso horário do usuário.

Esta 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. Consulte indexed macros.

{ITEM.VALUE.TIME}

O momento em que o valor do item foi coletado.
 Exibido no formato `HHMMSS`.
 O fuso horário do server é usado em nomes de trigger (na lista *Monitoring* > *Problems*), nomes de eventos e nomes e valores de tags. Em todos os outros casos, é usado o fuso horário do usuário.

Esta 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. Consulte indexed macros.

{ITEM.VALUE.TIMESTAMP}

O timestamp UNIX em que 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. Consulte indexed macros.

{ITEM.VALUETYPE}

O tipo de valor do N-ésimo item na expressão do trigger que causou uma notificação.
 Valores possíveis: *0* - Número decimal, *1* - Caractere, *2* - Log, *3* - Número inteiro 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 do trigger. Veja indexed macros.

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 estado mais recente da regra de descoberta em baixo nível.
 Valores possíveis: *Not supported*, *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 indexed macros.

{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-se em um dos seguintes casos:

- Proxy do item N na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui;
- Proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.DESCRIPTION} aqui, sem indexação;
- Proxy ao qual um agent ativo se registrou (em notificações de autoregistro). 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.
 Corresponde a um dos seguintes:

- Proxy do N-ésimo item na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui;
- Proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.NAME} aqui, sem indexação;
- Proxy ao qual um agent ativo se registrou (em notificações de autoregistro). 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. Consulte 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 da configuração do serviço. Expandido para uma string vazia se não existirem tags.

{SERVICE.TAGSJSON}

Um array 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 um array vazio se não existirem tags.

{SERVICE.TAGS.<tag name>}

O valor da tag do evento de serviço referenciado pelo nome da tag.
 As tags do evento de serviço podem ser definidas na seção de tags da configuração do serviço.
 Um nome de tag que contenha caracteres não alfanuméricos (incluindo caracteres UTF multibyte 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.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.
 As 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.
 As funções baseadas em item são avaliadas e substituídas pelos resultados no momento da geração do evento, enquanto todas as outras funções são exibidas exatamente como 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 da expressão do trigger.
 Valores possíveis: *Desconhecido*, *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*.

{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 estado OK, *1* - o trigger está no estado 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.

Macros indexadas

A sintaxe de macro indexada **{MACRO<1-9>}** pode referenciar apenas o N-ésimo item, função ou host no campo *Expression* de um trigger:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** resolvem para os endereços IP do primeiro, segundo e terceiro hosts na expressão do trigger (se presentes).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** resolvem 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}** resolvem para os valores das primeiras, segunda e terceira funções baseadas em item no momento do evento (se presentes).

No contexto de triggers, as macros indexadas sempre se referem ao campo *Expression* da configuração do trigger, e não à *Recovery expression*. Por exemplo, em um evento de recuperação, **{ITEM.VALUE2}** resolverá 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 **graph names**. Por exemplo, `{?func(/{HOST.HOST2}/key,param)}` em um graph name resolverá para o host do segundo item no graph.

Warning:
As macros indexadas não serão resolvidas em nenhum outro contexto além dos casos mencionados aqui. Para outros contextos, use macros sem índice (**{HOST.HOST}, {HOST.IP}**, etc.).

Macros de usuário suportadas por localização

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

Localização		Múltiplas macros/misturar com texto[1]
	Duração do passo	não

Hosts/protótipos de host

Na configuração de um host e de um protótipo de host, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
IP/DNS da interface		apenas DNS
Porta da interface		não
SNMP v1, v2		
	Comunidade SNMP	sim
SNMP v3		
	Nome do contexto	sim
	Nome de segurança	sim
	Senha de autenticação	sim
	Senha de privacidade	sim
IPMI		
	Nome de usuário	sim
	Senha	sim
Tags[2]		
	Nomes das tags	sim
	Valores das tags	sim

Itens / protótipos de item

Na configuração de um item ou de um protótipo de item, as macros de usuário podem ser usadas nos seguintes campos:

Local	Múltiplas macros/mistura com texto[1]
Nome do item	sim
Parâmetros da chave do item	sim[4]
Intervalo de atualização	não
Intervalos personalizados	não
Tempo limite (disponível para tipos de item compatíveis)	não

Local	Múltiplas macros/mistura com texto[1]
Armazenar até (para histórico e tendên-cias)	não
Descrição	sim
Item cal-cu-lado/agregado	
Fórmula (constantes de expressão e parâmetros de função; parâmetros da chave do item; (*somente item agregado*) condições de filtro (nome do grupo de hosts e nome da tag))	sim
Monitor de banco de da-dos	
Nome de usuário	sim
Senha	sim
Consulta SQL	sim
HTTP agent	
URL[3]	sim
Campos de consulta	sim
Corpo da solicitação	sim
Cabeçalhos (nomes e valores)	sim
Códigos de status obrigatórios	sim
Proxy HTTP	sim
Nome de usuário da autenticação HTTP	sim
Senha da 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 nave-g-ador	
Nomes e valores dos parâmetros	sim
Item de con-sulta de teleme-tria	
Deslocamento de tempo	não
Limite de retrospectiva	não
Granularidade	não

Local	Múltiplas macros/mistura com texto[1]
SNMP agent	
OID SNMP	sim
SSH agent	
Nome de usuário	sim
Arquivo de chave pública	sim
Arquivo de chave privada	sim
Senha	sim
Script	sim
TELNET agent	
Nome de usuário	sim
Senha	sim
Script	sim
Zabbix trapper	
Hosts permitidos	sim
Tags[2]	
Nomes das tags	sim
Valores das tags	sim
Etapas de pré-processamento	
Parâmetros (incluindo scripts personalizados)	sim
Parâmetros personalizados de tratamento de erros (campos *Definir valor como* e *Definir erro como*)	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/mistura com texto[1]
Parâmetros da chave	sim
Intervalo de atualização	não
Intervalo personalizado	não

Localização	Múltiplas macros/mistura com texto[1]
Timeout (disponível para tipos de item suportados)	não
Excluir recursos perdidos	não
Desabilitar recursos perdidos	não
Descrição	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
Monitor de banco de dados	
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

Localização	Múltiplas macros/mistura com texto[1]
Cabeçalhos (nomes e valores)	sim
Códigos de status obrigatórios	sim
Nome de usuário para autenticação HTTP	sim
Senha para 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:

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the proxy belongs to a group)	no
Interface address and port (for passive proxy)	no
Timeouts for item types	no

Grupos de proxy

Em uma configuração de grupo de proxy, macros de usuário podem ser usadas nos seguintes campos:

Location	Multiple macros/mix with text[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
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	

Localização		Múltiplas macros/mistura com texto[1]
	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, macros de usuário podem ser usadas nos seguintes campos:

Local	Múltiplas macros/mistura com texto[1]
Scripts globais (URL, script, SSH, Telnet, IPMI), incluindo texto de confirmação	yes
Webhooks	
Script JavaScript	no
Nome do parâmetro do script JavaScript	no
Valor do parâmetro do script JavaScript	yes
Dashboards	
Coluna do tipo de dado *Text* no widget *Top hosts* do dashboard	yes
Parâmetro *Description* no widget *Item value* e *Gauge* do dashboard	yes
Parâmetro *Text* do rótulo primário/secundário no widget *Honeycomb* do dashboard	yes
Parâmetro *URL*[3] no widget *URL* do dashboard	yes
Monitoring > *Problems*, *Hosts*, *Latest data*	
Coluna *Custom text*	yes

Local	Múltiplas macros/mistura com texto[1]
Data collection > *Templates*, *Hosts*	
Coluna *Custom text*	yes
Alerts > *Media types* > *Message templates*	
Assunto	yes
Mensagem	yes
Alerts > *Media types* > *Script*	
Parâmetros do script	yes
Alerts > *Media types* > *Media type*	
Campos *Username* e *Password* para o tipo de mídia *Email* (quando *Authentication* estiver definido como ”Username and password”; secret macros recommended)	yes
Users > *Users* > *Media*	
Quando ativo	yes
Administration > *General* > *GUI*	
Horário de trabalho	no
Administration > *General* > *Timeouts*	

Local	Múltiplas macros/mistura com texto[1]
Timeouts para tipos de item	no
Administration > *General* > *Connectors*	
URL	yes
Username	yes
Password	yes
Bearer token	yes
Timeout	no
HTTP proxy	yes
SSL certificate file	yes
SSL key file	yes
SSL key password	yes

Para uma lista completa de todas as macros suportadas no Zabbix, consulte supported macros.

Notas de rodapé

[1] Se múltiplas macros em um campo ou macros misturadas com texto não forem suportadas para o local, uma única macro deve preencher todo o campo.

[2] Macros usadas em nomes e valores de tags são resolvidas apenas durante o processo de geração de eventos.

[3] URLs que contêm uma macro secreta não funcionarão, pois a macro nelas será resolvida como "******".

[4] Macros não são suportadas na chave do item de itens Zabbix trapper.

Compatibilidade de versão

Agents compatíveis

Para ser compatível com o Zabbix 8.0, o Zabbix agent não deve ser anterior à versão 1.6 nem posterior à versão 8.0.

Talvez seja necessário revisar a configuração de agents mais antigos, pois alguns parâmetros foram alterados, por exemplo, os parâmetros relacionados ao registro em log em versões anteriores à 3.0.

Para aproveitar totalmente as funcionalidades mais recentes, as métricas, o desempenho aprimorado e o menor uso de memória, use o agent compatível mais recente.

Observações para Windows XP

- No Windows XP de 32 bits, não use agents do Zabbix mais recentes que a versão 6.0.x;
- No Windows XP/Server 2003, não use templates de agents mais recentes que o Zabbix 4.0.x. Os templates mais recentes usam contadores de desempenho em inglês, que são compatíveis somente 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
Desatualizado	Sim	Não	Comandos remotos (por exemplo, scripts de shell); Verificações imediatas do valor do item (ou seja, *Executar agora*); Observação: testes de pré-processamento com um 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.

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`

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 oferece suporte aos 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 *(somente funções de tendência)*;
 y - anos *(somente funções de tendência)*.

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 de item

Os sufixos também podem ser usados para exibir valores numéricos de item em um formato legível para humanos.

Para habilitar isso, use um dos seguintes sufixos no campo *Units* ao configurar um item:
 B - bytes;
 Bps - bytes por segundo;
 s - segundos, exibidos usando até três unidades de tempo não zero maiores;
 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 à forma 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 unidades de tempo não zero maiores são exibidas (por exemplo, 1M 10d 4h).
  - Se uma unidade for zero e estiver entre duas unidades não zero, ela será omitida (por exemplo, 10d 56m em vez de 10d 0h 56m).

Quando *Units* são usados, os seguintes sufixos multiplicadores são aplicados automaticamente aos valores de item:
 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 unidade, use o prefixo `!` (por exemplo, !B ou !s).

Os exemplos a seguir mostram como os valores de item 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 fixa de unidades bloqueadas, composta por `ms`, `rpm`, `RPM`, `%`. Essa lista foi descontinuada; portanto, a forma correta de impedir a conversão dessas unidades é `!ms`, `!rpm`, `!RPM`, `!%`.

Execução de comandos

O Zabbix usa funcionalidade comum para verificações externas, user parameters, 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 de 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 server do Zabbix e no proxy do Zabbix.
- Qualquer código de saída diferente de 0 é considerado uma falha de execução.
- O conteúdo da saída de erro padrão e da saída padrão para execuções com falha é coletado e fica disponível no frontend (onde o resultado da execução é exibido).
- Uma entrada de log adicional pode ser criada para comandos remotos executados no agent/proxy do Zabbix ao habilitar o parâmetro LogRemoteCommands em agent/proxy.

Possíveis mensagens do frontend e entradas de log para comandos/scripts com falha:

- O conteúdo da saída de erro padrão e da saída padrão para execuções com falha (se houver).
- ”Process exited with code: N.” (para saída vazia e código de saída não igual a 0).
- ”Process killed by signal: N.” (para processo encerrado por um sinal, apenas no Linux).
- ”Process terminated unexpectedly.” (para processo encerrado por motivos desconhecidos).

Veja também

- Verificações externas
- Parâmetros de usuário
- items system.run
- Scripts de alerta personalizados
- Comandos remotos
- Scripts globais

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+)$<br>` Chave do item: `net.tcp.service[tcp,,"{{$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]+)<br>` 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]+)<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` 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]+<br>` 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]+)<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` 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]+<br>` Macro de usuário com contexto: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Expressão: `concat("abc, "{$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])+<br>` Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Macro de usuário com contexto: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Expressão: `concat("abc, "{$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]+<br>` Macro de usuário com contexto: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Dentro da string de alguma expressão, por exemplo: `func(arg1, arg2, arg3)="{$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.

O que é o Zabbix

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

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

O Zabbix é um software que monitora diversos parâmetros de uma rede, bem como a integridade e o estado de servidores, máquinas virtuais, aplicativos, serviços, bancos de dados, sites, da nuvem e muito mais. O Zabbix utiliza um mecanismo flexível de notificações que permite aos usuários configurar alertas baseados em e-mail para praticamente qualquer evento. Isso possibilita uma reação rápida a problemas nos servidores. O Zabbix oferece excelentes recursos de geração de relatórios e visualização de dados com base nos dados armazenados. Isso torna o Zabbix ideal para o planejamento de capacidade.

O Zabbix oferece suporte tanto à coleta por polling quanto ao recebimento de traps. Todos os relatórios e estatísticas do Zabbix, assim 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 configurado corretamente, o Zabbix pode desempenhar um papel importante no monitoramento da infraestrutura de TI. Isso se aplica igualmente a pequenas organizações com poucos servidores e a grandes empresas com uma grande quantidade de servidores.

O Zabbix está disponível gratuitamente. 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 por seus parceiros em todo o mundo.

Saiba mais sobre os recursos do Zabbix.

Usuários do Zabbix Muitas organizações de diferentes portes ao redor do mundo dependem do Zabbix como uma plataforma principal de monitoramento.

Arquitetura O Zabbix consiste em vários componentes de software principais. Suas responsabilidades estão descritas abaixo.

Server

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

Armazenamento do 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 facilitar o acesso ao Zabbix de qualquer lugar e de qualquer plataforma, é fornecida a interface baseada na web. A interface faz parte do Zabbix server e, normalmente (mas não necessariamente), é executada na mesma máquina física que aquela em que o server está em execução.

Proxy

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

Agent

Os agents do 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, há dois tipos de agents disponíveis: o Zabbix agent (leve, compatível com muitas plataformas, escrito em C) e o Zabbix agent 2 (extremamente flexível, facilmente extensível com plugins, escrito em Go).

Fluxo de dados Além disso, é importante dar um passo atrás e observar 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 em *Server X* está muito alta, primeiro você deve criar uma entrada de host para *Server X*, seguida de um item para monitorar sua CPU, depois um trigger que seja ativado se a CPU estiver muito alta, seguido de uma ação que envie um e-mail. Embora isso possa parecer muitas etapas, com o uso de templates isso realmente não é. No entanto, devido a esse design, é possível criar uma configuração muito flexível.

Recursos 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 trapping quanto polling), IPMI, JMX, monitoramento VMware
- verificações personalizadas
- coleta dos dados desejados em intervalos personalizados
- realizada por server/proxy e por agents

Definições flexíveis de limiar

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

Alertas altamente configuráveis

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

Geração de gráficos em tempo real

- os itens monitorados são imediatamente plotados usando a funcionalidade de gráficos integrada

Capacidades de monitoramento web

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

Amplas opções de visualização

- capacidade de criar gráficos personalizados que podem combinar vários itens em uma única visualização
- mapas de rede
- apresentações em sequência em uma visão 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 housekeeping integrado

Configuração fácil

- adicione dispositivos monitorados como hosts
- hosts são incluídos para monitoramento assim que estão 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
- autoregistro de agent
- descoberta de sistemas de arquivos, interfaces de rede e OIDs SNMP

Frontend web rápido

- um frontend baseado na web em PHP
- acessível de qualquer lugar

- você pode navegar clicando
- log de auditoria

API do Zabbix

- a API do Zabbix 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 determinadas visualizações

Agent completo e facilmente extensível

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

Daemons binários

- escritos em C, para desempenho e baixo consumo de memória
- facilmente portáveis

Pronto para ambientes complexos

- monitoramento remoto facilitado pelo uso de um Zabbix proxy

Glossário Nesta seção, você pode aprender o significado de alguns termos usados com frequência no Zabbix.

host

- qualquer dispositivo físico ou virtual, aplicativo, 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 valores

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

trigger

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

Quando os dados recebidos estão acima do limite, os triggers passam do estado *Ok* para o estado *Problem*. Quando os dados recebidos estão abaixo do limite, os triggers permanecem ou retornam ao estado *Ok*.

template

- um conjunto de entidades (itens, triggers, gráficos, regras de descoberta de baixo nível, cenários web) pronto para ser aplicado a um ou vários hosts.

A função dos templates é acelerar a implantação de tarefas de monitoramento em um host e também facilitar a aplicação de alterações em massa às tarefas de monitoramento. Os templates são vinculados diretamente a hosts individuais.

grupo de templates

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

evento

- uma ocorrência única de algo que merece atenção, como uma mudança de estado de um trigger ou a realização de uma descoberta/autoregistro de agent.

tag de evento

- um marcador predefinido para o evento. Ele pode ser usado na correlação de eventos, na granularidade de permissões etc.

correlação de eventos

- um método de correlacionar problemas às suas resoluções 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 mesmo usar um método diferente de coleta de dados.

problema

- um trigger que está no estado "Problem".

atualização de problema

- opções de gerenciamento de problemas fornecidas pelo Zabbix, como adicionar um 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 realizada).

escalonamento

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

macro

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

mídia

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

notificação

- uma mensagem sobre algum evento enviada a um usuário por meio do canal de mídia escolhido.

comando remoto

- um comando predefinido que é executado automaticamente em um host monitorado quando determinada condição é atendida.

cenário web

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

frontend

- a interface web fornecida com o Zabbix.

dashboard

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

widget

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

API do Zabbix

- a API do Zabbix permite usar o protocolo JSON RPC para criar, atualizar e obter objetos do Zabbix (como hosts, itens, 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 e funciona como um repositório central de dados.

Zabbix proxy

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

Zabbix agent

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

Zabbix agent 2

- uma nova geração do Zabbix agent para monitorar ativamente recursos e aplicativos 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 representa tanto o Zabbix agent quanto o Zabbix agent 2 quando o comportamento funcional é o mesmo. O Zabbix agent 2 é mencionado especificamente apenas quando sua funcionalidade é diferente.

criptografia

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

autoregistro de agent

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

descoberta de rede

- *descoberta automatizada de dispositivos de rede.*

descoberta de baixo nível

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

regra de descoberta de baixo nível

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

protótipo de item

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

protótipo de trigger

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

Zabbix Cloud

Zabbix Cloud

Descubra as opções de configuração disponíveis e os controles de acesso para organizações e nós no Zabbix Cloud.

Observe que os nós do Zabbix Cloud atualmente executam o **Zabbix 7.0**.
 Para informações detalhadas sobre o Zabbix 7.0, consulte a documentação oficial.

Configurar e gerenciar o Zabbix Cloud

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

Configuração do nó Aprenda como configurar seu nó Zabbix Cloud, com foco em acesso, manipulação de dados e configurações de recursos.

Adicionando usuários Descubra como adicionar e atribuir funções a usuários dentro de uma organização Zabbix Cloud e seus nós.

Saiba mais sobre o Zabbix Cloud

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

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

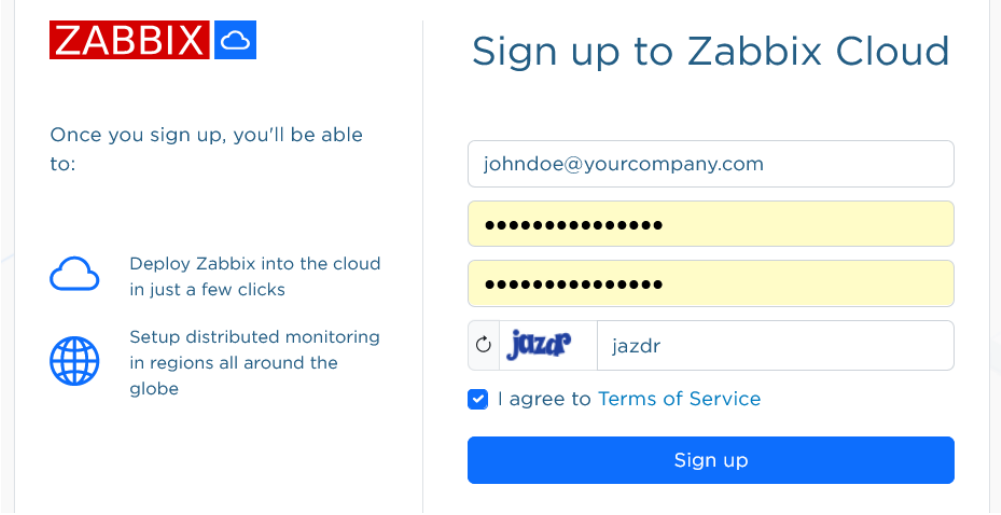
Implante o Zabbix na nuvem

Para iniciar o Zabbix na nuvem, siga este processo:

1. Faça login ou cadastre-se em cloud.zabbix.com.
2. Crie um nó de nuvem.
3. Comece a usar o Zabbix na nuvem.

Fazer login ou cadastrar-se Você pode fazer login em cloud.zabbix.com com uma conta existente do Google, GitHub ou Microsoft.

Como alternativa, acesse a página de cadastro para se registrar.



Após o cadastro, você receberá um e-mail confirmando a criação da sua nova conta.

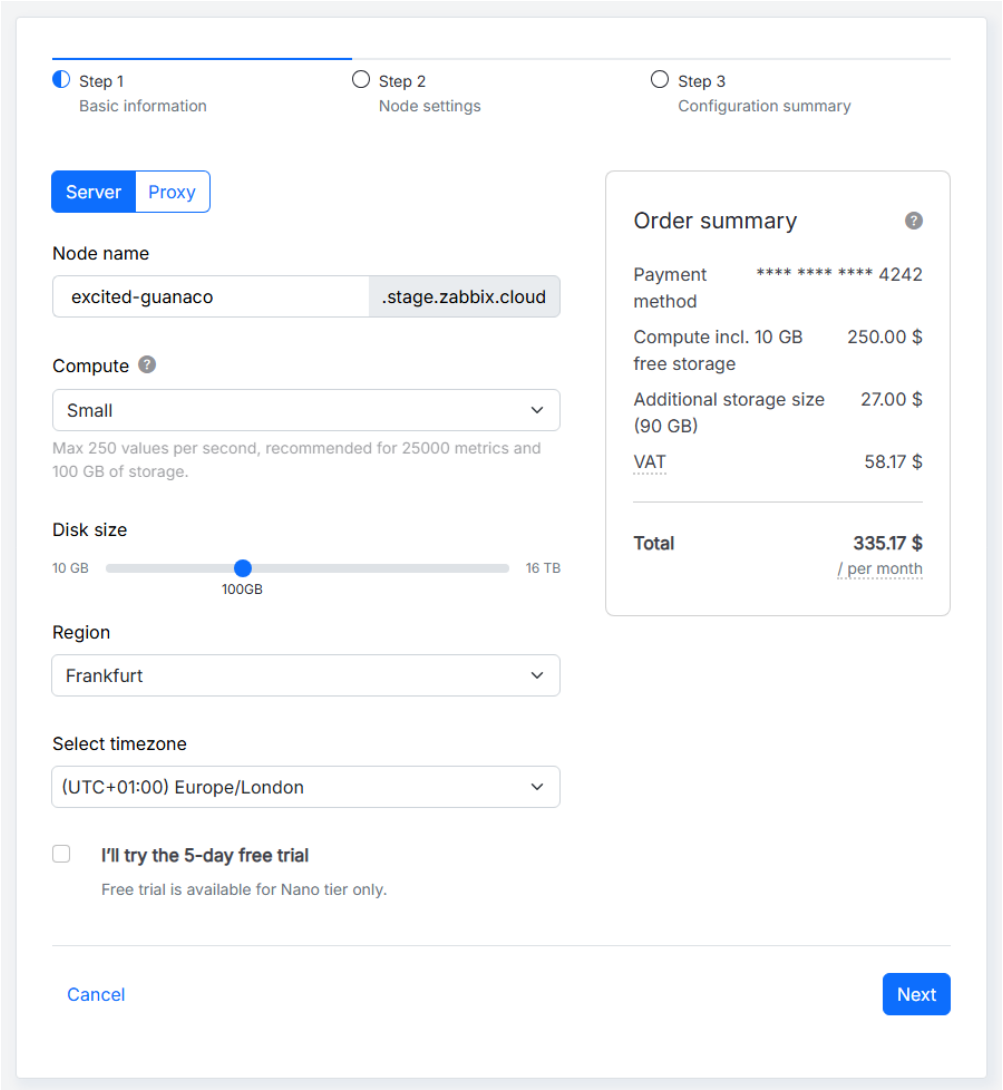
Acesse cloud.zabbix.com para fazer login. Será solicitada uma senha de uso único durante o login. Essa senha será enviada a você por e-mail.

Criar um nó de nuvem Um nó de nuvem é uma instância do Zabbix implantada no Zabbix Cloud. Ele pode ser implantado como um nó de server ou um nó de proxy.

Quando estiver conectado, você verá que uma organização *Private* já existe para sua conta por padrão. Você poderá renomear a organização padrão posteriormente.

Clique em *Create New Node*. Um formulário de configuração do nó, dividido em três etapas, será aberto.

Na primeira etapa, você selecionará as informações básicas.



Parâmetro	Descrição
Server/Proxy	Selecione se deseja implantar uma instância de server do Zabbix ou uma instância de proxy do Zabbix.
Node name	O nome do nó forma um subdomínio exclusivo no formato `<your-name>.zabbix.cloud`, usado para acessar os serviços do nó.
Compute	Selecione a capacidade computacional do nó. Para determinar o nível de recursos necessário, consulte os detalhes de preços. Você sempre poderá fazer um upgrade posteriormente se precisar de mais capacidade. Será necessário um método de cobrança antes de criar um nó que não seja de avaliação. Adicione as informações de cobrança pelo menu *Billing*. Está disponível uma **avaliação gratuita** de 5 dias para testar um nó da categoria *Nano*. Nenhuma cobrança será feita durante o período de avaliação, mas, quando ele terminar, seu nó será removido caso não seja convertido em um nó pago.
Disk size	O controle deslizante usa como padrão um mínimo recomendado com base na categoria. Aumente esse valor se você armazenar grandes volumes de histórico ou tendências. Observe que discos maiores têm 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.
Region	Selecione a região onde o nó será implantado.
Select timezone	Selecione o fuso horário do server do Zabbix. Esse fuso horário é usado pelo server do Zabbix e é independente dos fusos horários de manutenção e do perfil do usuário.

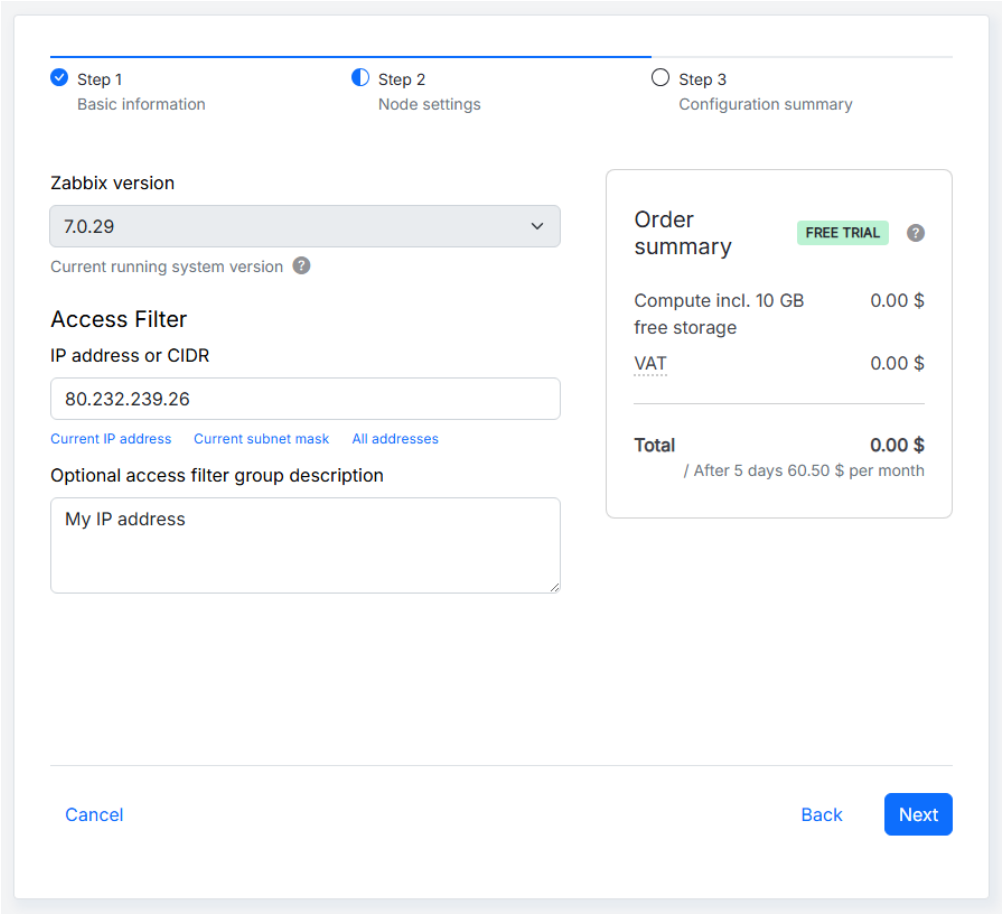
Depois de concluir a configuração, clique em *Next*.

Na segunda etapa, você selecionará as configurações do nó.

As opções de configuração da segunda etapa dependem de você estar criando um nó de server ou de proxy.

Configurar um nó de server

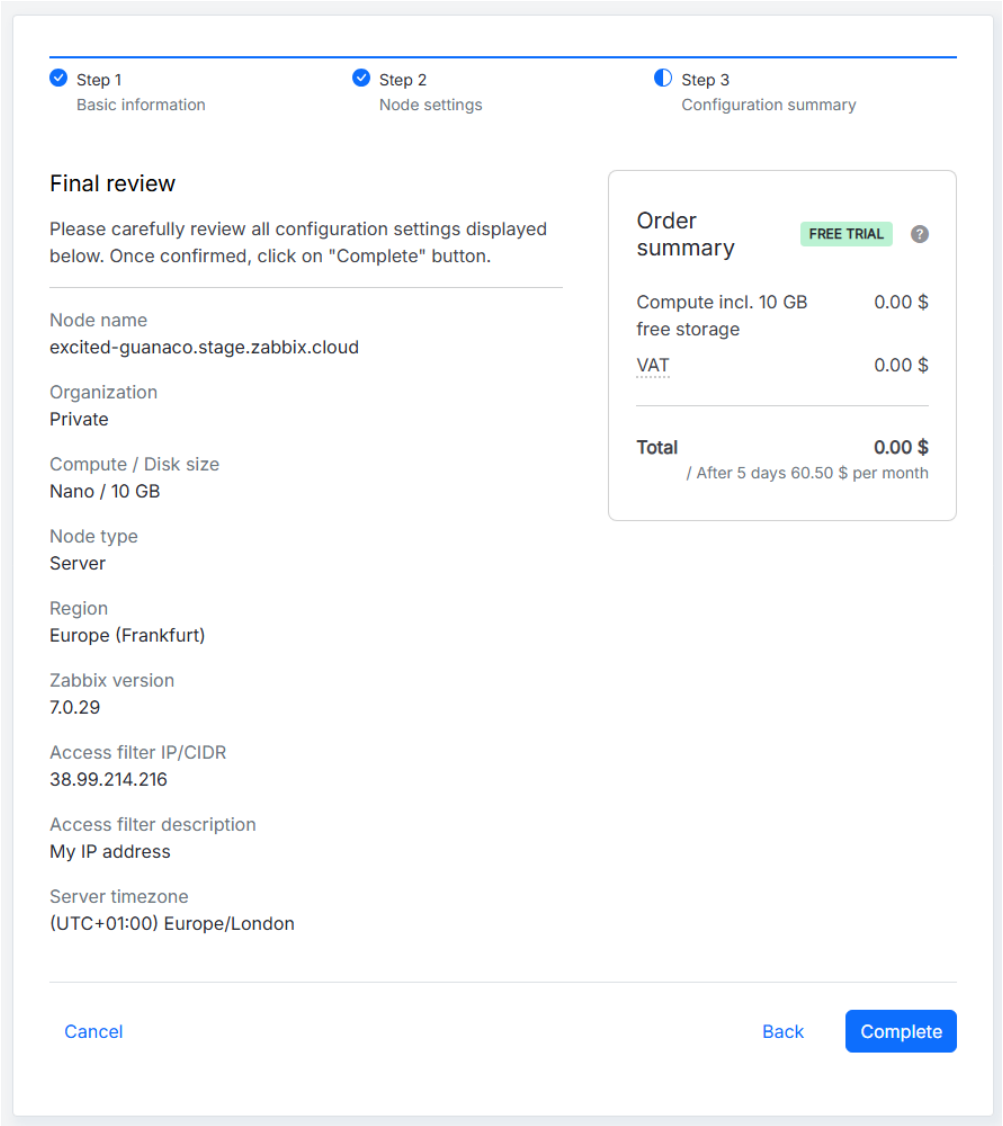
Durante a criação de um nó de server, os seguintes parâmetros estão disponíveis:



Parâmetro	Descrição
Zabbix version	Selecione a versão do Zabbix. O Zabbix Cloud oferece suporte à versão LTS mais recente e aplica automaticamente atualizações de versões secundárias.
IP address or CIDR	Especifique o filtro de acesso inicial do nó. Insira o endereço IP ou CIDR manualmente ou preencha o campo automaticamente selecionando a opção correspondente. Para nós de server, esse filtro se aplica às conexões com o server, a API e o frontend. O filtro pode ser alterado posteriormente na aba Access filters.
Optional access filter group description	Adicione uma descrição opcional para o grupo de filtros de acesso, que pode conter vários endereços IP ou CIDRs.

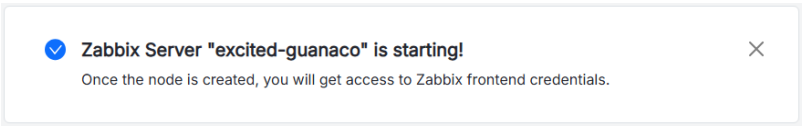
Warning:
Durante a criação do nó, o campo *IP address or CIDR* é preenchido previamente com o endereço IP usado para acessar o Zabbix Cloud. Você pode alterar esse valor para especificar um filtro de acesso inicial diferente. Depois que o nó for criado, você poderá adicionar, editar ou excluir filtros de acesso na aba *Access filters*.

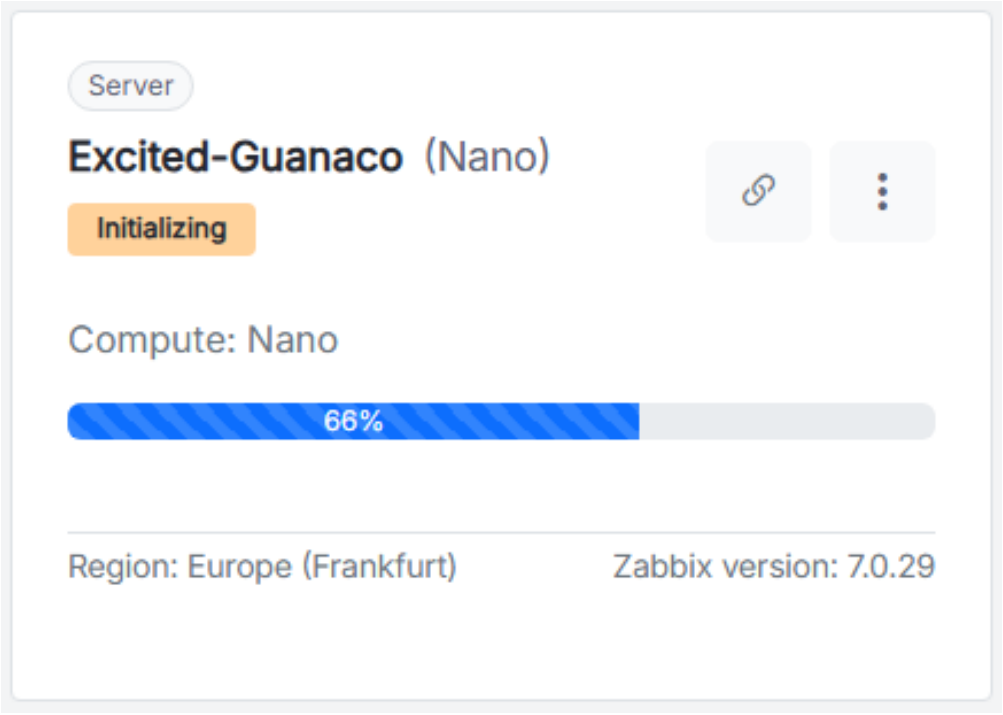
A terceira etapa exibe um resumo da configuração do nó.



Depois de revisar as configurações e confirmar que tudo está correto, clique em *Complete* para um nó de avaliação ou em *Proceed to payment* para um nó pago. Caso contrário, retorne a uma etapa anterior clicando no botão *Back* ou clique em *Cancel* para começar novamente.

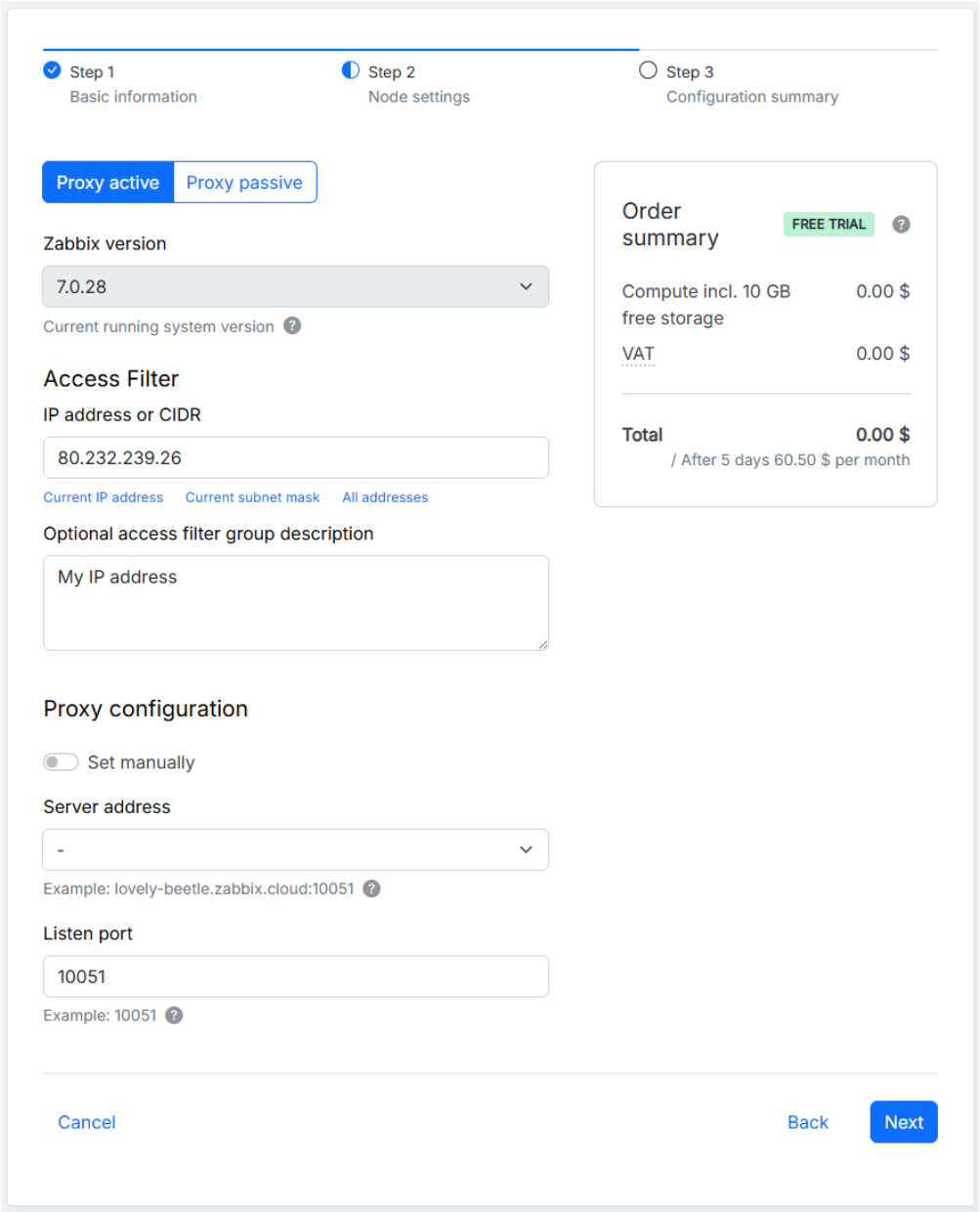
Enquanto o nó é inicializado, uma mensagem correspondente é exibida, e o nó já aparece em *Organizations*.





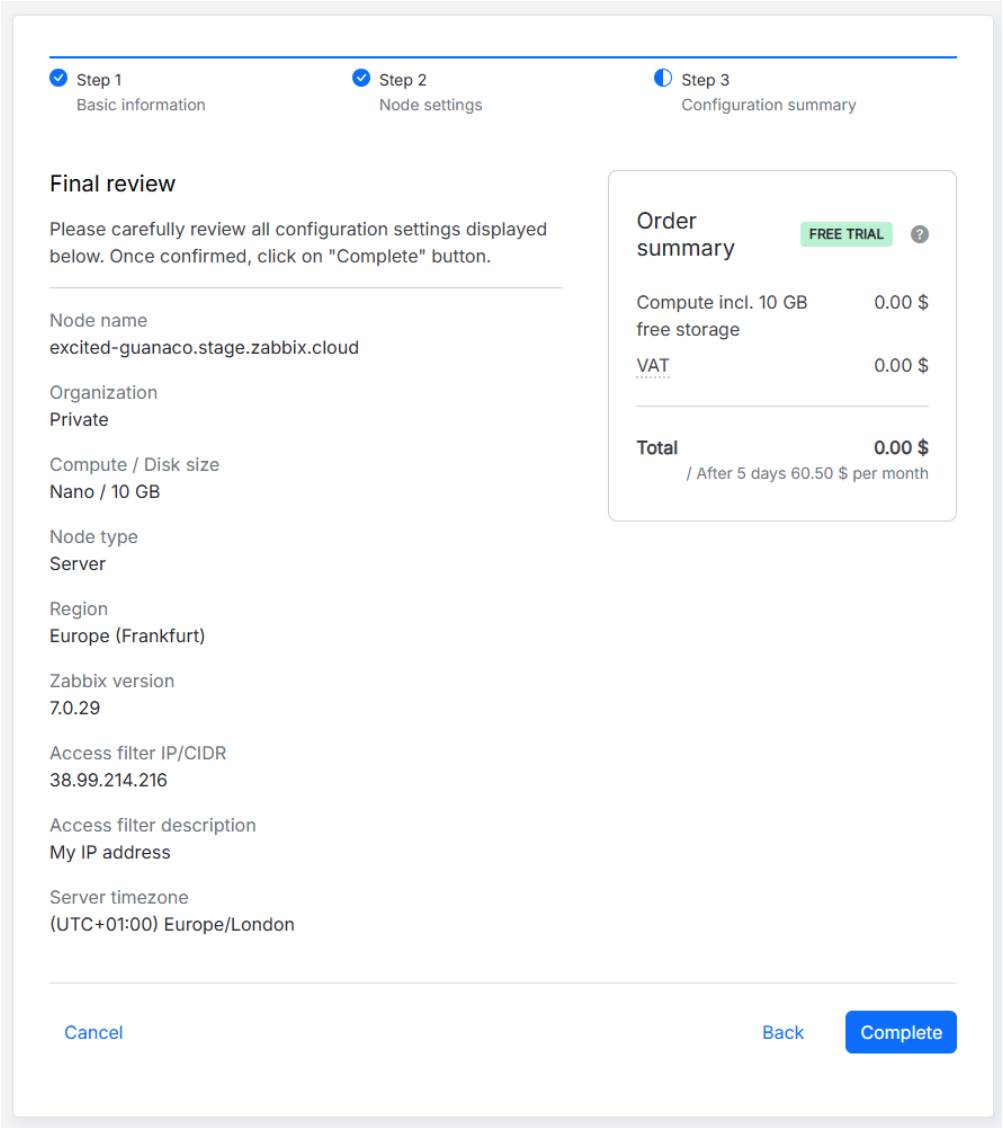
Configurar um nó de proxy

Durante a criação do nó de proxy, os seguintes parâmetros estão disponíveis:



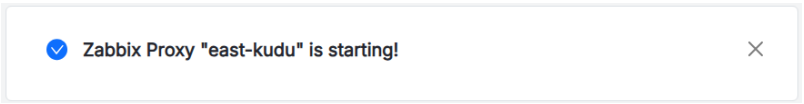
Parâmetro	Descrição
Proxy active/Proxy passive	Selecione o modo do proxy. No modo ativo, o proxy inicia as conexões com o server do Zabbix; no modo passivo, o server do Zabbix inicia as conexões com o proxy.
Zabbix version	Selecione a versão do Zabbix. O Zabbix Cloud oferece suporte à versão LTS mais recente e aplica automaticamente atualizações de versões secundárias.
IP address or CIDR	Especifique o filtro de acesso IP de entrada inicial do proxy. Insira o endereço IP ou CIDR manualmente ou preencha o campo automaticamente selecionando a opção correspondente. O filtro define quais solicitações de rede podem chegar ao proxy e pode ser alterado posteriormente na aba Access filters.
Optional access filter group description	Adicione uma descrição opcional para o grupo de filtros de acesso, que pode conter vários endereços IP ou CIDRs.
Set manually	Selecione como especificar o server do Zabbix. Quando desabilitado, selecione um nó de server na lista de nós de server disponíveis para você. Quando habilitado, especifique manualmente os endereços do server.
Server address	Especifique o endereço do server do Zabbix. Quando *Set manually* está desabilitado, selecione um nó de server na lista. Quando habilitado, especifique até 10 endereços de server. Para um proxy ativo, especifique o endereço IP ou o nome DNS do server do Zabbix. Para um proxy passivo, especifique o endereço IP, o nome DNS ou o CIDR do server do Zabbix. Os filtros de acesso são configurados separadamente e não são atualizados automaticamente quando um endereço de server é especificado.
Listen port	Especifique a porta na qual o proxy escuta conexões de entrada. A porta padrão é 10051. A porta de escuta não pode ser alterada após a criação do nó.

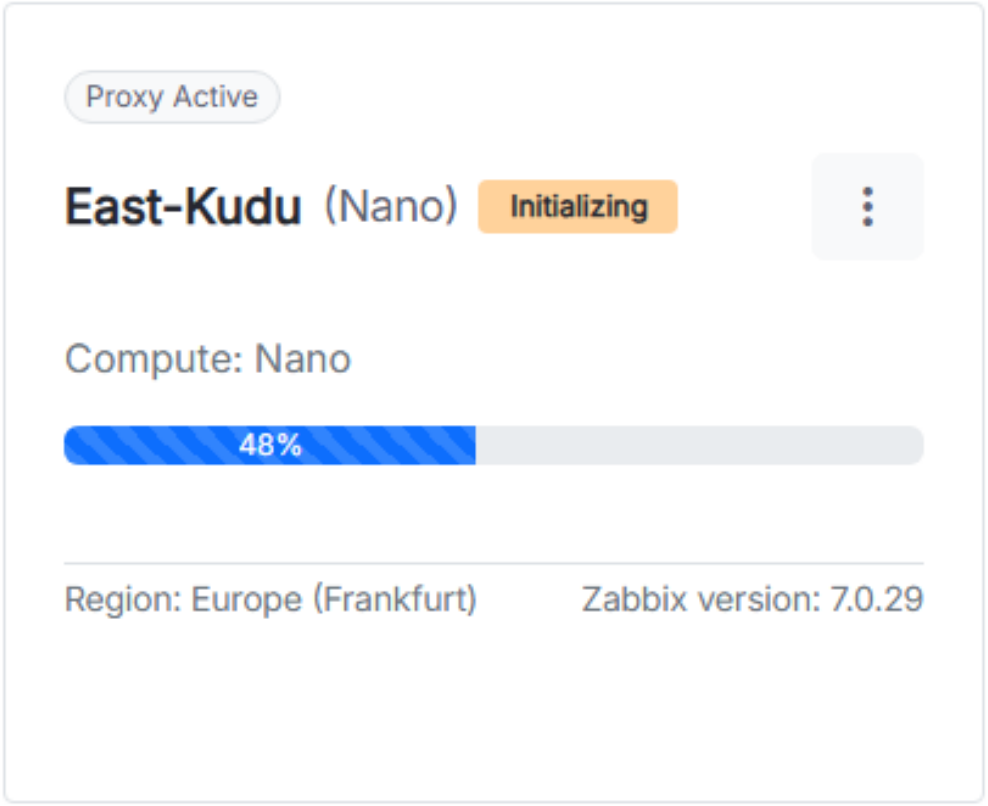
Na terceira etapa, você verá o resumo da configuração do nó.



Depois de revisar as configurações e confirmar que tudo está correto, clique em *Complete* para um nó de avaliação ou em *Proceed to payment* para um nó pago. Caso contrário, retorne às etapas anteriores clicando no botão *Back* ou clique em *Cancel* para começar novamente.

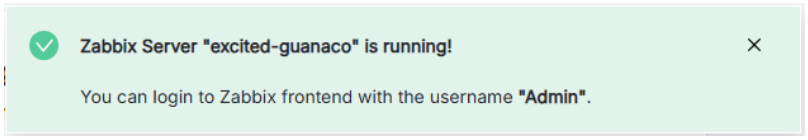
Enquanto o nó é inicializado, uma mensagem correspondente é exibida, e o nó já aparece em *Organizations*.





Para configurar um proxy do Zabbix Cloud, consulte Configurar um proxy do Zabbix Cloud.

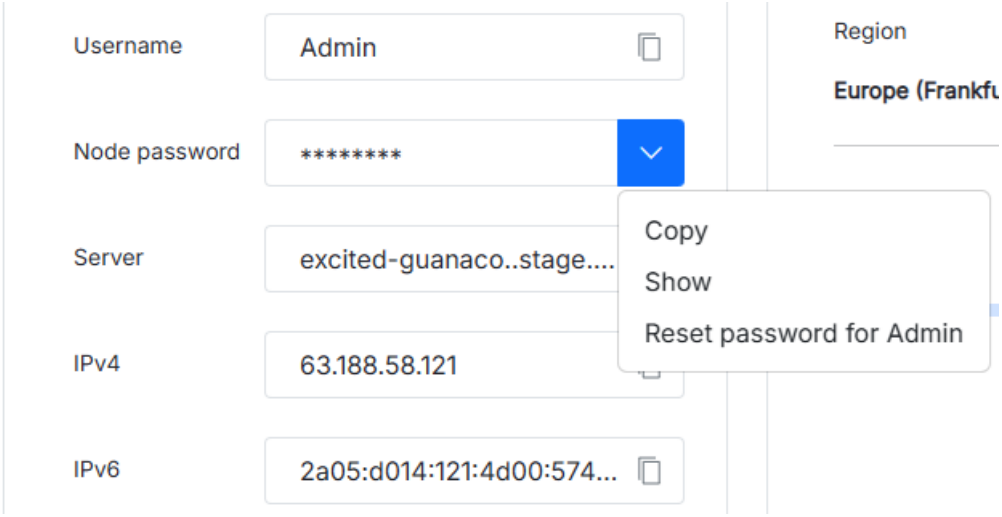
Comece a usar o Zabbix



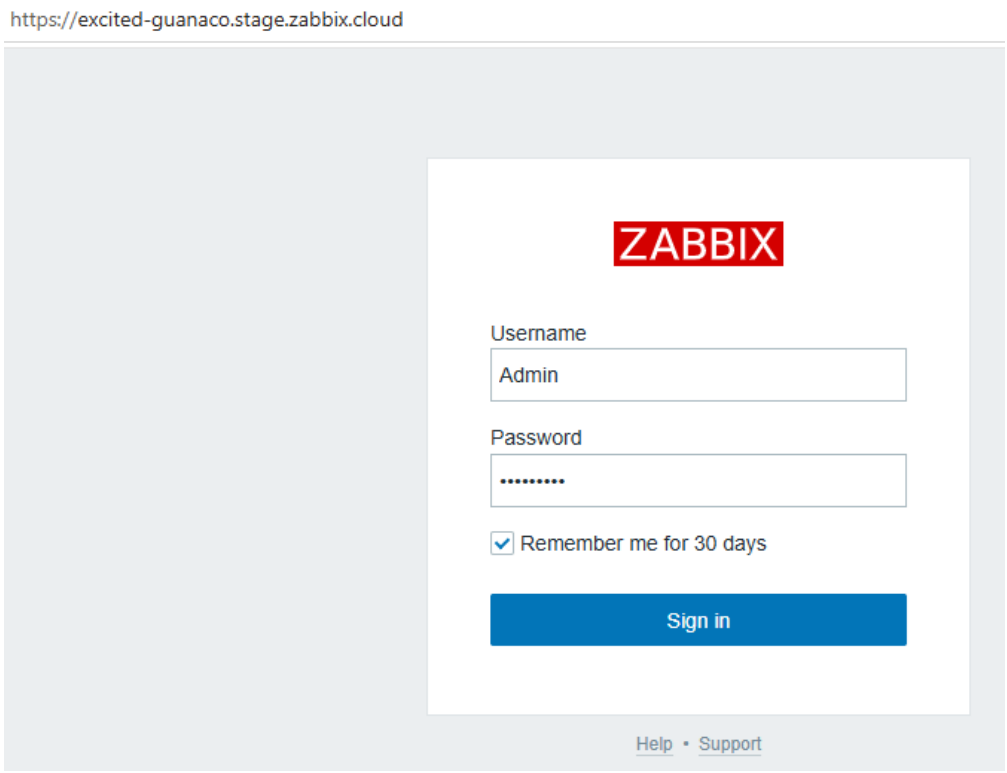
Quando o status do server node mudar de *Initializing* para *Running*, você poderá fazer login no frontend do Zabbix.

Para acessar a senha inicial do frontend, clique no node criado.

Você será direcionado para as configurações do node (visão geral, filtros de acesso, criptografia, backups etc.). Em **Connection info**, localize o campo *Node password* e selecione *Reset password for Admin* no menu suspenso. Depois que a senha for redefinida, clique no mesmo menu suspenso e selecione *Copy*.



Clique na URL do frontend para abrir o frontend do Zabbix (`excited-guanaco.zabbix.cloud`).



Faça login usando as seguintes credenciais:

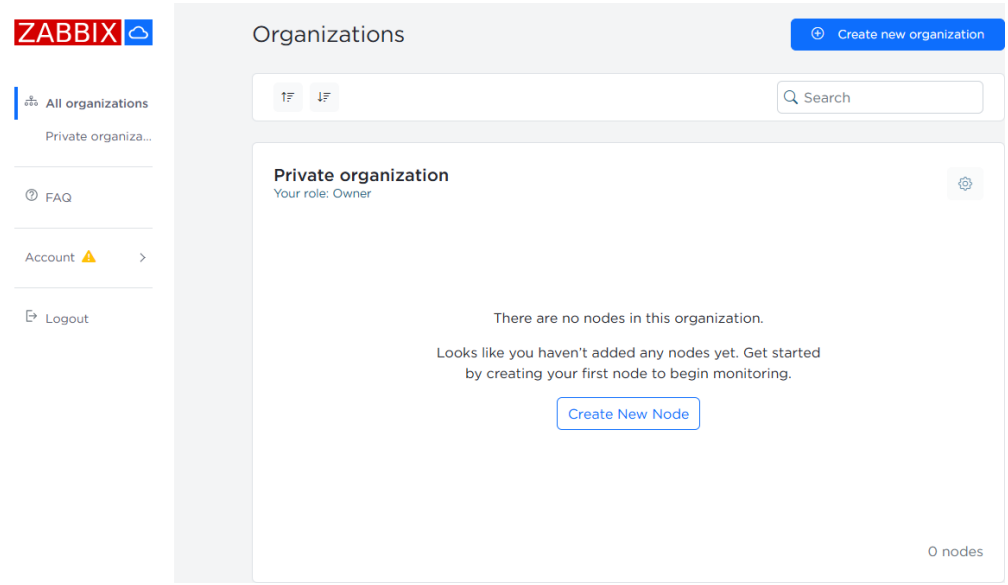
- *Username* - `Admin`
- *Password* - forneça a senha copiada

Para iniciar o monitoramento, você precisa definir filtros de acesso - o endereço IP ou os endereços IP que têm permissão para se conectar ao server do Zabbix Cloud. Pelo menos um deve ser definido.

Gerenciamento de organizações Uma organização é um contêiner para um ou mais nodes. As organizações também ajudam a estruturar seus recursos e delegar responsabilidades, como:

- Gerenciamento de nodes
- Faturamento

Ao fazer login no Zabbix Cloud, a página *Todas as organizações* é exibida. Por padrão, uma organização privada é criada para sua conta.

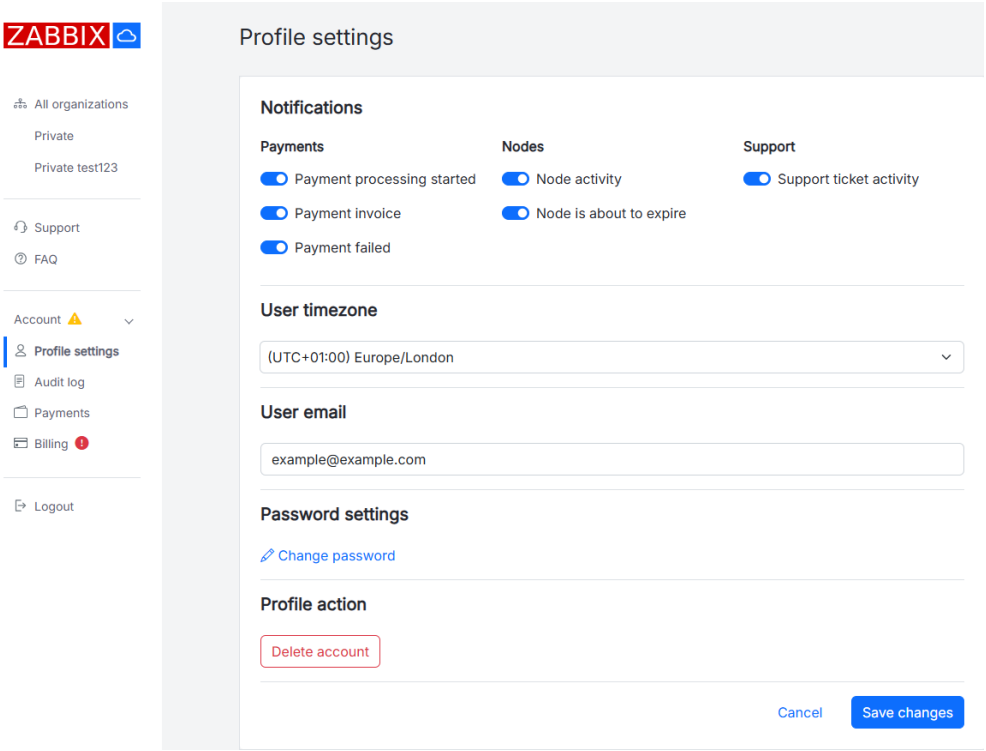


Como proprietário desta organização, você pode gerenciar usuários, administradores, nodes e o nome da organização. Clique no botão de configurações  para acessar as opções de gerenciamento.

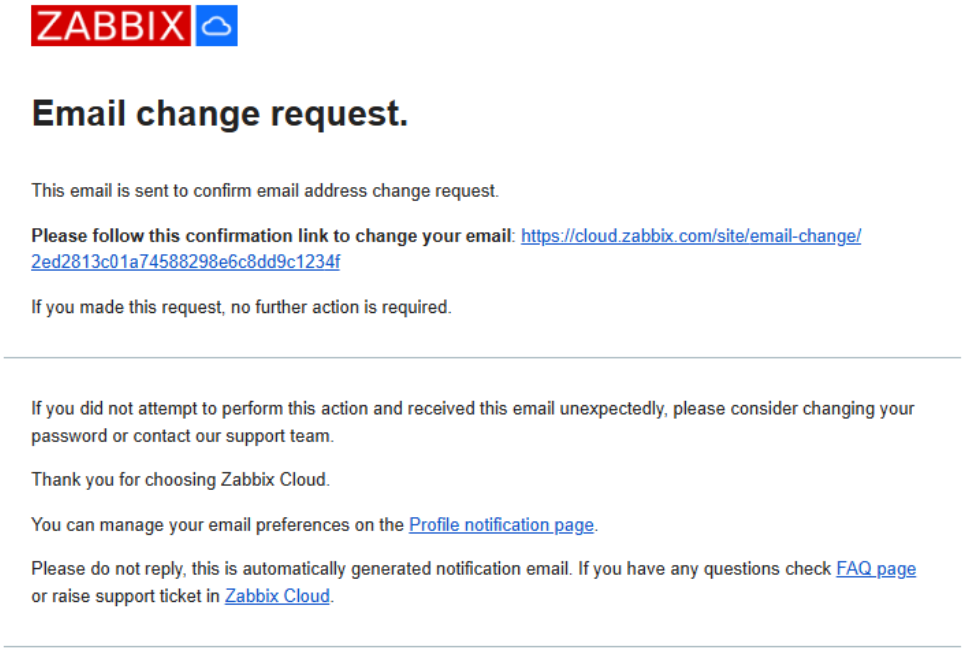
Gerenciamento de contas Alterando o endereço de e-mail da conta

Para alterar o endereço de e-mail da conta, acesse *Account* > *Profile settings*.

No campo *User email*, insira o novo endereço de e-mail e clique em *Save changes*.



Uma mensagem de confirmação é enviada para o novo endereço de e-mail. Siga o link na mensagem para confirmar a alteração do endereço de e-mail.



O link de confirmação abre a página *Profile settings* com uma mensagem de confirmação e o endereço de e-mail atualizado no campo User email.

Após a confirmação da alteração, um e-mail confirmando a alteração é enviado para os endereços de e-mail novo e anterior. A mensagem enviada para o endereço anterior inclui os endereços de e-mail anterior e novo.

Configuração do nó

Visão geral

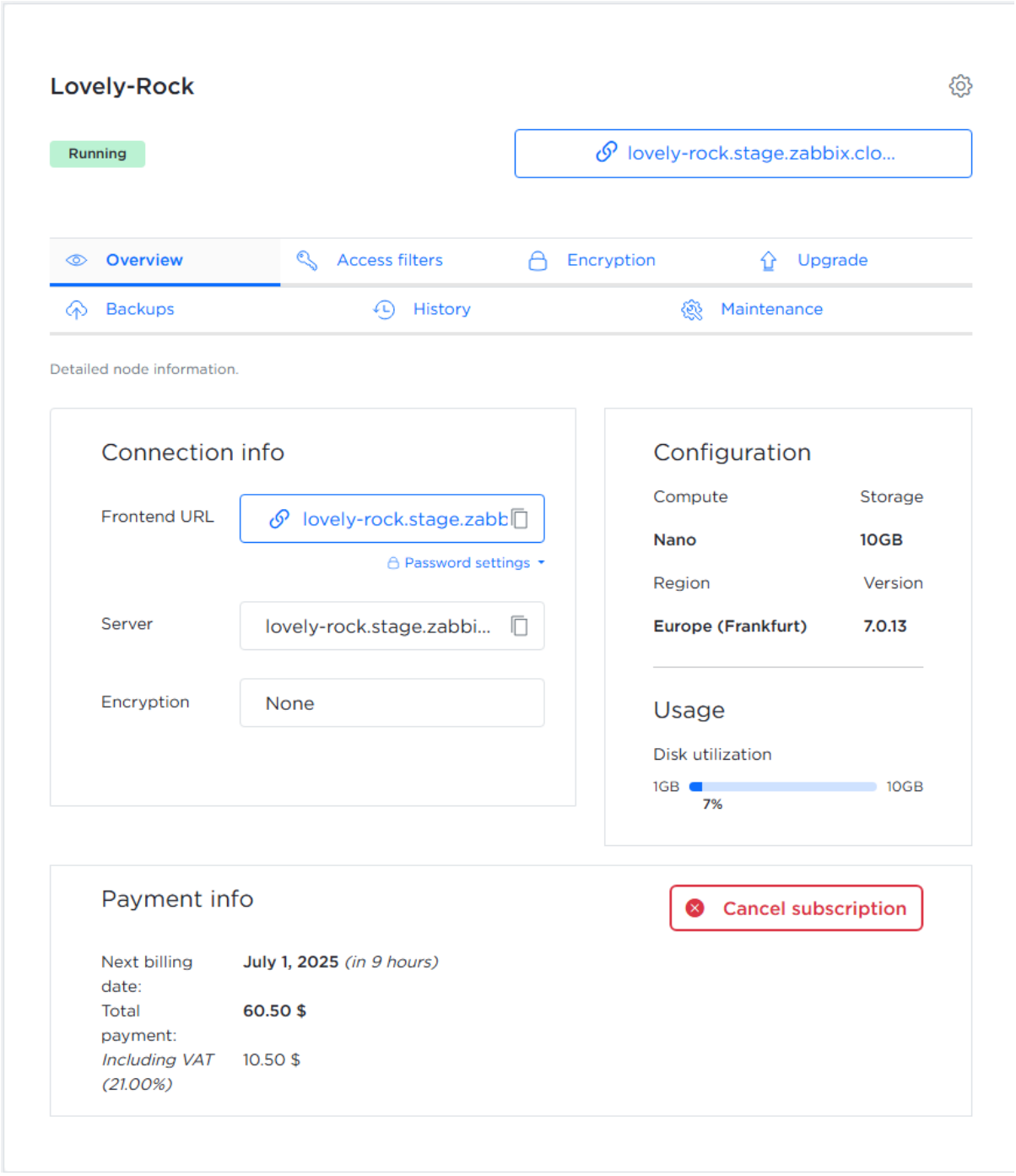
Um **cloud node** é um componente do Zabbix implantado na nuvem. Os nodes do server incluem o Zabbix server, o banco de dados e o web frontend, enquanto os nodes do proxy incluem o Zabbix proxy e o banco de dados. Um cloud node possui recursos computacionais dedicados (CPU, memória) e armazenamento em disco.

Os usuários podem criar um ou mais cloud nodes. Os cloud nodes podem ser agrupados em **organizações** (estruturais, regionais etc.), criando uma configuração de monitoramento distribuído.

Os cloud nodes e as organizações são criados usando a interface do usuário do Zabbix Cloud. Para acessar a página de configuração do node, clique no node em *Todas as organizações*, no menu da sua organização ou no link *Página de configuração do node* exibido depois que o node é iniciado.

Na visualização de configuração do node, cada aba permite gerenciar um aspecto específico do seu node:

- Visão geral - visualizar os detalhes do node, abrir o frontend e gerenciar os usuários do node.
- Filtros de acesso - controlar o acesso por endereço IP ou sub-rede.
- Criptografia - configurar TLS carregando certificados, chaves e CRLs opcionais.
- Backups - criar, restaurar e excluir snapshots do node; visualizar o próximo backup agendado.
- Histórico - definir períodos de retenção para logs de auditoria, histórico e tendências, com unidades predefinidas e um limite de atualização de 3 vezes por dia.
- Manutenção - definir janelas semanais de manutenção por fuso horário, dia e hora.
- Upgrade - dimensionar os recursos de computação/armazenamento ou gerenciar o status da avaliação/pagamento.

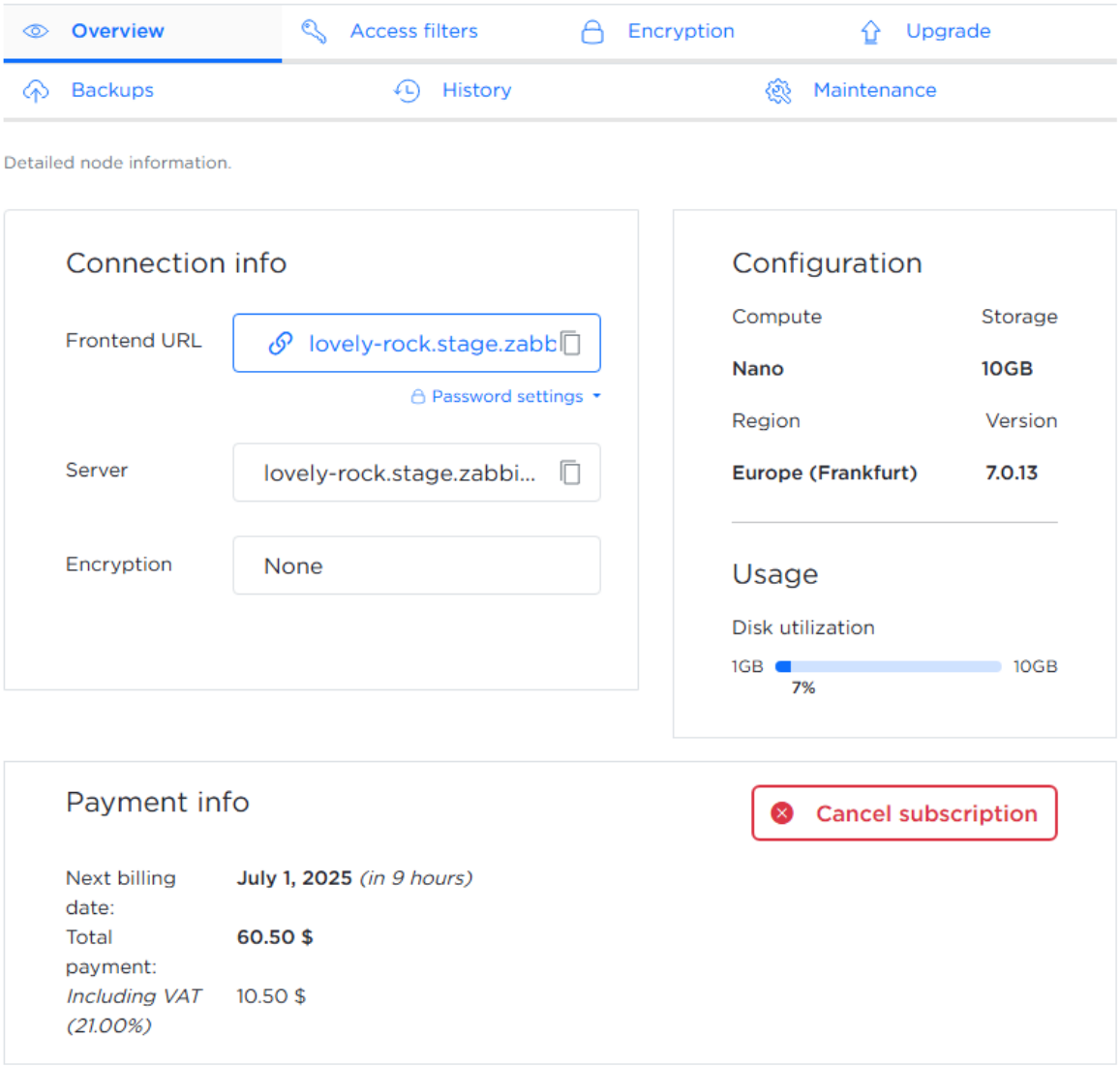


Note:
Algumas abas podem estar indisponíveis dependendo do tipo de node (server ou proxy), do tipo de usuário que visualiza a página e do status do node (avaliação/não avaliação).

Aba Visão geral

A aba *Visão geral* fornece informações detalhadas sobre o node. As informações exibidas dependem de o node ser um server ou proxy.

Nó do server A aba *Visão geral* do nó do server contém as seguintes seções:



Informações de conexão:

- URL clicável do frontend, com opção de copiar para a área de transferência
- Nome de usuário
- Senha do nó e suas configurações
- Nome do host do server, com opção de copiar para a área de transferência
- Endereço IPv4, com opção de copiar para a área de transferência
- Endereço IPv6, com opção de copiar para a área de transferência
- Método de criptografia atual

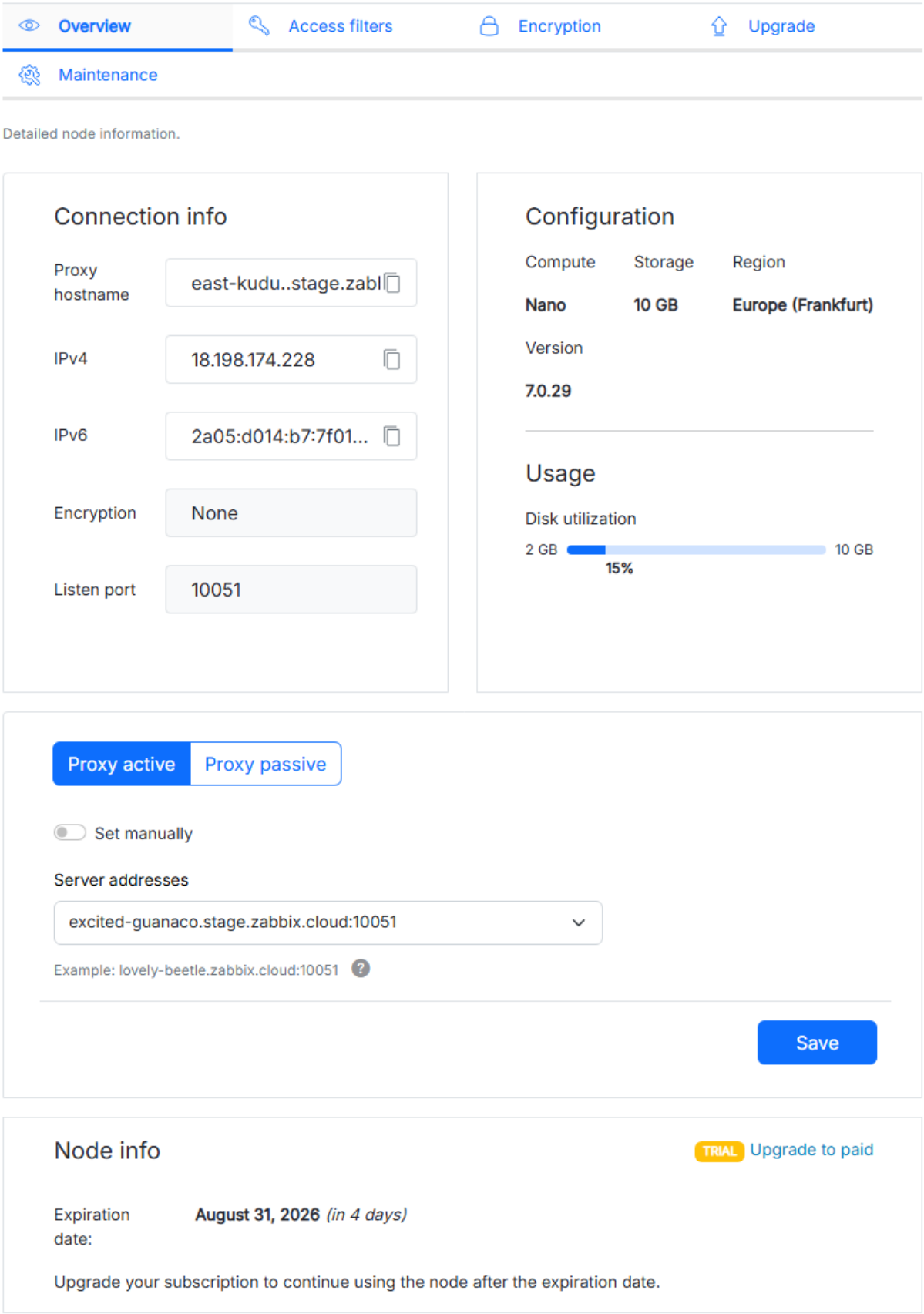
Configuração e uso:

- Camada de computação, armazenamento máximo e 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 específico ao contexto é exibido, dependendo do tipo de nó, após o nó concluir a inicialização:
  - *Fazer upgrade para pago* - disponível para nós em período de avaliação gratuito
  - *Cancelar assinatura* - disponível para nós pagos com uma assinatura ativa
  - *Assinar novamente o plano pago* - exibido quando uma assinatura foi cancelada
  - Nenhum botão de ação é exibido para nós pré-pagos, que têm validade longa e não podem ser cancelados

Proxy node A aba *Overview* do proxy node contém as seguintes seções:



Connection info:

- Nome do host do proxy, com uma opção para copiar para a área de transferência
- Endereço IPv4, com uma opção para copiar para a área de transferência
- Endereço IPv6, com uma opção para copiar para a área de transferência
- Método de criptografia atual
- Porta de escuta atual

Configuration and usage:

- Camada de computação, armazenamento máximo, região
- Versão dos componentes do Zabbix
- Utilização do disco

Proxy mode:

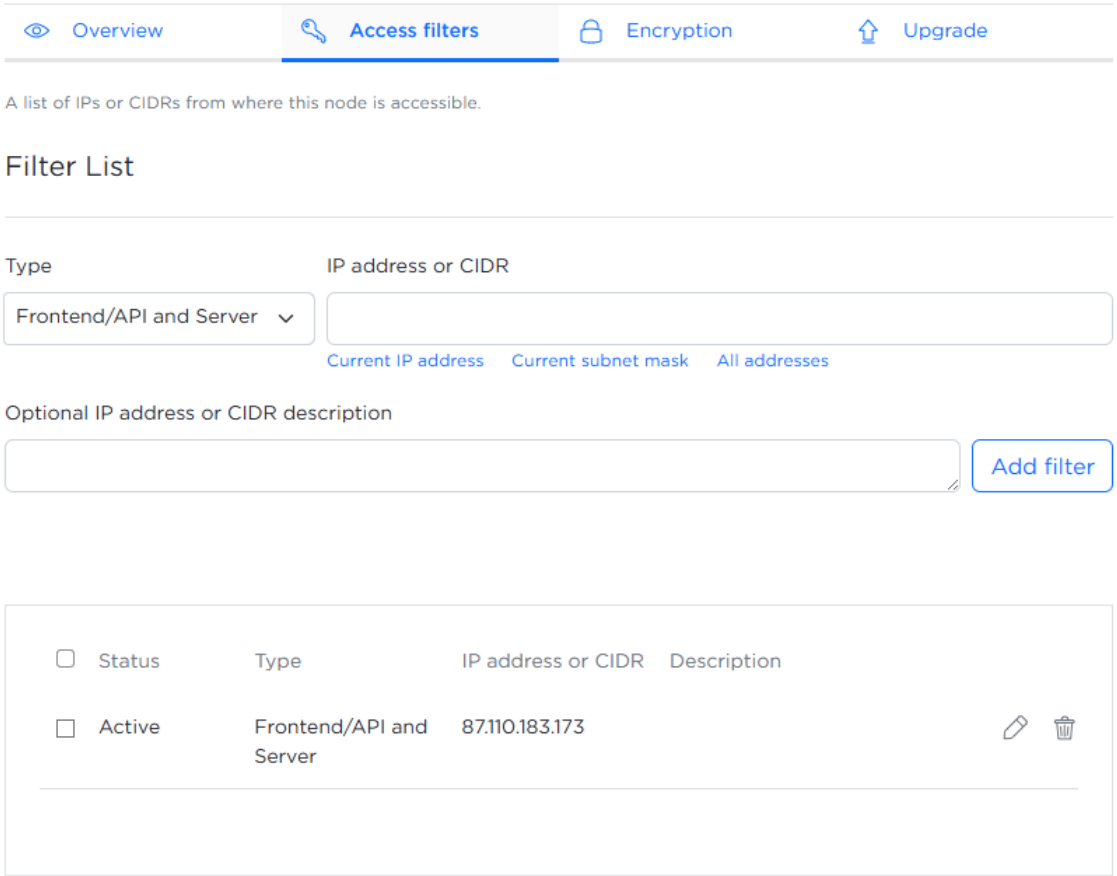
- Alternar entre os modos ativo e passivo do proxy
- Selecione o endereço do server no menu suspenso *Server addresses* ou habilite *Set manually* para inserir o endereço do server manualmente

Node info/Payment Info:

- Data de expiração (para nodes de avaliação e pré-pagos)
- Um botão específico ao contexto é exibido de acordo com o tipo de node após a conclusão da inicialização do node: - *Upgrade to paid* - disponível para nodes em uma avaliação gratuita - *Cancel subscription* - disponível para nodes pagos com uma assinatura ativa - *Resubscribe to paid* - exibido se uma assinatura tiver sido cancelada - Nenhum botão de ação é exibido para nodes pré-pagos, que têm validade longa e não podem ser cancelados

Guia de filtros de acesso

A guia *Filtros de acesso* permite restringir o acesso ao seu nó do Zabbix por endereço IP ou sub-rede.



Para configurar um filtro de acesso, preencha os seguintes campos:

- *Tipo* - para nós server, use a lista suspensa para escolher qual componente do Zabbix limitar: somente frontend/API, somente server ou ambos. Para nós proxy, somente *IP de entrada* está disponível. Isso define quais solicitações de rede recebidas podem chegar ao proxy.
- *Endereço IP ou CIDR* - insira o endereço IP ou CIDR manualmente ou preencha o campo automaticamente selecionando a opção relevante.
- *Descrição opcional do endereço IP ou CIDR* - adicione uma descrição opcional para cada IP ou sub-rede.

Depois de preencher os campos, clique em *Adicionar filtro*. Os filtros adicionados aparecem na tabela abaixo, mostrando o status, o tipo, o endereço IP ou a sub-rede e a descrição.

Warning:
Durante a inicialização do nó, um filtro de acesso é criado automaticamente para o endereço IP usado. Se você se conectar posteriormente a partir de um IP ou rede diferente, deverá adicionar esse endereço aqui; caso contrário, o acesso será bloqueado.

Você pode:

-  - editar um filtro.
-  - excluir um filtro ou vários filtros usando as caixas de seleção.

Aba Encryption

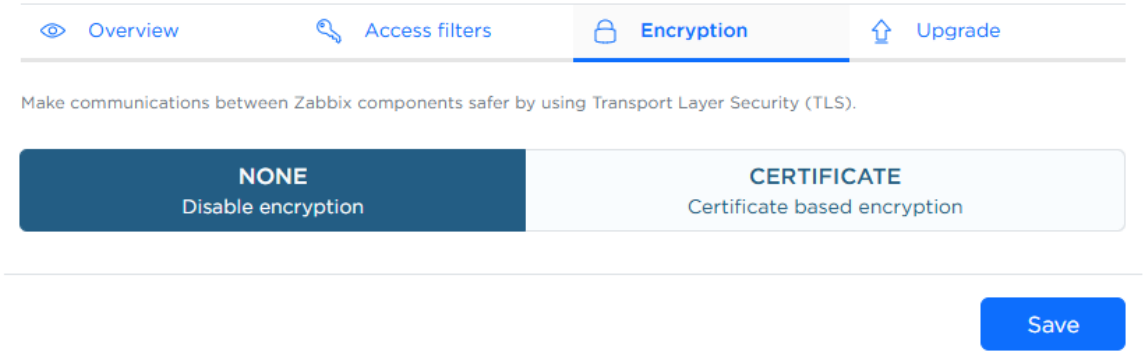
A aba *Encryption* permite configurar a criptografia TLS para a comunicação entre os componentes do Zabbix fazendo upload dos arquivos de certificado necessários.

Para habilitar a comunicação segura usando certificados, selecione *CERTIFICATE* e faça upload dos seguintes arquivos do seu computador:

- *Certificados da CA raiz* - certificados de CA confiáveis usados para verificar os certificados dos pares. Clique em *Escolher arquivo* e selecione o arquivo que contém os certificados da CA raiz confiáveis (normalmente um arquivo `.crt` ou `.pem`, codificado em PEM).
- *Cadeia de certificados* - representa a cadeia de confiança do seu próprio certificado. Clique em *Escolher arquivo* e selecione o arquivo codificado em PEM (por exemplo, `.crt` ou `.pem`) que inclui seu certificado e quaisquer certificados de CA intermediários.

- *Chave privada* - a chave privada correspondente ao seu certificado. Clique em *Escolher arquivo* e selecione o arquivo `.key` codificado em PEM.
- *Certificados revogados* (opcional) - se você mantiver uma Lista de Revogação de Certificados (CRL), clique em *Escolher arquivo* e selecione o arquivo CRL para especificar os certificados que não devem mais ser considerados confiáveis.

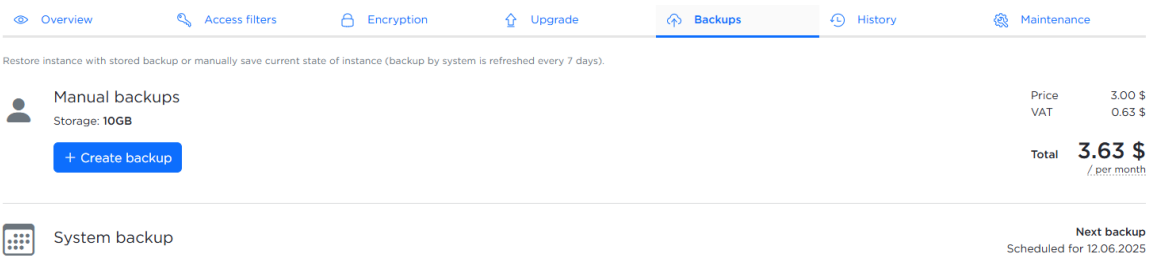
Depois de selecionar os arquivos, clique em *Salvar*. Os arquivos serão enviados, validados e aplicados para configurar a criptografia TLS entre os componentes do Zabbix.



Aba Backups

A aba *Backups* permite visualizar, criar, restaurar e excluir backups do seu nó do 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:

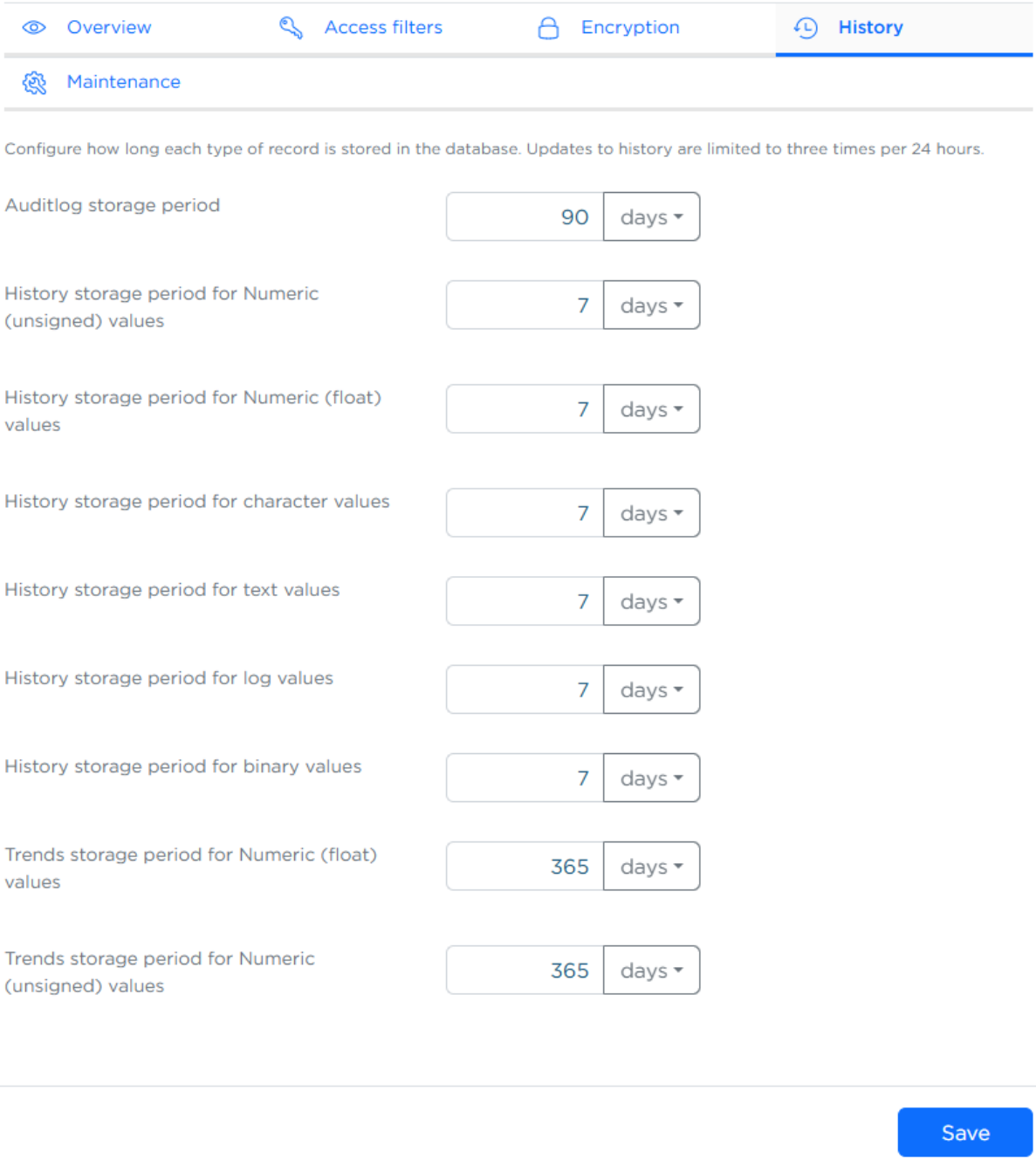
1. Clique no botão *Create backup*.
2. Insira uma descrição opcional na janela pop-up.

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 serão armazenados.



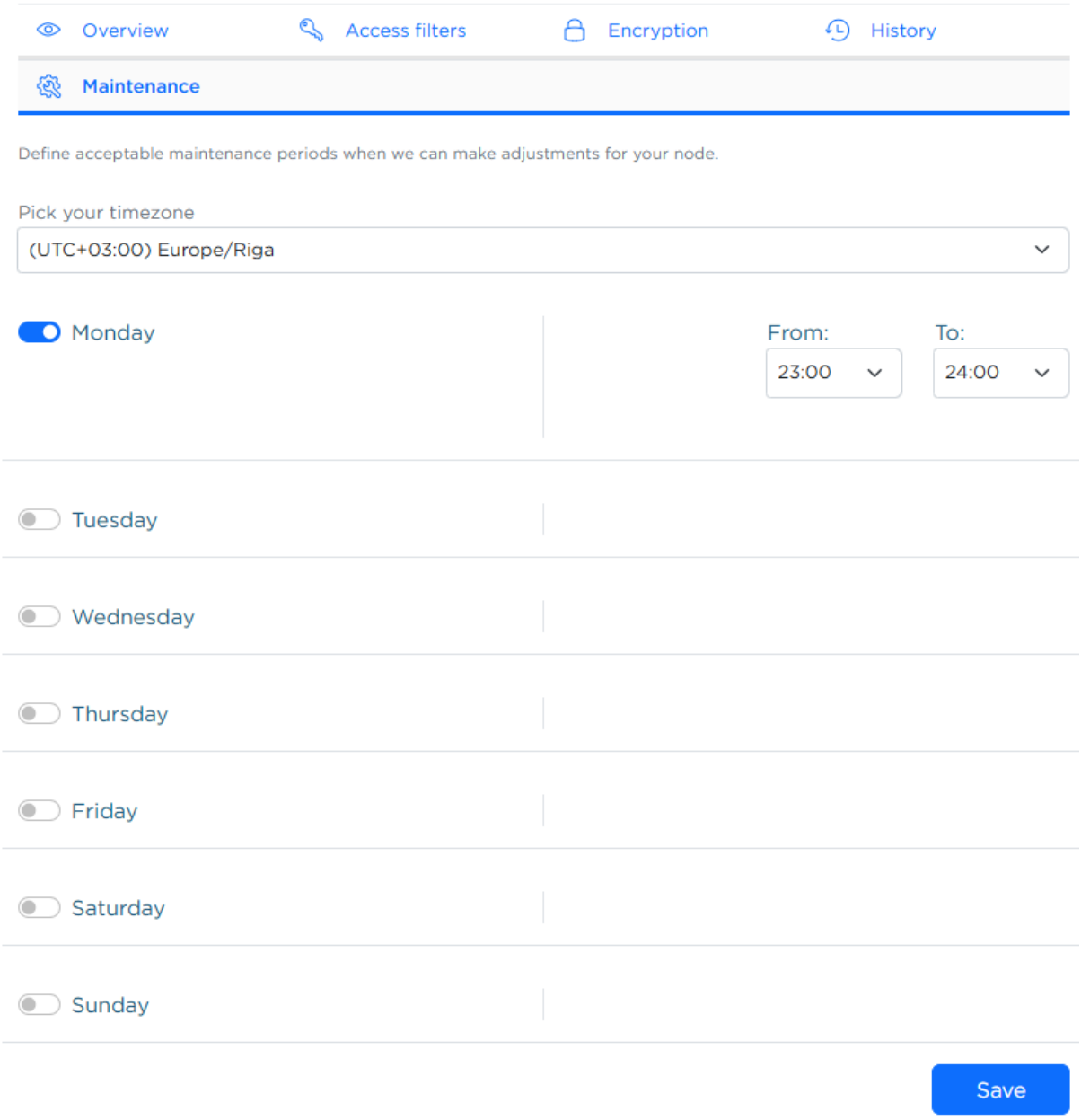
Você pode definir o período de armazenamento para cada tipo de dado (por exemplo, numérico, texto e logs) e escolher a unidade (dias, meses ou anos). As alterações podem ser feitas até três vezes a cada 24 horas.

Depois de atualizar os valores, clique em *Salvar* para aplicar as alterações.

Warning:
Períodos de retenção mais longos podem aumentar o uso do disco e afetar o desempenho. Recomenda-se configurar os períodos de retenção com base nos seus dados reais e nas necessidades de análise.
 O Zabbix Cloud usa 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 gerenciamento de retenção mais previsível.

Aba Maintenance

A aba *Maintenance* permite definir janelas de manutenção aceitáveis nas quais atualizações ou ajustes podem ser realizados no seu node.

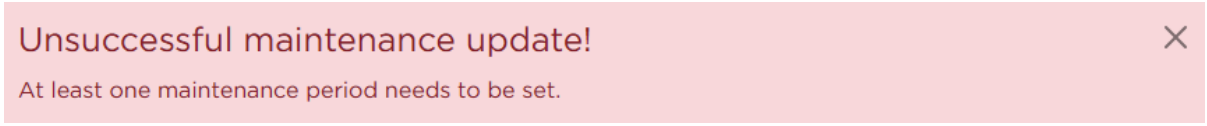


Você deve:

- Selecionar seu fuso horário na lista suspensa.
- Escolher pelo menos um dia e uma janela de horário para a manutenção.

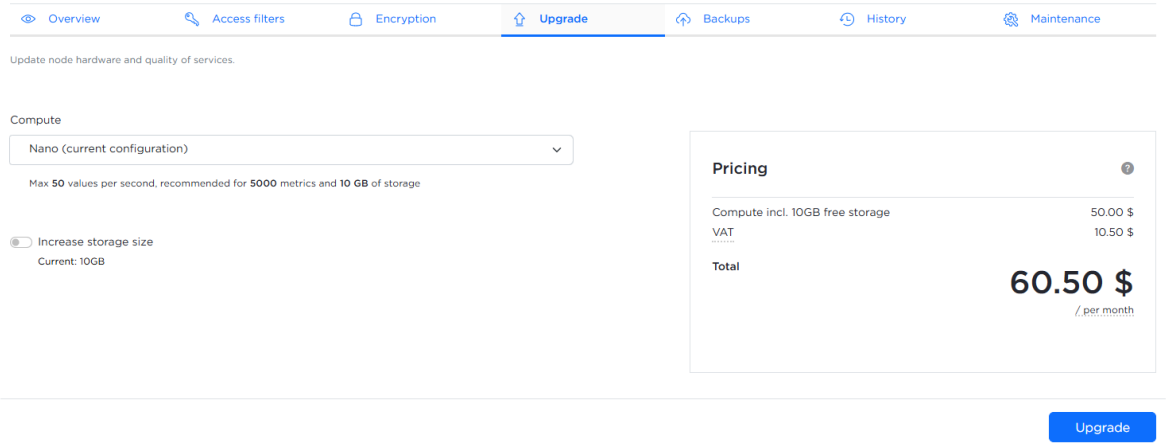
Clique em *Save* para aplicar as alterações.

Se nenhum dia ou janela de horário for selecionado, um erro será exibido.



Aba Upgrade

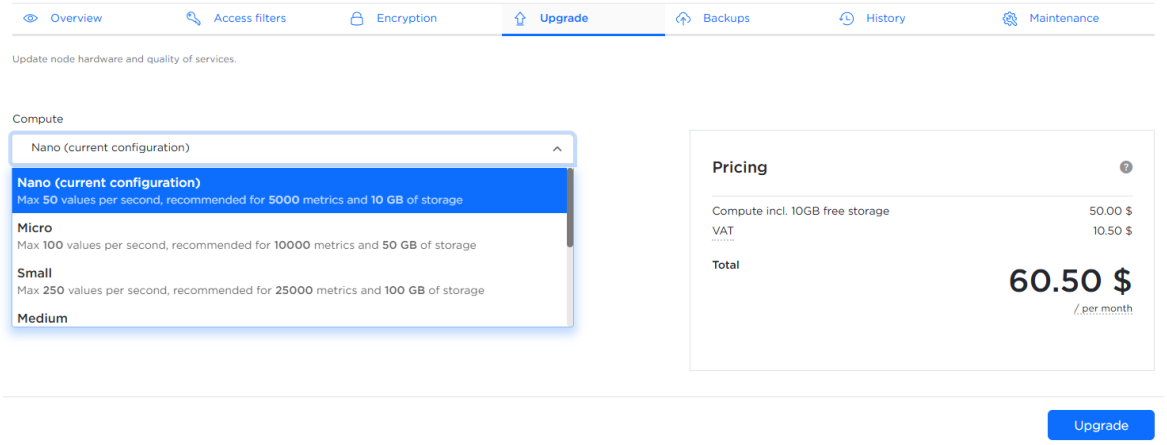
A aba *Upgrade* permite dimensionar os recursos do seu node ou gerenciar sua assinatura.



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 do usuário (entre em contato com o suporte da Zabbix para solicitar qualquer redução). Os nodes do período de avaliação gratuita podem ser convertidos em assinaturas pagas.

Para aumentar o nível de computação:

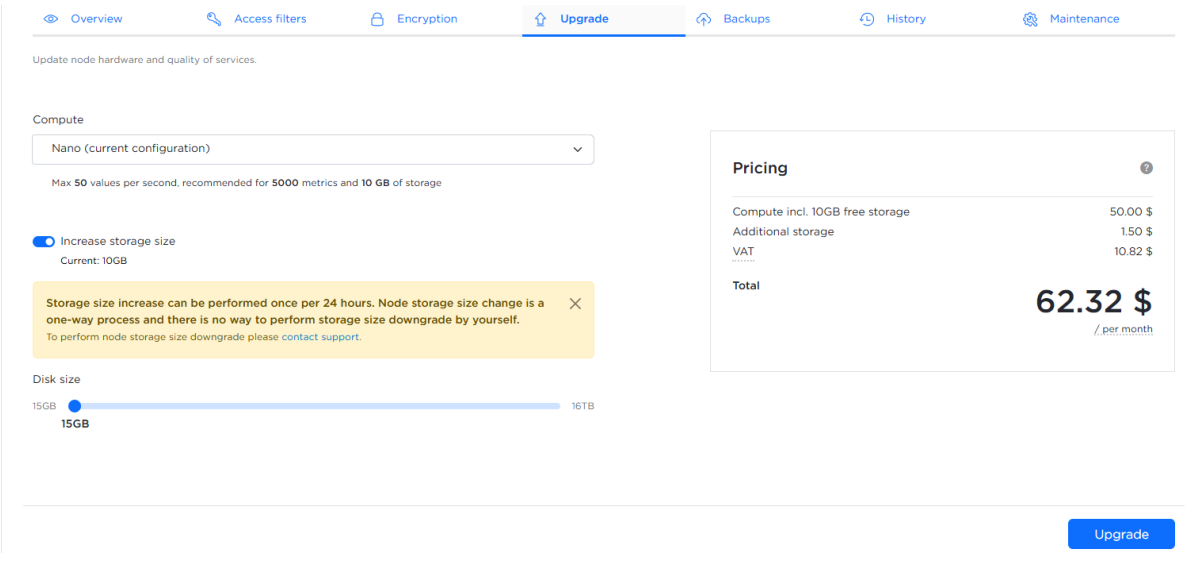
1. Clique no menu suspenso *Compute*.



2. Selecione o nível de configuração desejado.
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 em *Upgrade*.

Note:
O preço mensal total atualizado é exibido na tabela *Pricing* antes da confirmação das alterações.

Adicionando usuários

Introdução

Os exemplos a seguir descrevem como adicionar usuários à sua organização e aos nós.

Note:
Copiar, limpar e redefinir a senha do frontend só pode ser realizado pelo proprietário.

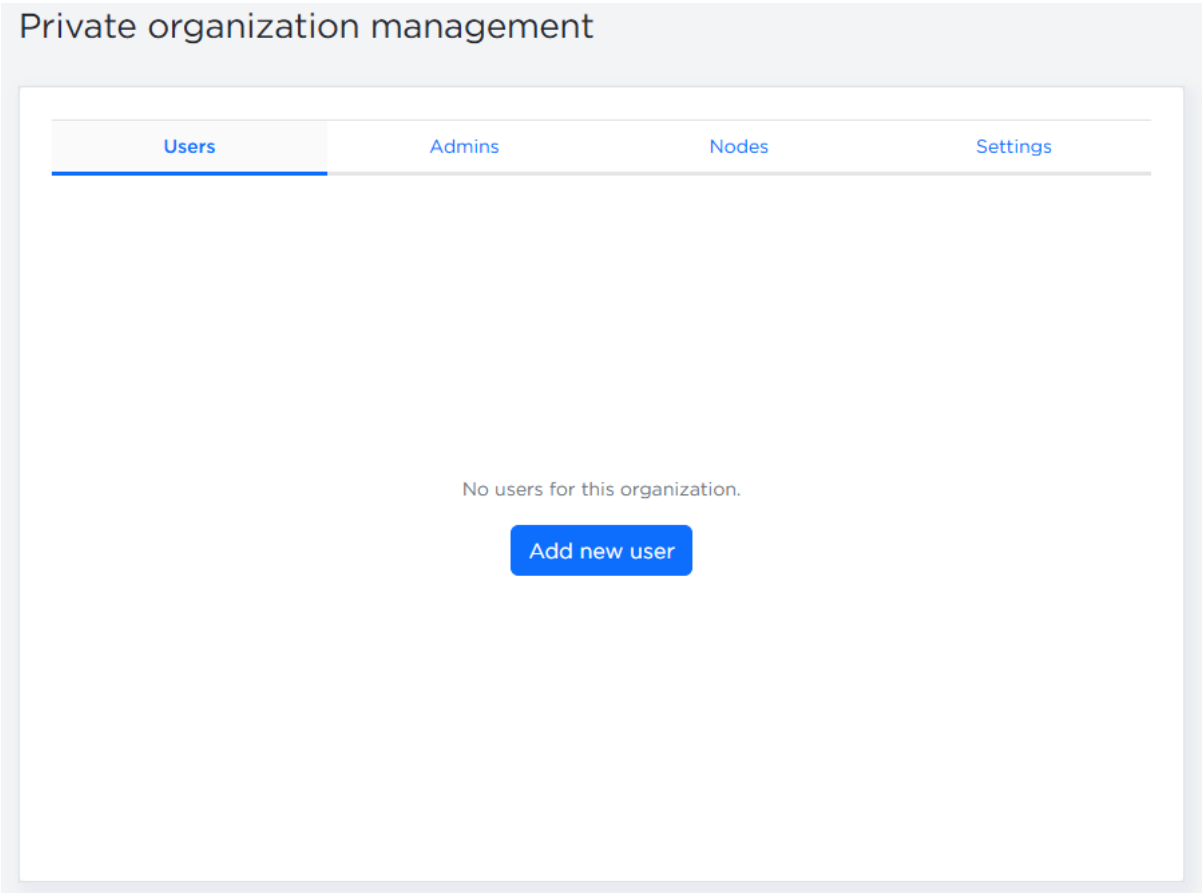
Adicionando usuários às organizações

Semelhante aos usuários no Zabbix, você pode criar usuários no Zabbix Cloud e atribuir 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.

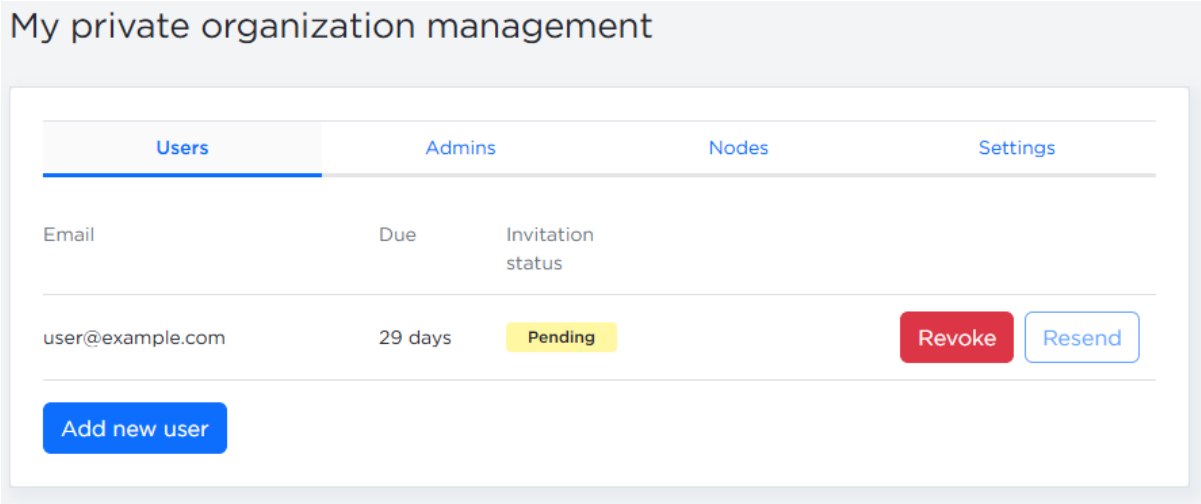
Cada função concede um nível diferente de acesso:

- **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.

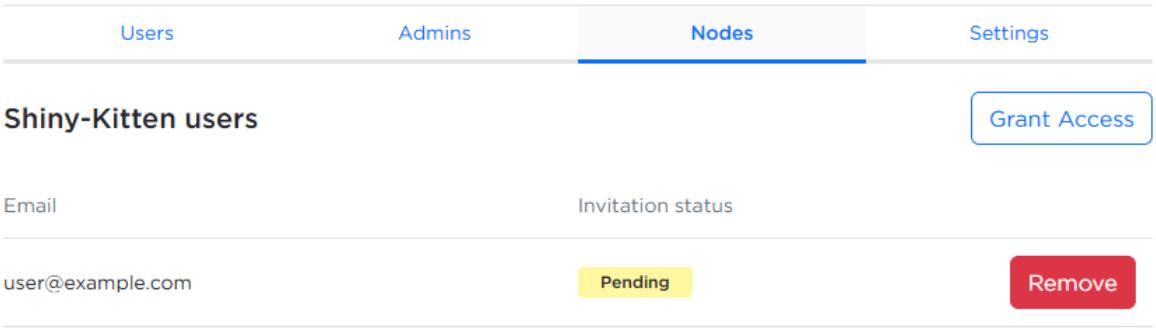
O usuário convidado será adicionado com status *Pendente*. Você pode revogar o convite ou reenviá-lo (o reenvio está disponível uma vez a cada 10 minutos—o botão ficará desabilitado se esse limite for atingido).



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.

Principais diferenças do Zabbix Cloud

Introdução

Zabbix Cloud é uma plataforma baseada em nuvem que permite aos seus usuários executar instâncias autônomas do Zabbix na nuvem (nós em nuvem).

Em comparação com uma implantação regular do Zabbix no local:

- Não é necessária instalação nem manutenção.
- Os nós em nuvem são totalmente gerenciados pela Zabbix.
- Os nós em nuvem podem ser iniciados em poucos minutos.

Este serviço é oferecido no modelo de pagamento conforme o uso (um **teste gratuito** está disponível).

Tabela de comparação Embora a maior parte das funcionalidades se sobreponha entre as instalações do Zabbix on-premises e os nós na nuvem, também existem diferenças. Uma tabela de comparação resumida é apresentada abaixo. As seções seguintes descrevem alguns itens selecionados em detalhes.

Recurso / Aspecto	Zabbix Cloud	Zabbix on-premises
Ciclo de lançamento	Alinhado às releases LTS	Você escolhe a versão do Zabbix (estável, beta, builds personalizados)
Facilidade de uso	Plataforma pronta para uso, totalmente gerenciada pelo Zabbix	Requer instalação, configuração e manutenção
Aplicação de patches de vulnerabilidades	Aplicação automática de patches do SO/Zabbix	Aplicação manual de patches pelo usuário
Certificado HTTPS	Certificado válido pronto para uso (sem Let's Encrypt)	Autoassinado por padrão (Let's Encrypt ou outra CA pode ser configurada)
Desempenho do DB (INSERT/UPDATE/SELECT)	Particionamento automático para máxima velocidade; sem DELETEs de longa duração	Particionamento e limpeza manuais (transações DELETE longas podem ocorrer)
Ajuste de desempenho	Pré-ajustado pela equipe do Zabbix Cloud (sem changelogs)	Visibilidade e controle completos sobre os parâmetros de ajuste
Controle do firewall	Gerenciado pela equipe do Zabbix Cloud; as portas da GUI/trapper não são expostas publicamente	Controle completo das regras do firewall e da exposição de portas
Tráfego de saída na porta 25	Bloqueado por padrão para evitar spam e abuso; use a porta SMTP 587 ou 465	Não é restringido pelo Zabbix; depende da sua rede e do provedor de hospedagem
Isolamento e proteção do DB	Altamente isolado, sem acesso direto ao DB/SSH	Depende da sua rede e da segurança do host
Alta disponibilidade (HA)	Executado em uma plataforma de nuvem altamente disponível, proxies do Zabbix podem ser usados para maior resiliência da coleta de dados	HA gerenciada pelo usuário
SNMP traps	Somente por meio de um proxy dedicado (sem receptor de traps com HA/balanceamento de carga)	Diretamente ou por meio de um proxy; HA/balanceamento de carga é possível
Polling SNMP	Difícil sem proxies (cada dispositivo precisa de NAT/porta personalizada)	Polling SNMP nativo; proxy opcional
Configurações de retenção	Histórico/Tendências/Auditoria somente pela UI (sem API ou arquivo de configuração)	Configurável no arquivo de configuração do server do Zabbix ou por meio da API
Scripts personalizados	`AlertScriptsPath`, `ExternalScripts`, módulos do frontend e da comunidade não são compatíveis	Totalmente compatível (caminho de scripts, módulos, integrações)

Recurso / Aspecto	Zabbix Cloud	Zabbix on-premises
Monitoramento ODBC	Somente PostgreSQL (driver `{postgresql}`); o plugin do MariaDB está presente como placeholder	ODBC para PostgreSQL, MySQL, Oracle etc. - totalmente configurável
Limitação de chamadas ODBC	Não é possível (somente `StartODBCPollers=1`); consultas de sincronização pesadas bloqueiam outras tarefas	Quantidade e agendamento dos pollers totalmente ajustáveis
Upload de certificado SAML	Não compatível	Compatível pela UI ou API
Relatórios agendados	É necessário criar sua própria mídia Email (mídia de script não compatível)	Compatível com mídias de script e Email prontas para uso
Interface do host para verificações ativas	Endereço IP fornecido pela plataforma (criado automaticamente)	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 requer que cada dispositivo seja exposto via NAT e portas personalizadas.

Configurações de retenção

Os períodos de retenção de histórico, tendências e log de auditoria só podem ser configurados pela interface web na nuvem. Essas configurações não podem ser feitas através do `zabbix_server.conf` ou da API.

Substituições manuais de histórico por item não são suportadas (o particionamento é controlado globalmente). As URLs dos nós em nuvem não aceitam os mesmos parâmetros de consulta que as instalações locais.

Scripts personalizados

Os seguintes tipos de scripts personalizados não são suportados pelo Zabbix Cloud:

- Scripts de alerta (`AlertScriptsPath`)
- Scripts externos (`ExternalScripts`)
- Scripts de frontend no Zabbix Server

Módulos de frontend desenvolvidos pela comunidade não podem ser instalados.

Monitoramento ODBC

O Zabbix Cloud suporta monitoramento ODBC apenas para PostgreSQL. Use o template ODBC oficial do Zabbix e defina a string de conexão no template usando:

```
Driver={postgresql}
```

Um plugin MariaDB também está instalado, mas atualmente não está funcional. Se 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 ter sucesso, 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 da nuvem ou da API, garantindo que as camadas do sistema operacional e do banco de dados permaneçam isoladas e seguras.

Tráfego de saída na porta 25

Para evitar spam e abuso, as conexões de saída dos nós do Zabbix Cloud para a porta TCP 25 são bloqueadas por padrão. Isso afeta:

- Tipos de mídia Email que enviam mensagens por meio de um servidor SMTP na porta 25. Como 25 é a *porta do servidor SMTP* padrão para um novo tipo de mídia Email, altere-a para 587 (com *Segurança da conexão* definida como STARTTLS) ou 465 (SSL/TLS), dependendo do que o seu servidor SMTP oferece. O provedor de e-mail *Office365 relay* não pode ser usado, pois ele se conecta ao endpoint de relay da Microsoft na porta 25.
- Relatórios agendados, que exigem um tipo de mídia Email.
- Verificações que o próprio nó de nuvem realiza na porta 25, como `net.tcp.service[smtp]` verificações simples ou verificações SMTP em regras de descoberta de rede. Essas verificações informam que o serviço SMTP está indisponível mesmo quando ele está em execução; em vez disso, realize essas verificações por meio de um proxy ou agent do Zabbix na sua própria rede.

O tipo de mídia *Cloud Email* padrão não é afetado.

Interface de active check

Active checks na nuvem criam automaticamente uma interface de host com um IP não relacionado à sua rede local. Por padrão, esse IP pode ser externo. Para manter a consistência, você pode excluir ou ajustar manualmente a interface de host após a criação.

Upload de certificado SAML

A autenticação SAML não é suportada porque poucos provedores aceitam mensagens SAML não criptografadas ou não assinadas.

Relatórios agendados

O tipo de mídia *Cloud Email* padrão usa um transporte de script e não pode ser usado para relatórios agendados. Para enviar relatórios por e-mail:

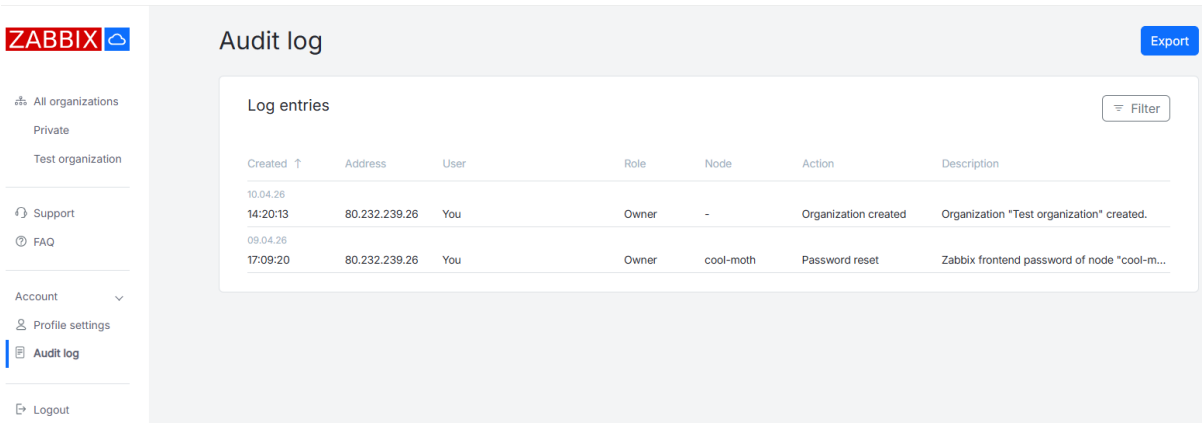
1. Crie um novo tipo de mídia Email.
2. Atribua a mídia de e-mail ao seu usuário em *Configurações do usuário* > *Mídia*.
3. Selecione esse tipo de mídia ao configurar os relatórios agendados.

Attention:
Não use a porta 25 como *porta do server SMTP* para esse tipo de mídia; consulte Tráfego de saída na porta 25.

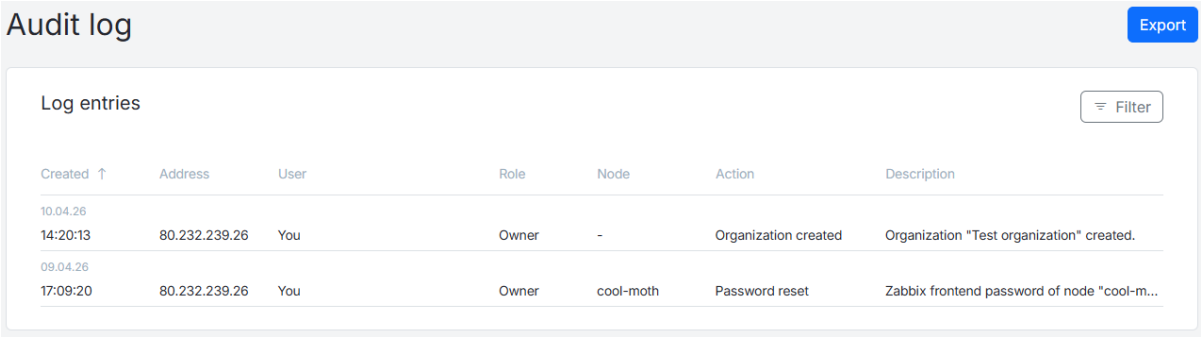
Log de auditoria

Visão geral No Zabbix Cloud, o log de auditoria registra a atividade de usuários e do sistema.

Após fazer login em uma conta do Zabbix Cloud, vá para *Account* > *Audit log* para visualizar os registros do log de auditoria. As entradas do log de auditoria disponíveis para um usuário dependem da função do usuário na organização.



Visualizando registros do log de auditoria Os registros do log de auditoria são exibidos em uma tabela.



A tabela contém as seguintes informações:

Column	Description
Created	Data e hora em que a entrada do log foi criada. O horário exibido corresponde ao fuso horário do usuário selecionado no menu suspenso *User timezone* em *Account > Profile settings*.
User	Nome do usuário da organização que executou a ação.
Role	A função do usuário que executou a ação.
Node	Nome do nó relacionado à ação.
Action	Nome da ação executada.
Description	Descrição da ação executada.

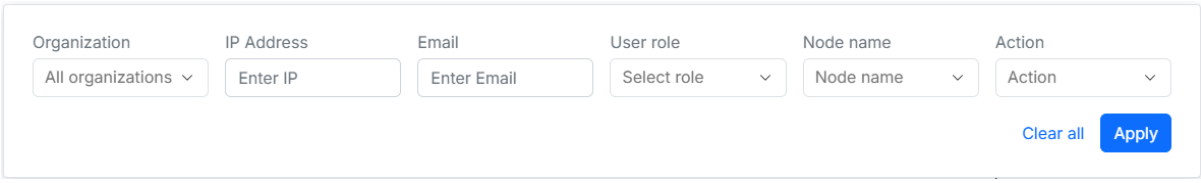
Os nomes das colunas podem ser clicados para classificar os registros em ordem crescente ou decrescente.

Os registros do log de auditoria filtrados podem ser exportados no formato `.csv` usando o botão *Export*.

Usando filtros

Use o botão *Filter* para abrir o filtro.

Para exibir os resultados filtrados, clique em *Apply* após selecionar um ou mais parâmetros de filtragem. Para limpar o filtro, clique em *Clear all*.



É possível filtrar registros por um ou mais parâmetros. Os seguintes parâmetros de filtragem estão disponíveis:

Parâmetro	Descrição
Organization	Selecione todas as organizações ou uma específica.
IP Address	Especifique um endereço IP.
Email	Especifique um endereço de e-mail associado ao usuário.
User role	Selecione todas as funções ou uma específica: Owner, Admin ou User.
Node name	Selecione todos os nodes ou um específico.
Action	Selecione um tipo de ação específico.

O log de auditoria registra as seguintes ações:

- *Node criado*
- *Node excluído*
- *Node atualizado*
- *Versão do node atualizada*
- *Senha redefinida*
- *Manutenção atualizada*
- *Configuração de Histórico atualizada**
- *Data de expiração do node estendida*
- *Configuração do proxy atualizada*
- *Backup criado*
- *Backup excluído*

- *Backup restaurado*
- *Backup atualizado*
- *Filtro de acesso adicionado*
- *Filtro de acesso atualizado*
- *Filtro de acesso excluído*
- *Criptografia habilitada*
- *Criptografia desabilitada*
- *Assinatura cancelada*
- *Assinatura renovada*
- *Falha no pagamento da assinatura*
- *O pagamento requer confirmação*
- *Organização criada*
- *Organização excluída*
- *Admin convidado*
- *Usuário convidado*
- *Admin ingressou*
- *Usuário ingressou*
- *Admin removido da organização*
- *Usuário removido da organização*
- *Node adicionado à organização*
- *Node removido da organização*
- *Acesso ao node concedido*
- *Acesso ao node revogado*
- *Configurações do usuário atualizadas*
- *Pagamento criado*
- *Pagamento concluído*
- *Falha no pagamento*
- *Cobrança de VAT aplicada*
- *Falha na validação de VAT*
- *Informações de cobrança atualizadas*
- *Método de pagamento atualizado*
- *Método de pagamento removido*
- *Node pré-pago criado*
- *Node pré-pago atualizado*
- *Incidente iniciado*
- *Incidente encerrado*

Configurar um proxy do Zabbix Cloud

Visão geral Esta página descreve como configurar um proxy do Zabbix implantado no Zabbix Cloud para coletar dados de monitoramento de hosts em um ambiente remoto.

O procedimento destina-se principalmente a um server e proxy do Zabbix criados no Zabbix Cloud.

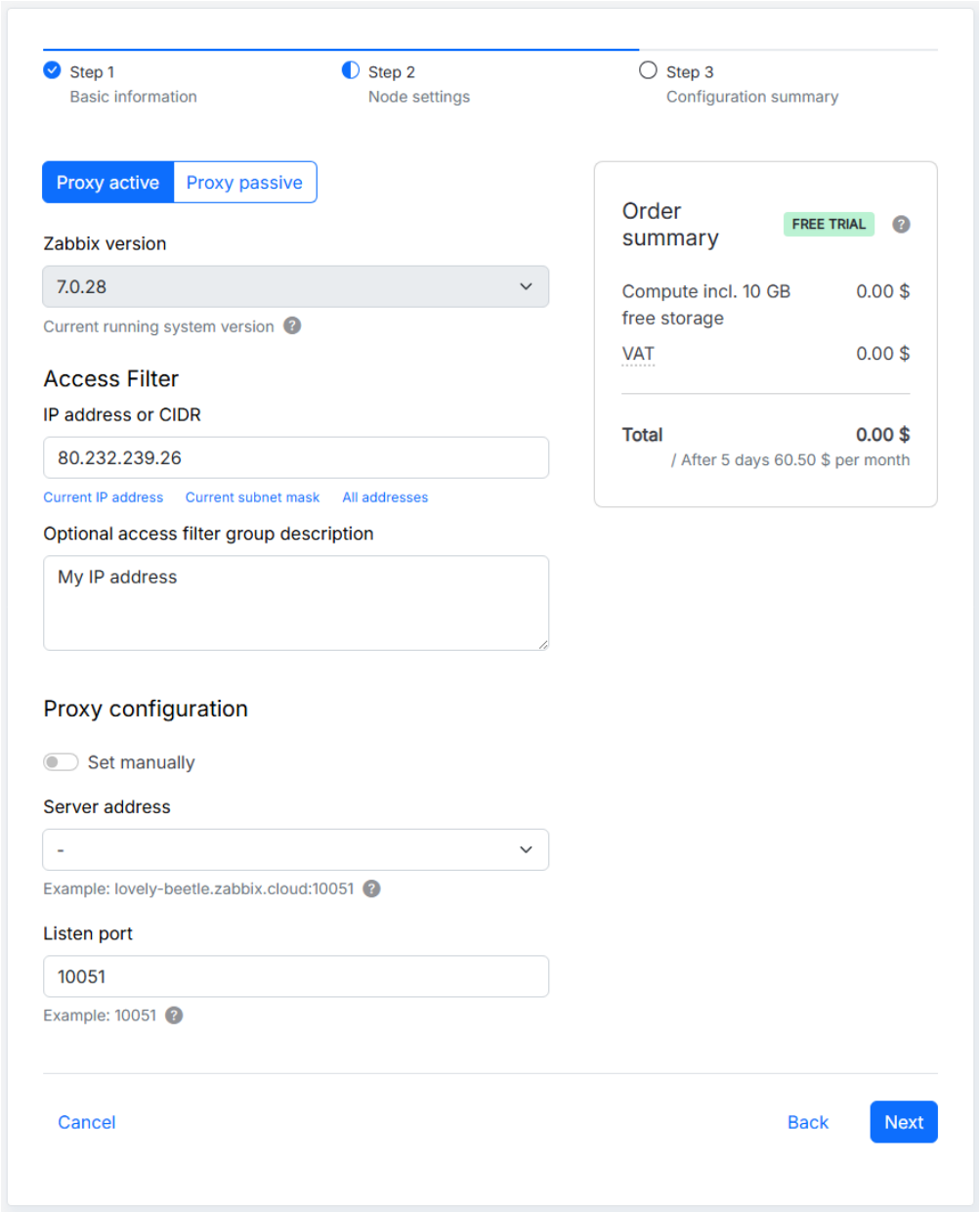
O procedimento também descreve a configuração necessária quando o server do Zabbix não está hospedado no Zabbix Cloud.

Pré-requisitos Um Zabbix server está instalado, configurado e em execução, ou um nó do Zabbix server está em execução no Zabbix Cloud.

Criar um node proxy Crie um node proxy do Zabbix Cloud.

Durante a criação do node:

1. Selecione *Proxy*.
2. Selecione *Proxy active* ou *Proxy passive*, dependendo de como o proxy deve se comunicar com o server do Zabbix.
3. Configure os filtros de acesso por endereço IP ou CIDR do node (isso pode ser feito após a criação do node).
4. Especifique o server do Zabbix ao qual o proxy se conectará.



5. Conclua o processo de criação do node.

Para obter mais informações sobre como criar um node proxy, consulte Implantar o Zabbix na nuvem.

Configurar filtros de acesso O Zabbix Cloud usa filtros de acesso para controlar quais endereços IP ou sub-redes podem se conectar a um node na nuvem.

Durante a inicialização do node, um filtro de acesso é criado automaticamente para o endereço IP usado para acessar o node. Isso não inclui necessariamente os endereços dos hosts monitorados ou de outros componentes do Zabbix. Adicione manualmente os endereços necessários.

Consulte Configuração do node — Filtros de acesso para obter mais informações.

Nó proxy

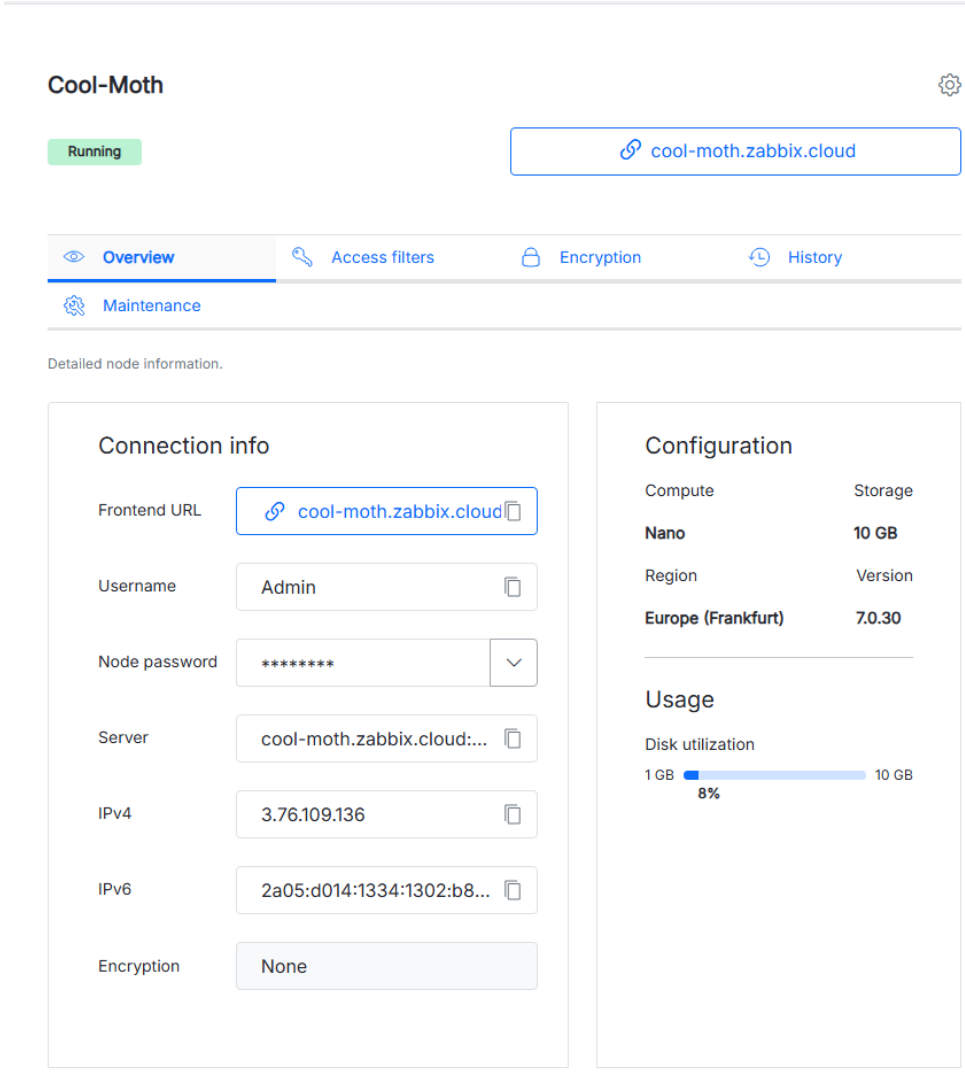
Adicione os endereços IP ou as sub-redes a partir dos quais são esperadas conexões com o proxy.

Se os hosts monitorados usarem verificações ativas do agent, os agents iniciarão conexões com o proxy; portanto, o endereço IP ou a sub-rede dos hosts monitorados deverá ser permitido pelos filtros de acesso do nó proxy.

Se *Proxy Passive* tiver sido definido como o modo do proxy, o server Zabbix se conectará ao proxy; portanto, o endereço IP ou a sub-rede do server Zabbix deverá ser permitido pelos filtros de acesso do nó proxy.

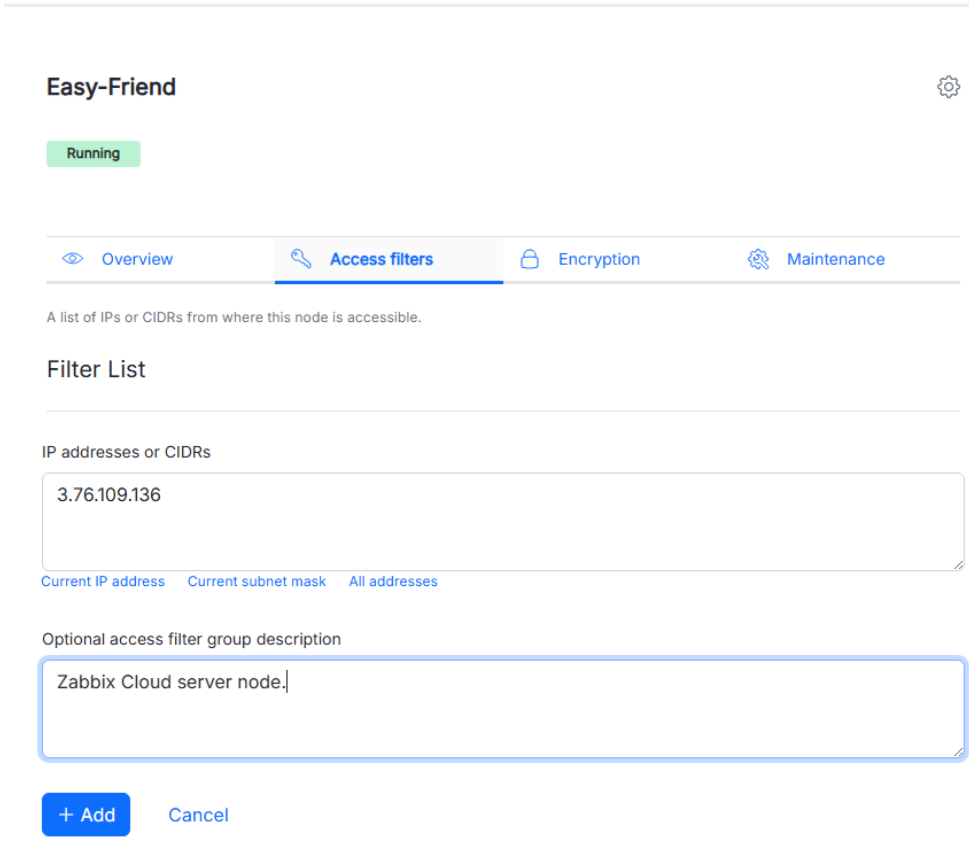
Se *Proxy passive* tiver sido selecionado e um nó server for usado:

1. Acesse a aba *Overview* do seu nó server e copie o endereço IPv4 ou IPv6 do nó.

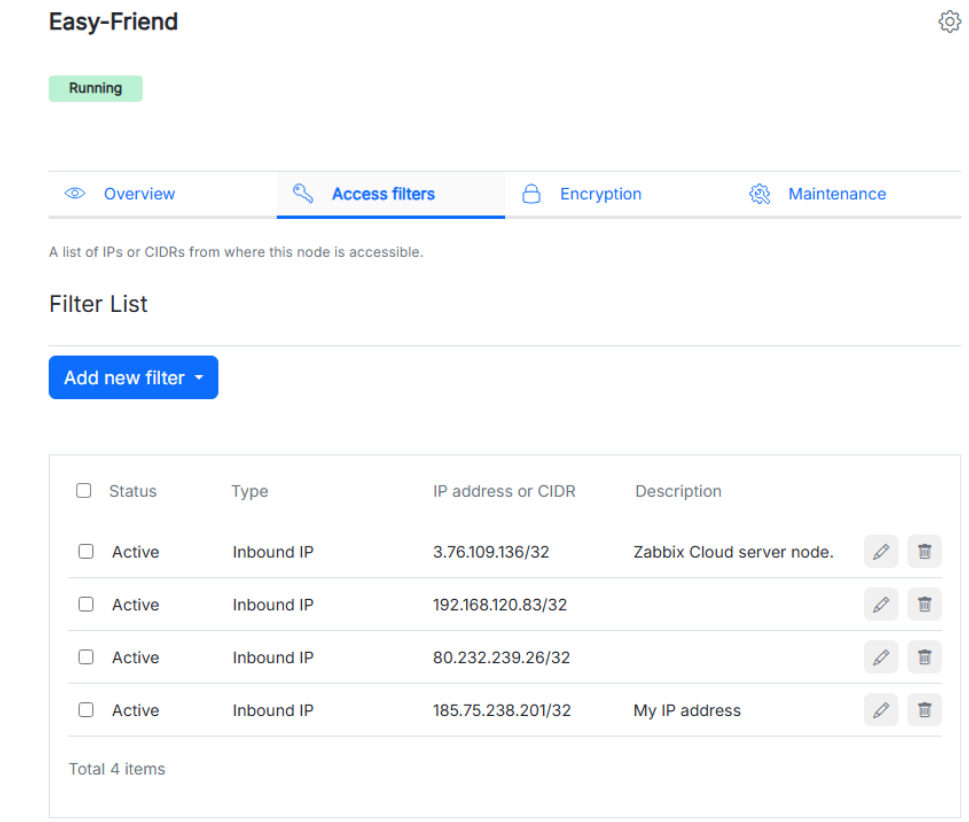


2. Acesse a aba *Access filters* do seu nó proxy e clique em *Add new filter* > *Add*.

3. Cole o endereço copiado no campo *IP addresses or CIDRs* e adicione uma descrição ao campo *Optional access filter group description*.



4. Clique em *Add*.



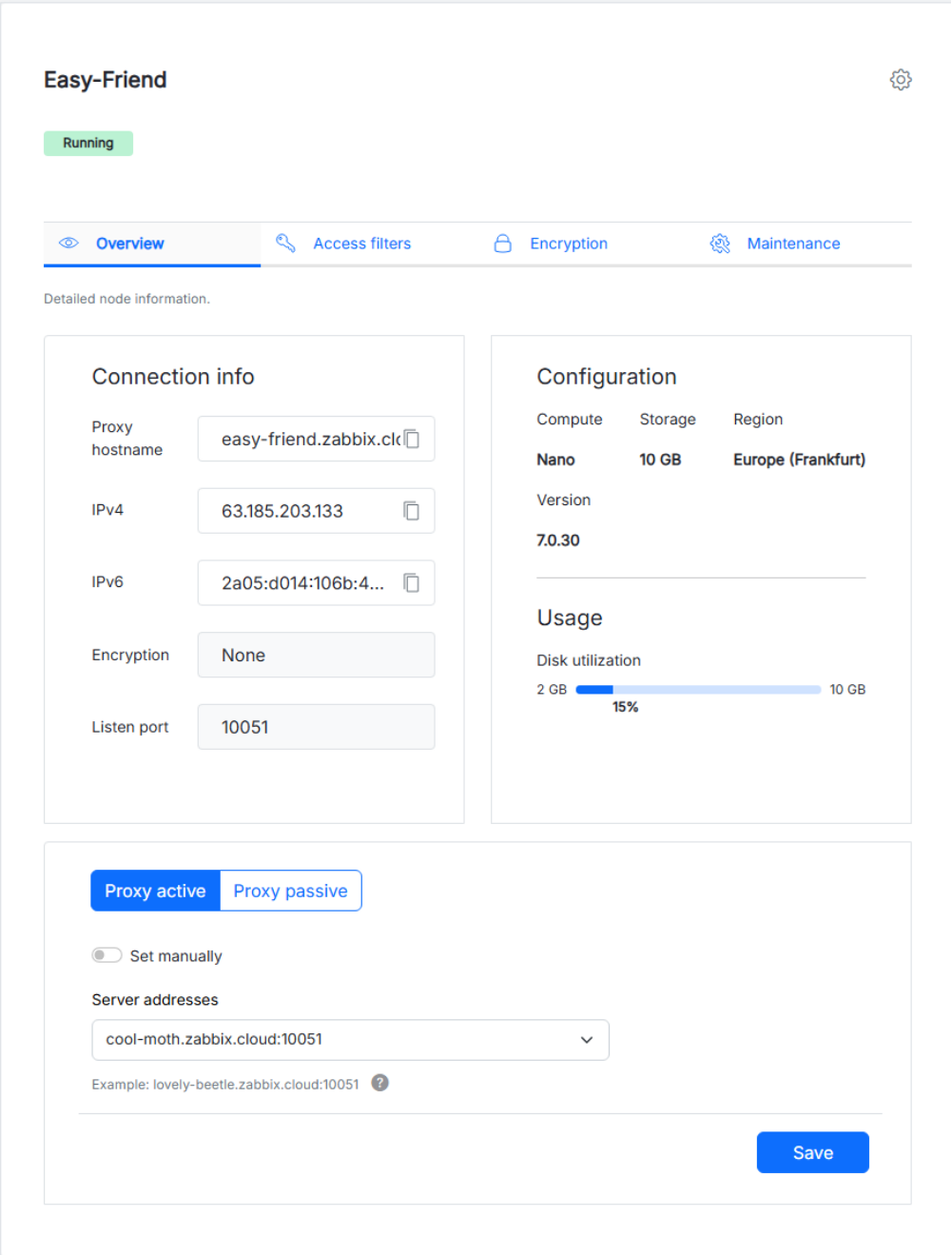
Repita as etapas 2 a 4 para o endereço IP de um agent Zabbix que use verificações ativas.

Nó do server

Se o Zabbix server estiver hospedado no Zabbix Cloud, configure também os filtros de acesso no nó do server.

Para um proxy no modo ativo, adicione o endereço IP ou o CIDR do nó do proxy aos filtros de acesso do nó do server, pois o proxy inicia a conexão com o server.

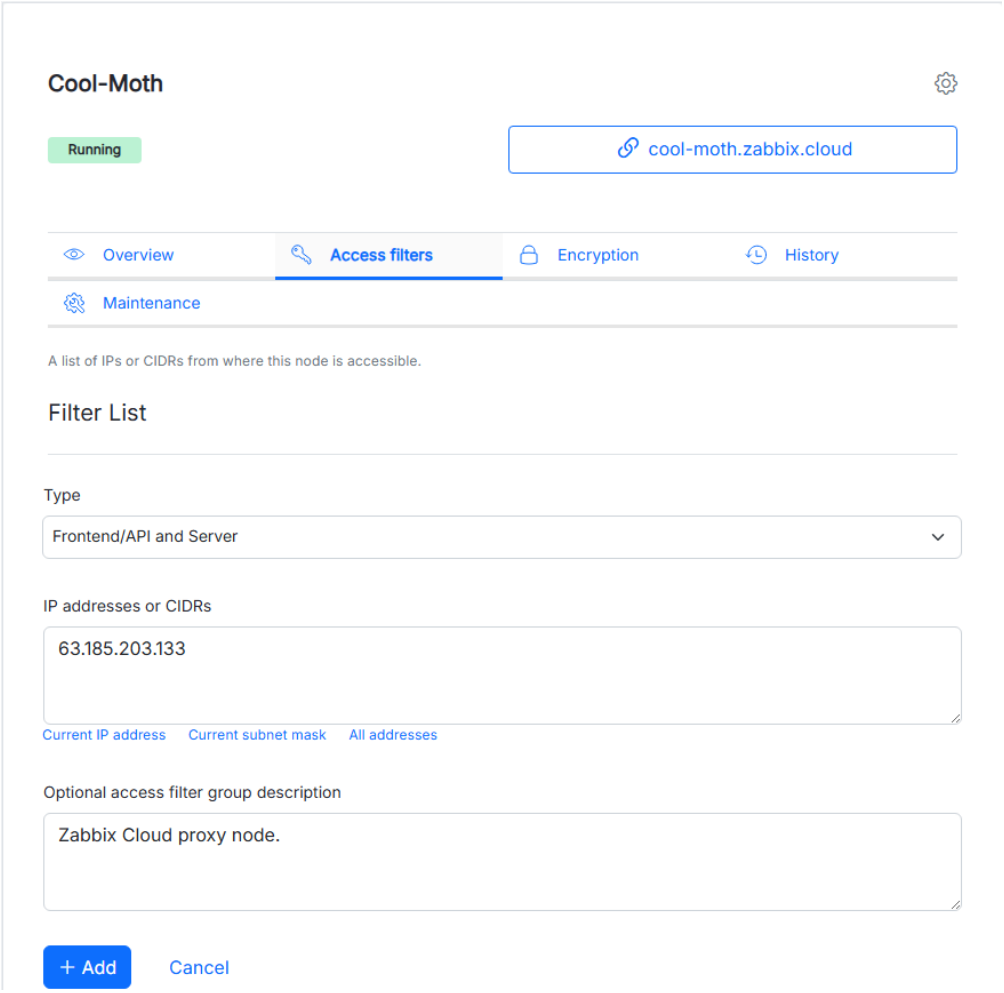
1. Acesse a aba *Overview* do nó do proxy e copie o endereço IPv4 ou IPv6 do nó.



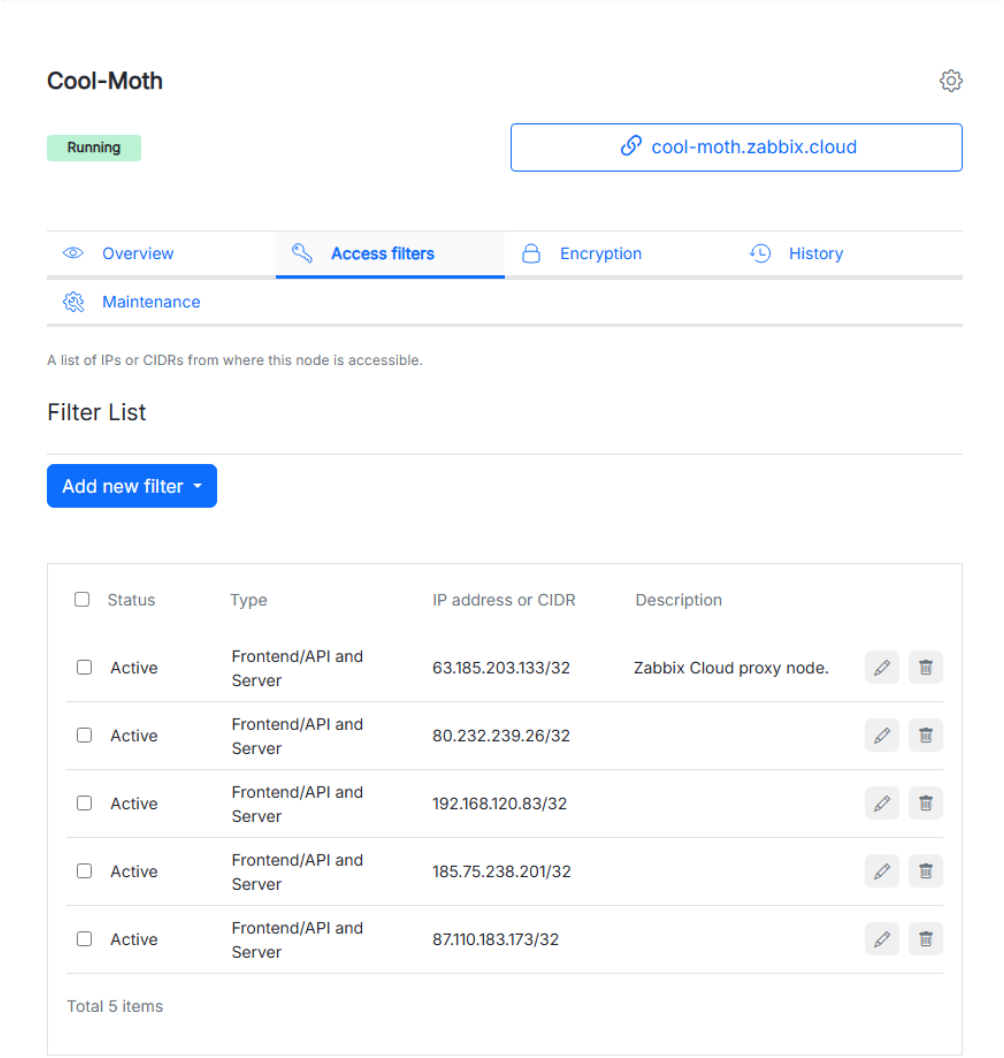
2. Acesse a aba *Access filters* do nó do server e clique em *Add new filter* > *Add*.

3. Cole o endereço copiado no campo *IP addresses or CIDRs* e adicione uma descrição ao campo *Optional access filter group description*.

4. Selecione o tipo de filtro apropriado.



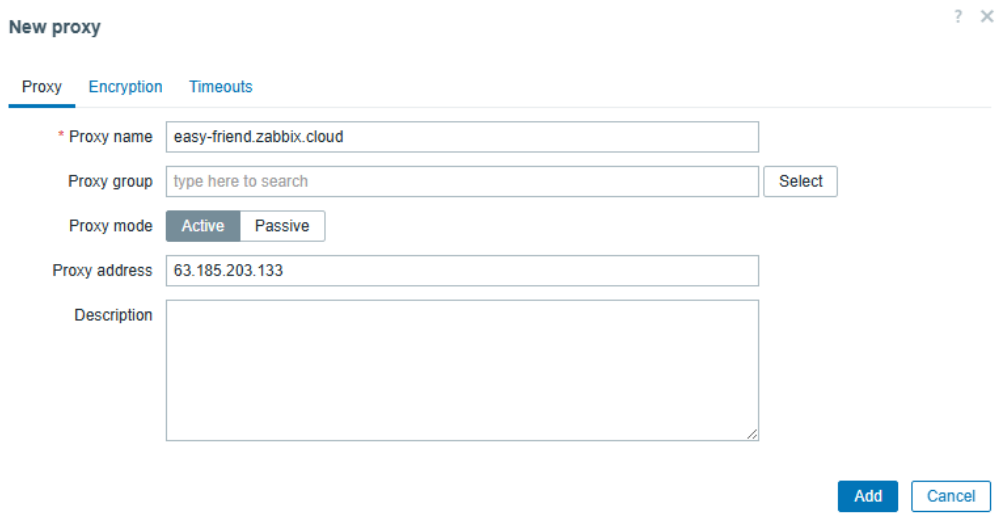
5. Clique em *Add*.



Configurar o proxy O proxy deve ser configurado no frontend do Zabbix para se comunicar com o server do Zabbix.

Abra *Administration* > *Proxies* no frontend do Zabbix, clique em *Create proxy* e configure o proxy conforme necessário:

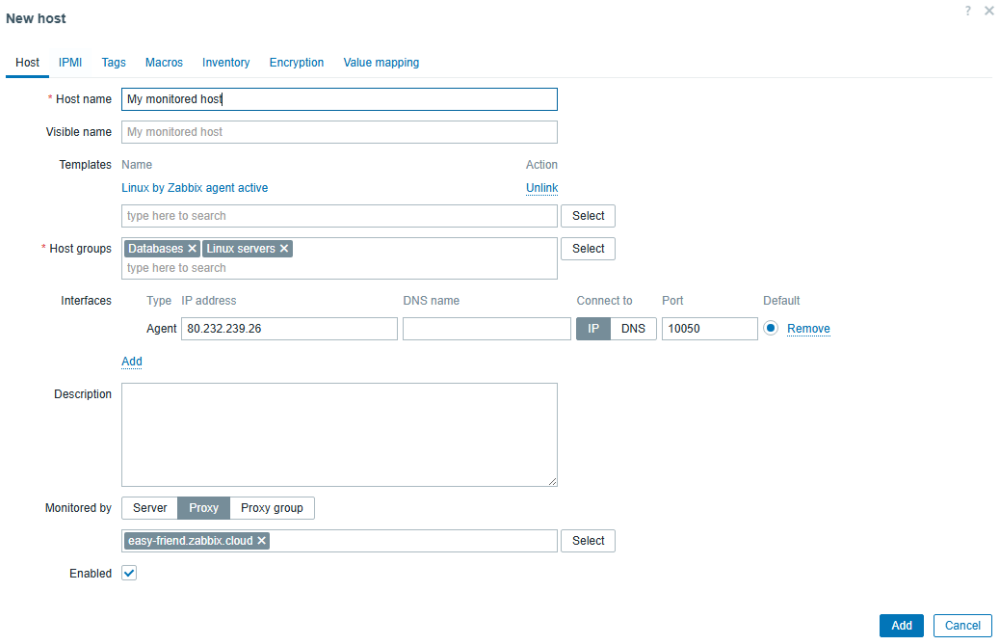
1. Adicione o nome do proxy ao campo *Proxy name*. Ele deve corresponder ao nome do host do proxy configurado para o proxy. Digite-o ou copie-o da aba *Overview* do seu proxy.
2. Selecione o modo do proxy correspondente ao selecionado durante a criação do proxy.
3. Para um proxy ativo, insira o endereço IP ou o nome DNS do proxy.
4. Clique em *Add*.



Para obter mais informações, consulte Monitoramento distribuído usando proxies.

Configurar hosts monitorados Adicione os hosts que devem ser monitorados por meio do proxy.

1. Acesse *Coleta de dados* > *Hosts*.
2. Crie um host ou abra um host existente.
3. Em *Monitorado por*, selecione o *Proxy* e digite o nome do proxy para pesquisá-lo e selecioná-lo ou pressione o botão *Selecionar* para abrir uma lista de proxies disponíveis.
4. Configure a interface do host e os demais parâmetros.
5. Clique em *Adicionar* ou *Atualizar*.



Um host configurado para ser monitorado por um proxy envia ou recebe dados de monitoramento por meio desse proxy, dependendo do tipo de item e do modo do agent.

Configurar o Zabbix agent Quando um host monitorado usa o Zabbix agent ou o Zabbix agent 2, configure o agent para se comunicar com o componente Zabbix apropriado.

A configuração depende de o agent usar verificações *passivas* ou *ativas*.

Verificações passivas do agent

Nas verificações passivas, o proxy do Zabbix inicia conexões com o agent e solicita dados de monitoramento.

Abra o arquivo de configuração do agent do Zabbix e atualize o parâmetro `Server` para permitir conexões do endereço IP do proxy do Zabbix Cloud.

Por exemplo:

```
Server=<proxy_IP>
```

Certifique-se de que o host monitorado permita conexões TCP de entrada do proxy na porta do agent, que é 10050 por padrão.

O proxy deve conseguir acessar o host monitorado pela rede.

Verificações ativas do agent

Com as verificações ativas, o agent inicia conexões com o proxy do Zabbix e envia os dados de monitoramento coletados.

Abra o arquivo de configuração do agent do Zabbix e atualize o parâmetro `ServerActive` com o endereço do proxy do Zabbix Cloud.

Por exemplo:

`ServerActive=<proxy_IP>:<proxy_listen_port>`

Substitua `<proxy_IP>` pelo endereço IP do proxy do Zabbix Cloud e `<proxy_listen_port>` pela porta de escuta configurada quando o nó proxy foi criado. A porta de escuta padrão é 10051.

Se um proxy for especificado em `ServerActive`, o agent obterá sua configuração de verificações ativas do proxy, em vez de diretamente do server do Zabbix.

Certifique-se de que o host monitorado possa estabelecer uma conexão TCP de saída com o proxy na porta de escuta configurada.

Para obter mais informações sobre verificações passivas e ativas do agent, consulte Verificações passivas e ativas do agent.

Zabbix server externo Um proxy do Zabbix Cloud também pode se comunicar com um Zabbix server implantado fora do Zabbix Cloud.

Nesse caso, o Zabbix server externo deve ser capaz de se comunicar com o proxy de nuvem de acordo com o modo de proxy selecionado.

Para um proxy no modo ativo, o proxy inicia a conexão com o Zabbix server externo. Configure o Zabbix server externo para aceitar conexões do proxy e certifique-se de que o proxy possa alcançar o server pela rede.

Para um proxy passivo, o Zabbix server inicia a conexão com o proxy. Adicione o endereço IP ou a sub-rede do Zabbix server externo aos filtros de acesso do nó do proxy e certifique-se de que o server possa alcançar o proxy pela rede.

O Zabbix server externo não usa os filtros de acesso do Zabbix Cloud. Em vez disso, configure as regras de firewall e de rede necessárias no ambiente externo.

Os hosts monitorados podem estar localizados no mesmo ambiente que o Zabbix server externo, em outra rede ou em uma rede acessível pelo proxy.

Quando os agents usam verificações ativas, configure o parâmetro `ServerActive` para apontar para o proxy do Zabbix Cloud. Quando os agents usam verificações passivas, configure o parâmetro `Server` para permitir conexões do proxy.

Verificar a configuração Após configurar o proxy, os agents, os filtros de acesso e a conectividade de rede, verifique se o proxy está se comunicando com o Zabbix server.

1. Acesse *Administration* > *Proxies*.
2. Verifique o *State* do proxy.
3. Acesse *Monitoring* > *Latest data* e verifique se há valores recentes de item disponíveis.

Se o proxy não estiver se comunicando com o server, verifique:

- o modo do proxy;
- o endereço e a porta do server;
- os filtros de acesso do proxy;
- os filtros de acesso do server, quando o server estiver hospedado no Zabbix Cloud;
- a configuração do agent;
- as regras de firewall e roteamento;
- o endereço configurado em `Server` ou `ServerActive`.

Aplicativo móvel

Aplicativo móvel

O aplicativo móvel do Zabbix permite monitorar seu ambiente Zabbix pelo dispositivo móvel. Visualize problemas, confira seus detalhes, atualize-os e receba notificações push quando algo precisar de atenção.

Obter o aplicativo (em breve)

Configure o Zabbix para o aplicativo

Configurar o adaptador de ponte do Zabbix Instale o adaptador de ponte do Zabbix para permitir que dispositivos móveis se conectem ao Zabbix.

Adicionar dispositivos Adicione seu dispositivo móvel na interface web do Zabbix para conectá-lo ao seu Zabbix server.

Configurar notificações push Receba alertas no seu dispositivo móvel quando ocorrer um problema.

Saiba mais sobre o aplicativo

Visão geral do aplicativo Aprenda os conceitos básicos do uso do aplicativo móvel do Zabbix.

Acesso móvel seguro Proteja o aplicativo, seus dispositivos móveis e o adaptador de ponte do Zabbix, ou revogue o acesso de dispositivos perdidos ou comprometidos.

Configure o Zabbix para o aplicativo

Para usar o aplicativo móvel do Zabbix (em breve), primeiro você deve:

1. Configurar o adaptador de ponte do Zabbix.
2. Configurar as notificações push.
3. Adicionar seu dispositivo móvel na interface web do Zabbix.

As seções abaixo abordam cada etapa.

Após a configuração, consulte Acesso móvel seguro para manter o aplicativo e o adaptador de ponte do Zabbix seguros.

Configurar o adaptador de bridge do Zabbix Para permitir que seu Zabbix server se comunique com o aplicativo móvel, você deve configurar o adaptador de bridge do Zabbix.

O adaptador de bridge pode ser instalado no mesmo sistema que o seu Zabbix server ou em um sistema diferente, usando pacotes:

> **Attention:**
> As etapas abaixo são para o Ubuntu 24.04. Para outros sistemas operacionais, obtenha os comandos de instalação do repositório e dos pacotes (por exemplo, `wget`, `dpkg`, `apt`) na página Obter o Zabbix para o seu sistema operacional e use-os para instalar o pacote `zabbix-bridge-adapter`.

1. Inicie uma nova sessão do shell com privilégios de root:

```
sudo -s
```

2. Instale o repositório do Zabbix:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Instale o adaptador de bridge do Zabbix:

```
apt install zabbix-bridge-adapter
```

4. Edite o arquivo de configuração do Zabbix server:

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Reinicie o Zabbix server, por exemplo (consulte outras opções):

```
systemctl restart zabbix-server
```

6. Copie o ID do Zabbix server em *Relatórios* > *Informações do sistema*; você o usará na próxima etapa.

7. Edite o arquivo de configuração do adaptador de bridge do Zabbix:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

> **Attention:**
> O diretório especificado em `DataDir` deve ter permissões 0700, e todos os arquivos armazenados nele devem ter permissões 0600.

8. Revise e aceite o contrato de licença de usuário final (EULA) do adaptador de bridge do Zabbix:

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Inicie o adaptador de bridge do Zabbix:

```
systemctl start zabbix-bridge-adapter
```

Attention:
Quando iniciado, o adaptador de bridge do Zabbix usa o ID do server para registrar seu Zabbix server no Zabbix Bridge hospedado centralmente. O estado do registro é armazenado em um arquivo de estado no `DataDir`. Se você reinstalar o adaptador de bridge e o arquivo de estado for perdido, o registro do mesmo Zabbix server poderá falhar. Para evitar isso, faça backup do arquivo de estado antes de reinstalar o adaptador de bridge e restaure-o depois.

Configurar notificações push Para receber alertas no seu dispositivo móvel quando algo exigir atenção no ambiente monitorado:

1. Crie um tipo de mídia de notificação push (ou habilite o padrão).
2. Crie uma ação (ou habilite a ação de trigger padrão *Report problems to Zabbix administrators*).
3. Adicione um dispositivo para adicionar automaticamente o tipo de mídia ao seu usuário (ou adicione mídia do usuário manualmente).

Adicionar dispositivos Para usar o aplicativo móvel do Zabbix em seu dispositivo, você deve adicionar esse dispositivo na interface web do Zabbix. Para obter instruções, consulte:

- *Usuários* > *Dispositivos* (como *Super admin*, adicione seus próprios dispositivos móveis ou os de outro usuário)
- *Configurações do usuário* > *Dispositivos* (como qualquer usuário, adicione seus próprios dispositivos móveis)

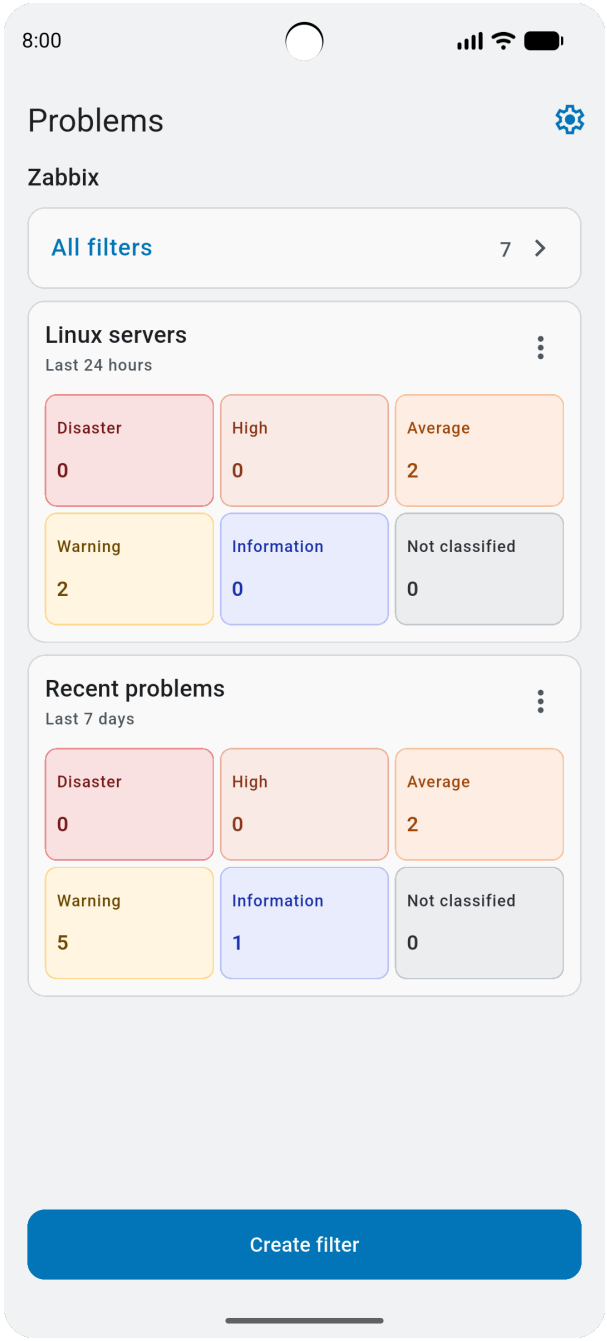
Visão geral do aplicativo

O aplicativo móvel do Zabbix (em breve) permite monitorar seu ambiente Zabbix a partir do seu dispositivo móvel.
Visualize problemas, verifique seus detalhes, atualize problemas e receba notificações push quando algo precisar de atenção.

Esta página apresenta uma visão geral do aplicativo móvel e explica como usá-lo.

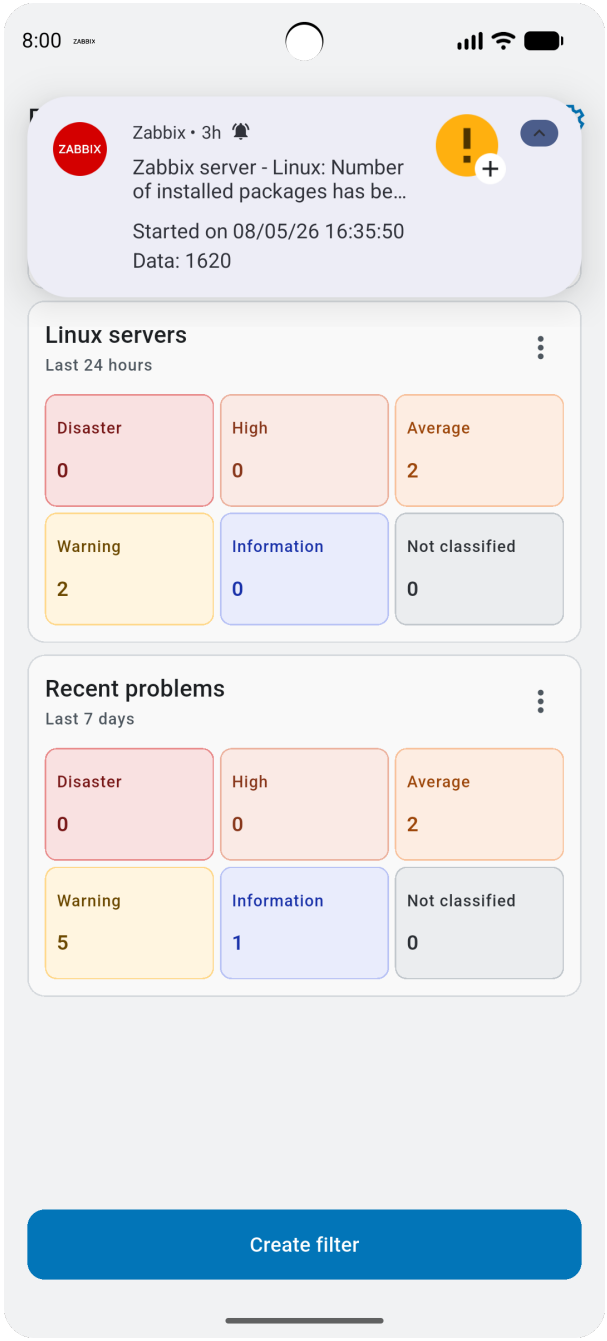
Problemas A tela *Problemas* mostra uma visão geral dos problemas com base nos seus filtros fixados. Ela contém:

- Ícone de *Configurações* - toque para ver as configurações do aplicativo.
- *Todos os filtros* - toque para ver e gerenciar os filtros.
- Nome do Zabbix server (ou *Zabbix*, se não tiver sido configurado durante a instalação da interface web do Zabbix).
- Filtros fixados - toque no nome ou em uma gravidade para abrir a tela *Lista de problemas*. Toque no menu de três pontos para gerenciar o filtro: editar, duplicar, desafixar ou excluir. Observe que os filtros fixados são ordenados alfabeticamente, e o filtro *Problemas recentes* não pode ser editado nem removido.
- *Criar filtro* - toque para criar um novo filtro.



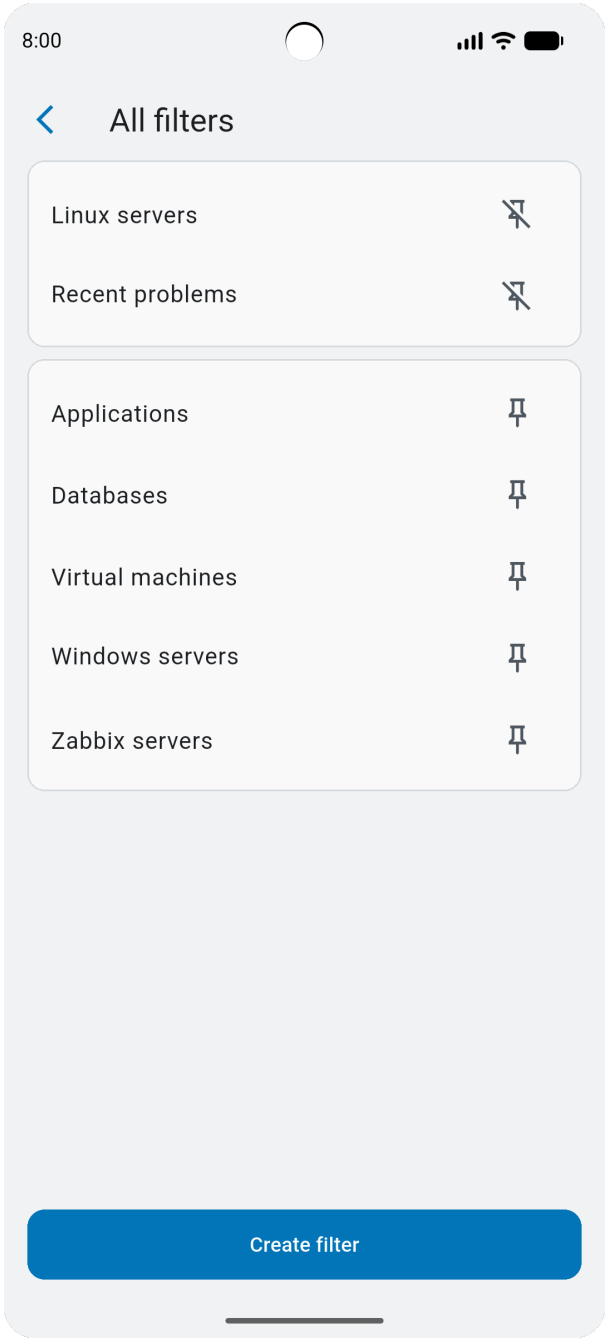
Notificações Quando o Zabbix server detecta um problema, ele pode enviar uma notificação para o seu dispositivo móvel. A notificação contém o host com problema, o nome e uma mensagem gerada a partir dos templates de mensagem configurados para o tipo de mídia de notificação push.

Toque na notificação para abrir a tela *Detalhes do problema*.



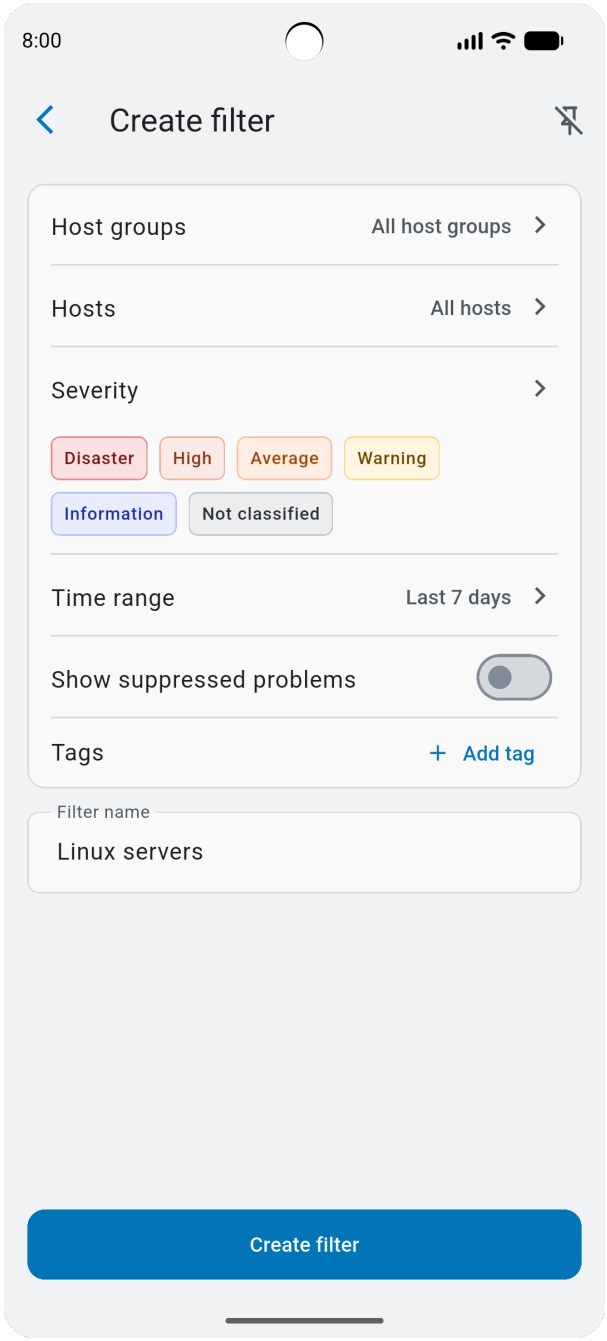
Todos os filtros A tela *Todos os filtros* permite gerenciar filtros. Ela contém:

- Nomes dos filtros - toque para editar o filtro.
- Ícones de fixação - toque para fixar ou desafixar o filtro na tela *Problemas*.
- *Criar filtro* - toque para criar um novo filtro.



Criar filtro A tela *Criar filtro* permite criar um filtro, com opções semelhantes às disponíveis em *Monitoramento* > *Problemas* (consulte detalhes). Você também pode definir um *Intervalo de tempo* para especificar por quanto tempo os problemas não resolvidos devem ser exibidos.

Toque em *Criar filtro* para salvar o filtro. Os novos filtros são fixados na tela *Problemas*; toque no ícone de fixação para alterar essa configuração.



Lista de problemas A tela *Lista de problemas* mostra os problemas que correspondem ao filtro (por exemplo, *Problemas recentes*).

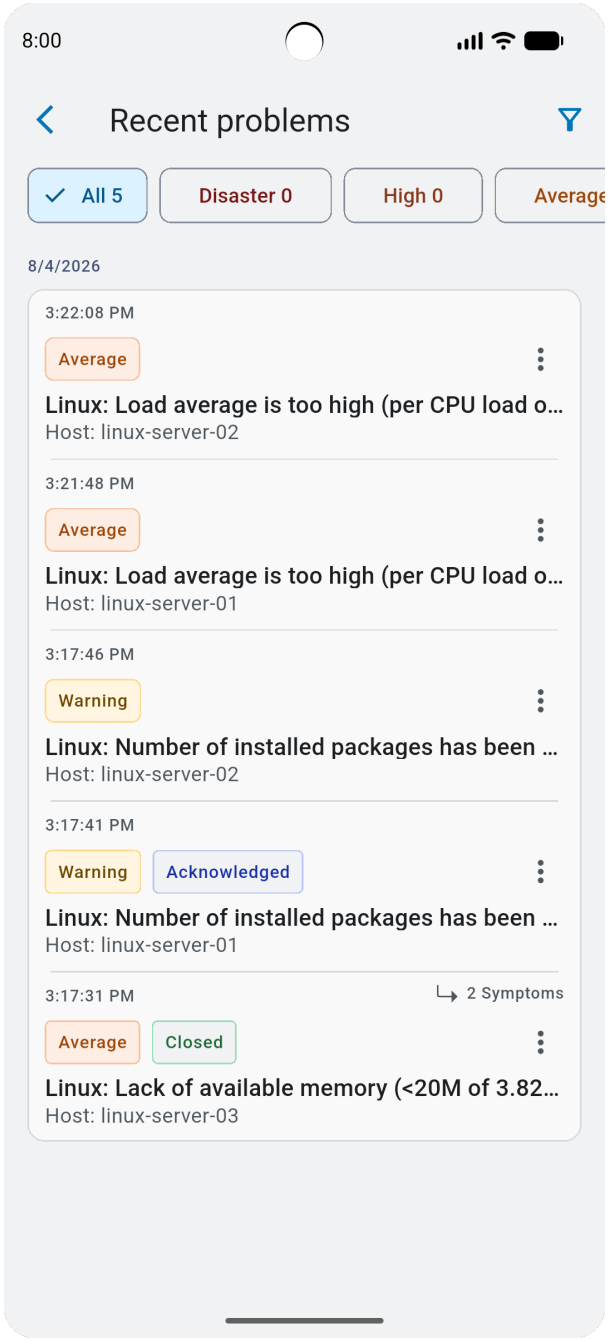
Toque no ícone de filtro ou em uma severidade de problema para restringir ainda mais a lista.

Para cada problema, você pode ver:

- Hora e data de início.
- Severidade, nome e host.
- Se foi reconhecido ou fechado recentemente (resolvido automaticamente ou fechado manualmente).
- Se possui sintomas - toque para ver a lista de problemas de sintomas.

Toque em um problema para abrir a tela *Detalhes do problema*.

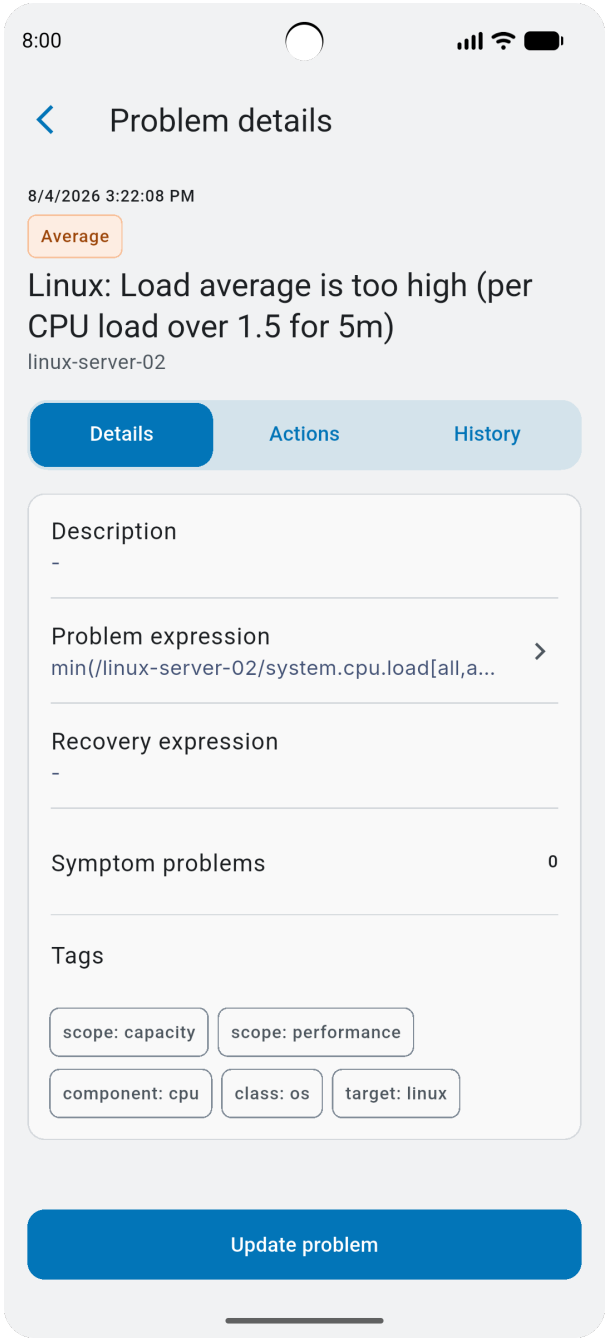
Toque no menu de três pontos para reconhecer rapidamente o problema ou suprimi-lo pelo período que você escolher.



Detalhes do problema A tela *Detalhes do problema* mostra os detalhes, as ações e o histórico do problema.

- Guia *Detalhes* - mostra a descrição do problema, as expressões, os problemas sintomáticos (toque para ver a lista de problemas) e as tags.
- Guia *Ações* - mostra a linha do tempo do problema: quando ele começou ou foi reconhecido, se uma notificação foi enviada e outras ações.
- Guia *Histórico* - mostra a idade, a duração e o tempo de recuperação do problema (ou – se o problema ainda estiver em andamento).

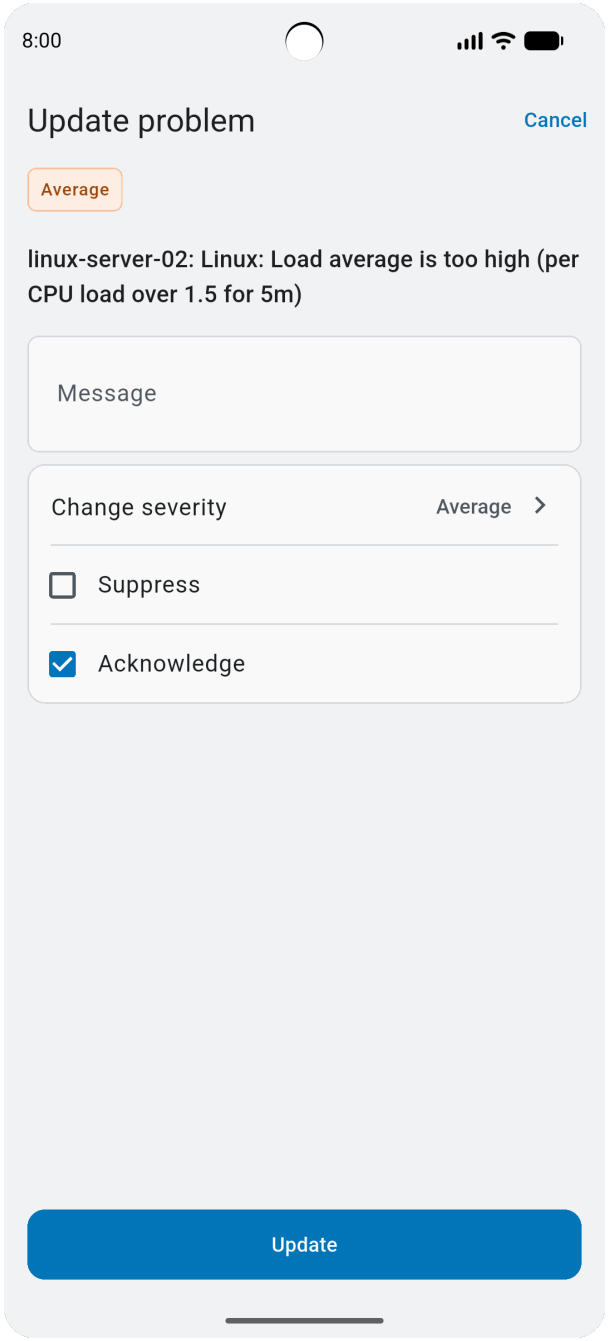
Toque em *Atualizar problema* para atualizar o problema.



Atualizar problema A tela *Atualizar problema* permite atualizar um problema:

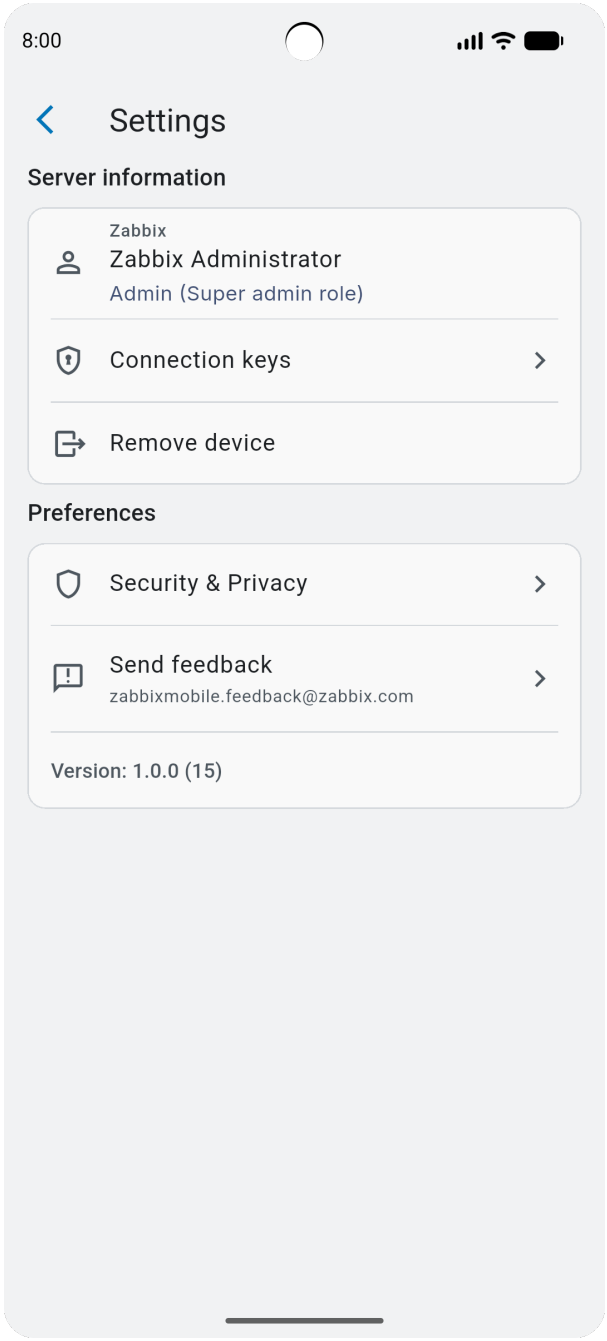
- Adicionar uma mensagem.
- Alterar a severidade do problema.
- Suprimir o problema pelo período que você escolher.
- Reconhecer ou remover o reconhecimento do problema.
- Fechar o problema.

Toque em *Atualizar* para salvar suas alterações.



Configurações A tela *Configurações* permite acessar as configurações do aplicativo. Ela contém:

- Nome do server Zabbix (ou *Zabbix*, se não tiver sido configurado durante a instalação da interface web do Zabbix).
- Nome, tipo e função do usuário ao qual o dispositivo móvel foi adicionado.
- *Chaves de conexão* - abra para visualizar e copiar o ID do server, o ID do dispositivo e outros detalhes.
- *Remover dispositivo* - toque para remover o dispositivo móvel do Zabbix.
- *Segurança e privacidade* - abra para ativar o bloqueio do aplicativo e gerenciar o compartilhamento de dados de diagnóstico.
- *Enviar feedback* - abra para enviar feedback ao Zabbix.
- Versão do aplicativo.



Acesso móvel seguro

Esta página contém práticas recomendadas para manter o aplicativo móvel do Zabbix e o adaptador Zabbix bridge seguros.

Proteja dispositivos móveis Para impedir que outras pessoas abram o aplicativo em seu dispositivo móvel, ative o bloqueio do aplicativo em *Configurações*.

Se um dispositivo for perdido ou comprometido, remova-o na interface web do Zabbix. Isso revoga e remove suas credenciais de acesso, impedindo que o dispositivo acesse o Zabbix. Dependendo das permissões da sua função de usuário, você pode remover dispositivos nas seguintes seções:

- *Configurações do usuário > Dispositivos* - qualquer usuário pode remover seus próprios dispositivos (requer a permissão *Gerenciar os próprios dispositivos*).
- *Usuários > Dispositivos* - o *Super admin* pode remover os dispositivos de qualquer usuário (requer a permissão *Gerenciar dispositivos de usuários*).

Você também pode limitar quais partes do Zabbix podem ser acessadas a partir de um dispositivo móvel. Para isso, atribua aos usuários uma função de usuário que conceda apenas as permissões necessárias para o aplicativo. As permissões mínimas são as seguintes:

Seção	Configuração
Acesso aos elementos da interface	Desmarque todas as caixas de seleção.
Acesso aos serviços	Defina *Acesso de leitura e escrita aos serviços* e *Acesso somente leitura aos serviços* como None.
Acesso aos módulos	Desmarque todas as caixas de seleção.
Acesso à API	Desmarque a caixa de seleção *Habilitado*.
Acesso às ações	Desmarque todas as caixas de seleção. Para permitir que os usuários atualizem problemas no aplicativo, marque também as ações necessárias: - *Adicionar comentários aos problemas* - *Alterar a severidade* - *Reconhecer problemas* - *Suprimir problemas* - *Fechar problemas* - *Criar e editar a própria mídia* Observe que os problemas que os usuários podem ver no aplicativo também dependem do acesso aos hosts.
Dispositivos	Marque a caixa de seleção *Habilitado*. Para permitir que os usuários adicionem e removam seus próprios dispositivos móveis, marque também a caixa de seleção *Gerenciar os próprios dispositivos*; caso contrário, um *Super admin* deverá gerenciar os dispositivos em *Usuários > Dispositivos*.

Proteja o adaptador de bridge do Zabbix O adaptador de bridge do Zabbix armazena seus dados (arquivo de estado e chaves de criptografia) no diretório especificado no parâmetro `DataDir` do arquivo de configuração do adaptador de bridge do Zabbix. Certifique-se de que o diretório tenha permissões 0700 e que todos os arquivos armazenados nele tenham permissões 0600.

O adaptador de bridge do Zabbix encaminha solicitações de dispositivos móveis para a API do Zabbix. Para restringir quais métodos da API podem ser chamados por meio do adaptador de bridge do Zabbix, use o parâmetro `AllowedAPIMethods`. Por exemplo, permita apenas o conjunto mínimo de métodos exigido pelo aplicativo móvel, ou até menos, caso não queira que alguns recursos do aplicativo móvel sejam usados (por exemplo, não permita `event.acknowledge` se os usuários do aplicativo móvel não devem atualizar problemas).

Centro do desenvolvedor

Central do desenvolvedor

Comece a desenvolver extensões personalizadas para o Zabbix com orientações sobre plugins, módulos e widgets.

Desenvolva no Zabbix

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

Biblioteca Python para o Zabbix Recursos para usar a biblioteca oficial `zabbix_utils` para interagir com a API do Zabbix e criar integrações.

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

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

Criando um plugin

Esta página destina-se àqueles que têm interesse em desenvolver seu próprio plugin (coletor de dados independente) para o Zabbix agent 2.

Um plugin personalizado amplia a funcionalidade do Zabbix agent 2 para além dos plugins integrados e plugins carregáveis fornecidos.

Cada plugin é um pacote *Go* que define a estrutura e implementa uma ou várias interfaces (*Exporter*, *Configurator*, *Runner*). Para obter mais informações, consulte Interfaces de plugin e Diagrama de conexão.

Este guia ajudará você a criar um plugin carregável personalizado.

Para obter orientações adicionais, consulte também os repositórios de:

- Plugin de exemplo
- Plugins carregáveis existentes

O que você criará Este é um tutorial passo a passo para criar um plugin carregável simples chamado **MyIP**. Este plugin implementará uma única chave de item (`myip`) que retorna o endereço IP externo do host do agent Zabbix.

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 # Mude para 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 o versionamento de `golang.zabbix.com/sdk` atualmente não é suportado, mas isso pode mudar no futuro.

Dependências adicionais podem ser instaladas conforme necessário usando `go get`.

3. Crie um arquivo `main.go` vazio para o código-fonte do plugin:

```
touch main.go
```

Agora a configuração inicial está concluída, e o plugin está pronto para desenvolvimento.

Etapa 2: Estrutura do plugin O módulo `golang.zabbix.com/sdk`, instalado na etapa anterior, fornece a estrutura necessária para o desenvolvimento de plugins e garante que todos os plugins tenham uma estrutura consistente.

1. Configure o fluxo básico de execução.

Comece definindo o fluxo principal de execução do plugin. Adicione o código a seguir a `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 do plugin. Posteriormente, a função run conterá a lógica principal do plugin.

2. Explore as interfaces do plugin.

Um plugin do Zabbix agent 2 deve ser representado por uma struct que implemente 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 tempos limite de chaves de item.
- Uma ou mais das seguintes interfaces funcionais de plugin:

- *Exporter* - realiza uma consulta e retorna um valor (ou valores), nada ou um erro; geralmente é usada em conjunto com a interface *Collector*.
- *Collector* - gerencia a coleta periódica de dados.
- *Runner* - define os procedimentos de inicialização e encerramento do plugin.
- *Watcher* - permite a consulta independente de métricas, ignorando o agendador interno do agent; é útil para monitoramento baseado em traps 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 usa as implementações padrão.

3. Crie a struct do plugin.

Agora, importe o pacote *plugin* e crie uma struct `myIP` que incorpore a struct `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

Atualmente, a struct myIP satisfaz a interface Accessor. Um método para implementar uma das interfaces funcionais de plugin, a *Exporter*, será adicionado posteriormente neste tutorial.

Etapa 3: Definir chaves de item Seu plugin precisa das chaves de item para coletar dados e fornecê-los ao 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{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

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", "Metric one description.", "metric.two", "Metric two 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 código para criar e configurar um handler:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
```

```
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Etapa 5: Implementar a coleta de dados A coleta de dados é feita por meio da interface Exporter, que descreve o método Export:

```
func Export(
  key string,             // A chave do item a ser coletado.
  params []string,        // Argumentos passados para a chave do item (`myip[arg1, arg2]`).
  context ContextProvider // Metadados sobre a coleta de dados da chave do item.
) (any, error)
```

1. Importe os pacotes necessários para requisições HTTP e leitura da 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 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
}
```

Etapa 6: Compilar e configurar o plugin

1. Para compilar o plugin, execute:

```
go mod tidy
go build
```

Isso deve criar um executável `myip` no diretório atual.

2. Configure o Zabbix agent 2 para usar o plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Substitua `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` pelo caminho do `myip` criado na etapa 5.

O nome do plugin no nome do 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/zabbix_agent2.conf -t myip
```

A saída deve conter um endereço IP externo do seu host e se parecer com isto:

```
myip                                          [s|192.0.2.1]
```

Com isso, você criou um plugin carregável simples 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)
    }
}
```

```go
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")
    }

    // Cria um novo handler.
    h, err := container.NewHandler("MyIP") // Nome do plugin
    if err != nil {
        return errs.Wrap(err, "falha ao criar novo 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 que uma solicitação de encerramento do agent seja recebida.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "falha ao executar o handler do plugin")
    }

    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("chave de item desconhecida %q", key)
    }

    // O log será encaminhado para o log 2 do agent.
    p.Infof(
        "solicitação recebida para processar a chave %q com %d parâmetros",
        key,
        len(params),
    )

    // Coleta os dados e os retorna.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "falha ao obter o endereço IP")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "falha ao ler o corpo da resposta")
```

```
    }

    return string(body), nil
}
```

Código-fonte completo

Interfaces de plugin

Esta seção descreve as interfaces de plugin disponíveis.

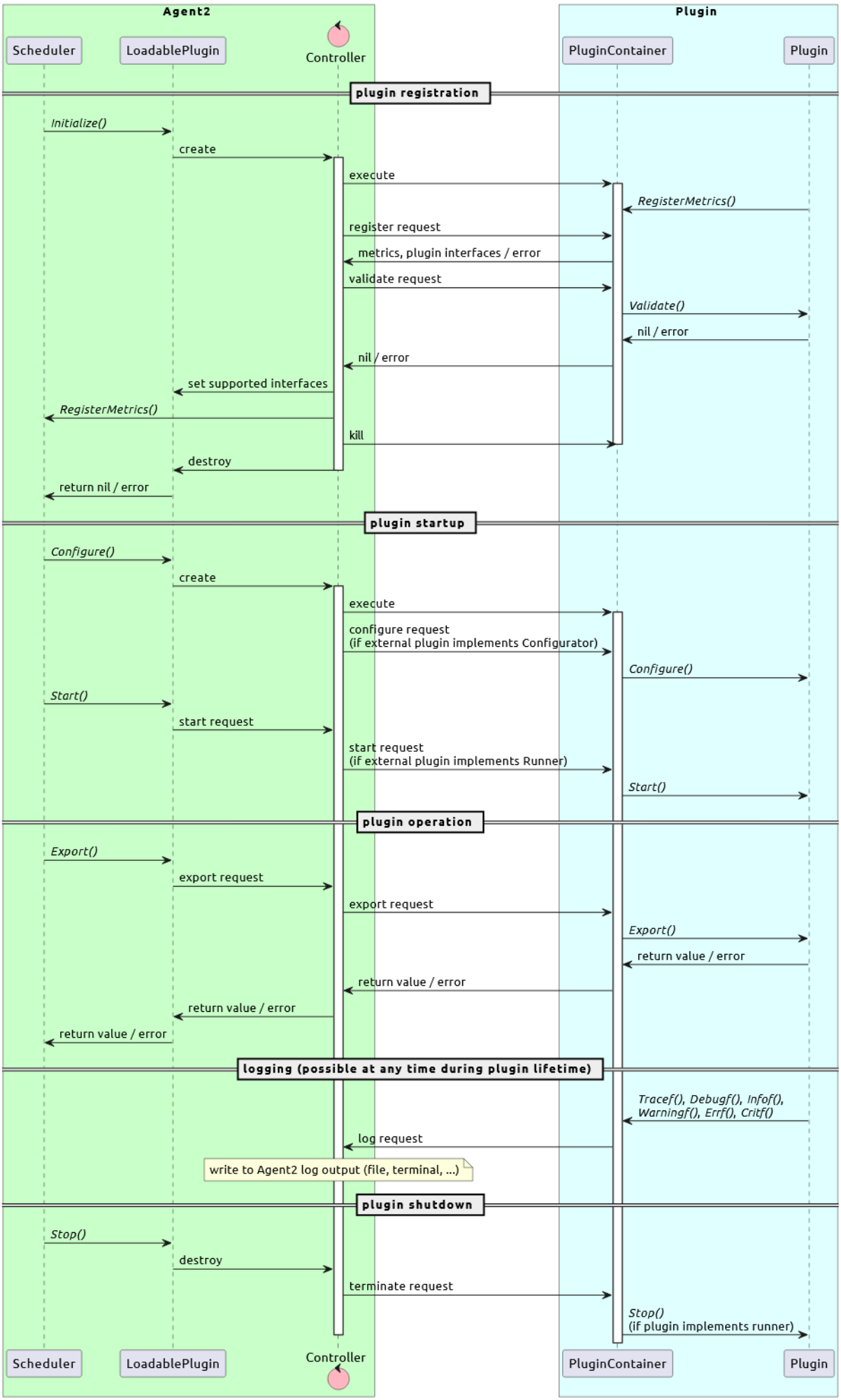
plugin.Exporter *Exporter* é a interface mais simples que executa uma coleta e retorna um valor (valores), nada ou erro. Ela aceita uma chave de item pré-analisada, parâmetros e contexto. O acesso a todas as outras interfaces de plugin é exclusivo e nenhum método pode ser chamado se um plugin já estiver 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.

Diagrama de conexão O Zabbix agent 2 se conecta 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 Zabbix agent 2 e um plugin carregável, bem como 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

Biblioteca Python para Zabbix

zabbix_utils é a biblioteca oficial do Python para o Zabbix.

Com o zabbix_utils você pode:

- Usar a API do Zabbix para gerenciar objetos do Zabbix (criar hosts, atualizar items, obter eventos, etc.).
- Coletar dados do agent do Zabbix (semelhante ao Zabbix get).
- Enviar dados para o server ou proxy do Zabbix (semelhante ao Zabbix sender).

A biblioteca suporta execução de scripts síncronos e assíncronos, permitindo lidar tanto com fluxos de trabalho simples quanto com operações paralelas em larga escala, como coleta massiva de dados, exportações em massa, processamento de alertas e dimensionamento eficiente do sistema.

A biblioteca é compatível com as versões suportadas do server Zabbix e com as versões modernas do Python.

Para começar, instale a biblioteca e siga o guia quickstart.

Instalação

zabbix_utils pode ser instalado usando qualquer um dos métodos descritos abaixo.

Requisitos:

- Zabbix 6.0 ou posterior (testado em 6.0, 7.0, 7.2, 7.4)
- Python 3.8 ou posterior (testado em 3.8–3.13)
- Framework HTTP assíncrono aiohttp (necessário apenas para o modo assíncrono)

Python Package Index (PyPI)

Este é o método mais comum para a maioria dos ambientes:

```
pip install zabbix_utils
pip install zabbix_utils[async] # Somente para suporte assíncrono
```

Repositório Zabbix

Use este método se preferir gerenciar dependências através do gerenciador de pacotes do seu sistema.

Baixe os pacotes Zabbix para sua distribuição Linux e execute os seguintes comandos.

No RHEL e derivados:

```
dnf install python3-zabbix-utils
dnf install epel-release # Apenas para suporte assíncrono
dnf install python3-aiohttp # Apenas para suporte assíncrono
```

No Debian/Ubuntu e derivados:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Apenas para suporte assíncrono
```

Fonte (GitHub)

Use este método se você preferir a versão de desenvolvimento mais recente:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Somente para suporte assíncrono
python3 setup.py install
```

Para instalar as dependências necessárias para o modo assíncrono, você também pode usar um dos métodos descritos acima.

Guia de início rápido

Depois de instalar o zabbix_utils, você pode usá-lo em seu script.

Este guia de início rápido mostra como:

- Recuperar o nome do host do agent Zabbix.
- Criar um host e um item trapper usando a API do Zabbix.
- Enviar um valor para o item.

O guia apresenta o script passo a passo, explicando cada parte à medida que é introduzida. O script completo é fornecido ao final da página.

O guia também pressupõe que seu server, agent e API do Zabbix estejam sendo executados localmente.

Recuperar o nome do host do agent Zabbix

Comece recuperando o nome do host do sistema onde o agent Zabbix está em execução. Você precisará desse nome de host para criar um host usando a API do Zabbix.

1. Importe a classe `Getter` de zabbix_utils. Esta classe funciona como o Zabbix get e permite solicitar dados do agent Zabbix.

2. Crie uma instância de `Getter` que se conecta ao agent Zabbix local em `127.0.0.1` na porta 10050.

3. Chame o método `get()` na instância de `Getter` para solicitar o nome do host ao agent Zabbix.

- O método `get()` recebe uma chave de item como parâmetro e envia uma solicitação ao agent Zabbix para esse item.
- O método `get()` retorna um objeto, e o nome do host é armazenado no atributo `value` desse objeto.
- `system.hostname` é uma chave interna do agent Zabbix que retorna o nome do host.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Conecte-se e faça login na API do Zabbix

Em seguida, conecte-se à API do Zabbix. Isso permite que seu script gerencie objetos do Zabbix, como hosts e items.

1. Importe a classe `ZabbixAPI` de zabbix_utils.

2. Crie uma instância de `ZabbixAPI` e forneça a URL da interface web do Zabbix.

3. Chame o método `login()` na instância de `ZabbixAPI` e forneça seu nome de usuário e senha. A conta deve ter permissão para acessar a API do Zabbix.

4. Chame o método `logout()` para fechar a sessão após as operações da API.

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### As operações da API vão aqui

api.logout()
```

Criar host no Zabbix

Agora que você tem o nome do host do agent Zabbix e está conectado à API do Zabbix, pode criar um novo host no Zabbix.

1. Chame o método da API `host.create()` na instância `ZabbixAPI` e forneça os detalhes do host:

- `host` - defina como a variável `hostname` contendo o nome do host que você recuperou do agent Zabbix.
- `interfaces` - contém detalhes de conexão para o agent Zabbix que está sendo executado no host.
- `groups` - contém pelo menos um grupo de hosts ao qual o host deve pertencer.

2. Como a API do Zabbix retorna o ID do host recém-criado, salve esse ID na variável `host_id` para uso posterior.

3. Imprima uma mensagem para confirmar que o host foi criado.

```
api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
```

```
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' criado com ID {host_id}")
```

Crie um novo item no Zabbix

Após criar um host, você pode adicionar um item a ele.

1. Defina uma chave exclusiva para o item a ser criado.

2. Chame o método `item.create()` na instância `ZabbixAPI` e forneça os detalhes do item:

- `hostid` - defina como a variável `host_id` contendo o ID do host que você acabou de criar.
- `name` - um nome para o item.
- `key_` - defina como a variável `item_key` que você acabou de definir.
- `type` - defina como 2 (item trapper), necessário para receber valores enviados do seu script nas etapas seguintes.
- `value_type` - defina como 3 (numérico sem sinal), o tipo de dado que este item armazena.

3. Como a API do Zabbix retorna o ID do item recém-criado, salve este ID na variável `item_id` para uso posterior.

4. Imprima uma mensagem para confirmar que o item foi criado.

```
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3,
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' criado com ID {item_id}")
```

Enviar valor para o host

Agora que você criou seu host e item, pode enviar dados para ele.

1. Importe a classe `time` e aguarde alguns segundos antes de enviar os dados. Isso garante que o Zabbix tenha processado totalmente seu novo host e item.

2. Importe a classe `Sender` de zabbix_utils. Esta classe funciona como o Zabbix sender e permite que seu script envie dados para o Zabbix.

3. Crie uma instância de `Sender` que se conecta ao Zabbix server local em `127.0.0.1` na porta 10051.

4. Chame o método `send_value()` na instância de `Sender` para enviar um valor para o item do seu host e forneça estes detalhes:

- `hostname` - defina como a variável `hostname` que contém o nome do host que você recuperou do Zabbix agent.
- `item_key` - defina como a variável `item_key` que você acabou de definir anteriormente em seu script.
- 1 - o valor a ser enviado.

4. Imprima uma mensagem para confirmar que o valor foi enviado.

```
import time
time.sleep(10)

from zabbix_utils import Sender

sender = Sender(server='127.0.0.1', port=10051)

response = sender.send_value(hostname, item_key, 1)
```

```
print(f"Sender response: {response}")
```

Após a execução bem-sucedida deste código, você deverá ver uma mensagem de sucesso indicando que o Zabbix recebeu seu valor:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Agora você também pode verificar sua interface web do Zabbix (*Monitoramento* > *Últimos dados*) para ver o valor.

Script completo

Abaixo está o script completo que combina todas as etapas: recuperando o nome do host, criando o host e o item usando a API do Zabbix e enviando dados para o Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Recupera o nome do host do agent do Zabbix
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Conecta e faz login na API do Zabbix
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Cria o host no Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' criado com ID {host_id}")

### Cria o item no Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' criado com ID {item_id}")

### Faz logout da API
api.logout()

### Aguarda o Zabbix processar o novo host
time.sleep(10)

### Envia valor para o host
```

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Resposta do sender: {response}")
```

Usar a API do Zabbix

O zabbix_utils permite que você use a API do Zabbix para gerenciar objetos do Zabbix, incluindo a criação de hosts, atualização de items, recuperação de eventos e muito mais.

As requisições à API podem ser feitas em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python envia uma requisição e aguarda uma resposta antes de continuar, o que é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script envia requisições sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para requisições lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório do GitHub zabbix_utils.

Importação

Para usar o zabbix_utils para trabalhar com a API do Zabbix, importe a classe `ZabbixAPI` em seu script:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Se você já usa uma biblioteca Python da comunidade, normalmente pode substituir essa importação por `ZabbixAPI` do zabbix_utils.

Fazer login

Antes de fazer solicitações à API, você deve criar uma instância `ZabbixAPI` e fazer login na API do Zabbix.

Escolha o método que melhor se adapta à forma como você gerencia as credenciais. Você pode usar seu nome de usuário e senha ou tokens de API.

Durante a inicialização

Ao criar uma instância do `ZabbixAPI`, você pode fornecer a URL da interface web do Zabbix e suas credenciais de uma só vez:

```
### Nome de usuário e senha:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### Token da API:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Se preferir agrupar os parâmetros de conexão, você pode passá-los como um dicionário Python e desempacotá-los usando a sintaxe de argumentos nomeados do Python:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Usando `login()`

Você pode primeiro criar uma instância `ZabbixAPI` apenas com a URL da interface web do Zabbix e, posteriormente, usar o método `login()` com suas credenciais:

```
### Nome de usuário e senha:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")
```

```
### Token da API:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Usando variáveis de ambiente

Você pode primeiro armazenar suas credenciais como variáveis de ambiente, por exemplo, usando um terminal:

```
### Nome de usuário e senha:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
```

Em seguida, se você iniciar seu script a partir da mesma sessão do terminal, a instância `ZabbixAPI` no script requer apenas a URL da interface web do Zabbix:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Você também pode armazenar a URL em uma variável de ambiente:

```
### Nome de usuário e senha:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

Quando a URL e as credenciais estão todas definidas como variáveis de ambiente, a instância `ZabbixAPI` no script não requer nenhum argumento:

```
api = ZabbixAPI()
```

Sair

Se você fez login usando um nome de usuário e senha, chame o método `logout()` após concluir suas operações de API:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### As ações da API vão aqui

api.logout()
```

Note:
Chamar o método `logout()` não é necessário ao usar tokens de API.

Requisições de API

Depois de fazer login, você pode fazer qualquer requisição de API chamando os métodos descritos na referência de métodos da API do Zabbix.

Os métodos da API são chamados usando o seguinte formato:

```
api_instance.zabbix_object.method(parameters)
```

Note:
Alguns nomes de métodos ou objetos da API do Zabbix usam palavras que são palavras reservadas em Python (por exemplo, `import`). Para evitar erros em Python, adicione um sublinhado ao nome do método ou objeto ao usá-lo em Python (por exemplo, `api.configuration.import_`).

Por exemplo, para ver uma lista de hosts com `host.get`:

```
### 1. Importe ZabbixAPI de zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Crie a instância do ZabbixAPI e faça login:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Recupere hosts que correspondam a um filtro:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Itere sobre os hosts retornados e imprima os detalhes de cada host:
for host in hosts:
    print(host)

### 5. Faça logout da API para encerrar a sessão:
api.logout()
```

Modo assíncrono

O modo assíncrono permite que seu script envie solicitações de API sem esperar que cada uma termine. Isso pode tornar seu script mais eficiente quando ele precisa executar muitas solicitações de API ou quando algumas solicitações demoram muito.

Ao usar o modo assíncrono, há diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncZabbixAPI` em vez de `ZabbixAPI`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar métodos da API, incluindo `login()` e `logout()`.

> **Attention:**
> Ao criar uma instância de `AsyncZabbixAPI`, você não pode fazer login durante a inicialização.

Por exemplo, para ver uma lista de hosts com `host.get` no modo assíncrono:

```
### 1. Importe asyncio para modo assíncrono e AsyncZabbixAPI de zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Defina a função principal async onde todas as operações da API serão executadas:
async def main():

    # 3. Crie uma instância de AsyncZabbixAPI e faça login (deve usar await):
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Recupere hosts que correspondam a um filtro (deve usar await):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Itere sobre os hosts retornados e imprima os detalhes de cada host:
    for host in hosts:
```

```
        print(host)

    # 6. Faça logout da API para fechar a sessão (deve usar await):
    await api.logout()

### 7. Execute a função async main() usando o event loop do asyncio:
asyncio.run(main())
```

Exemplos

Os exemplos a seguir mostram tarefas comuns da API do Zabbix usando a biblioteca zabbix_utils.

Exemplos adicionais estão disponíveis no diretório `examples` do repositório zabbix_utils no GitHub.

Coletar dados do agent Zabbix

Visão geral

zabbix_utils permite coletar dados do agent Zabbix (semelhante ao Zabbix get).

Os dados podem ser coletados em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python solicita e aguarda os dados antes de continuar, o que é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script solicita dados sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para solicitações lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório do GitHub zabbix_utils.

Importação

Para usar o zabbix_utils para coletar valores de item, importe a classe `Getter` em seu script Python:

```
from zabbix_utils import Getter
```

Dados da solicitação

Para solicitar o valor de um item:

1. Crie uma instância de `Getter`, especificando o endereço IP e a porta do seu agent do Zabbix.
2. Chame o método `get()` na instância de `Getter`, especificando a chave do item que você deseja recuperar.

Por exemplo, para solicitar dados do item `system.uname`:

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Usando IP não padrão

Se o server que executa seu script tiver vários endereços IP, você pode especificar um `source_ip` para o `Getter` usar ao se conectar ao agent do Zabbix:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Usando timeout

Você pode definir um `timeout` de resposta para o `Getter` para controlar por quanto tempo seu script deve aguardar uma resposta do agent do Zabbix antes de desistir:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Usando criptografia

O `Getter` não inclui suporte nativo a criptografia, mas você pode fornecê-lo criando um wrapper usando bibliotecas de terceiros:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementação do wrapper TLS PSK para o socket
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Resposta

A resposta do agent Zabbix é processada pela biblioteca e retornada como um objeto `AgentResponse`:

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

Modo assíncrono

O modo assíncrono permite que seu script colete valores sem esperar que cada um deles chegue. Isso pode tornar seu script mais eficiente quando ele precisa coletar muitos valores ou quando alguns valores demoram muito para serem coletados.

Ao usar o modo assíncrono, há diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncGetter` em vez de `Getter`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar o método `get()`.

Por exemplo, para coletar um único valor usando o modo assíncrono:

```
### 1. Importe asyncio para o modo assíncrono e AsyncGetter de zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Defina a função principal async onde todas as solicitações de dados serão executadas:
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Busque o valor system.uname do agent do Zabbix (é necessário usar await):
    response = await agent.get('system.uname')

    # 4. Exiba o valor retornado pelo agent do Zabbix:
    print(response.value)

### 5. Execute a função principal async main() usando o event loop do asyncio:
asyncio.run(main())
```

Enviar dados para o Zabbix server ou proxy

O zabbix_utils permite enviar valores de item para um item trapper no Zabbix server ou proxy (de forma semelhante ao Zabbix sender).

Você pode enviar um único valor, múltiplos valores ou até mesmo direcionar múltiplos clusters Zabbix.

Os dados podem ser enviados em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python envia valores e aguarda uma resposta antes de continuar; isso é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script envia valores sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para solicitações lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório GitHub do zabbix_utils.

Importação

Para usar o zabbix_utils para enviar valores de item, importe a classe `Sender` em seu script:

```
from zabbix_utils import Sender
```

Para enviar múltiplos valores, você também pode importar a classe `ItemValue`:

```
from zabbix_utils import Sender, ItemValue
```

Enviar valor único

Para enviar o valor de um item:

1. Crie uma instância de `Sender`, especificando o endereço IP e a porta do seu Zabbix server ou proxy.
2. Chame o método `send_value()` na instância de `Sender` usando o seguinte formato:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Por exemplo, para enviar 1 para o item trapper `service.status` no host `Linux server`:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Usando IP não padrão

Se o servidor que executa seu script tiver vários endereços IP, você pode especificar um `source_ip` para o `Sender` usar ao enviar valores para o Zabbix server ou proxy:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Usando timeout

Você pode definir um `timeout` de resposta para o `Sender` para controlar quanto tempo seu script deve aguardar por uma resposta do Zabbix server ou proxy antes de desistir:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Usando o arquivo de configuração do agent

Você pode permitir que `zabbix_utils` leia os parâmetros `Server` ou `ServerActive` de um arquivo de configuração local do agent ou do agent 2 do Zabbix. Nesses casos, não é necessário especificar os parâmetros de conexão ao criar uma instância de `Sender`:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Se `ServerActive` contiver um ou mais clusters do Zabbix com várias instâncias de server, `Sender` enviará os dados para o primeiro server disponível em cada cluster. Se `ServerActive` não estiver definido, será utilizado o endereço de `Server` com a porta padrão (10051).

Usando criptografia

O `Sender` não inclui suporte de criptografia embutido, mas você pode fornecê-lo criando um wrapper usando bibliotecas de terceiros:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementação do wrapper TLS PSK para o socket
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Resposta para valor único

A resposta retornada pelo server ou proxy Zabbix é processada pela biblioteca e retornada como um objeto `TrapperResponse`:

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

Modo assíncrono

O modo assíncrono permite que seu script Python envie valores sem esperar por uma resposta do Zabbix server ou proxy. Isso pode tornar seu script mais eficiente quando ele precisa enviar muitos valores ou quando alguns valores demoram muito para serem enviados.

Ao usar o modo assíncrono, há algumas diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncSender` em vez de `Sender`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar o método `send_value()`.

Por exemplo, para enviar um único valor usando o modo assíncrono:

```
### 1. Importe asyncio para o modo assíncrono e AsyncSender do zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Defina a função principal async onde todas as operações de envio de dados (devem usar await) serão
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Imprima a resposta retornada pelo Zabbix server ou proxy:
    print(response)

### 4. Execute a função async main() usando o event loop do asyncio:
asyncio.run(main())
```

Enviar múltiplos valores

Para enviar múltiplos valores:

1. Prepare um array de objetos `ItemValue`, cada um usando o mesmo formato do método `send_value()`.
2. Crie uma instância de `Sender`, especificando o endereço IP e a porta do seu Zabbix server ou proxy.
3. Chame o método `send()` (em vez de `send_value()`) na instância de `Sender`, especificando o array de objetos com os valores a serem enviados.

Por exemplo, para enviar cinco valores para diferentes hosts:

```
items = [
    ItemValue('server-de', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-fr', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-uk', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-nl', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-pl', 'service.status', 'up', 1770887205, 100),
]

sender = Sender(server='127.0.0.1', port=10051)
response = sender.send(items)
```

Usando tamanho de bloco personalizado

Se você precisar enviar mais valores do que um item trapper pode aceitar em uma única solicitação, poderá dividi-los em blocos.

Por padrão, o tamanho do bloco é de 250 valores. Você pode alterá-lo definindo o parâmetro `chunk_size` ao criar uma instância de `Sender`.

Por exemplo, para enviar cinco valores em três blocos (2-2-1), defina o parâmetro `chunk_size` como 2:

```
items = [
    ItemValue('server-de', 'service.status', 'up'),
    ItemValue('server-fr', 'service.status', 'up'),
    ItemValue('server-uk', 'service.status', 'up'),
    ItemValue('server-nl', 'service.status', 'up'),
    ItemValue('server-pl', 'service.status', 'up'),
]

sender = Sender(server='127.0.0.1', port=10051, chunk_size=2)
response = sender.send(items)
```

Enviar valores para vários clusters Zabbix

Para enviar valores para vários clusters Zabbix:

1. Prepare um array de clusters Zabbix. Se um cluster tiver vários nós, o valor será enviado para o primeiro nó **disponível** de cada cluster.
2. Crie um `Sender`, especificando seu array de clusters Zabbix.
3. Chame o método `send_value()` na instância do `Sender` usando o mesmo formato do método `send_value()`.

Por exemplo, para enviar um valor para o primeiro nó disponível em cada cluster:

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

Resposta para múltiplos valores

Por padrão, o `Sender` retorna um resultado agregado do envio de valores em todos os hosts ou clusters:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Se você precisar de informações mais detalhadas, pode inspecionar os resultados para cada cluster e cada chunk usando o atributo `response.details`:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

```
if response.failed == 0:
    print(f"Value sent successfully in {response.time}")
else:
    print(response.details)
    # {
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"Processed {resp.processed} of {resp.total} at {node.address}:{node.port}")
            # Processed 1 of 1 at 127.0.0.1:10051
            # Processed 1 of 1 at zabbix.example.local:10051
```

Log de depuração

Para solucionar problemas ao trabalhar com a API do Zabbix, coletar dados do agent do Zabbix ou enviar dados para o server ou proxy do Zabbix, você pode habilitar o log de depuração.

A biblioteca zabbix_utils utiliza o módulo padrão `logging` do Python.

Para ver mensagens detalhadas de depuração da biblioteca:

1. Importe o módulo `logging` do Python em seu script.
2. Configure o sistema de logging para exibir mensagens em um formato legível e defina o nível de logging como DEBUG.

```
import logging
from zabbix_utils import Getter

logging.basicConfig(
    format=u'[%(asctime)s] %(levelname)s %(message)s',
    level=logging.DEBUG
)

agent = Getter(host='127.0.0.1', port=10050)
resp = agent.get('system.uname')

print(resp.value)
```

Com o log de depuração ativado, informações detalhadas sobre solicitações e respostas são impressas, por exemplo:

```
[2023-10-01 12:00:01,587] DEBUG Content of the packet: b'ZBXD\x01\x0c\x00\x00\x00\x00\x00\x00\x00system.un
[2023-10-01 12:00:01,722] DEBUG Zabbix response header: b'ZBXD\x01C\x00\x00\x00C\x00\x00\x00'
[2023-10-01 12:00:01,723] DEBUG Zabbix response body: Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
[2023-10-01 12:00:01,724] DEBUG Response from [127.0.0.1:10050]: Linux test_server 5.15.0-3.60.5.1.el9uek.
```

```
Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
```

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 module Um processo de escrita de um módulo de exemplo consiste nas seguintes etapas (quando disponível, clique no nome do arquivo ou da 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 do dashboard) e defina as regras de inicialização e tratamento de eventos.
6. Crie a pasta *assets* para arquivos JavaScript (coloque em *assets/js*), estilos CSS (coloque em *assets/css*) ou quaisquer outros arquivos adicionais.
7. Certifique-se de especificar as views, actions e arquivos de assets 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. Depois que o módulo estiver habilitado, você poderá visualizar imediatamente todas as alterações feitas nos arquivos do módulo ao atualizar 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*).

Como 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 ver uma descrição detalhada):

Parameter	Description	Required
manifest_version	Versão do manifesto do módulo.	Yes
id	ID exclusivo 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*	Yes for widgets, otherwise no
widget	Configuração do widget. Usado apenas para widgets.	
actions	Ações a serem registradas com o módulo.	
assets	Arquivos CSS e JavaScript a serem incluídos.	No
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, esse link será aberto ao clicar no ícone de ajuda ? na janela *Adicionar widget* ou *Editar widget*. Se **url** não for especificado, clicar no ícone de ajuda abrirá 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 requisição 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 true, o Zabbix chamará 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:

```
/**
 * Validate input parameters.
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * Check user permissions.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Execute action and generate response object.
 */
protected function doAction(): void {
    $data = [
        'hosts_count' => API::Host()->get([
```

```
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
Você pode ver a lista completa de classes de controller disponíveis no código-fonte do Zabbix.

Views

O arquivo de view recebe os dados de um controller e, em seguida, prepara a aparência HTML deles.

Note:
Definir view(s) para um módulo do frontend é opcional, a menos que o módulo seja um widget.
 Widgets de dashboard precisam de pelo menos duas views: uma para o modo de edição e outra para o modo de visualização (devem ser armazenadas no diretório *views*).

É possível usar classes HTML predefinidas do Zabbix (de */zabbix/ui/include/classes/html*) na view, bem como adicionar novas classes HTML e CSS. As novas classes devem ser armazenadas na pasta *assets* do módulo.

Exemplo:

```
...
(new CColHeader(_('Name')))
```

Isso adicionará um novo nome de coluna *Name* e estilizará a linha superior da tabela como nas 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']
- **Azul (clássico)** - [theme='blue-classic-theme']
- **Escuro** - [theme='dark-theme']
- **Azul escuro** - [theme='dark-blue-theme']
- **Escuro (clássico)** - [theme='dark-classic-theme']
- **Claro de alto contraste** - [theme='hc-light']
- **Escuro de alto contraste** - [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 *Administration* > *General* > *Modules* e clique em *Scan directory*.



Localize seu módulo na lista e ative-o.

Para ativar um módulo, clique no hiperlink *Disabled* - o estado do módulo mudará para *Enabled*.

Clique no nome do módulo para ver informações adicionais sobre o módulo, como autor, versão ou 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
<?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* oferece os seguintes métodos:

- *addField()* - recebe uma instância da classe *CWidgetField* como parâmetro; cada classe *CWidgetField* tem uma classe *CWidgetFieldView* correspondente 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 combina visualmente (com uma borda) os campos em um grupo.
- *includeJsFile()* - permite adicionar um arquivo JavaScript à visualização de configuração do widget.
- *addJavaScript()* - permite adicionar JavaScript embutido, que será executado assim que a visualização de configuração do widget for carregada.

A classe *CWidgetFormView* deve estar localizada no diretório *views*.

Exemplo de views/widget.edit.php

```php
<?php

/**
 * My custom widget form view.
 *
```

```
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

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 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 of int32	Várias caixas de seleção em um único campo de configuração.
CWidgetFieldColor	string	Campo de seleção de cor.
CWidgetFieldDatePicker	string	Campo de seleção de data.
CWidgetFieldIntegerBox	int32	Campo para inserir um número inteiro. Pode ser usado para configurar valores mínimos e máximos.
CWidgetFieldLatLng	string	Caixa de texto para inserir latitude, longitude e nível de zoom do mapa, separados por vírgulas.
CWidgetFieldMultiSelect	-	Classe base para campos de seleção múltipla, estendida por todas as classes `CWidgetFieldMultiSelect*`.
CWidgetFieldMultiSelectAction	ID	Campo de seleção múltipla para selecionar ações na 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.
CWidgetFieldMultiSelectHostInventory	array of int32	Campo de seleção múltipla para selecionar campos do inventário de hosts.
CWidgetFieldMultiSelectItem	ID	Campo de seleção múltipla para selecionar itens.
CWidgetFieldMultiSelectItemPrototype	ID	Campo de seleção múltipla para selecionar protótipos de itens.
CWidgetFieldMultiSelectMap	ID	Campo de seleção múltipla para selecionar mapas.
CWidgetFieldMultiSelectMediaType	ID	Campo de seleção múltipla para selecionar tipos de mídia.
CWidgetFieldMultiSelectOverrideHost	ID	Campo de seleção múltipla para selecionar uma fonte de dados para hosts de um dashboard ou de outro widget.
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.
CWidgetFieldNavTree	array of (multiple mixed)	Armazena uma estrutura de árvore de navegação de mapas para uso no widget *Árvore de navegação de mapas*.
CWidgetFieldNumericBox	string	Campo para inserir um número de ponto flutuante.
CWidgetFieldPatternSelect	-	Classe base para campos de seleção por padrão, estendida por todas as classes `CWidgetPatternSelect*`.
CWidgetFieldPatternSelectHost	string	Campo de seleção múltipla para selecionar hosts. Oferece suporte a padrões de nomes de hosts.
CWidgetFieldPatternSelectItem	string	Campo de seleção múltipla para selecionar itens. Oferece suporte a padrões de nomes de itens.
CWidgetFieldRadioButtonList	int32	Grupo de botões de opção composto por um ou mais botões de opção.
CWidgetFieldRangeControl	int32	Controle deslizante para selecionar um valor do tipo inteiro.
CWidgetFieldReference	string	Armazena um identificador exclusivo para este widget no dashboard; usado para a comunicação entre widgets.
CWidgetFieldSelect	int32	Caixa de seleção suspensa.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* predefinido com severidades de trigger.
CWidgetFieldSparkline	array of (multiple mixed)	Permite configurar um gráfico sparkline.
CWidgetFieldTags	array of (string, int32, string)	Permite configurar uma ou mais linhas de filtros de tags.
CWidgetFieldTextArea	string	Área de texto para inserir texto com várias linhas.
CWidgetFieldTextBox	string	Caixa de texto para inserir texto de uma única linha.
CWidgetFieldThresholds	array of (string, string)	Permite configurar pares de cores e números.
CWidgetFieldTimePeriod	array of string	Campo de seleção de período de tempo.
CWidgetFieldTimeZone	string	Lista suspensa com fusos horários.
CWidgetFieldUrl	string	Caixa de texto para inserir URLs.

> **Note:**
> O comprimento máximo dos campos `int32` é de 10 caracteres, dos campos `ID` é de 20 caracteres e dos campos `string` é de 65535 caracteres. Para tipos compostos (por exemplo, `array of (string, int32, string)`), o limite se aplica a cada elemento individualmente. Restrições de comprimento específicas do campo podem ser definidas usando métodos como `setMaxLength()`.

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 invocado quando o dashboard entra no modo de edição, normalmente quando um usuário interage com o botão *Editar* do widget ou com o botão *Editar dashboard* do dashboard.

Exemplo:

```
onEdit() {
    this._deactivateGraph();
}
```

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

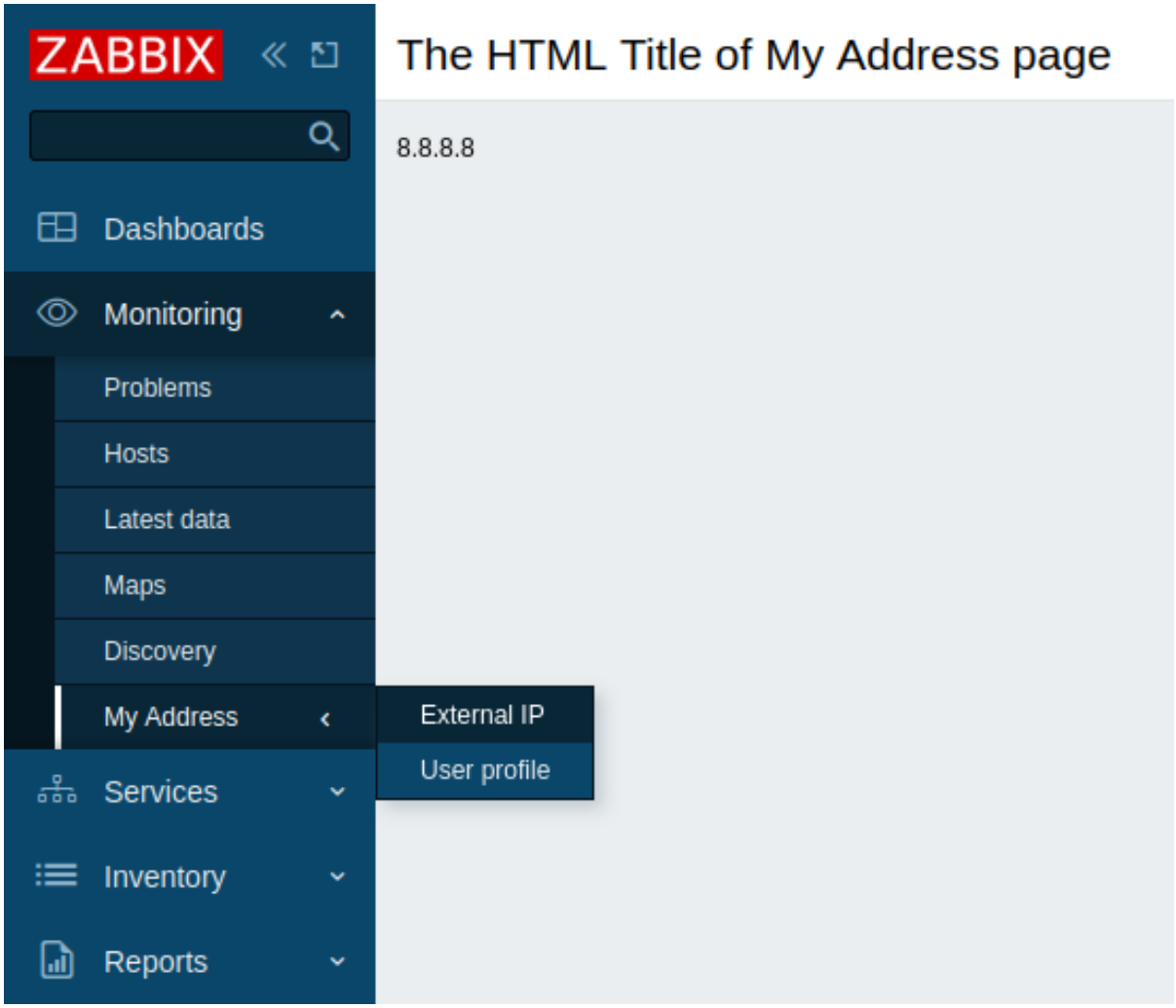
Esta seção contém tutoriais práticos passo a passo para ilustrar como construir um módulo e um widget personalizados no Zabbix.

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 vai criar

Durante este tutorial, você primeiro criará um módulo de frontend que adiciona uma nova seção de menu *My Address* e depois o converterá 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 *My Address* recém-criada. Veja como o módulo final ficará:



Parte I - Nova seção de menu

Adicionar 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).

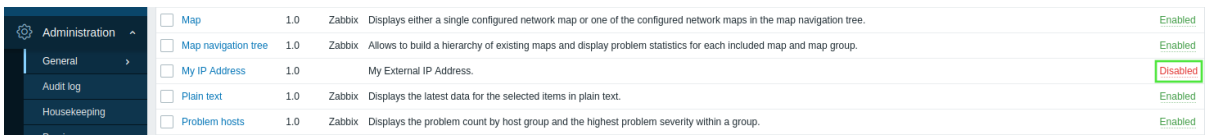
ui/modules/MyAddress/manifest.json

```
{
    "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 *Administration* > *General* > *Modules* e clique no botão *Scan directory*.



4. Localize o novo módulo *My IP Address* na lista e clique no hiperlink ”Disabled” para alterar o status do módulo de ”Disabled” para ”Enabled” (se o módulo não estiver listado, consulte a seção de troubleshooting).



O módulo agora está registrado no frontend. No entanto, ele ainda não está visível, porque você ainda precisa definir a funcionalidade do módulo. Depois de adicionar conteúdo ao diretório do módulo, você verá imediatamente as alterações no frontend do Zabbix ao atualizar a página.

Criar uma seção de menu

1. Crie um arquivo *Module.php* no diretório *MyAddress*.

Esse arquivo implementa uma nova classe *Module* que estende a classe padrão *CModule*. A classe *Module* inserirá uma nova seção de menu *My Address* no menu principal.

O método *setAction()* especifica uma ação a ser executada ao clicar na seção de menu. Para começar, você pode usar a ação predefinida *userprofile.edit*, que abrirá a página *User profile*. Na parte III deste tutorial, você aprenderá a criar uma ação personalizada.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

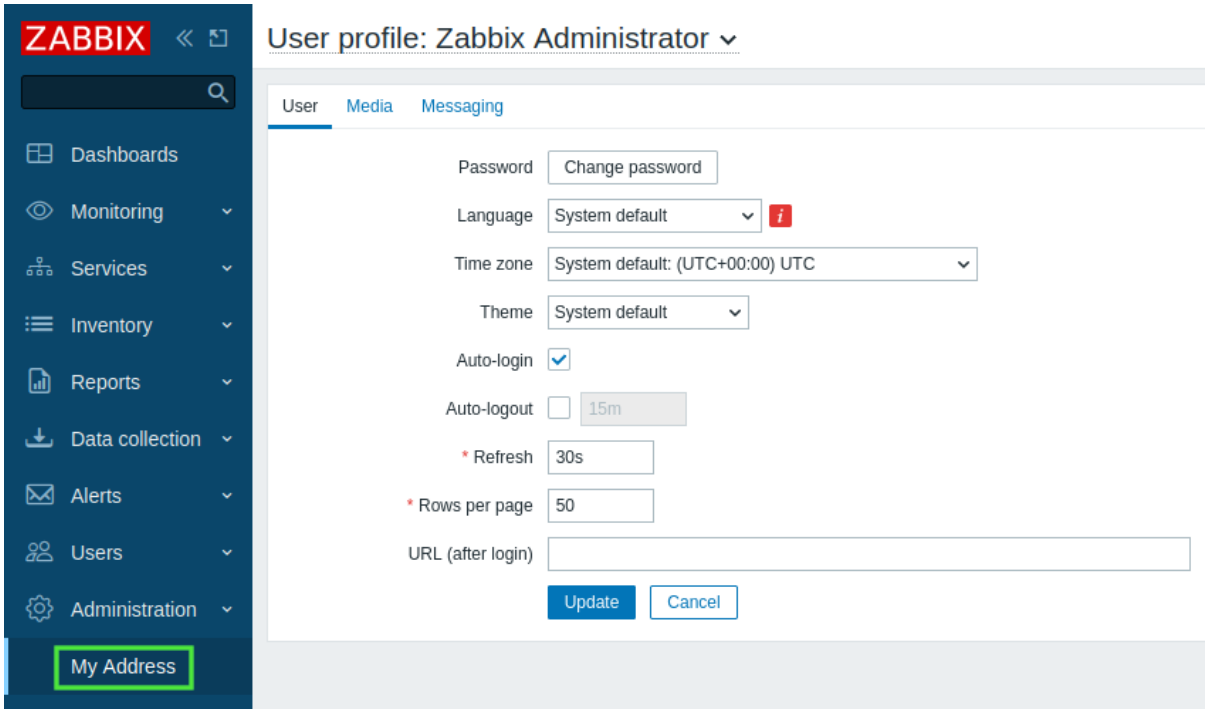
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Você pode substituir `'userprofile.edit'` por outras ações, por exemplo, `'charts.view'` (abre gráficos personalizados), `'problems.view'` (abre *Monitoring* > *Problems*), ou `'report.status'` (abre o relatório *System information*).

3. Atualize o frontend do Zabbix. Agora há uma nova seção *My Address* na parte inferior do menu principal do Zabbix. Clique em *My Address* para abrir a página *User profile*.



Parte II - Alteração da localização da seção de menu

Nesta parte, você moverá a seção de menu *My Address* para a seção *Monitoring* e, em seguida, adicionará a ela um menu aninhado. Como resultado, os usuários poderão acessar duas páginas de submenu a partir da seção de menu *Monitoring* > *My Address*.

1. Abra e edite o arquivo *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

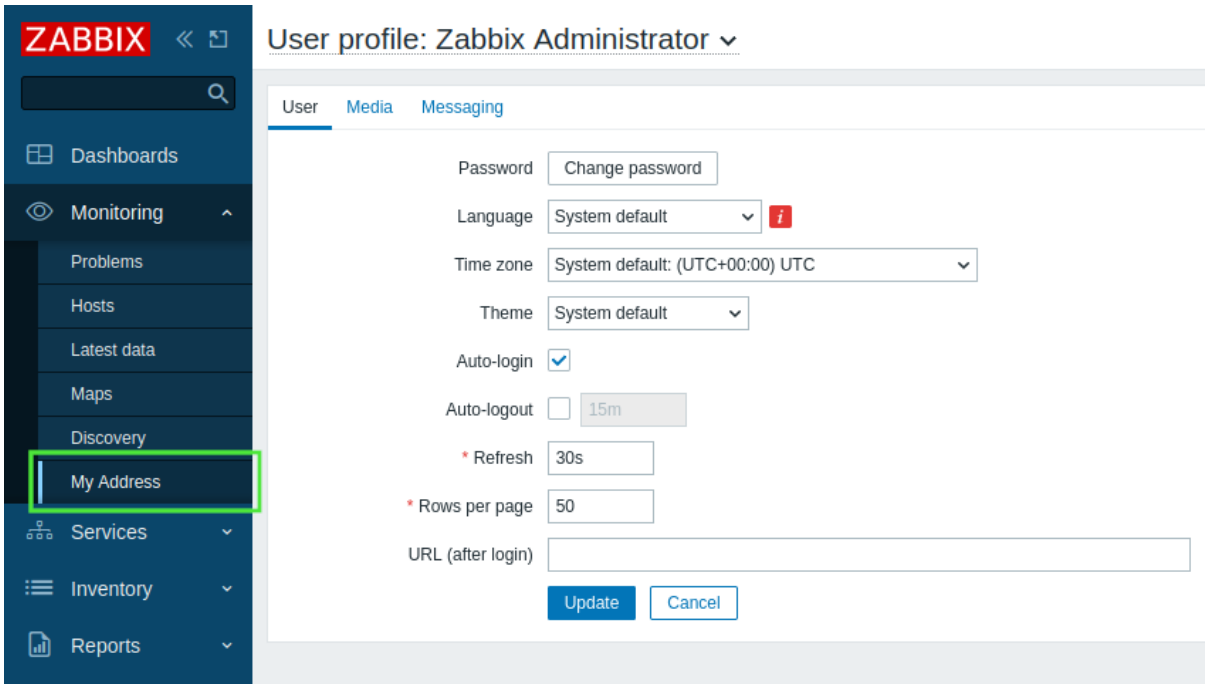
namespace Modules\MyAddress;

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 *Monitoring* e observe que a seção *My address* agora está localizada abaixo da seção *Discovery*.



3. Para adicionar páginas aninhadas à seção de menu *My Address*, abra e edite o arquivo *Module.php* novamente.

Esta etapa cria duas subseções:

- *External IP*, que executa uma nova ação 'my.address' que será definida nas próximas etapas;
- *User profile*, que executa a ação predefinida 'userprofile.edit' para abrir a página *User profile*.

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

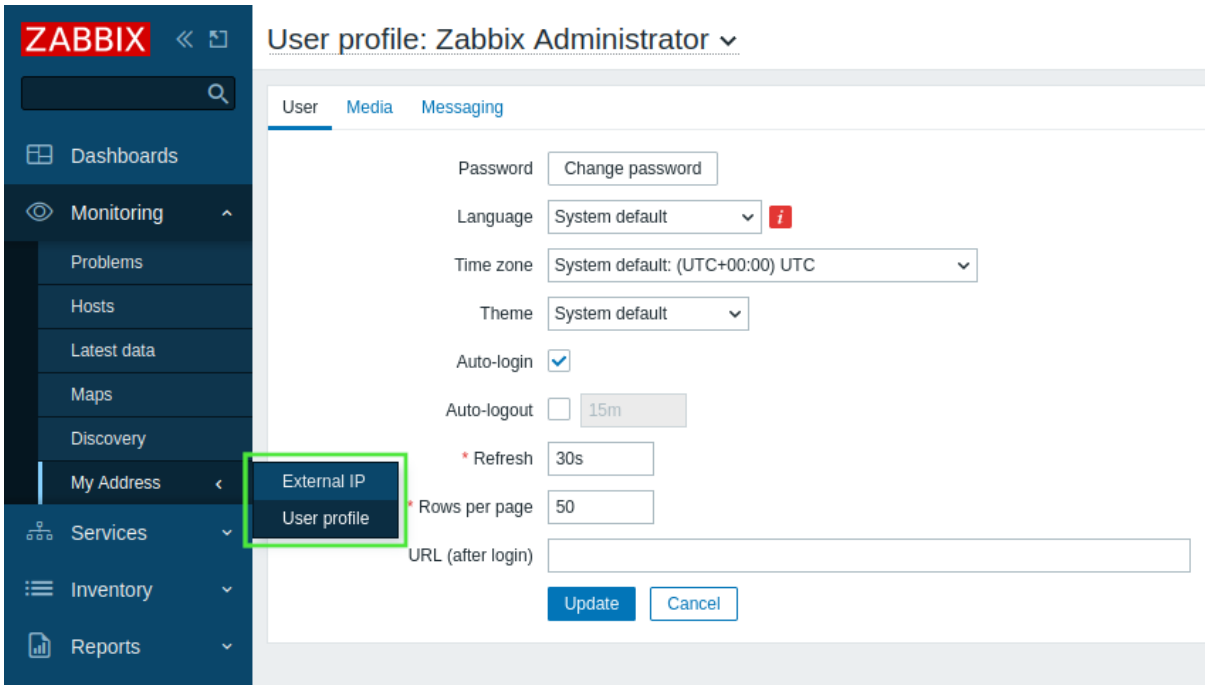
namespace Modules\MyAddress;

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 *My address* agora contém um menu de terceiro nível com duas páginas - *External IP* e *User profile*.



Parte III - Ação do módulo

Uma action 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 view.

1. Crie um diretório *actions* no diretório *MyAddress*.
2. Crie um arquivo *MyAddress.php* no diretório *actions*.

A lógica da action será definida na classe *MyAddress*. Esta classe de action implementará quatro funções: *init()*, *checkInput()*, *checkPermissions()* e *doAction()*. O frontend do Zabbix chama a função *doAction()* quando a action 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 view.

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 view do módulo.

Observe que a variável $data está disponível na view sem precisar ser definida especificamente. O framework passa automaticamente o array associativo para a view.

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();
```

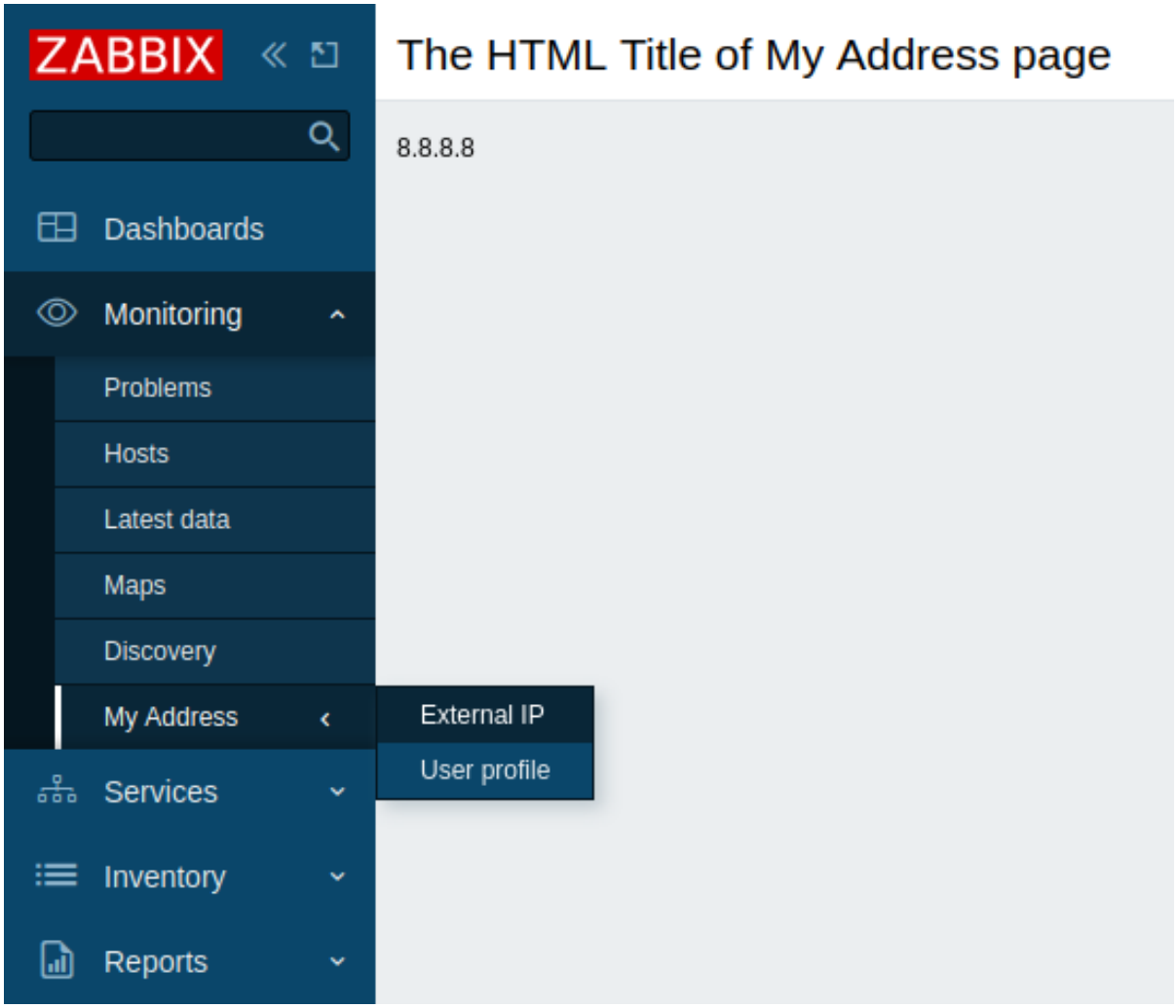
5. A action do módulo precisa ser registrada no arquivo *manifest.json*. Abra *manifest.json* e adicione um novo objeto `actions` que contenha:

- a chave da action com o nome da action escrito em minúsculas (a-z) e com as palavras separadas por pontos (por exemplo, `my.address`);
- o nome da classe da action (`MyAddress`) como um valor para a chave `class` do objeto `my.address`;
- o nome da view da action (`my.address`) como um valor para a chave `view` do objeto `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

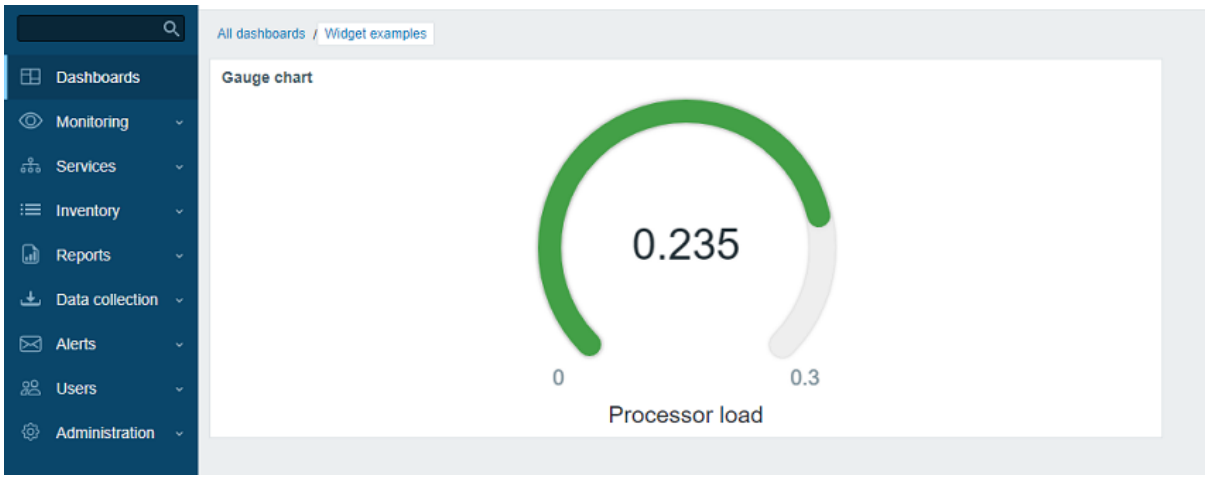
6. Atualize o frontend do Zabbix. Clique em *My address* > *External IP* para ver o endereço IP do seu computador.



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 vai criar Durante este tutorial, você primeiro criará um widget básico "Olá, mundo!" e depois o converterá em um widget mais avançado que exibe o valor de um item como um gráfico de medidor. Veja como o widget final ficará:



Parte I - "Olá, mundo!" Nesta seção, você aprenderá como criar os elementos mínimos necessários 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 sua 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 sua instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*). O diretório `zabbix/ui/widgets` é reservado para os 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 (consulte 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"
}
```

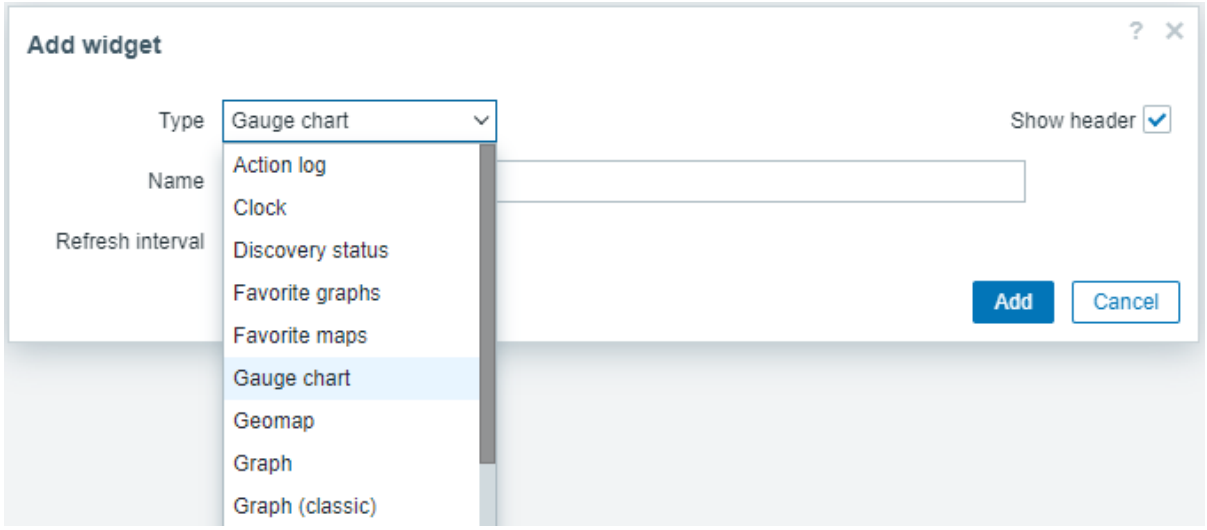
3. No frontend do Zabbix, vá para a seção *Administration* > *General* > *Modules* e clique no botão *Scan directory*.



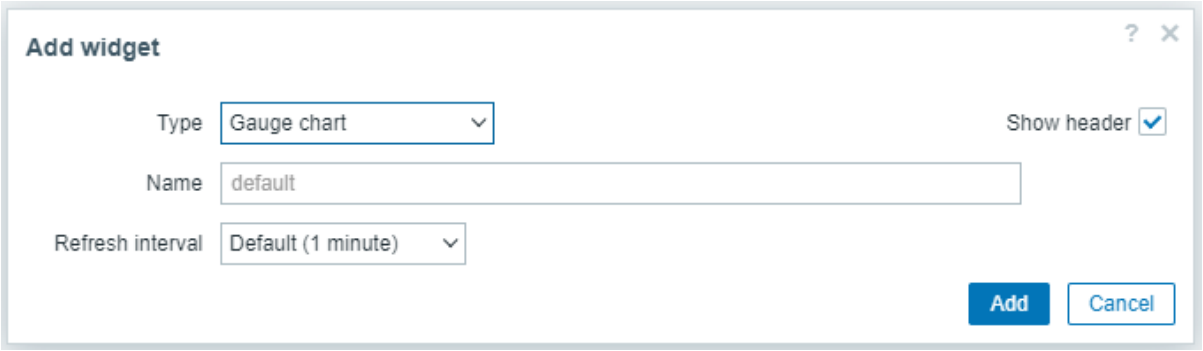
4. Localize o novo módulo *Gauge chart* na lista e clique no hiperlink "Disabled" para alterar o status do módulo de "Disabled" para "Enabled" (se o módulo não estiver listado, consulte a seção de troubleshooting).



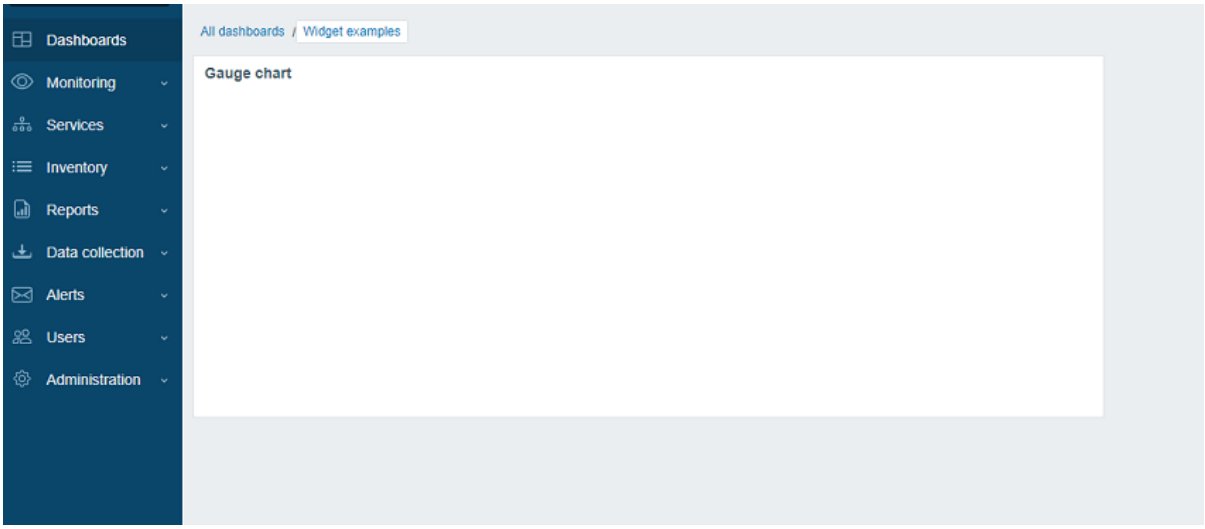
5. Abra um dashboard, coloque-o no modo de edição e adicione um novo widget. No campo *Type*, selecione "Gauge chart".



6. Neste ponto, a configuração do widget *Gauge chart* contém apenas os campos comuns do widget *Name* e *Refresh interval*. Clique em *Add* para adicionar o widget ao dashboard.



7. Um widget em branco deve aparecer no dashboard. Clique em *Save changes* no canto superior direito para salvar o dashboard.



Adicionar uma visualização de widget

> **Note:**
> O arquivo **view** do widget deve estar localizado no diretório `views` (para este tutorial, `ui/modules/lesson_gauge_chart/views/`). Se o arquivo tiver o nome padrão `widget.view.php`, você não precisa registrá-lo no arquivo `manifest.json`. Se o arquivo tiver um nome diferente, especifique-o na seção `actions/widget.lesson_gauge_chart.view` do arquivo `manifest.json`.

1. Crie um diretório `views` no diretório `lesson_gauge_chart`.
2. Crie um arquivo `widget.view.php` no diretório `views`.

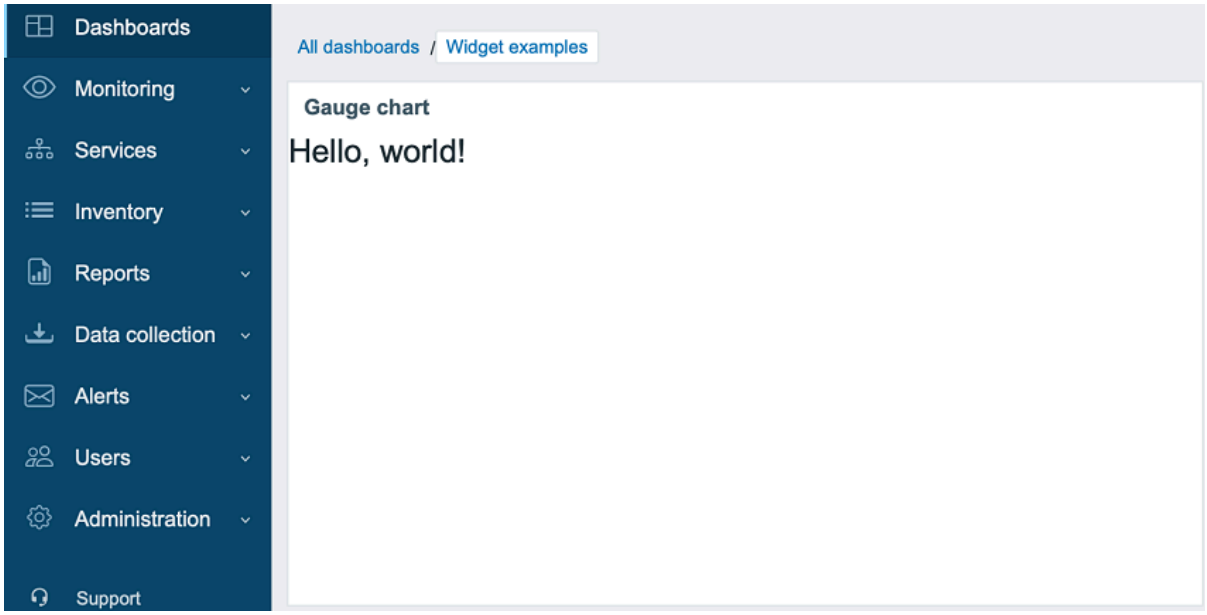
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @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 Adicionar configurações a uma view de configuração e usá-las em uma view de widget

Nesta seção, você aprenderá como adicionar um campo de configuração de widget e exibir o valor inserido na view do widget como texto.

A configuração do widget consiste em um formulário (`Zabbix\Widgets\CWidgetForm`) e uma view de formulário do widget (`widget.edit.php`). Para adicionar campos (`Zabbix\Widgets\CWidgetField`), você precisa criar uma classe `WidgetForm`, que estenderá `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 de **formulário** 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, no qual 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 view de configuração do widget `widget.edit.php` e adicione uma view para o novo campo *Description*. Para a classe de campo `CWidgetFieldTextBox`, a view é `CWidgetFieldTextBoxView`.

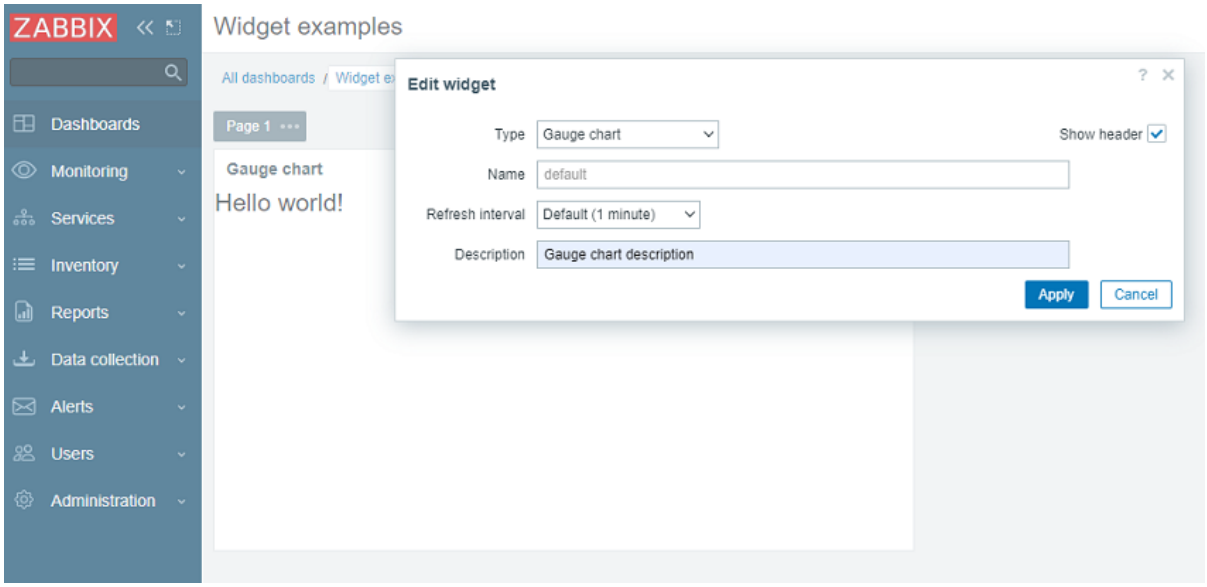
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Acesse 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*. Insira qualquer valor, por exemplo, *Gauge chart description*.



7. Clique em *Apply* no formulário de configuração do widget. Em seguida, clique em *Save changes* no canto superior direito para salvar o dashboard. Observe que a nova descrição não está visível em lugar nenhum, 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 view 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` estenderá a classe `CControllerDashboardW`

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
<?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 `$data['description']`, na view do widget. Abra `views/widget.view.php` e substitua o texto estático "Hello, world!" por `$data['description']`.

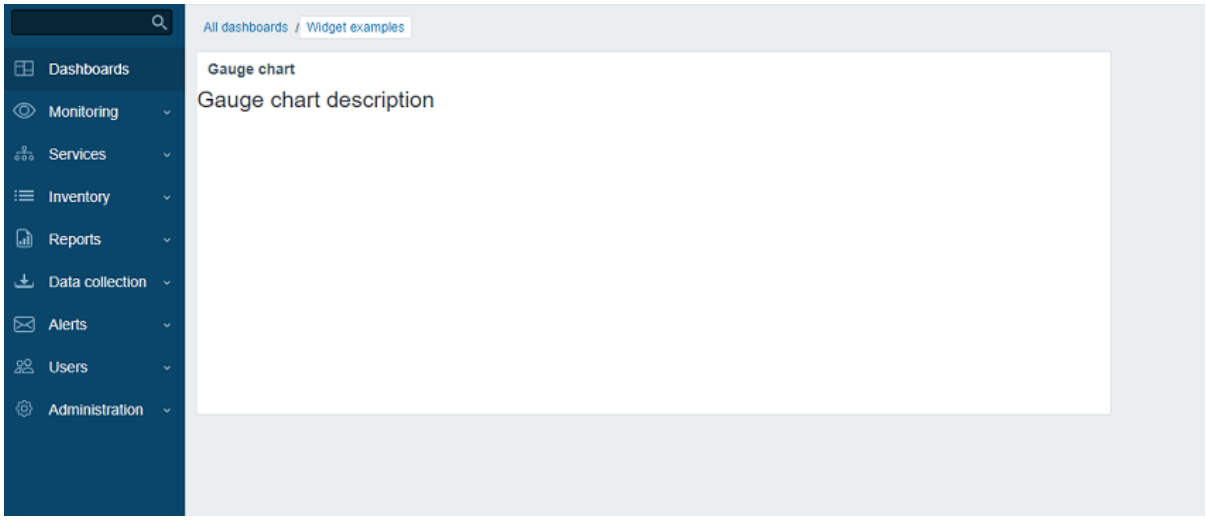
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```php
<?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 exibir o último valor de um item escolhido pelo usuário. Para isso, você precisa 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 view de configuração. Depois disso, o controller do widget poderá recuperar os dados do item e seu valor por meio de uma requisição à API. Assim que o valor for recebido, ele poderá ser exibido na view 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 view de configuração.

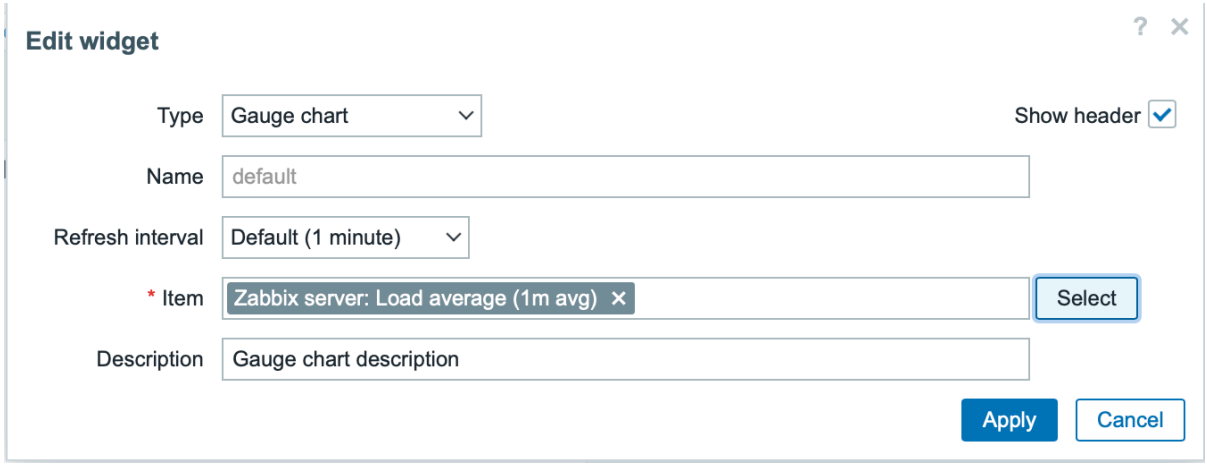
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```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 *Apply* no formulário de configuração do widget. Depois, clique em *Save changes* 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 controller do widget em `$this->fields_values['itemid']`. O método `doAction()` do controller 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 view do widget.

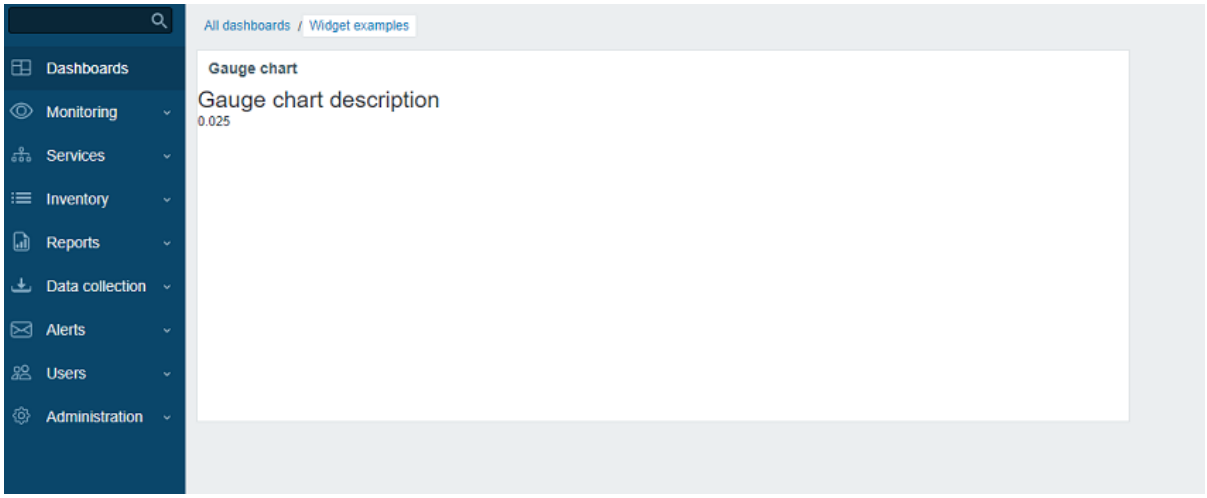
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?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 a uma view de configuração

Nesta seção, você aprenderá como adicionar uma seção expansível/recolhível *Advanced configuration* com parâmetros opcionais, como cor, valores mínimo e máximo, unidades e o campo *Description* criado anteriormente.

1. Crie um arquivo `Widget.php` no diretório principal do widget `lesson_gauge_chart` para criar uma nova classe `Widget`.

A classe `Widget` estenderá a classe base `CWidget` para adicionar/substituir as configurações padrão do widget (neste caso - traduções). O JavaScript, fornecido abaixo, exibe a string "No data" no caso de ausência de dados. A string "No data" está presente nos arquivos de tradução da interface do usuário do Zabbix.

Se houver constantes do widget, também é recomendável 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 *Color* (seletor de cor), *Min* (campo numérico), *Max* (campo numérico) e *Units* (select), e defina a paleta de cores padrão para o seletor de cor, para que ela possa ser usada nas próximas etapas.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};
```

```
use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Abra `views/widget.edit.php` e adicione os componentes visuais dos campos à view de configuração.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
```

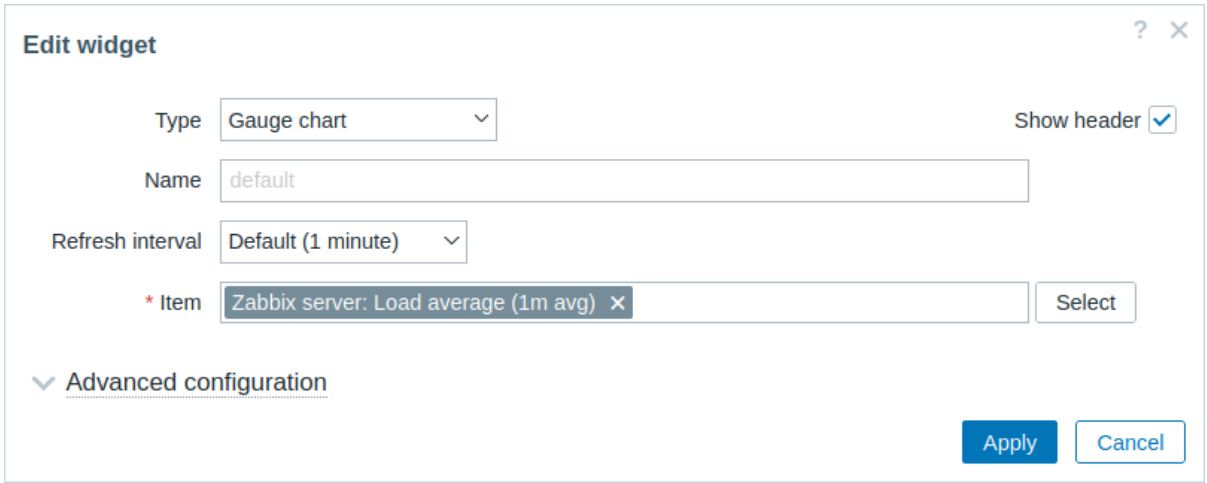
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

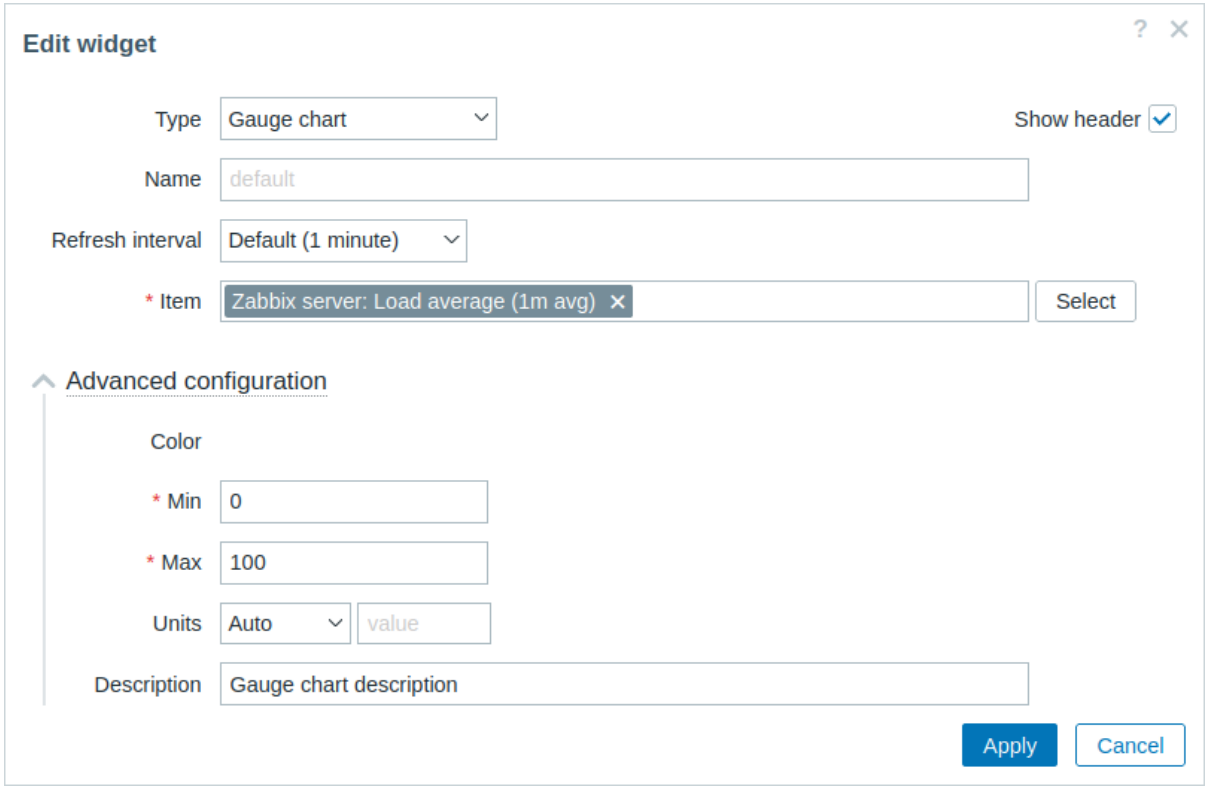
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
O método `addField()` da classe `CWidgetFormView` recebe uma string de classe CSS como segundo parâmetro.

4. Volte ao dashboard, mude 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 *Advanced configuration*.



5. Expanda a seção *Advanced configuration* para ver campos adicionais de configuração do widget. Observe que o campo *Color* ainda não tem um seletor de cor. Isso ocorre porque o seletor de cor precisa ser inicializado com JavaScript, que será adicionado na próxima seção - *Add JavaScript to the widget*.



Adicionar JavaScript ao widget

Nesta seção, você aprenderá como adicionar um gráfico de gauge - criado com JavaScript - que mostra se o último valor está normal ou se está muito alto/muito baixo.

1. Crie um arquivo `widget.edit.js.php` no diretório `views`.

O JavaScript será responsável por inicializar o seletor de cores na tela 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 tela de configuração. Para isso, use o método `includeJsFile()`. Para adicionar JavaScript inline, use o método `addJavaScript()`.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

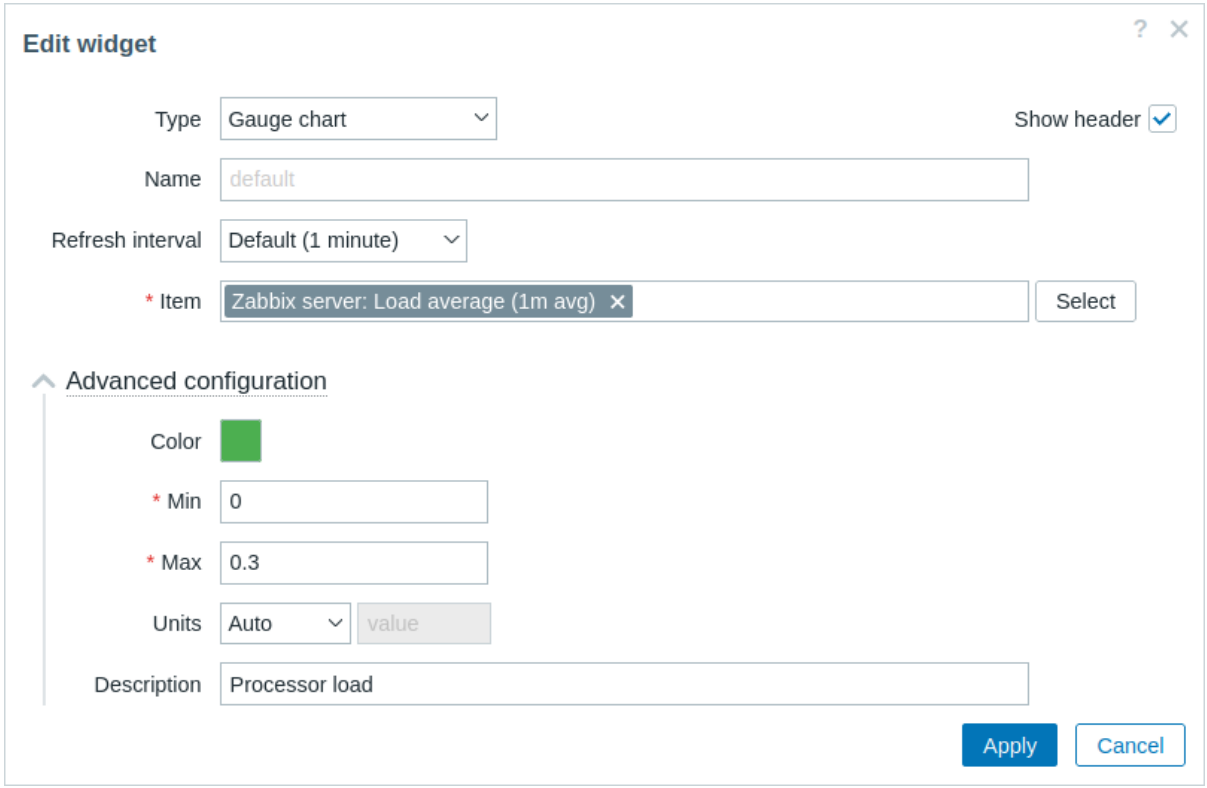
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. Volte ao dashboard, clique no ícone de engrenagem no widget para abrir o formulário de configuração do widget. Agora, expanda a seção *Advanced configuration* para ver o seletor de cores inicializado. Preencha os campos com valores e selecione uma cor para o gráfico de gauge.



4. Clique em *Apply* no formulário de configuração do widget. Depois, clique em *Save changes* no canto superior direito para salvar o dashboard.

5. Abra `actions/WidgetView.php` e atualize o controller.

A propriedade `$this->fields_values` agora contém os valores de todos os campos de *Advanced configuration*. Finalize o controller para permitir a passagem da configuração e do valor do item selecionado para a view 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 contêiner para o gráfico de gauge, que será desenhado nas próximas etapas, e um contêiner 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`. Esse diretório será usado para armazenar JavaScript, CSS e, potencialmente, outros assets, como fontes ou imagens.
8. Para o JavaScript da view 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 estenderá a classe JavaScript base de todos os widgets do dashboard - `CWidget`.

O dashboard depende de uma implementação correta de um widget e comunica qualquer informação relevante ao widget por meio da chamada dos respectivos métodos JavaScript. O dashboard também espera que o widget gere eventos quando alguma interação ocorrer. Assim, a classe `CWidget` contém um conjunto de métodos com a implementação padrão do comportamento do widget, que pode ser personalizada ao estender a classe.

Neste caso, alguma personalização é necessária; portanto, a lógica personalizada será implementada para os seguintes comportamentos do widget:

- inicialização do widget, responsável por definir o estado inicial do widget (veja o método `onInitialize()`);
- exibição do conteúdo do widget (isto é, desenhar o gráfico de gauge) se o processo de atualização do widget tiver sido 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`, consulte: JavaScript.

Como este JavaScript é necessário para a view do widget, ele deve ser carregado 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 na etapa 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 elementos do widget, use o seletor `div.dashboard-widget-{widget id}`. Para configurar o CSS para o widget inteiro, 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;
}
```

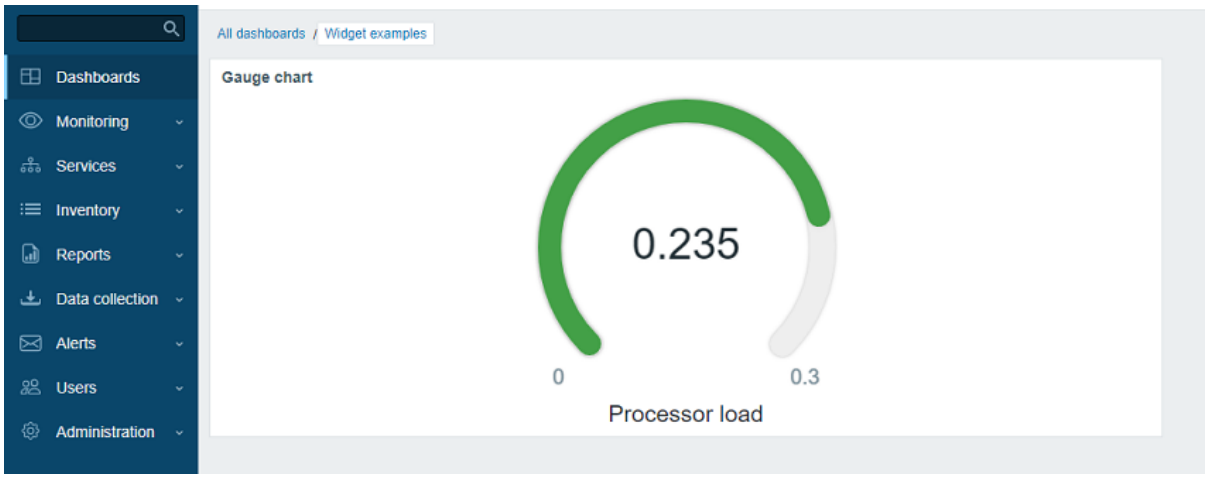
3. Abra `manifest.json` e adicione o nome do arquivo CSS (`widget.css`) ao array na seção `assets/css`. Isso permitirá que os estilos CSS definidos em `widget.css` sejam carregados junto com a página do dashboard.

`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 final 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.

Alterações no desenvolvimento de extensões

Alterações da versão 7.4 para a 8.0 ZBXNEXT-10445 O comprimento máximo dos campos `string` de widgets foi aumentado de 2048 para 65535 caracteres. Campos `string` específicos em widgets nativos do Zabbix agora também têm suas próprias restrições de comprimento (por exemplo, definidas por meio da função `setMaxLength()`).

Zabbix manpages

Documentação de referência para ferramentas e serviços de linha de comando do Zabbix, listando comandos disponíveis, opções e exemplos de uso.

zabbix_agent2

```
ZABBIX_AGENT2(8)            Manual do Administrador de Sistema           ZABBIX_AGENT2(8)

NOME
       zabbix_agent2 - Zabbix agent 2

SINOPSE
       zabbix_agent2 [-c config-file]
       zabbix_agent2 [-c config-file] [-v] -p
       zabbix_agent2 [-c config-file] [-v] -t item-key
       zabbix_agent2 [-c config-file] -R runtime-option
       zabbix_agent2 [-c config-file] -T
       zabbix_agent2 -h
       zabbix_agent2 -V

DESCRIÇÃO
       zabbix_agent2  é  um  aplicativo para monitorar parâmetros de vários
       serviços.

OPÇÕES
       -c, --config config-file
              Usa o config-file alternativo em vez do padrão.

       -R, --runtime-control runtime-option
              Executa funções administrativas de acordo com runtime-option.

           Opções de controle em tempo de execução
               userparameter_reload
                   Recarrega os parâmetros de usuário do arquivo de configuração

               log_level_increase
                   Aumenta o nível de log

               log_level_decrease
                   Diminui o nível de log

               help
                   Lista as opções de controle em tempo de execução disponíveis

               metrics
                   Lista as métricas disponíveis
               version
                   Exibe a versão

       -p, --print
              Exibe os items conhecidos e sai. Para cada item, são usados os
              padrões genéricos ou são fornecidos padrões específicos para
              teste. Esses padrões são listados entre colchetes como parâme-
              tros 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 valores de
              retorno possíveis. Os items exibidos como funcionais não têm
              garantia de funcionar a partir do Zabbix server ou do
              zabbix_get ao consultar um daemon agent em execução, pois as
              permissões ou o ambiente podem ser diferentes. Os tipos de valor
              retornado são:

           d   Número com parte decimal.

           m   Não suportado. Isso pode ser causado pela consulta de um item
               que só funciona no modo ativo, como um item de monitoramento de
               log, ou um item que requer vários valores coletados. Problemas
               de permissão ou parâmetros de usuário incorretos também podem
               resultar no estado de 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 a descrição da
              saída.

       -v, --verbose
              Habilita saída detalhada ao testar item ou exibir items
              conhecidos.

       -T, --test-config
              Valida o arquivo de configuração e sai.

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações de versão e sai.

ARQUIVOS
       /usr/local/etc/zabbix_agent2.conf
              Localização padrão do arquivo de configuração do Zabbix agent 2
              (se não tiver sido modificada durante o tempo de compilação).

VEJA TAMBÉM
       Documentação

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENT2(8)
```

zabbix_agentd

```
ZABBIX_AGENTD(8)            Manual do Administrador de Sistema           ZABBIX_AGENTD(8)

NOME
       zabbix_agentd - daemon do Zabbix agent

SINOPSE
       zabbix_agentd [-c config-file]
```

```
zabbix_agentd [-c config-file] -p
zabbix_agentd [-c config-file] -t item-key
zabbix_agentd [-c config-file] -R runtime-option
zabbix_agentd [-c config-file] -T
zabbix_agentd -h
zabbix_agentd -V

DESCRIÇÃO
zabbix_agentd é um daemon para monitorar vários parâmetros do server.

OPÇÕES
-c, --config config-file
       Usa o config-file alternativo em vez do padrão.

-f, --foreground
       Executa o Zabbix agent em primeiro plano.

-R, --runtime-control runtime-option
       Executa funções administrativas de acordo com runtime-option.

    Opções de controle em tempo de execução
        userparameter_reload
            Recarrega os parâmetros de usuário do arquivo de configuração

        log_level_increase[=target]
            Aumenta o nível de log; afeta todos os processos se target
            não for especificado

        log_level_decrease[=target]
            Diminui o nível de log; afeta todos os processos se target
            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 e número do processo (por exemplo, listener,3)
        pid Identificador do processo, até 65535. Para valores maiores,
            especifique target como "process-type,N"

-p, --print
       Imprime os items conhecidos e sai. Para cada item, são usados os
       padrões genéricos ou fornecidos padrões específicos para teste.
       Esses padrões são listados entre colchetes como parâmetros da
       chave do item. Os valores retornados são colocados entre colchetes
       e prefixados com o tipo do valor retornado, separado por um
       caractere pipe. Para parâmetros de usuário, o tipo é sempre t,
       pois o agent não consegue determinar todos os valores de retorno
       possíveis. Os items exibidos como funcionais não têm garantia de
       funcionar a partir do Zabbix server ou do zabbix_get ao consultar
       um daemon do 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 pela consulta de 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 de 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 a descrição da saída.

       -T, --test-config
              Valida o arquivo de configuração e sai.

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações de versão e sai.

ARQUIVOS
       /usr/local/etc/zabbix_agentd.conf
              Localização padrão do arquivo de configuração do Zabbix agent (se
              não modificada durante o tempo de compilação).

VEJA TAMBÉM
       Documentation

       zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                          2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)              Manual Geral de Comandos              ZABBIX_GET(1)

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-sub-
       ject cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k item-key
       zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t
       timeout] --tls-connect psk --tls-psk-identity PSK-identity
       --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] -k item-key
       zabbix_get -h
       zabbix_get -V

DESCRIÇÃO
       zabbix_get é um utilitário de linha de comando para obter dados
       do agent Zabbix.

OPÇÕES
       -s, --host host-name-or-IP
              Especifica o nome do host ou o endereço IP de um host.
```

-p, --port port-number
       Especifica o número da porta do agent em execução no host. O
       padrão é 10050.

-I, --source-address IP-address
       Especifica o endereço IP de origem.

-t, --timeout seconds
       Especifica o tempo limite. Intervalo válido: 1-600 segundos (pa-
       drão: 30)

-k, --key item-key
       Especifica a chave do item para o qual o valor será recuperado.

-P, --protocol value
       Protocolo usado para se comunicar com o agent. Valores:

            auto   conecta usando o protocolo JSON, com retorno e
                        nova tentativa usando o protocolo de texto puro
                        (padrão)

            json   conecta usando o protocolo JSON

            plaintext   conecta usando o protocolo de texto puro,
                        onde apenas a chave do item é enviada (versões
                        6.4.x e anteriores)

--tls-connect value
       Como se conectar ao agent. Valores:

            unencrypted
                        conecta sem criptografia (padrão)

            psk   conecta usando TLS e uma chave pré-compartilhada

            cert   conecta usando TLS e um certificado

--tls-ca-file CA-file
       Caminho completo de um arquivo contendo os certificados da(s) CA
       de nível superior para verificação do certificado do par.

--tls-crl-file CRL-file
       Caminho completo de um arquivo contendo certificados revogados.

--tls-agent-cert-issuer cert-issuer
       Emissor de certificado do agent permitido.

--tls-agent-cert-subject cert-subject
       Assunto do certificado do agent permitido.

--tls-cert-file cert-file
       Caminho completo de um arquivo contendo o certificado ou a
       cadeia de certificados.

--tls-key-file key-file
       Caminho completo de um arquivo contendo a chave privada.

--tls-psk-identity PSK-identity
       String de identidade PSK.

--tls-psk-file PSK-file
       Caminho completo de um arquivo contendo a chave pré-compartilhada.

```
       --tls-cipher13 cipher-string
              String de cifras para OpenSSL 1.1.1 ou mais recente para TLS 1.3.
              Substitui os critérios padrão de seleção do conjunto de cifras.
              Esta opção não está disponível se a versão do OpenSSL for infe-
              rior a 1.1.1.

       --tls-cipher cipher-string
              String de prioridade do GnuTLS (para TLS 1.2 e superior) ou
              string de cifras do OpenSSL (somente para TLS 1.2). Substitui os
              critérios padrão de seleção do conjunto de cifras.

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações de versão e sai.
STATUS DE SAÍDA
       O status de saída é 0 se o zabbix_get recuperar com sucesso um valor do
       agent Zabbix. Se a recuperação do valor falhar, o status de saída será
       1.

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  op-
       erations,O=Example  Corp,DC=example,DC=com"  --tls-cert-file /home/zab-
       bix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
       zabbix_get  -s  127.0.0.1  -p  10050   -k   "system.cpu.load[all,avg1]"
       --tls-connect   psk   --tls-psk-identity   "PSK   ID   Zabbix   agentd"
       --tls-psk-file /home/zabbix/zabbix_agentd.psk

VEJA TAMBÉM
       Documentação

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                     ZABBIX_GET(1)
```

zabbix_js

```
ZABBIX_JS(1)               Manual Geral de Comandos               ZABBIX_JS(1)

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
       testar 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
              stdin.

       -p, --param input-param
              Especifica o parâmetro de entrada passado ao script. O parâmetro
              é passado literalmente como uma string e fica disponível no
              script por meio da variável 'value'. Mutuamente exclusivo com
              -i.

       -i, --input input-file
              Especifica um arquivo contendo o parâmetro de entrada passado ao
              script. O conteúdo do arquivo é passado literalmente como uma
              string e fica disponível no script por meio da variável 'value'.
              Se '-' for especificado como nome do arquivo, a entrada será
              lida da stdin. Mutuamente exclusivo com -p.

       -w, --webdriver url
              Especifica a URL do webdriver.

       -l, --loglevel log-level
              Especifica o nível de log (intervalo: 0-5).

       -t, --timeout timeout
              Especifica o timeout em segundos. Intervalo válido: 1-600
              segundos (padrão: 10)

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações de versão e sai.

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

VEJA TAMBÉM
       Documentação

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                       ZABBIX_JS(1)
```

zabbix_proxy

```
ZABBIX_PROXY(8)             Manual do Gerenciador do Sistema            ZABBIX_PROXY(8)

NOME
       zabbix_proxy - daemon do Zabbix proxy

SINOPSE
       zabbix_proxy [-c arquivo-de-configuração]
       zabbix_proxy [-c arquivo-de-configuração] -R opção-de-tempo-de-execução
       zabbix_proxy [-c arquivo-de-configuração] -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 arquivo-de-configuração
Usa o arquivo-de-configuração alternativo em vez do padrão.

-f, --foreground
Executa o Zabbix proxy em primeiro plano.

-R, --runtime-control opção-de-tempo-de-execução
Executa funções administrativas de acordo com a opção-de-tempo-de-execução.

Opções de controle em tempo de execução

config_cache_reload
Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O Zabbix proxy ativo se conectará ao Zabbix server e solicitará os dados de configuração. O arquivo de configuração padrão (a menos que a opção -c seja especificada) será usado para localizar o arquivo PID e o sinal será enviado ao processo listado no arquivo PID.

snmp_cache_reload
Recarrega o cache SNMP.

housekeeper_execute
Executa o housekeeper. Ignorado se o housekeeper estiver sendo executado no momento.

diaginfo[=section]
Registra informações internas de diagnóstico da seção especificada. A seção pode ser historycache, preprocessing, locks. Por padrão, as informações de diagnóstico de todas as seções são registradas.

prof_enable[=target]
Ativa o profiling, afeta todos os processos se target não for especificado

prof_disable[=target]
Desativa o profiling, afeta todos os processos se target não for especificado

log_level_increase[=target]
Aumenta o nível de log, afeta todos os processos se target não for especificado.

log_level_decrease[=target]
Diminui o nível de log, afeta todos os processos se target não for especificado.

Destinos 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, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, preprocessing worker, self-monitoring, snmp trapper, supervisor, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Tipo e número do processo (por exemplo, poller,3)

pid Identificador do processo

Destinos de controle de profiling

```
                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, internal poller, ipmi manager, ipmi poller, java
                    poller, odbc poller, poller, agent poller, http agent poller, snmp poller,
                    preprocessing manager, self-monitoring, snmp trapper, task manager, trapper,
                    unreachable poller, vmware collector)

                process-type,N
                    Tipo e número do processo (por exemplo, history syncer,1)

                pid Identificador do processo

                scope
                    (rwlock, mutex, processing) pode ser usado com process-type
                    (por exemplo, history syncer,1,processing)

        -T, --test-config
               Valida o arquivo de configuração e encerra.

        -h, --help
               Exibe esta ajuda e encerra.

        -V, --version
               Exibe informações da versão e encerra.

ARQUIVOS
        /usr/local/etc/zabbix_proxy.conf
               Local padrão do arquivo de configuração do Zabbix proxy (se não for
               modificado durante a compilação).

VEJA TAMBÉM
        Documentação

        zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8),
        zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                    ZABBIX_PROXY(8)
```

zabbix_sender

```
ZABBIX_SENDER(1)            Manual de comandos gerais           ZABBIX_SENDER(1)

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] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
       [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
       cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
       value
```

```
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
[-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender -h
zabbix_sender -V
```

DESCRIÇÃO

zabbix_sender é um utilitário de linha de comando para enviar dados de monitoramento ao Zabbix server ou proxy. No Zabbix server, deve ser criado um item do tipo Zabbix trapper com a chave correspondente. Observe que os valores recebidos somente serão aceitos de hosts especificados no campo Allowed hosts desse item.

OPÇÕES

-c, --config config-file

Usa config-file. O Zabbix sender lê os detalhes do server no arquivo de configuração do agentd. Por padrão, o Zabbix sender não lê nenhum arquivo de configuração. Somente os parâmetros Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile, TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject, TLSCertFile, TLSKeyFile, TLSPSKIdentity e TLSPSKFile são compatíveis. O Hostname definido por meio do parâmetro HostnameItem não será obtido; nesse caso, o nome do host deve ser especificado pela linha de comando (consulte 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 serão 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, deverá ser usado o nome do host ou o

endereço IP do proxy. Quando usado junto com --config, substitui as entradas do parâmetro ServerActive especificadas no arquivo de configuração do agentd.

-p, --port port
Especifica o número da porta em que o Zabbix server trapper está em execução no server. O padrão é 10051. Quando usado junto com --config, substitui as entradas de porta do parâmetro ServerActive especificadas no arquivo de configuração do agentd.

-I, --source-address IP-address
Especifica o endereço IP de origem. Quando usado junto com --config, substitui o parâmetro SourceIP especificado no arquivo de configuração do agentd.

-t, --timeout seconds
Especifica o tempo limite. 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 e o nome DNS do host não funcionarão. Quando usado junto com --config, substitui o parâmetro Hostname especificado no arquivo de configuração do agentd.

-k, --key key
Especifica a chave do item para o qual o valor será enviado.

-o, --value value
Especifica o valor do item.

-i, --input-file input-file
Carrega os valores do arquivo de entrada. Especifique - como <input-file> para ler os valores da entrada padrão. Cada linha do arquivo contém, separados por espaços em branco: <hostname> <key> <value>. Cada valor deve ser especificado em sua própria linha. Cada linha deve conter 3 entradas separadas por espaços em branco: <hostname> <key> <value>, em que "hostname" é o nome do host monitorado conforme registrado no frontend do Zabbix, "key" é a chave do item de destino e "value" é o valor a ser enviado. Especifique - como <hostname> para usar o nome do host 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 estar definido corretamente na configuração do item no frontend do Zabbix. O Zabbix sender enviará até 250 valores em uma conexão. 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 de acordo com as seguintes regras:

- Entradas entre aspas e sem aspas são compatíveis.
- As aspas duplas são o caractere de delimitação.
- Entradas com espaços em branco devem estar entre aspas.
- Os caracteres de aspas duplas e barra invertida dentro de uma entrada entre aspas devem ser escapados com uma barra

invertida.

- O escape não é compatível com entradas sem aspas.
- Sequências de escape de avanço de linha (\n) são compatíveis com strings entre aspas.
- As sequências de escape de avanço de linha são removidas do final de uma entrada.

-T, --with-timestamps
Esta opção somente pode ser usada com a opção --input-file.

Cada linha do arquivo de entrada deve conter 4 entradas separadas por espaços em branco: <hostname> <key> <timestamp> <value>. O timestamp deve ser especificado no formato de timestamp Unix. Se o item de destino tiver triggers que façam referência a ele, todos os timestamps deverão estar em ordem crescente; caso contrário, o cálculo do evento não estará correto.

Um exemplo de linha de um arquivo de entrada:

"Linux DB3" db.connections 1429533600 43

Para obter mais detalhes, consulte a opção --input-file.

Se um valor com timestamp for enviado para um host que esteja em um período de manutenção do tipo “no data”, 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 somente pode ser usada com a opção --with-timestamps.

Cada linha do arquivo de entrada deve conter 5 entradas separadas por espaços em branco: <hostname> <key> <timestamp> <ns> <value>.

Um exemplo de linha de um arquivo de entrada:

"Linux DB3" db.connections 1429533600 7402561 43

Para obter mais detalhes, consulte a opção --input-file.

-r, --real-time
Envia os valores um por um assim que são recebidos. Isso pode ser usado ao ler a entrada padrão.

--tls-connect value
Como se conectar ao server ou proxy. Valores:

unencrypted
conecta sem criptografia (padrão)

psk
conecta usando TLS e uma chave pré-compartilhada

cert
conecta usando TLS e um certificado

--tls-ca-file CA-file
Caminho completo de um arquivo que contém os certificados da CA de nível superior para verificação do certificado do peer.

--tls-crl-file CRL-file
Caminho completo de um arquivo que contém certificados revogados.

```
       --tls-server-cert-issuer cert-issuer
              Emissor de certificado do server permitido.

       --tls-server-cert-subject cert-subject
              Assunto do certificado do server permitido.

       --tls-cert-file cert-file
              Caminho completo de um arquivo que contém o certificado ou a
              cadeia de certificados.

       --tls-key-file key-file
              Caminho completo de um arquivo que contém a chave privada.

       --tls-psk-identity PSK-identity
              String de identidade PSK.

       --tls-psk-file PSK-file
              Caminho completo de um arquivo que contém a chave
              pré-compartilhada.

       --tls-cipher13 cipher-string
              String de cifras para OpenSSL 1.1.1 ou mais recente para TLS 1.3.
              Substitui os critérios padrão de seleção do conjunto de cifras.
              Esta opção não está disponível se a versão do OpenSSL for
              inferior a 1.1.1.

       --tls-cipher cipher-string
              String de prioridade do GnuTLS (para TLS 1.2 e superior) ou
              string de cifras do OpenSSL (somente para TLS 1.2). Substitui os
              critérios padrão de seleção do conjunto de cifras.

       -v, --verbose
              Modo detalhado; use -vv para obter mais detalhes.

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações da versão e sai.

STATUS DE SAÍDA
       O status de saída será 0 se os valores forem enviados e todos forem
       processados com sucesso pelo server. Se os dados forem enviados, mas o
       processamento de pelo menos um dos valores falhar, o status de saída
       será 2. Se o envio dos dados falhar, o status de saída será 1.

EXEMPLOS
       zabbix_sender -c  /etc/zabbix/zabbix_agentd.conf  -k  mysql.queries  -o
       342.45
              Envia 342.45 como o valor do 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 o valor do 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 chaves 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 hífen
              nos dados de entrada indica que o nome do host 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 ao 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 o valor do 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 o valor do item mysql.queries no host Monitored
              Host para o server com IP 192.168.1.113, usando TLS com uma chave
              pré-compartilhada (PSK).

CONSULTE TAMBÉM
       Documentação

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SENDER(1)
```

zabbix_server

```
ZABBIX_SERVER(8)            Manual do Gerenciador de Sistema           ZABBIX_SERVER(8)

NOME
       zabbix_server - daemon do server Zabbix

SINOPSE
       zabbix_server [-c arquivo-de-configuração]
       zabbix_server [-c arquivo-de-configuração] -R opção-de-tempo-de-execução
       zabbix_server [-c arquivo-de-configuração] -T
       zabbix_server -h
       zabbix_server -V

DESCRIÇÃO
       zabbix_server é o daemon central do software Zabbix.

OPÇÕES
       -c, --config arquivo-de-configuração
              Usa o arquivo-de-configuração alternativo em vez do padrão.

       -f, --foreground
              Executa o server Zabbix em primeiro plano.
```

```
-R, --runtime-control opção-de-tempo-de-execução
       Executa funções administrativas de acordo com a opção-de-tempo-de-execução.

    Opções de controle em tempo de execução
        config_cache_reload
            Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momen

        snmp_cache_reload
            Recarrega o cache SNMP.

        housekeeper_execute
            Executa o housekeeper. Ignorado se o housekeeper estiver sendo executado no momento.

        trigger_housekeeper_execute
            Executa o housekeeper de trigger (remove problemas de triggers excluídos). Ignorado se

        secrets_reload
            Recarrega segredos do Vault. Ignorado se o cache de configuração ou os segredos estiver

        diaginfo[=section]
            Registra informações internas de diagnóstico da seção especificada. A seção pode ser hi

        prof_enable[=target]
            Ativa o profiling, afeta todos os processos se target não for especificado

        prof_disable[=target]
            Desativa o profiling, afeta todos os processos se target não for especificado

        ha_status
            Exibe o status do cluster de alta disponibilidade. Pode ser executado somente no nó ati

        ha_remove_node
            ha_remove_node[=name|ID] Remove um nó de alta disponibilidade pelo nome ou ID listado e

        ha_set_failover_delay[=delay]
            Define o atraso de failover de alta disponibilidade. O atraso pode ser especificado usa

        log_level_increase[=target]
            Aumenta o nível de log, afeta todos os processos se target não for especificado

        log_level_decrease[=target]
            Diminui o nível de log, afeta todos os processos se target não for especificado

    Destinos de controle do nível de log
        process-type
            Todos os processos do tipo especificado (alerter, alert manager, alert syncer, availabi

        process-type,N
            Tipo de processo e número (por exemplo, poller,3)

        pid Identificador do processo

    Destinos de controle de profiling
        process-type
            Todos os processos do tipo especificado (alerter, alert manager, alert syncer, availabi

        process-type,N
            Tipo de processo e número (por exemplo, history syncer,1)

        pid Identificador do processo

        scope
```

```
                    (rwlock, mutex, processing) pode ser usado com process-type (por exemplo, history synce

       -T, --test-config
              Valida o arquivo de configuração e encerra.

       -h, --help
              Exibe esta ajuda e encerra.

       -V, --version
              Exibe informações da versão e encerra.

ARQUIVOS
       /usr/local/etc/zabbix_server.conf
              Local padrão do arquivo de configuração do server Zabbix (se não tiver sido modificado duran

VEJA TAMBÉM
       Documentação

       zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       Manual do Administrador de Sistemas      ZABBIX_WEB_SERVICE(8)

NOME
       zabbix_web_service - Serviço web do Zabbix

SINOPSE
       zabbix_web_service [-c config-file]
       zabbix_web_service [-c config-file] -T
       zabbix_web_service -h
       zabbix_web_service -V

DESCRIÇÃO
       zabbix_web_service é um aplicativo para fornecer serviços web aos com-
       ponentes do Zabbix.

OPÇÕES
       -c, --config config-file
              Usa o config-file alternativo em vez do padrão.

       -T, --test-config
              Valida o arquivo de configuração e sai.

       -h, --help
              Exibe esta ajuda e sai.

       -V, --version
              Exibe informações de versão e sai.

ARQUIVOS
       /usr/local/etc/zabbix_web_service.conf
              Localização padrão do arquivo de configuração do serviço web do Zabbix (se
              não modificada durante o tempo de compilação).

VEJA TAMBÉM
       Documentation
```

```
       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06             ZABBIX_WEB_SERVICE(8)
```

Aviso de direitos autorais

A documentação do Zabbix NÃO é distribuída sob a licença AGPL-3.0. O uso da documentação do Zabbix está sujeito aos seguintes termos:

Você pode criar uma cópia impressa desta documentação exclusivamente para seu uso pessoal. A conversão para outros formatos é permitida, desde que o conteúdo real não seja alterado ou editado de forma alguma. Você não deve publicar ou distribuir esta documentação em qualquer forma ou mídia, exceto se você distribuir a documentação de maneira semelhante à forma como o Zabbix a divulga (ou seja, eletronicamente para download em um site do Zabbix) ou em um USB ou mídia similar, desde que a documentação seja divulgada 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 não expressamente concedidos acima.