Documentation 8.0

ZABBIX

29.12.2025

Contents

Zabbix マニュアル **6**
著作権表示 . . . 6
1 インストールと最初のステップ . . . 6
Zabbix 8.0 へのアップグレードに関する注意事項 . . . 7
1 Web インターフェースのインストール . . . 7
2 クイックスタートガイド . . . 13
3 インストール: 追加ガイド . . . 32
4 アップグレード . . . 70
既知の問題 . . . 85
要件 . . . 93
2 Zabbix 8.0 の新機能 . . . 102
テンプレート . . . 102
フロントエンド . . . 102
ウィジェット . . . 103
ドキュメント . . . 104
プロセス . . . 104
アイテム . . . 104
プラグイン . . . 105
ローレベルディスカバリ . . . 105
Zabbix 8.0.x の新機能 . . . 105
3 Zabbix アプライアンス . . . 105
4 Zabbix プロセス . . . 109
1 Zabbix サーバー . . . 109
2 Zabbix エージェント . . . 117
3 Zabbix エージェント 2 . . . 121
4 Zabbix プロキシ . . . 124
5 Zabbix Java ゲートウェイ . . . 128
6 zabbix_sender コマンド . . . 132
7 zabbix_get コマンド . . . 133
8 zabbix_js コマンド . . . 134
9 Zabbix Web サービス . . . 134
5 設定 . . . 135
1 ホストとホストグループ . . . 144
2 アイテム . . . 158
3 トリガー . . . 354
4 イベント . . . 372
5 イベント相関 . . . 375
6 タグ付け . . . 381
7 視覚化 . . . 384
8 テンプレートとテンプレートグループ . . . 408
9 すぐに使えるテンプレート . . . 408
10 イベント時の通知 . . . 421
11 マクロ . . . 466
12 ユーザーとユーザーグループ . . . 480
13 secrets の保存 . . . 488
14 スケジュールレポート . . . 495
15 データのエクスポート . . . 498
6 サービス監視 . . . 504
1 サービストリー . . . 505
2 SLA . . . 508

3 セットアップ例 . . . 510
7 Web 監視 . . . 514
1 Web 監視アイテム . . . 522
2 実際のシナリオ . . . 523
8 仮想マシンの監視 . . . 529
1 VMware 監視アイテムキー . . . 531
2 仮想マシンディスカバリキーフィールド . . . 549
3 VMware アイテムの JSON の例 . . . 552
4 VMware 監視設定例 . . . 557
9 メンテナンス . . . 560
10 正規表現 . . . 564
11 問題の認識 . . . 569
1 障害の抑制 . . . 571
12 設定のエクスポート/インポート . . . 572
1 テンプレートグループ . . . 573
2 ホストグループ . . . 574
3 テンプレート . . . 574
4 ホスト . . . 589
5 ネットワークマップ . . . 605
6 メディアタイプ . . . 613
13 ディスカバリ . . . 620
1 ネットワークディスカバリ . . . 620
2 アクティブエージェントの自動登録 . . . 628
3 ローレベルディスカバリ . . . 631
14 分散監視 . . . 682
1 プロキシ . . . 683
15 暗号化 . . . 690
1 証明書の利用 . . . 697
2 事前共有鍵の使用 . . . 704
3 トラブルシューティング . . . 706
16 Web インターフェース . . . 709
1 メニュー . . . 709
2 Web インターフェースセクション . . . 716
3 ユーザー設定 . . . 948
4 グローバル検索 . . . 951
5 フロントエンドメンテナンスモード . . . 953
6 ページパラメーター . . . 954
7 定義 . . . 957
8 独自テーマ作成 . . . 957
9 デバッグモード . . . 958
10 Zabbix 用 Cookie . . . 959
11 タイムゾーン . . . 960
12 パスワードのリセット . . . 961
13 時間帯とホストセレクター . . . 961
17 ベストプラクティス . . . 962
1 セキュリティのベストプラクティス . . . 962
2 設定のベストプラクティス . . . 970
18 API . . . 971
メソッドリファレンス . . . 976
付録 1. 参考文献の解説 . . . 1799
付録 2. 7.4 から 8.0 への変更点 . . . 1805
付録 3. 8.0 の変更点 . . . 1805
19 拡張機能 . . . 1805
1 ローダブルモジュール . . . 1807
2 プラグイン . . . 1816
3 フロントエンドモジュール . . . 1824
20 付録 . . . 1825
1 インストールとセットアップ . . . 1825
2 プロセスの設定 . . . 1879
3 プロトコル . . . 1960
4 アイテム . . . 1984
5 サポートされている関数 . . . 2019
6 マクロ . . . 2050

7 単位記号 . . . 2088
8 日時フォーマット . . . 2090
9 コマンドの実行 . . . 2090
10 バージョン間の互換性 . . . 2091
12 Windows 用 Zabbix sender ダイナミックリンクライブラリ . . . 2092
13 Zabbix API の Python ライブラリ . . . 2093
14 サービス監視のアップグレード . . . 2093
15 その他の問題 . . . 2094
16 エージェントとエージェント 2 の比較 . . . 2095
17 エスケープの例 . . . 2096
21 クイックリファレンスガイド . . . 2097
概要 . . . 2097
1 Zabbix エージェントで Linux を監視する . . . 2097
2 Zabbix エージェントで Windows を監視する . . . 2101
3 HTTP 経由で Apache を監視する . . . 2105
4 Zabbix エージェント 2 で MySQL を監視する . . . 2109
5 Zabbix で VMware を監視する . . . 2115
6 Zabbix でネットワークトラフィックを監視する . . . 2119
7 アクティブチェックを使用してネットワークトラフィックを監視する . . . 2122
8 ブラウザアイテムでウェブサイトの監視 . . . 2125
9 Zabbix エージェント 2(パッシブ) で Web サイトの証明書を監視する . . . 2129
10 Zabbix でネットワークスイッチやルーターを監視する . . . 2131
11 アクティブチェックを使用して Windows イベントログを監視する . . . 2138
Zabbix とは . . . 2141

Zabbix Cloud **2145**
著作権表示 . . . 2146
1 Zabbix Cloud で Zabbix をデプロイする . . . 2146
はじめに . . . 2146
組織管理 . . . 2146
ノードの作成 . . . 2147
Zabbix フロントエンドへのアクセス . . . 2149
2 ノードの設定 . . . 2151
概要 . . . 2151
概要タブ . . . 2152
アクセスフィルタータブ . . . 2153
暗号化タブ . . . 2154
バックアップタブ . . . 2155
履歴タブ . . . 2155
メンテナンスタブ . . . 2156
アップグレードタブ . . . 2157
3 ユーザーの追加 . . . 2159
はじめに . . . 2159
組織へのユーザー追加 . . . 2159
ノードへのユーザー追加 . . . 2160
4 Zabbix Cloud とオンプレミスの主な違い . . . 2161
はじめに . . . 2161
機能の違い . . . 2161

デベロッパーセンター **2162**
著作権表示 . . . 2162
モジュール . . . 2163
モジュールファイルの構造 . . . 2163
ウィジェット . . . 2171
チュートリアル . . . 2179
例 . . . 2205
プラグイン . . . 2205
例 . . . 2208
プラグインの作成 (チュートリアル) . . . 2208
プラグインインターフェース . . . 2213
拡張機能開発の変更点 . . . 2213

Zabbix リファレンスマニュアル **2213**
zabbix_agent2 . . . 2213

名前 . . . 2213
概要 . . . 2214
説明 . . . 2214
オプション . . . 2214
ファイル . . . 2215
参照 . . . 2215
目次 . . . 2215
zabbix_agentd . . . 2215
名前 . . . 2215
概要 . . . 2215
説明 . . . 2216
オプション . . . 2216
ファイル . . . 2217
参照 . . . 2217
目次 . . . 2217
zabbix_get . . . 2217
名前 . . . 2217
概要 . . . 2217
説明 . . . 2217
オプション . . . 2218
例 . . . 2219
参照 . . . 2219
目次 . . . 2219
zabbix_js . . . 2219
名前 . . . 2219
概要 . . . 2219
説明 . . . 2219
オプション . . . 2220
例 . . . 2220
参照 . . . 2220
目次 . . . 2220
zabbix_proxy . . . 2220
名前 . . . 2221
概要 . . . 2221
説明 . . . 2221
オプション . . . 2221
ファイル . . . 2222
参照 . . . 2222
目次 . . . 2222
zabbix_sender . . . 2222
名前 . . . 2223
概要 . . . 2223
説明 . . . 2223
オプション . . . 2223
終了ステータス . . . 2225
例 . . . 2225
参照 . . . 2226
目次 . . . 2226
zabbix_server . . . 2226
名前 . . . 2226
概要 . . . 2226
説明 . . . 2226
オプション . . . 2227
ファイル . . . 2228
参照 . . . 2228
目次 . . . 2228
zabbix_web_service . . . 2229
名前 . . . 2229
概要 . . . 2229
説明 . . . 2229
オプション . . . 2229
ファイル . . . 2229
参照 . . . 2229

目次 2229

Zabbix マニュアル

Zabbix のユーザーマニュアルにようこそ。このマニュアルは、ユーザーが Zabbix を使用して、単純なことからより複雑なことまで、監視タスクを正常に管理できるようにするために作成されています。

著作権表示

Zabbix のドキュメントは、AGPL-3.0 ライセンスの下で配布されていません。Zabbix ドキュメントの使用には、次の条件が適用されます。

個人的な利用の場合、このドキュメントの印刷物を作成することができます。実際の内容が変更または編集されない限り、他の形式への変換は許可されます。Zabbix が配布する方法と同様の方法（つまり、Zabbix Web サイトから電子的にダウンロードする）または USB などの媒体でドキュメントを配布する場合を除き、いかなる形式、メディアでも、このドキュメントを公開、配布してはなりません。ただし、ドキュメントがソフトウェアと一緒に同じ媒体で配布されることを条件とします。印刷物の配布や、このドキュメントの全体または一部を別の出版物で使用するなどのその他の利用には、Zabbix の正式な代表者からの事前の書面による同意が必要です。Zabbix は、上記で明示的に付与されていない、このドキュメントに対するすべての権利を留保します。

1 インストールと最初のステップ

Zabbix を最初からセットアップするには、次の手順に従います。

1) 必要なソフトウェアがインストールされ、実行されていることを確認する
2) Zabbix をインストールする
3) Zabbix の Web インターフェースをインストールする
4) 監視を開始する

各ステップの詳細は以下の通りです。

必要なソフトウェア　Zabbix は、以下のソフトウェア（LAMP スタックとも呼ばれます）で正常に動作します。

- **Linux** オペレーティングシステム**:** Ubuntu、Debian、CentOS、またはその他
- **Web** サーバー**:** Apache または NGINX
- データベースサーバー**:** MySQL/MariaDB または PostgreSQL
- **PHP**

ほとんどの最新の Linux システムでは、PHP はすでにインストールされている場合があります。その場合は、データベースサーバーと Web サーバーのセットアップのみが必要です。

サポートされているソフトウェアのバージョンなど、より詳細な情報については、必要なソフトウェアを参照してください。

Zabbix のインストール　最も簡単な Zabbix のインストール方法は、ディストリビューションパッケージからインストールすることです。パッケージは、必要なソフトウェアの依存関係もインストールします。

お使いのオペレーティングシステムに合わせた正確なインストール手順は、**Zabbix** 公式サイトで入手できます。

プラットフォームを選択し、表示される手順に従ってください。最初は、Zabbix サーバー、Zabbix エージェント 2、およびフロントエンド（Web インターフェース）のインストールを推奨します。後から Zabbix プロキシや追加のエージェントを追加することもできます。

インストールプロセスの一部として、(設定や履歴データを保存するための）データベースの作成と、初期データのインポートが含まれます。

Zabbix Web インターフェースのインストール　Web インターフェースのインストールウィザードの手順を完了してください。

フロントエンドがデータベースと通信できるように、(前の手順で作成した）データベースのパスワードが求められます。

監視の開始　Zabbix Web インターフェースにログインします。

クイックスタートガイドを使用して、一般的なリソース（Linux、Apache、MySQL など）の監視を開始します。これらのクイックガイドでは、一般的なリソースの監視を開始するために使用するテンプレートについて説明しています。

Linux サーバーの監視を始めるには、クイックスタートガイドから始めるとよいでしょう。

Zabbix 8.0 へのアップグレードに関する注意事項

これらの注意事項は、Zabbix 7.4.x から Zabbix 8.0.0 へのアップグレードに関するものです。

すべての注意事項は以下のグループに分類されています:

- 重大な変更 - 既存のインストールに影響を与える可能性のある変更や、アップグレードプロセスに関連する重要な情報
- その他 - Zabbix の機能変更に関するその他の情報

また、以下も参照してください:

- Zabbix 8.0.0 以前のバージョンからのアップグレードに関するすべての情報については、アップグレード手順を参照してください。
- 高可用性 (HA) クラスター内のサーバーのアップグレード手順については、HA クラスターのアップグレードを参照してください。

互換性のない変更点　データベースバージョン

必要なデータベースバージョンの最小値が引き上げられました:

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

プラグイン

Zabbix agent 2 用の Ceph プラグインは、ロード可能なプラグインとなり、追加のインストール手順が必要になりました。詳細は Ceph プラグインの readme を参照してください。

その他

Zabbix 8.0.x のアップグレードノート

このページでは、Zabbix のメジャーバージョンのマイナーリリースに関するアップグレードノートをまとめています。

また、メジャーバージョンのアップグレードノートも参照してください。

8.0.1 のアップグレードに関する注意事項　このバージョンはまだリリースされていません。

テンプレートの変更

このページでは、Zabbix に同梱されている標準テンプレートのすべての変更点を一覧表示しています。

最新の Zabbix バージョンにアップグレードしても、使用中のテンプレートは自動的にはアップグレードされません。既存のインストールでテンプレートを変更するには、Zabbix Git リポジトリから最新のテンプレートをダウンロードし、Zabbix に手動でインポートすることを推奨します。

同じ名前のテンプレートがすでに存在する場合は、インポート時に *Delete missing* オプションをチェックしてクリーンなインポートを実現してください。この方法では、更新されたテンプレートに含まれなくなった古いアイテムが削除されます（ただし、これらの古いアイテムの履歴が失われることに注意してください）。

> **Note:**
> Zabbix 6.0 以降、すべてのテンプレートが更新されたフォーマットに従っているため、6.0 より前のテンプレートのインポートに影響する場合があります。詳細については、6.0 のテンプレートの変更を参照してください。

8.0.0 の変更点

1 Web インターフェースのインストール

このセクションでは、Zabbix Web インターフェースをインストールするための手順を説明します。
Zabbix バックエンド（Zabbix サーバー、エージェント、データベース）をインストールした後に、これらの手順に従ってください。

Zabbix フロントエンドは PHP で記述されているため、実行するには PHP 対応の Web サーバーが必要です。

Note:
Zabbix フロントエンドの SSL 設定の詳細については、ベストプラクティスを参照してください。

ようこそ画面

ブラウザで Zabbix フロントエンドの URL を開きます。パッケージから Zabbix をインストールした場合、URL は以下の通りです。

- Apache の場合: *http://<server_ip_or_name>/zabbix*
- Nginx の場合: *http://<server_ip_or_name>*

フロントエンドインストールウィザードの最初の画面が表示されます。

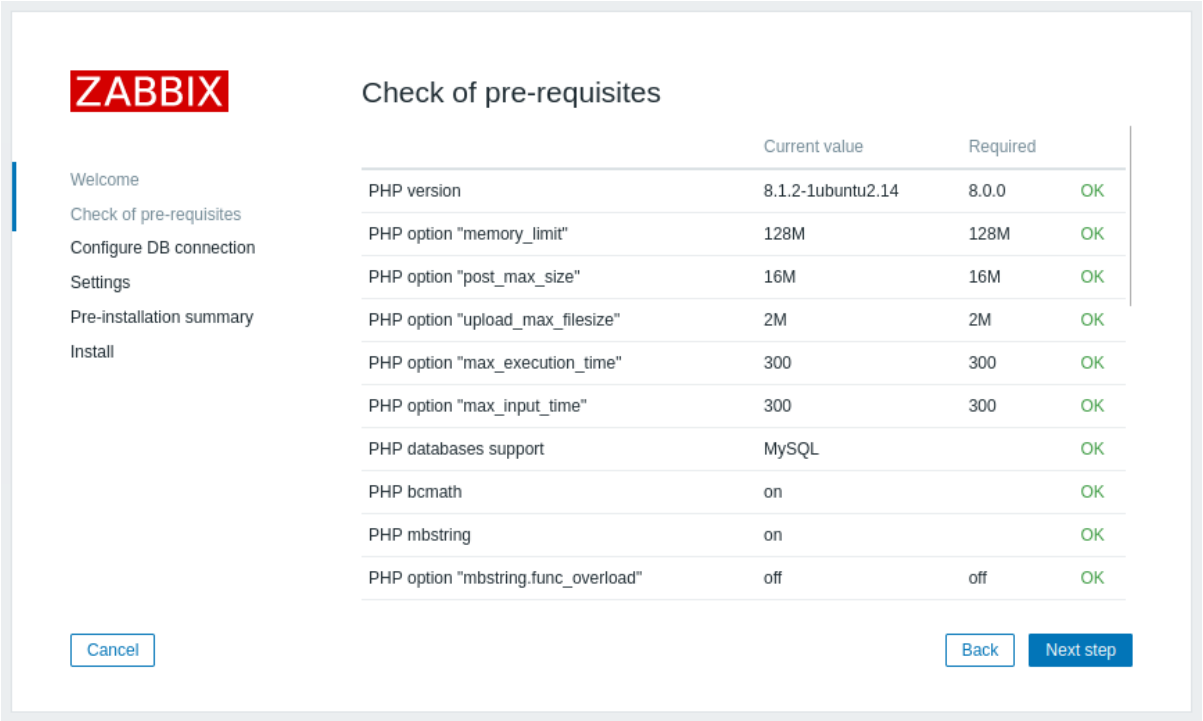
Default language ドロップダウンメニューを使用して、システムのデフォルト言語を変更し、選択した言語でインストールプロセスを続行できます（オプション）。詳細は、追加フロントエンド言語のインストールを参照してください。

言語を *English (en_US)* に設定すると、フロントエンドで米国の時刻/日付フォーマットも有効になることに注意してください。



必要条件のチェック

Zabbix フロントエンドの必須要件がすべて満たされていることを確認してください。

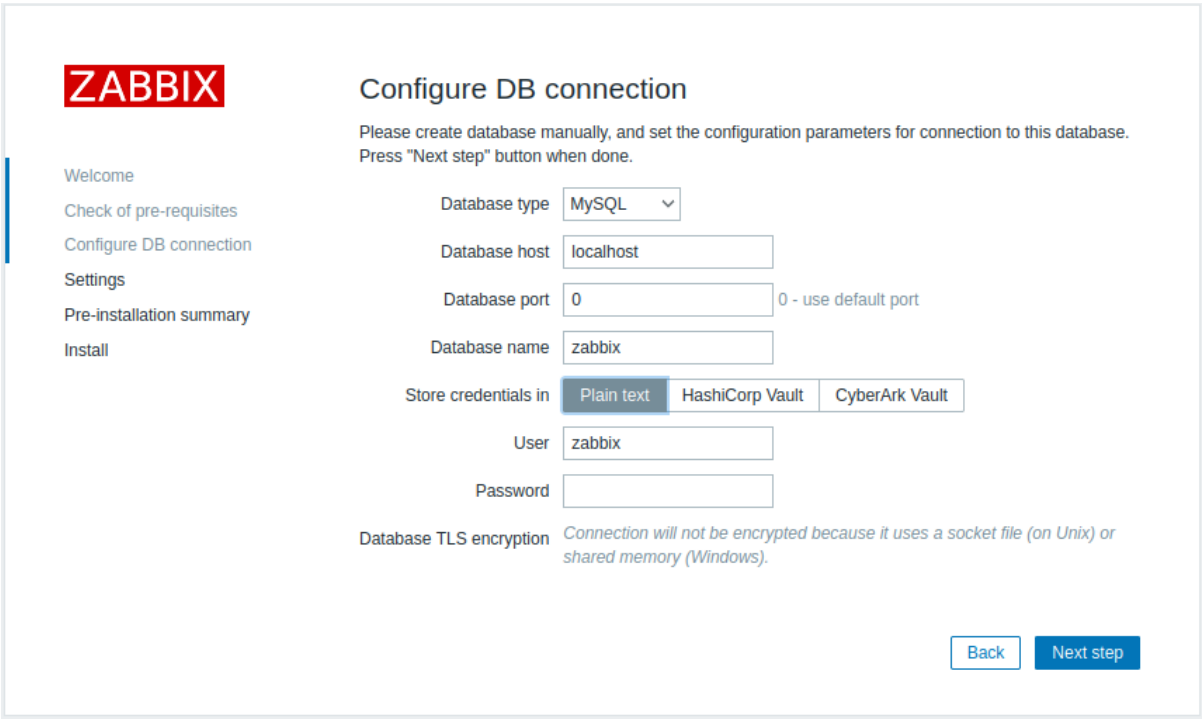


必要条件	最小値	説明
PHP バージョン	8.0.0	
PHP オプション*"memory_limit"*	128MB	php.ini: memory_limit = 128M
PHP オプション*"post_max_size"*	16MB	php.ini: post_max_size = 16M
PHP オプション*"upload_max_filesize"*	2MB	php.ini: upload_max_filesize = 2M
PHP オプション*"max_execution_time"*	300 秒	php.ini: max_execution_time = 300 (値 0 および-1 も許可されます)
PHP オプション*"max_input_time"*	300 秒	php.ini: max_input_time = 300 (値 0 および-1 も許可されます)
PHP データベースサポート	MySQL、PostgreSQL のいずれか	すべての必須およびオプションの PHP 拡張機能のリストについては、要件を参照してください。失敗したオプションの必要条件は赤い *Warning* ステータスで表示され、満たされていなくてもセットアッププロセスを続行できます。
PHP bcmath	有効である必要があります	
PHP mbstring	有効である必要があります	
PHP オプション*"mbstring.func_overload"*	無効である必要があります	php.ini: mbstring.func_overload = 0
PHP オプション*"session.auto_start"*	無効である必要があります	php.ini: session.auto_start = 0
PHP オプション*"arg_separator.output"*	&	php.ini: arg_separator.output = "&" (値"&" も許可されます)

Attention:
Apache ユーザーまたはユーザーグループを変更する必要がある場合は、セッションフォルダーへの権限を確認してください。そうしないと、Zabbix セットアップを続行できない場合があります。

DB 接続の設定

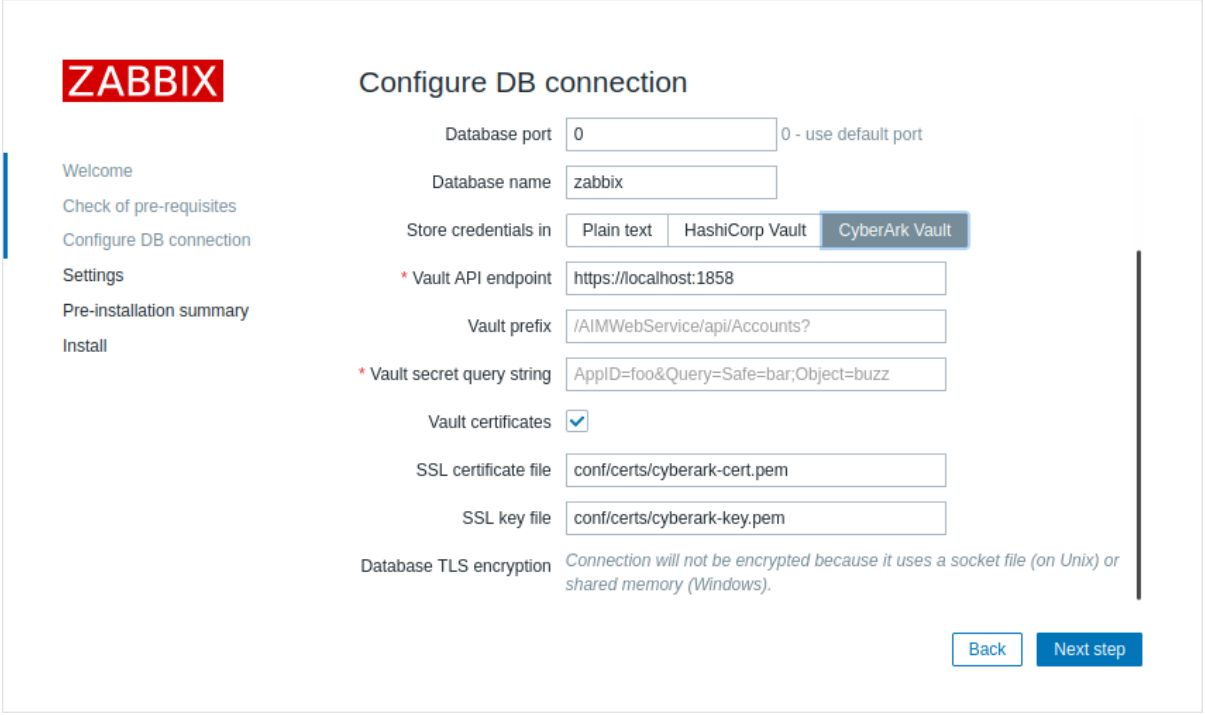
データベースへの接続に必要な情報を入力します。Zabbix データベースはすでに作成されている必要があります。



Database TLS encryption オプションをチェックすると、データベースへのTLS 接続の設定のための追加フィールドがフォームに表示されます（MySQL または PostgreSQL のみ）。

Store credentials in が HashiCorp Vault または CyberArk Vault に設定されている場合、追加のパラメータが利用可能になります:

- HashiCorp Vaultの場合: Vault API エンドポイント、vault プレフィックス、シークレットパス、認証トークン
- CyberArk Vaultの場合: Vault API エンドポイント、vault プレフィックス、シークレットクエリ文字列、証明書。*Vault certificates* チェックボックスをマークすると、SSL 証明書ファイルと SSL キーファイルのパスを指定するための 2 つの新しいフィールドが表示されます。

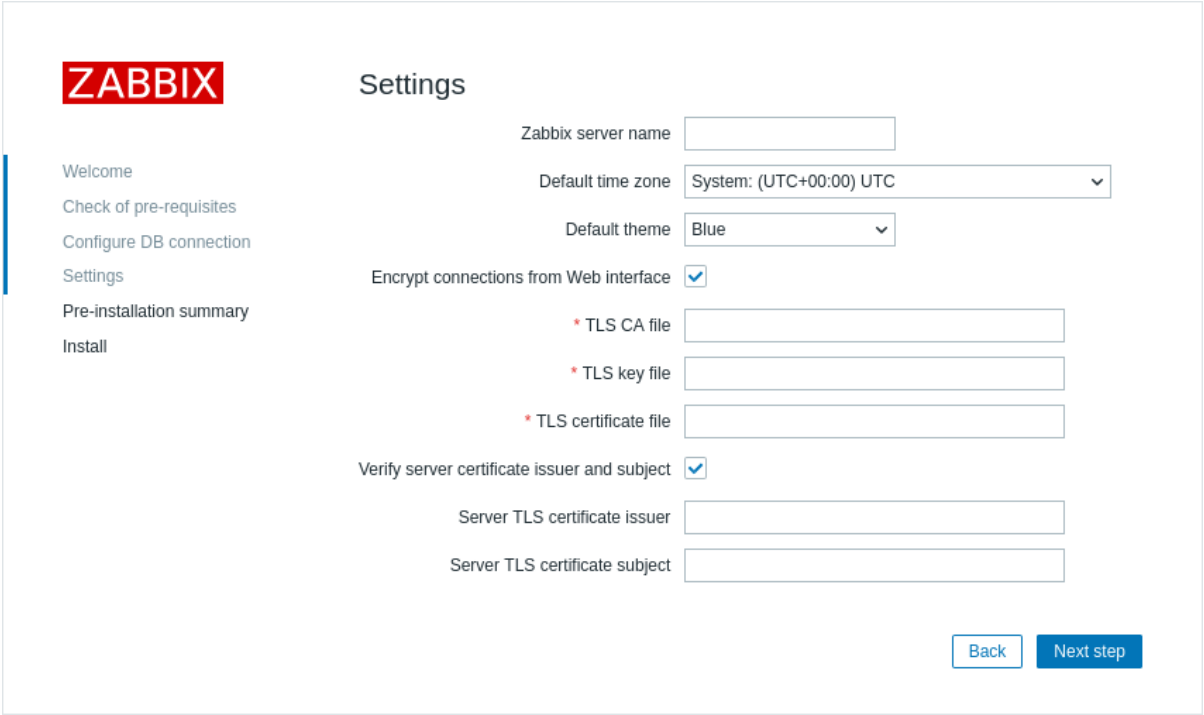


設定

Zabbix サーバーの名前の入力は任意ですが、入力した場合はメニューバーやページタイトルに表示されます。

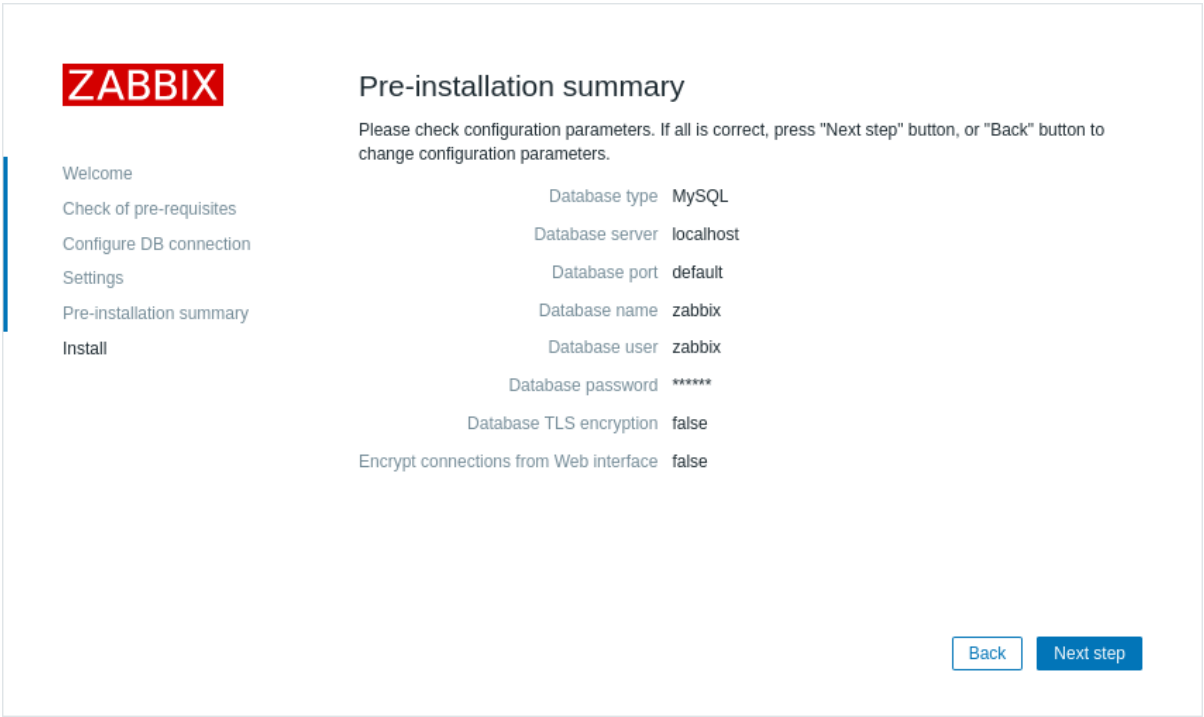
フロントエンドのデフォルトのタイムゾーンとテーマを設定します。

Web インターフェースからの接続を暗号化オプションをチェックすると、Zabbix サーバーとフロントエンド間のTLS 接続の設定用の追加フィールドがフォームに表示されます。

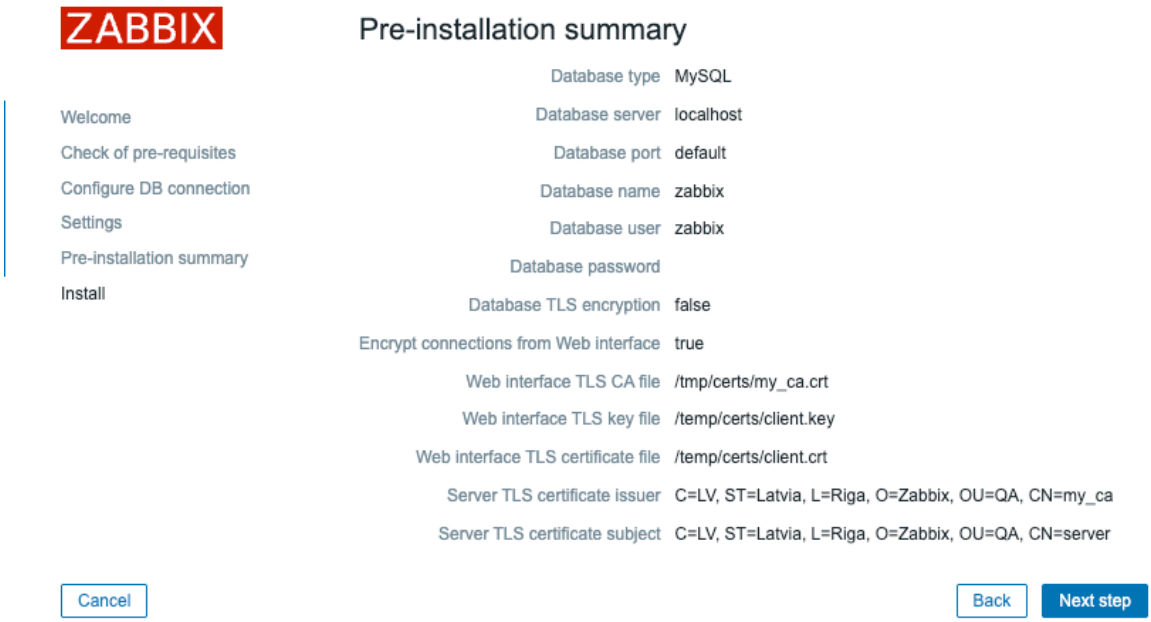


インストール前の概要

設定の概要を確認します。



TLS 構成が追加されている場合は、サブページにデータが表示されます。



インストール

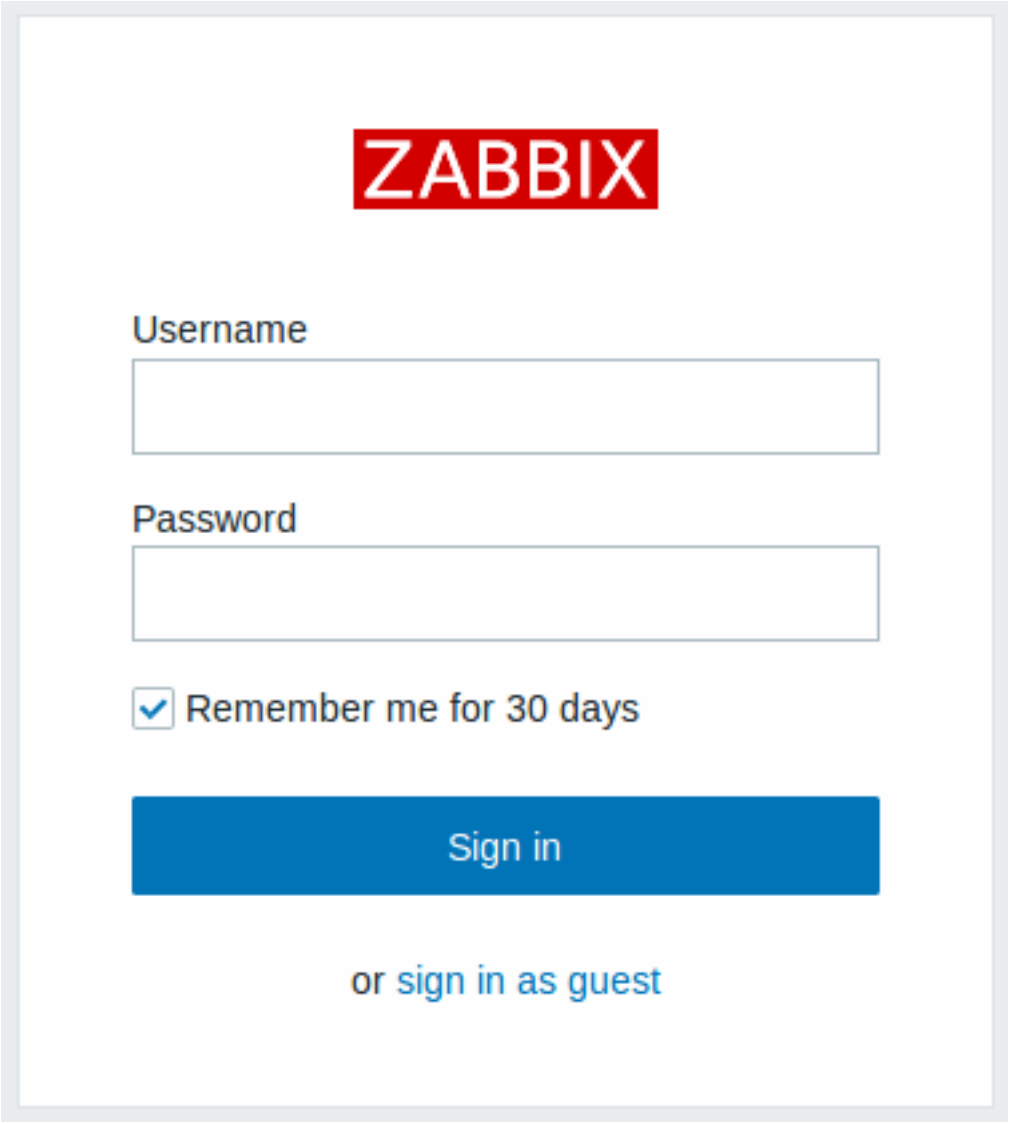
Zabbix をソースからインストールする場合は、設定ファイルをダウンロードし、Zabbix の PHP ファイルをコピーした Web サーバーの HTML ドキュメントサブディレクトリの conf/配下に配置します。

Note:
Web サーバーユーザーに conf/ディレクトリへの書き込み権限があれば、設定ファイルは自動的に保存され、すぐに次のステップに進むことができます。

インストールを完了します。

ログイン

Zabbix フロントエンドの準備ができました！デフォルトのユーザー名は **Admin**、パスワードは **zabbix** です。



Linux 監視ガイドに進みます。

2 クイックスタートガイド

ZabbixとそのWeb インターフェースのインストールが完了したら、監視を開始しましょう。

これらのクイックガイドを使用して、一般的なリソースの監視を開始できます:

- Linux サーバーの監視
- Apache Web サーバーの監視
- MySQL サーバーの監視

クイックスタートガイドでは、監視に公式の Zabbix テンプレートを使用することに重点を置いています。テンプレートとは、あらかじめ設定された監視要素のセットです。

テンプレートはすぐに監視を始めるには便利ですが、その要素をカスタマイズしたい場合もあるでしょう。

テンプレートを構成する基本要素についてより理解するには、基本設定を参照してください。

基本設定

これらのガイドでは、Zabbix の基本的な設定要素について説明します:

- ログインとユーザーの設定
- 新しいホスト
- 新しいアイテム
- 新しいトリガー

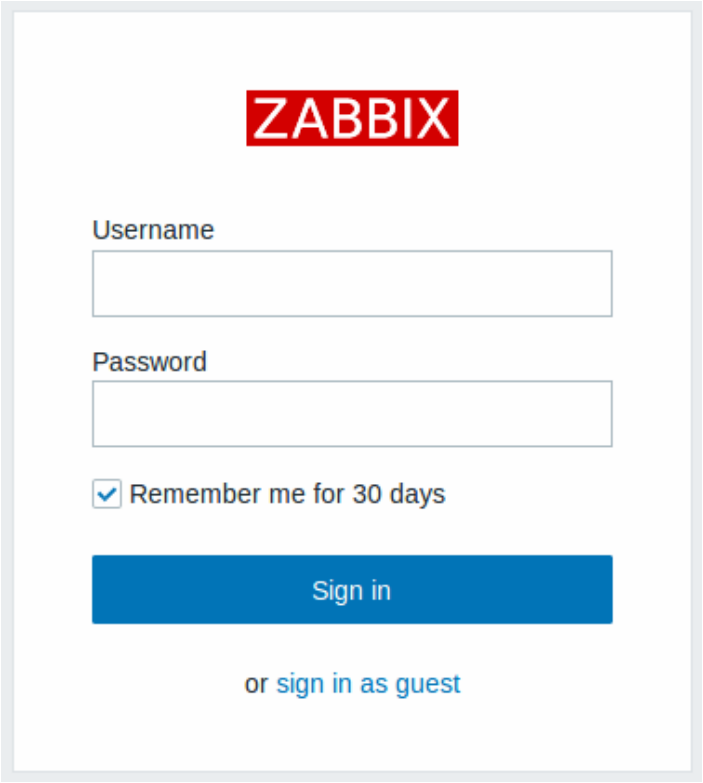
- 障害通知の受信
- 新しいテンプレート

1 ログインとユーザーの設定

概要

このセクションでは、Zabbix にログインし、システムユーザーを設定する方法について説明します。

ログイン



これは Zabbix のウェルカム画面です。**Admin** というユーザー名と **zabbix** というパスワードを入力して、Zabbix スーパーユーザーとしてログインします。すべてのメニューセクションへのアクセスが許可されます。

セキュリティ上の理由から、最初のログイン後すぐに Admin アカウントのデフォルトパスワードを変更することを強く推奨します。

永続的なログイン

最大 30 日間ログイン状態を維持するには、サインインをクリックする前に *30* 日間記憶するを選択します。

30 日間記憶するを有効にした場合:

- セッションは 30 日間アクティブなままです。
- 自動ログアウトが上書きされ、期間が終了するまでサインイン状態が維持されます。
- 30 日以内の次回アクセス時に、認証情報を再入力せずに自動的にログインされます。

30 日間記憶するを無効にした場合:

- 以前有効にした自動ログインがクリアされます。
- セッションは設定された自動ログアウト間隔に従って期限切れになります。

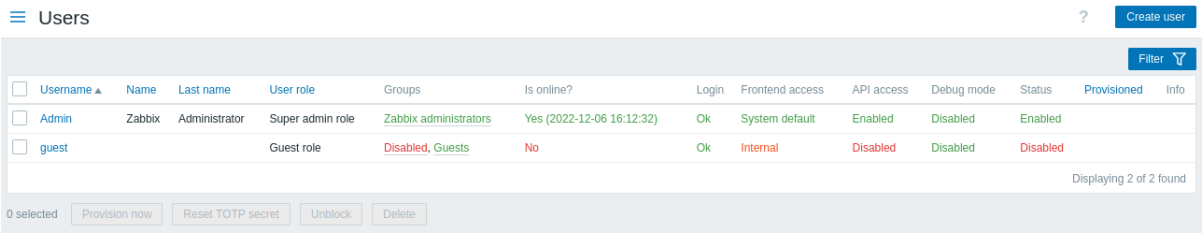
ブルートフォース攻撃からの保護

5 回連続してログインに失敗した場合、Zabbix インターフェースはブルートフォース攻撃や辞書攻撃を防ぐために 30 秒間停止します。

ログインに失敗した IP アドレスは、ログインに成功した後に表示されます。

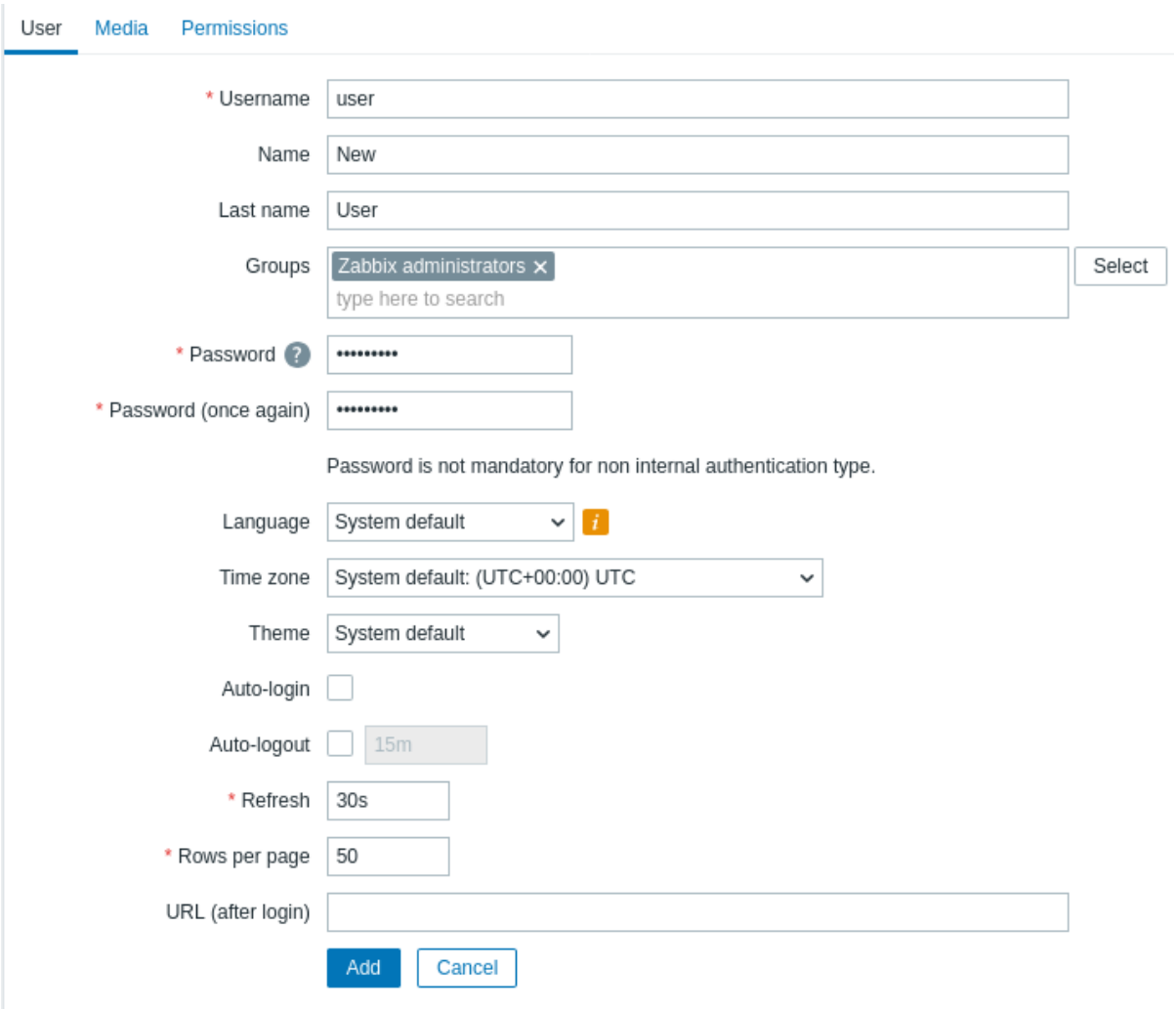
ユーザーの追加

ユーザー情報を表示するには、サイドバーの垂直メニューで ユーザー > ユーザーに移動します。



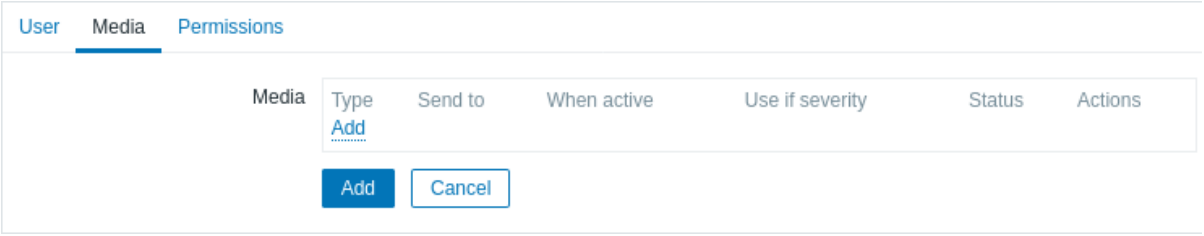
新しいユーザーを追加するには、右上隅の ユーザーの作成を選択します。

新しいユーザーフォームでは、必ずユーザーを既存のユーザーグループのいずれか（例：'Zabbix administrators'）に追加してください。



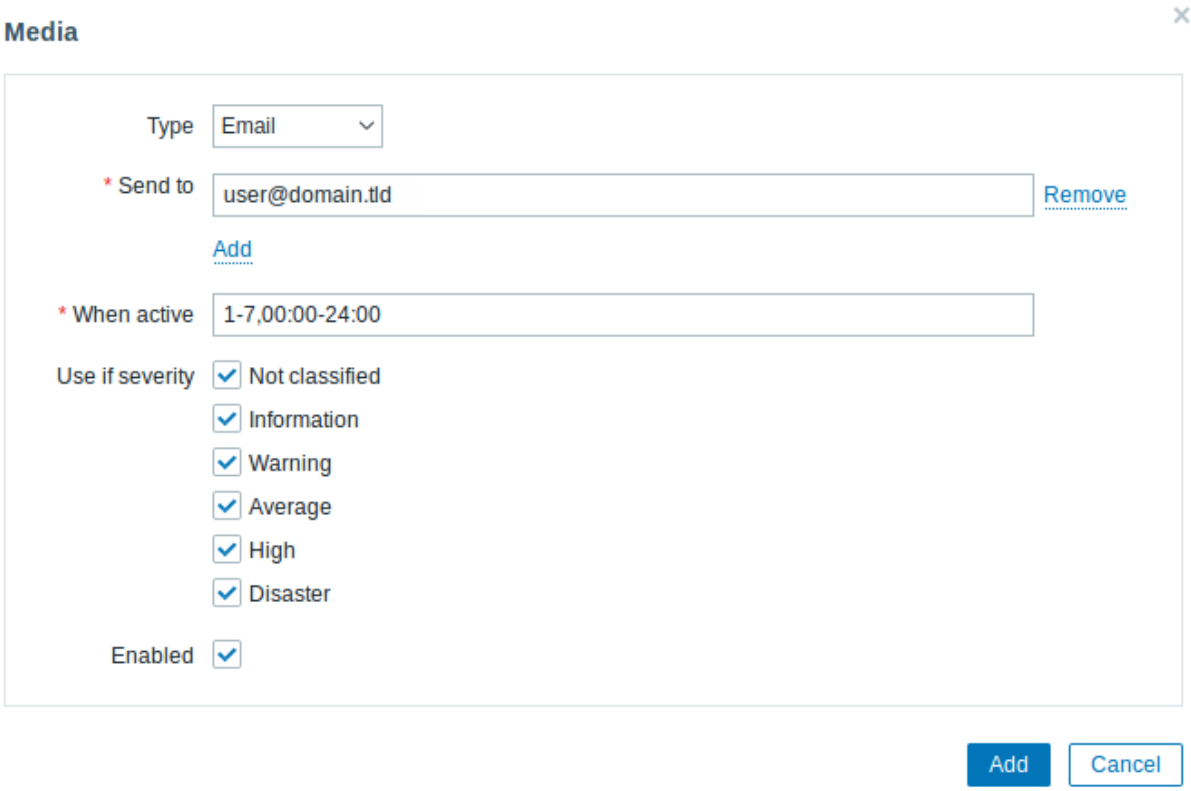
すべての必須入力フィールドには赤いアスタリスクが付いています。この設定フォームの入力フィールドの詳細については、ユーザー設定ページを参照してください。

デフォルトでは、新しいユーザーにはメディア（通知配信方法）が定義されていません。作成するには、「メディア」タブに移動し、追加をクリックします。



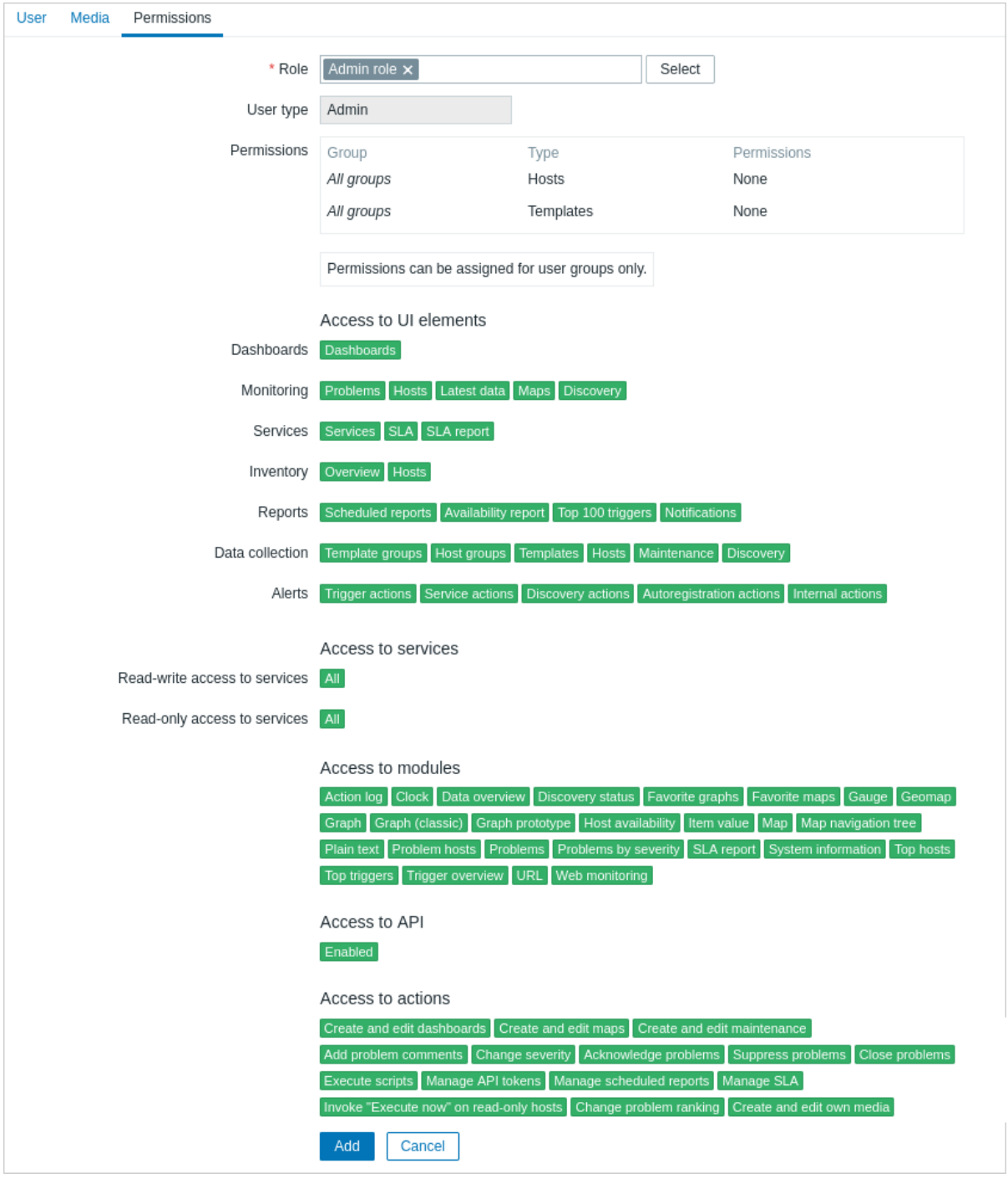
ポップアップで、ユーザーのメールアドレスを入力します。

メディアが有効になる期間を指定できます（フォーマットの説明については期間指定ページを参照）。デフォルトでは、メディアは常に有効です。また、メディアが有効になるトリガーの深刻度レベルをカスタマイズできますが、今はすべて有効のままにしておきます。



追加をクリックしてメディアを保存し、次に「権限」タブに移動します。

「権限」タブには必須フィールドロールがあります。ロールは、ユーザーがどのフロントエンド要素を表示できるか、どの操作を実行できるかを決定します。[選択] を押してリストからロールを選択します。たとえば、*Admin role* を選択すると、管理を除くすべての Zabbix フロントエンドセクションへのアクセスが許可されます。後で権限を変更したり、さらにユーザーロールを作成したりできます。ロールを選択すると、同じタブに権限が表示されます。

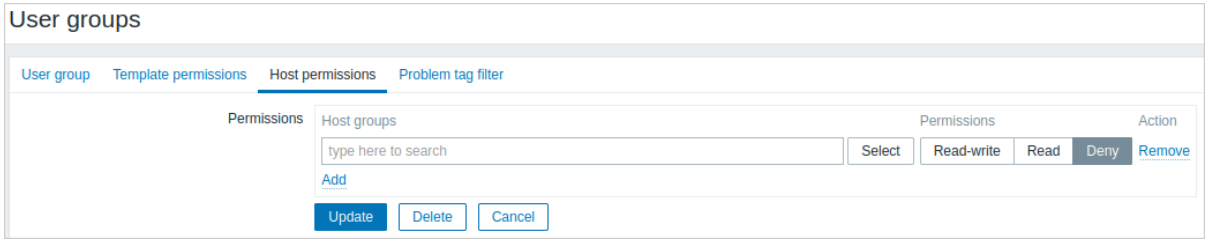


ユーザーのプロパティフォームで追加をクリックしてユーザーを保存します。新しいユーザーがユーザーリストに表示されます。

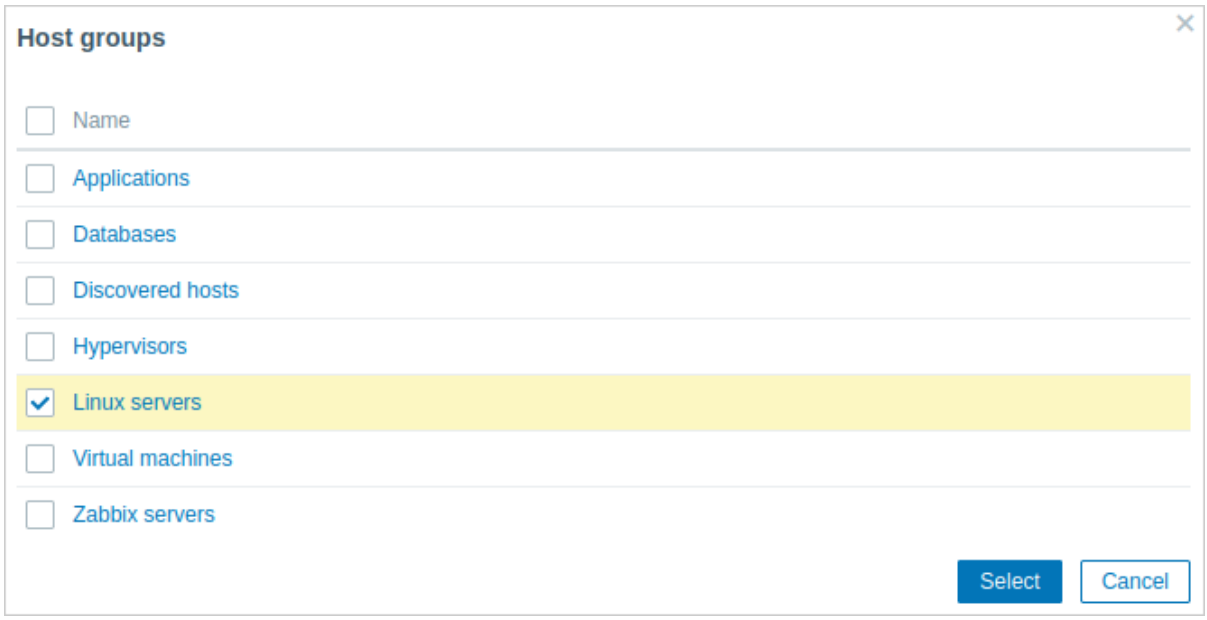


権限の追加

デフォルトでは、新しいユーザーにはホストやテンプレートへのアクセス権がありません。ユーザーに権限を付与するには、グループ列のユーザーグループ（この場合は「Zabbix administrators」）をクリックします。ユーザーグループのプロパティフォームで、ホスト権限タブに移動し、ホストグループへの権限を割り当てます。ホストグループ選択フィールドを表示するには、Add をクリックします。



次に、フィールドの横にある選択をクリックすると、ホストグループのリストが表示されます。このユーザーには *Linux servers* グループへの読み取り専用アクセス権を付与するため、リスト内の該当するチェックボックスをマークし、選択をクリックして選択を確定します。



読み取りボタンをクリックして権限レベルを設定し、更新をクリックしてユーザーグループ構成の変更を保存します。

テンプレートへの権限を付与するには、テンプレート権限タブに切り替えてテンプレートグループを指定する必要があります。手順はホストグループへの権限割り当てと同じです。テンプレートの概要については、このクイックスタートの新しいテンプレートセクションを参照してください。

Attention:
Zabbix では、ホストやテンプレートへのアクセス権はユーザーグループに割り当てられ、個々のユーザーには割り当てられません。

完了です！新しいユーザーの資格情報を使用してログインしてみてください。

2 新しいホスト

概要

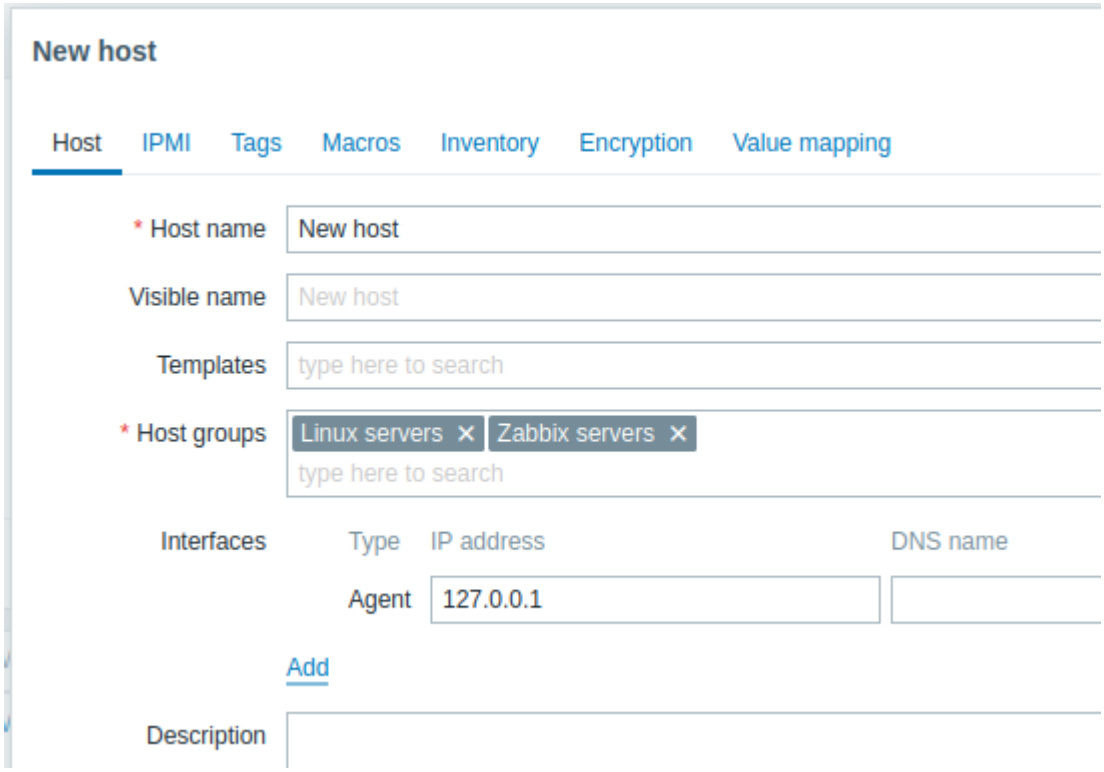
このセクションでは、新しいホストの設定方法について説明します。

Zabbix におけるホストとは、監視したいネットワーク上のエンティティ（物理、仮想）のことです。Zabbix での「ホスト」の定義は非常に柔軟です。物理サーバー、ネットワークスイッチ、仮想マシン、またはアプリケーションなどが該当します。

ホストの追加

Zabbix で設定されたホストの情報は、データ収集 > ホストおよび 監視 > ホストメニューセクションで確認できます。すでに「Zabbix server」という事前定義されたホストが 1 つありますが、ここでは別のホストを追加する方法を学びます。

新しいホストを追加するには、ホストの作成をクリックします。ホストの設定フォームが開きます。



すべての必須入力フィールドには赤いアスタリスクが付いています。

ここで最低限入力する必要があるのは次のとおりです。

ホスト名

- ホスト名を入力します。英数字、スペース、ドット、ハイフン、アンダースコアが使用できます。

ホストグループ

- 選択ボタンをクリックして既存のグループを 1 つまたは複数選択するか、存在しないグループ名を入力して新しいグループを作成します。

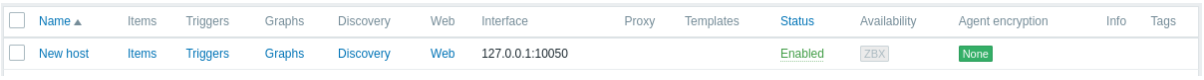
Note:
すべてのアクセス権限はホストグループに割り当てられ、個々のホストには割り当てられません。そのため、ホストは少なくとも 1 つのグループに属している必要があります。

インターフェース***:*** ***IP*** アドレス

- 技術的には必須フィールドではありませんが、特定のメトリクスを収集するにはホストインターフェースが必要です。Zabbix エージェントのパッシブチェックを使用する場合は、このフィールドにエージェントの IP または DNS を指定します。また、Zabbix エージェントの設定ファイルの「Server」ディレクティブにも Zabbix サーバーの IP または DNS を指定する必要があることに注意してください。Zabbix エージェントと Zabbix サーバーが同じマシンにインストールされている場合は、両方の場所で同じ IP/DNS を指定する必要があります。

その他のオプションは、デフォルト値のまま変更しなくてもかまいません。

入力が完了したら、追加をクリックします。新しいホストがホストリストに表示されます。



「可用性」列には、各インターフェースごとのホストの可用性を示すインジケーターが表示されます。Zabbix エージェントインターフェースを定義したので、エージェントの可用性アイコン（「ZBX」と表示されているもの）を使ってホストの可用性を確認できます。

- ZBX - ホストの状態が確立されていません。まだメトリクスのチェックが行われていません
- ZBX - ホストは利用可能で、メトリクスのチェックが成功しています
- ZBX - ホストは利用できず、メトリクスのチェックが失敗しています（アイコンにマウスカーソルを合わせるとエラーメッセージが表示されます）。通信に何らかのエラーが発生している可能性があり、インターフェースの認証情報が正しくない場合があります。Zabbix サーバーが稼働していることを確認し、後でページを更新してみてください。

3 新しいアイテム

概要

このセクションでは、アイテムの設定方法について説明します。

アイテムは Zabbix でデータを収集する基本単位です。アイテムがなければデータはありません。なぜなら、アイテムだけが単一のメトリック、つまりホストからどのようなデータを収集するかを定義するからです。

アイテムの追加

すべてのアイテムはホストごとにグループ化されています。そのため、サンプルアイテムを設定するには、データ収集 > ホストに移動し、作成した「New host」を探します。

「New host」の行のアイテムリンクをクリックし、アイテムの作成をクリックします。アイテムの設定フォームが開きます。

すべての必須入力フィールドには赤いアスタリスクが付いています。

サンプルアイテムで入力する主な情報は次のとおりです。

名前

- 値として *CPU load* を入力します。これはリストや他の場所で表示されるアイテム名になります。

キー

- 値として *system.cpu.load* を手動で入力します。これは収集する情報の種類を識別するアイテムの技術的な名前です。この特定のキーは、Zabbix エージェントに付属する事前定義済みのキーの 1 つです。

情報のタイプ

- この属性は、期待されるデータの形式を定義します。*system.cpu.load* キーの場合、このフィールドは自動的に数値（浮動小数点）に設定されます。

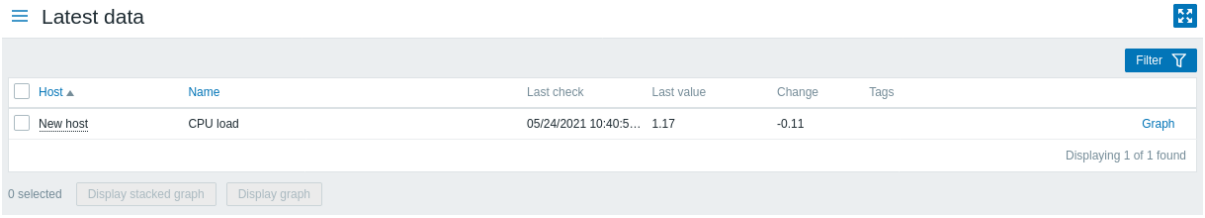
> **Note:**
> アイテムの履歴の保持日数を 7 日または 14 日に減らすことも検討してください。これは、データベースが大量の履歴値を保持する負担を軽減するための良い方法です。

その他のオプションは、デフォルト値のまま変更しなくてもかまいません。

完了したら追加をクリックします。新しいアイテムがアイテムリストに表示され、成功メッセージが表示されます。

データの表示

アイテムが定義されたら、実際にデータが収集されているかどうか気になるかもしれません。そのためには、監視データ > 最新データに移動し、フィルタで「New host」を選択して 適用をクリックします。



ただし、最初のデータが到着するまで最大 60 秒かかる場合があります。これは、デフォルトでサーバーが設定変更を読み取り、新しいアイテムを実行する頻度です。

「変更」列に値が表示されない場合は、これまでに 1 つの値しか受信されていない可能性があります。もう 1 つの値が到着するまで 30 秒待ってください。

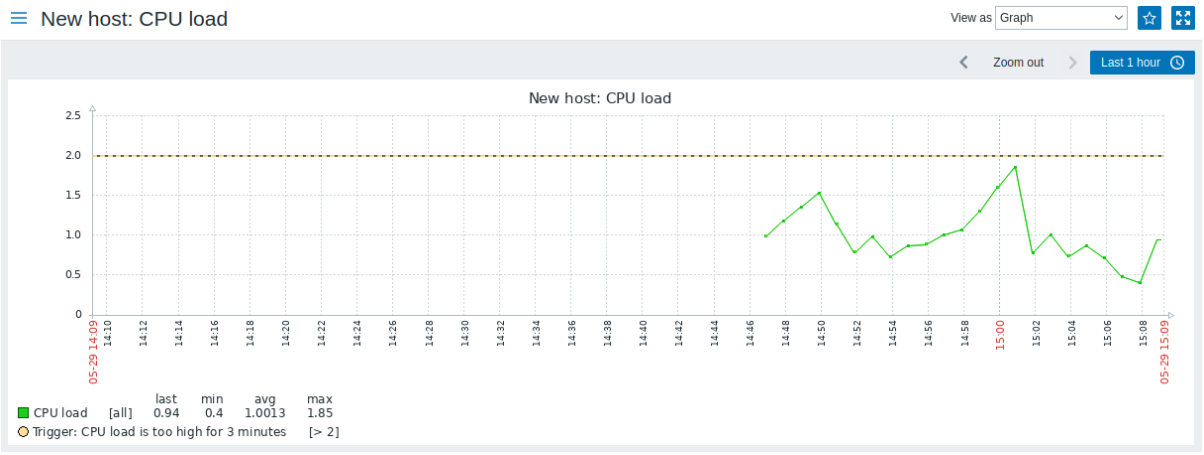
スクリーンショットのようにアイテムに関する情報が表示されない場合は、以下を確認してください。

- アイテムの「キー」と「情報のタイプ」フィールドがスクリーンショットとまったく同じように入力されていること
- エージェントとサーバーの両方が稼働していること
- ホストのステータスが「監視中」で、可用性アイコンが緑色であること
- ホストフィルタで選択されているホストが正しいこと
- アイテムが有効になっていること

グラフ

アイテムがしばらく動作している場合、何か視覚的なものを見てみる時期かもしれません。シンプルグラフは、追加の設定なしで監視されている数値アイテムすべてに利用可能です。これらのグラフは実行時に生成されます。

グラフを表示するには、監視 > 最新データに移動し、アイテムの横にある「グラフ」リンクをクリックします。



4 新しいトリガー

概要

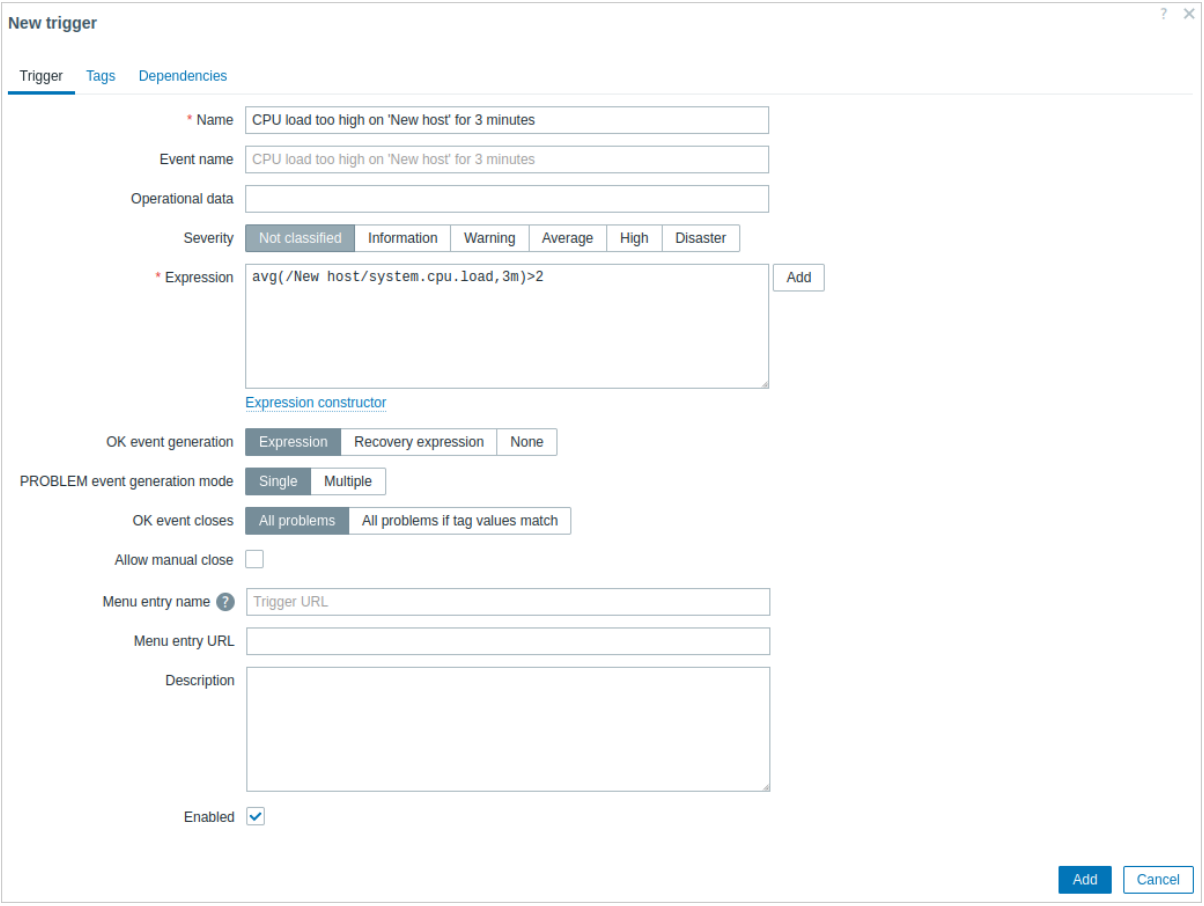
このセクションでは、トリガーの設定方法について説明します。

アイテムはデータを収集するだけです。受信データを自動的に評価するには、トリガーを定義する必要があります。トリガーには、データにとって許容できるレベルのしきい値を定義する式が含まれています。

そのレベルを受信データが超えると、トリガーが「発火」し、「障害」状態になります。これは、注意が必要な何かが発生したことを示します。レベルが再び許容範囲内になると、トリガーは「正常」状態に戻ります。

トリガーの追加

アイテムのトリガーを設定するには、データ収集 > ホストに移動します。「New host」の行の トリガーリンクをクリックし、トリガーの作成をクリックします。トリガーの設定フォームが開きます。



このトリガーで入力する主な情報は次のとおりです。

名前

- 値として *CPU load too high on 'New host' for 3 minutes* を入力します。これはリストなどに表示されるトリガー名になります。

式

- 次のように入力します: avg(/New host/system.cpu.load,3m)>2

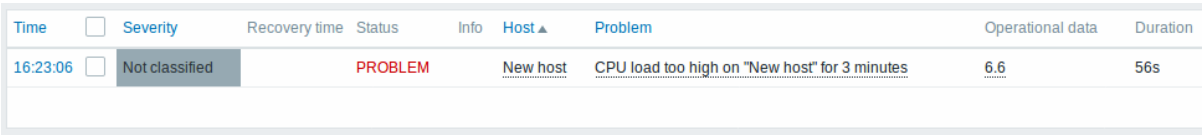
これはトリガーの式です。式が最後の記号まで正しく入力されていることを確認してください。ここでのアイテムキー（system.cpu.load）はアイテムを参照するために使用されます。この特定の式は、3 分間の CPU 負荷の平均値が 2 を超えた場合に問題の閾値を超えたことを意味します。トリガー式の構文について詳しく学ぶことができます。

完了したら、追加をクリックします。新しいトリガーがトリガーリストに表示されるはずです。

トリガーステータスの表示

トリガーが定義されたら、そのステータスを確認したくなるかもしれません。

トリガーで定義したしきい値を CPU 負荷が超えた場合、監視データ > 障害に障害が表示されます。



ステータス列の点滅は、トリガーステータスが最近（過去 30 分以内）に変化したことを示しています。

5 問題通知の受信

概要

このセクションでは、Zabbix で通知という形でアラートを設定する方法を学びます。

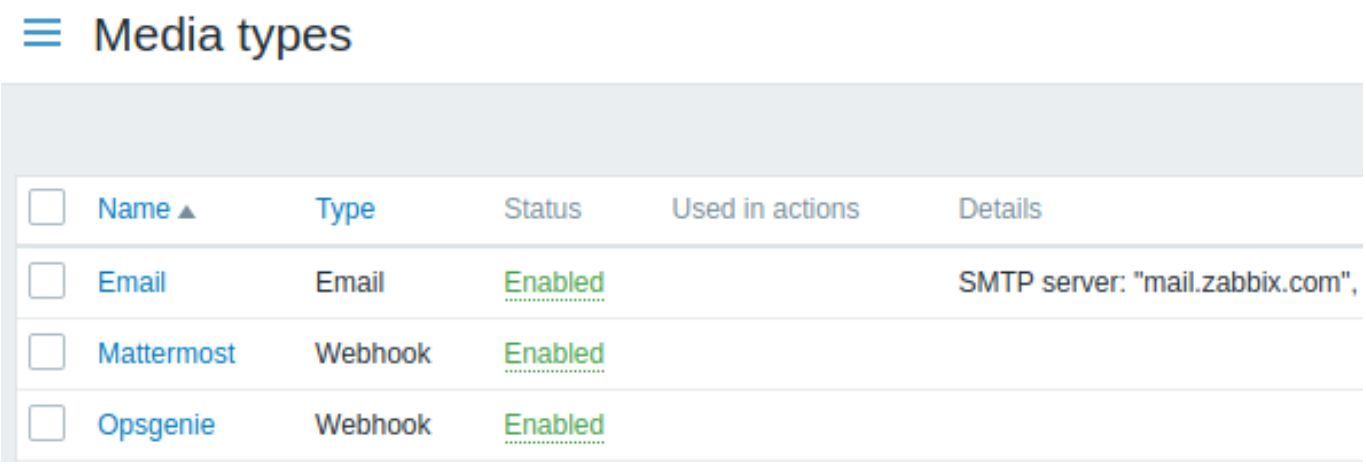
アイテムがデータを収集し、トリガーが問題発生時に「発火」するように設計されている場合でも、Zabbix フロントエンドを積極的にチェックしていないときでも重要なイベントを通知するアラートメカニズムがあると便利です。

これが通知の役割です。問題通知の配信方法として最も一般的なメールについて、メール通知の設定方法を学びます。

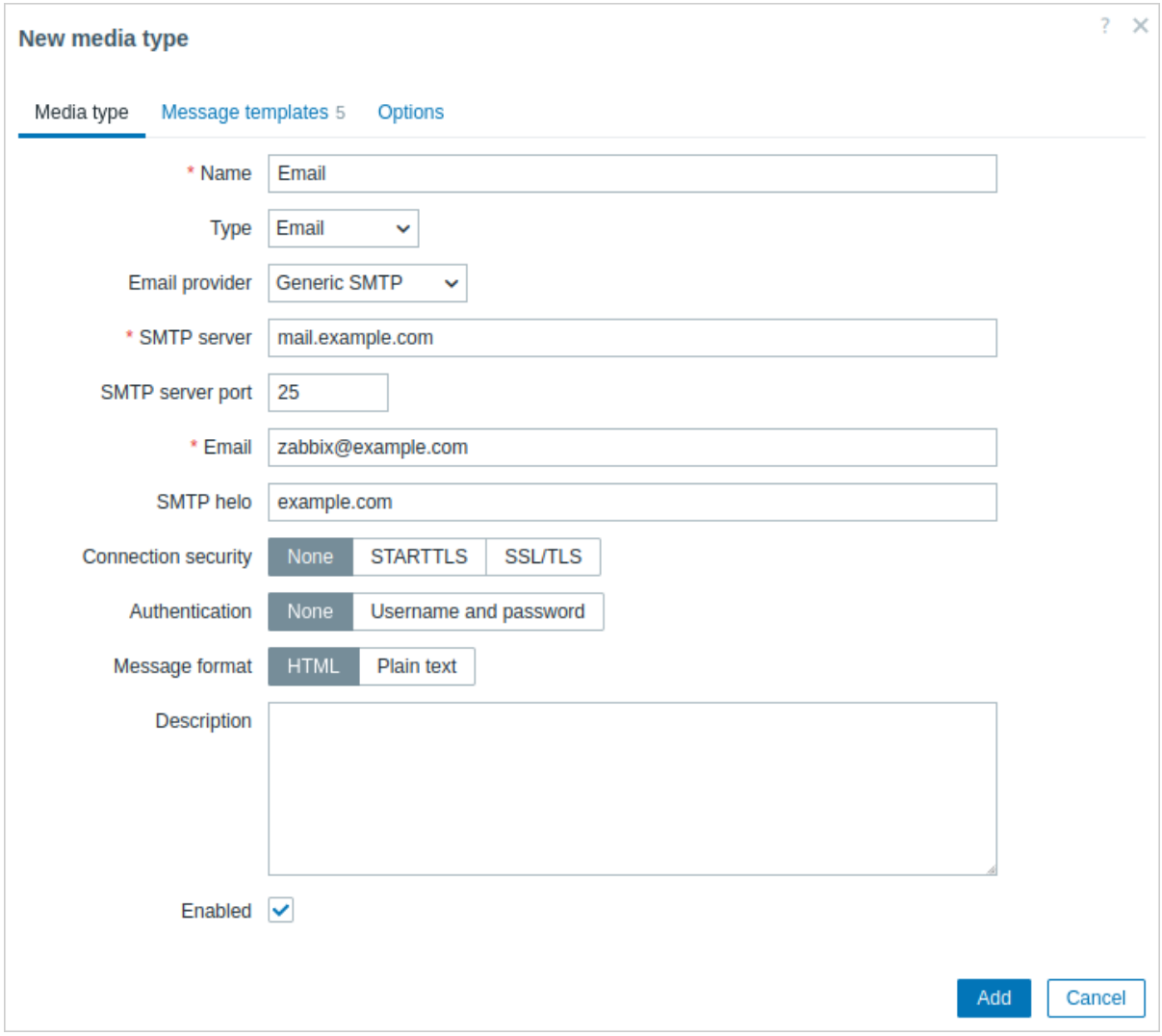
メール設定

最初に、Zabbix にはいくつかの事前定義された通知配信方法があります。メールはその 1 つです。

メール設定を構成するには、アラート > メディアタイプに移動し、事前定義されたメディアタイプのリストから *Email* をクリックします。



メール設定の構成フォームが開きます。



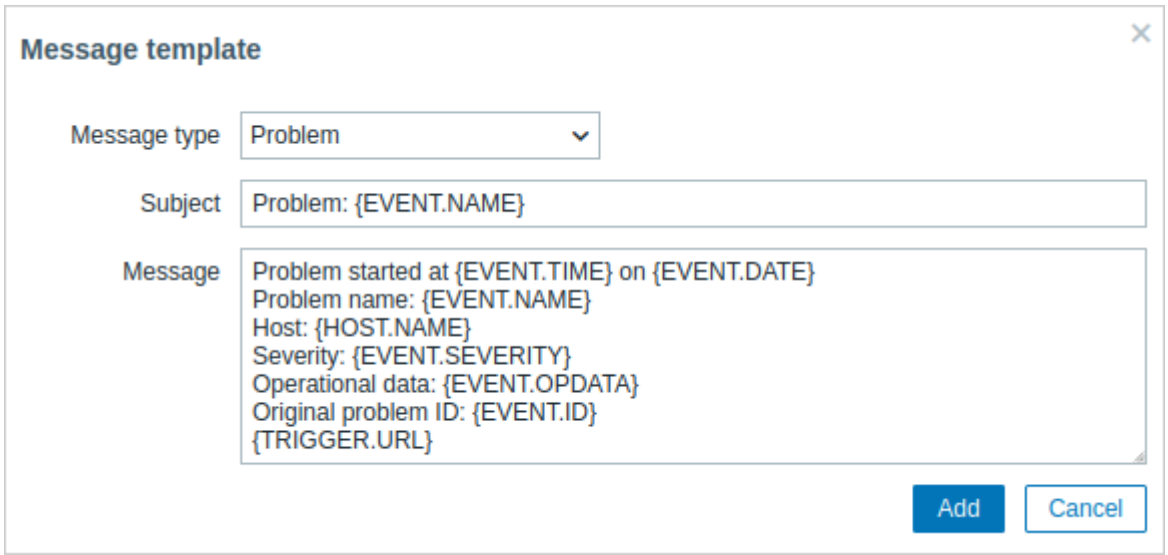
すべての必須入力フィールドには赤いアスタリスクが付いています。

メディアタイプタブで、*SMTP* サーバー、*SMTP helo*、*Email* の値を環境に適したものに設定します。

Note:
Email フィールドの値は、Zabbix から送信される通知の「From」アドレスとして使用されます。

次に、問題メッセージの内容を定義する必要があります。内容は、メッセージテンプレートタブで設定されたメッセージテンプレートによって定義されます。

追加をクリックしてメッセージテンプレートを作成し、メッセージタイプとして問題を選択します。

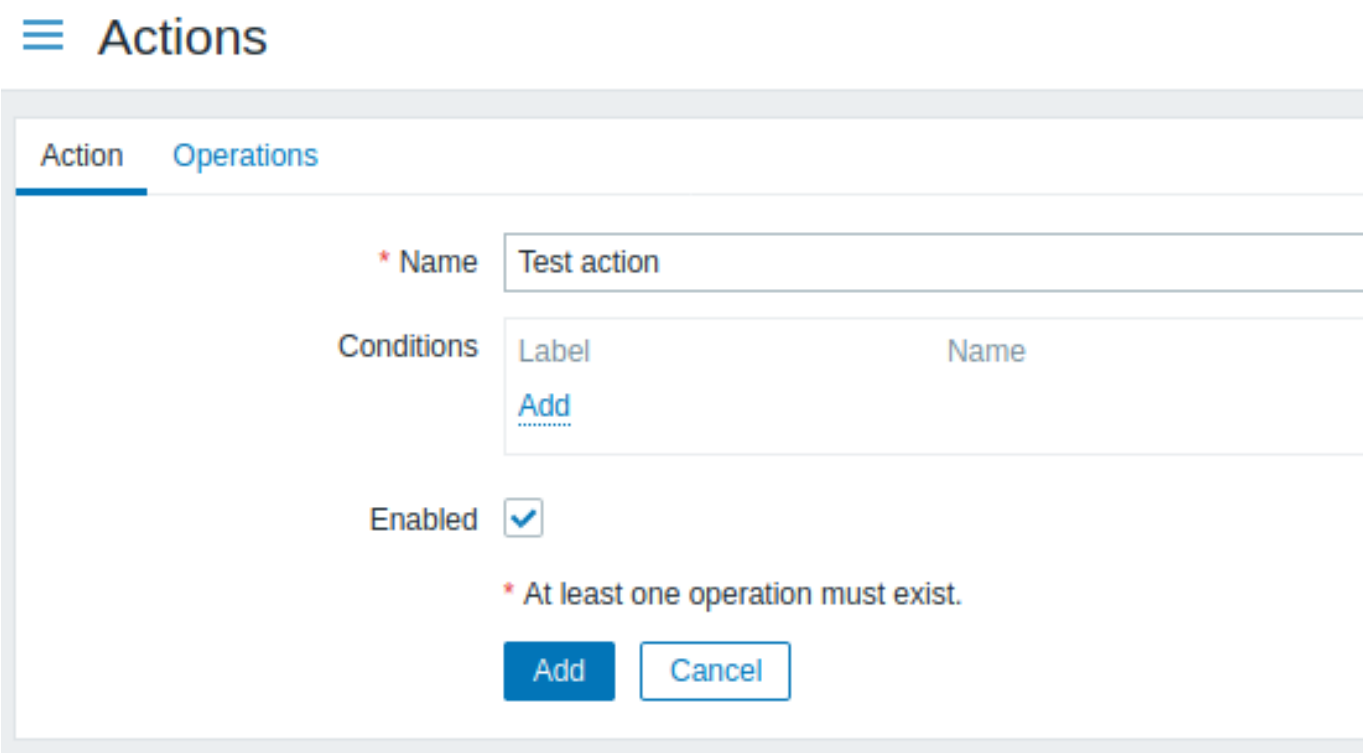


準備ができたら追加をクリックし、フォームを保存します。

これで *Email* を動作するメディアタイプとして構成できました。メディアタイプは、特定の配信先アドレスを定義することでユーザーにもリンクする必要があります（新しいユーザーの設定時と同様）、そうしないと使用されません。

新しいアクション

通知の配信は、Zabbix のアクションが行うことの 1 つです。したがって、通知を設定するには、アラート > アクション > トリガーアクションに移動し、アクションの作成をクリックします。

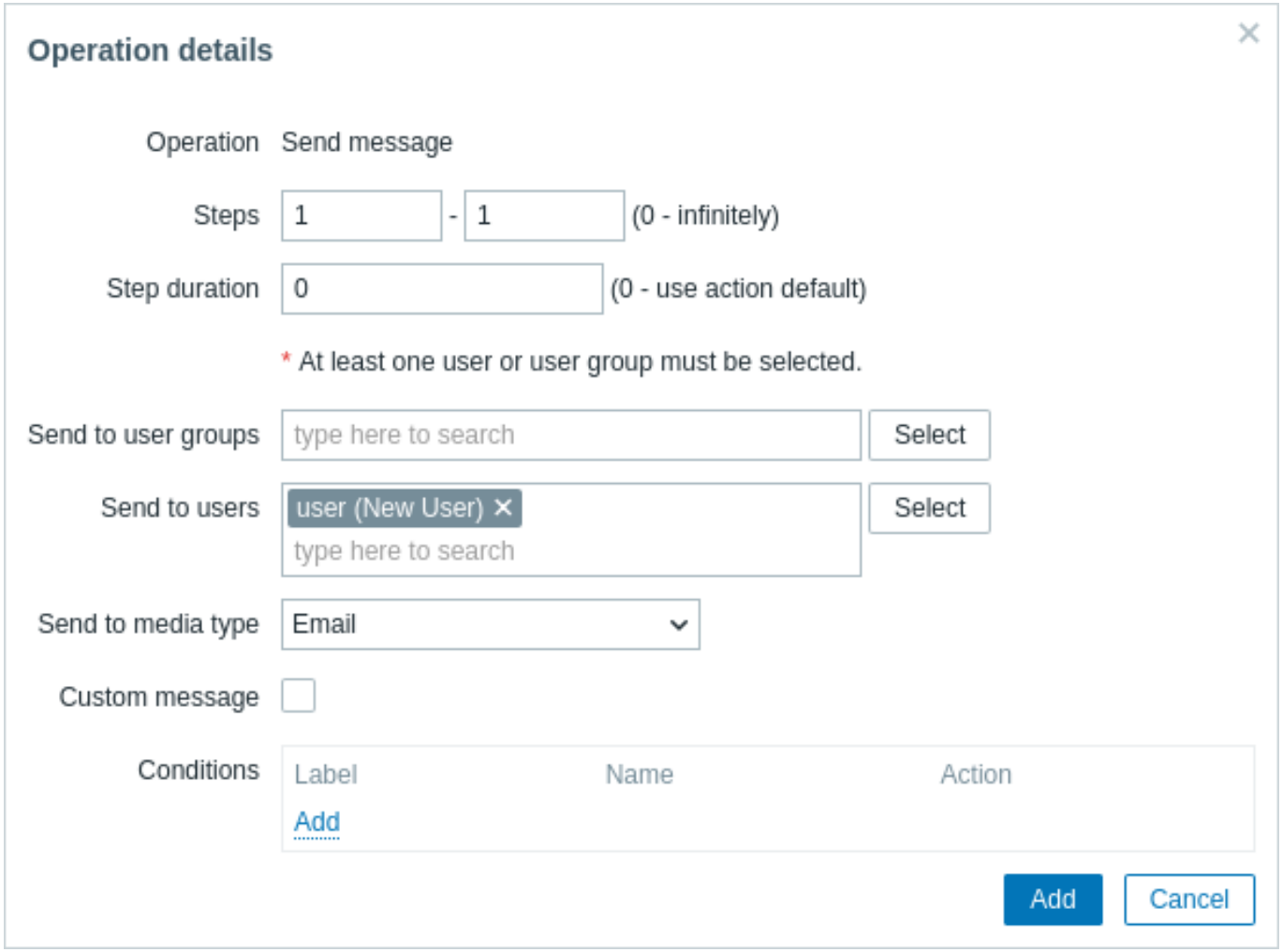


必須入力フィールドはすべて赤いアスタリスクでマークされています。

このフォームでは、アクションの名前を入力します。

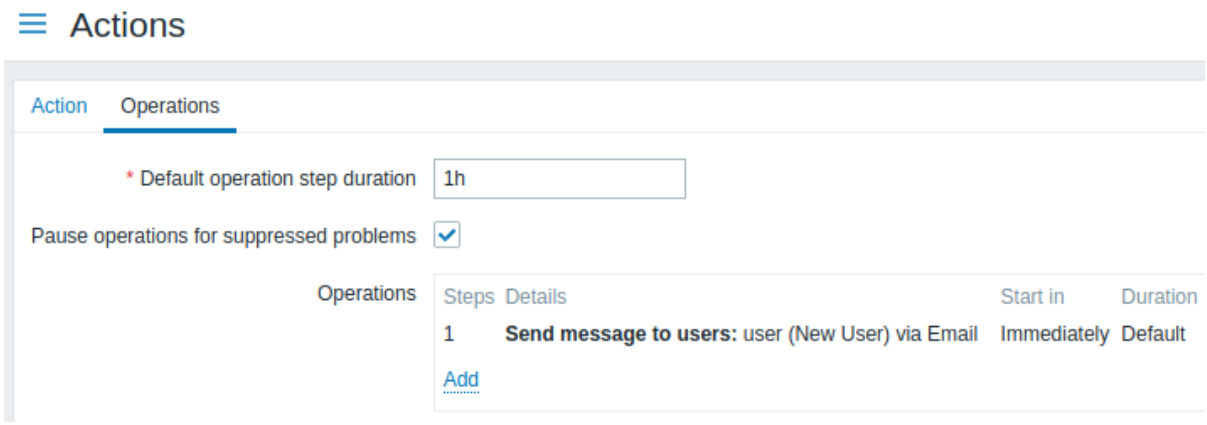
最も単純な場合、より具体的な条件を追加しなければ、アクションは「正常」から「障害」へのトリガーの変更時に実行されます。

アクションが何をすべきかを定義する必要があります。これはオペレーションタブで行います。オペレーションブロックで追加をクリックすると、新しいオペレーションフォームが開きます。



必須入力フィールドはすべて赤いアスタリスクでマークされています。

ここで、ユーザーに送信ブロックの選択をクリックし、定義したユーザー（「user」）を選択します。メディアタイプに送信の値として「Email」を選択します。これが完了したら追加をクリックし、オペレーションが追加されていることを確認します。



これでシンプルなアクション設定は完了なので、アクションフォームで追加をクリックします。

通知の受信

これで通知の配信が設定できたので、実際に通知を受け取ってみましょう。そのために、意図的にホストの負荷を上げて、トリガーが「発生」し、問題通知を受け取れるようにします。

ホストのコンソールを開いて、次のコマンドを実行します。

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

このプロセスを 1 つまたは複数実行してもかまいません。

監視データ > 最新データに移動し、「CPU Load」の値がどのように増加したかを確認します。トリガーが「発生」するには、「CPU Load」の値が 3 分間連続して「2」を超える必要があることを思い出してください。条件を満たすと:

- 監視データ > 障害で、トリガーが点滅する「障害」ステータスで表示されます。
- メールで問題通知を受け取ります。

Attention:
通知が動作しない場合:
- メール設定とアクションの両方が正しく設定されていることを再度確認してください
- ユーザーの追加の手順で説明したように、作成したユーザーがイベントを生成したホストに対して少なくとも読み取り権限を持っていることを確認してください。ユーザーは「Zabbix administrators」ユーザーグループの一員であり、ホストが属する「Linux servers」ホストグループに対して少なくとも読み取り権限を持っている必要があります。
- さらに、レポート > アクションログでアクションログを確認することもできます。

6 新しいテンプレート

概要

このセクションでは、テンプレートの設定方法について説明します。

前のセクションでは、アイテム、トリガーの設定方法と、ホストの問題通知の受信方法について学びました。

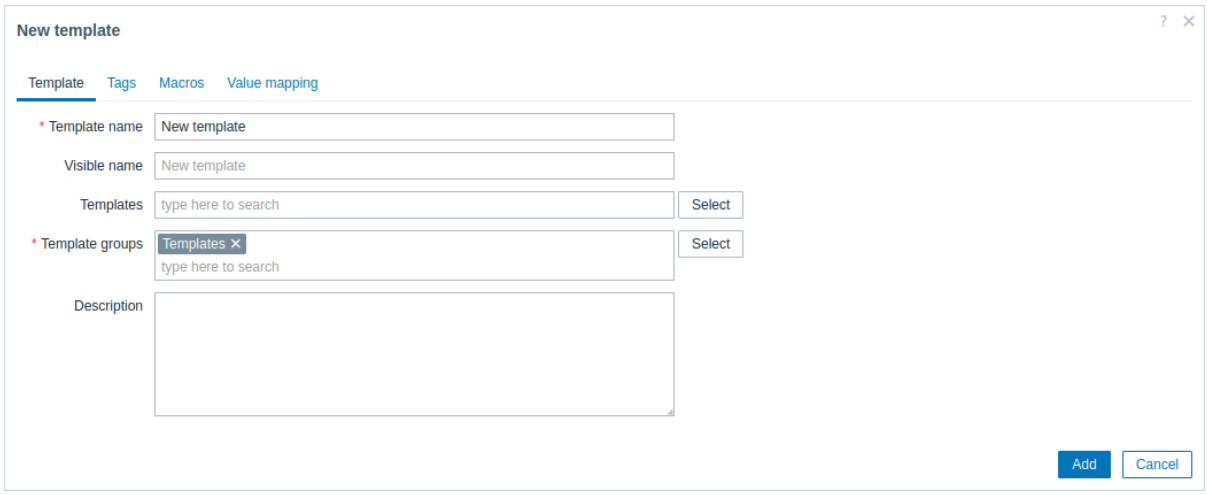
これらの手順はそれ自体で非常に柔軟性がありますが、例えば 1000 台のホストに必要な場合は、多くの手順が必要に思えるかもしれません。自動化があれば便利です。

ここでテンプレートが役立ちます。テンプレートを使用すると、便利なアイテム、トリガー、その他のエンティティをグループ化し、それらをホストに 1 ステップで適用することで何度も再利用できます。

テンプレートがホストにリンクされると、ホストはテンプレートのすべてのエンティティを継承します。つまり、事前に用意された一連のチェックを非常に迅速に適用できます。

テンプレートの追加

テンプレートの作業を開始するには、まずテンプレートを作成する必要があります。そのためには、データ収集 > テンプレートで テンプレートの作成をクリックします。これにより、テンプレートの設定フォームが表示されます。



すべての必須入力フィールドには赤いアスタリスクが付いています。

ここで入力する必要があるパラメータは次のとおりです。

テンプレート名

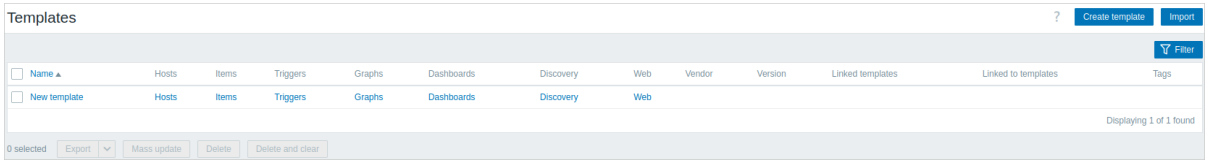
- テンプレート名を入力します。英数字、スペース、アンダースコアが使用できます。

テンプレートグループ

- 選択ボタンをクリックして、1 つまたは複数のグループを選択します。テンプレートはグループに属している必要があります。

Note:
テンプレートグループへのアクセス権限は、ユーザーグループの設定でテンプレート権限タブからホスト権限と同様に割り当てます。すべてのアクセス権限はグループに割り当てられ、個々のテンプレートには割り当てられません。そのため、テンプレートを少なくとも 1 つのグループに含めることが必須です。

完了したら、追加をクリックします。新しいテンプレートがテンプレートの一覧に表示されます。また、フィルターを使用してテンプレートを検索することもできます。



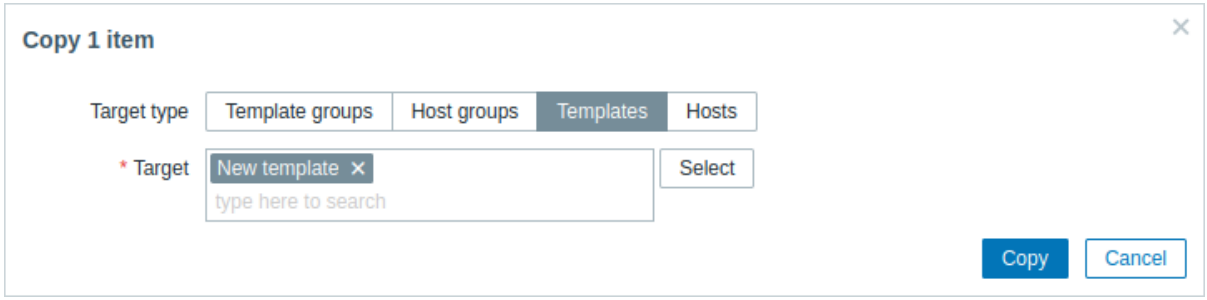
ご覧のとおり、テンプレートは存在しますが、アイテムやトリガーなどのエンティティはまだ何も含まれていません。

テンプレートへのアイテムの追加

テンプレートにアイテムを追加するには、データ収集 → ホストに移動し、「New host」の横にある アイテムをクリックして、「New host」のアイテムリストを開きます。

次に:

- リスト内の「CPU Load」アイテムのチェックボックスをオンにします。
- リストの下にある コピーをクリックします。
- テンプレートタブを選択します。
- アイテムをコピーするテンプレートを選択します。



すべての必須入力フィールドには赤いアスタリスクが付いています。

- コピーをクリックします。

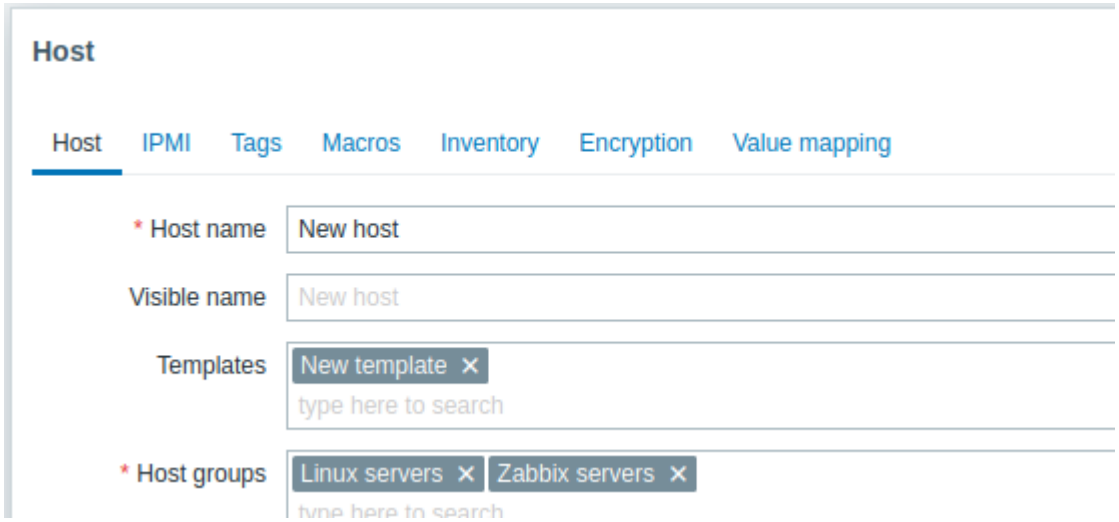
データ収集 > テンプレートに移動すると、「New template」に新しいアイテムが 1 つ追加されているはずです。

今回は 1 つのアイテムだけにしますが、同様にして他のアイテムやトリガー、その他のエンティティもテンプレートに追加でき、目的（OS の監視、単一アプリケーションの監視など）に応じてかなり完全なエンティティのセットにすることができます。

テンプレートのホストへのリンク

テンプレートの準備ができたら、あとはそれをホストに追加するだけです。そのためには、データ収集 > ホストに移動し、「新規ホスト」をクリックして設定フォームを開き、テンプレートフィールドを探します。

テンプレートフィールドに *New template* と入力し始めます。作成したテンプレート名がドロップダウンリストに表示されるはずです。スクロールして選択します。テンプレートフィールドに表示されていることを確認してください。



フォームの更新をクリックして変更を保存します。これでテンプレートがホストに追加され、テンプレートが保持しているすべてのエンティティが追加されます。

この方法で、他のホストにも適用できます。テンプレートレベルでアイテム、トリガー、その他のエンティティに変更を加えると、そのテンプレートがリンクされているホストにも反映されます。

事前定義されたテンプレートをホストにリンクする

ご存知のように、Zabbix にはさまざまな OS、デバイス、アプリケーション用の事前定義されたテンプレートが用意されています。非常に迅速に監視を開始するには、それらの中から適切なものをホストにリンクすることができますが、これらのテンプレートは環境に合わせて微調整する必要があることに注意してください。一部のチェックは不要かもしれませんし、ポーリング間隔が頻繁すぎる場合もあります。

テンプレートに関する詳細情報もご覧ください。

Apache Web サーバーの監視

Apache Web サーバーの監視を開始するには、次の手順に従います。

1) Apache の統計情報へのアクセスを確認する
2) Apache 監視テンプレートを適用する
3) 最新データを確認する

このガイドを使用するには、Zabbix サーバーと Web インターフェースがインストールされている必要があります。Zabbix にはデフォルトの *Admin* ユーザーとしてログインしてください。

Apache 統計情報へのアクセス

このステップの目的は、Zabbix が Apache Web サーバーの統計情報を読み取れることを確認することです。

Apache では、サーバーのパフォーマンスやアクティビティに関するリアルタイム情報は、組み込みの `mod_status` モジュールによって提供されます。有効にすると、パフォーマンスやアクティビティのデータを表示する Web アクセス可能なページ (通常は`/server-status`) が作成されます。

Apache Web サーバーで `mod_status` が有効になっていることを確認するには、次のコマンドを実行します。

Debian/Ubuntu の場合	RHEL 系システムの場合
`apache2ctl -M | grep status`	`httpd -M | grep status`

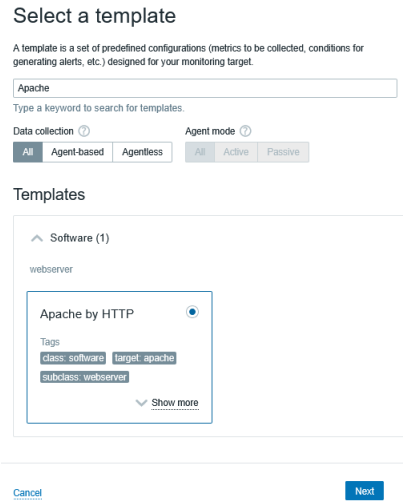
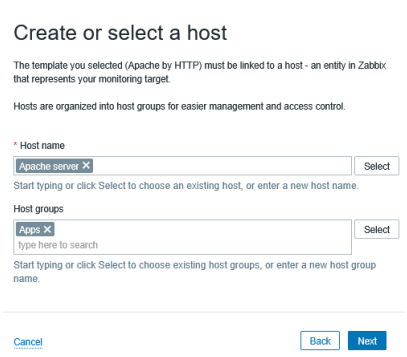
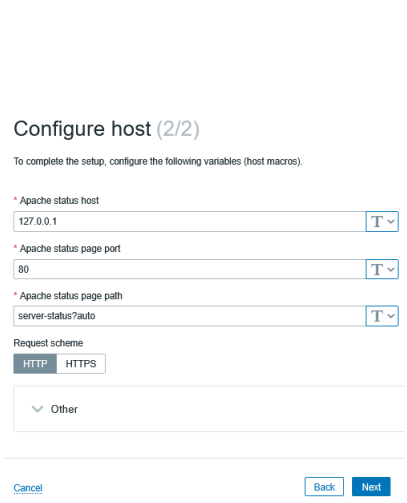
出力に"`status_module (shared)`" が表示されていれば、status モジュールはすでに有効になっています。表示されない場合は、status モジュールを有効にする必要があります。

Apache 監視テンプレートの適用

Zabbix では、このテンプレートはすぐに使用できます。データ収集、可視化、分析のために事前設定された要素が含まれています。

テンプレートを適用するには、ホストウィザード（データ収集 > ホスト）を開始します。

- テンプレートを選択（"Apache by HTTP"）
- Apache ホストを作成し、グループを追加します。この仮想ホストには Apache 関連のデータが含まれます。
- ホストを設定します。重要なのは、Apache Web サーバーのアドレス（ローカルマシンの場合は 127.0.0.1、リモートマシンの場合は IP アドレス）を入力することです。

ステップ 1	ステップ 2	ステップ 3
		
テンプレートの選択。	ホスト/ホストグループの作成。	ホストの設定。

他の設定はデフォルトのままにしておくことができます。

ホストウィザードを完了するには、作成と完了をクリックします。

最新データを確認する

Latest data

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

おめでとうございます。Apache Web サーバーの監視が開始されました！

Zabbix は可視化オプションを提供し、問題が発生した場合にアラートを配信します。メールでアラートを設定するには、問題通知の受信を参照してください。

Linux サーバーの監視

Linux サーバーの監視を開始するには、以下の手順に従います。

1) Linux 監視テンプレートを適用する
2) サーバーに Zabbix エージェントをインストールする
3) 最新データを確認する

ローカルでない Linux サーバーの監視を開始するには、その IP アドレスと Zabbix サーバーの IP アドレスを知っておく必要があります。

Linux サーバーの監視には Zabbix エージェントを使用します。エージェントのインストール手順はこのプロセスで案内されます。

このガイドを利用するには、Zabbix サーバーと Web インターフェースがインストールされている必要があります。デフォルトの *Admin* ユーザーで Zabbix にログインしてください。

Linux 監視テンプレートの適用

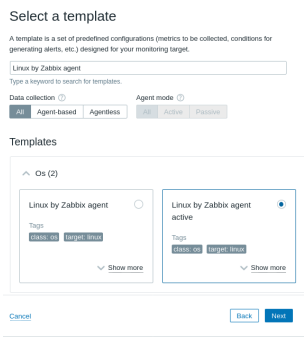
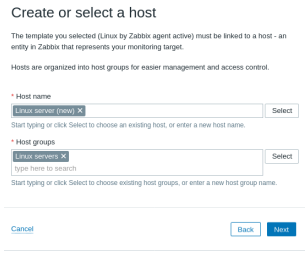
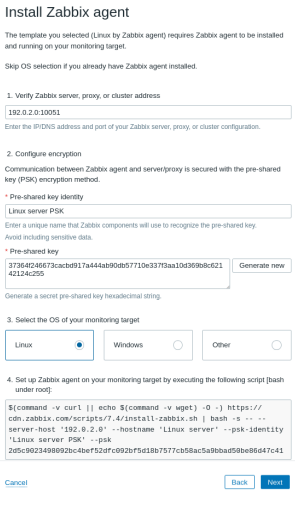
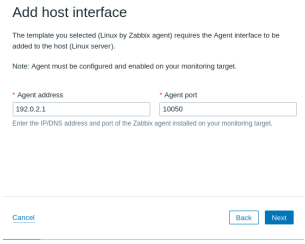
Zabbix では、このテンプレートはすぐに使用できます。データ収集、可視化、分析のために事前に設定された要素が含まれています。

テンプレートを適用するには、ホストウィザード（データ収集 > ホスト）を開始します。

- テンプレートを選択（”Linux by Zabbix agent active”）
- Linux サーバー用のホストを作成し、グループを追加します。この仮想ホストには、Linux サーバー監視に関連するすべてのデータが含まれます。
- エージェントを設定してインストールします。エージェントをインストールするには、Linux サーバー上で提供されたコマンドを実行します。これにより、エージェントがインストールされ、暗号化された Zabbix サーバーへの接続が設定されます。

すでにエージェントがインストールされている場合は、このステップをスキップできます（次へをクリック）。ただし、エージェントに接続するための事前共有キー ID と事前共有キーの入力が求められます。

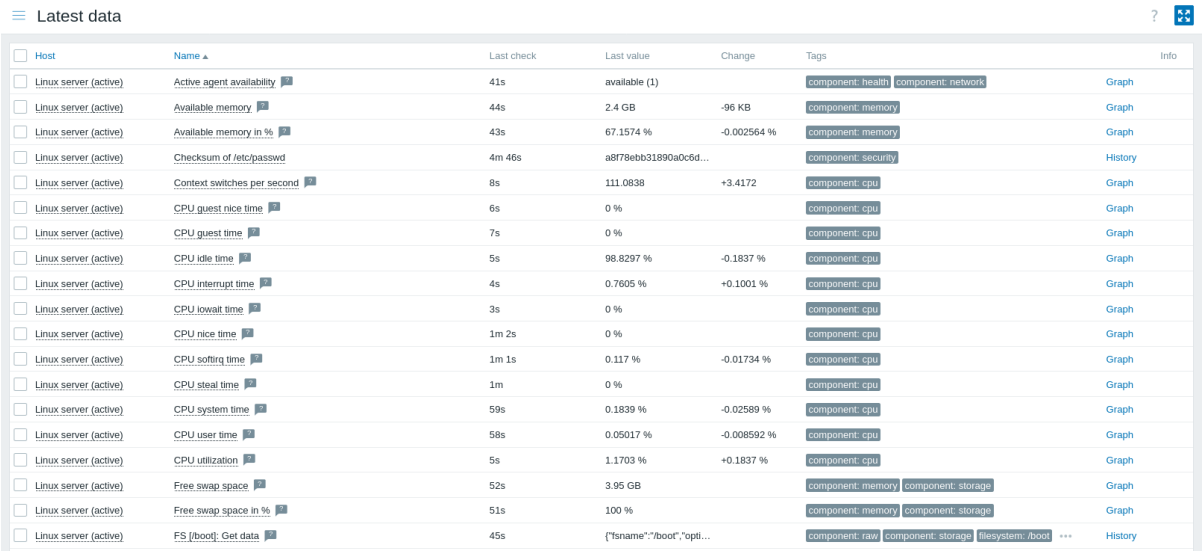
- ホストの設定を完了します。エージェント（Linux サーバー）の IP アドレスとポートを入力します。

ステップ 1	ステップ 2	ステップ 3	ステップ 4
			
テンプレートの選択。	ホスト/ホストグループの作成。	ホスト上でのエージェントの設定とインストール。	Linux サーバーの IP アドレス/ポートをエージェントインターフェースとして追加。

他の設定はデフォルトのままにしておくことができます。

ホストウィザードを完了するには、作成と完了をクリックします。

最新データを確認する



Latest data

Host	Name	Last check	Last value	Change	Tags	Info
Linux server (active)	Active agent availability	41s	available (1)		component: health, component: network	Graph
Linux server (active)	Available memory	44s	2.4 GB	-96 KB	component: memory	Graph
Linux server (active)	Available memory in %	43s	67.1574 %	-0.002564 %	component: memory	Graph
Linux server (active)	Checksum of /etc/passwd	4m 46s	a8f78ebb31890a0c6d…		component: security	History
Linux server (active)	Context switches per second	8s	111.0838	+3.4172	component: cpu	Graph
Linux server (active)	CPU guest nice time	6s	0 %		component: cpu	Graph
Linux server (active)	CPU guest time	7s	0 %		component: cpu	Graph
Linux server (active)	CPU idle time	5s	98.8297 %	-0.1837 %	component: cpu	Graph
Linux server (active)	CPU interrupt time	4s	0.7605 %	+0.1001 %	component: cpu	Graph
Linux server (active)	CPU iowait time	3s	0 %		component: cpu	Graph
Linux server (active)	CPU nice time	1m 2s	0 %		component: cpu	Graph
Linux server (active)	CPU softirq time	1m 1s	0.117 %	-0.01734 %	component: cpu	Graph
Linux server (active)	CPU steal time	1m	0 %		component: cpu	Graph
Linux server (active)	CPU system time	59s	0.1839 %	-0.02589 %	component: cpu	Graph
Linux server (active)	CPU user time	58s	0.05017 %	-0.008592 %	component: cpu	Graph
Linux server (active)	CPU utilization	5s	1.1703 %	+0.1837 %	component: cpu	Graph
Linux server (active)	Free swap space	52s	3.95 GB		component: memory, component: storage	Graph
Linux server (active)	Free swap space in %	51s	100 %		component: memory, component: storage	Graph
Linux server (active)	FS [/boot]: Get data	45s	{"fsname":"/boot","opti…		component: raw, component: storage, filesystem: /boot	History

おめでとうございます。Linux サーバーの監視が開始されました！

Zabbix は可視化オプションを提供し、問題が発生した場合にアラートを配信します。メールでアラートを設定するには、問題通知の受信を参照してください。

MySQL サーバーの監視

MySQL サーバーの監視を開始するには、以下の手順に従ってください。

1) 制限付き MySQL ユーザーの作成
2) MySQL 監視テンプレートの適用
3) 最新データの確認

MySQL サーバーの監視を開始するには、その DSN と Zabbix サーバーの IP アドレスを知っておく必要があります。

MySQL サーバーの監視には Zabbix エージェントを使用します。エージェントのインストール手順はこのプロセスで説明します。

このガイドを使用するには、Zabbix サーバーと Web インターフェースがインストールされている必要があります。デフォルトの *Admin* ユーザーで Zabbix にログインしている必要があります。

制限付き MySQL ユーザーの作成

MySQL の監視に必要な権限のみを持つ制限付きの MySQL ユーザーを作成することを推奨します。

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

MySQL 監視テンプレートの適用

Zabbix では、このテンプレートはすぐに使用できます。データ収集、可視化、分析のために事前に設定された要素が含まれています。

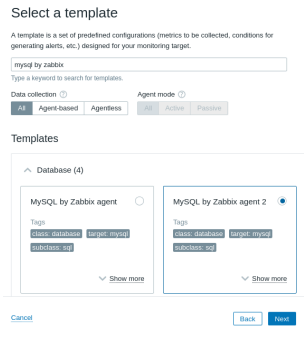
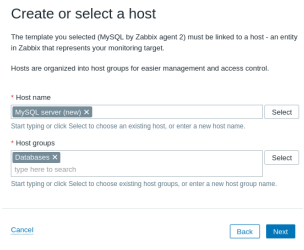
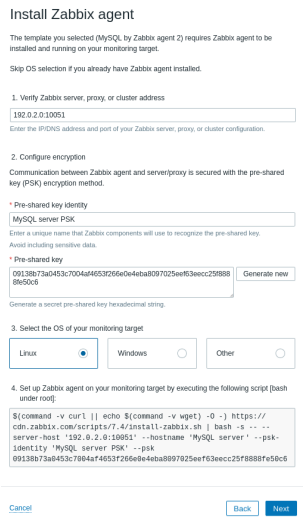
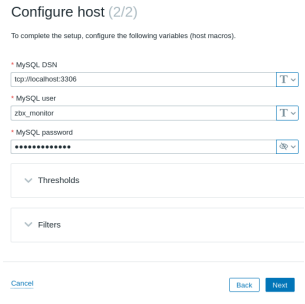
テンプレートを適用するには、ホストウィザード（データ収集 > ホスト）を開始します。

- テンプレートを選択（”MySQL by Zabbix agent 2”）
- MySQL サーバーホストを作成し、グループを追加します。この仮想ホストには、MySQL サーバー監視に関連するすべてのデータが含まれます。
- エージェント 2 を設定してインストールします（Zabbix agent 2 は MySQL 監視を標準で提供します）。

エージェントをインストールするには、MySQL サーバーをホストしているマシンで提供されたコマンドを実行します。これにより、エージェントがインストールされ、Zabbix サーバーへの接続が暗号化付きで設定されます。

すでにエージェント 2 がインストールされている場合は、このステップをスキップできます（次へをクリック）。ただし、エージェントに接続するための事前共有キー ID と事前共有キーの入力が求められます。

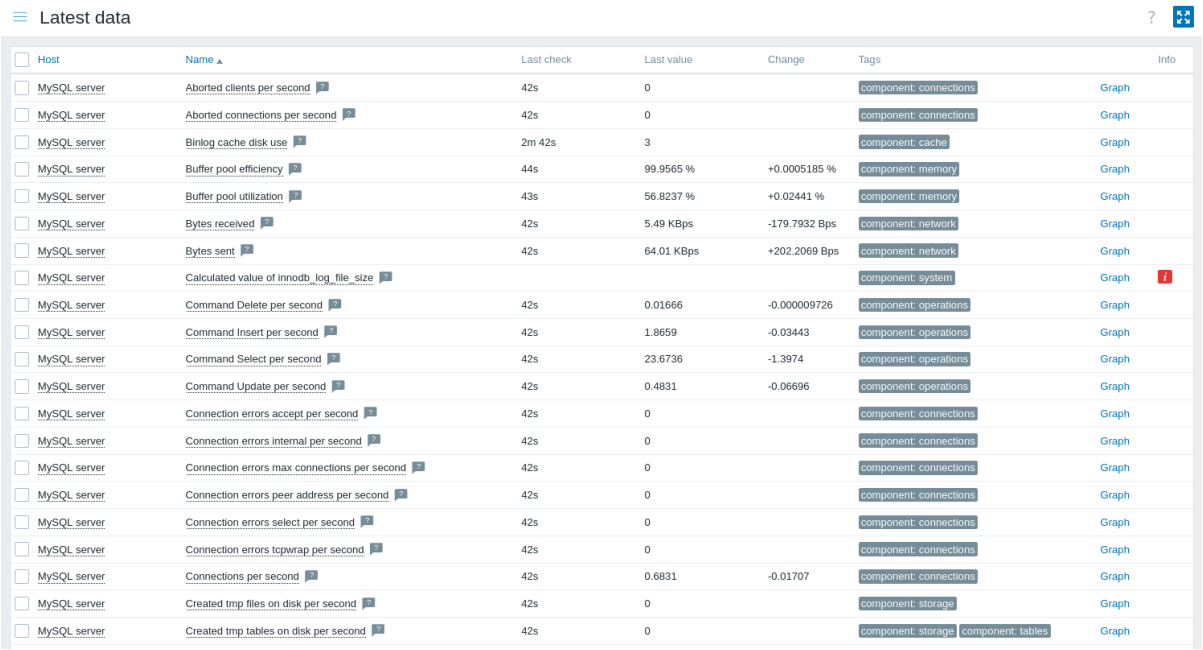
- ホストの設定を完了します。重要なのは、MySQL サーバーの DSN、MySQL ユーザー、およびパスワードの入力が求められることです。作成した `zbx_monitor` ユーザーとそのパスワードを入力してください。

ステップ 1	ステップ 2	ステップ 3	ステップ 4
			
テンプレートの選択。	ホスト/ホストグループの作成。	ホスト上でのエージェントの設定とインストール。	MySQL サーバーの DNS、ユーザー、パスワードの追加。

他の設定はデフォルトのままにしておくことができます。

ホストウィザードを完了するには、作成と完了をクリックします。

最新データを確認する



おめでとうございます。MySQL サーバーの監視が開始されました！

Zabbix は可視化オプションを提供し、問題が発生した場合にアラートを通知します。メールでアラートを設定するには、問題通知の受信を参照してください。

3 インストール**:** 追加ガイド

Zabbix の最も簡単なインストール方法をお探しの場合は、インストールと初期設定をご覧ください。

以下のガイドは、Zabbix またはそのコンポーネントをインストールする追加またはシステム固有の方法を説明しています:

- コンテナからのインストール
- ソースからのインストール
- パッケージからのインストールに関する注意事項
- Windows MSI からの Zabbix エージェントのインストール
- Windows での Zabbix agent 2 のビルド
- Windows での Zabbix エージェントのビルド
- PKG からの macOS エージェントのインストール
- macOS での Zabbix エージェントのビルド
- Zabbix アプライアンス

macOS での **Zabbix** エージェントのビルド

概要

このセクションでは、TLS の有無にかかわらず、ソースから Zabbix macOS エージェントのバイナリをビルドする方法を説明します。

前提条件

コマンドライン開発者ツール Automake、pkg-config、および PCRE(v8.x) または PCRE2(v10.x) が必要です (Xcode は必要ありません)。TLS を使用してエージェントバイナリをビルドする場合は、OpenSSL または GnuTLS も必要になります。

Automake と pkg-config をインストールするには、https://brew.sh/の Homebrew パッケージマネージャーが必要です。インストールするため、ターミナルを開いて次のコマンドを実行します。

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

次に Automake と pkg-config をインストールします。

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

PCRE、OpenSSL、および GnuTLS ライブラリの準備は、エージェントへのリンク方法によって異なります。

これらのライブラリが既にある macOS マシンでエージェントバイナリを実行する場合は、Homebrew が提供するコンパイル済みのライブラリを使用できます。これらは通常、Zabbix エージェントバイナリの構築やその他の目的で Homebrew を使用する macOS マシンです。

共有バージョンのライブラリがない macOS マシンでエージェントバイナリを使用する場合は、ソースから静的ライブラリをコンパイルし、Zabbix エージェントをリンクする必要があります。

共有ライブラリを使用したエージェントバイナリのビルド

PCRE2 をインストール (必要に応じて、以下のコマンドの *pcre2* を *pcre* に置き換えてください)。

```
brew install pcre2
```

TLS でビルドする場合は OpenSSL と GnuTLS の両方、あるいはどちらかをインストール。

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

Zabbix ソースをダウンロード。

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

TLS なしでエージェントをビルド。

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

OpenSSL を使用してエージェントをビルド。

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

GnuTLS を使用してエージェントをビルド。

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

TLS を使用しない静的ライブラリを使用したエージェントバイナリのビルド

$HOME/static-libs に PCRE 静的ライブラリがインストールされると仮定します。PCRE2 10.39 を使用します。

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

Unicode プロパティをサポートする PCRE をダウンロードしてビルドします。

```
mkdir static-libs-source
cd static-libs-source
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
```

Zabbix ソースとビルドエージェントをダウンロードします。

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

OpenSSL と静的ライブラリを使用してエージェントバイナリをビルドする

OpenSSL をビルドするとき、ビルドが成功した後に `make test` を実行することをお勧めします。ビルドが成功した場合でも、テストが失敗することがあります。この場合は、続行する前に問題を調査して解決する必要があります。

PCRE と OpenSSL の静的ライブラリが `$HOME/static-libs` にインストールされると仮定します。PCRE2 10.39 と OpenSSL 1.1.1a を使用します。

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

`static-libs-source` で静的ライブラリをビルドします。

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

Unicode プロパティをサポートする PCRE をダウンロードしてビルドする。

```
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
cd ..
```

OpenSSL をダウンロードしてビルドする。

```
curl --remote-name https://www.openssl.org/source/openssl-1.1.1a.tar.gz
tar xf openssl-1.1.1a.tar.gz
cd openssl-1.1.1a
./Configure --prefix="$OPENSSL_PREFIX" --openssldir="$OPENSSL_PREFIX" --api=1.1.0 no-shared no-capieng no-
make
make test
make install_sw
cd ..
```

Zabbix ソースとビルドエージェントをダウンロードする。

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

GnuTLS と静的ライブラリを使用してエージェントバイナリをビルドする

GnuTLS は、Nettle 暗号バックエンドと GMP 演算ライブラリに依存しています。完全な GMP ライブラリを使用する代わりに、このガイドでは、Nettle に含まれている mini-gmp を使用します。

GnuTLS と Nettle をビルドするときは、ビルドが成功した後に `make check` を実行することをお勧めします。ビルドが成功した場合でも、テストが失敗することがあります。この場合は、続行する前に問題を調査して解決する必要があります。

PCRE、Nettle、および GnuTLS 静的ライブラリが `$HOME/static-libs` にインストールされると仮定します。PCRE2 10.39、Nettle 3.4.1、および GnuTLS 3.6.5 を使用します。

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
NETTLE_PREFIX="$HOME/static-libs/nettle-3.4.1"
GNUTLS_PREFIX="$HOME/static-libs/gnutls-3.6.5"
```

`static-libs-source` で静的ライブラリをビルドします。

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

Nettle をダウンロードしてビルドします。

```
curl --remote-name https://ftp.gnu.org/gnu/nettle/nettle-3.4.1.tar.gz
tar xf nettle-3.4.1.tar.gz
cd nettle-3.4.1
./configure --prefix="$NETTLE_PREFIX" --enable-static --disable-shared --disable-documentation --disable-a
```

```
make
make check
make install
cd ..
```

GnuTLS をダウンロードしてビルドします。

```
curl --remote-name https://www.gnupg.org/ftp/gcrypt/gnutls/v3.6/gnutls-3.6.5.tar.xz
tar xf gnutls-3.6.5.tar.xz
cd gnutls-3.6.5
PKG_CONFIG_PATH="$NETTLE_PREFIX/lib/pkgconfig" ./configure --prefix="$GNUTLS_PREFIX" --enable-static --dis
make
make check
make install
cd ..
```

Zabbix ソースとビルドエージェントをダウンロードをします。

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
CFLAGS="-Wno-unused-command-line-argument -framework Foundation -framework Security" \
> LIBS="-lgnutls -lhogweed -lnettle" \
> LDFLAGS="-L$GNUTLS_PREFIX/lib -L$NETTLE_PREFIX/lib" \
> ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFI
make
make install
```

MSI からの **Windows** エージェントのインストール

概要

Zabbix エージェントは、32 ビットまたは 64 ビットの MSI インストーラーパッケージを使用して Windows にインストールできます。これらはダウンロードできます。

MSI インストールの最小 OS 要件は以下の通りです。

- **Zabbix** エージェントの場合**:** Windows XP (64 ビット) または Windows Server 2003
- **Zabbix agent 2** の場合**:** Windows 10 (32 ビット) または Windows Server 2016

32 ビットパッケージは 64 ビットシステムにはインストールできません。

パッケージには以下が含まれます。

- TLS サポート（TLS の設定は任意）
- Zabbix get およびZabbix sender ユーティリティ（Zabbix エージェント/agent 2 と一緒、または個別にインストール可能）

> **Attention:**
> Zabbix agent 2 パッケージにはロード可能なプラグイン（MongoDB、PostgreSQL、MSSQL）は含まれていません。これらは別途ダウンロードしてインストールする必要があります。

インストールは、セットアップウィザードまたはコマンドラインを使用して行うことができます。

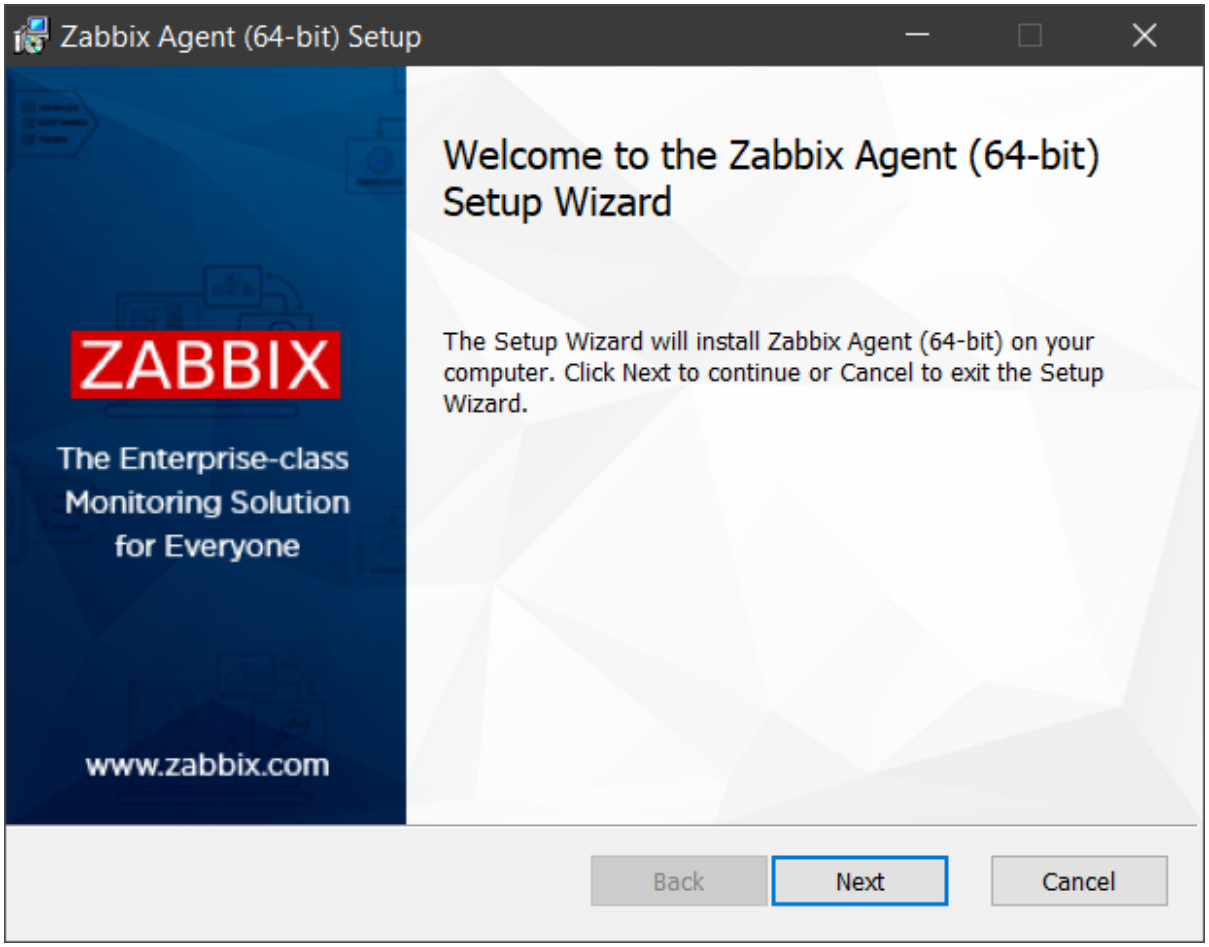
MSI パッケージによるインストールは完全にサポートされていますが、適切なエラー処理のために少なくともMicrosoft .NET Framework 2のインストールを推奨します。

> **Attention:**
> インストーラーが提供するデフォルトのインストール場所を使用することを推奨します。必要な権限がないカスタムの場所を使用すると、インストールのセキュリティが損なわれる可能性があります。

セットアップウィザードからのインストール

以下のインストール手順は、Zabbix agent および Zabbix agent 2 の両方に適用されます。

1. ダウンロードした MSI ファイルをダブルクリックしてインストールを開始します。

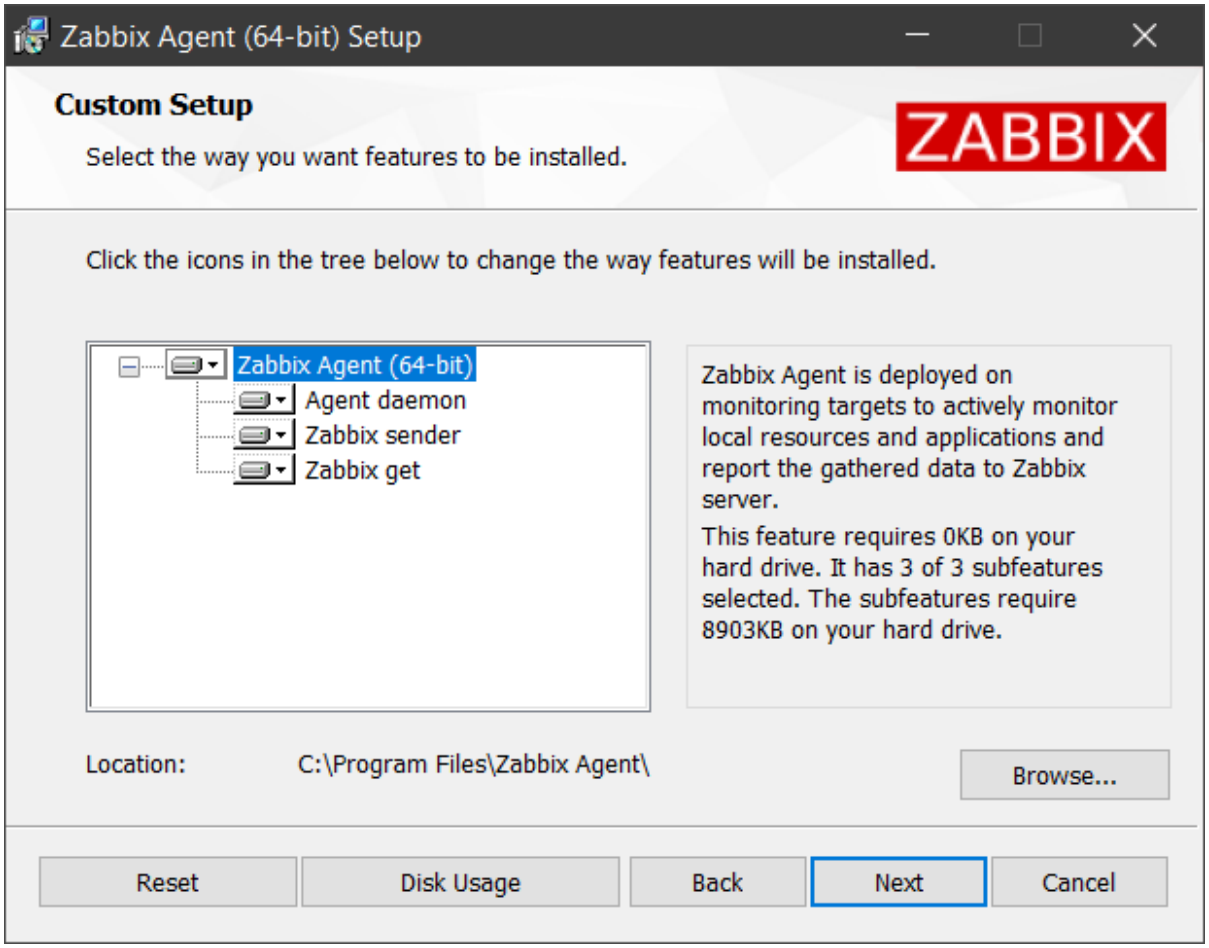


2. エンドユーザーライセンス契約に同意します。



3. インストールする Zabbix コンポーネント（Agent daemon、Zabbix sender、Zabbix get）を選択します。

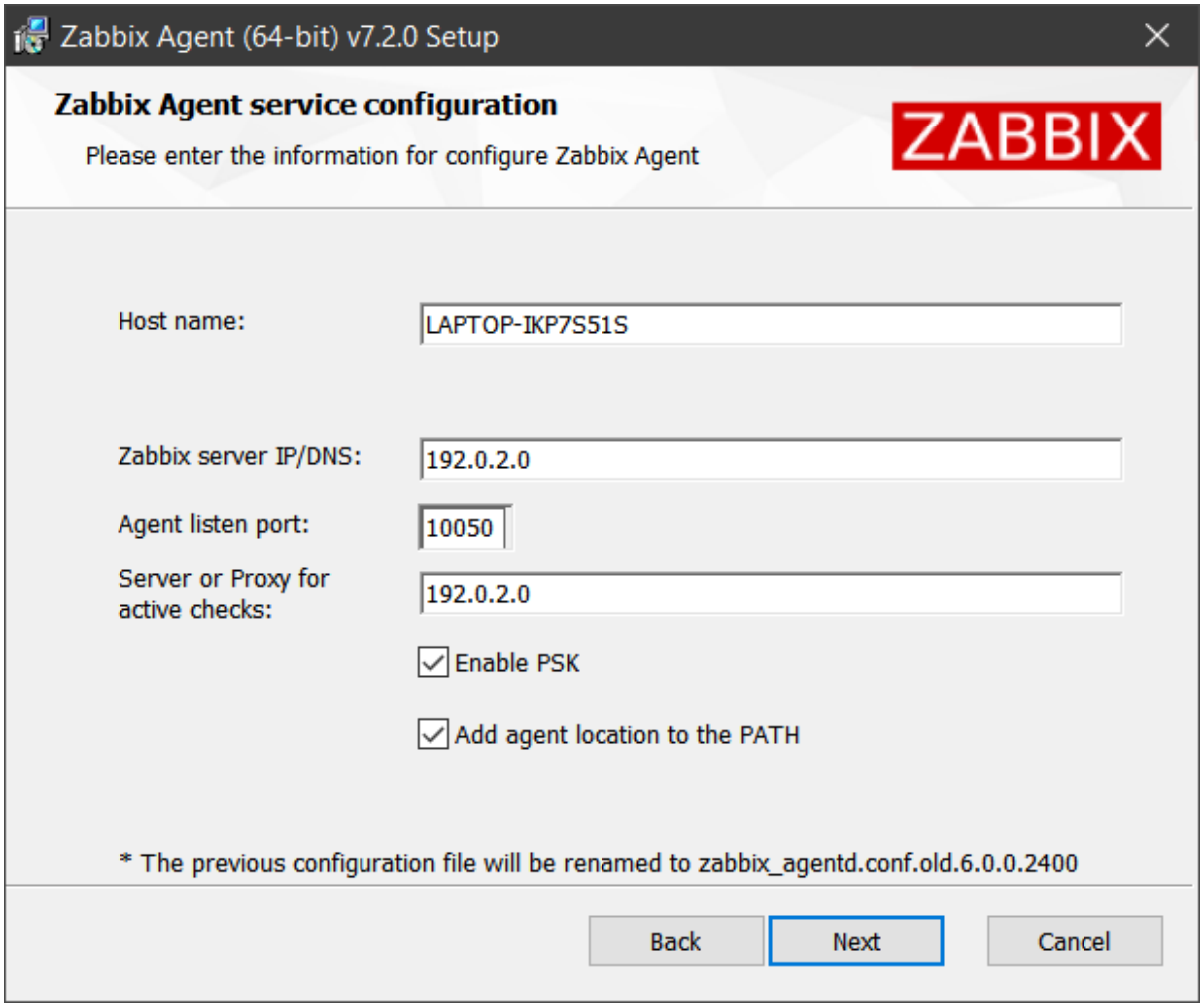
Attention:
インストーラーが提供するデフォルトのインストール場所を使用することを推奨します。必要な権限がないカスタムの場所を使用すると、インストールのセキュリティが損なわれる可能性があります。



4. 以下のパラメータを設定します。これらの値は Zabbix agent の設定ファイルに設定されます。

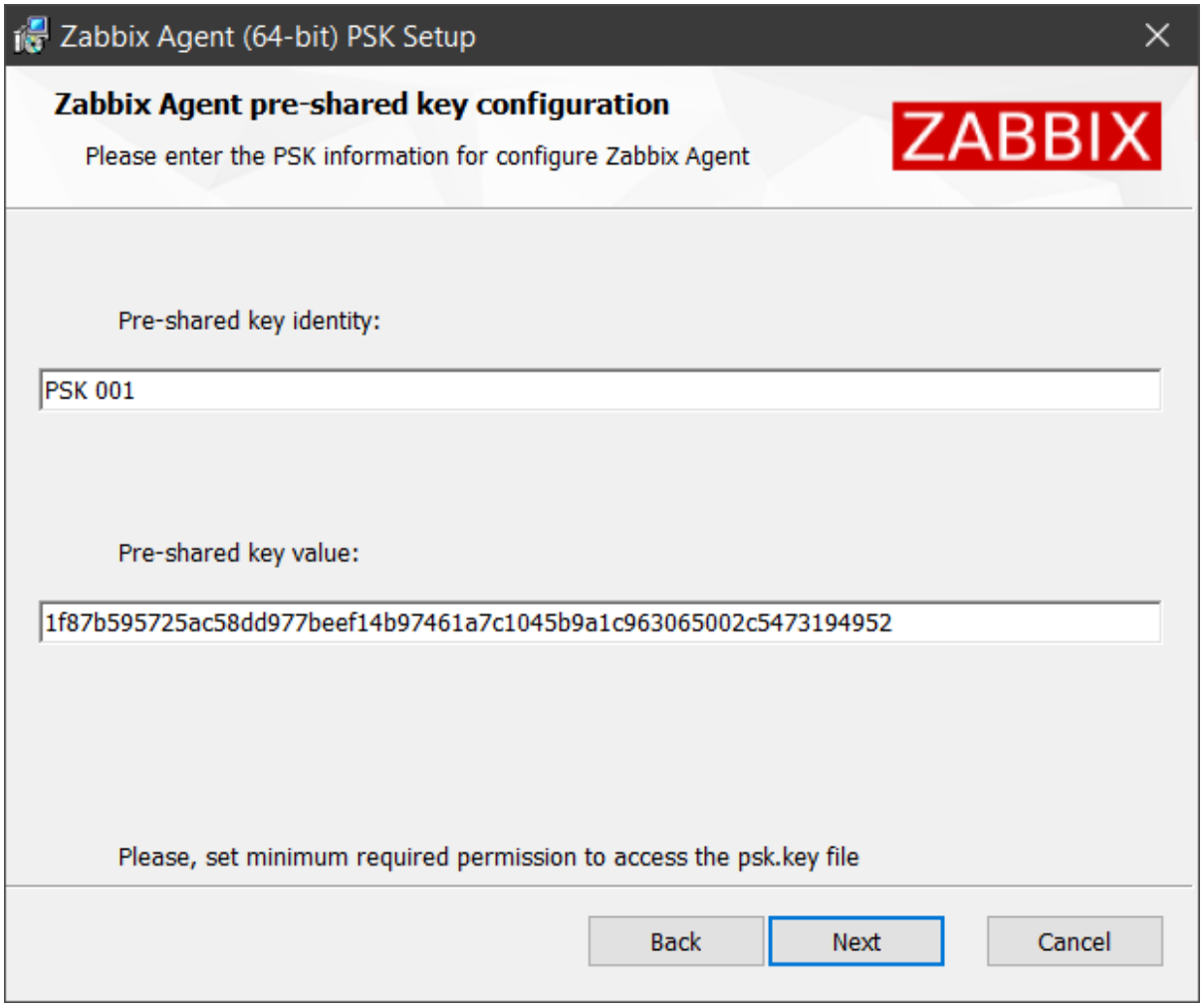
パラメータ	説明
ホスト名	Zabbix agent をインストールするマシンのホスト名。Hostnameパラメータを設定します。
Zabbix サーバー IP/DNS	カンマ区切りの IP アドレス（CIDR 表記も可）または Zabbix サーバーまたは Zabbix プロキシの DNS 名のリスト。このパラメータは必須です。Serverパラメータを設定します。
エージェントリッスンポート	サーバーからの接続を受け付けるポート。ListenPortパラメータを設定します。
アクティブチェック用のサーバーまたはプロキシ	アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ構成。サーバー/プロキシのアドレスは IP アドレスまたは DNS 名、オプションでコロン区切りのポート番号。ServerActiveパラメータを設定します。
PSK を有効にする	チェックボックスをオンにすると、事前共有鍵を使用したTLS サポートが有効になります。TLSConnectおよびTLSAcceptパラメータを `psk` に設定します。
エージェントの場所を *PATH* に追加	チェックボックスをオンにすると、Zabbix agent の場所がシステムの PATH 変数に追加されます。

Note:
既存の Zabbix agent が検出された場合、その設定ファイルのパラメータ値が表示されます。また、インストール時に既存の設定ファイルはリネームされ、新しい設定ファイルが作成されます。



5. 前の手順で *PSK* を有効にするチェックボックスをオンにした場合は、PSK パラメータを設定します。これらのパラメータも Zabbix agent の設定ファイルに設定されます。

パラメータ	説明
事前共有鍵 *ID*	事前共有鍵の ID 文字列。TLSPSKIdentityパラメータを設定します。
事前共有鍵値	事前共有鍵の文字列値。キーを含む psk.key ファイルを作成し、TLSPSKFileパラメータをキーの場所（デフォルト: `C:\Program Files\Zabbix Agent\psk.key`）に設定します。事前共有鍵ファイルへのアクセスを制限し、Zabbix agent（または agent を実行するユーザー）のみが読み取れるようにファイルのセキュリティ設定を調整することを推奨します。

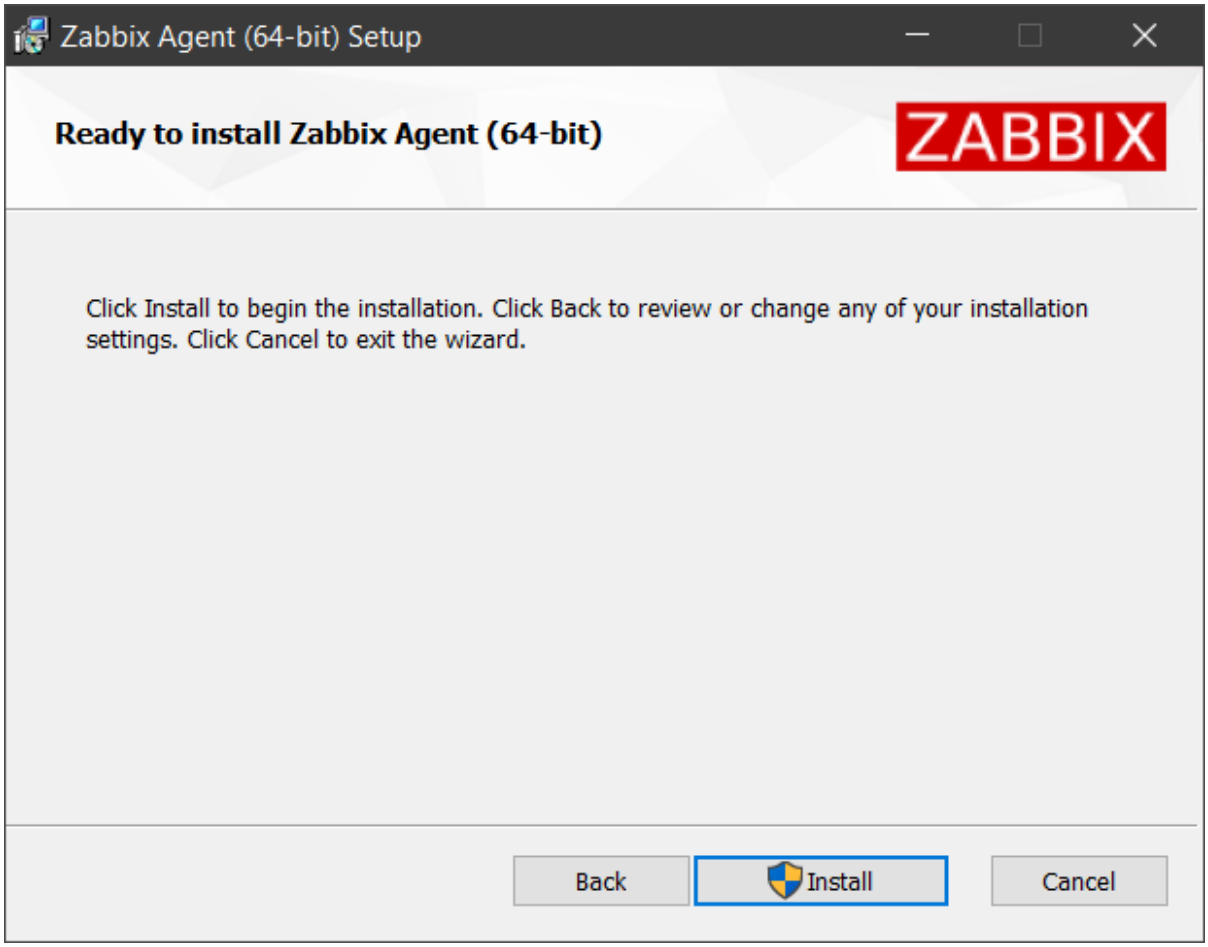


6. インストールをクリックしてインストールを開始します。

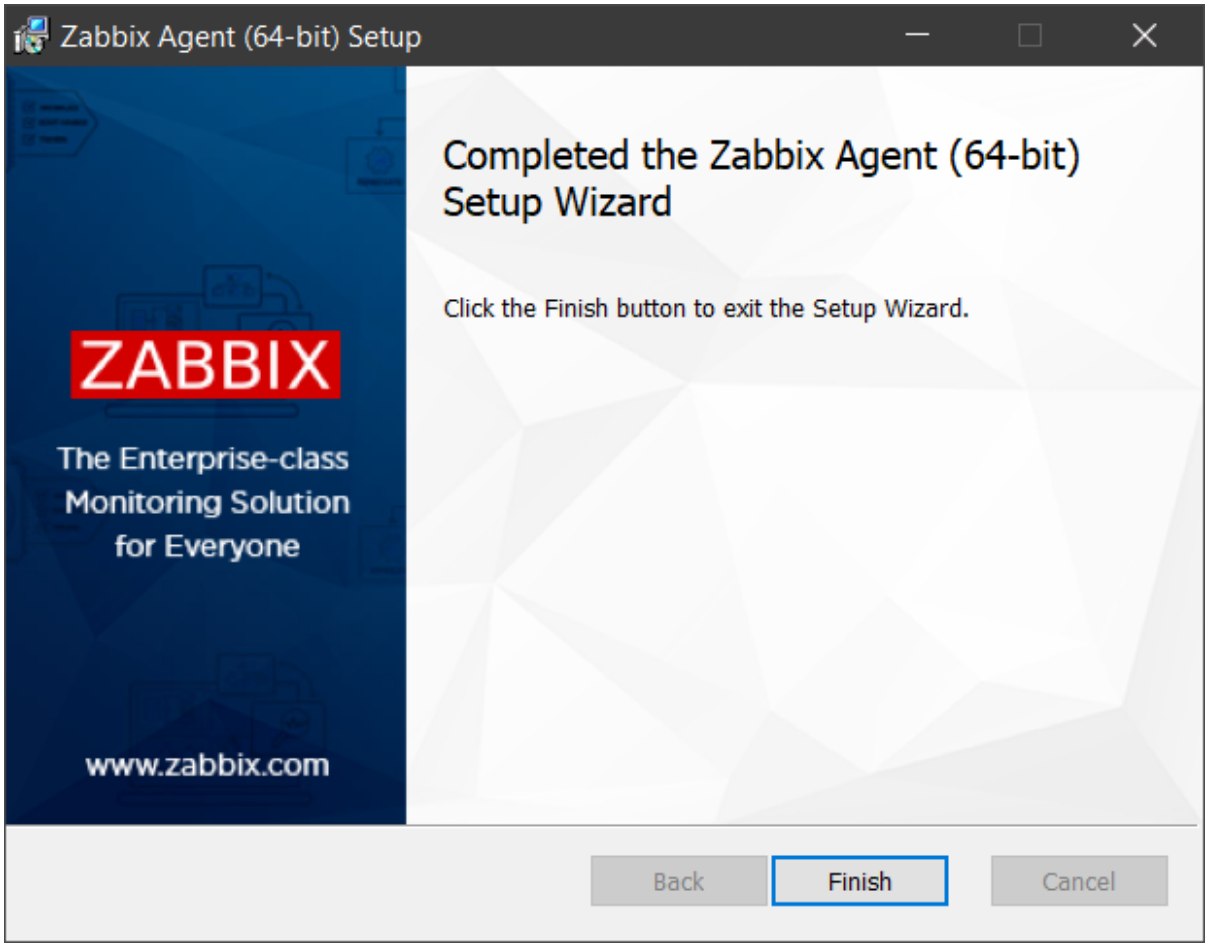
選択したすべての Zabbix コンポーネントと Zabbix agent の設定ファイルが、指定した場所（デフォルト: `C:\Program Files\Zabbix Agent`）にインストールされます。Zabbix agent 2 の場合も同様ですが、組み込みプラグイン用の追加設定ファイルが `zabbix_agent2.d\plugins.d` サブフォルダにインストールされます。

さらに、zabbix_agentd.exe（または zabbix_agent2.exe）は、遅延自動起動（または Windows Vista/Server 2008 以前の Windows バージョンでは自動起動）で Windows サービスとしてセットアップされます。

インストール中に異なるバージョンの Zabbix agent が実行されている場合、アプリケーションを閉じて再起動を試みるか、開いたままにして再起動が必要かを選択するよう求められます。



7. 完了ボタンをクリックしてセットアップウィザードを終了します。



コマンドラインからのインストール

Zabbix エージェントは、msiexecで MSI インストーラーを実行することで、コマンドラインからインストールできます。例えば:

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

この方法では、パラメータを使用して無人インストールやカスタム構成が可能です。

サポートされているパラメータ

Zabbix エージェント MSI インストーラーパッケージは、Zabbix エージェントおよび Zabbix エージェント 2 の両方で以下のパラメータをサポートしています。

Note:
Zabbix エージェント/エージェント 2 のパラメータは、インストール時に設定ファイルに設定されます。パラメータ名をクリックすると、Zabbix エージェント (Windows)ページで詳細な説明と設定例を表示できます。Zabbix エージェント 2 については、Zabbix エージェント 2(Windows)ページを参照してください。

パラメータ	説明
ADDDEFAULT	デフォルト設定でインストールするコンポーネントのカンマ区切りリスト。詳細はADDDEFAULT プロパティを参照。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDDEFAULT=AgentProgram,GetProgram
ADDLOCAL	ローカルにインストールするコンポーネントのカンマ区切りリスト。詳細はADDLOCAL プロパティを参照。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDLOCAL=AgentProgram,SenderProgram
ALLOWDENYKEY	Zabbix エージェントチェックの制限のための AllowKey または DenyKey パラメータのセミコロン区切りリスト。必要に応じてバックスラッシュでデリミタ (\;) をエスケープしてください。エージェント設定ファイルのAllowKeyおよびDenyKeyパラメータを設定します。 例: ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"
CONF	Zabbix エージェント用のテンプレート設定ファイルへのフルパス名。インストール時、このファイルがエージェントの設定ファイルになります。ファイルには少なくともServerおよびLogFileパラメータを含める必要があります。 例: CONF="C:\full\path\to\example.conf"
DONOTSTART	MSI インストーラーが Zabbix エージェントサービスを開始しないようにするには DONOTSTART=1 を使用します。
ENABLEPATH	Zabbix エージェントの場所をシステム PATH 変数に追加するには ENABLEPATH=1 を使用します。
ENABLEPERSISTENTBUFFER	Zabbix エージェント 2 のみ。アクティブアイテム用のローカル永続ストレージの使用を有効にします。
HOSTINTERFACE	ホストインターフェースを定義するオプションパラメータ。
HOSTMETADATA	ホストメタデータを定義するオプションパラメータ。
HOSTMETADATAITEM	ホストメタデータ取得に使用するアイテムを定義するオプションパラメータ。
HOSTNAME	ホスト名を定義するオプションパラメータ。
HOSTNAMEITEM	ホスト名取得に使用するアイテムを定義するオプションパラメータ。
INCLUDE	Zabbix エージェント設定ファイルに含める個別ファイルまたはディレクトリ内のすべてのファイルのセミコロン区切りリスト。
INSTALLFOLDER	Zabbix コンポーネントおよび Zabbix エージェント設定ファイルをインストールするフォルダへのフルパス名。Zabbix エージェント 2 の場合、組み込みプラグイン用の追加設定ファイルが zabbix_agent2.d\plugins.d サブフォルダにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent"
LISTENIP	エージェントがリッスンする IP アドレスのカンマ区切りリスト。
LISTENPORT	サーバーからの接続を受け付けるポート。
LOGFILE	Zabbix エージェントのログファイル名。
LOGTYPE	ログ出力のタイプ。
NONMSICONFNAME	Zabbix エージェント用のカスタム設定ファイルへのフルパス名。インストール時、このファイルに存在する有効なエージェント設定パラメータ (この表に記載されているものに限る) が新規作成されるエージェント設定ファイルに書き込まれます。ファイルには少なくともServerパラメータを含める必要があります。 例: NONMSICONFNAME="C:\full\path\to\example.conf"
PERSISTENTBUFFERFILE	Zabbix エージェント 2 のみ。Zabbix エージェント 2 が SQLite データベースを保持するファイル。
PERSISTENTBUFFERPERIOD	Zabbix エージェント 2 のみ。サーバーまたはプロキシに接続できない場合にデータを保存する期間。

パラメータ	説明
SERVER	Zabbix サーバーまたは Zabbix プロキシの IP アドレス (カンマ区切り、CIDR 表記可) または DNS 名のリスト。このパラメータは必須です (STARTAGENTS が 0 の場合を除く)。
SERVERACTIVE	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスタ設定。
SKIP	Zabbix エージェント用の Windows ファイアウォール例外ルールを MSI インストーラーが追加しないようにするには `SKIP=fw` を使用します。
SOURCEIP	Zabbix サーバーまたは Zabbix プロキシへの送信接続、または一部のアイテム (web.page.get, net.tcp.port など) 実行時の接続に使用する送信元 IP アドレス。
STARTAGENTS	パッシブチェックを処理する zabbix_agentd の事前フォークインスタンス数。 0 に設定するとパッシブチェックが無効になり、エージェントは TCP ポートでリッスンしません。
STARTUPTYPE	Zabbix エージェントサービスの起動タイプ。指定可能な値: **automatic** - Windows 起動時に自動的にサービスを開始; **delayed** - *(デフォルト)* 自動開始サービスの起動完了後にサービスの開始を遅延 (Windows Vista/Server 2008 以降で利用可能); **manual** - サービスを手動 (ユーザーまたはアプリケーション) で開始; **disabled** - サービスを無効化し、ユーザーやアプリケーションが開始できないようにする。 例: `STARTUPTYPE=disabled`
STATUSPORT	Zabbix エージェント 2 のみ。設定すると、エージェントはこのポートで HTTP ステータスリクエスト (http://localhost:<port>/status) を受け付けます。
TIMEOUT	Zabbix プロキシまたはサーバーへの接続確立およびデータ交換の待機時間 (秒単位)。
TLSACCEPT	受け付ける着信接続 (パッシブチェック用)。`psk` に設定すると、TLSCONNECT も `psk` に設定されます (別途指定がない限り)。
TLSCAFILE	ピア証明書検証用の最上位 CA 証明書を含むファイルへのフルパス名。
TLSCERTFILE	エージェント証明書または証明書チェーンを含むファイルへのフルパス名。
TLSCONNECT	Zabbix サーバーまたはプロキシへの接続方法 (アクティブチェック用)。`psk` に設定すると、TLSACCEPT も `psk` に設定されます (別途指定がない限り)。
TLSCRLFILE	失効証明書を含むファイルへのフルパス名。
TLSKEYFILE	Zabbix エージェントの秘密鍵を含むファイルへのフルパス名。
TLSPSKFILE	Zabbix エージェントの事前共有鍵を含むファイルへのフルパス名。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値が TLSPSKFILE で指定されたファイルに書き込まれます。推奨: 事前共有鍵ファイルへのアクセスは、Zabbix エージェント (またはエージェントを実行するユーザー) のみが読み取れるようにファイルのセキュリティ設定を調整してください。
TLSPSKIDENTITY	事前共有鍵の識別文字列。
TLSPSKVALUE	事前共有鍵の文字列値。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値が TLSPSKFILE で指定されたファイルに書き込まれます。 例: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	許可するサーバー (プロキシ) 証明書の発行者。
TLSSERVERCERTSUBJECT	許可するサーバー (プロキシ) 証明書のサブジェクト。
UNSAFEUSERPARAMETERS	ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。

例

次の例では、カスタム設定で Zabbix エージェントをインストールします。また、事前共有鍵を使用したTLS サポートも有効にします。

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.0^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

次の例では、Zabbix エージェントの新しいバージョンをインストールし、テンプレート設定ファイル(`CONF="C:\agent-template.conf"`)を使用します。インストール中、このファイルがエージェントの設定ファイルになります。古い設定ファイルからパラメータを継承するには、パラメータプレースホルダー（例：`[AllowDenyKey]`）を使用します。

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

#### agent-template.conf の例:
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

または、カスタム設定ファイル（NONMSICONFNAME="C:\agent-custom.conf"）を使用することもできます。インストール中、このファイルに存在する有効なエージェント設定パラメータ（上記の表に記載されているものに限定）は、新しく作成されたエージェント設定ファイルに書き込まれます。既存のエージェント設定を保持するには、保持するパラメータを定義します。

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

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

Zabbix agent 2 ロード可能なプラグイン

Zabbix agent 2 のロード可能なプラグインは、64 ビットの MSI インストーラーパッケージを使用して Windows にインストールできます。パッケージはダウンロード可能です。

MSI インストールの最小 OS 要件は、Windows 10 (64 ビット) または Windows Server 2016 です。

Attention:
プラグインをインストールする前に、README ファイルを確認してください。特定の要件やインストール手順が記載されている場合があります。

Zabbix agent/agent2 と同様に、ロード可能なプラグインもセットアップウィザードまたはコマンドラインを使用してインストールできます。

セットアップウィザードからのインストール

1. ダウンロードした MSI ファイルをダブルクリックしてインストールを開始します。

2. エンドユーザーライセンス契約に同意します。

3. インストールする Zabbix agent 2 のロード可能なプラグインを選択します。

Attention:
インストーラーが提供するデフォルトのインストール場所を使用することを推奨します。必要な権限がないカスタムの場所を使用すると、インストールのセキュリティが損なわれる可能性があります。

4. インストールをクリックしてインストールを開始します。

選択したすべての Zabbix agent 2 のロード可能なプラグインが指定した場所（デフォルト: C:\Program Files\Zabbix Agent 2）にインストールされ、設定ファイルは zabbix_agent2.d サブフォルダーにインストールされます。

5. 完了ボタンをクリックしてセットアップウィザードを終了します。

コマンドラインからのインストール

Zabbix agent 2 のロード可能なプラグインは、msiexecで MSI インストーラーを実行することでコマンドラインからインストールできます。例:

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

Zabbix agent 2 のロード可能なプラグイン MSI インストーラーパッケージは、以下のパラメータをサポートしています。

パラメータ	説明
ADDDEFAULT	デフォルト構成でインストールするコンポーネントのカンマ区切りリスト。詳細はADDDEFAULT プロパティを参照してください。 指定可能な値: ALL, CephPlugin, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDDEFAULT=MongodbPlugin,PostgresqlPlugin
ADDLOCAL	ローカルにインストールするコンポーネントのカンマ区切りリスト。詳細はADDLOCAL プロパティを参照してください。 指定可能な値: ALL, CephPlugin, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDLOCAL=MongodbPlugin,MssqlPlugin
INSTALLFOLDER	Zabbix コンポーネントをインストールするフォルダーへのフルパス名。設定ファイルは zabbix_agent2.d サブフォルダーにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"

PKG からの **macOS** エージェントのインストール

概要

Zabbix エージェントは、ダウンロード可能な PKG インストーラーパッケージを使用して macOS にインストールできます。

Zabbix エージェントパッケージは、暗号化の有無にかかわらず利用できます。

エージェントのインストール

エージェントは、グラフィカルユーザーインターフェースまたはコマンドラインからインストールできます。例えば:

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

コマンドで正しい Zabbix パッケージバージョンを使用していることを確認してください。ダウンロードしたパッケージの名前と一致している必要があります。

エージェントの実行

エージェントはインストールまたは再起動後に自動的に起動します。

必要に応じて、/usr/local/etc/zabbix/zabbix_agentd.conf の設定ファイルを編集できます。

エージェントを手動で起動するには、次のコマンドを実行します:

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

エージェントを手動で停止するには:

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

アップグレード時には、既存の設定ファイルは上書きされません。代わりに新しい zabbix_agentd.conf.NEW ファイルが作成され、必要に応じて既存の設定ファイルの確認や更新に使用できます。設定ファイルを手動で変更した後は、エージェントを再起動することを忘れないでください。

トラブルシューティングとエージェントの削除

このセクションでは、Zabbix エージェントのインストールのトラブルシューティングや削除に使用できる便利なコマンドをいくつか紹介します。

Zabbix エージェントが実行中かどうかを確認します:

```
ps aux | grep zabbix_agentd
```

Zabbix エージェントがパッケージからインストールされているかどうかを確認します:

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

インストーラーパッケージからインストールされたファイルを表示します（このビューでは先頭の/は表示されません）:

```
pkgutil --only-files --files com.zabbix.pkg.ZabbixAgent
Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
usr/local/bin/zabbix_get
usr/local/bin/zabbix_sender
usr/local/etc/zabbix/zabbix_agentd/userparameter_examples.conf.NEW
```

```
usr/local/etc/zabbix/zabbix_agentd/userparameter_mysql.conf.NEW
usr/local/etc/zabbix/zabbix_agentd.conf.NEW
usr/local/sbin/zabbix_agentd
```

launchctl で起動した Zabbix エージェントを停止します:

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

インストーラーパッケージでインストールされたファイル（設定ファイルやログを含む）を削除します:

```
sudo rm -f /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
sudo rm -f /usr/local/sbin/zabbix_agentd
sudo rm -f /usr/local/bin/zabbix_get
sudo rm -f /usr/local/bin/zabbix_sender
sudo rm -rf /usr/local/etc/zabbix
sudo rm -rf /var/log/zabbix
```

Zabbix エージェントがインストールされたことを忘れさせます:

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

Windows での **Zabbix agent 2** のビルド

概要

このページでは、Windows 10 (64 ビットまたは 32 ビット) で Zabbix agent 2 をソースからビルドする方法を説明します。

64 ビットプラットフォームでは 32 ビット版と 64 ビット版の両方をビルドできますが、32 ビットプラットフォームでは 32 ビット版のみビルドできます。

Zabbix agent 2 のビルドには以下が必要です:

- MinGW ビルドツール
- Go プログラミング言語
- OpenSSL (Zabbix の暗号化機能用)
- PCRE2 (Perl Compatible Regular Expressions; Zabbix の正規表現パターンマッチング機能用)

Zabbix agent 2 は、次のいずれかの方法でビルドできます:

- vcpkg を使用—C++ パッケージマネージャーを使用して依存関係の管理を簡素化する自動化アプローチ。
- 手動ビルド—すべての依存関係をインストールしてからエージェントをコンパイルする手動アプローチ。

> **Attention:**
> ビルドプロセスを開始する前に、次の点にご注意ください:

> - コマンドの実行には、保護されたフォルダーへの書き込み権限を持つユーザーで起動したコマンドプロンプトを使用してください。ただし、OpenSSLおよびPCRE2のインストール時は MSYS2 ターミナルを使用してください。
> - すべてのソースファイルとビルドフォルダー用に `C:\Zabbix` に作業ディレクトリを作成することを推奨します。ただし、コンパイル済みコンポーネントは `C:\Zabbix\x64` (32 ビットビルドの場合は `C:\Zabbix\x86`) にインストールしてください。

vcpkg を使用した Zabbix agent 2 のビルド

このセクションでは、依存関係の管理と C++ プロジェクトとの統合を簡素化するパッケージマネージャである vcpkg を使用して Zabbix agent をビルドする手順を説明します。

1. Build Tools for Visual Studio 2022をダウンロードしてインストールします。インストール時に、vcpkg パッケージマネージャを含む *Desktop development with C++* ワークロードを選択してください。

2. Go（MSI インストーラとして利用可能）をダウンロードしてインストールします。インストール時に、インストールフォルダとして `C:\Zabbix\Go` を指定してください。

3. Microsoft Visual C ランタイムライブラリを使用するMinGW ディストリビューションをダウンロードします。例えば:

- 64 ビットビルドの場合: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- 32 ビットビルドの場合: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

その後、`C:\Zabbix\mingw64`（32 ビットビルドの場合は `C:\Zabbix\mingw32`）に展開します。

4. vcpkg を初期化し、Zabbix agent 2 のビルドに必要な依存関係をインストールします（この処理には時間がかかる場合があります）:

```
cd C:\Zabbix

set PATH=%PATH%;"C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\vcpkg"
vcpkg new --application
vcpkg add port pcre2
vcpkg add port libiconv
vcpkg add port openssl

#### 64 ビットビルドの場合:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### 32 ビットビルドの場合:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
vcpkg install --triplet x86-mingw-static --x-install-root=x86
```

5. Zabbix ソースアーカイブをダウンロードし、C:\Zabbix\zabbix-8.0.0 に展開します。

6. Zabbix ビルドディレクトリ（C:\Zabbix\zabbix-8.0.0\build\mingw）に移動し、以下の build.bat スクリプトを作成します。

- 64 ビットビルドの場合:

```
:: 現在のセッションのシステム`PATH`変数に MinGW と Go を追加:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: vcpkg インストールパスを設定:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Crypt32 ライブラリ用のリンカフラグを設定:
SET CGO_LDFLAGS="-lCrypt32"

:: ビルドプロセスを実行:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- 32 ビットビルドの場合:

```
:: 現在のセッションのシステム`PATH`変数に MinGW と Go を追加:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: vcpkg インストールパスを設定:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Crypt32 ライブラリ用のリンカフラグを設定:
SET CGO_LDFLAGS="-lCrypt32"

:: ビルドプロセスを実行:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. スクリプトを実行して Zabbix agent 2 をコンパイルします:

```
build.bat
```

コンパイル後、Zabbix agent 2 のバイナリは C:\Zabbix\zabbix-8.0.0\bin\win64（64 ビットビルドの場合）または C:\Zabbix\zabbix-8.0.0\bin\win32（32 ビットビルドの場合）に配置されます。Zabbix agent 2 の設定ファイルは C:\Zabbix\zabbix-8.0.0\src\go\conf にあります。

エージェントを実行するには、zabbix_agent2.exe とその設定ファイルを専用フォルダ（例: C:\Zabbix\agent2）にコピーし、エージェントを実行します:

```
mkdir C:\Zabbix\agent2

#### 64 ビットビルドの場合:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### 32 ビットビルドの場合:
copy C:\Zabbix\zabbix-8.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

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

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

必要に応じて、Zabbix agent 2 のロード可能プラグインのコンパイルを続行してください。

Zabbix agent 2 ロード可能プラグインのコンパイル

1. Zabbix agent 2 のバージョンに一致するZabbix プラグインソース(例：zabbix-agent2-plugin-postgresql-8.0.0.tar.gz)をダウンロードし、C:\Zabbix に展開します。

> **Attention:**
> プラグインをインストールする前に、README ファイルを確認してください。特定の要件やインストール手順が記載されている場合があります。

2. 展開したプラグインディレクトリに移動し、プラグインをコンパイルします:

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

#### 64 ビットビルドの場合:
mingw32-make ARCH=AMD64

#### 32 ビットビルドの場合:
mingw32-make ARCH=x86
```

コンパイル後、Zabbix agent 2 プラグインのバイナリとその設定ファイルは同じプラグインディレクトリに配置されます。

プラグインの実行ファイルは、Zabbix agent 2 からロード可能であれば、どこに配置してもかまいません。プラグインの設定ファイル（例：PostgreSQLプラグインの場合は postgresql.conf）で、プラグインバイナリへのパスを指定します:

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

プラグイン設定ファイルへのパスは、Zabbix agent 2 の設定ファイルのIncludeパラメータで指定する必要があります:

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

プラグインの設定の詳細については、セットアップに進んでください。

Zabbix agent 2 の手動ビルド

> **Attention:**
> この方法は、ビルド環境を完全に制御する必要がある場合や、vcpkg の使用ができない制限された環境での利用に適しています。

このセクションでは、必要なビルドツールや依存関係のインストール、エージェントのコンパイルを含む、Zabbix agent 2 を手動でビルドする手順を説明します。

ビルドツールのセットアップ

1. MSYS2（MSI インストーラーとして利用可能）をダウンロードしてインストールします。インストール時に、インストールフォルダとして C:\Zabbix\msys64 を指定してください。

2. Go（MSI インストーラーとして利用可能。現在サポートされているGo バージョンを参照）をダウンロードしてインストールします。インストール時に、インストールフォルダとして C:\Zabbix\Go を指定してください。

3. Microsoft Visual C ランタイムライブラリを使用するMinGW ディストリビューションをダウンロードします。例えば:

- 64 ビットビルドの場合: x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z
- 32 ビットビルドの場合: i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z

その後、C:\Zabbix\mingw64（32 ビットビルドの場合は C:\Zabbix\mingw32）に展開します。

OpenSSL のインストール

> **Note:**
> TLS サポートなしで Zabbix エージェントをコンパイルする場合は、PCRE2 のインストールセクションに進んでください。

1. 管理者権限で MSYS2 MSYS ターミナルを開き、以下のコマンドを実行します。

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

2. OpenSSL のソースアーカイブをダウンロードし、C:\Zabbix\openssl-3.5.0 に展開します。

3. 展開した OpenSSL ディレクトリに移動し、以下の build.sh スクリプトを作成します。

- 64 ビットビルドの場合:

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw64/bin:$PATH"
export d="/c/Zabbix/x64/OpenSSL-Win64-350-static"

perl Configure mingw64 no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2

make
make install
```

- 32 ビットビルドの場合:

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw32/bin:$PATH"
export d="/c/Zabbix/x86/OpenSSL-Win64-350-static"

perl Configure mingw no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2-m

make
make install
```

> **Attention:**
> C:\Zabbix\x64\OpenSSL-Win64-350-static ディレクトリへの書き込み権限を管理者以外のユーザーから必ず取り消してください。そうしないと、エージェントが特権のないユーザーによって変更可能なパスから SSL 設定を読み込むことになり、潜在的なセキュリティ脆弱性につながります。

- no-shared オプションは、libcrypto.lib および libssl.lib の OpenSSL 静的ライブラリを自己完結型にし、Zabbix バイナリが外部 DLL を必要とせずに OpenSSL を含むようにします。つまり、Zabbix バイナリを OpenSSL ライブラリなしで他の Windows マシンにコピーできますが、新しい OpenSSL のバグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションなしでは、Zabbix は実行時に OpenSSL DLL に依存します。つまり、OpenSSL の更新時に Zabbix エージェントの再コンパイルが不要な場合がありますが、他のマシンにコピーする際は OpenSSL DLL もコピーする必要があります。

その他の OpenSSL 構成オプションについては、OpenSSL ドキュメントを参照してください。

4. スクリプトを実行して OpenSSL を構成・インストールします（この処理には時間がかかる場合があります）。

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

PCRE2 のインストール

1. PCRE2 のソースアーカイブをダウンロードし、C:\Zabbix\pcre2-10.45 に展開します。

2. 管理者権限で MSYS2 MSYS ターミナルを開きます。次に、展開した PCRE2 ディレクトリ内に build ディレクトリを作成し、そこに移動します:

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

3. PCRE2 を構成します:

```
#### 64 ビットビルドの場合:
export PATH="/c/Zabbix/mingw64/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX="/c/Zabbix/x64/PCRE2" ..

#### 32 ビットビルドの場合:
export PATH="/c/Zabbix/mingw32/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE
```

Note:
エラーが発生した場合は、CMake のビルドプロセスを繰り返す前に CMake キャッシュを削除することを推奨します。キャッシュ（CMakeCachecache.txt）は、展開した PCRE2 ディレクトリの build ディレクトリ内にあります。

4. PCRE2 をインストールします:

```
make install
```

Zabbix agent 2 のコンパイル

1. Zabbix ソースアーカイブをダウンロードし、C:\Zabbix\zabbix-8.0.0 に展開します。

生のソースリポジトリからソースアーカイブを生成する必要がある場合（例：カスタムパッチを適用する場合や最新のソースコードからビルドする場合）、Goがインストールされた **Linux** マシンで以下のコマンドを実行します（Zabbix agent 2 の設定に必要です）:

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

これによりソースアーカイブが作成され、Windows マシンにコピーできます。

2. 管理者権限でコマンドプロンプトを開きます。次に、Zabbix のビルドディレクトリに移動し、Zabbix agent をコンパイルします。OpenSSL および PCRE2 がインストールされているディレクトリを正しく指定してください:

- 64 ビットビルドの場合:

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

#### TLS サポートあり:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### TLS サポートなし:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- 32 ビットビルドの場合:

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

#### TLS サポートあり:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win64-350-static"

#### TLS サポートなし:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

コンパイル後、Zabbix agent 2 のバイナリは C:\Zabbix\zabbix-8.0.0\bin\win64（32 ビットビルドの場合は C:\Zabbix\zabbix-8.0.0\bin\win32）に配置されます。Zabbix agent 2 の設定ファイルは C:\Zabbix\zabbix-8.0.0\src\go\conf にあります。

エージェントを実行するには、zabbix_agent2.exe バイナリとその設定ファイルを専用のフォルダ（例：C:\Zabbix\agent2）にコピーし、エージェントを実行します:

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

必要に応じて、Zabbix agent 2 ロード可能プラグインのコンパイルを続行してください。

Windows での **Zabbix** エージェントのビルド

概要

このページでは、Windows 10 (64 ビット) で Zabbix エージェントをソースからビルドする方法を説明します。

これらの手順は、Visual Studio 2022 をサポートする Windows バージョンに適用されます。

Zabbix エージェントのビルドには以下が必要です:

- C コンパイラ (Build Tools for Visual Studio 2022 に含まれています)
- OpenSSL (Zabbix の暗号化機能用)
- PCRE2 (Perl Compatible Regular Expressions; Zabbix の正規表現パターンマッチング機能用)

Zabbix エージェントは、以下のいずれかの方法でビルドできます:

- vcpkg を使用—C++ パッケージマネージャーを使用して依存関係の管理を簡素化する自動化アプローチ。
- 手動ビルド—すべての依存関係をインストールしてからエージェントをコンパイルする手動アプローチ。

監視のニーズによっては、追加のライブラリが必要になる場合があります。詳細は要件をご覧ください。

Attention:
ビルドプロセスを開始する前に、以下にご注意ください:

- コマンドの実行には、Build Tools for Visual Studio 2022 に含まれる x64 Native Tools Command Prompt を使用し、保護されたフォルダーへの書き込み権限を持つユーザーで起動してください。
- すべてのソースファイルとビルドフォルダー用に `C:\Zabbix` に作業ディレクトリを作成することを推奨します。ただし、コンパイルされたコンポーネントは `C:\Program Files\Zabbix\x64` にインストールしてください。

vcpkg を使用した Zabbix エージェントのビルド

このセクションでは、依存関係の管理と C++ プロジェクトとの統合を簡素化するパッケージマネージャである vcpkg を使用して Zabbix エージェントをビルドする手順を説明します。

1. Build Tools for Visual Studio 2022をダウンロードしてインストールします。インストール時に、vcpkg でエージェントをビルドするために必要なツールが含まれる *Desktop development with C++* ワークロードを必ず選択してください。

- C コンパイラ（Microsoft Visual C++）
- NMake コマンドラインツール
- vcpkg パッケージマネージャ
- x64 Native Tools Command Prompt

2. vcpkg を初期化し、Zabbix エージェントのビルドに必要な依存関係をインストールします（この処理には時間がかかる場合があります）。

```
cd C:\Zabbix
vcpkg new --application
vcpkg add port pcre2
vcpkg add port openssl
vcpkg install --triplet x64-windows-static --x-install-root="C:\Program Files\Zabbix\x64"
```

3. Zabbix ソースアーカイブをダウンロードし、`C:\Zabbix\zabbix-8.0.0` に展開します。

4. Zabbix のビルドディレクトリ（`C:\Zabbix\zabbix-8.0.0\build\win32\project`）に移動し、以下の `build.bat` スクリプトを作成します。OpenSSL と PCRE2 がインストールされているディレクトリを正しく指定してください。

```
:: vcpkg のインストールパスを設定:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: ビルドプロセスを実行:
nmake -f Makefile CPU=AMD64 ^
```

```
    PCRE2INCDIR="%vcpkg%\include" ^
    PCRE2LIBDIR="%vcpkg%\lib" ^
    TLS=openssl ^
    TLSINCDIR="%vcpkg%\include" ^
    TLSLIBDIR="%vcpkg%\lib" ^
    LIBS="$(LIBS) Crypt32.lib" ^
    all
```

5. スクリプトを実行して Zabbix エージェントをコンパイルします。

```
build.bat
```

コンパイル後、Zabbix コンポーネントのバイナリは `C:\Zabbix\zabbix-8.0.0\bin\win64` に配置されます。Zabbix エージェントの設定ファイルは `C:\Zabbix\zabbix-8.0.0\conf` にあります。

エージェントを実行するには、`zabbix_agent.exe` とその設定ファイルを専用のフォルダ（例：`C:\Zabbix\agent`）にコピーし、エージェントを実行します。

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

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

Zabbix エージェントの手動ビルド

> **Attention:**
> この方法は、ビルド環境を完全に制御したいユーザーや、vcpkg の使用ができない制限された環境での利用に適しています。

このセクションでは、必要なビルドツールと依存関係（Perl、OpenSSL、PCRE2）のインストール、およびエージェントのコンパイルを含む、Zabbix エージェントを手動でビルドする手順について説明します。

ビルドツールのインストール

1. Visual Studio 2022 のビルドツールをダウンロードしてインストールします。インストール時に、*C++* によるデスクトップ開発ワークロードを選択してください。これにはエージェントを手動でビルドするために必要なツールが含まれています:

- C コンパイラ（Microsoft Visual C++）
- NMake コマンドラインツール
- x64 Native Tools コマンドプロンプト

OpenSSL のインストール

> **Note:**
> TLS サポートなしで Zabbix エージェントをコンパイルする場合は、PCRE2 のインストールセクションに進んでください。

1. Strawberry Perl（MSI インストーラーとして提供）をダウンロードしてインストールします。インストール時に、インストールフォルダーとして `C:\Zabbix\Strawberry` を必ず指定してください。

2. `Text::Template` Perl モジュールをインストールします:

```
cpanm Text::Template
```

3. Netwide Assembler（NASM; OpenSSL のコンパイルに必要）が Strawberry Perl のインストール時にコンパイルされたことを確認します:

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

NASM がコンパイルされていない場合は、手動でインストールしてください。詳細はNASM ドキュメントを参照してください。

4. OpenSSL ソースアーカイブをダウンロードし、`C:\Zabbix\openssl-3.5.0` に展開します。

5. 展開したディレクトリに移動し、OpenSSL を設定します。例:

```
cd C:\Zabbix\openssl-3.5.0
perl Configure VC-WIN64A no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1
```

Attention:
Windows で Zabbix エージェントをコンパイルする際に OpenSSL のカスタムディレクトリ（例：C:\zabbix や C:\openssl-64bit）を選択した場合は、このディレクトリへの書き込み権限を管理者以外のユーザーから必ず取り消してください。そうしないと、エージェントが特権のないユーザーによって変更可能なパスから SSL 設定を読み込むことになり、潜在的なセキュリティ脆弱性につながります。

- no-shared オプションは、libcrypto.lib および libssl.lib の OpenSSL 静的ライブラリを自己完結型にし、Zabbix バイナリが外部 DLL を必要とせずに OpenSSL を含むようにします。これにより、Zabbix バイナリを OpenSSL ライブラリなしで他の Windows マシンにコピーできますが、新しい OpenSSL バグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションを指定しない場合、Zabbix は実行時に OpenSSL DLL に依存します。これにより、OpenSSL の更新時に Zabbix エージェントの再コンパイルが不要になる場合がありますが、他のマシンにコピーする際には OpenSSL DLL もコピーする必要があります。

その他の OpenSSL 設定オプションについては、OpenSSL ドキュメントを参照してください。

6. OpenSSL をコンパイルし、テストを実行します（これには時間がかかる場合があります）：

Attention:
テストは管理者権限なしで実行してください。そうしないと、予期しない結果やセキュリティリスクにつながる可能性があります。一部のテストが失敗した場合は、OpenSSL ドキュメントを参照してトラブルシューティングを行ってください。

```
nmake
nmake test
...
All tests successful.
Files=325, Tests=3101, 822 wallclock secs ( 4.81 usr +  0.81 sys =  5.62 CPU)
Result: PASS
```

7. OpenSSL をインストールします:

```
nmake install
```

ソフトウェアコンポーネント（ライブラリ、ヘッダーファイル、ドキュメントなし）のみをインストールする場合は、nmake install_sw を使用できます。

PCRE2 のインストール

1. CMake（MSI インストーラーとして利用可能）をダウンロードしてインストールします。
 インストール時に、インストールフォルダーとして C:\Zabbix\CMake を指定し、*Add CMake to the PATH environment variable* オプションを選択してください。

2. PCRE2 のソースアーカイブをダウンロードし、C:\Zabbix\pcre2-10.45 に展開します。

3. 展開した PCRE2 ディレクトリに build ディレクトリを作成し、そこに移動します。

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

4. PCRE2 を構成します。

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

Note:
エラーが発生した場合は、CMake のビルドプロセスを再試行する前に CMake キャッシュを削除することを推奨します。
 キャッシュ（CMakeCachecache.txt）は、展開した PCRE2 ディレクトリの build ディレクトリにあります。

5. NMake を使用して PCRE2 をビルドします。

```
nmake
```

6. PCRE2 をインストールします。

```
cmake --install .
```

Zabbix エージェントのコンパイル

1. Zabbix ソースアーカイブをダウンロードし、C:\Zabbix\zabbix-8.0.0 に展開します。

生のソースリポジトリからソースアーカイブを生成する必要がある場合（例：カスタムパッチを適用する場合や最新のソースコードからビルドする場合）、**Linux** マシンで以下のコマンドを実行します。

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

これによりソースアーカイブが作成され、Windows マシンにコピーできます。

2. Zabbix のビルドディレクトリに移動し、Zabbix エージェント（または他のコンポーネント）をコンパイルします。OpenSSL および PCRE2 がインストールされているディレクトリを正しく指定してください。

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

#### TLS サポートあり:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra

#### TLS サポートなし:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra
```

コンパイル後、Zabbix コンポーネントのバイナリは `C:\Zabbix\zabbix-8.0.0\bin\win64` に配置されます。Zabbix エージェントの設定ファイルは `C:\Zabbix\zabbix-8.0.0\conf` にあります。

エージェントを実行するには、zabbix_agent.exe とその設定ファイルを専用のフォルダ（例：C:\Zabbix\agent）にコピーし、エージェントを実行します。

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

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

コンテナからのインストール

概要

このセクションでは、Docker またはDocker Compose を使用して Zabbix をデプロイする方法について説明します。

Zabbix では公式に以下を提供しています:

- 各 Zabbix コンポーネントごとに分離された Docker イメージ。これにより、移植性が高く自己完結型のコンテナとして実行できます。
- Docker で複数の Zabbix コンテナを定義・実行するための Compose ファイル。

> **Attention:**
> Zabbix 6.0 以降、決定論的トリガーはインストール時に作成する必要があります。MySQL/MariaDB でバイナリログが有効になっている場合、これにはスーパーユーザー権限、または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。変数の設定方法についてはデータベース作成スクリプト を参照してください。
> コンソールから実行した場合、この変数は一時的にしか設定されず、Docker を再起動すると解除されることに注意してください。この場合、SQL サービスは稼働させたまま、'docker compose down zabbix-server' で zabbix-server サービスのみを停止し、'docker compose up -d zabbix-server' で再起動してください。
> または、この変数を設定ファイルに記述することもできます。

ソースファイル

Docker ファイルのソースは、GitHub の Zabbix 公式リポジトリに保存されています。ここでは、最新のファイル変更を追跡したり、プロジェクトをフォークして独自のイメージを作成したりできます。

Docker

Zabbix は、さまざまな OS ベースのイメージに基づいたイメージを提供しています。特定の Zabbix コンポーネントに対応するベースオペレーティングシステムイメージの一覧については、Docker Hubの各コンポーネントの説明を参照してください。すべての Zabbix イメージは、ベースイメージが更新された場合に最新イメージを再構築するように設定されています。

インストール

Zabbix コンポーネントイメージを取得するには、次のコマンドを実行します。

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

`zabbix/zabbix-server-mysql` を必要な Docker リポジトリ名に置き換えてください。

このコマンドは、Alpine Linux OS ベースの最新の安定版 Zabbix コンポーネントバージョンを取得します。タグをリポジトリ名に追加することで、別のオペレーティングシステムベースのイメージや、特定の Zabbix メジャーまたはマイナーバージョンのイメージを取得できます。

Docker Hub で利用可能なリポジトリは以下の通りです。

コンポーネント		Docker リポジトリ
Zabbix agent		zabbix/zabbix-agent
Zabbix server		
	MySQL サポート	zabbix/zabbix-server-mysql
	PostgreSQL サポート	zabbix/zabbix-server-pgsql
Zabbix web インターフェース		
	Apache2 Web サーバー（MySQL サポート）	zabbix/zabbix-web-apache-mysql
	Apache2 Web サーバー（PostgreSQL サポート）	zabbix/zabbix-web-apache-pgsql
	Nginx Web サーバー（MySQL サポート）	zabbix/zabbix-web-nginx-mysql
	Nginx Web サーバー（PostgreSQL サポート）	zabbix/zabbix-web-nginx-pgsql
Zabbix proxy		
	SQLite3 サポート	zabbix/zabbix-proxy-sqlite3
	MySQL サポート	zabbix/zabbix-proxy-mysql
Zabbix Java gateway		zabbix/zabbix-java-gateway

Note:
SNMP トラップのサポートは、zabbix/zabbix-snmptrapsという別リポジトリで提供されています。これは Zabbix server や Zabbix proxy と連携できます。

タグ

公式 Zabbix コンポーネントイメージには、以下のタグが含まれる場合があります。

タグ	説明	例
latest	Alpine Linux イメージをベースにした Zabbix コンポーネントの最新安定版。	zabbix-agent:latest
<OS>-trunk	特定のオペレーティングシステム上で現在開発中の Zabbix バージョンの最新ナイトリービルド。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	特定のオペレーティングシステム上での Zabbix コンポーネントの最新安定版。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu	zabbix-agent:ol-latest

タグ	説明	例
<OS>-X.X-latest	特定のメジャーバージョンおよびオペレーティングシステムの Zabbix コンポーネントの最新マイナーバージョン。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu **X.X** - Zabbix のメジャーバージョン（例: *6.0*, *7.4*, *8.0*）。	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	特定のメジャーバージョンおよびオペレーティングシステムの Zabbix コンポーネントの最新マイナーバージョン。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu **X.X** - Zabbix のメジャーバージョン（例: *6.0*, *7.4*, *8.0*）。 * - Zabbix のマイナーバージョン	zabbix-agent:alpine-8.0.1

初期設定

イメージをダウンロードした後、`docker run` コマンドを実行し、必要な環境変数やマウントポイントを指定する追加の引数を付けてコンテナを起動します。いくつかの設定例を以下に示します。

Note:
Zabbix コンポーネント間の通信を有効にするために、Zabbix サーバー（トラッパー）用の 10051/TCP、Zabbix エージェント用の 10050/TCP、SNMP トラップ用の 162/UDP、Zabbix Web インターフェース用の 80/TCP などのポートがホストマシンに公開されます。Zabbix コンポーネントで使用されるデフォルトポートの一覧は、要件ページで確認できます。Zabbix サーバーおよびエージェントの場合、デフォルトポートは ZBX_LISTENPORT環境変数を設定することで変更できます。

環境変数

すべての Zabbix コンポーネントイメージは、設定を制御するための環境変数を提供します。サポートされている環境変数は、コンポーネントリポジトリに記載されています。

これらの環境変数は Zabbix の設定ファイルのオプションですが、命名方法が異なります。たとえば、`ZBX_LOGSLOWQUERIES` は、Zabbix サーバーまたは Zabbix プロキシの設定ファイルの `LogSlowQueries` と同じです。

Attention:
一部の設定オプション（例：`PIDFile` や `LogType`）は変更できません。

以下の環境変数は Docker コンポーネント固有のものであり、Zabbix の設定ファイルには存在しません。

変数	コンポーネント	デフォルト値	説明
`DB_SERVER_HOST`	サーバー プロキシ Web インターフェース	MYSQL の場合は `mysql-server` PostgreSQL の場合は `postgres-server`	MySQL または PostgreSQL サーバーの IP または DNS 名。
`DB_SERVER_PORT`	サーバー プロキシ Web インターフェース	MYSQL の場合は 3306 PostgreSQL の場合は 5432	MySQL または PostgreSQL サーバーのポート。

変数	コンポーネント	デフォルト値	説明
MYSQL_USER	サーバー プロキシ Web インターフェース	zabbix	MySQL データベースユーザー。
MYSQL_PASSWORD	サーバー プロキシ Web インターフェース	zabbix	MySQL データベースのパスワード。
MYSQL_DATABASE	サーバー プロキシ Web インターフェース	Zabbix サーバーの場合は zabbix Zabbix プロキシの場合は zabbix_proxy	Zabbix データベース名。
POSTGRES_USER	サーバー Web インターフェース	zabbix	PostgreSQL データベースユーザー。
POSTGRES_PASSWORD	サーバー Web インターフェース	zabbix	PostgreSQL データベースのパスワード。
POSTGRES_DB	サーバー Web インターフェース	Zabbix サーバーの場合は zabbix Zabbix プロキシの場合は zabbix_proxy	Zabbix データベース名。
PHP_TZ	Web インターフェース	Europe/Riga	PHP 形式のタイムゾーン。サポートされているタイムゾーンの一覧はphp.netで確認できます。
ZBX_SERVER_NAME	Web インターフェース	Zabbix Docker	Web インターフェースの縦メニューの Zabbix ロゴ下に表示される Zabbix インストール名。
ZBX_JAVAGATEWAY_ENABLE	サーバー プロキシ	false	Java 関連のチェックを収集するために Zabbix Java ゲートウェイとの通信を有効にします。
ZBX_ENABLE_SNMP_TRAPS	サーバー プロキシ	false	SNMP トラップ機能を有効にします。**zabbix-snmptraps** インスタンスと、Zabbix サーバーまたは Zabbix プロキシへの共有ボリューム*/var/lib/zabbix/snmptraps* が必要です。

ボリューム

イメージでは、以下のマウントポイントを使用してボリュームをマウントできます。

ボリューム	説明
Zabbix agent	
/etc/zabbix/zabbix_agentd.d	このボリュームに **.conf* ファイルを含めることで、UserParameter 機能を使って Zabbix agent を拡張できます
/var/lib/zabbix/modules	このボリュームに追加モジュールを配置することで、LoadModule 機能を使って Zabbix agent を拡張できます
/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。ファイル名は ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE 環境変数で指定します
Zabbix server	
/usr/lib/zabbix/alertscripts	このボリュームはカスタムアラートスクリプト用です。zabbix_server.conf の AlertScriptsPath パラメータです
/usr/lib/zabbix/externalscripts	このボリュームは外部チェックで使用されます。zabbix_server.conf の ExternalScripts パラメータです
/var/lib/zabbix/modules	このボリュームに追加モジュールを配置することで、LoadModule 機能を使って Zabbix server を拡張できます

ボリューム		説明
	/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。ファイル名は ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE 環境変数で指定します
	/var/lib/zabbix/ssl/certs	このボリュームはクライアント認証用の SSL クライアント証明書ファイルの保存場所です。zabbix_server.conf の SSLCertLocation パラメータです
	/var/lib/zabbix/ssl/keys	このボリュームはクライアント認証用の SSL 秘密鍵ファイルの保存場所です。zabbix_server.conf の SSLKeyLocation パラメータです
	/var/lib/zabbix/ssl/ssl_ca	このボリュームは SSL サーバ証明書検証用の認証局 (CA) ファイルの保存場所です。zabbix_server.conf の SSLCALocation パラメータです
	/var/lib/zabbix/snmptraps	このボリュームは snmptraps.log ファイルの保存場所です。zabbix-snmptraps コンテナと共有し、Zabbix server の新しいインスタンス作成時に volumes_from Docker オプションで継承できます。共有ボリュームを使用し、ZBX_ENABLE_SNMP_TRAPS 環境変数を'true' に切り替えることで SNMP トラップ処理機能を有効化できます
	/var/lib/zabbix/mibs	このボリュームに新しい MIB ファイルを追加できます。サブディレクトリはサポートされていません。すべての MIB は /var/lib/zabbix/mibs に配置する必要があります
Zabbix proxy		
	/usr/lib/zabbix/externalscripts	このボリュームは外部チェックで使用されます。zabbix_proxy.conf の ExternalScripts パラメータです
	/var/lib/zabbix/db_data	このボリュームはデータベースファイルを外部デバイスに保存するためのものです。SQLite3 を使用した Zabbix proxy でのみサポートされます
	/var/lib/zabbix/modules	このボリュームに追加モジュールを配置することで、LoadModule 機能を使って Zabbix server を拡張できます
	/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。ファイル名は ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE 環境変数で指定します
	/var/lib/zabbix/ssl/certs	このボリュームはクライアント認証用の SSL クライアント証明書ファイルの保存場所です。zabbix_proxy.conf の SSLCertLocation パラメータです
	/var/lib/zabbix/ssl/keys	このボリュームはクライアント認証用の SSL 秘密鍵ファイルの保存場所です。zabbix_proxy.conf の SSLKeyLocation パラメータです
	/var/lib/zabbix/ssl/ssl_ca	このボリュームは SSL サーバ証明書検証用の認証局 (CA) ファイルの保存場所です。zabbix_proxy.conf の SSLCALocation パラメータです
	/var/lib/zabbix/snmptraps	このボリュームは snmptraps.log ファイルの保存場所です。zabbix-snmptraps コンテナと共有し、Zabbix server の新しいインスタンス作成時に volumes_from Docker オプションで継承できます。共有ボリュームを使用し、ZBX_ENABLE_SNMP_TRAPS 環境変数を'true' に切り替えることで SNMP トラップ処理機能を有効化できます
	/var/lib/zabbix/mibs	このボリュームに新しい MIB ファイルを追加できます。サブディレクトリはサポートされていません。すべての MIB は /var/lib/zabbix/mibs に配置する必要があります
Apache2 Web サーバーベースの Zabbix Web インターフェース		

ボリューム		説明
	/etc/ssl/apache2	このボリュームにより、Zabbix Web インターフェースで HTTPS を有効化できます。Apache2 の SSL 接続用に準備した `ssl.crt` と `ssl.key` の 2 つのファイルを含める必要があります
Nginx Web サーバーベースの Zabbix Web インターフェース		
	/etc/ssl/nginx	このボリュームにより、Zabbix Web インターフェースで HTTPS を有効化できます。Nginx の SSL 接続用に準備した `ssl.crt`、`ssl.key`、`dhparam.pem` の 3 つのファイルを含める必要があります
Zabbix snmptraps		
	/var/lib/zabbix/snmptraps	このボリュームには受信した SNMP トラップの `snmptraps.log` ログファイルが含まれます
	/var/lib/zabbix/mibs	このボリュームに新しい MIB ファイルを追加できます。サブディレクトリはサポートされていません。すべての MIB は `/var/lib/zabbix/mibs` に配置する必要があります

詳細については、Docker Hub の Zabbix 公式リポジトリを参照してください。

例

例 **1**

この例では、MySQL データベースサポート付きの Zabbix サーバー、Nginx ウェブサーバーをベースにした Zabbix ウェブインターフェース、および Zabbix Java ゲートウェイを実行する方法を示します。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

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

2. 空の MySQL サーバーインスタンスを起動します。

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

3. Zabbix Java ゲートウェイインスタンスを起動します。

```
docker run --name zabbix-java-gateway -t \
```

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

4. Zabbix サーバーインスタンスを起動し、作成した MySQL サーバーインスタンスとリンクします。

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

5. Zabbix ウェブインターフェースを起動し、作成した MySQL サーバーおよび Zabbix サーバーインスタンスとリンクします。

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

例 **2**

この例では、PostgreSQL データベースサポート付きの Zabbix サーバー、Nginx ウェブサーバーをベースにした Zabbix ウェブインターフェース、および SNMP トラップ機能を実行する方法を示します。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

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

2. 空の PostgreSQL サーバーインスタンスを起動します。

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

3. Zabbix snmptraps インスタンスを起動します。

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

4. Zabbix サーバーインスタンスを起動し、作成した PostgreSQL サーバーインスタンスとリンクします。

```
docker run --name zabbix-server-pgsql -t \
      -e DB_SERVER_HOST="postgres-server" \
      -e POSTGRES_USER="zabbix" \
      -e POSTGRES_PASSWORD="zabbix_pwd" \
      -e POSTGRES_DB="zabbix" \
      -e ZBX_ENABLE_SNMP_TRAPS="true" \
      --network=zabbix-net \
```

```
      -p 10051:10051 \
      --volumes-from zabbix-snmptraps \
      --restart unless-stopped \
      -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

5. Zabbix ウェブインターフェースを起動し、作成した PostgreSQL サーバーおよび Zabbix サーバーインスタンスとリンクします。

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

例 **3**

この例では、MySQL データベースサポート付きの Zabbix サーバー、Nginx ウェブサーバーをベースにした Zabbix ウェブインターフェース、および Red Hat 8 上で podman を使用した Zabbix Java ゲートウェイを実行する方法を示します。

1. zabbix という名前の新しい pod を作成し、ポート（ウェブインターフェース、Zabbix サーバートラッパー）を公開します。

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

2. （オプション）zabbix pod の場所で Zabbix エージェントコンテナを起動します。

```
podman run --name zabbix-agent \
    -e ZBX_SERVER_HOST="127.0.0.1,localhost" \
    --restart=always \
    --pod=zabbix \
    -d registry.connect.redhat.com/zabbix/zabbix-agent-74:latest
```

3. ホスト上に./mysql/ディレクトリを作成し、Oracle MySQL サーバー 8.4 を起動します。

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

4. Zabbix サーバーコンテナを起動します。

```
podman run --name zabbix-server-mysql -t \
                  -e DB_SERVER_HOST="127.0.0.1" \
                  -e MYSQL_DATABASE="zabbix" \
                  -e MYSQL_USER="zabbix" \
                  -e MYSQL_PASSWORD="zabbix_pwd" \
                  -e MYSQL_ROOT_PASSWORD="root_pwd" \
                  -e ZBX_JAVAGATEWAY="127.0.0.1" \
                  --restart=always \
                  --pod=zabbix \
                  -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-74
```

5. Zabbix Java ゲートウェイコンテナを起動します。

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

6. Zabbix ウェブインターフェースコンテナを起動します。

```
podman run --name zabbix-web-mysql -t \
                  -e ZBX_SERVER_HOST="127.0.0.1" \
                  -e DB_SERVER_HOST="127.0.0.1" \
                  -e MYSQL_DATABASE="zabbix" \
                  -e MYSQL_USER="zabbix" \
                  -e MYSQL_PASSWORD="zabbix_pwd" \
                  -e MYSQL_ROOT_PASSWORD="root_pwd" \
                  --restart=always \
                  --pod=zabbix \
                  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-74
```

Note:
Pod zabbix は、zabbix-web-mysql コンテナの 8080/TCP からホストマシンの 80/TCP ポート（HTTP）を公開します。

Docker Compose

また、Zabbix は Docker Compose プラグインを使用してインストールすることもできます。複数コンテナの Zabbix コンポーネントを定義・実行するための Compose ファイルは、公式のZabbix Docker リポジトリで提供されています。

Attention:
公式の Zabbix Compose ファイルは Docker Compose のバージョン 3 をサポートしています。

これらの Compose ファイルは例として追加されています。内容は多岐にわたります。例えば、MySQL と SQLite3 の両方をサポートするプロキシが含まれています。

Zabbix が提供する Docker Compose ファイルを取得するには、リポジトリをクローンします:

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

必要なバージョンに切り替えます:

```
git checkout 8.0
```

Compose 設定ファイルを指定して、コンテナを作成・起動します:

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

上記コマンドの docker-compose_v3_alpine_mysql_latest.yaml は、必要な設定ファイルに置き換えてください。

利用可能なオプションは以下の通りです:

ファイル名	説明
docker-compose_v3_alpine_mysql_latest.yaml	Alpine Linux 上で MySQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。
docker-compose_v3_alpine_mysql_local.yaml	Alpine Linux 上で MySQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。
docker-compose_v3_alpine_pgsql_latest.yaml	Alpine Linux 上で PostgreSQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。
docker-compose_v3_alpine_pgsql_local.yaml	Alpine Linux 上で PostgreSQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。
docker-compose_v3_ol_mysql_latest.yaml	Oracle Linux 上で MySQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。
docker-compose_v3_ol_mysql_local.yaml	Oracle Linux 上で MySQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。
docker-compose_v3_ol_pgsql_latest.yaml	Oracle Linux 上で PostgreSQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。
docker-compose_v3_ol_pgsql_local.yaml	Oracle Linux 上で PostgreSQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_mysql_latest.yaml	Ubuntu 22.04 上で MySQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。
docker-compose_v3_ubuntu_mysql_local.yaml	Ubuntu 22.04 上で MySQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_pgsql_latest.yaml	Ubuntu 22.04 上で PostgreSQL データベースをサポートする Zabbix 8.0 コンポーネントの最新バージョンを実行します。

ファイル名	説明
docker-compose_v3_ubuntu_pgsql_local.yaml	Ubuntu 22.04 上で PostgreSQL データベースをサポートする Zabbix 8.0 の最新バージョンをローカルでビルドし、Zabbix コンポーネントを実行します。

ストレージ

Compose ファイルは、ホストマシン上のローカルストレージをサポートするように設定されています。Docker Compose は、compose ファイルを使用して Zabbix コンポーネントを実行すると、compose ファイルのあるフォルダに zbx_env ディレクトリを作成します。このディレクトリには、Volumesセクションで説明されているのと同じ構造と、データベースストレージ用のディレクトリが含まれます。

また、/etc/localtime および/etc/timezone ファイル用に読み取り専用モードのボリュームもあります。

環境変数

変数ファイルは、.env_< コンポーネントの種類 > という名前構造を持ち、*env_vars* ディレクトリに配置されています。
 変数の命名や選択可能な内容の詳細については、環境変数を参照してください。

例

例 **1**

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

このコマンドは、各 Zabbix コンポーネントの最新の Zabbix 8.0 イメージをダウンロードし、デタッチモードで実行します。

> **Attention:**
> compose ファイル付きの github.com 公式 Zabbix リポジトリから.env_<type of component> ファイルをダウンロードするのを忘れないでください。

例 **2**

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

このコマンドは、ベースイメージ Ubuntu 24.04（noble）をダウンロードし、Zabbix 8.0 コンポーネントをローカルでビルドしてデタッチモードで実行します。

ソースからのインストール

Zabbix の最新バージョンは、ソースからコンパイルすることで入手できます。Zabbix ソースコードの取得も参照してください。

ソースから Zabbix をインストールするためのステップバイステップのチュートリアルをここに示します。

1 Zabbix デーモンのインストール

1 ソースアーカイブのダウンロード

Zabbix ダウンロードページにアクセスし、ソースアーカイブをダウンロードします。ダウンロードしたら、以下のコマンドを実行してソースを展開します。

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

> **Note:**
> コマンドには正しい Zabbix バージョンを入力してください。ダウンロードしたアーカイブ名と一致する必要があります。

2 ユーザーアカウントの作成

すべての Zabbix デーモンプロセスは、特権のないシステムユーザーで実行されます。Zabbix デーモンが特権のないユーザーアカウントから起動された場合、そのユーザーで実行され続けます。

デフォルトの設定では、デーモンが root として起動された場合、zabbix ユーザーアカウントに切り替わります。このアカウントは存在している必要があります。zabbix システムユーザーとグループを作成するには、以下のコマンドを実行してください。

RedHat 系システムの場合:

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

Debian 系システムの場合:

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

Zabbix フロントエンド用に別のユーザーアカウントを作成する必要はありません。

セキュリティ推奨

Zabbix のサーバーとエージェントが同じマシンで動作する場合は、別々のユーザーアカウントで実行することを推奨します。両方を同じユーザーで実行すると、エージェントがサーバーの設定ファイルにアクセスできるようになり、データベースのパスワードなどの機密情報が Zabbix の管理者レベルのユーザーに漏洩する可能性があります。

Attention:
Zabbix を root、bin、または特別な権限を持つ他のアカウントで実行することは、セキュリティリスクとなります。

ホームディレクトリ（オプション）

Zabbix プロセスはホームディレクトリを必要としないため、通常は作成を推奨しません。ただし、$HOME/.my.cnf に MySQL の認証情報を保存するなど、ホームディレクトリが必要な機能を利用する場合は、以下のコマンドで作成できます。

RedHat 系システムの場合:

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

Debian 系システムの場合:

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

3 Zabbix データベースの作成

Zabbix のサーバーおよびプロキシデーモン、ならびに Zabbix フロントエンドにはデータベースが必要です。Zabbixエージェントの実行には必要ありません。

データベーススキーマの作成とデータセットの挿入には、SQLスクリプトが提供されています。Zabbix プロキシのデータベースにはスキーマのみが必要ですが、Zabbix サーバーのデータベースにはスキーマに加えてデータセットも必要です。

Zabbix データベースを作成したら、Zabbix のコンパイルの次の手順に進みます。

4 ソースの設定

Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントをビルドするには、GNU 拡張付きの C99 が必要です。このバージョンは、CFLAGS="-std=gnu99" を設定することで明示的に指定できます。

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

Note:
Zabbix Git リポジトリからインストールする場合は、最初に次を実行する必要があります。

```
./bootstrap.sh
```

Zabbix サーバーまたはプロキシのソースを設定する場合は、使用するデータベースタイプを指定する必要があります。サーバーまたはプロキシプロセスで一度にコンパイルできるデータベースタイプは 1 つだけです。

サポートされているすべての設定オプションを表示するには、展開した Zabbix ソースディレクトリ内で次を実行します。

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

Zabbix サーバーとエージェントのソースを設定するには、次のように実行できます。

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

Zabbix サーバー（PostgreSQL など）用のソースを設定するには、次のように実行できます。

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

Zabbix プロキシ（SQLite など）用のソースを設定するには、次のように実行できます。

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

Zabbix エージェント用のソースを設定するには、次のように実行できます。

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

または、Zabbix agent 2 の場合は次のようにします。

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

> **Note:**
> Zabbix agent 2 をビルドするには、サポートされている Go バージョンが必要です。

コンパイルオプションに関する注意事項:

- `--enable-agent` - Zabbix エージェント、およびZabbix getやZabbix senderコマンドラインユーティリティもコンパイルします。
- `--with-libcurl` - 仮想マシンの監視、SMTP 認証、および `web.page.*` Zabbix エージェントアイテムに必要です。詳細は要件（libcurl）も参照してください。
- `--with-libxml2` - 仮想マシンの監視に必要です。
- `--with-libpcre2[=DIR]` - Zabbix は常に PCRE2 ライブラリでコンパイルされます。このオプションはカスタムの PCRE2 インストールパスを指定する場合のみ使用します。
- `--with-mysql=/path/to/mysql_config` - 特定の MySQL クライアントライブラリ設定へのパスを指定します。複数のバージョンの MySQL や MariaDB がインストールされている場合に便利です。
- `--enable-static` - ライブラリを静的にリンクします（Solarisではサポートされていません）。必要なライブラリがないシステムにコンパイル済みバイナリを配布する場合に使用します。Zabbix サーバーのビルド時には推奨されません。サーバーを静的にビルドするには、すべての外部ライブラリの静的バージョンが必要です。configure スクリプトはこれを自動的にチェックしません。
- `--with-stacksize=<value>` - スレッドごとのスタックサイズをキロバイト単位で設定します(例：`--with-stacksize=512`)。Zabbix がスタックオーバーフロー（例：前処理時）でクラッシュまたはフリーズする場合（デフォルトのスレッドスタック制限が低いシステムなど）、この値を増やすことができます。

> **Attention:**
> `./configure` がライブラリの不足やその他の問題で失敗した場合は、詳細なエラー情報を `config.log` ファイルで確認してください。
> たとえば、libssl が不足している場合、直前のエラーメッセージは誤解を招く場合があります。
> `checking for main in -lmysqlclient... no`
> `configure: error: Not found mysqlclient library`
> この場合、`config.log` には実際の原因が記載されています。
> `/usr/bin/ld: cannot find -lssl`
> `/usr/bin/ld: cannot find -lcrypto`

関連情報:

- 暗号化サポート付き Zabbix のコンパイル
- 既知のコンパイル問題

5 すべてを作成してインストールする

> **Note:**
> Zabbix Git リポジトリからインストールする場合は、最初に次を実行する必要があります:
>
> `$ make dbschema`

`make install`

このステップは、十分な権限を持つユーザー（通常は'root'、または `sudo` を使用して）として実行する必要があります。

`make install` を実行すると、デフォルトでデーモンバイナリ（zabbix_server、zabbix_agentd、zabbix_proxy）は/usr/local/sbin に、クライアントバイナリ（zabbix_get、zabbix_sender）は/usr/local/bin にインストールされます。

> **Note:**
> /usr/local 以外の場所を指定するには、前のソース構成ステップで--prefix キーを使用します。例えば--prefix=/home/zabbix のように指定します。この場合、デーモンバイナリは <prefix>/sbin、ユーティリティは <prefix>/bin にインストールされます。Man ページは <prefix>/share にインストールされます。

6 設定ファイルの確認と編集

- Zabbix エージェントの設定ファイル **/usr/local/etc/zabbix_agentd.conf** を編集します。

このファイルは、zabbix_agentd をインストールしたすべてのホストで設定する必要があります。

ファイル内で Zabbix サーバーの **IP** アドレスを指定する必要があります。他のホストからの接続は拒否されます。

- Zabbix サーバーの設定ファイル **/usr/local/etc/zabbix_server.conf** を編集します。

データベース名、ユーザー、およびパスワード（使用している場合）を指定する必要があります。

他のパラメータは、小規模なインストール（監視対象ホストが 10 台まで）の場合はデフォルトのままで問題ありません。ただし、Zabbix サーバー（またはプロキシ）のパフォーマンスを最大化したい場合は、デフォルトのパラメータを変更する必要があります。

- Zabbix プロキシをインストールした場合は、プロキシの設定ファイル **/usr/local/etc/zabbix_proxy.conf** を編集します。

サーバーの IP アドレスとプロキシのホスト名（サーバーで認識されている必要があります）、およびデータベース名、ユーザー、パスワード（使用している場合）を指定する必要があります。

> **Note:**
> SQLite を使用する場合は、データベースファイルへのフルパスを指定する必要があります。DB ユーザーとパスワードは不要です。

7 デーモンの起動

サーバー側で zabbix_server を実行します。

```
zabbix_server
```

> **Note:**
> システムが 36MB（またはそれ以上）の共有メモリの割り当てを許可していることを確認してください。そうでない場合、サーバーが起動せず、サーバーログファイルに「Cannot allocate shared memory for <type of cache>.」と表示されることがあります。これは FreeBSD や Solaris 8 で発生する可能性があります。

監視対象のすべてのマシンで zabbix_agentd を実行します。

```
zabbix_agentd
```

> **Note:**
> システムが 2MB の共有メモリの割り当てを許可していることを確認してください。そうでない場合、エージェントが起動せず、エージェントログファイルに「Cannot allocate shared memory for collector.」と表示されることがあります。これは Solaris 8 で発生する可能性があります。

Zabbix プロキシをインストールした場合は、zabbix_proxy を実行します。

```
zabbix_proxy
```

2 Zabbix Web インターフェースのインストール

PHP ファイルのコピー

Zabbix フロントエンドは PHP で書かれているため、実行するには PHP をサポートする Web サーバーが必要です。インストールは、ui ディレクトリから Web サーバーの HTML ドキュメントディレクトリに PHP ファイルをコピーするだけで行えます。

Apache Web サーバーの HTML ドキュメントディレクトリの一般的な場所は以下の通りです:

- /usr/local/apache2/htdocs (ソースから Apache をインストールした場合のデフォルトディレクトリ)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

HTML ルートの代わりにサブディレクトリを使用することを推奨します。サブディレクトリを作成し、Zabbix フロントエンドファイルをコピーするには、<htdocs> を実際のディレクトリに置き換えて以下のコマンドを実行します:

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

英語以外の言語を使用する場合は、追加フロントエンド言語のインストールの手順を参照してください。

フロントエンドのインストール

Zabbix フロントエンドのインストールウィザードについては、Web インターフェースのインストールページを参照してください。

3 Zabbix agent 2 ローダブルプラグインのインストール

Zabbix agent 2 ローダブルプラグインのインストールは、組み込みプラグインでカバーされていないターゲット（例：MongoDB サーバーやクラスター、PostgreSQL やそのフォークなど）を監視したい場合にのみ必要です。ローダブルプラグインと組み込みプラグインの全リストを参照してください。

Attention:
プラグインをインストールする前に、必ずその README ファイルを確認してください。特定の要件やインストール手順が記載されている場合があります。

ソースからインストールするには、まずダウンロードしてローダブルプラグインのソースアーカイブを展開します。

プラグインをコンパイルするには、展開したプラグインディレクトリに移動し、make を実行します。

```
make
```

Note:
Zabbix agent 2 ローダブルプラグインのビルドにはサポートされている Go バージョンが必要です。

プラグインの実行ファイルは、Zabbix agent 2 からロード可能であればどこに配置しても構いません。プラグインの設定ファイル（例：PostgreSQLプラグインの場合は postgresql.conf）で、プラグインバイナリへのパスを指定します。

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

プラグイン設定ファイルへのパスは、Zabbix agent 2 の設定ファイルの Include パラメータで指定する必要があります。

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

プラグインの設定の詳細については、セットアップに進んでください。

Zabbix が提供するローダブルプラグインは、以下のビルドターゲットを持つシンプルな makefile を使用しています。

- `make` - プラグインのビルド
- `make clean` - プラグインのビルドで作成されたすべてのファイルを削除
- `make check` - セルフテストの実行（実際の監視ターゲット、例：PostgreSQL データベースが必要）
- `make style` - `golangci-lint` で Go コードスタイルをチェック
- `make format` - `go fmt` で Go コードをフォーマット
- `make dist` - すべての依存関係を含むソースアーカイブを作成

4 Java ゲートウェイのインストール

JMX アプリケーションを監視する場合のみ、Java ゲートウェイのインストールが必要です。Java ゲートウェイは軽量で、データベースを必要としません。

ソースからインストールするには、まずソースアーカイブをダウンロードして展開します。

Java ゲートウェイをコンパイルするには、`./configure` スクリプトを`--enable-java` オプション付きで実行します。Java ゲートウェイのインストールでは単一の実行ファイルだけでなくディレクトリツリー全体が作成されるため、`--prefix` オプションでデフォルトの/usr/local 以外のインストールパスを指定することを推奨します。

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

Java ゲートウェイをコンパイルして JAR ファイルにパッケージ化するには、`make` を実行します。このステップでは、`javac` および `jar` の実行ファイルがパスに含まれている必要があります。

```
make
```

これで、src/zabbix_java/bin に zabbix-java-gateway-$VERSION.jar ファイルが作成されます。配布ディレクトリの src/zabbix_java から Java ゲートウェイを実行しても問題ない場合は、Java ゲートウェイの設定と実行の手順に進んでください。そうでない場合は、十分な権限があることを確認し、`make install` を実行してください。

```
make install
```

Java ゲートウェイの設定と実行の詳細については、セットアップに進んでください。

5 Zabbix Web サービスのインストール

Zabbix Web サービスのインストールは、スケジュールレポートを使用する場合にのみ必要です。

ソースからインストールするには、まずソースアーカイブをダウンロードして解凍します。

Zabbix Web サービスをコンパイルするには、`--enable-webservice` オプションを付けて`./configure` スクリプトを実行します。

Note:
Zabbix Web サービスをビルドするには、サポートされている Go バージョンが必要です。

Web サービスをインストールしたマシンで zabbix_web_service を実行します:

```
zabbix_web_service
```

スケジュールレポート生成の設定の詳細については、セットアップに進んでください。

6 Zabbix ソースコードの取得

Zabbix ソースコードを取得する方法はいくつかあります。

- 公式 Zabbix ウェブサイトからリリースされた安定バージョンをダウンロードできます
- 公式 Zabbix ウェブサイトの開発者ページからナイトリービルドをダウンロードできます
- Git ソースコードリポジトリシステムから最新の開発バージョンを取得できます:
  - フルリポジトリの主な場所はhttps://git.zabbix.com/scm/zbx/zabbix.gitです
  - マスターおよびサポートされているリリースは、Github のhttps://github.com/zabbix/zabbixにもミラーリングされています

リポジトリをクローンするには、Git クライアントがインストールされている必要があります。公式のコマンドライン Git クライアントパッケージは、ディストリビューションでは一般的に **git** と呼ばれています。例えば、Debian/Ubuntu でインストールするには、次のコマンドを実行します。

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

すべての Zabbix ソースを取得するには、コードを配置したいディレクトリに移動し、次のコマンドを実行します。

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

7 コンパイルの問題

これらは、ソースからの Zabbix のコンパイルに関する既知の問題です。他のすべての場合については、既知の問題のページを参照してください。

非標準の場所にあるライブラリ

Zabbix では、非標準の場所にあるライブラリを指定することができます。以下の例では、Zabbix は指定された非標準の場所から curl-config を実行し、その出力を使用して使用する正しい libcurl を決定します。

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

これは、システムにインストールされている libcurl がこれだけの場合は動作しますが、標準の場所（たとえばパッケージマネージャによって）に別の libcurl がインストールされている場合は動作しないことがあります。たとえば、Zabbix 用に新しいバージョンのライブラリが必要で、他のアプリケーション用に古いバージョンが必要な場合などです。

したがって、非標準の場所にあるコンポーネントを指定しても、同じコンポーネントが標準の場所にも存在する場合は常に動作するとは限りません。

たとえば、/usr/local にインストールされた新しい libcurl を libcurl パッケージがまだインストールされている状態で使用すると、Zabbix が間違ったものを取得してコンパイルに失敗することがあります。

```
usr/bin/ld: ../../src/libs/zbxhttp/libzbxhttp.a(http.o): in function 'zbx_http_convert_to_utf8':
/tmp/zabbix-master/src/libs/zbxhttp/http.c:957: undefined reference to 'curl_easy_header'
collect2: error: ld returned 1 exit status
```

ここで、関数 curl_easy_header() は古い/usr/lib/x86_64-linux-gnu/libcurl.so にはありませんが、新しい/usr/local/lib/libcurl.so にはあります。

問題はリンカーフラグの順序にあり、解決策の 1 つは LDFLAGS 変数でライブラリへのフルパスを指定することです。

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

一部のシステムでは-Wl,--no-as-needed オプションが必要な場合があることに注意してください（参考：Debian 系システムのデフォルトリンクオプション）。

一部のシステムでスタックサイズが小さい

Zabbix がスタックオーバーフローによりクラッシュまたはフリーズする場合は、ソースの設定時に--with-stacksize オプションを使用してスレッドごとのスタックサイズを増やすことができます。この問題は、デフォルトのスレッドスタック制限が低いシステム、特に事前処理中に複数のスレッドが作成される場合に発生することがあります。

次の例では、スレッドごとのスタックサイズを 512KB に設定しています。

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

Linux ベースのシステムでは、`ulimit -s` コマンドを使用して実行時にシステムのスレッドスタック制限を確認できます。

パッケージからのインストールに関する注意

概要

Zabbix 8.0 の公式 プレリリースパッケージは、Zabbix ウェブサイト から入手できます。

ご利用の環境に適したインストール手順を生成するには、オペレーティングシステムと Zabbix コンポーネントを選択してください。重要な追加情報については、このページのパッケージインストールに関する注意事項も参照してください。

パッケージは以下の Linux ディストリビューション向けに提供されています:

- Red Hat Enterprise Linux およびその派生ディストリビューション: AlmaLinux、Amazon Linux 2023、CentOS Stream、CentOS 7、Oracle Linux、Rocky Linux
- Debian、Ubuntu、Raspberry Pi OS、Raspbian
- SUSE Linux Enterprise Server、openSUSE Leap

Attention:
一部の OS ディストリビューション（特に Debian 系ディストリビューション）は、独自の Zabbix パッケージを提供しています。これらのパッケージは Zabbix ではサポートされておらず、最新の機能やバグ修正が含まれていない場合があります。Zabbix 公式リポジトリからの公式パッケージのみを使用することを推奨します。以前に OS のリポジトリから Zabbix をインストールした場合は、OS リポジトリからの Zabbix パッケージのアップグレードの手順を参照してください。

パッケージは MySQL/PostgreSQL データベースおよび Apache/Nginx ウェブサーバーをサポートしています。Zabbix サーバーとプロキシは同じデータベースを共有できないため、両方を同じホストにインストールする場合は異なるデータベース名を使用してください。

必要に応じて、Zabbix agent/agent 2、Zabbix get、Zabbix sender の個別パッケージが Zabbix 公式リポジトリ で入手できます。

Zabbix では、非 Linux オペレーティングシステム向けに Zabbix agent の事前コンパイル済みバイナリも提供しています。詳細は以下を参照してください:

- Windows エージェントの MSI からのインストール
- macOS エージェントの PKG からのインストール
- レガシーバイナリ（HP-UX、NetBSD、Tru64、古いバージョンの SLES など、古い/あまり一般的でないシステム向け）

パッケージインストールの注意事項

以下の注意事項はすべてのシステムに適用されます:

- PostgreSQL を使用する場合、Zabbix サーバー/プロキシの設定で `DBHost=localhost`（または IP アドレス）を指定すると、PostgreSQL はローカルの UNIX ソケットではなくネットワークソケットを使用します。関連するセットアップ手順についてはSELinux の設定を参照してください。
- TimescaleDB を使用する場合は、追加のTimescaleDB セットアップを参照してください。
- Zabbix Java ゲートウェイ（JMX アプリケーションの監視用）をインストールする場合は、RHEL 系システムおよびDebian 系システムの追加セットアップ手順を参照してください。
- Zabbix エージェントを root として実行する場合は、エージェントを root で実行するを参照してください。

以下の注意事項は RHEL およびその派生製品に適用されます:

- EL9 用に EPEL リポジトリを有効にしている場合、EPEL リポジトリも Zabbix パッケージを提供しているため、公式 Zabbix パッケージをインストールする前にパッケージ解決から除外する必要があります。EPEL Zabbix パッケージの誤インストールを参照してください。
- Red Hat UBI 環境で Zabbix パッケージをインストールする場合は、Red Hat UBI 環境での RHEL 用 Zabbix パッケージを参照してください。
- ICMP ping アイテムを使用する場合、`fping` 用のパッケージもZabbix 公式リポジトリで利用可能です。

SELinux の設定

Zabbix はソケットベースのプロセス間通信を使用します。Security-Enhanced Linux（SELinux）が有効になっているシステムでは、Zabbix が SocketDir ディレクトリ内で UNIX ドメインソケットを作成/使用できるように SELinux ルールを追加する必要がある場合があります。ソケットファイルは Zabbix サーバー（alerter、preprocessing、IPMI）および Zabbix プロキシ（IPMI）によって使用され、プロセスが実行中の間存在します。

SELinux が enforcing モードで有効になっている場合、Zabbix フロントエンドとサーバー間の通信を有効にするには、以下のコマンドを実行します。

RHEL 7（以降）、AlmaLinux、CentOS Stream、Oracle Linux、Rocky Linux 8（以降）の場合：

```
setsebool -P httpd_can_connect_zabbix on
```

データベースにネットワーク経由でアクセスする場合（PostgreSQL の `localhost` を含む）、Zabbix フロントエンドがデータベースに接続できるようにします。

```
setsebool -P httpd_can_network_connect_db on
```

RHEL 7 以前の場合：

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

SELinux の設定を適用した後、Apache を再起動します。

```
systemctl restart httpd
```

オプションとして、Zabbix 公式リポジトリから事前定義された `zabbix-selinux-policy` パッケージをインストールできます。このパッケージは、すべてのサポートされている OS バージョンで提供されており、Zabbix の導入を簡素化し、設定の複雑さから SELinux を無効にすることを防ぎます。

> **Attention:**
> 最大限のセキュリティを確保するために、カスタム SELinux 設定を行うことを推奨します。

`zabbix-selinux-policy` パッケージには、Zabbix がソケットを作成・使用できるようにし、HTTPd から PostgreSQL への接続（フロントエンドで使用）を有効にする基本的な SELinux ポリシーが含まれています。

ソースの `zabbix_policy.te` ファイルには、以下のルールが含まれています。

```
module zabbix_policy 1.2;

require {
    type zabbix_t;
    type zabbix_port_t;
    type zabbix_var_run_t;
    type postgresql_port_t;
    type httpd_t;
    class tcp_socket name_connect;
    class sock_file { create unlink };
    class unix_stream_socket connectto;
}

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

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

Debuginfo パッケージ

Debuginfo パッケージには、Zabbix バイナリのデバッグシンボルが含まれています。通常のインストールや運用には必要ありませんが、高度なトラブルシューティングに役立ちます。

`zabbix-debuginfo` リポジトリを有効にするには:

- RHEL 7 の場合、`/etc/yum.repos.d/zabbix.repo` を編集し、`zabbix-debuginfo` セクションの `enabled=1` を設定します:

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

- SUSE の場合、/etc/zypp/repos.d/zabbix.repo を編集し、zabbix-debuginfo セクションの enabled=1 を設定します:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

有効化したら、パッケージをインストールします:

- RHEL の場合、すべての Zabbix コンポーネントのデバッグ情報を含む単一のパッケージをインストールします:

```
dnf install zabbix-debuginfo
```

- SUSE の場合、コンポーネントごとの debuginfo パッケージをインストールします:

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

不安定リリースリポジトリの有効化

以下の手順は、Zabbix のマイナーバージョンのリリース候補で使用される、不安定な Zabbix リリースリポジトリ（デフォルトでは無効）を有効にするためのものです。

まず、zabbix-release パッケージを最新バージョンにインストールまたはアップデートしてください。rc パッケージをシステムで有効にするには、以下の操作を行います。

Red Hat Enterprise Linux

zabbix-unstable リポジトリの enabled=1 を設定するために、/etc/yum.repos.d/zabbix.repo ファイルを開きます。

```
[zabbix-unstable]
name=Zabbix Official Repository (unstable) - $basearch
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Debian/Ubuntu

/etc/apt/sources.list.d/zabbix.list を開き、「Zabbix unstable repository」のコメントを外します。

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

SUSE

zabbix-unstable リポジトリの enable=1 を設定するために、/etc/zypp/repos.d/zabbix.repo ファイルを開きます。

```
[zabbix-unstable]
name=Zabbix Official Repository
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/x86_64/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/x86_64/repodata/repomd.xml.key
enabled=1
update=1
```

4 アップグレード

アップグレードする前に、関連するすべてのアップグレードノートを参照することを強くお勧めします。

また、どのコンポーネントをアップグレードする必要があるかを知っておくと便利です。

概要

段階的なアップグレード手順は以下の通りです::

- Red Hat Enterprise Linux（パッケージを使用）
- Debian/Ubuntu（パッケージを使用）
- コンテナを使用したアップグレード
- ソースを使用したアップグレード

関連手順:

- 高可用性（HA）クラスタのサーバーについては、HA クラスタのアップグレードを参照してください
- TimescaleDB データベースについては、TimescaleDB スキーマのアップグレードを参照してください

アップグレードするコンポーネント

Zabbix エージェントのアップグレードは推奨されますが、必須ではありません。

Zabbix プロキシのアップグレードは強く推奨されます。Zabbix サーバーは、サーバーと同じメジャーバージョンのプロキシを完全にサポートします。Zabbix サーバーは、**Zabbix** サーバーの前の **LTS** リリースバージョンよりも古くないプロキシもサポートしますが、機能は制限されます（データ収集、リモートコマンドの実行、即時アイテム値チェック）。設定の更新も無効になり、古いプロキシは古い設定でのみ動作します。

Attention:
Zabbix サーバーの一つ前の LTS リリースバージョンより古いプロキシ、または Zabbix サーバーのメジャーバージョンより新しいプロキシはサポートされていません。Zabbix サーバーは、サポートされていないプロキシからのデータを無視し、Zabbix サーバーとのすべての通信は失敗し、警告が表示されます。詳細については、Version compatibilityを参照してください。

アップグレード中のダウンタイムとデータ損失を最小限に抑えるには、Zabbix サーバーを停止、アップグレード、起動してから、Zabbix プロキシを順番に停止、アップグレード、起動することをお勧めします。サーバーのダウンタイム中も、実行中のプロキシはデータ収集を継続します。サーバーが稼働すると、旧式のプロキシはデータを新しいサーバーに送信し (ただし、プロキシ設定は更新されません)、部分的には機能し続けます。Zabbix サーバーのダウンタイム中に発生した障害に関する通知は、アップグレードされたサーバーが起動した後にのみ生成されます。

Zabbix プロキシが初めて起動され、SQLite データベースファイルが存在しない場合、プロキシは自動的にファイルを作成します。

注意: Zabbix プロキシが SQLite3 を使用していて、起動時に既存のデータベースファイルのバージョンが必要なバージョンよりも古いことを検出した場合、データベースファイルを自動的に削除し、新しいものを作成します。そのため、SQLite データベースファイルに保存されている履歴データは失われます。Zabbix プロキシのバージョンがデータベースファイルのバージョンよりも古い場合、Zabbix はエラーを記録して終了します。

データベースのサイズによっては、バージョン 8.0 へのデータベースのアップグレードに長い時間がかかる場合があります。

アップグレードに関する注意事項

Zabbix 8.0.x への直接アップグレードは、バージョン **2.0**.x 以降からサポートされています。それ以前のバージョンからアップグレードする場合は、Zabbix 2.0 以前のドキュメントを参照してください。

Note:
アップグレード後、外部ソフトウェアがアップグレードされた Zabbix バージョンと互換性がない場合、Zabbix の一部のサードパーティ製ソフトウェアとの連携に影響が出る可能性があることにご注意ください。

以下のアップグレードノートが利用可能です:

アップグレード元	完全なアップグレードノートを読む	バージョン間の最も重要な変更点
7.4.x	対象: Zabbix 8.0	必要な最小データベースバージョンが引き上げられました。
7.2.x	対象: Zabbix 7.4 Zabbix 8.0	PCRE ライブラリのサポートが廃止されました。
7.0.x	対象: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Oracle DB のサポートが廃止されました。

アップグレード元	完全なアップグレードノートを読む	バージョン間の最も重要な変更点
6.4.x	対象: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 PHP バージョンが 7.4.0 から 8.0.0 に引き上げられました。 エージェント、HTTP エージェント、SNMP walk[oid] チェック用の非同期ポーラー。 プロキシ用の個別のデータベーステーブル。 Windows エージェントの設定ファイルのデフォルトの場所が変更されました。 Oracle DB は非推奨となりました。 古い数値（float）値タイプが廃止されました。
6.2.x	対象: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 MySQL バージョンが 8.0.0 から 8.0.30 に引き上げられました。 Zabbix サーバー/プロキシには'libevent_pthreads' ライブラリが必要です。 アップグレード後の最初の起動時に、SQLite3 を使用する Zabbix プロキシは自動的に古いバージョンのデータベース（すべての履歴を含む）を削除し、新しいデータベースを作成します。
6.0.x LTS	対象: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 PHP バージョンが 7.2.5 から 7.4.0 に引き上げられました。 サービス監視が大幅に再設計されました。 決定論的トリガーはアップグレード時に作成する必要があります。 MySQL/MariaDB でバイナリログが有効な場合、これにはスーパーユーザー権限または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。変数の設定方法についてはデータベース作成スクリプトを参照してください。
5.4.x	対象: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小データベースバージョンが引き上げられました。 古いデータベースの場合、サーバー/プロキシは起動しません。 データベース構造の変更により監査ログ記録が失われます。
5.2.x	対象: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小データベースバージョンが引き上げられました。 集計アイテムが個別のタイプとして削除されました。
5.0.x LTS	対象: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 PHP バージョンが 7.2.0 から 7.2.5 に引き上げられました。 パスワードハッシュアルゴリズムが MD5 から bcrypt に変更されました。
4.4.x	対象: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	IBM DB2 のサポートが廃止されました。 必要な最小 PHP バージョンが 5.4.0 から 7.2.0 に引き上げられました。 必要な最小データベースバージョンが引き上げられました。 Zabbix PHP ファイルディレクトリが変更されました。

アップグレード元	完全なアップグレードノートを読む	バージョン間の最も重要な変更点
4.2.x	対象: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Jabber、Ez Texting メディアタイプが削除されました。
4.0.x LTS	対象: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	古いプロキシはアップグレードされたサーバーにデータを報告できなくなりました。 新しいエージェントは古い Zabbix サーバーと動作できなくなります。
3.4.x	対象: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	'libpthread' および'zlib' ライブラリが必須になりました。 プレーンテキストプロトコルのサポートが廃止され、ヘッダーが必須になりました。 バージョン 1.4 以前の Zabbix エージェントはサポートされなくなりました。 パッシブプロキシ設定の Server パラメータが必須になりました。
3.2.x	対象: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Zabbix サーバー/フロントエンドのバックエンドデータベースとしての SQLite サポートが廃止されました。 POSIX 拡張の代わりに Perl 互換正規表現（PCRE）がサポートされます。 Zabbix サーバーには'libpcre' および'libevent' ライブラリが必須です。 ユーザーパラメータ、リモートコマンド、'nowait' フラグなしの system.run[] アイテム、および Zabbix サーバーで実行されるスクリプトに対して終了コードチェックが追加されました。 新機能をサポートするには Zabbix Java ゲートウェイのアップグレードが必要です。

アップグレード元	完全なアップグレードノートを読む	バージョン間の最も重要な変更点
3.0.x LTS	対象: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	データベースのアップグレードは、ヒストリーテーブルのサイズによっては遅くなる場合があります。
2.4.x	対象: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 PHP バージョンが 5.3.0 から 5.4.0 に引き上げられました。 LogFile エージェントパラメータを指定する必要があります。
2.2.x LTS	対象: Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	ノードベースの分散監視が削除されました。

アップグレード元	完全なアップグレードノートを読む	バージョン間の最も重要な変更点
2.0.x	対象: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な最小 PHP バージョンが 5.1.6 から 5.3.0 に引き上げられました。 サーバーが正しく動作するには大文字小文字を区別する MySQL データベースが必要です。Zabbix サーバーが MySQL データベースで正しく動作するには、文字セット utf8 および utf8_bin 照合順序が必要です。データベース作成スクリプトを参照してください。 'mysql' の代わりに'mysqli' PHP 拡張が必要です。

パッケージからのアップグレード

概要

このセクションでは、Zabbix が提供する公式の RPM および DEB パッケージを使用して、以下の OS でアップグレードを成功させるために必要な手順を説明します。

- Red Hat Enterprise Linux
- Debian/Ubuntu

OS リポジトリからの Zabbix パッケージ

一部の OS ディストリビューション（特に Debian 系ディストリビューション）は、独自の Zabbix パッケージを提供しています。これらのパッケージは **Zabbix** によってサポートされておらず、古いバージョンであったり、最新の機能やバグ修正が含まれていない場合があります。Zabbix 公式リポジトリからの公式パッケージのみを使用することを推奨します。

OS ディストリビューションが提供するパッケージからアップグレードする場合（または一度でもインストールしたことがある場合）は、公式の Zabbix パッケージに切り替えるために次の手順に従ってください。

1. 古いパッケージをアンインストールします。
2. 古いインストールから残っているファイルがないか確認し、削除します。
3. Zabbix が提供するインストール手順に従って公式の Zabbix パッケージをインストールします。

直接アップグレードは行わないでください。インストールが壊れる可能性があります。

Debian/Ubuntu

概要

このセクションでは、Debian/Ubuntu 用の公式 Zabbix パッケージを使用して、Zabbix **7.4.x** から最新バージョンの Zabbix **8.0.x** へのアップグレード手順を説明します。

Warning:
アップグレードの前に、関連するアップグレードノートを確認し、システムが Zabbix 8.0 の要件を満たしていることを確認してください。

Note:
アップグレード中は、2 つの SSH セッションを並行して実行することを検討してください。1 つはアップグレード手順の実行用、もう 1 つはサーバー/プロキシのログ監視用です。たとえば、2 つ目のセッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行し、最新のログエントリやエラーをリアルタイムで確認できます。これは本番環境では非常に重要です。

Zabbix 8.0.x のマイナーバージョン間（例：8.0.1 から 8.0.3 へのアップグレード）のアップグレード手順については、マイナーバージョン間のアップグレードを参照してください。

アップグレード手順

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

```
systemctl stop zabbix-server
```

Zabbix プロキシやエージェント、エージェント 2 もアップグレードする場合は、同様に停止します。

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

2 Zabbix データベースのバックアップ

既存の Zabbix データベースをバックアップして、アップグレードの失敗 (ディスク容量の問題、電源喪失、予期せぬ問題など) から保護します。

3 Zabbix 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

既存の Zabbix 設定ファイル、PHP ファイル、および Zabbix バイナリをバックアップします。

設定ファイルの場合は、次を実行します。

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

PHP ファイルと Zabbix バイナリの場合は、次を実行します。

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

4 リポジトリ設定パッケージの更新

アップグレードを進める前に、現在の Zabbix リポジトリパッケージをアンインストールしてください:

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

また､作業ディレクトリから古い Zabbix パッケージを手動で削除する必要がある場合があります(例: `rm zabbix-release_latest+debian1`
これは、アップグレード時にパッケージマネージャが古いバージョンを再利用するのを防ぐためです。

次に、最新のリポジトリ設定パッケージをインストールして、最新のパッケージとの互換性を確保し、最近のセキュリティパッチやバグ修正を含めます。

Debian 12 では、以下を実行します:

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

> **Note:**
> 古い Debian バージョンの場合は、上記のリンクをZabbix リポジトリから正しいものに置き換えてください。ただし、これらのバージョンのパッケージにはすべての Zabbix コンポーネントが含まれていない場合があり、パッケージからそれらのコンポーネントをアップグレードするには OS のアップグレードを検討してください。含まれるコンポーネントの一覧については、Zabbix パッケージを参照してください。

Ubuntu 24.04 では、以下を実行します:

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

Ubuntu 22.04 では、以下を実行します:

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

Note:
古い Ubuntu バージョンの場合は、上記のリンクをZabbix リポジトリから正しいものに置き換えてください。ただし、これらのバージョンのパッケージにはすべての Zabbix コンポーネントが含まれていない場合があり、パッケージからそれらのコンポーネントをアップグレードするには OS のアップグレードを検討してください。含まれるコンポーネントの一覧については、Zabbix パッケージを参照してください。

Zabbix リポジトリ設定に関するプロンプトが表示される場合があります:

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Deleted (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Zabbix リポジトリ設定のパッケージメンテナのバージョンをインストールするには、Y（または I）を入力してください。

その後、リポジトリ情報を更新します:

```
apt update
```

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

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

- PostgreSQL を使用している場合は、コマンドの `mysql` を `pgsql` に置き換えます。
- プロキシをアップグレードする場合は、コマンドの `server` を `proxy` に置き換えます。
- Zabbix エージェント 2 をアップグレードする場合は､コマンドの `zabbix-agent` を `zabbix-agent2 zabbix-agent2-plugin-*` に置き換えます。

Attention:
'apt install zabbix-agent2' コマンドを使用して Zabbix エージェント 2 をアップグレードすると、エラーが発生する可能性があります。詳細については既知の問題点を参照してください。

Zabbix サーバー (またはプロキシ) の設定に関するプロンプトが表示される場合があります。

```
Configuration file '/etc/zabbix/zabbix_server.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix_server.conf (Y/I/N/O/D/Z) [default=N] ?
```

状況に最も適したオプションを入力してください。たとえば、D と入力して現在の設定と新しい設定を比較し、パッケージ管理者のバージョン (Y または I) をインストールするかどうかを決定します。

次に、Apache を使用した Zabbix フロントエンドをアップグレードし、再起動するには、次のコマンドを実行します。

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

6 コンポーネントの設定パラメータの確認

関連するアップグレードノートを確認し、設定パラメータに変更が必要かどうかを確認してください。

新しいオプションパラメータについては、新機能ページを参照してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix Web インターフェースが正しく動作するために、Web ブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

マイナーバージョン間のアップグレード

Zabbix 8.0.x のマイナーバージョン間（例: 8.0.1 から 8.0.3 へのアップグレード）が可能です。

まず、リポジトリ情報を更新します:

```
apt update
```

次に、すべての Zabbix コンポーネントをアップグレードするには、以下を実行します:

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

- Zabbix サーバーのみをアップグレードする場合は、コマンド内の `'zabbix*'` を `'zabbix-server*'` に置き換えてください。
- Zabbix プロキシのみをアップグレードする場合は、コマンド内の `'zabbix*'` を `'zabbix-proxy*'` に置き換えてください。
- Zabbix エージェントのみをアップグレードする場合は、コマンド内の `'zabbix*'` を `'zabbix-agent*'` に置き換えてください。
- Zabbix エージェント 2 のみをアップグレードする場合は、コマンド内の `'zabbix*'` を `'zabbix-agent2*'` に置き換えてください。

Red Hat Enterprise Linux

概要

このセクションでは、Red Hat Enterprise Linux またはその派生ディストリビューション（AlmaLinux、CentOS Stream、Oracle Linux、Rocky Linux）用の公式 Zabbix パッケージを使用して、Zabbix **7.4.x** から最新バージョンの Zabbix **8.0.x** へアップグレードする手順を説明します。

> **Warning:**
> アップグレードを行う前に、関連するアップグレードノートを確認し、システムが Zabbix 8.0 の要件を満たしていることを確認してください。

> **Note:**
> アップグレード中は、2 つの SSH セッションを並行して実行することを検討してください。1 つはアップグレード手順の実行用、もう 1 つはサーバー/プロキシのログ監視用です。たとえば、2 つ目のセッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行し、最新のログエントリやエラーをリアルタイムで確認できます。これは本番環境では非常に重要です。

Zabbix 8.0.x のマイナーバージョン間（例：8.0.1 から 8.0.3 へのアップグレード）のアップグレード手順については、マイナーバージョン間のアップグレードを参照してください。

アップグレード手順

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

```
systemctl stop zabbix-server
```

Zabbix プロキシやエージェント、エージェント 2 もアップグレードする場合は、同様に停止します。

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

2 Zabbix データベースのバックアップ

既存の Zabbix データベースをバックアップして、アップグレードの失敗 (ディスク容量の問題、電源喪失、予期せぬ問題など) から保護します。

3 Zabbix 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

既存の Zabbix 設定ファイル、PHP ファイル、および Zabbix バイナリをバックアップします。

設定ファイルの場合は、次を実行します。

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

PHP ファイルと Zabbix バイナリの場合は、次を実行します。

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

4 リポジトリ設定パッケージの更新

アップグレードを進める前に、現在のリポジトリパッケージを最新バージョンに更新し、最新パッケージとの互換性を確保し、最近のセキュリティパッチやバグ修正を含めてください。

RHEL 10 では、以下を実行します:

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

RHEL 9 では、以下を実行します:

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

> **Note:**
> 古い RHEL バージョンやその派生ディストリビューションの場合は、上記のリンクをZabbix リポジトリから正しいものに置き換えてください。ただし、これらのバージョンのパッケージにはすべての Zabbix コンポーネントが含まれていない場合があり、パッケージからそれらのコンポーネントをアップグレードするには、OS のアップグレードを検討してください。含まれるコンポーネントの一覧については、Zabbix パッケージを参照してください。

次に、dnf パッケージマネージャのキャッシュ（ヘッダー、メタデータ、以前のインストールやアップデートでダウンロードされたパッケージファイルを含む）をクリーンアップします:

```
dnf clean all
```

次回の dnf 操作時に、古いメタデータがクリアされているため、dnf はリポジトリから新しいメタデータをダウンロードします。

RHEL でリポジトリ設定パッケージを更新する際の既知の問題も参照してください。

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

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

- PostgreSQL を使用している場合は、コマンドの `mysql` を `pgsql` に置き換えます。
- プロキシをアップグレードする場合は、コマンドの `server` を `proxy` に置き換えます。
- エージェント 2 をアップグレードする場合は,コマンドの `zabbix-agent` を `zabbix-agent2 zabbix-agent2-plugin-*` に置き換えます。

> **Attention:**
> 'dnf install zabbix-agent2' コマンドを使用して Zabbix エージェント 2 をアップグレードすると、エラーが発生する可能性があります。詳細については既知の問題点を参照してください。

次に、Apache を使用した Web フロントエンドをアップグレードし、再起動するには、次のコマンドを実行します。

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

6 コンポーネントの設定パラメータの確認

関連するアップグレードノートを確認し、設定パラメータに変更が必要かどうかを確認してください。

新しいオプションパラメータについては、新機能ページを参照してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix Web インターフェースが正しく動作するために、Web ブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

マイナーバージョン間のアップグレード

Zabbix 8.0.x のマイナーバージョン間（例：8.0.1 から 8.0.3 など）でアップグレードすることが可能です。

すべての Zabbix コンポーネントをアップグレードするには、以下を実行します。

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

- Zabbix サーバーのみをアップグレードする場合は、コマンド内の `'zabbix-*'` を `'zabbix-server-*'` に置き換えてください。
- Zabbix プロキシのみをアップグレードする場合は、コマンド内の `'zabbix-*'` を `'zabbix-proxy-*'` に置き換えてください。
- Zabbix エージェントのみをアップグレードする場合は、コマンド内の `'zabbix-*'` を `'zabbix-agent-*'` に置き換えてください。
- Zabbix エージェント 2 のみをアップグレードする場合は、コマンド内の `'zabbix-*'` を `'zabbix-agent2-*'` に置き換えてください。

アップグレードの問題

エスケープ関連　アップグレード後の関数パラメータの文字数制限超過

Zabbix 7.0 では、ヒストリ関数の文字列パラメータでバックスラッシュの適切なエスケープが追加されました。

7.0 以前の Zabbix バージョンからアップグレードする際に追加のバックスラッシュが追加されるため、パラメータが長くなり、パラメータの長さが最大データサイズの 255 文字を超えるとトリガー関数が壊れる可能性があります。

データベースのアップグレード中にどのような問題が発生し、それらの影響、認識方法、修正方法を説明するために、3 つのアイテムと 2 つのトリガーを持つ例を挙げます。

セットアップ

1) Zabbix 6.0 バージョン。

2) 114 個のバックスラッシュと末尾に'a' がある /tmp/ONE ファイル:

```
\\\
```

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

そして、トリガー式を更新してこれらを使用します:

Trig_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Trig_232:

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

これで、両方のトリガーは以前と同じように動作します - Trig_228 のみが発火します。

計算アイテムには問題ありませんが、一貫性のために更新することもできます:

Calc_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Calc_232:

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

コンテナからのアップグレード

概要

このセクションでは、Zabbix **8.0**.x コンテナへのアップグレードを成功させるために必要な手順について説明します。

個々の Zabbix コンポーネントイメージや Docker compose ファイルのアップグレード用に、別々の手順が用意されています。

> **Warning:**
> アップグレードの前に、関連するアップグレードノートを必ずお読みください！

> **Attention:**
> アップグレードを開始する前に、ユーザーがアップグレードのために必要なデータベース権限を持っていることを確認してください。
>
> Zabbix 6.0 以前からのアップグレードでは、アップグレード中に決定論的トリガーを作成する必要があります。MySQL/MariaDB でバイナリログが有効になっている場合、これにはスーパーユーザー権限、または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。変数の設定方法についてはデータベース作成スクリプトを参照してください。
> コンソールから実行する場合、変数は一時的にしか設定されず、Docker が再起動されると解除されることに注意してください。この場合、SQL サービスは稼働させたまま、'docker compose down zabbix-server' で zabbix-server サービスのみを停止し、その後'docker compose up -d zabbix-server' で再起動してください。
> または、この変数を設定ファイルに記述することもできます。

データベースのサイズによっては、バージョン 8.0 へのアップグレードにかなりの時間がかかる場合があります。

Zabbix イメージのアップグレード

以下の手順は、任意の Zabbix コンポーネントのアップグレードに使用できます。`zabbix-server-mysql` を必要なコンポーネントのイメージ名に置き換えてください。

1. 現在のイメージバージョンを確認します:

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

2. 希望するイメージバージョンをプルします。例:

```
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest
```

`zabbix/zabbix-server-mysql:alpine-8.0-latest` は、Alpine Linux ベースの MySQL サポート付き Zabbix サーバー 8.0 の最新のマイナーバージョンをプルします。必要な Docker リポジトリとタグの組み合わせの名前に置き換えてください。利用可能なオプションの一覧については、コンテナからのインストールを参照してください。

3. コンテナを停止します:

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

4. コンテナを削除します:

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

5. docker run コマンドを実行し、必要な環境変数やマウントポイントを指定する追加引数を続けて、更新されたコンテナを起動します。

設定例

MySQL を使用した Zabbix サーバー:

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

PostgreSQL を使用した Zabbix サーバー:

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

他の Zabbix コンポーネントを含む、より多くの設定例は、コンテナからのインストールページで利用できます。

6. アップデートを確認します:

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

Compose ファイル

compose ファイルを使用して Zabbix をインストールした場合は、このセクションのアップグレード手順に従ってください。

1. 現在のイメージバージョンを確認します:

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

2. GitHub のリポジトリから最新の更新を取得し、必要なブランチに切り替えます:

```
git pull
git checkout 8.0
```

3. 新しい compose ファイルを使用して Zabbix コンポーネントを起動します:

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

4. アップデートを確認します:

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

サポートされている環境変数やボリュームマウントポイントのリストなど、詳細についてはコンテナからのインストールを参照してください。

ソースからのアップグレード

概要

このセクションでは、公式の Zabbix ソースを使用して Zabbix **7.4**.x から Zabbix **8.0**.x へのアップグレードを成功させるために必要な手順を説明します。

> **Warning:**
> アップグレードの前に、関連するアップグレードノートを必ずお読みください！

8.0 の要件も確認することをお勧めします。

Note:
アップグレード中は、2 つの SSH セッションを並行して実行すると便利です。1 つのセッションでアップグレード手順を実行し、もう 1 つのセッションでサーバー/プロキシのログを監視します。たとえば、2 つ目の SSH セッションで `tail -f zabbix_server.log` や `tail -f zabbix_proxy.log` を実行すると、最新のログファイルのエントリやリアルタイムで発生する可能性のあるエラーを確認できます。これは本番環境では非常に重要です。

サーバーのアップグレードプロセス

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

2 既存 Zabbix データベースのバックアップ

これは非常に重要なステップです。データベースのバックアップがあることを確認してください。アップグレード手順が失敗（ディスク容量の不足、電源オフ、予期しない問題）した場合に役立ちます。

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

Zabbix バイナリ、設定ファイル、PHP ファイルディレクトリのバックアップコピーを作成します。

4 新しいサーバーバイナリのインストール

これらの手順 を使用して、ソースから Zabbix サーバーをコンパイルします。

5 サーバーの設定パラメータを確認する

アップグレードノートを確認し、設定パラメータの変更が必要かどうかを確認してください。

新しいオプションパラメータについては、新機能ページを参照してください。

6 新しい Zabbix バイナリの開始

新しいバイナリを開始します。ログファイルをチェックして、バイナリが正常に開始されたかどうかを確認します。

Zabbix サーバーはデータベースを自動的にアップグレードします。起動時に、Zabbix サーバーは現在の（必須およびオプションの）データベースバージョンと必要なデータベースバージョンを確認します。現在の必須バージョンが必要なバージョンよりも古い場合、Zabbix サーバーは必要なデータベースアップグレードパッチを自動的に実行します。データベースアップグレードの開始レベルと進行レベル（パーセンテージ）は、Zabbix サーバーのログファイルに書き込まれます。アップグレードが完了すると、「データベースのアップグレードが完全に完了しました」というメッセージがログファイルに書き込まれます。アップグレードパッチのいずれかが失敗した場合、Zabbix サーバーは起動しません。現在のデータベースバージョンが必要なバージョンよりも新しい場合も、Zabbix サーバーは起動しません。Zabbix サーバーは、現在のデータベースバージョンがバージョンに対応している場合にのみ起動します。

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

サーバーを起動する前に：

- データベースユーザーに十分な権限があることを確認してください（テーブルの作成、テーブルの削除、インデックスの作成、インデックスの削除）
- 十分な空きディスク容量があることを確認してください。

7 新しい Web インターフェースのインストール

PHP の必要最小バージョンは 8.0.0 です。必要に応じて更新しインストール手順に従ってください。

8 Web ブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix Web インターフェースが正しく機能するために、Web ブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

プロキシのアップグレードプロセス

1 プロキシのを停止

Zabbix プロキシを停止します。

2 設定ファイルと Zabbix プロキシバイナリのバックアップ

Zabbix プロキシバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいプロキシバイナリのインストール

これらの手順 を使用して、ソースから Zabbix プロキシをコンパイルします。

4 プロキシ設定パラメーターの確認

このバージョンでは、プロキシのパラメーターに必須の変更はありません。

5 新しい Zabbix プロキシの起動

新しい Zabbix プロキシを起動し、ログファイルをチェックしてプロキシが正常に開始されたかどうかを確認してください。

Zabbix プロキシはデータベースを自動的にアップグレードします。データベースのアップグレードは、Zabbix サーバーを起動するときと同様に行われます。

エージェントのアップグレードプロセス

> **Attention:**
> エージェントのアップグレードは必須ではありません。新しい機能にアクセスする必要がある場合にのみ、エージェントをアップグレードする必要があります。

このセクションで説明するアップグレード手順は、Zabbix エージェントと Zabbix エージェント 2 の両方をアップグレードするために使用できます。

1 エージェントの停止

Zabbix エージェントを停止します。

2 設定ファイルと Zabbix エージェントバイナリのバックアップ

Zabbix エージェントバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいエージェントバイナリのインストール

Zabbix エージェントをソースからコンパイルするには、この手順を使用してください。

または、Zabbix ダウンロードページから事前にコンパイルされた Zabbix エージェントをダウンロードすることもできます。

4 エージェント設定パラメーターの確認

このバージョンではエージェントまたはエージェント 2のパラメーターに必須の変更はありません。

5 新しい Zabbix エージェントの起動

新しい Zabbix エージェントを起動し、ログファイルをチェックして、エージェントが正常に起動したかどうかを確認します。

マイナーバージョン間のアップグレード

8.0.x のマイナーバージョン間（例: 8.0.1 から 8.0.3）でアップグレードする場合、メジャーバージョン間のアップグレード時と同じアクションをサーバー/プロキシ/エージェントに対して実行する必要があります。唯一の違いは、マイナーバージョン間のアップグレードではデータベースに変更が加えられないことです。

既知の問題

参考: コンパイルの問題

アップグレード

正常なアップグレードのための SQL モード設定

MySQL/MariaDB の `sql_mode` 設定には、”STRICT_TRANS_TABLES” モードが設定されている必要があります。これが設定されていない場合、Zabbix データベースのアップグレードは失敗します（ZBX-19435も参照してください）。

MariaDB 10.2.1 以前でのアップグレード

MariaDB 10.2.1 以前でデータベーステーブルが作成された場合、Zabbix のアップグレードが失敗することがあります。これらのバージョンではデフォルトの行フォーマットが compact であるためです。この問題は、行フォーマットを dynamic に変更することで修正できます（ZBX-17690も参照）。

テンプレート

デュアルスタック（IPv4/IPv6）環境におけるテンプレートの互換性

デュアルスタック環境（IPv4 と IPv6 の両方をサポートするように構成されたシステム）では、ホスト名 `localhost` は通常、IPv4 アドレスと IPv6 アドレスの両方に解決されます。多くのオペレーティングシステムや DNS リゾルバが IPv4 よりも IPv6 を優先するという一般的な優先順位付けのため、監視対象のサービスが IPv4 のみでリッスンするように構成されている場合、Zabbix テンプレートが正しく動作しないことがあります。

IPv6 アドレスでリッスンするように構成されていないサービスはアクセスできなくなり、監視が失敗する可能性があります。ユーザーは IPv4 で正しくアクセスを構成していても、IPv6 がデフォルトで優先される動作により接続の問題に直面することがあります。

この回避策としては、サービス（Nginx、Apache、PostgreSQL など）が IPv4 と IPv6 の両方のアドレスでリッスンするように構成されており、Zabbix サーバー/エージェントが IPv6 経由でアクセスできるようにすることです。さらに、Zabbix のテンプレートや設定では、`127.0.0.1` の代わりに `localhost` を明示的に使用して、IPv4 と IPv6 の両方との互換性を確保してください。

例えば、PostgreSQL by Zabbix agent 2 テンプレートで PostgreSQL を監視する場合、`zbx_monitor` ユーザーの接続を許可するために `pg_hba.conf` ファイルを編集する必要があるかもしれません。デュアルスタック環境で IPv6 が優先されている場合（システムが localhost を `::1` に解決する）に `localhost` を構成し、IPv4 エントリ（`127.0.0.1/32`）のみを追加した場合、対応する IPv6 エントリがないため接続は失敗します。

次の `pg_hba.conf` ファイルの例は、`zbx_monitor` ユーザーがローカルマシンから IPv4 および IPv6 アドレスの両方を使用して、異なる認証方式で任意のデータベースに接続できることを保証します。

```
### TYPE     DATABASE     USER            ADDRESS          METHOD
  host     all          zbx_monitor     localhost        trust
  host     all          zbx_monitor     127.0.0.1/32     md5
  host     all          zbx_monitor     ::1/128          scram-sha-256
```

必要に応じて、接続文字列のテンプレートマクロを構成する際に、PostgreSQL by Zabbix agent 2 テンプレートで IPv4 アドレス（`127.0.0.1`）を直接使用することもできます。

EPEL Zabbix パッケージの誤インストール

EPEL リポジトリがインストールされ有効になっている場合、Zabbix パッケージのインストール時に公式の Zabbix パッケージではなく EPEL バージョンがインストールされることがあります。これを解決するには:

1. EPEL からインストールされた Zabbix パッケージを削除します:

```
dnf remove zabbix-server-mysql
```

2. `/etc/yum.repos.d/epel.repo` ファイルに以下の行を追加して、EPEL から Zabbix パッケージを除外します:

```
[epel]
...
excludepkgs=zabbix*
```

3. 公式の Zabbix サーバーパッケージを再インストールします:

```
dnf install zabbix-server-mysql
```

インストール時､公式の Zabbix パッケージはバージョン文字列に `release` という単語が含まれており (例: `7.0.0-release1.el8`)、EPEL パッケージと区別できます。

Red Hat UBI 環境の RHEL 用 Zabbix パッケージ

Red Hat Universal Base Image環境で Red Hat Enterprise Linux パッケージから Zabbix をインストールする場合は、必要なリポジトリと依存関係にアクセスできることを確認してください。Zabbix パッケージは libOpenIPMI.so および libOpenIPMIposix.so ライブラリに依存していますが、これらは UBI システムで有効になっているデフォルトのパッケージマネージャーリポジトリのどのパッケージでも提供されていないため、インストールが失敗します。

libOpenIPMI.so および libOpenIPMIposix.so ライブラリは、redhat-#-for-<arch>-appstream-rpms リポジトリによって提供される OpenIPMI-libs パッケージで使用できます。このリポジトリへのアクセスはサブスクリプションによって管理され UBI 環境の場合、RHEL ホストのリポジトリ構成およびシークレットディレクトリをコンテナーファイルシステムの名前空間にマウントすることによって伝播されます。

詳細についてはZBX-24291を参照してください。

RHEL パッケージの署名鍵の有効期限切れ

Red Hat Enterprise Linuxまたはその派生ディストリビューションで Zabbix をアップグレードする際、Zabbix リポジトリのパッケージに対して署名鍵の有効期限切れの問題が発生する場合があります。署名鍵が有効期限切れになると、パッケージ署名の検証を試みても、証明書または鍵が有効でないことを示すエラーが発生します。例:

```
error: Verifying a signature using certificate D9AA84C2B617479C6E4FCF4D19F2475308EFA7DD (Zabbix LLC (Jul 2
  1. Certificate 19F2475308EFA7DD invalid: certificate is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
  2. Key 19F2475308EFA7DD invalid: key is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
```

このような問題を解決するには、RHEL のバリアントに合わせて最新の `zabbix-release` パッケージを手動で再インストールしてください（下記のリンクはZabbix リポジトリから正しいものに置き換えてください）。

例えば、**RHEL 10** の場合は次のコマンドを実行します:

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

その後、リポジトリ情報を更新します:

```
dnf update
```

詳細はZBX-24761を参照してください。

Timescale DB: パーティション数が多い場合のメモリ使用量の増加

PostgreSQL バージョン 9.6-12 では、多数のパーティションを持つテーブルを更新する際に過剰なメモリを使用します。この問題は、Zabbix が TimescaleDB を使用しているシステムでトレンドを比較的小さなチャンク（例：1 日）に分割している場合に発生します。これにより、デフォルトのハウスキーピング設定ではトレンドテーブルに数百のチャンクが存在し、PostgreSQL がメモリ不足に陥る可能性があります。

この問題は、TimescaleDB を使用した新規インストールの Zabbix 5.0.1 以降で解決されていますが、それ以前に TimescaleDB をセットアップした場合は、ZBX-16347の移行手順をご参照ください。

Timescale DB 2.5.0: 整数を含むテーブルで圧縮ポリシーが失敗する可能性

この問題は、TimescaleDB 2.5.0/2.5.1 を使用している場合に発生します。TimescaleDB 2.5.2 以降で解決されています。

詳細については、TimescaleDB Issue #3773をご参照ください。

MariaDB とのデータベース TLS 接続

MariaDB を使用している場合、DBTLSConnect パラメータの'verify_ca' オプションによるデータベース TLS 接続はサポートされていません。

MySQL/MariaDB で発生する可能性のあるデッドロック

高負荷で動作し、複数の LLD ワーカーが関与している場合、行ロック戦略に関連する InnoDB エラーが原因でデッドロックが発生する可能性があります (上流バグを参照)。このエラーは MySQL 8.0.29 以降で修正されていますが、MariaDB では修正されていません。詳細はZBX-21506を参照してください。

グローバルイベントの相関

最初のイベントと 2 番目のイベントの間隔が非常に短い場合、つまり 0.5 秒以下の場合、イベントが正しく相関されないことがあります。

PostgreSQL 11 以前の数値（浮動小数点）データ型の範囲

PostgreSQL 11 以前のバージョンでは、約-1.34E-154 から 1.34E+154 までの浮動小数点値の範囲のみがサポートされています。

NetBSD 8.0 以降

NetBSD 8.X および 9.X バージョンでは、さまざまな Zabbix プロセスが起動時にランダムにクラッシュすることがあります。これは、デフォルトのスタックサイズ（4MB）が小さすぎるためであり、次のコマンドを実行して増やす必要があります。

```
ulimit -s 10240
```

詳細については、関連する問題レポートをご覧ください: ZBX-18275。

Zabbix エージェント 2 の正規表現の制限

Zabbix エージェント 2 は、標準の Go 正規表現ライブラリの制限により、正規表現の先読みと後読みをサポートしていません。

IPMI チェック

Debian 9（stretch）以前および Ubuntu 16.04（xenial）以前の標準 OpenIPMI ライブラリパッケージでは、IPMI チェックは動作しません。これを修正するには、ZBX-6139で説明されているように、OpenSSL を有効にして OpenIPMI ライブラリを再コンパイルしてください。

IPMI — 信頼できないホストが OpenIPMI をクラッシュさせる可能性がある

Zabbix が IPMI データのポーリングに使用する OpenIPMI ライブラリには、信頼できないデバイスからの細工された応答によってトリガーされるバグがあります。
信頼できない IPMI デバイスが細工されたデータを送信すると、OpenIPMI ライブラリがクラッシュし、結果として IPMI ポーリングを実行する Zabbix サーバープロセスが終了する可能性があります。

SSH チェック

- Debian、Ubuntu などの一部の Linux ディストリビューションでは、libssh2 ライブラリがパッケージからインストールされている場合、暗号化された秘密鍵（パスフレーズ付き）はサポートされません。詳細については、ZBX-4850をご覧ください。
- 一部の Linux ディストリビューションで OpenSSH 8 と libssh 0.9.x を使用している場合、SSH チェックで「Cannot read data from SSH server」と報告されることがあります。これは libssh の問題（詳細なレポート）が原因です。このエラーは、安定版 libssh 0.9.5 リリースで修正されているはずです。詳細については、ZBX-17756もご覧ください。
- SSH スクリプトでパイプ「|」を使用すると、「Cannot read data from SSH server」エラーが発生する場合があります。この場合は、libssh ライブラリのバージョンアップを推奨します。詳細については、ZBX-21337もご覧ください。

ODBC チェック

- MySQL unixODBC ドライバは、MariaDB コネクタライブラリでコンパイルされた Zabbix サーバーまたは Zabbix プロキシと一緒に使用しないでください。また、その逆も同様です。可能であれば、上流のバグのため、ドライバと同じコネクタの使用も避けるのが望ましいです。推奨されるセットアップ:

PostgreSQL、SQLite、または Oracle コネクタ → MariaDB または MySQL unixODBC ドライバ MariaDB コネクタ → MariaDB unixODBC ドライバ MySQL コネクタ → MySQL unixODBC ドライバ

詳細および利用可能な回避策については、ZBX-7665を参照してください。

- Microsoft SQL Server からクエリされた XML データは、Linux および UNIX システム上でさまざまな方法で切り捨てられる場合があります。
- Linux 用のさまざまなバージョンの Oracle Instant Client を使用して Oracle データベースを監視するために ODBC チェックを使用すると、Zabbix サーバーがクラッシュすることが観察されています。
  詳細はZBX-18402、ZBX-20803も参照してください。
- FreeTDS UnixODBC ドライバを使用する場合、SQL クエリの先頭に「SET NOCOUNT ON」ステートメントを追加する必要があります（例: `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`）。そうしないと、Zabbix のデータベース監視アイテムが「SQL query returned empty result」というエラーで情報の取得に失敗します。
  詳細についてはZBX-19917を参照してください。

アイテムのリクエストメソッドパラメータが正しくありません

HTTP チェックでのみ使用されるリクエストメソッドパラメータが、Zabbix 4.0 以前のバージョンからのアップグレードの結果として、すべてのアイテムでデフォルト以外の値「1」に誤って設定されている場合があります。この状況を修正する方法の詳細については、ZBX-19308を参照してください。

Web 監視と HTTP エージェント

Zabbix サーバーは、Web シナリオまたは HTTP エージェントで「SSL verify peer」が有効になっている場合、一部の Linux ディストリビューションで上流のバグによりメモリリークが発生します。詳細および利用可能な回避策については、ZBX-10486を参照してください。

シンプルチェック

fping v3.10 より前のバージョンには、重複したエコーリプレイパケットを誤って処理するバグがあります。これにより、`icmpping`、`icmppingloss`、`icmppingsec` アイテムで予期しない結果が発生する可能性があります。**fping** の最新バージョンを使用することを推奨します。詳細はZBX-11726をご覧ください。

rootless コンテナでの fping 実行時のエラー

コンテナが rootless モードまたは特定の制限環境で実行されている場合、ICMP チェックを実行する際に fping の実行に関連するエラー（例：`fping: Operation not permitted` や、すべてのリソースへのパケットが失われる）に直面することがあります。

この問題を修正するには、”docker run” または”podman run” コマンドに`--cap-add=net_raw` を追加してください。

さらに、非 root 環境で fping を実行するには sysctl の変更が必要な場合があります。例：

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

ここで”1995” は zabbix の GID です。詳細については、ZBX-22833を参照してください。

SNMP チェック

OpenBSD オペレーティングシステムを使用している場合、Net-SNMP ライブラリの 5.7.3 バージョンまでの use-after-free バグにより、Zabbix サーバーの設定ファイルで SourceIP パラメータが設定されていると、Zabbix サーバーがクラッシュする可能性があります。回避策として、SourceIP パラメータを設定しないでください。同じ問題は Linux にも当てはまりますが、Zabbix サーバーの動作停止は引き起こしません。OpenBSD では、net-snmp パッケージにローカルパッチが適用されており、OpenBSD 6.3 でリリースされる予定です。

SNMP データのスパイク

SNMP データにスパイクが発生することがあり、これは主電源の電圧スパイクなどの物理的要因に関連している可能性があります。詳細はZBX-14318を参照してください。

SNMP トラップ

SNMP トラップに必要な"net-snmp-perl" パッケージは、RHEL 8.0-8.2 で削除され、RHEL 8.3 で再追加されました。

そのため、RHEL 8.0-8.2 を使用している場合は、RHEL 8.3 へのアップグレードが最善の解決策です。

詳細については、ZBX-17192 も参照してください。

RHEL 7 でのアラートプロセスのクラッシュ

RHEL 7 で Zabbix サーバーのアラートプロセスがクラッシュする事例が報告されています。詳細については、ZBX-10461 を参照してください。

Zabbix agent 2 (6.0.5 以前) のアップグレード

パッケージから Zabbix agent 2 (バージョン 6.0.5 以前) をアップグレードする際、プラグイン関連のファイル競合エラーが発生する場合があります。このエラーを修正するには、必要に応じて agent 2 の設定をバックアップし、agent 2 をアンインストールして再インストールしてください。

RHEL 系システムでは、以下を実行します:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Debian 系システムでは、以下を実行します:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

詳細はZBX-23250を参照してください。

フロントエンドのロケールが切り替わる

フロントエンドのロケールが明確な理由もなく切り替わる、つまり、あるページ（またはページの一部）がある言語で表示され、他のページ（またはページの一部）が別の言語で表示されることがあることが確認されています。この問題は、複数のユーザーがいて、そのうちの一部があるロケールを使用し、他のユーザーが別のロケールを使用している場合によく発生します。

この問題の既知の回避策は、PHP および Apache でマルチスレッドを無効にすることです。

この問題は、PHP でのロケールの設定の仕組みに関連しています。ロケール情報はスレッドごとではなく、プロセスごとに保持されます。したがって、マルチスレッド環境では、同じ Apache プロセスで複数のプロジェクトが実行されている場合、別のスレッドでロケールが変更され、その結果 Zabbix のスレッドでデータの処理方法が変わる可能性があります。

詳細については、関連する問題レポートを参照してください:

- ZBX-10911（フロントエンドのロケールが切り替わる問題）
- ZBX-16297（BC Math 関数の `bcdiv` 関数を使用したグラフでの数値処理の問題）

グラフ

サマータイム

サマータイム (DST) の変更により、X 軸ラベルの表示に不規則性 (日付の重複、日付の欠落など) が発生します。

合計集計

グラフで合計集計を 1 時間未満の期間で使用する場合、トレンドからデータが取得されると、グラフに不正な (乗算された) 値が表示されます。

テキストの重なり

一部のフロントエンド言語 (例: 日本語) では、ローカルフォントによりグラフの凡例でテキストが重なる場合があります。これを回避するには、PHP GD 拡張モジュールのバージョン 2.3.0(またはそれ以降) を使用してください。

ログファイルの監視

ファイルシステムが 100% フルで、ログファイルが追記されている場合、`log[]` および `logrt[]` アイテムはログファイルの先頭から繰り返し再読み込みします（詳細は ZBX-10884 を参照してください）。

遅い MySQL クエリ

Zabbix サーバーは、アイテムの値が存在しない場合に遅い `SELECT` クエリを生成します。この問題は MySQL 5.6/5.7 バージョンで発生することが知られています（詳細な議論についてはZBX-10652を参照）。また、特定の場合にはそれ以降の MySQL バージョンでも発生する可能性があります。これに対する回避策としては、MySQL で`index_condition_pushdown`または`prefer_ordering_index`オプティマイザを無効にすることです。ただし、この回避策ですべての遅いクエリに関連する問題が解決するわけではないことに注意してください。

リンクからのフィルター設定の永続化

フィルター設定（時間セレクターを含む）を含む Zabbix フロントエンドページへのリンクを開くと、そのフィルターは自動的にデータベースにユーザーごとに保存され、そのページの以前に保存されたフィルターや時間セレクターの設定が置き換えられます。これらの設定は、ユーザーが手動で更新またはリセットするまで有効です。

SNMPv3 トラップにおける IPv6 アドレスの問題

net-snmp のバグにより、SNMP トラップで SNMPv3 を使用する場合、IPv6 アドレスが正しく表示されないことがあります。詳細および回避策については、ZBX-14541を参照してください。

失敗したログイン情報での長い IPv6 IP アドレスの切り捨て

失敗したログイン試行メッセージでは、データベースフィールドの文字数制限が 39 文字のため、格納された IP アドレスの最初の 39 文字のみが表示されます。つまり、39 文字を超える IPv6 IP アドレスは不完全に表示されます。

Windows での Zabbix エージェントチェック

Zabbix エージェントの設定ファイル（zabbix_agentd.conf）の `Server` パラメータに存在しない DNS エントリがあると、Windows 上で Zabbix エージェントの応答時間が長くなる場合があります。これは、Windows の DNS キャッシュデーモンが IPv4 アドレスの否定的な応答をキャッシュしないために発生します。ただし、IPv6 アドレスの場合は否定的な応答がキャッシュされるため、これに対する回避策としては、ホストで IPv4 を無効にすることが考えられます。

YAML エクスポート/インポート

YAML のエクスポート/インポートには、いくつか既知の問題があります。

- エラーメッセージが翻訳できません。
- .yaml ファイル拡張子の有効な JSON がインポートできない場合があります。
- クォートされていない人間が読める日付が自動的に Unix タイムスタンプに変換されます。

SUSE で NGINX と php-fpm を使用したセットアップウィザード

フロントエンドのセットアップウィザードは、NGINX + php-fpm を使用した SUSE で設定ファイルを保存できません。これは、/usr/lib/systemd/system/php-fpm.service ユニットの設定によるもので、Zabbix が/etc に書き込むことを防ぎます。(PHP 7.4で導入)

回避策としては、以下の 2 つのオプションがあります。

- php-fpm の systemd ユニットのProtectSystemオプションを'full' ではなく'true' に設定する。
- /etc/zabbix/web/zabbix.conf.php ファイルを手動で保存する。

認証ヘッダーの転送

Apache または NGINX が、API リクエスト内の認証ヘッダーを Zabbix に転送できない場合があります。これにより、Zabbix API や Okta を使用した SAML といったシングルサインオン (SSO) サービスの使用時に認証の問題が発生する可能性があります。

この問題を解決するには、Web サーバーの設定を更新してください。

Apache をリバースプロキシ (CGI 以外の設定) として使用する場合は、(RHEL ベースシステムの場合) `/etc/httpd/conf/httpd.conf` または (Debian/Ubuntu の場合) `/etc/apache2/apache2.conf` に次のディレクティブを追加してください。

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Apache がリクエストを処理するためにスクリプトを直接実行する場合 (例: mod_cgi を使用)、代わりに次のディレクティブを追加してください。

```
CGIPassAuth On
```

一方、**NGINX** は Authorization ヘッダーを自動的に処理します。ただし、NGINX がリバースプロキシとして動作している場合は、`/etc/nginx/nginx.conf` (Zabbix フロントエンドの場所) に以下のディレクティブを追加することで、Authorization ヘッダーを明示的に転送できます。

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

設定を更新したら、Web サーバーを再起動してください。

詳細については、以下を参照してください:

- ZBX-22952
- Apache 2.4 + PHP-FPM および Authorization ヘッダー
- SetEnvIfNoCaseおよびCGIPassAuthディレクティブ
- NGINX リバースプロキシ

Ubuntu 20 での Zabbix Web サービス用 Chromium

ほとんどの場合、Zabbix Web サービスは Chromium で動作しますが、Ubuntu 20.04 で Chromium を使用すると、次のエラーが発生します。

```
Cannot fetch data: chrome failed to start:cmd_run.go:994:
WARNING: cannot create user data directory: cannot create
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

このエラーは、ユーザー'zabbix' のホームディレクトリとして/var/lib/zabbix が使用されているために発生します。

MySQL のカスタムエラーコード

Zabbix はバックエンドデータベースにアクセスできないことを検出すると、通知を送信し、接続を試み続けます。特定のデータベースエンジンでは、特定のエラーコードが認識されます。MySQL では、これらの認識されるエラーコードには以下が含まれます:

- CR_CONN_HOST_ERROR
- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

さらに、Azure 上の MySQL インストールで Zabbix を使用している場合、Zabbix ログに一般的なエラーメッセージ *[9002] Some errors occurred* が表示されることがあります。このメッセージは、データベースから Zabbix サーバーまたはプロキシに送信されます。エラーの原因を特定するには、Azure のログを参照してください。

PCRE2 への切り替え後の無効な正規表現

Zabbix 6.0 で PCRE2 のサポートが追加されました。PCRE も引き続きサポートされていますが、RHEL 7 以降、SLES（全バージョン）、Debian 9 以降、Ubuntu 16.04 以降の Zabbix インストールパッケージは PCRE2 を使用するように更新されています。多くの利点がある一方で、PCRE2 への切り替えにより、既存の PCRE 正規表現パターンの一部が無効になったり、異なる動作をする場合があります。特に、パターン *^[\w-\.]* に影響します。この正規表現を意味を変えずに再び有効にするには、式を *^[-\w\.]* に変更してください。これは、PCRE2 がダッシュ記号を区切り文字として扱い、文字クラス内で範囲を作成するために発生します。

Geomap ウィジェットのエラー

NGINX を使用している古い Zabbix バージョンからアップグレードし、アップグレード時に新しい NGINX 設定ファイルに切り替えなかった場合、Geomap ウィジェットの地図が正しく読み込まれないことがあります。

この問題を解決するには、古い設定ファイルを破棄し、現在のバージョンパッケージの設定ファイルを使用し、ダウンロード手順の *e. Zabbix* フロントエンド用の *PHP* の設定セクションに記載されているように再設定してください。

または、既存の NGINX 設定ファイル（通常は*/etc/zabbix/nginx.conf*）を手動で編集することもできます。その場合は、ファイルを開き、次のブロックを探します。

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

次に、このブロックを以下のように置き換えます。

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

前処理 — グローバル変数は安全ではありません

前処理の JavaScript はリクエストごとに実行されますが、宣言されていない識別子への代入（例: `secret = value`）は暗黙的なグローバル変数を作成し、現在の実行を超えて持続する可能性があります。暗黙的なグローバル変数に機密データ（トークン、パスワードなど）を保存すると、後続の前処理実行や同じ環境で実行される他の統合によって、偶発的な漏洩や再利用のリスクが高まります。

暗黙的なグローバル変数に依存しないでください。常に `var` や `const` で変数を宣言し、グローバルオブジェクト（例: `globalThis` や `window`）に秘密情報を付与しないでください。前処理内から組み込みのグローバルオブジェクトを上書きするサポートされた方法はありません。

安全な例:

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

Windows のプロセッサグループ

Microsoft のドキュメントによると、64 個未満の論理プロセッサを持つシステムには常に 1 つのプロセッサグループ（グループ 0）しかありません。しかし、Zabbix ユーザーからは、64 個以下の論理プロセッサを持つシステムで 2 つのプロセッサグループが存在するという稀なバグZBX-20260が報告されています。この結果、2 つのプロセッサグループのうち 1 つのプロセッサグループに対してのみ"\Processor(n)" パフォーマンスカウンタが存在することになります。このバグの実際の根本原因は不明です。しかし、stackoverflow.comで同様のケースが説明されており、そこでは BIOS と Windows の相互運用に根本原因がありました。

utf8mb4 照合順序によるフィルタリングの制限

フィルタ（例：データ収集 > メンテナンス）は、特定の Unicode 文字（例：𝓮、𝔢）を含むエンティティに適用した場合、正しく機能しないことがあります。この問題は、MySQL または MariaDB データベースのデフォルトの utf8mb4_bin 照合順序が、Unicode 文字のソートや比較をどのように処理するかによって発生します。

この制限に対処するには、データベースのカラムの照合順序を utf8mb4_0900_bin、utf8mb4_0900_ai_ci、utf8mb4_unicode_520_ci などの代替案に変更することができます。ただし、照合順序を変更すると、空白の扱いや他の文字のソートやフィルタリングで予期しない動作が発生する可能性があることに注意してください。

照合順序の変更についての詳細は、MySQL ドキュメントまたはMariaDB ドキュメントを参照してください。照合順序の違いについては、MySQL ドキュメントのUnicode 文字セットを参照してください。

MariaDB 10.5.1–10.5.9 での UI 要素へのアクセス

Super Admin 以外のロールで Zabbix ウェブフロントエンドにアクセスすると、「システムエラーが発生しました。Zabbix 管理者に連絡してください。」というメッセージが表示される場合があります。この問題は、MariaDB のバージョン 10.5.1 から 10.5.9 を使用しているインストールに影響します。

この問題を回避するには、MariaDB を 10.5.9 以降のバージョンにアップデートしてください。詳細はZBX-25746を参照してください。

tcmalloc による過剰なメモリ使用量のプロファイリング

Zabbix のインストールでメモリ使用量が多すぎると疑われる場合は、tcmallocのメモリプロファイリング機能を使用して、Zabbix サーバー/プロキシのメモリ消費を調査できます。

1. Zabbix をソースからインストールする際に、追加のフラグを設定します:

```
export CFLAGS="-std=gnu99 -g -O0"
```

`-std=gnu99` フラグは、Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントのビルドに必要です。`-g` フラグは追加のデバッグ情報を付加し、`-O0` は最適化を無効にします。最適化は tcmalloc のプロファイリングに干渉する可能性があります。

2. Zabbix サーバーを起動する前に、以下の環境変数を設定します。これらの変数は、tcmalloc にメモリ使用量の追跡とレポート方法を指示します:

```
LD_PRELOAD="/usr/lib/aarch64-linux-gnu/libtcmalloc.so" \
HEAPPROFILE=./heap_profile \
HEAP_PROFILE_ALLOCATION_INTERVAL=0 \
HEAP_PROFILE_INUSE_INTERVAL=4294967296 \
HEAPPROFILESIGNAL=5 \
MALLOCSTATS=1 \
./sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

3. シグナル 5 を対象プロセスに送信してプロファイルダンプをトリガーします。1234 を実際のプロセス ID（PID）に置き換えてください:

```
kill -5 1234
```

4. 生成されたプロファイルを表示します:

```
pprof-symbolize -text ./sbin/zabbix_server ./heap_profile.0001.heap

Using local file ./sbin/zabbix_server.
Using local file ./heap_profile.0001.heap.
Total: 1078.1 MB
  1076.8  99.9%  99.9%   1076.8  99.9% zbx_malloc2
     1.0   0.1% 100.0%      1.0   0.1% __GI___strdup
     0.2   0.0% 100.0%      0.2   0.0% CRYPTO_zalloc@@OPENSSL_3.0.0
     0.1   0.0% 100.0%      0.1   0.0% OPENSSL_LH_insert@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% zbx_realloc2
     0.0   0.0% 100.0%      0.1   0.0% PKCS7_decrypt@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% find_best_tree_node
     0.0   0.0% 100.0%      0.0   0.0% CRYPTO_strndup@@OPENSSL_3.0.0
     ...
     0.0   0.0% 100.0%      0.0   0.0% preprocessing_flush_value
     0.0   0.0% 100.0%   1074.0  99.6% preprocessor_add_request
```

この例では、zbx_malloc2 がほぼすべてのメモリアロケーションの原因となっています。

関連情報:

- 関連する問題報告については、ZBX-25050およびZBX-25584を参照してください。
- コンパイルオプション（-std=gnu99、-g、-O0 など）については、GCC Option Summaryを参照してください。
- tcmalloc プロファイリング用の環境変数については、Gperftools Heap Profilerのドキュメントを参照してください。

マルチプライマリモードでの MySQL 8.0 Group Replication

マルチプライマリモードで MySQL 8.0 Group Replication を使用している場合、トランザクションコミット時に次のようなエラーが発生することがあります。

```
1531697:20250128:064734.697 query [txnlev:1] [update alerts set status=1,retries=0,error='' where alertid=
1531697:20250128:064734.713 query [txnlev:1] [commit;]
1531697:20250128:064734.753 [Z3005] query failed: [3101] Plugin instructed the server to rollback the curr
```

このエラーは、外部キー制約を含むロールバック操作の問題によって引き起こされるようです。

関連情報:

- ZBX-26060 - 関連する問題レポート
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" - 上流の問題

要件

ハードウェア要件

メモリ

Zabbix には物理メモリとディスクメモリの両方が必要です。必要なディスクメモリの量は、監視するホスト数やパラメータ数によって異なります。監視パラメータの履歴を長期間保存する予定がある場合は、データベースに履歴を保存するために少なくとも数ギガバイトの空き容量を確保する必要があります。各 Zabbix デーモンプロセスは、データベースサーバーへの複数の接続を必要とします。接続に割り当てられるメモリの量は、データベースエンジンの設定によって異なります。

> **Note:**
> 物理メモリが多いほど、データベース（および Zabbix）の動作が高速になります。

CPU

Zabbix、特に Zabbix データベースは、監視するパラメータの数や選択したデータベースエンジンによっては、かなりの CPU リソースを必要とする場合があります。

その他のハードウェア

Zabbix で SMS 通知サポートを使用するには、シリアル通信ポートとシリアル GSM モデムが必要です。USB-シリアル変換器も動作します。

ハードウェア構成の例

この表は、**Linux/BSD/Unix** プラットフォームを前提としたハードウェア構成の例を示しています。

これらは、開始時の規模とハードウェア構成の例です。各 Zabbix インストールは固有のものです。Zabbix インストールを本番環境に展開する前に、ステージングまたは開発環境で Zabbix システムのパフォーマンスをベンチマークし、要件を十分に理解してください。

インストール規模	監視メトリック数[1]	CPU/vCPU コア数	メモリ (GiB)	データベース	Amazon EC2[2]
小規模	1,000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
中規模	10,000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
大規模	100,000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
超大規模	1,000,000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] 1 メトリック = 1 アイテム + 1 トリガー + 1 グラフ
 [2] Amazon の汎用 EC2 インスタンスの例。ARM64 または x86_64 アーキテクチャを使用し、Zabbix インストールの評価およびテスト時に、Compute/Memory/Storage 最適化など適切なインスタンスタイプを選択してから本番環境にインストールしてください。

Note:
実際の構成は、アクティブアイテム数やリフレッシュレートに大きく依存します（詳細はこのページのデータベースサイズセクションを参照してください）。大規模なインストールの場合は、データベースを別サーバーで実行することを強く推奨します。

サポートされているプラットフォーム

セキュリティ要件と監視サーバーのミッションクリティカルな性質により、UNIX は必要なパフォーマンス、フォールトトレランス、およびレジリエンスを一貫して提供できる唯一のオペレーティングシステムです。Zabbix は市場をリードするバージョンで動作します。

Zabbix コンポーネントは、以下のプラットフォームで利用可能でテストされています。

プラットフォーム	サーバー	エージェント	エージェント 2	コメント
Linux	x	x	x	
Windows	-	x	x	Zabbix エージェントは、Windows XP (64-bit)/Server 2003 以降のすべてのデスクトップおよびサーバーバージョンでサポートされています。 Zabbix エージェント 2 は、サポートされている Go バージョンでのみコンパイルされ、重大なセキュリティ脆弱性を防ぐため、Windows 10 (32-bit)/Server 2016 以降のすべてのデスクトップおよびサーバーバージョンでサポートされています。Go 1.21 以降、必要な最小 Windows バージョンが引き上げられ、Windows 10/Server 2016 が Zabbix エージェント 2 の最小バージョンとなります。
macOS	x	x	-	
IBM AIX	x	x	-	Zabbix エージェントは、AIX 6.1 TL07 / 7.1 TL01 未満のプラットフォームでは動作しません。
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	

プラットフォーム	サーバー	エージェント	エージェント 2	コメント
HP-UX	x	x	-	

Note:
Zabbix サーバー/エージェントは、他の Unix 系オペレーティングシステムでも動作する場合があります。

Attention:
Zabbix は、暗号化を有効にしてコンパイルされた場合、コアダンプを無効にし、コアダンプの無効化を許可しないシステムでは起動しません。

必要なソフトウェア

Zabbix は、最新の Web サーバー、主要なデータベースエンジン、PHP スクリプト言語を中心に構築されています。

サードパーティの外部周辺ソフトウェア

必須と記載されている場合は、必要なソフトウェア/ライブラリが厳密に必要です。オプションのものは、特定の機能をサポートするために必要です。

ソフトウェア	必須ステータス	サポートされているバージョン	コメント
MySQL/Percona	いずれか	8.4.0-9.5.X	MySQL (または Percona) を Zabbix のバックエンドデータベースとして使用する場合に必要です。InnoDB エンジンが必要です。 サーバー/プロキシのビルドには C API (libmysqlclient) ライブラリの使用を推奨します。
MariaDB		10.11.00-12.0.X	InnoDB エンジンが必要です。 推奨バージョンは 11.4 です。 サーバー/プロキシのビルドには MariaDB Connector/C ライブラリの使用を推奨します。 また、MariaDB でのデッドロックの可能性 およびMariaDB 10.5.1–10.5.9 での UI 要素へのアクセス も参照してください。
PostgreSQL		15.0-18.X	PostgreSQL を Zabbix のバックエンドデータベースとして使用する場合に必要です。 インストール規模によっては、特定の操作でデータベースが使用するメモリ量が十分であり、クエリの実行に時間がかかりすぎないように、PostgreSQL の *work_mem* 設定プロパティ (デフォルト値は 4MB) を増やす必要がある場合があります。
PostgreSQL 用 *TimescaleDB*		2.20.X-2.23.X	TimescaleDB を PostgreSQL データベース拡張として使用する場合に必要です。圧縮をサポートする TimescaleDB Community Edition をインストールしてください。 PostgreSQL 15 は TimescaleDB 2.10 以降でサポートされています。 PostgreSQL と TimescaleDB のバージョン互換性の詳細については、TimescaleDB ドキュメント も参照してください。
SQLite	オプション	3.3.5-3.34.X	SQLite は Zabbix プロキシでのみサポートされています。Zabbix プロキシデータベースとして SQLite を使用する場合に必要です。
smartmontools		7.1 以降	Zabbix agent 2 で必要です。
who			ユーザー数プラグインで必要です。
dpkg			system.sw.packages プラグインで必要です。
pkgtool			system.sw.packages プラグインで必要です。
rpm			system.sw.packages プラグインで必要です。
pacman			system.sw.packages プラグインで必要です。
q applets			Gentoo Linux で system.sw.packages プラグインを使用する場合、q applets の一部である `qlist` および `qsize` が必要です。

Note:
Zabbix はオペレーティングシステムで利用可能なデータベースでも動作しますが、最良の体験のために、公式のデータベース開発者リポジトリからインストールされたデータベースの使用を推奨します。

フロントエンド

Zabbix フロントエンドの最小サポート画面幅は 1200px です。

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必要です。オプションのものは、特定の機能をサポートするために必要です。

ソフトウェア	必須ステータス	サポートバージョン	コメント
PHP	はい	8.0.0 - 8.4.X	
Apache	いずれか	2.4 以降	
Nginx		1.20 以降	
MySQL	いずれか	サードパーティ外部周辺ソフトウェアを参照	
PostgreSQL			
PHP 拡張			
mysqli	はい		MySQL を Zabbix バックエンドデータベースとして使用する場合に必要です。
pgsql			PostgreSQL を Zabbix バックエンドデータベースとして使用する場合に必要です。
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*)；ユーザースクリプトサポートに必要です。
gd		2.0.28 以降	php-gd（ディストリビュータが別パッケージとして提供している場合）；PHP GD 拡張は PNG 画像（*--with-png-dir*）、JPEG 画像（*--with-jpeg-dir*）、FreeType 2（*--with-freetype-dir*）をサポートしている必要があります。バージョン 2.3.0 以降は、一部のフロントエンド言語でグラフのテキスト重なりを回避するために必要な場合があります。
libxml		2.6.15 以降	php-xml（ディストリビュータが別パッケージとして提供している場合）
xmlwriter			php-xmlwriter（ディストリビュータが別パッケージとして提供している場合）
xmlreader			php-xmlreader（ディストリビュータが別パッケージとして提供している場合）
ctype			php-ctype (*--enable-ctype*)
session			php-session（ディストリビュータが別パッケージとして提供している場合）
ldap	いいえ		php-ldap；LDAP 認証に必要です。
openssl			php-openssl；SAML 認証に必要です。
gettext			php-gettext (*--with-gettext*)；翻訳に必要です。
cURL		7.19.4 以降	php-curl；Duo Universal Prompt MFAおよびSMTP 認証に必要です。

Zabbix に同梱されているサードパーティ製フロントエンドライブラリ：

ライブラリ	必須ステータス	最小バージョン	コメント
jQuery JavaScript Library	はい	3.6.0	クロスブラウザ開発を簡素化する JavaScript ライブラリ。
jQuery UI		1.12.1	jQuery 上に構築されたユーザーインターフェースのインタラクション、エフ
SAML PHP Toolkit		4.0.0	Zabbix へのサインインを可能にする SAML 2.0 認証サポートを追加する PHP
Symfony Yaml Component		5.1.0	Zabbix 設定要素の YAML 形式でのエクスポートおよびインポートをサポート

Note:
Zabbix は、以前のバージョンの Apache、MySQL、PostgreSQL でも動作する場合があります。

Attention:
デフォルトの DejaVu 以外のフォントを使用する場合、PHP 関数imagerotateが必要になる場合があります。この関数がない場合、グラフ表示時にこれらのフォントが正しく表示されないことがあります。この関数は、PHP がバンドルされた GD でコンパイルされている場合のみ利用可能であり、Debian や他のディストリビューションではそうではありません。

Zabbix フロントエンドコードの作成およびデバッグに使用されるサードパーティ製ライブラリ：

ライブラリ	必須ステータス	最小バージョン	説明
Composer	いいえ	2.4.1	PHP の依存関係および必要なライブラリを管理するための標準フォーマットを提供する、PHP 用のアプリケーションレベルのパッケージマネージャ。
PHPUnit		8.5.29	Zabbix フロントエンドのテスト用 PHP ユニットテストフレームワーク。
SASS		3.4.22	カスケーディングスタイルシート（CSS）に解釈およびコンパイルされるプリプロセッサスクリプト言語。

クライアント側の Web ブラウザ

Cookie と JavaScript を有効にする必要があります。

Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari、Opera の最新の安定バージョンがサポートされています。

Warning:
IFrame の同一生成元ポリシーが実装されているため、Zabbix を異なるドメインのフレームに配置することはできません。

 ただし、Zabbix フレームに配置されたページは、フレームに配置されたページと Zabbix フロントエンドが同じドメインにある場合、Zabbix フロントエンドに（JavaScript を介して）アクセスできます。`http://secure-zabbix.com/cms/page.html` のようなページを `http://secure-zabbix.com/zabbix/`のダッシュボードに配置すると、Zabbix への完全な JS アクセスが可能になります。

サーバー/プロキシ

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必要です。オプションのものは、特定の機能をサポートするために必要です。

要件	必須ステータス	説明
libpcre2	はい	Perl Compatible Regular Expression（PCRE）サポートに PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。
libevent		プロセス間通信に必要です。バージョン 2.0.10 以上。
libevent-pthreads		プロセス間通信に必要です。
libpthread		ミューテックスおよびリード/ライトロックサポートに必要です（libc の一部である場合があります）。
libresolv		DNS 解決に必要です（libc の一部である場合があります）。
libiconv		テキストエンコーディング/フォーマット変換に必要です（libc の一部である場合があります）。Linux 上の Zabbix サーバーでは必須です。
libz		圧縮サポートに必要です。
libm		数学ライブラリ。Zabbix サーバーのみで必要です。
libmysqlclient	いずれか	MySQL を使用する場合に必要です。
libmariadb		MariaDB を使用する場合に必要です。
libpq5		PostgreSQL を使用する場合に必要です。*libpq5* のバージョンは、使用する PostgreSQL データベースのバージョンと同じかそれ以上である必要があります。
libsqlite3		SQLite を使用する場合に必要です。Zabbix プロキシのみで必要です。
libOpenIPMI	いいえ	IPMI サポートに必要です。Zabbix サーバーのみで必要です。
libssh2 または *libssh*		SSH チェックに必要です。バージョン 1.8.0 以上（libssh2）；0.9.0 以上（libssh）。

要件	必須ステータス	説明
libcurl		以下の機能に必要です： - Web 監視、VMware 監視、およびHTTP エージェントアイテム（すべて：バージョン 7.19.1 以上）； - Zabbix エージェントのweb.page.*アイテム（バージョン 7.19.1 以上；エージェントの要件も参照）； - SMTP 認証（Basic：バージョン 7.20.0 以上；OAuth：バージョン 7.33 以上；フロントエンドの要件も参照）； - Elasticsearch（バージョン 7.28.0 以上）。 すべての機能でバージョン 7.28.0 以上を推奨します。 `web.page.*` アイテムでアップグレードされた cURL 機能を使用するには、Zabbix サーバー/プロキシを再起動してください。 SMTP 認証には、実行時に `libcurl-full` パッケージを使用してください。
libxml2		VMware 監視および XML XPath 前処理に必要です。
net-snmp		SNMP サポートに必要です。バージョン 5.3.0 以上。 強力な暗号化プロトコル（AES192/AES192C、AES256/AES256C）のサポートは、net-snmp ライブラリ 5.8 以降で利用可能です。RHEL 8+ ベースのシステムでは、net-snmp 5.8.15 以降の使用を推奨します。
libunixodbc		データベース監視に必要です。
libgnutls または *libopenssl*		暗号化を使用する場合に必要です。 最小バージョン：*libgnutls* - 3.1.18、*libopenssl* - 1.0.1
libldap		LDAP サポートに必要です。
fping		ICMP ping アイテムに必要です。
c-ares		Zabbix が`--with-ares` オプションで構成されている場合、非同期 DNS 解決に必要です。それ以外の場合は *libevent* が使用されます。 最小バージョン：1.16.0

エージェント

要件	必須ステータス	説明
libpcre2	はい	Perl Compatible Regular Expression (PCRE) サポートに PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。 ログ監視に必要です。Windows でも必要です。
libpthread		ミューテックスおよびリード/ライトロックのサポートに必要です（libc の一部である場合があります）。Windows では不要です。
libresolv		DNS 解決に必要です（libc の一部である場合があります）。Windows では不要です。
libiconv		ログアイテム、ファイル内容、ファイル正規表現および regmatch アイテムで UTF-8 へのテキストエンコーディング/フォーマット変換に必要です（libc の一部である場合があります）。Windows では不要です。
libgnutls または *libopenssl*	いいえ	暗号化を使用する場合に必要です。 最小バージョン: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Microsoft Windows では OpenSSL 1.1.1 以降が必要です。
libldap		LDAP を使用する場合に必要です。Windows ではサポートされていません。
libcurl		Zabbix エージェントのweb.page.*アイテムの拡張サポートに必要です。 libcurl がない場合、基本的な機能（例: web.page.get[http://example.com]）が利用できます。libcurl がある場合、エージェントは認証情報付き HTTP URL（例: http://user:password@example.com）や HTTPS URL などの追加機能をサポートします。 バージョン 7.19.1 以上が必要です（7.28.0 以上を推奨）。 アップグレードされた cURL 機能を使用するには、Zabbix エージェントを再起動してください。
libmodbus		Modbus 監視を使用する場合のみ必要です。 バージョン 3.0 以上。

Agent 2

要件	必須ステータス	説明
Go	はい	Zabbix agent 2 およびそのプラグインをソースからビルドするために必要です。 Go 1.24.10 以降がサポートされています。インストール手順についてはgo.devを参照してください。 Zabbix agent 2 およびそのプラグインで使用される Go ライブラリは、Zabbix Git リポジトリにリストされています（リポジトリで `indirect` とマークされているライブラリは、他の必要なライブラリの依存関係です）： - Zabbix agent 2 - プラグインサポート - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - サンプルプラグイン
libpcre2	はい	Perl Compatible Regular Expression（PCRE）サポートのために PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。 ログ監視に必要です。Windows でも必要です。
libopenssl	いいえ	暗号化を使用する場合に必要です。 UNIX プラットフォームでは OpenSSL 1.0.1 以降が必要です。 OpenSSL ライブラリには PSK サポートが有効になっている必要があります。 LibreSSL はサポートされていません。 Microsoft Windows システムでは OpenSSL 1.1.1 以降が必要です。

Web サービス

Zabbix Web サービスを使用して定期レポートを生成するのに、Google Chrome の最新の安定バージョンがサポートされています。

Web サービスの構築に必要な Go のバージョンは、Zabbix エージェント 2で使用されているバージョンと同じです。

Java ゲートウェイ

Zabbix をソースリポジトリまたはアーカイブから取得した場合、必要な依存関係はすでにソースツリーに含まれています。

Zabbix をディストリビューションのパッケージから取得した場合、必要な依存関係はパッケージングシステムによってすでに提供されています。

上記のいずれの場合も、ソフトウェアはすぐに使用でき、追加のダウンロードは必要ありません。

ただし、これらの依存関係の独自バージョンを提供したい場合（たとえば、Linux ディストリビューション用のパッケージを準備している場合）、Java ゲートウェイが動作することが確認されているライブラリバージョンのリストを以下に示します。Zabbix は、これらのライブラリの他のバージョンでも動作する可能性があります。

次の表は、オリジナルコードの Java ゲートウェイに現在バンドルされている JAR ファイルを示しています。

ライブラリ	必須ステータス	最小バージョン	コメント
android-json	Yes	4.3r1	JSON（JavaScript Object Notation）は軽量なデータ交換フォーマットです。これは、Android SDK から抽出された org.json 互換の Android 実装です。
logback-classic		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

Java ゲートウェイは、Oracle Java またはオープンソースの OpenJDK（バージョン 1.6 以降）を使用してビルドできます。Zabbix が提供するパッケージは OpenJDK を使用してコンパイルされています。次の表は、ディストリビューションごとに Zabbix パッケージのビルドに使用される OpenJDK パッケージを示しています。

ディストリビューション	OpenJDK パッケージ
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)

ディストリビューション	OpenJDK パッケージ
CentOS Stream 9	`java-11-openjdk-devel` (amd64, arm64: 11.0.18.0.10-3)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4.0.1; arm64: 11.0.20.0.8-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

デフォルトのポート番号

各コンポーネントごとの開放ポートのリストは、デフォルト設定の場合に適用されます。

Zabbix コンポーネント	ポート番号	プロトコル	接続タイプ
Zabbix エージェント	10050	TCP	オンデマンド
Zabbix agent 2	10050	TCP	オンデマンド
Zabbix サーバー	10051	TCP	オンデマンド
Zabbix プロキシ	10051	TCP	オンデマンド
Zabbix Java ゲートウェイ	10052	TCP	オンデマンド
Zabbix ウェブサービス	10053	TCP	オンデマンド
Zabbix フロントエンド	80	HTTP	オンデマンド
	443	HTTPS	オンデマンド
Zabbix トラッパー	10051	TCP	オンデマンド

Note:
Zabbix の通信を有効にするためには、これらのポート番号をファイアウォールで開放する必要があります。通常、発信側の TCP 接続には明示的なファイアウォール設定は不要です。

データベースサイズ

Zabbix の設定データは固定量のディスク容量を必要とし、あまり増加しません。

Zabbix データベースのサイズは、主に保存される履歴データの量を定義する以下の変数に依存します。

- 1 秒あたりの処理値数

これは、Zabbix サーバーが毎秒受信する新しい値の平均数です。例えば、監視するアイテムが 3000 個あり、更新間隔が 60 秒の場合、1 秒あたりの値の数は 3000/60 = **50** となります。

つまり、毎秒 50 個の新しい値が Zabbix データベースに追加されることになります。

- 履歴のハウスキーパー設定

Zabbix は値を一定期間（通常は数週間または数ヶ月）保持します。各新しい値には、データとインデックスのために一定量のディスク容量が必要です。

したがって、30 日分の履歴を保持し、1 秒あたり 50 個の値を受信する場合、値の総数は (**30***24*3600)* **50** = 129,600,000、つまり約 1 億 3 千万個となります。

使用するデータベースエンジンや受信する値の種類（浮動小数点、整数、文字列、ログファイルなど）によって、1 つの値を保持するために必要なディスク容量は 40 バイトから数百バイトまで異なります。通常、数値アイテムの場合は 1 値あたり約 90 バイトです[2]。この場合、1 億 3 千万個の値を保持するには 130M * 90 バイト = **10.9GB** のディスク容量が必要です。

Note:
テキスト/ログアイテムの値のサイズは正確に予測することはできませんが、1 値あたり約 500 バイトと見込んでください。

- トレンドのハウスキーパー設定

Zabbix は、各アイテムについて 1 時間ごとの最大/最小/平均/カウント値のセットを **trends** テーブルに保持します。このデータはトレンド分析や長期間のグラフに使用されます。1 時間の期間はカスタマイズできません。

Zabbix データベースは、データベースの種類によって異なりますが、各合計値ごとに約 90 バイト必要です。トレンドデータを 5 年間保持したいと仮定します。3000 アイテムの値は、3000*24*365* **90** = **2.2GB**/年、5 年間で **11GB** 必要です。

- イベントのハウスキーパー設定

各 Zabbix イベントは約 250 バイトのディスク容量を必要とします[1]。Zabbix が 1 日に生成するイベント数を見積もるのは困難です。最悪の場合、Zabbix が 1 秒ごとに 1 つのイベントを生成すると仮定できます。

各復旧イベントごとに event_recovery レコードが作成されます。通常、ほとんどのイベントは復旧するため、イベントごとに 1 つの event_recovery レコードがあると仮定できます。つまり、イベントごとに追加で 80 バイト必要です。

オプションでイベントにはタグを付けることができ、各タグレコードは約 100 バイトのディスク容量を必要とします[1]。イベントごとのタグ数（#tags）は設定によって異なります。したがって、追加で #tags * 100 バイトのディスク容量が必要です。

つまり、3 年間イベントを保持したい場合、3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B のディスク容量が必要です[2]。

Note:
[1] イベント名、タグ、値に非 ASCII 文字が含まれる場合はさらに多くなります。
 [2] サイズの概算は MySQL に基づいており、他のデータベースでは異なる場合があります。

以下の表は、Zabbix システムに必要なディスク容量を計算するための式を示しています。

パラメータ	必要なディスク容量の計算式（バイト単位）
Zabbix 設定	固定サイズ。通常は 10MB 以下。
履歴	days*(items/refresh rate)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 refresh rate：アイテムの平均更新間隔 bytes：1 値を保持するために必要なバイト数。データベースエンジンによって異なるが、通常は ~90 バイト。
トレンド	days*(items/3600)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 bytes：1 トレンドを保持するために必要なバイト数。データベースエンジンによって異なるが、通常は ~90 バイト。
イベント	days*events*24*3600*bytes events：1 秒あたりのイベント数。最悪の場合は 1 秒あたり 1 イベント。 days：履歴を保持する日数 bytes：1 イベントを保持するために必要なバイト数。データベースエンジンによって異なるが、通常は ~330 + イベントごとの平均タグ数 *100 バイト。

したがって、必要なディスク容量の合計は次のように計算できます。

設定 + 履歴 + トレンド + イベント

ディスク容量は Zabbix インストール直後にすぐに使用されるわけではありません。データベースサイズは増加し、ハウスキーパー設定に応じてある時点で増加が止まります。

時刻同期

Zabbix が稼働しているサーバーで正確なシステム時刻を持つことは非常に重要です。ntpdは、ホストの時刻を他のマシンの時刻と同期させる最も一般的なデーモンです。Zabbix コンポーネントが稼働しているすべてのシステムで、システム時刻を同期させておくことを強く推奨します。

ネットワーク要件

各コンポーネントごとに開放する必要があるポートの一覧です（デフォルト設定の場合）。

コンポーネント	ポート
フロントエンド	http:80, https:443
サーバー	10051 (アクティブプロキシ/エージェントで使用)
アクティブプロキシ	10051
パッシブプロキシ	10051
エージェント 2	10050
トラッパー	
JavaGateway	10052
WebService	10053

Note:
Zabbix と外部通信を行うためには、これらのポート番号をファイアウォールで開放する必要があります。通常、外向きの TCP 接続には明示的なファイアウォール設定は不要です。

2 Zabbix 8.0 の新機能

Zabbix 8.0.0 は Zabbix 7.4.0 から継続し、新機能や機能強化が追加されています。

このバージョンの重大な変更点を参照してください。

テンプレート

新しいテンプレート

- Aruba CX 8300s by SNMPは、Aruba CX 8300 スイッチシリーズの SNMP ベースの監視を提供するテンプレートです。
- AWS by HTTP テンプレートセットに *AWS Backup Vault by HTTP* テンプレートが追加されました。
- Ciena 3906 by SNMPは、Ciena 3906 デバイスの監視用テンプレートです。
- Cisco Secure Firewall Threat Defense by HTTPは、REST API を使用して Cisco Secure Firewall Threat Defense デバイスの監視機能を提供するテンプレートです。
- OpenAI Platform by HTTPは、OpenAI の開発者プラットフォームを監視するためのテンプレートです。
- Ribbon SBC Edge by HTTPは、HTTP 経由で Ribbon SBC Edge（旧 SWe Lite）デバイスを監視するテンプレートです。
- Ribbon SBC SWe Core by HTTPは、HTTP 経由で SBC SWe Core デバイスを監視するテンプレート、およびRibbon SBC SWe CE by HTTPは、HTTP 経由で Ribbon SBC SWe Call Engine（CE）インスタンスを監視するテンプレートです。
- Stormshield SNS by SNMPは、SNMP 経由で Stormshield Network Security（SNS）デバイスを監視するテンプレートです。
- VeloCloud SD-WAN Edge by HTTPは、HTTP 経由で VeloCloud SD-WAN Edge デバイスを監視するテンプレートです。
- Vyatta Virtual Router by SNMPは、Vyatta 1908e 仮想ルーターを監視するテンプレートです。

テンプレートの更新

- Proxmox VE by HTTP テンプレートがネストされた LLD 機能で更新されました。さらに、パーセンテージを表示するアイテムの単位フォーマットが明確になるよう調整されました。
- MySQL by Zabbix agent、MySQL by Zabbix agent 2、MySQL by Zabbix agent active、MySQL by Zabbix agent 2 active、MySQL by ODBCは、`SHOW SLAVE STATUS`（旧構文）と `SHOW REPLICA STATUS`（新構文）の両方をサポートするように更新されました。
- VeloCloud SD-WAN by HTTPは、以前の *VMWare SD-WAN VeloCloud by HTTP* から名称が変更され、最新の VeloCloud SD-WAN プラットフォームに合わせて更新されました。

フロントエンド

自動登録フォームでのインラインバリデーション　フロントエンドの自動登録アクション設定フォームでインラインバリデーションがサポートされました。入力エラーはフィールド入力後すぐに表示されるため、使いやすさが向上し、設定ミスを減らすことができます。

シングルサインオン用のインポート可能な **SAML** 証明書　スーパー管理者ユーザーは、SAML設定のために証明書と秘密鍵をフロントエンドから直接インポートできるようになりました。管理 > 認証 > *SAML* に 3 つの新しいフィールドが追加されました:

- *IdP* 証明書 - アイデンティティプロバイダーによって提示される X.509 証明書

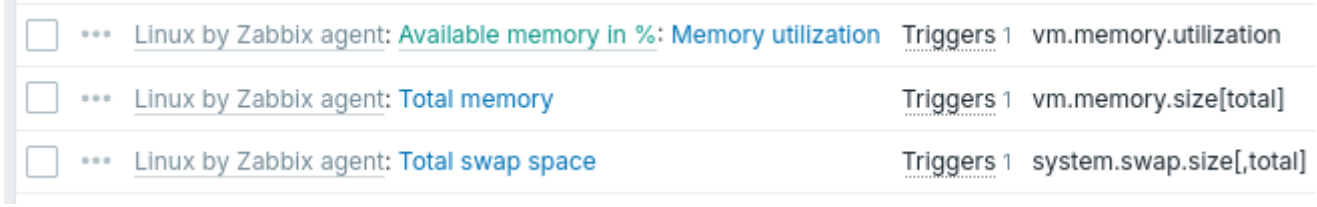
- *SP* 証明書 - SAML 交換に使用されるサービスプロバイダー証明書
- *SP* 秘密鍵 - SP 証明書に対応する秘密鍵

これらのコントロールにより、SAML 設定タブから新しい値の追加や既存値の変更が可能です。証明書と秘密鍵は、選択したストレージバックエンドに保存される前に検証され、不正または不正な値は説明付きのエラーで拒否されます。

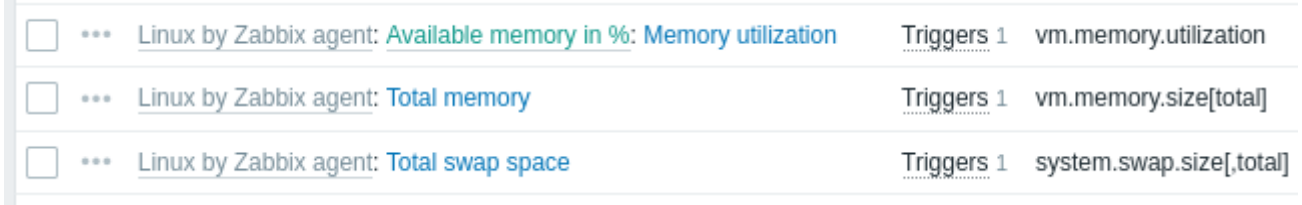
ディスカバリーフォームの **JSON** への変換チェックボックス　ディスカバリールールフォームとディスカバリープロトタイプフォームに新しい *JSON* に変換チェックボックスが追加されました。これは、タイプドロップダウンで"HTTP エージェント" が選択されている場合に表示されます。このオプションを有効にすると、取得したデータを自動的に JSON に変換してからさらに処理することができます。

新しいフォントと埋め込みフォント　Zabbix には、新しい埋め込みフォントが追加され、可読性が向上し、読み込みが速くなり、追加のフロントエンド言語が最小限のレイアウト影響でレンダリングされます。埋め込みフォントであるため、システム間で一貫した外観も保証されます。

新しいフォント:



従来のフォント:



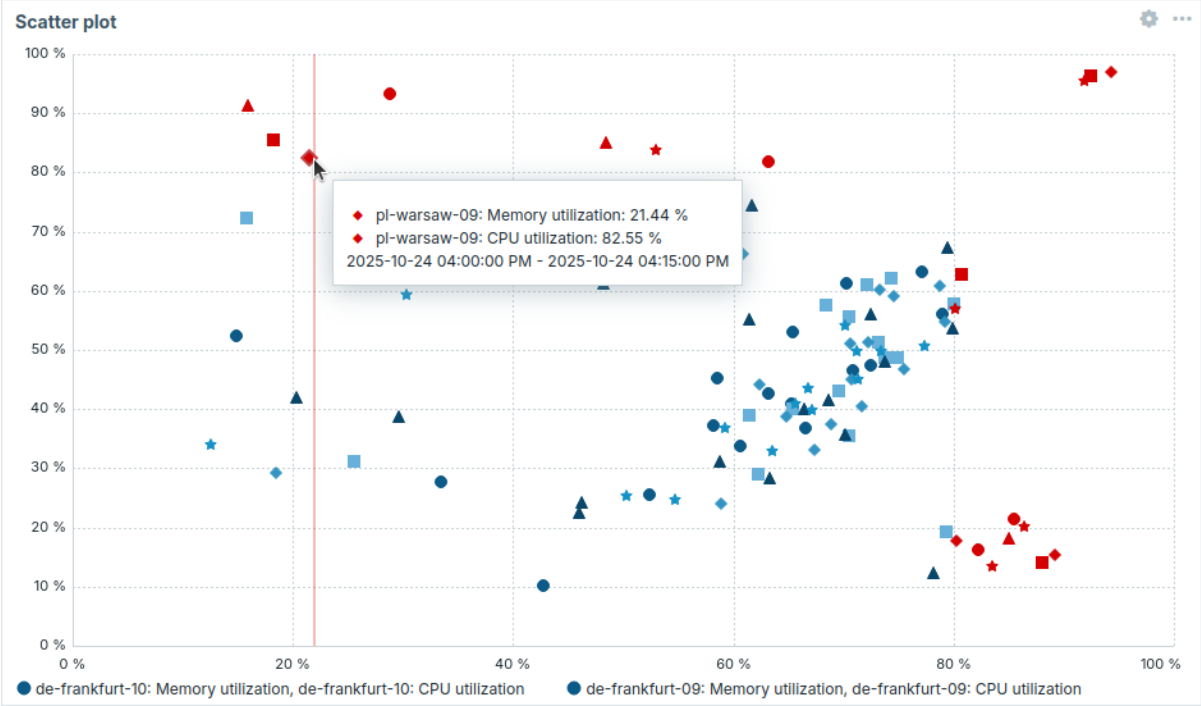
新しいフォントは、ほぼすべてのテーマで使用されています。必要に応じて、新たに追加された *Blue (classic)* および *Dark (classic)* テーマが従来のフォントで利用できます。

等幅フォントやグラフ用のフォントファミリーは変更されていません。

モーダルフォーム　低レベルディスカバリのセットアップでは、ホストプロトタイプの設定フォームがモーダル（ポップアップ）ウィンドウで開くようになりました。

ウィジェット

散布図　散布図ウィジェットがダッシュボードウィジェットに追加されました。このウィジェットは、X 軸と Y 軸に沿って個々のデータポイントをプロットすることで、2 つのメトリクス間の関係を表示します。これにより、データセット内のパターン、クラスター、相関関係、外れ値を明らかにするのに役立ちます。



ジオマップホストマーカーのクラスタリング　ジオマップウィジェットは、近くのホストマーカーをカウント付きの単一マーカーにまとめる方法を制御するクラスタリングパラメータをサポートするようになりました。クラスタリングのためのマップズームレベルを設定することで、大きなマップの可読性を保ち、ズームイン時の正確な可視性を維持できます。

ドキュメント

マイナーリリースのドキュメントページの統合　メジャーな Zabbix リリースのマイナーバージョンのリリースドキュメントは、それぞれ新機能とアップグレードノートの単一のドキュメントページにまとめられるようになりました。

プロセス

SNMPv3 EngineID のキャッシュと再利用　Zabbix は、SNMPv3 EngineID → IP のマッピングをキャッシュし、以降の SNMPv3 チェックでキャッシュされた EngineID を再利用しようとします。これにより、プローブトラフィックが削減され、ポーラーのパフォーマンスが向上します。再利用された EngineID が応答しない場合、ポーラーは EngineID プローブにフォールバックし、インターフェースの変更や永続的な障害の後に古いエントリを削除することがあります。

履歴キャッシュ回復時のプロキシスロットリングの改善　プロキシのスロットリングロジックが改良され、履歴キャッシュ回復時のサーバーの安定性が向上しました。履歴キャッシュの使用率がスロットリングの閾値に達すると、従来通りサーバーはプロキシデータの受け入れを停止し続けます。キャッシュの使用率が 60% まで下がると、サーバーはスロットリングリストの処理を開始しますが、キャッシュの圧力がさらに下がるまで、非常に大きなバッチ（約 10,000 レコード以上）を含むプロキシのアップロードは引き続き拒否される場合があります。この変更により、サーバーの回復中にキャッシュの過負荷が繰り返されるリスクが軽減されます。

zabbix_get および **zabbix_js** の最大タイムアウト値の増加　zabbix_getおよびzabbix_jsコマンドラインユーティリティの `timeout` パラメータの最大値が 600 秒に増加しました。

削除されたトリガーのイベントのハウスキーパーによるクリーンアップ　トリガーが削除されると、ハウスキーパーは、そのトリガーの問題に関連するすべてのイベントも削除するようになりました。以前は、トリガーの問題のみが削除され、イベントはハウスキーパーのトリガーデータ保存期間が満了した後にのみ削除されていました。

手動クローズ ― 復旧イベントにトリガータグが継承される　手動クローズ後に作成された復旧イベントは、アイテムタグやホストタグに加えてトリガータグも継承します。これらのタグはイベントタグ配列に存在し、`{EVENT.RECOVERY.TAGS}`や`{EVENT.RECOVERY.TAGSJSO`どの通知マクロで利用できます。

アイテム

smart.disk.discovery ― 新しい type パラメータ　smart.disk.discovery アイテム（Zabbix agent 2 S.M.A.R.T. プラグイン）は、ディスクをスキャンする値を指定するためのオプションの **type** パラメータを受け入れるようになりました。

プラグイン

Ceph プラグイン　このプラグインは現在、2 つのモードで動作します。

- **native** - このモードは go-ceph ライブラリを使用して、ネイティブの Ceph API（msgr2 プロトコル）を使用して Ceph クラスターと直接通信します。これは最新の Ceph インストールに推奨されるモードですが、**Linux** のみでサポートされており、Ceph 16 以降でのみ利用可能です。
- **restful**（非推奨）- このモードは通信に Ceph RESTful API を使用します。これは後方互換性のためのデフォルトモードですが、mgr/restful モジュールの削除により、Ceph バージョン 20（Tentacle）以降では動作しません。

どちらのモードを使用するかは、mode パラメータ（native/restful）の値によって決まります。

- `Plugins.Ceph.Default.Mode=native` - プラグインのネイティブモードを設定
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - 指定したセッションのネイティブモードを設定

ユーザー認証情報のセットは各モードで異なり、互換性がないことに注意してください。`Plugins.Ceph.InsecureSkipVerify` パラメータは、接続のセキュリティが msgr2 プロトコル（デフォルトでセキュア）によって Ceph クラスター側で定義されているため、ネイティブモードでは無視されます。

Zabbix agent 2 用の Ceph プラグインは現在ロード可能なプラグインとなっており、追加のインストール手順が必要であることに注意してください。これは librados パッケージ（ネイティブモード用）が必要なためです。詳細は Ceph プラグインのreadmeを参照してください。

Oracle プラグイン　Zabbix agent 2 のOracle プラグインは、TCPS（TLS）プロトコルを使用した Oracle データベースへの暗号化接続をサポートするようになりました。これにより、TLS で保護されたソケットを介して Oracle インスタンスを監視できるようになり、リモート監視のセキュリティが向上します。

ロード可能なプラグインのテスト実行モード　ロード可能なプラグインは、`-t` (`--test`) フラグを使用してテストモードで起動できるようになりました。このとき、アイテムキーを引数として渡します。このモードでは、プラグインはデバッグや開発目的で実行され、プラグインの設定ファイルは無視されます。

Redis プラグイン **— TLS** サポートと起動時検証　Zabbix agent 2 のRedis プラグインに TLS サポートが追加されました。

プラグインの TLS 設定の起動時検証が実装され、検証/エラーメッセージが改善されました。無効な設定ロジック（例：`TLSCAFile` を指定せずに接続タイプ `verify_full` を使用するなど）は、Zabbix agent 2 の起動を妨げる可能性があります。

ローレベルディスカバリ

ネストされたローレベルディスカバリのマクロサポート　ローレベルディスカバリマクロは、以下のネストされたローレベルディスカバリルールでサポートされるようになりました:

- JSONPath の前処理パラメータ
- カスタム LLD マクロの JSONPath フィールド

Zabbix 8.0.x の新機能

このページでは、Zabbix のメジャーバージョンのマイナーリリースに含まれる新機能の情報をまとめて提供します。

メジャーバージョンの新機能も参照してください。

既存テンプレートの変更や、さまざまなデバイス、サービス、パートナーソリューションをカバーする新しいテンプレートの情報については、テンプレートの変更を参照してください。

Zabbix 8.0.1 の新機能　このバージョンはまだリリースされていません。

3 Zabbix アプライアンス

概要　Zabbix アプライアンスは、Zabbix サーバーとフロントエンドを手動で設定したり、Zabbix 用の既存のサーバーを再利用したりする代わりに、即座に展開する方法を提供します。

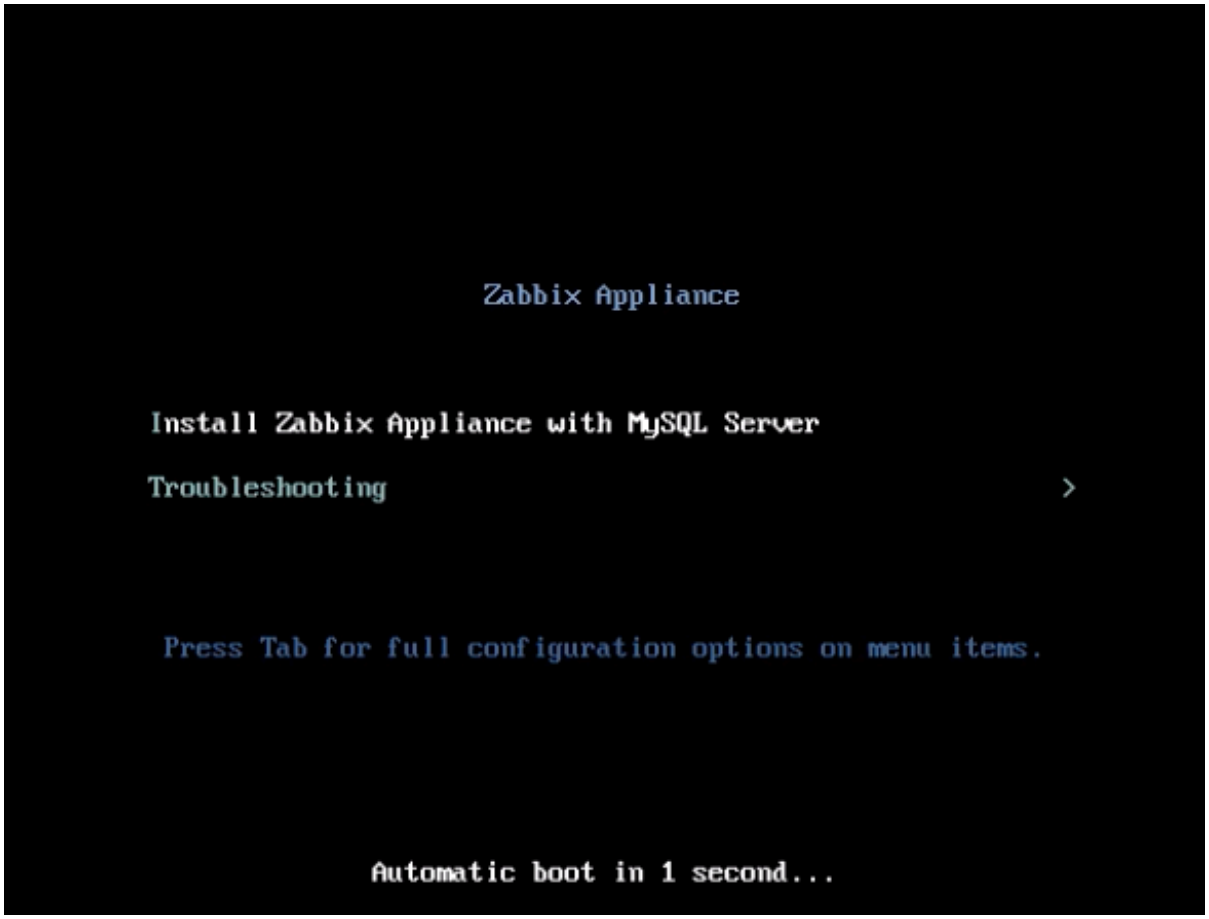
このアプライアンスは AlmaLinux 8 (x86_64) に基づいており、MySQL 上で実行される設定済みの Zabbix サーバーと Nginx Web サーバー上で実行されるフロントエンドが含まれています。

Attention:
このアプライアンスは、Zabbix を評価するために設計されています。本番環境での使用は推奨していません。

アプライアンスイメージは、次の形式でダウンロードできます。

- インストール CD/DVD (.iso)
- VMware (.vmx) - 注意点
- オープン仮想化フォーマット (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - 注意点
- KVM、Parallels、QEMU、USB スティック、VirtualBox、Xen (.raw) - 注意点
- KVM、QEMU (.qcow2)

Zabbix インストール CD/DVD のブートメニュー:



クイックスタート　前提条件

1. ホストマシンに、仮想マシンのシステム要件を満たすのに十分なリソースがあることを確認してください。

- *RAM*: 4 GB
- ディスク容量: 仮想マシンに最低 8GB
- *CPU*: 最低 2 コア

2. まだインストールされていない場合は、アプライアンスイメージを起動するための仮想化ソフトウェアをインストールします (例: VirtualBox)。
3. 仮想化ソフトウェアでサポートされている形式でアプライアンスをダウンロードします。
4. ネットワーク設定をチェックして、ホストマシンで DHCP が有効になっていることを確認します。

インストール

1. ダウンロードしたイメージからアプライアンス仮想マシンを起動します。
2. ホストマシン上のブラウザからアクセスできるように仮想マシンのネットワーク設定を構成します。これは *Bridged mode* を有効にすることで実現できます。
3. デフォルトのシステム認証情報を使用して仮想マシンにログインします。
4. IP アドレスを取得するには、仮想マシンで次のコマンドを実行します:

```
ip addr show
```

5. ホストマシンでブラウザを開き、アプライアンスが DHCP 経由で受信した IP アドレスを指定します。
6. デフォルトの認証情報を使用して Zabbix にログインし、監視を開始します。

設定　このセクションでは、頻繁に必要となるデフォルトの設定と、利用可能なカスタマイズオプションについて説明します。

ユーザー認証情報

システム

- ユーザー名: root
- パスワード: zabbix

Zabbix フロントエンド

- ユーザー名: Admin
- パスワード: zabbix

ログイン後、ユーザープロファイル設定でデフォルトのパスワードを変更したり、新しいユーザーを作成してデフォルトのパスワードを削除したりできます。

データベース

すべてのデータベースユーザーのパスワードは、インストールプロセス中にランダムに生成されます。データベースには次のユーザーが定義されています。

Root:

- ユーザー名: root
- パスワード: パスワードは`/root/.my.cnf` ファイルに保存されます。root アカウントではパスワードを入力する必要はありません。

Zabbix サーバー*:*

- ユーザー名: zabbix_srv
- パスワード: パスワードは`/etc/zabbix/zabbix\_server.conf` に保存されます。

Zabbix フロントエンド*:*

- ユーザー名: zabbix_web
- パスワード: パスワードは`/etc/zabbix/web/zabbix.conf.php` に保存されます。

データベースユーザーのパスワードを変更するには、MySQL および対応する設定ファイルでパスワードを変更します。

フロントエンドのアクセス

Zabbix フロントエンドには、*http://*< 仮想マシンの *IP*> からアクセスできます。

デフォルトでは、どこからでもアクセスが許可されます。アクセスを制限するには`/etc/nginx/conf.d/zabbix.conf` を変更します。編集したファイルを保存したら、root ユーザーとして SSH 経由でログインし、次のコマンドを実行して Nginx を再起動します。

```
systemctl restart nginx
```

静的 IP アドレス

デフォルトでは、アプライアンスは DHCP を使用して IP アドレスを取得します。静的 IP アドレスを指定するには:

- root ユーザーとしてログインします。
- 次のコマンドを実行し、値をカスタム IP アドレスに置き換えます。

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # アプライアンスのIPアドレス/CIDRプレフィッ
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # ゲートウェイのIPアドレス
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # DNSサーバーのIPアドレス
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

あるいは、ファイル`/etc/NetworkManager/system-connections/eth0.nmconnection` を開いて次の行を追加することもできます。

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

編集したファイルを保存した後、`systemctl restart network` コマンドを実行します。

ファイアウォールの構成

ファイアウォール設定を管理するために、アプライアンスは事前定義されたルールを持つ iptables を使用します。

- SSH (22 TCP) ポートを開放
- Zabbix エージェント (10050 TCP) と Zabbix トラッパー (10051 TCP) ポートを開放
- HTTP (80 TCP) と HTTPS (443 TCP) ポートを開放
- SNMP trap (162 UDP) ポートを開放
- NTP (123 UDP) ポートの発信接続を開放
- ICMP パケットを 1 秒あたり 5 パケットに制限
- 他のすべての着信接続はドロップ

追加のポートを開くには、`/etc/sysconfig/iptables` ファイルを変更し、ファイアウォールルールをリロードします。

```
systemctl reload iptables
```

リポジトリ

Zabbix アプライアンスは、Zabbix リポジトリの *zabbix-release* パッケージを使用します。リポジトリは`/etc/yum.repos.d/*` ディレクトリで設定されています。

タイムゾーン

デフォルトではアプライアンスはシステムクロックに UTC を使用します。タイムゾーンを変更するには適切なファイルを`/usr/share/zoneinfo` から`/etc/localtime` にコピーします。例:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

> **Note:**
> Zabbixフロントエンドのタイムゾーンは個別に設定され、フロントエンドの設定で変更できます。Zabbix フロントエンドのデフォルトのタイムゾーンは Europe/Riga です。

ファイルの場所

- 設定ファイルは`/etc/zabbix` にあります。
- Zabbix サーバー、プロキシ、エージェントのログファイルは`/var/log/zabbix` にあります。
- Zabbix フロントエンドは`/usr/share/zabbix` にあります。
- ユーザー *zabbix* のホームディレクトリは`/var/lib/zabbix` です。

システムサービス

Systemd サービスが利用可能です。Zabbix サービスのリストを表示するには、仮想マシンで次のコマンドを実行します。

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

イメージフォーマット固有の注意点　VMware

vmdk 形式のイメージは、VMware Player、Server および Workstation 製品で直接使用できます。ESX、ESXi および vSphere で使用するには、VMware vCenter コンバーター (authentication required for download) を使用して変換する必要があります。VMware vCenter コンバーターを使用する場合、ハイブリッドネットワークアダプタで問題が発生する可能性があります。その場合は、変換プロセス中に E1000 アダプタを指定してみてください。あるいは、変換の完了後、既存のアダプタを削除して E1000 アダプタを追加することもできます。

HDD/フラッシュイメージ (raw)

イメージを起動するには、次のコマンドを実行します。

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

`/dev/sdc` をフラッシュ/HDD ディスクデバイスのパスに置き換えてください。

Hyper-V

アプライアンスが Hyper-V で起動しない場合は、**Ctrl+Alt+F2** を押して TTY セッションを切り替えてみてください。

トラブルシューティング　フロントエンドへのログイン時に *Access denied for user 'replace_user'@'localhost' (using password: YES)* というエラーメッセージが表示された場合、インストールがまだ進行中であることを示している可能性があります。



数分待ってもエラーが解消しない場合や、他に予期しない動作が見られる場合は、インストールプロセスが正常に完了していない可能性があります。この場合は、現在のアプライアンスを削除し、同じインストール手順に従って再デプロイすることをお勧めします。この手順で通常は問題が解決します。

壊れたインストールを手動で修正しようとすることは、さらなる問題を引き起こす可能性があるため推奨されませんのでご注意ください。

4 Zabbix プロセス

サイドバーを使用して、Zabbix プロセスセクションのコンテンツにアクセスしてください。

1 Zabbix サーバー

概要

Zabbix サーバーは、Zabbix ソフトウェアの中心的なプロセスです。

サーバーはデータのポーリングとトラッピングを実行し、トリガーを計算し、ユーザーに通知を送信します。Zabbix エージェントやプロキシが、システムの可用性や整合性に関するデータを報告する中心的なコンポーネントです。サーバー自体も、シンプルなサービスチェックを使用して、ネットワークサービス（Web サーバーやメールサーバーなど）をリモートでチェックできます。

サーバーは、すべての設定、統計、運用データが保存される中央リポジトリであり、監視対象システムのいずれかで問題が発生した場合に、管理者に積極的にアラートを送信する Zabbix のエンティティです。

基本的な Zabbix サーバーの機能は、3 つの異なるコンポーネントに分かれています。それらは、Zabbix サーバー、Web フロントエンド、データベースストレージです。

Zabbix のすべての設定情報はデータベースに保存されており、サーバーと Web フロントエンドの両方がデータベースとやり取りします。たとえば、Web フロントエンド（または API）を使用して新しいアイテムを作成すると、それはデータベースの items テーブルに追加されます。その後、約 1 分ごとに Zabbix サーバーは items テーブルをクエリし、アクティブなアイテムのリストを取得し、それを Zabbix サーバー内のキャッシュに保存します。このため、Zabbix フロントエンドで行った変更が最新データセクションに反映されるまでに最大 2 分かかる場合があります。

Zabbix サーバーの起動

パッケージとしてインストールした場合

Zabbix サーバーはデーモンプロセスとして動作します。サーバーは以下のコマンドで起動できます。

```
systemctl start zabbix-server
```

これはほとんどの GNU/Linux システムで動作します。他のシステムでは、以下のコマンドを実行する必要があるかもしれません。

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

同様に、停止/再起動/ステータス表示には以下のコマンドを使用します。

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

手動で起動する

上記でうまくいかない場合は、手動で起動する必要があります。zabbix_server バイナリのパスを探して、以下を実行します:

```
zabbix_server
```

Zabbix サーバーでは、以下のコマンドラインパラメータが利用できます:

```
-c --config <file>              設定ファイルのパス（デフォルトは /usr/local/etc/zabbix_server.conf）
-f --foreground                 Zabbix サーバーをフォアグラウンドで実行
-R --runtime-control <option>   管理機能の実行
-T --test-config                設定ファイルの検証と終了
-h --help                       このヘルプを表示
-V --version                    バージョン番号の表示
```

コマンドラインパラメータを指定して Zabbix サーバーを起動する例:

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

ランタイム制御

ランタイム制御オプション:

オプション	説明	ターゲット
config_cache_reload	設定キャッシュをリロードします。キャッシュが現在ロード中の場合は無視されます。	
history_cache_clear[=**target**]	ID で指定されたアイテムの履歴キャッシュをクリアします。 アイテムの最初と最後の値を除くすべての値に影響します。	**target** - アイテムの ID
diaginfo[=**section**]	サーバーログファイルに診断情報を収集します。	**historycache** - 履歴キャッシュの統計情報 **valuecache** - 値キャッシュの統計情報 **preprocessing** - 前処理マネージャの統計情報 **alerting** - アラートマネージャの統計情報 **lld** - LLD マネージャの統計情報 **locks** - ミューテックスのリスト（*BSD* システムでは空） **connector** - 最大キューを持つコネクタの統計情報
ha_status	高可用性（HA）クラスタのステータスをログに記録します。	
ha_remove_node=**target**	名前または ID で指定された高可用性（HA）ノードを削除します。 アクティブ/スタンバイノードは削除できませんのでご注意ください。	**target** - ノードの名前または ID（ha_status の実行で取得可能）
ha_set_failover_delay=**delay**	高可用性（HA）のフェイルオーバー遅延を設定します。 時間のサフィックスがサポートされています（例: 10s, 1m）。	
proxy_config_cache_reload[=**target**]	プロキシの設定キャッシュをリロードします。	**target** - カンマ区切りのプロキシ名リスト ターゲットが指定されていない場合、すべてのプロキシの設定をリロードします
secrets_reload	Vault からシークレットをリロードします。	
service_cache_reload	サービスマネージャのキャッシュをリロードします。	
snmp_cache_reload	SNMP キャッシュをリロードします — すべてのホストの SNMP エンジンプロパティ（エンジンタイム、エンジンブーツ、エンジン ID、認証情報）をクリアします。SNMP の問題をトラブルシューティングする際にグローバルキャッシュのクリアを強制するために使用します。	
housekeeper_execute	ハウスキーピング手順を開始します。 ハウスキーピング手順が現在進行中の場合は無視されます。	

オプション	説明	ターゲット
trigger_housekeeper_execute	サーバーのトリガーハウスキーピング手順を開始し、削除されたトリガーによって発生した問題や、そのような問題によって生成されたサービス問題（ハウスキーピング時に解決済みと見なされる）を削除します。 ハウスキーピング手順が開始されるまで、削除されたトリガーによって発生した問題がサービス問題を生成し、サービスに割り当てられる可能性があることに注意してください。 頻繁に検出/未検出されるトリガーに基づく多くのサービスステータス計算ルールがセットアップに含まれている場合は、ProblemHousekeepingFrequencyサーバー設定パラメータを調整して、トリガーハウスキーピング手順の頻度を増やすことを検討してください。 トリガーハウスキーピング手順が現在進行中の場合は無視されます。	
log_level_increase[=**<target>**]	ログレベルを上げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	**process type** - 指定したタイプのすべてのプロセス（例: poller） すべてのサーバープロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: poller,3） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを’process type,N’ として指定してください。
log_level_decrease[=**<target>**]	ログレベルを下げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	
prof_enable[=**<target>**]	プロファイリングを有効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。 有効なプロファイリングは、関数名ごとのすべての rwlock/mutex の詳細を提供します。	**process type** - 指定したタイプのすべてのプロセス（例: history syncer） プロファイリングターゲットとしてサポートされるプロセスタイプ: alerter, alert manager, availability manager, configuration syncer, discovery manager, escalator, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, odbc poller, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **process type,N** - プロセスタイプと番号（例: history syncer,1） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを’process type,N’ として指定してください。 **scope** - `rwlock`, `mutex`, `processing` はプロセスタイプと番号（例: history syncer,1,processing）またはタイプのすべてのプロセス（例: history syncer,rwlock）で使用できます

オプション	説明	ターゲット
prof_disable[=<**target**>]	プロファイリングを無効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。	**process type** - 指定したタイプのすべてのプロセス（例: history syncer） プロファイリングターゲットとしてサポートされるプロセスタイプ: `prof_enable` を参照 **process type,N** - プロセスタイプと番号（例: history syncer,1） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを'process type,N' として指定してください。

サーバーの設定キャッシュをリロードするためにランタイム制御を使用する例:

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

プロキシの設定をリロードするためにランタイム制御を使用する例:

```
### すべてのプロキシの設定をリロード:
zabbix_server -R proxy_config_cache_reload

### Proxy1 と Proxy2 の設定をリロード:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

アイテムのヒストリキャッシュをクリアするためにランタイム制御を使用する例:

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

診断情報を収集するためにランタイム制御を使用する例:

```
### サーバーログファイルに利用可能なすべての診断情報を収集:
zabbix_server -R diaginfo

### サーバーログファイルにヒストリキャッシュの統計情報を収集:
zabbix_server -R diaginfo=historycache
```

SNMP キャッシュをリロードするためにランタイム制御を使用する例:

```
zabbix_server -R snmp_cache_reload
```

Attention:
SNMPv3 インターフェースが Zabbix UI 経由で更新された場合、ほとんどの場合 Zabbix はそのインターフェースの新しい SNMPv3 認証情報を自動的にリロードします。認証情報の変更後もポーリングが失敗する場合 (例えば、engineBoots/engineID の不整合や RFC 非準拠デバイスなど) や、トラブルシューティングのためにグローバルな SNMP キャッシュのクリアを強制する必要がある場合のみ、`-R snmp_cache_reload` を使用してください。

ハウスキーパーの実行をトリガーするためにランタイム制御を使用する例:

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

ログレベルを変更するためにランタイム制御を使用する例:

```
### すべてのプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### 2 番目のポーラープロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### すべての http poller プロセスのログレベルを下げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

HA フェイルオーバー遅延を最小の 10 秒に設定する例:

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

プロセスユーザー

Zabbix サーバーは、root 以外のユーザーとして実行するように設計されています。起動した root 以外のユーザーとして実行されます。したがって、サーバーは問題なく任意の root 以外のユーザーとして実行できます。

'root' として実行しようとすると、ハードコーディングされた'zabbix' ユーザーに切り替わります。このユーザーは、システム上に存在している必要があります。サーバーを'root' として実行するには、サーバーの設定ファイルで'AllowRoot' パラメータを適切に変更する必要があります。

Zabbix サーバーとエージェントを同じマシンで実行する場合は、サーバーとエージェントで異なるユーザーを使用することを推奨します。そうしないと、両方を同じユーザーで実行した場合、エージェントがサーバーの設定ファイルにアクセスでき、Zabbix の管理者レベルのユーザーであれば、たとえばデータベースのパスワードなどを簡単に取得できてしまいます。

設定ファイル

zabbix_server の設定の詳細については、設定ファイルのオプションを参照してください。

起動スクリプト

これらのスクリプトは、システムの起動/シャットダウン時に Zabbix プロセスを自動的に開始/停止するために使用されます。スクリプトは misc/init.d ディレクトリにあります。

サーバープロセスタイプとスレッド

- `agent poller` - ワーカースレッドを持つパッシブチェック用の非同期ポーラープロセス
- `alert manager` - アラートキュー管理プロセス
- `alert syncer` - アラート DB 書き込みプロセス
- `alerter` - 通知送信用プロセス
- `availability manager` - ホストの可用性更新用プロセス
- `browser poller` - ブラウザアイテムチェック用ポーラー
- `configuration syncer` - 設定データのインメモリキャッシュ管理プロセス
- `configuration syncer worker` - アイテム名のユーザーマクロ値を解決・同期するプロセス
- `connector manager` - コネクタの管理プロセス
- `connector worker` - コネクタマネージャからのリクエストを処理するプロセス
- `discovery manager` - デバイスのディスカバリ管理プロセス
- `discovery worker` - ディスカバリマネージャからのディスカバリタスクを処理するプロセス
- `escalator` - アクションのエスカレーションプロセス
- `ha manager` - 高可用性管理プロセス
- `history poller` - データベース接続を必要とする計算チェック処理プロセス
- `history syncer` - ヒストリ DB 書き込みプロセス
- `housekeeper` - 古い履歴データの削除プロセス
- `http agent poller` - ワーカースレッドを持つ HTTP チェック用の非同期ポーラープロセス
- `http poller` - Web 監視用ポーラー
- `icmp pinger` - icmpping チェック用ポーラー
- `internal poller` - 内部チェック用ポーラー
- `ipmi manager` - IPMI ポーラーマネージャ
- `ipmi poller` - IPMI チェック用ポーラー
- `java poller` - Java チェック用ポーラー
- `lld manager` - ローレベルディスカバリタスクの管理プロセス
- `lld worker` - ローレベルディスカバリタスクのワーカープロセス
- `odbc poller` - ODBC チェック用ポーラー
- `poller` - パッシブチェック用の通常ポーラー
- `preprocessing manager` - 前処理タスクの管理プロセス（前処理ワーカースレッド付き）
- `preprocessing worker` - データ前処理用スレッド
- `proxy poller` - パッシブプロキシ用ポーラー
- `proxy group manager` - プロキシの負荷分散と高可用性の管理プロセス
- `report manager`- 定期レポート生成タスクの管理プロセス
- `report writer` - 定期レポート生成プロセス
- `self-monitoring` - サーバー内部統計収集プロセス
- `service manager` - history syncer、task manager、alert manager から問題、問題タグ、問題復旧の情報を受信してサービスを管理するプロセス
- `snmp poller` - SNMP チェック用のワーカースレッド付き非同期ポーラープロセス（`walk[OID]` および `get[OID]` アイテムのみ）
- `snmp trapper` - SNMP トラップ用トラッパー
- `task manager` - 他のコンポーネントから要求されたタスク（例：問題のクローズ、問題の確認、アイテム値の即時チェック、リモートコマンド機能）のリモート実行プロセス
- `timer` - メンテナンス処理用タイマー

- `trapper` - アクティブチェック、トラップ、プロキシ通信用トラッパー
- `trigger housekeeper` - 削除されたトリガーによって生成された問題を削除するプロセス
- `unreachable poller` - 到達不能デバイス用ポーラー
- `vmware collector` - VMware サービスからデータ収集を担当する VMware データコレクタ

これらのプロセスタイプはサーバーログファイルで確認できます。

サーバーログファイルは、ファイル所有者のみが読み書き可能な権限で作成されます。さらに、所有者グループも読み取り可能です。他のすべての権限は拒否されます。

さまざまなタイプの Zabbix サーバープロセスは、**zabbix[process,<type>,<mode>,<state>]** 内部アイテムで監視できます。

ヒストリーシンカーのトランザクション統計

ヒストリーシンカーのプロセスタイトルには、ヒストリーシンカーのトランザクションに関する詳細な統計情報が表示されます。

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

”A+B triggers” の意味:

- A - ヒストリ値によって処理されたトリガー数
- B - タイマーによって処理されたトリガー数

”processed...in N (<timings>) sec” のタイミングは以下の通りです:

- アイテム値をデータベースに書き込むのにかかった時間
- アイテムデータ (状態、エラー、ホストインベントリなど) の更新にかかった時間
- トレンドをデータベースにフラッシュするのにかかった時間
- トリガーの計算にかかった時間
- イベントとアクションの処理にかかった時間

サポートされているプラットフォーム

セキュリティ要件やサーバー運用のミッションクリティカルな性質から、UNIX は必要なパフォーマンス、フォールトトレランス、レジリエンスを一貫して提供できる唯一のオペレーティングシステムです。Zabbix は市場をリードするバージョンで動作します。

Zabbix サーバーは以下のプラットフォームでテストされています:

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

Note:
Zabbix は他の Unix 系オペレーティングシステムでも動作する場合があります。

ロケール

サーバーが一部のテキスト項目を正しく解釈できるように、UTF-8 ロケールが必要であることに注意してください。ほとんどの最新の Unix 系システムではデフォルトで UTF-8 ロケールが設定されていますが、システムによっては明示的に設定する必要がある場合があります。

1 高可用性 (HA)

概要

実質的にダウンタイムを発生させられない重要なインフラストラクチャーでは、高可用性 (HA) が求められます。したがって、停止してしまう可能性があるサービスについては、現在のサービスが停止してしまったときに処理を引き継げるようにするためのフェイルオーバーオプションが必要です。

Zabbix は、セットアップが簡単で、以前の HA の専門知識を必要としない組み込まれた高可用性のソリューションを提供します。組み込まれた Zabbix の HA は、Zabbix サーバーのソフトウェア/ハードウェアの障害に対する保護やメンテナンスに伴うダウンタイムを削減するのに役に立つでしょう。

Zabbix の高可用性モードでは、複数の Zabbix サーバーがクラスター内のノードとして実行されます。クラスター内の 1 台の Zabbix サーバーがアクティブである間、他のサーバーはスタンバイ状態となり、必要に応じて引き継ぐことができるよう準備されます。



Zabbix の HA への切り替えは戻すことが可能で、いつでもスタンドアロン環境に戻すことができます。

参照: 実装の詳細

高可用性の有効化

Zabbix サーバーをクラスターノードとして起動

Zabbix サーバーをクラスターノードとして起動するには、サーバー設定で 2 つのパラメーターを設定する必要があります。

- **HANodeName** パラメーターは、HA クラスターノードとなる Zabbix サーバーごとに指定する必要があります。

これは、エージェントおよびプロキシ設定でサーバーが参照される一意のノード識別子 (例: `zabbix-node-01`) です。HANodeName が指定されていない場合、サーバーはスタンドアロンモードで起動されます。

- **NodeAddress** パラメーターは、ノードごとに指定する必要があります。

NodeAddress パラメーター (アドレス: ポート番号) は、Zabbix フロントエンドがアクティブなサーバーノードに接続するために使用されます。NodeAddress は、それぞれの Zabbix サーバーの IP または FQDN 名と一致する必要があります。

設定ファイル変更後、すべての Zabbix サーバーを再起動します。これでクラスターノードとして機能します。サーバーの最新ステータスは、レポート → システム情報で確認できます。

```
zabbix_server -R ha_status
```

上記ランタイムコマンドは、現在の HA クラスターの状態を Zabbix サーバーログ (および stdout) に記録します。

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

Web インターフェースの準備

Web インターフェースの設定ファイル (Web インターフェースのファイルがあるディレクトリ内の `conf/zabbix.conf.php`) 内で、Zabbix サーバーのアドレス: ポート番号が 設定されていないことを確認してください。

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

Zabbix の Web インターフェースは、Zabbix データベースのノードテーブルから設定を読み取って、アクティブノードを自動的に検出します。アクティブノードのノードアドレスが Zabbix サーバーのアドレスとして使用されます。

Zabbix プロキシの設定

HA クラスターノード (サーバー) は、パッシブまたはアクティブモードの Zabbix プロキシの設定ファイルにリスト化されている必要があります。

パッシブプロキシの場合、ノード名をプロキシ設定ファイルの Serverパラメーターにカンマ区切りでリスト化する必要があります。

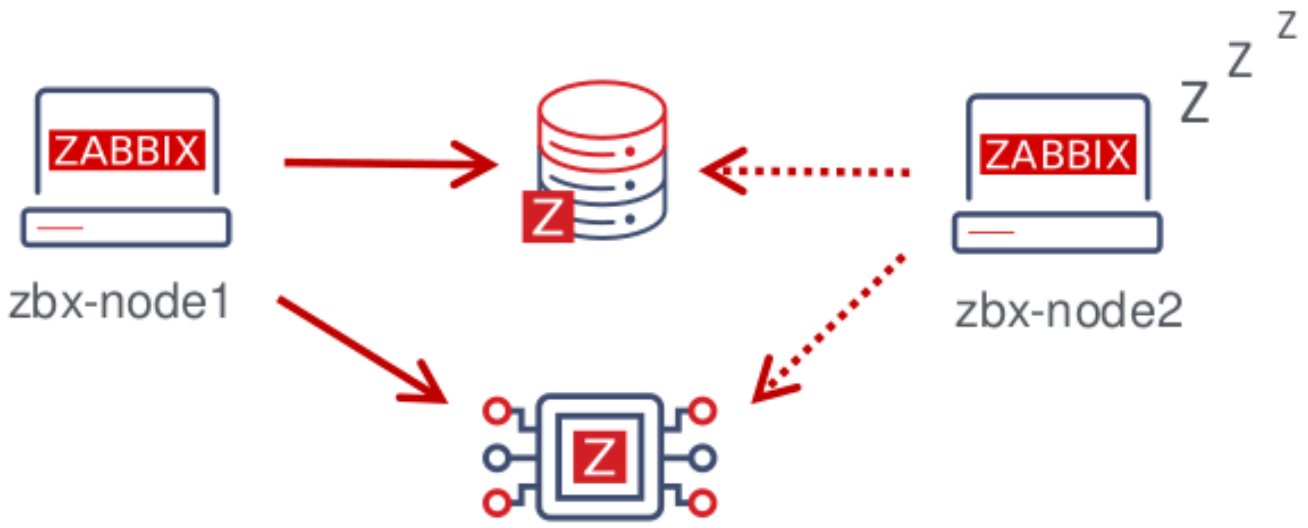
```
Server=zabbix-node-01,zabbix-node-02
```

アクティブプロキシの場合、ノード名をプロキシ設定ファイルの Serverパラメーターにセミコロン区切りでリスト化する必要があります。

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

Zabbix エージェントの設定

HA クラスターのノード (サーバー) は、Zabbix エージェントまたは Zabbix エージェント 2 の設定ファイルに並べて記述することが必要です。



パッシブチェックを行うのならば、ノード名を Serverパラメーターにカンマ区切りで設定する必要があります。

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

アクティブチェックを行うのならば、ノード名を ServerActiveパラメーターに設定する必要があります。ノードと他のサーバーとはカンマで、他のノードとはセミコロン区切ります。例:

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

スタンバイノードへのフェイルオーバー

アクティブなノードが停止すると、Zabbix は自動的に別のノードにフェイルオーバーします。フェイルオーバーが発生するには、少なくとも 1 つのノードがスタンバイ状態である必要があります。

フェイルオーバーはどのくらいの速さで実行されるか。すべてのノードは 5 秒ごとに最終アクセス時刻 (および変更されている場合はステータス) を更新します。したがって:

- アクティブノードがシャットダウンし、そのステータスが” 停止” と報告された場合、別のノードが **5** 秒以内に引き継ぎます。
- アクティブノードがシャットダウンするか、ステータスを更新できずに使用できなくなった場合、スタンバイノードはフェイルオーバー遅延 +5 秒待機して引き継ぎます。

フェイルオーバー遅延は設定可能で、サポートされる範囲は 10 秒から 15 分 (デフォルト 1 分) です。フェイルオーバー遅延の変更は次のコマンドを実行します:

```
zabbix_server -R ha_set_failover_delay=5m
```

HA クラスターの管理

HA クラスターの現在のステータスは、専用のランタイムコントロールオプションを使用して管理できます。

- `ha_status` - Zabbix サーバーログ (および標準出力) に HA クラスターのステータスを記録します。
- `ha_remove_node=target` - <target> で識別される HA ノードを削除します。<target> はノードの名前または ID であり、実行中の ha_status の出力から取得できます。例:

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

アクティブ/スタンバイノードは削除できないことに注意してください。

- `ha_set_failover_delay=delay` - HA フェイルオーバー遅延の設定 (10 秒から 15 分まで。タイムサフィックスがサポートされています。例: 10s, 1m)

ノードのステータスは次のように監視できます:

- レポート → システム情報
- システム情報のダッシュボードウィジェット
- サーバーの `ha_status` ランタイム制御オプション (上記を参照)

`zabbix[cluster,discovery,nodes]` の内部アイテムは、高可用性ノード情報を含む JSON を返すため、ノード検出に使用できます。

高可用性の無効化

高可用性クラスターを無効にするには:

- 設定ファイルのバックアップコピーを作成する
- スタンバイノードを停止する
- アクティブなプライマリサーバーから HANodeName パラメータを削除
- プライマリサーバーを再起動 (スタンドアロンモードで起動します)

HA クラスターのアップグレード

HA ノードのメジャーバージョンアップグレードを実行するには、次の手順を実行します:

- 全てのノードを停止する。
- データベースのフルバックアップを作成する。
- データベースがレプリケーションを使用している場合は、すべてのノードが同期していて問題がないことを確認してください。レプリケーションが壊れている場合はアップグレードしないでください。
- データベースのアップグレードを実行する単一ノードを選択し、HANodeName をコメントアウトしてその構成をスタンドアロンモードに変更して、アップグレードします。
- データベースのアップグレードが完了したことを確認します (システム情報には、Zabbix サーバーが実行中であると表示されます)。
- ノードを HA モードで再起動します。
- 残りのノードをアップグレードして起動します (この時点でデータベースはすでにアップグレードされているため、スタンドアロンモードに変更する必要はありません)。

マイナーバージョンのアップグレードでは、最初のノードをアップグレードし、それがアップグレードされて実行されていることを確認してから、次のノードでのアップグレードを開始するだけで十分です。

実装の詳細

高可用性 (HA) クラスターはオプトインソリューションであり、Zabbix サーバーでサポートされています。ネイティブ HA ソリューションは、簡単に使用できるように設計されており、複数のサイトで機能し、Zabbix が認識するデータベースに関する特定の要件はありません。ユーザーは、環境の高可用性要件に最適なものに応じて、ネイティブの Zabbix HA ソリューションまたはサードパーティの HA ソリューションを自由に使用できます。

このソリューションは複数の zabbix_server インスタンスまたはノードで構成されています。すべてのノードについて:

- 個別に構成
- 同じデータベースを使用
- いくつかのモードがあるかもしれません: アクティブ、スタンバイ、使用不可、停止

一度にアクティブ (稼働) できるノードは 1 つだけです。スタンバイノードは、HA マネージャーという 1 つのプロセスのみを実行します。スタンバイノードは、データの収集、処理、その他の通常のサーバーの仕事を行いません。ポートをリッスンしません。データベース接続は最小限です。

アクティブノードとスタンバイノードは両方とも、最終アクセス時刻を 5 秒ごとに更新します。各スタンバイノードは、アクティブノードの最終アクセス時刻を監視します。アクティブノードの最終アクセス時刻が'failover delay' 秒を超えている場合、スタンバイノードは自身をアクティブノードに切り替え、以前のアクティブノードに' 使用不可' ステータスを割り当てます。

アクティブノードは自身のデータベース接続を監視します。接続が `failover delay`-5 秒を超えて失われた場合は、すべての処理を停止してスタンバイモードに切り替ります。アクティブノードは、スタンバイノードのステータスも監視します。スタンバイノードの最終アクセス時刻が'failover delay' 秒を超えている場合、スタンバイノードには' 使用不可' ステータスが割り当てられます。

ノードは、Zabbix のマイナーバージョン間で互換性があるように設計されています。

2 Zabbix エージェント

概要

Zabbix エージェントは、監視対象にデプロイされ、ローカルリソースやアプリケーション（ハードドライブ、メモリ、プロセッサの統計情報など）を積極的に監視します。

エージェントはローカルで運用情報を収集し、さらなる処理のために Zabbix サーバーにデータを報告します。障害（ハードディスクの容量不足やサービスプロセスのクラッシュなど）が発生した場合、Zabbix サーバーは障害を報告した特定のマシンの管理者に積極的にアラートを送信できます。

Zabbix エージェントは、統計情報の収集にネイティブのシステムコールを使用するため、非常に効率的です。

パッシブチェックとアクティブチェック

Zabbix エージェントは、パッシブチェックとアクティブチェックを行うことができます。

- パッシブチェック - Zabbix エージェントは、Zabbix サーバー (またはプロキシ) からのリクエストに対して応答します。例えばサーバーが CPU load のデータを要求すると、エージェントが結果を返します。
- アクティブチェック - Zabbix エージェントは、Zabbix サーバー (またはプロキシ) からのリクエストを待たずにデータを収集および送信します。まず、サーバーから監視アイテムのリスト (CPU load、使用可能なメモリなど) を取得し、必要なデータを収集し、定期的に新しい値を送信します。

エージェントチェックタイプは、それぞれの監視アイテムのタイプを選択することで設定されます。Zabbix エージェントは、アイテムの”Zabbix エージェント” または”Zabbix エージェント (アクティブ)” のタイプを処理します。

サポートしているプラットフォーム

サポートしているプラットフォームについては、要件ページを参照してください。

UNIX 系 OS 上の Zabbix エージェント

UNIX 系 OS 上の Zabbix エージェントは、監視対象のホストで実行されます。

インストール

Zabbix エージェントは、以下のいずれかの方法で Linux ベースのシステムにインストールできます。

- Zabbix パッケージ - Zabbix のバージョン、OS ディストリビューション、OS バージョンを選択した後、エージェントコンポーネントを選択し、表示される手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent` オプションでソースを設定して Zabbix エージェントをコンパイルします。

Attention:
一般的に、32 ビットの Zabbix エージェントは 64 ビットシステムでも動作しますが、場合によっては失敗することがあります。

Note:
macOS、IBM AIX、FreeBSD、OpenBSD、Solaris 用の事前コンパイル済み Zabbix エージェントバイナリはダウンロードできます。現在の Zabbix サーバー/プロキシバージョンと互換性のあるレガシーバイナリは、NetBSDおよびHP-UX向けに利用可能です。

パッケージでインストールした場合

Zabbix エージェントはデーモンプロセスとして実行されます。エージェントを起動するには以下を実行します。

```
systemctl start zabbix-agent
```

これはほとんどの GNU/Linux システムで動きます。他のシステムでは、以下のように実行する必要がある場合があります。

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

Zabbix エージェントを停止/再起動/ステータス表示するには以下を実行します。

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

手動での起動

zabbix_agentd バイナリを見つけて直接実行することで、Zabbix エージェントを起動できます。例:

```
zabbix_agentd
```

Windows 上での Zabbix エージェント

Windows 上の Zabbix エージェントは、Windows サービスとして実行されます。

インストール

Zabbix エージェントは、次のいずれかの方法で Windows にインストールできます。

- 事前コンパイル済み Zabbix エージェントバイナリ - Zabbix エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントインストールページの手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix エージェントのビルドページの手順に従ってください。

Zabbix エージェント（ZIP アーカイブから）を Windows サービスとしてインストールする追加の詳細については、Microsoft Windows 上の Zabbix エージェントページを参照してください。

オプション

1 つのホストで複数のエージェントインスタンスを実行することができます。1 つのインスタンスはデフォルトの設定ファイルまたはコマンドラインで指定された設定ファイルを使用できます。複数のインスタンスの場合、各エージェントインスタンスは独自の設定ファイルを持つ必要があります（インスタンスの 1 つはデフォルトの設定ファイルを使用できます）。

Zabbix エージェントで使用できるコマンドラインパラメータは以下の通りです。

パラメータ	説明
UNIX および **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパス。 このオプションを使用して、デフォルト以外の設定ファイルを指定できます。 UNIX の場合、デフォルトは /usr/local/etc/zabbix_agentd.conf またはコンパイル時の変数 *--sysconfdir* または *--prefix* で設定されたもの Windows の場合、デフォルトは C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Zabbix エージェントをフォアグラウンドで実行（デフォルト: true）。
-p --print	既知のアイテムを表示して終了。 注: ユーザーパラメータの結果も返すには、設定ファイルを指定する必要があります（デフォルトの場所にない場合）。
-t --test <item key>	指定したアイテムをテストして終了。 注: ユーザーパラメータの結果も返すには、設定ファイルを指定する必要があります（デフォルトの場所にない場合）。
-T --test-config	設定ファイルを検証して終了。
-h --help	ヘルプ情報を表示。
-V --version	バージョン番号を表示。
UNIX エージェントのみ	
-R --runtime-control <option>	管理機能を実行。ランタイム制御を参照。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用（-i, -d, -s, -x オプションと併用）。 インスタンスのサービス名を区別するため、各サービス名には指定した設定ファイルの Hostname 値が含まれます。
-S --startup-type <value>	Zabbix Windows エージェントサービスの起動タイプを設定。許可される値: `automatic` - *(デフォルト)* Windows 起動時に自動的にサービスを開始; `delayed` - 自動起動サービスの起動完了後にサービスの開始を遅延（Windows Server 2008/Vista 以降で利用可能）; `manual` - サービスを手動で開始（ユーザーまたはアプリケーションによる）; `disabled` - サービスを無効化し、ユーザーやアプリケーションによる開始を禁止。 このオプションは-i オプションと併用、または既にインストール済みのサービスの起動タイプを変更するために単独で使用できます。
-i --install	Zabbix Windows エージェントをサービスとしてインストール。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストール。
-s --start	Zabbix Windows エージェントサービスを開始。
-x --stop	Zabbix Windows エージェントサービスを停止。

コマンドラインパラメータの具体的な例:

- すべての組み込みエージェントアイテムを値付きで表示
- 指定した設定ファイルで定義された”mysql.ping” キーのユーザーパラメータをテスト
- デフォルトの設定ファイルパス C:\Program Files\Zabbix Agent\zabbix_agentd.conf を使用して Windows に”Zabbix Agent” サービスをインストール
- エージェント実行ファイルと同じフォルダにある設定ファイル zabbix_agentd.conf を使用し、サービス名を設定ファイルの Hostname 値で拡張して一意にすることで、Windows に”Zabbix Agent [Hostname]” サービスをインストール
- エージェント実行ファイルと同じフォルダにある設定ファイル zabbix_agentd.conf を使用して、インストール済みの”Zabbix Agent” サービスの起動タイプを変更

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

ランタイム制御

ランタイム制御オプションを使用すると、エージェントプロセスのログレベルを変更できます。

オプション	説明	ターゲット
log_level_increase[=<target>]	ログレベルを上げます。 ターゲットが指定されていない場合、すべてのプロセスが影響を受けます。	ターゲットは次のように指定できます: **process type** - 指定されたタイプ (例　listener) のすべてのプロセス すべてのエージェントプロセスタイプを参照してください。 **process type,N** - プロセスのタイプと数 (例: listener,3) **pid** - プロセス ID (1 ~ 65535)。より大きな値の場合は、'process-type,N' の指定方法にしてください。
log_level_decrease[=<target>]	ログレベルを下げます。 ターゲットが指定されていない場合、すべてのプロセスが影響を受けます。	
userparameter_reload	現在の構成ファイルから *UserParameter* および *Include* オプションの値を再ロードします。	

例:

- すべてのプロセスのログレベルを上げる
- 3 番目の listener プロセスのログレベルを上げる
- PID 1234 のプロセスのログレベルを上げる
- すべてのアクティブなチェックプロセスのログレベルを下げる

```
zabbix_agentd -R log_level_increase
zabbix_agentd -R log_level_increase=listener,3
zabbix_agentd -R log_level_increase=1234
zabbix_agentd -R log_level_decrease="active checks"
```

Note:
ランタイム制御は、OpenBSD、NetBSD、および Windows ではサポートしません。

エージェントプロセスタイプ

- `active checks` - アクティブチェックを実行するプロセス
- `collector` - データ収集のためのプロセス
- `listener` - パッシブチェックをリッスンするプロセス

これらのプロセスタイプは、エージェントのログファイルで確認できます。

エージェントのログファイルは、ファイル所有者のみに読み書き権限で作成されます。さらに、ファイル所有者のグループにも読み取り可能です。他のすべての権限は拒否されます。

プロセスユーザー

UNIX 上の Zabbix エージェントは、root 以外のユーザーとして実行するように設計されていて、起動された root 以外のユーザーとして実行されます。したがって、root 以外のユーザーでエージェントを実行しても何も問題ありません。

'root' として実行しようとすると、ハードコードされた'zabbix' ユーザーに切り替わるため、'zabbix' ユーザーがシステム上に存在している必要があります。'root' として実行するためには、エージェントの設定ファイルの'AllowRoot' パラメーターを適切に変更する必要があります。

設定ファイル

Zabbix エージェントの設定の詳細については、Zabbix エージェント (UNIX)またはZabbix エージェント (Windows)の設定ファイルのオプションを参照してください。

ロケール

一部の文字型のエージェントアイテムで期待される値を返すには、UTF-8 ロケールがエージェントに必要であることに注意してください。最新の Unix ライクシステムのほとんどはデフォルトで UTF-8 ロケールを持っていますが、一部のシステムでは具体的に設定する必要がある可能性があります。

終了コード

Zabbix エージェントは、正常に終了した場合は 0 を返し、失敗した場合は 1 を返します。

3 Zabbix エージェント 2

概要

Zabbix エージェント 2 は新世代のZabbix エージェント](/manual/concepts/agent) であり、Go 言語で記述されています (Zabbix エージェントの C 言語コードが再利用されています)。次のように設計されています:

- TCP 接続の数を減らす
- チェックの同時実行性の改善
- プラグイン拡張の簡易化。最小限のコードで簡単なチェックを提供し、定期的なレポートを使用して、長期にわたるスクリプトとスタンドアロンのデータ収集で構成される複雑なチェックをサポートします。
- Zabbix エージェントの代替品として機能し、以前のすべての機能をサポートします。

パッシブチェックとアクティブチェック

Zabbix エージェント 2 は Zabbix エージェントと同様にパッシブおよびアクティブチェックをサポートしています。さらに、Zabbix エージェント 2 アクティブチェックは、柔軟な/スケジューリング間隔をサポートし、1 つのアクティブサーバー内の同時実行性をチェックします。

Note:
デフォルトでは、Zabbix エージェント 2 はリソース使用量の急増を防ぐために、アイテムの更新間隔内の条件付きランダム時間にアクティブチェックの最初のデータ収集をスケジュールします。エージェントの再起動直後に予定された更新確認間隔にないアクティブチェックを実行するには、設定ファイルで `ForceActiveChecksOnStart` パラメーター (グローバルレベル) または `Plugins.< プラグイン名 >.System.ForceActiveChecksOnStart` (特定のプラグインチェックにのみ影響します) を設定します。プラグインレベルのパラメーターが設定されている場合、グローバルパラメーターを上書きします。

同時実行性の確認

異なるプラグインからのチェックを同時に実行できます。1 つのプラグイン内の同時チェック数は、プラグイン容量設定によって制限されます。各プラグインにはハードコードされた容量設定 (デフォルトは 1000) があり、プラグイン設定パラメーターの `Plugins.< プラグイン名 >.System.Capacity=N` を使用して下げることができます。

サポートしているプラットフォーム

サポートしているプラットフォームについては、要件ページを参照してください。

UNIX 系 OS 上の Zabbix エージェント 2

UNIX 系 OS 上の Zabbix エージェント 2 は、監視対象のホストで実行されます。

インストール

Zabbix agent 2 は、Linux ベースのシステムに以下のいずれかの方法でインストールできます。

- Zabbix パッケージ - Zabbix のバージョン、OS ディストリビューション、OS バージョンを選択した後、Agent 2 コンポーネントを選択し、指示に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent2` オプションで設定してエージェントをコンパイルします。

Note:
Zabbix agent 2 の監視機能は、個別に提供されているロード可能なプラグインで拡張できます。詳細はロード可能なプラグインを参照してください。

パッケージでインストールした場合

Zabbix エージェント 2 はフォアグラウンドプロセスとして実行され、バックグラウンド実行には外部サービスマネージャー (systemd など) に依存します。Zabbix エージェント 2 には Linux 上でのデーモン化サポートが組み込まれていません。

エージェントは、次のコマンドを実行して起動できます:

```
systemctl start zabbix-agent2
```

Zabbix エージェント 2 を停止、再起動、またはステータスを確認するには、次のコマンドを使用します:

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

手動で起動

zabbix_agent2 バイナリを見つけて直接実行することで、Zabbix エージェントを起動できます。例:

```
zabbix_agent2
```

Windows システム上のエージェント 2

Zabbix エージェント 2 はスタンドアロンプロセスとして実行されますが、Windows サービスとして実行することもできます。

インストール

Zabbix agent 2 は、次のいずれかの方法で Windows にインストールできます。

- 事前コンパイル済み Zabbix エージェントバイナリ - エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントインストールページの手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix agent 2 のビルドページの手順に従ってください。

Note:
Zabbix agent 2 の監視機能は、個別に提供されているロード可能なプラグインで拡張できます。詳細は、ロード可能なプラグインを参照してください。

Zabbix agent 2（ZIP アーカイブから）を Windows サービスとしてインストールする追加の詳細については、Microsoft Windows 上の Zabbix エージェントページを参照してください。

オプション

Zabbix agent 2 では、以下のコマンドラインパラメータを使用できます。

パラメータ	説明
UNIX および **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパス。 このオプションを使用して、デフォルト以外の設定ファイルを指定できます。 UNIX のデフォルトは /usr/local/etc/zabbix_agent2.conf またはコンパイル時の変数 *--sysconfdir* または *--prefix* で設定されたもの Windows のデフォルトは C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Zabbix agent をフォアグラウンドで実行する（デフォルト: true）。
-p --print	既知のアイテムを表示して終了。 注: ユーザーパラメータの結果も返すには、設定ファイルがデフォルト以外の場所にある場合は設定ファイルを指定する必要があります。
-t --test <item key>	指定したアイテムをテストして終了。 注: ユーザーパラメータの結果も返すには、設定ファイルがデフォルト以外の場所にある場合は設定ファイルを指定する必要があります。
-T --test-config	設定ファイルを検証して終了。
-h --help	ヘルプ情報を表示して終了。
-v --verbose	デバッグ情報を表示する。-p および -t オプションと併用してください。
-V --version	エージェントのバージョンとライセンス情報を表示。
-R --runtime-control <option>	管理機能を実行します。ランタイム制御を参照してください。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用する（-i, -d, -s, -x オプションと併用）。 インスタンスのサービス名を区別するため、各サービス名には指定した設定ファイルの Hostname 値が含まれます。

パラメータ	説明
-S --startup-type <value>	Zabbix Windows エージェントサービスの起動タイプを設定します。許可される値: `automatic` - *(デフォルト)* Windows 起動時に自動的にサービスを開始; `delayed` - 自動的に開始されたサービスの起動完了後にサービスの開始を遅延; `manual` - サービスを手動（ユーザーまたはアプリケーション）で開始; `disabled` - サービスを無効にし、ユーザーまたはアプリケーションによる開始を禁止。 このオプションは -i オプションと併用するか、既にインストールされているサービスの起動タイプを変更するために単独で使用できます。
-i --install	Zabbix Windows エージェントをサービスとしてインストール。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストール。
-s --start	Zabbix Windows エージェントサービスを開始。
-x --stop	Zabbix Windows エージェントサービスを停止。

コマンドラインパラメータの具体的な使用例:

- すべての組み込みエージェントアイテムと値を表示
- 指定した設定ファイルで定義された”mysql.ping” キーのユーザーパラメータをテスト
- デフォルトの設定ファイルパス C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf を使用して Windows 用の”Zabbix Agent” サービスをインストール
- エージェント実行ファイルと同じフォルダにある設定ファイル zabbix_agent2.conf を使用して、インストール済みの Windows 用”Zabbix Agent” サービスの起動タイプを変更

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

ランタイム制御

ランタイム制御は、リモート制御のためのいくつかのオプションを提供します。

オプション	説明
log_level_increase	ログレベルを上げます。
log_level_decrease	ログレベルを下げます。
metrics	利用可能なメトリックを一覧表示します。
version	エージェントのバージョンを表示します。
userparameter_reload	現在の設定ファイルから *UserParameter* および *Include* オプションの値を再ロードします。
help	ランタイムコントロールのヘルプを表示します。

例:

- エージェント 2 のログレベルを上げます
- ランタイム制御オプションを出力します

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

設定ファイル

Zabbix エージェント 2 の設定パラメーターは、いくつかの例外を除いて、Zabbix エージェントと互換性があります。

新しいパラメーター	説明
ControlSocket	ランタイム制御ソケットパス。エージェント 2 はランタイムコマンドに制御ソケットを使用します。
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	これらのパラメーターは、アクティブなアイテムのエージェント 2 の永続ストレージを設定するために使用されます。
ForceActiveChecksOnStart	エージェントがアクティブチェックを再起動直後に実行するか、時間経過で均等に分散実行するかを決定します。
Plugins	プラグインは `Plugins.< プラグイン名 >.<Parameter>=<value>` の形式で独自のパラメーターを設定できます。一般的なプラグインパラメーターは *System.Capacity* で、同時に実行できるチェックの制限を設定します。

新しいパラメーター	説明
StatusPort	ポートエージェント 2 は、HTTP ステータスリクエストをリッスンし、設定されたプラグインといくつかの内部パラメーターのリストを表示します。
削除されたパラメーター	説明
AllowRoot, *User*	デーモン化がサポートされていないため、サポートされていません。
LoadModule, *LoadModulePath*	ロードモジュールはサポートされていません。
StartAgents	このパラメーターは、Zabbix エージェントではパッシブチェックの同時実行性を高めたり無効にしたりするために使用されていました。エージェント 2 では、同時実行はプラグインレベルで設定され、容量設定によって制限できます。一方、パッシブチェックの無効化は現在サポートされていません。
HostInterface, *HostInterfaceItem*	まだサポートされていません。

詳細についてはzabbix_agent2の設定ファイルオプションを参照してください。

終了コード

Zabbix エージェント 2 は古い OpenSSL バージョン (1.0.1、1.0.2) でコンパイルすることもできます。

この場合、Zabbix は OpenSSL でロックするためのミューテックスを提供します。ミューテックスのロックまたはロック解除が失敗した場合、エラーメッセージが標準エラーストリーム (STDERR) に出力され、エージェント 2 はそれぞれリターンコード 2 または 3 で終了します。

4 Zabbix プロキシ

概要

Zabbix プロキシは、1 つまたは複数の監視対象デバイスから監視データを収集し、その情報を Zabbix サーバーに送信するプロセスであり、本質的にはサーバーの代理として機能します。収集されたすべてのデータはローカルにバッファリングされ、その後プロキシが属する Zabbix サーバーに転送されます。

プロキシの導入は必須ではありませんが、単一の Zabbix サーバーの負荷を分散するために非常に有益です。プロキシのみがデータを収集する場合、サーバーでの処理は CPU やディスク I/O の負荷が少なくなります。

Zabbix プロキシは、リモート拠点、支店、ローカル管理者のいないネットワークの集中監視に最適なソリューションです。

Zabbix プロキシには、別途データベースが必要です。

> **Attention:**
> Zabbix プロキシでサポートされているデータベースは、SQLite、MySQL、PostgreSQL です。

参考: 分散環境でのプロキシの使用

プロキシの起動

パッケージとしてインストールした場合

Zabbix プロキシはデーモンプロセスとして実行されます。プロキシは次のコマンドで起動できます:

```
systemctl start zabbix-proxy
```

これはほとんどの GNU/Linux システムで動作します。他のシステムでは、次のコマンドを実行する必要があるかもしれません:

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

同様に、Zabbix プロキシの停止/再起動/ステータス表示には、以下のコマンドを使用します:

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

手動で起動する

上記でうまくいかない場合は、手動で起動する必要があります。zabbix_proxy バイナリのパスを探して、以下を実行します:

```
zabbix_proxy
```

Zabbix プロキシでは、以下のコマンドラインパラメータが使用できます:

```
-c --config <file>              設定ファイルへのパス
-f --foreground                 フォアグラウンドで Zabbix プロキシを実行
-R --runtime-control <option>   管理機能を実行
-T --test-config                設定ファイルを検証して終了
-h --help                       このヘルプを表示
-V --version                    バージョン番号を表示
```

コマンドラインパラメータを指定して Zabbix プロキシを起動する例:

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

ランタイム制御

ランタイム制御オプション:

オプション	説明	ターゲット
config_cache_reload	設定キャッシュをリロードします。キャッシュが現在ロード中の場合は無視されます。 アクティブ Zabbix プロキシは Zabbix サーバーに接続し、設定データを要求します。 パッシブ Zabbix プロキシは、サーバーがプロキシに接続した次のタイミングで Zabbix サーバーから設定データを要求します。	
history_cache_clear[=**target**]	ID で指定されたアイテムのヒストリーキャッシュをクリアします。 アイテムのすべての値に影響しますが、最初と最後の値は除きます。	**target** - アイテムの ID
diaginfo[=**section**]	プロキシのログファイルに診断情報を収集します。	**historycache** - ヒストリーキャッシュの統計情報 **preprocessing** - 前処理マネージャの統計情報 **locks** - ミューテックスのリスト（*BSD* システムでは空）
snmp_cache_reload	SNMP キャッシュをリロードします ― すべてのホストの SNMP エンジンプロパティ（engine time、engine boots、engine id、credentials）をクリアします。SNMP の問題をトラブルシューティングする際に、グローバルキャッシュのクリアを強制するために使用します。	
housekeeper_execute	ハウスキーピング手順を開始します。ハウスキーピング手順が現在進行中の場合は無視されます。	
log_level_increase[=**target**]	ログレベルを上げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	**process type** - 指定したタイプのすべてのプロセス（例: poller） すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: poller,3） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを'process type,N' として指定してください。
log_level_decrease[=**target**]	ログレベルを下げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	

オプション	説明	ターゲット
prof_enable[=<**target**>]	プロファイリングを有効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。 有効なプロファイリングは、関数名ごとのすべてのrwlock/mutex の詳細を提供します。	**process type** - 指定したタイプのすべてのプロセス（例: history syncer） すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: history syncer,1） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを'process type,N' として指定してください。 **scope** - `rwlock`, `mutex`, `processing` はプロセスタイプと番号（例: history syncer,1,processing）またはタイプのすべてのプロセス（例: history syncer,rwlock）で使用できます
prof_disable[=<**target**>]	プロファイリングを無効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。	**process type** - 指定したタイプのすべてのプロセス（例: history syncer） すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: history syncer,1） **pid** - プロセス ID（1～65535）。より大きな値の場合は、ターゲットを'process type,N' として指定してください。

プロキシ設定キャッシュをリロードするためのランタイム制御の使用例:

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

アイテムのヒストリーキャッシュをクリアするためのランタイム制御の使用例:

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

診断情報を収集するためのランタイム制御の使用例:

```
### プロキシログファイルに利用可能なすべての診断情報を収集:
zabbix_proxy -R diaginfo

### プロキシログファイルにヒストリーキャッシュの統計情報を収集:
zabbix_proxy -R diaginfo=historycache
```

SNMP キャッシュをリロードするためのランタイム制御の使用例:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
SNMPv3 インターフェースが Zabbix UI 経由で更新された場合、ほとんどの場合 Zabbix はそのインターフェースの新しい SNMPv3 認証情報を自動的にリロードします。認証情報の変更後もポーリングが失敗する場合（たとえば、engineBoots/engineID の不整合や RFC 非準拠デバイスなど）、またはトラブルシューティングのためにグローバルな SNMP キャッシュのクリアを強制する必要がある場合のみ、`-R snmp_cache_reload` を使用してください。

ハウスキーパーの実行をトリガーするためのランタイム制御の使用例:

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

ログレベルを変更するためのランタイム制御の使用例:

```
### すべてのプロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### 2 番目のポーラープロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234
```

```
### すべての http poller プロセスのログレベルを下げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

プロセスユーザー

Zabbix プロキシは、非 root ユーザーとして実行するように設計されています。起動した非 root ユーザーとして実行されます。したがって、プロキシは任意の非 root ユーザーとして問題なく実行できます。

'root' として実行しようとすると、ハードコードされた'zabbix' ユーザーに切り替わります。このユーザーはシステム上に存在する必要があります。プロキシを'root' として実行するには、プロキシの設定ファイルで'AllowRoot' パラメータを適切に変更する必要があります。

設定ファイル

zabbix_proxy の設定の詳細については、設定ファイルのオプションを参照してください。

プロキシプロセスタイプとスレッド

- agent poller - ワーカースレッドを持つパッシブチェック用の非同期ポーラープロセス
- availability manager - ホストの可用性更新用プロセス
- browser poller - ブラウザアイテムチェック用ポーラー
- configuration syncer - 設定データのインメモリキャッシュ管理プロセス
- data sender - プロキシデータ送信プロセス
- discovery manager - デバイス検出用のマネージャープロセス
- discovery worker - ディスカバリーマネージャからの検出タスクを処理するプロセス
- history syncer - ヒストリ DB 書き込みプロセス
- housekeeper - 古い履歴データの削除プロセス
- http agent poller - ワーカースレッドを持つ HTTP チェック用の非同期ポーラープロセス
- http poller - Web 監視用ポーラー
- icmp pinger - icmpping チェック用ポーラー
- internal poller - 内部チェック用ポーラー
- ipmi manager - IPMI ポーラーマネージャー
- ipmi poller - IPMI チェック用ポーラー
- java poller - Java チェック用ポーラー
- odbc poller - ODBC チェック用ポーラー
- poller - パッシブチェック用の通常のポーラー
- preprocessing manager - 前処理ワーカースレッドを持つ前処理タスクのマネージャー
- preprocessing worker - データ前処理用スレッド
- self-monitoring - 内部サーバ統計収集プロセス
- snmp poller - SNMP チェック用のワーカースレッドを持つ非同期ポーラープロセス（walk[OID] および get[OID] アイテムのみ）
- snmp trapper - SNMP トラップ用トラッパー
- task manager - 他のコンポーネントから要求されたタスクのリモート実行用プロセス（例：問題のクローズ、問題の確認、アイテム値の即時チェック、リモートコマンド機能）
- trapper - アクティブチェック、トラップ、プロキシ通信用トラッパー
- unreachable poller - 到達不能デバイス用ポーラー
- vmware collector - VMware サービスからのデータ収集を担当する VMware データコレクター

これらのプロセスタイプはプロキシのログファイルで確認できます。

プロキシのログファイルは、ファイル所有者のみが読み書きできる権限で作成されます。さらに、所有グループも読み取り可能です。他のすべての権限は拒否されます。

さまざまなタイプの Zabbix プロキシプロセスは、**zabbix[process,<type>,<mode>,<state>]** 内部アイテムを使用して監視できます。

ヒストリーシンカーのトランザクション統計

history syncer プロセスのタイトルには、history syncer トランザクションの詳細な統計情報が表示されます。

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

"processed...in N (<timings>) sec" のタイミングは以下の通りです:

- アイテムの値をデータベースに書き込むのにかかった時間
- アイテムデータ（状態、エラー）の更新にかかった時間

サポートされているプラットフォーム

Zabbix プロキシは、Zabbix サーバーと同じサポートされているプラットフォームで動作します。

メモリバッファ

メモリバッファは、新しいデータ（アイテム値、ネットワークディスカバリ、ホストの自動登録）をバッファに保存し、データベースにアクセスせずに Zabbix サーバーにアップロードすることを可能にします。メモリバッファは、Zabbix 7.0 からプロキシに導入されました。

Zabbix 7.0 以前のインストールでは、収集されたデータは Zabbix サーバーにアップロードする前にデータベースに保存されていました。これらのインストールでは、Zabbix 7.0 にアップグレードした後もデフォルトの動作として残ります。

最適なパフォーマンスのためには、プロキシでメモリバッファの使用を設定することを推奨します。これは、ProxyBufferModeの値を「disk」（既存インストールのハードコードされたデフォルト）から「hybrid」（推奨）または「memory」に変更することで可能です。また、メモリバッファサイズ（ProxyMemoryBufferSizeパラメータ）を設定する必要があります。

ハイブリッドモードでは、プロキシが停止した場合、バッファがいっぱいになった場合、またはデータが古くなった場合に、未送信データをデータベースにフラッシュすることで、バッファがデータ損失から保護されます。すべての値がデータベースにフラッシュされると、プロキシはメモリバッファの使用に戻ります。

メモリモードでは、メモリバッファが使用されますが、データ損失に対する保護はありません。プロキシが停止した場合やメモリがいっぱいになった場合、未送信データは破棄されます。

ハイブリッドモード（ProxyBufferMode=hybrid）は、Zabbix 7.0 以降のすべての新規インストールに適用されます。

ProxyMemoryBufferSizeやProxyMemoryBufferAgeなどの追加パラメータは、それぞれメモリバッファのサイズとバッファ内のデータの最大保持期間を定義します。

注意：設定が矛盾している場合、プロキシはエラーを出力して起動に失敗します。例えば、以下の場合です。

- ProxyBufferMode が「hybrid」または「memory」に設定されていて、ProxyMemoryBufferSize が「0」の場合
- ProxyBufferMode が「hybrid」または「memory」に設定されていて、ProxyLocalBuffer が「0」でない場合

ロケール

プロキシが一部のテキストアイテムを正しく解釈できるように、UTF-8 ロケールが必要であることに注意してください。ほとんどの最新の Unix 系システムではデフォルトで UTF-8 ロケールが設定されていますが、システムによっては明示的に設定する必要がある場合があります。

メンテナンス中のキューの計算

> **Attention:**
> Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

5 Zabbix Java ゲートウェイ

概要

Zabbix Java ゲートウェイは、ソースコードまたはパッケージからインストールできます。

JMX アプリケーションの監視をネイティブにサポートするために、「Zabbix Java ゲートウェイ」と呼ばれる Zabbix デーモンが用意されています。Zabbix Java ゲートウェイは Java で記述されたデーモンです。ホスト上の特定の JMX カウンタの値を取得するために、Zabbix サーバーは Zabbix Java ゲートウェイに問い合わせ、Zabbix Java ゲートウェイはJMX 管理 APIを使用して、対象のアプリケーションにリモートで問い合わせを行います。アプリケーションには追加のソフトウェアをインストールする必要はなく、コマンドラインで`-Dcom.sun.management.jmxremote` オプションを付けて起動するだけです。

Java ゲートウェイは Zabbix サーバーまたはプロキシからの着信接続を受け付け、「パッシブプロキシ」としてのみ使用できます。Zabbix プロキシとは異なり、Zabbix プロキシからも使用できます（Zabbix プロキシは連鎖できません）。各 Java ゲートウェイへのアクセスは、Zabbix サーバーまたはプロキシの設定ファイルで直接設定されるため、Zabbix サーバーまたは Zabbix プロキシごとに 1 つの Java ゲートウェイのみを設定できます。ホストに **JMX** エージェントタイプのアイテムと他のタイプのアイテムがある場合、**JMX** エージェントアイテムのみが Java ゲートウェイに渡されて取得されます。

Java ゲートウェイ経由でアイテムを更新する必要がある場合、Zabbix サーバーまたはプロキシは Java ゲートウェイに接続して値を要求し、Java ゲートウェイはその値を取得してサーバーまたはプロキシに返します。そのため、Java ゲートウェイは値をキャッシュしません。

Zabbix サーバーまたはプロキシには、オプション **StartJavaPollers** によって制御される、Java ゲートウェイに接続する特定のタイプのプロセスがあります。内部的には、Java ゲートウェイは **START_POLLERS**オプションによって制御される複数のスレッドを開始します。サーバー側では、接続に **Timeout** 秒以上かかると終了しますが、Java ゲートウェイは JMX カウンターから値を取得する

ためにまだビジーである可能性があります。これを解決するために、Java ゲートウェイには **TIMEOUT** オプションがあり、JMX ネットワーク操作のタイムアウトを設定できます。

Zabbix サーバーまたはプロキシは、可能な限り多くのリクエストを 1 つの JMX ターゲットにまとめてプールし（アイテムの間隔によって影響を受けます）、パフォーマンス向上のために 1 つの接続で Java ゲートウェイに送信しようとします。

StartJavaPollers は **START_POLLERS** 以下に設定することを推奨します。そうしないと、Java ゲートウェイで受信リクエストを処理するスレッドが利用できない状況が発生する可能性があります。その場合、Java ゲートウェイは ThreadPoolExecutor.CallerRunsPolicy を使用し、メインスレッドが受信リクエストを処理し、一時的に新しいリクエストを受け付けなくなります。

Zabbix Java ゲートウェイで Wildfly ベースの Java アプリケーションを監視しようとする場合は、Wildfly ダウンロードページで入手可能な最新の jboss-client.jar をインストールしてください。

1 ソースからのセットアップ

概要

ソースからインストールした場合、以下の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

ファイルの概要

ソースから Java ゲートウェイを取得した場合、シェルスクリプト、JAR、および設定ファイルのコレクションが $PREFIX/sbin/zabbix_java の下にあるはずです。これらのファイルの役割を以下にまとめます。

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

Java ゲートウェイ JAR ファイル本体

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

Java ゲートウェイの依存関係: Logback、SLF4J、および Android JSONライブラリ。

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

Logback の構成ファイル。

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

Java ゲートウェイを開始および停止するための便利なスクリプト。

`settings.sh`

上記の起動スクリプトとシャットダウンスクリプトによって提供される設定ファイル。

Java ゲートウェイの設定と実行

デフォルトでは、Java ゲートウェイはポート 10052 でリッスンします。別のポートで Java ゲートウェイを実行したい場合は、settings.sh スクリプトで指定できます。このオプションやその他のオプションの指定方法については、Java ゲートウェイ設定ファイルの説明を参照してください。

> **Warning:**
> ポート 10052 はIANAregisteredではありません。

設定完了後、起動スクリプトを実行して Java ゲートウェイを起動できます。

```
$ ./startup.sh
```

同様に Java ゲートウェイが不要になったら、シャットダウンスクリプトを実行して停止できます。

```
$ ./shutdown.sh
```

サーバーやプロキシとは異なり、Java ゲートウェイは軽量であり、データベースを必要としません。

Java ゲートウェイで使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイル で接続パラメータを指定します。

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

デフォルトでは、サーバーは JMX モニタリングに関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Java ゲートウェイに問題がある場合、またはフロントエンドのアイテムについて表示されるエラーメッセージが十分でない場合は、Java ゲートウェイのログファイルを確認することをお勧めします。

デフォルトでは、Java ゲートウェイはそのアクティビティを /tmp/zabbix_java.log ファイルにログレベル”info” で記録します。ですがその情報だけでは不十分な場合、ログレベル” デバッグ” の情報を取得する必要があります。ログレベルを上げるには、lib/logback.xml ファイルを変更し、<root> タグの level 属性を”debug” に変更します。

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

Zabbix サーバーや Zabbix プロキシとは異なり、logback.xml ファイルを変更した後に Zabbix Java ゲートウェイを再起動する必要はありません。logback.xml の変更は自動的に取得されます。デバッグが完了したら、ログレベルを”info” に戻すことができます。

別のファイルまたはデータベースのようなまったく別のメディアにログを記録する場合は、必要に応じて logback.xml ファイルを調整します。詳細については、Logback マニュアル を参照してください。

デバッグ目的の場合、デーモンではなくコンソールアプリケーションとして Java ゲートウェイを起動すると便利な場合があります。これを行うには、settings.sh で PID_FILE 変数をコメントアウトします。PID_FILE を省略した場合、startup.sh スクリプトは Java ゲートウェイをコンソールアプリケーションとして起動し、代わりに Logback に lib/logback-console.xml ファイルを使用させます。これにより、コンソールにログが記録されるだけでなく、ログレベル”debug” も有効になります。

最後に、Java ゲートウェイはロギングに SLF4J を使用するため、適切な JAR ファイルを lib ディレクトリに配置することで、Logback を選択したフレームワークに置き換えることができます。詳細については、SLF4J マニュアル を参照してください。

JMX モニタリング

詳しくはJMX モニタリングページをご覧ください。

2 RHEL パッケージでのセットアップ

概要

パッケージからインストールした場合、以下の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

Java ゲートウェイの設定と実行

Zabbix Java ゲートウェイの設定パラメーターは、次のファイルで調整できます。

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

詳細については、Zabbix Java ゲートウェイ設定パラメーターを参照してください。

Zabbix Java ゲートウェイを起動するには:

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

サーバー起動時に Zabbix Java ゲートウェイを自動的に起動するには:

RHEL 7 以降:

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

RHEL 7 より前:

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

Java ゲートウェイで使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメータを指定します。

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

デフォルトでは、サーバーは JMX モニタリングに関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

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

ログの内容を増やしたい場合は、下記ファイルを編集します。

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

level="info" を”debug” または”trace” に変更します (詳細なトラブルシューティング用):

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

</configuration>
```

JMX モニタリング

詳しくはJMX モニタリングページをご覧ください。

3 Debian/Ubuntu パッケージでのセットアップ

概要

パッケージからインストールした場合、以下の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

Java ゲートウェイの設定と実行

Java ゲートウェイの設定は次のファイルで行えます。

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

詳細については、Zabbix Java ゲートウェイ設定パラメーターを参照してください。

Zabbix Java ゲートウェイを起動するには:

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

サーバー起動時に Zabbix Java ゲートウェイを自動的に起動するには:

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

Java ゲートウェイで使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメータを指定します。

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

デフォルトでは、サーバーは JMX モニタリングに関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

/var/log/zabbix/zabbix_java_gateway.log

ログの内容を増やしたい場合は、下記ファイルを編集します。

/etc/zabbix/zabbix_java_gateway_logback.xml

level="info" を”debug” または”trace” に変更します (詳細なトラブルシューティング用):

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

</configuration>
```

JMX モニタリング

詳しくはJMX モニタリングページをご覧ください。

6 zabbix_sender コマンド

概要

Zabbix sender はパフォーマンスデータを Zabbix サーバーに送信して処理するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、可用性とパフォーマンスのデータを定期的に送信するために、長時間実行されるユーザースクリプトで使用されます。

結果を Zabbix サーバーまたはプロキシに直接送信するにはtrapper アイテムタイプを設定する必要があります。

参照: zabbix_utils - Zabbix sender のように動作する機能が組み込まれた Python ライブラリ

Zabbix sender の実行

Zabbix UNIX sender の実行例:

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

解説:

- z - Zabbix サーバーのホスト (IP アドレスも使用可能)
- s - 監視対象ホストの名称 (Zabbix フロントエンドに登録されているもの)
- k - アイテムキー
- o - 送信値

Attention:
空白を含むオプションは、ダブルクォーテーションで囲む必要があります。

Zabbix sender を使用して入力ファイルから複数の値を送信できます。詳細については、Zabbix sender のマニュアルページを参照してください。

設定ファイルが指定されている場合、Zabbix sender はエージェントの ServerActive 設定パラメータで定義されているすべてのアドレスを使用してデータを送信します。バッチデータの送信が 1 つのアドレスへの送信で失敗した場合、他のアドレスへの送信を試み、次のバッチはこのアドレスに送信されません。

Zabbix sender はファイルの最初にバイト順マーク (BOM) がない UTF-8 エンコーディング (UNIX ライクなシステムと Windows の両方) の文字列を受け入れます。

Windows 上の Zabbix sender も同様に実行できます:

```
zabbix_sender.exe [options]
```

zabbix_sender のリアルタイム送信シナリオは、連続して渡される複数の値を収集し、単一の接続でサーバーに送信します。前の値から 0.2 秒以内の値を同じスタックに入れることができますが、最大ポーリング時間は依然として 1 秒です。

Note:
指定された設定ファイルに無効な (*parameter=value* 表記に従っていない) パラメータエントリが存在する場合、Zabbix sender は終了します。

ローレベルディスカバリで Zabbix sender を実行する

ローレベルディスカバリの JSON 形式の値を送信する Zabbix sender を実行する例:

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

これが機能するには、ローレベルディスカバリルールに Zabbix トラッパーアイテムタイプ (この例では、`trapper.discovery.item` キー) が必要です。

7 zabbix_get コマンド

概要

Zabbix get は、Zabbix エージェントと通信し、エージェントから必要な情報を取得するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、Zabbix エージェントのトラブルシューティングに使用されます。

Zabbix get のように動作する組み込み機能を持つ Python ライブラリである zabbix_utils も参照してください。

Zabbix get の実行

UNIX で Zabbix get を実行して、エージェントからプロセッサの負荷値を取得する例:

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

Zabbix get を実行して Web サイトから文字列を取得する別の例:

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

ここでのアイテムキーにはスペースが含まれているため、シェルにアイテムキーを示すために引用符が使用されています。引用符はアイテムキーの一部ではなく、シェルによって削除され、Zabbix エージェントには渡されません。

アイテムキーがサポートされていない場合、Zabbix get は終了コード 1 を返します。

Zabbix get は以下のコマンドラインパラメータを受け付けます:

```
-s --host <host name or IP>              ホスト名または IP アドレスを指定
-p --port <port number>                  ホスト上で実行されているエージェントのポート番号を指定 (デフォルト: 10050)
-I --source-address <IP address>         ソース IP アドレスを指定
-t --timeout <seconds>                   タイムアウトを指定。有効範囲: 1-600 秒 (デフォルト: 30 秒)
-k --key <item key>                      取得するアイテムのキーを指定
-P --protocol <value>                    エージェントとの通信に使用するプロトコル。値:
                                             auto - JSON プロトコルで接続し、失敗した場合はプレーンテキストプロトコル
                                             json - JSON プロトコルで接続
                                             plaintext - アイテムキーのみを送信するプレーンテキストプロトコルで接続 (
-h --help                                Display this help message
-V --version                             バージョン番号を表示

--tls-connect <value>                    エージェントへの接続方法。値:
                                             unencrypted - 暗号化せずに接続 (デフォルト)
                                             psk - TLS および事前共有鍵を使用して接続
                                             cert - TLS および証明書を使用して接続
--tls-ca-file <CA file>                  ピア証明書の検証用の最上位 CA 証明書を含むファイルのフルパス
--tls-crl-file <CRL file>                失効証明書を含むファイルのフルパス
--tls-agent-cert-issuer <cert issuer>    許可するエージェント証明書の発行者
--tls-agent-cert-subject <cert subject>  許可するエージェント証明書のサブジェクト
--tls-cert-file <cert file>              証明書または証明書チェーンを含むファイルのフルパス
--tls-key-file <key file>                秘密鍵を含むファイルのフルパス
--tls-psk-identity <PSK-identity>        事前共有鍵を識別するための一意で大文字小文字を区別する文字列
--tls-psk-file <PSK-file>                事前共有鍵を含むファイルのフルパス
```

```
--tls-cipher13 <cipher-string>          TLS 1.3 用の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準
--tls-cipher <cipher-string>            GnuTLS の優先順位文字列（TLS 1.2 以降）または OpenSSL の暗号文字列（TLS 1
```

詳細はZabbix get man ページも参照してください。

Windows 上の Zabbix get も同様に実行できます:

```
zabbix_get.exe [options]
```

8 zabbix_js コマンド

概要

zabbix_js は、埋め込みスクリプトのテストに使用できるコマンドラインユーティリティです。

このユーティリティは、文字列パラメータでユーザースクリプトを実行し、結果を出力します。スクリプトは、埋め込み Zabbix スクリプトエンジンを使用して実行されます。

コンパイルまたは実行エラーが発生した場合、zabbix_js はエラーを stderr に出力し、コード 1 で終了します。

使用方法

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

zabbix_js は以下のコマンドラインパラメータを受け付けます:

```
-s, --script script-file            実行するスクリプトのファイル名を指定します。ファイル名に'-'を指定すると、スクリプトは標
-i, --input input-file              入力パラメータのファイル名を指定します。ファイル名に'-'を指定すると、入力は標準入力から
-p, --param input-param             入力パラメータを指定します。
-l, --loglevel log-level            ログレベルを指定します。
-t, --timeout timeout               タイムアウトを秒単位で指定します。有効範囲: 1-600 秒（デフォルト: 10 秒）。
-h, --help                          ヘルプ情報を表示します。
-V, --version                       バージョン番号を表示します。
-w <webdriver url>                  ブラウザ監視を有効にします。
```

例:

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

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部 Web サービスとの通信に使用されるプロセスです。現在、Zabbix Web サービスは、定期レポートの生成と送信に使用されており、将来的には機能が追加される予定です。

Zabbix サーバーは HTTP(S) 経由で Web サービスに接続します。Zabbix Web サービスはGoogle Chromeが同じホストにインストールされている必要があります。一部のディストリビューションでは Chromium でも機能する場合があります (既知の問題)。

インストール

公式の zabbix-web-service パッケージは、Zabbix リポジトリで入手できます。

Zabbix Web サービスをソースからコンパイルするには、`--enable-webservice` の configure オプションを指定してください。

Zabbix Web サービスを設定するには、`zabbix_web_service.conf`の設定ファイルパラメータを更新してください。

> **Attention:**
> Zabbix サーバーと Zabbix Web サービス間の暗号化は、証明書を使用して設定することを強く推奨します。デフォルトでは、Zabbix サーバーと Zabbix Web サービス間で送信されるデータは暗号化されていないため、不正アクセスにつながる可能性があります。

5 設定

サイドバーを使用して、設定セクションのコンテンツにアクセスしてください。

1 テンプレートの設定

概要

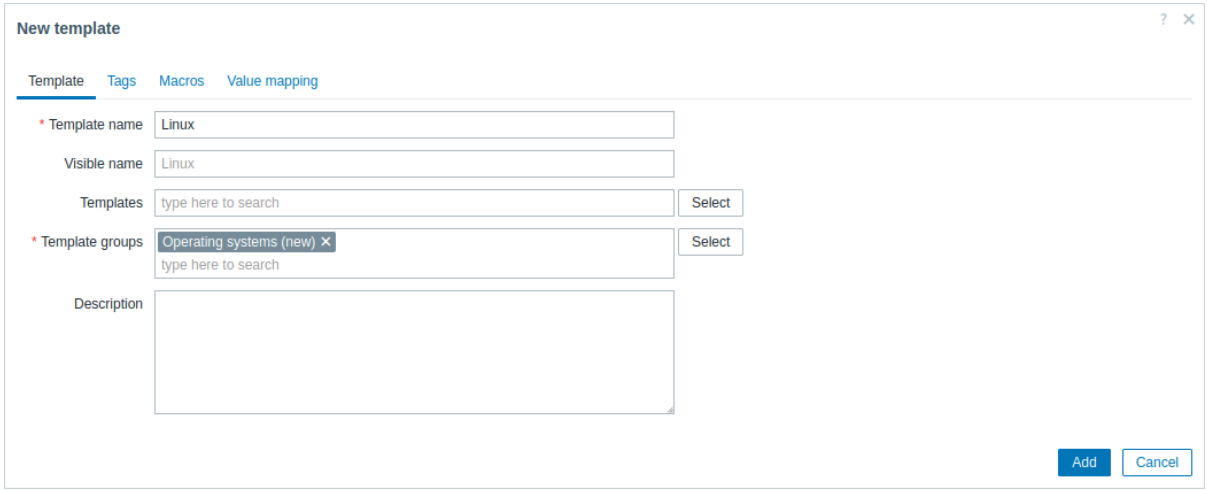
テンプレートを設定するには、最初に一般的なパラメーターを定義してテンプレートを作成し、次にエンティティ (アイテム、トリガー、グラフなど) をテンプレートに追加する必要があります。

テンプレートの作成

テンプレートを作成するには、以下の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. テンプレートの作成をクリックします。
3. テンプレート属性を編集します。

テンプレートタブには、テンプレートの一般的な属性が含まれています。



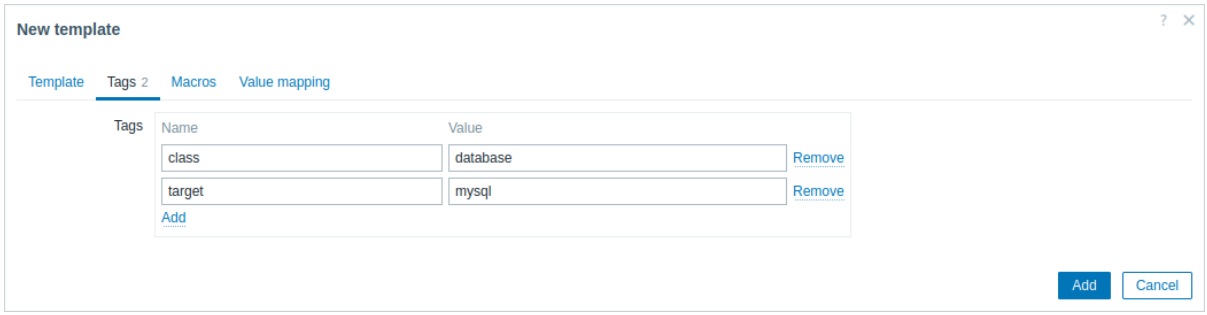
必須入力フィールドは赤いアスタリスクでマークされています。

テンプレート属性:

パラメータ	説明
テンプレート名	一意のテンプレート名。 英数字、スペース、ドット、ダッシュ、アンダースコアが使用できます。 先頭および末尾のスペースは使用できません。
表示名	この名前を設定すると、リストやマップなどで表示されます。
テンプレート	1 つ以上のテンプレートをこのテンプレートにリンクします。すべてのエンティティ（アイテム、トリガーなど）は、リンクされたテンプレートから継承されます。 新しいテンプレートをリンクするには、テンプレートフィールドにその名前を入力します。これにより、一致するテンプレートのドロップダウンリストが表示されます。または、選択をクリックして テンプレートポップアップを開きます。最初はテンプレートが表示されません。リストを表示するには、テンプレートグループフィールドにテンプレートグループ名を入力して（候補リストから 1 つ選択)、または テンプレートグループフィールドの横にある 選択ボタンを押して テンプレートグループポップアップを開きます。テンプレートグループを選択すると、テンプレートポップアップがそのグループに属するテンプレートを表示するように更新されます。選択したテンプレートは、設定を保存または更新したときにリンクされます。 テンプレートのリンクを解除するには、テンプレートブロックの 2 つのオプションのいずれかを使用します。 リンク解除 - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガーなど）は保持されます。 リンク解除とクリア - テンプレートのリンクを解除し、そのエンティティ（アイテム、トリガーなど）をすべて削除します。
テンプレートグループ	テンプレートが属するテンプレートグループ。

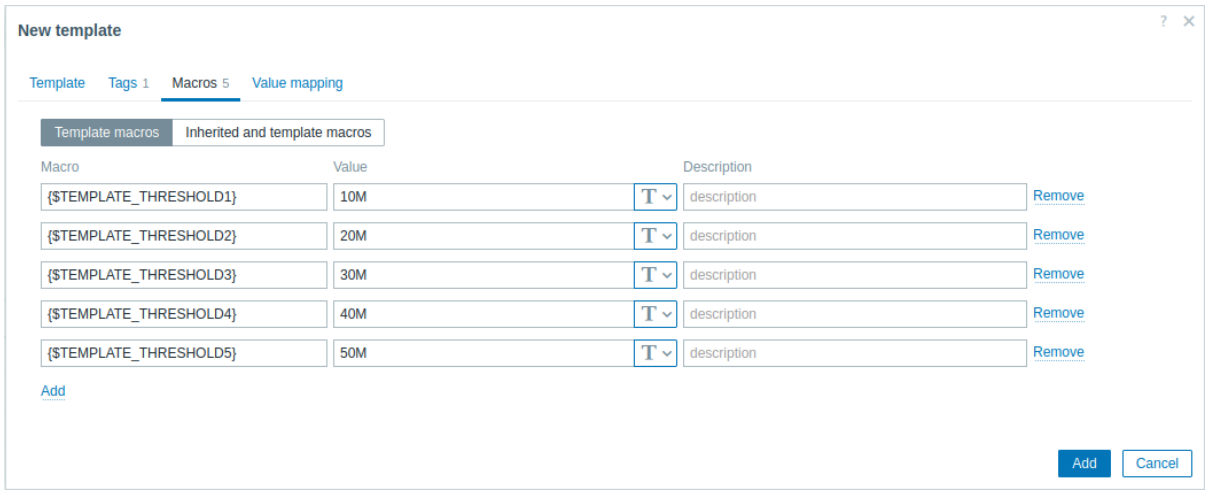
パラメータ	説明
説明	テンプレートの説明。
ベンダーとバージョン	テンプレートのベンダーとバージョン。既存のテンプレート（Zabbix が提供する標準テンプレート、インポートしたテンプレート、またはテンプレート APIで変更されたテンプレート）の更新時にのみ表示されます。テンプレート設定にその情報が含まれている場合のみ表示されます。 Zabbix フロントエンドでは変更できません。 標準テンプレートの場合、バージョンは次のように表示されます：Zabbix のメジャーバージョン、区切り文字（”-”）、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます）。例：7.0-0、7.0-5、8.0-0、8.0-3。

タグタブでは、テンプレートレベルのタグを定義できます。このテンプレートにリンクされたホストのすべての問題は、ここで入力した値でタグ付けされます。

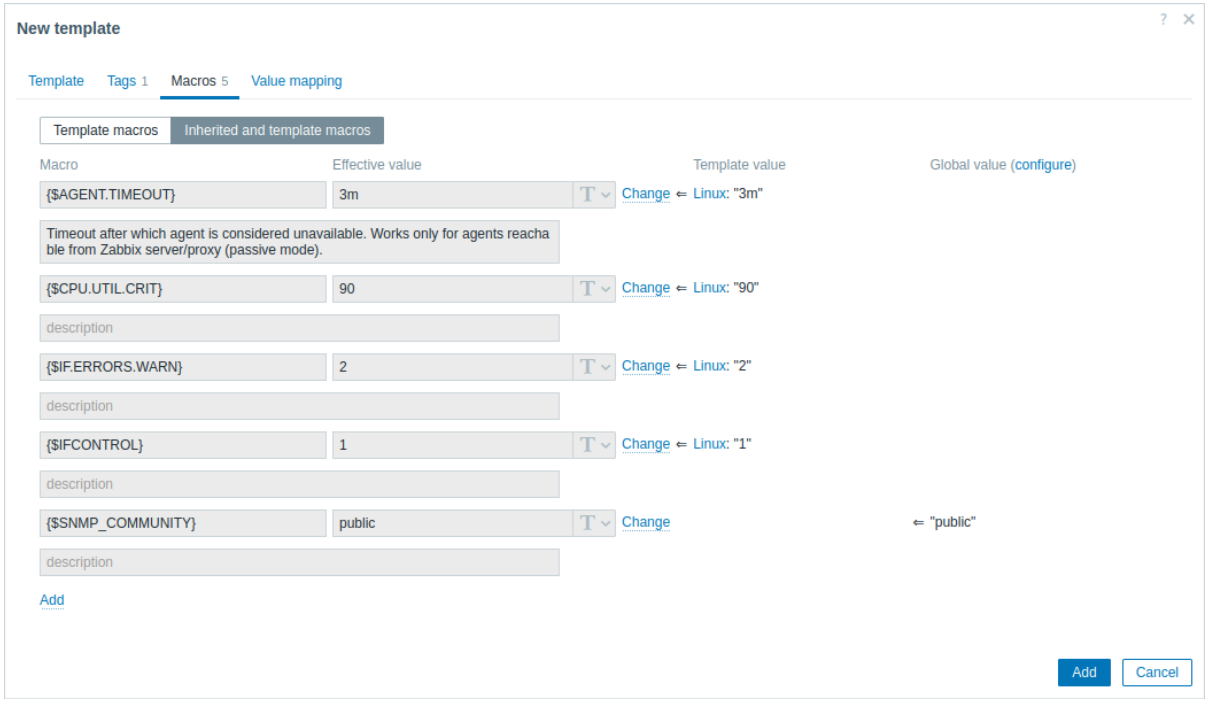


タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。

マクロタブでは、テンプレートレベルのユーザーマクロを名前と値のペアとして定義できます。マクロ値はプレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できます。説明の追加もサポートされています。



継承およびテンプレートマクロオプションを選択すると、リンクされたテンプレートやグローバルマクロから継承されるマクロ、およびマクロが解決される値も表示できます。



便宜上、該当するテンプレートへのリンクや、グローバルマクロ設定へのリンクも用意されています。また、テンプレートレベルでリンクされたテンプレートマクロやグローバルマクロを編集し、テンプレート上にマクロのコピーを作成することも可能です。

値のマッピングタブでは、値のマッピングでアイテムデータの人間に分かりやすい表現を設定できます。

ボタン:

Add	テンプレートを追加します。追加したテンプレートはリストに表示されます。
Update	既存のテンプレートのプロパティを更新します。
Clone	現在のテンプレートのプロパティに基づいて別のテンプレートを作成します。これには、リンクされたテンプレートから継承されたエンティティ（アイテム、トリガーなど）と、現在のテンプレートに直接関連付けられているエンティティの両方が含まれますが、クローンテンプレートが元のテンプレートと区別できるように、現在のテンプレートのベンダーとバージョンは含まれません。
Delete	テンプレートを削除します。テンプレートのエンティティ（アイテム、トリガーなど）はリンクされたホストに残ります。
Delete and clear	テンプレートとそのすべてのエンティティをリンクされたホストから削除します。
Cancel	テンプレートプロパティの編集をキャンセルします。

アイテム、トリガー、グラフの追加

Attention:
アイテムは最初にテンプレートに追加する必要があります。トリガーやグラフは、対応するアイテムがなければ追加できません。

テンプレートにアイテムを追加する方法は 2 つあります。

1. 新しいアイテムを作成するには、アイテムの作成のガイドラインに従ってください。

2. 既存のアイテムをテンプレートに追加するには:

- データ収集 → ホスト（またはテンプレート）に移動します。
- 必要なホスト/テンプレートの行でアイテムをクリックします。
- テンプレートに追加したいアイテムのチェックボックスをオンにします。

- アイテムリストの下にあるコピーをクリックします。
- アイテムをコピーするテンプレート（またはテンプレートグループ）を選択し、コピーをクリックします。
  選択したすべてのアイテムがテンプレートにコピーされます。

トリガーやグラフの追加も同様の方法で行います（それぞれトリガーやグラフのリストから）。ただし、必要なアイテムが先に追加されている場合のみ追加できることに注意してください。

ダッシュボードの追加

データ収集 → テンプレートでテンプレートにダッシュボードを追加するには、以下の手順を実行します。

1. テンプレートの行でダッシュボードをクリックします。
2. ダッシュボードの設定のガイドラインに従ってダッシュボードを設定します。

Attention:
テンプレートダッシュボード（グローバルダッシュボードではなく）でウィジェットを設定する場合、ホスト関連のパラメータは利用できず、一部のパラメータは異なるラベルになります。これは、テンプレートダッシュボードがテンプレートがリンクされているホストからのみデータを表示するためです。たとえば、障害ウィジェットのホストグループ、ホストグループを除外、ホストパラメータは利用できず、ホストの可用性ウィジェットのホストグループパラメータは利用できず、メンテナンス中のホストを表示パラメータはメンテナンス中のデータを表示に名前が変更されるなどです。テンプレートダッシュボードウィジェットで利用可能なパラメータの詳細については、各ダッシュボードウィジェットの特定のパラメータを参照してください。

Note:
テンプレートダッシュボードから作成されたホストダッシュボードへのアクセスの詳細については、ホストダッシュボードセクションを参照してください。

ローレベルディスカバリの設定

マニュアルのローレベルディスカバリセクションを参照

Web シナリオの追加

データ収集 → テンプレートでテンプレートに Web シナリオを追加するには、以下の手順を実行します。

1. テンプレートの行で *Web* をクリックします。
2. Web シナリオの設定の通常の方法に従って Web シナリオを設定します。

2 テンプレートグループの設定

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

設定

Attention:
テンプレートグループの作成は、スーパ管理者ユーザーのみが可能です。

Zabbix フロントエンドでテンプレートグループを作成する方法は 2 つあります。

方法 **1**:

- データ収集 → テンプレートグループ に移動します
- 画面右上のテンプレートグループの作成 をクリックします
- フォームにグループ名を入力します



方法 **2**: テンプレートの設定時に、テンプレートグループ入力フィールドに存在しないグループ名を入力します。

テンプレートグループが作成されると、データ収集 → テンプレートグループのリストでテンプレート名をクリックして、グループ名の編集、グループのクローン作成、グループの削除ができます。

テンプレートグループを削除しても、グループ内のテンプレートは削除されず、論理グループのみが削除されます。既存のテンプレートの唯一のグループであるテンプレートグループは削除できません。

テンプレートサブグループの作成

テンプレートサブグループ（またはネストされたテンプレートグループ）は、それを含む親テンプレートグループの子です。

サブグループは、グループ名入力フィールドでスラッシュ'/' を使用して親グループとの関係を示すことで作成されます。例:

- `Linux servers/Databases` と入力すると、親グループ `Linux servers` のサブグループ `Linux servers/Databases` が作成されます。
- `Linux servers/Databases/MySQL/Tokyo` と入力すると、ネストされた親グループ `Linux servers`、`Linux servers/Databases`、`Linux servers/Databases/MySQL` 内にそれぞれのサブグループが作成されます。

サブグループを作成する際、先頭または末尾にスラッシュを付けたり、連続して複数のスラッシュを使用することはできません。'/' のエスケープはサポートされていません。

サブグループを作成する前に、親テンプレートグループを作成する必要はありません。サブグループ（例: `Linux servers/Databases`）から作成を開始するか、親テンプレートグループ（この例では `Linux servers`）から作成を開始するかを選択できます。サブグループから作成を開始した場合、親テンプレートグループは自動的には作成されません。

テンプレートグループへの権限

- 既存の親テンプレートグループにサブグループを作成する場合（例: `Linux servers` が既に存在する場合に `Linux servers/Databases` を作成）、ユーザーグループのサブグループへの権限は親グループから継承されます。
- 既存のサブグループに親テンプレートグループを作成する場合（例: `Linux servers/Databases` が既に存在する場合に `Linux servers` を作成）、親グループへの権限は設定されません。

テンプレートグループを編集する際には、追加オプションとしてすべてのサブグループに権限を適用を設定することもできます。

このチェックボックスをマークして更新をクリックすると、編集中のテンプレートグループのすべての現在および将来のサブグループに同じレベルの権限が適用されます。

したがって、編集中のテンプレートグループのサブグループに対して異なる権限がユーザーグループに付与されている場合でも、チェックボックスをマークすると、すべての現在および将来のサブグループに編集中のグループと同じユーザー権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きすることに注意してください。このオプションによる変更は手動でのみ元に戻すことができます。

3 リンク作成/リンク削除

概要

リンクはテンプレートがホストに適用されるプロセスであり、リンク解除はホストからテンプレートとの関連付けを削除します。

テンプレートのリンク

テンプレートをホストにリンクするには、以下の手順を実行します。

1. データ収集 → ホストに移動します。
2. 必要なホストをクリックします。
3. テンプレートフィールドにテンプレート名の入力を開始します。一致するテンプレートのリストが表示されるので、スクロールして選択します。
 または、フィールドの横にある 選択をクリックし、ポップアップウィンドウのリストから 1 つまたは複数のテンプレートを選択することもできます。
4. ホスト属性フォームで 追加/更新をクリックします。

これでホストはテンプレートのすべてのエンティティを持つことになります。これには、アイテム、トリガー、グラフ、ローレベルディスカバリールール、Web シナリオ、ダッシュボードが含まれます。

Attention:
同じホストに複数のテンプレートをリンクする場合、それらのテンプレートに同じアイテムキーを持つアイテムが含まれていると失敗します。また、トリガーやグラフはアイテムを使用するため、同じアイテムキーを使用している場合、複数のテンプレートから 1 つのホストにリンクすることはできません。

テンプレートからエンティティ（アイテム、トリガーなど）が追加されるとき:

- 以前にホストに存在していた同一のエンティティはテンプレートのエンティティとして更新され、エンティティに対する既存のホストレベルのカスタマイズは失われます。
- テンプレートからエンティティが追加されます。
- テンプレートリンク前にホストのみに存在していた直接リンクされたエンティティはそのまま残ります。

リストでは、テンプレートからのすべてのエンティティがテンプレート名でプレフィックスされ、これらが特定のテンプレートに属していることを示します。テンプレート名自体（グレーのテキスト）はリンクになっており、テンプレートレベルでそれらのエンティティのリストにアクセスできます。

Note:
外部チェック、HTTP エージェントチェック、シンプルチェック、SSH チェック、Telnet チェックなど、一部のアイテムではホストインターフェースはオプションです。テンプレートをリンクする時点でホストにインターフェースが定義されていない場合、これらのアイテムはインターフェースなしで追加されます。後でホストインターフェースを追加しても、既存のアイテムには自動的に割り当てられません。新たに追加したホストインターフェースをすべてのテンプレートアイテムに一括で割り当てるには、テンプレートをアンリンクしてから再度リンクしてください。アイテムの履歴を保持するには アンリンクオプションを使用し、アンリンクしてクリアは使用しないでください。

テンプレート名でプレフィックスされていないエンティティがある場合、それは以前からホストに存在し、テンプレートによって追加されたものではないことを意味します。

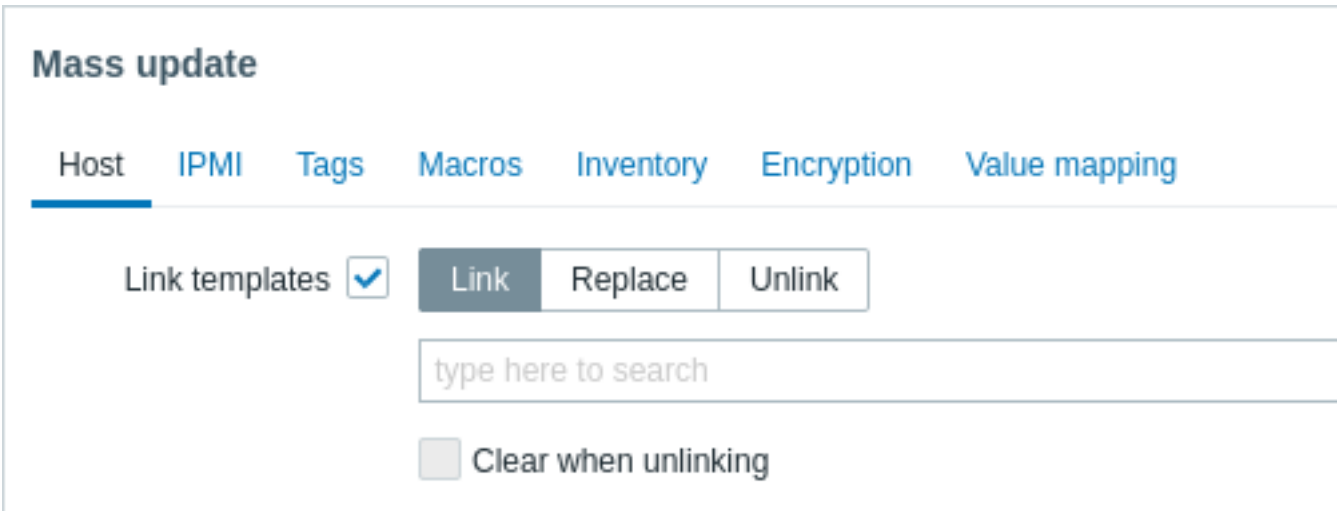
エンティティの一意性の基準

テンプレートからエンティティ（アイテム、トリガーなど）を追加する場合、ホスト上にすでに存在し更新が必要なエンティティと、異なるエンティティを知ることが重要です。同一性/差異を判断するための一意性の基準は以下の通りです。

- アイテムの場合 - アイテムキー
- トリガーの場合 - トリガー名と式
- カスタムグラフの場合 - グラフ名とそのアイテム

複数のホストへのテンプレートのリンク

多くのホストのテンプレートリンクを更新するには、データ収集 → ホストでいくつかのホストをチェックボックスで選択し、リストの下にある 一括更新 をクリックし、テンプレートのリンクを選択します。



追加のテンプレートをリンクするには、オートコンプリートフィールドにテンプレート名を入力し始めると、該当するテンプレートがドロップダウンで表示されます。スクロールしてリンクするテンプレートを選択してください。

置換オプションを使用すると、以前にホストにリンクされていたテンプレートをすべて解除し、新しいテンプレートをリンクできます。リンク解除オプションを使用すると、リンク解除するテンプレートを指定できます。リンク解除時にクリアオプションを使用す

ると、以前にリンクされていたテンプレートのリンクを解除するだけでなく、それらから継承されたすべてのエンティティ（アイテム、トリガーなど）も削除できます。

Note:
Zabbix には多くの定義済みテンプレートが用意されています。これらを参考にすることはできますが、アイテム数が多すぎたり、データの取得頻度が高すぎたりする場合があるため、変更せずに本番環境で使用することは避けてください。使用する場合は、実際のニーズに合わせて調整してください。

リンクされたエンティティの編集

テンプレートからリンクされたアイテムやトリガーを編集しようとすると、多くの主要なオプションが編集できないことに気付くかもしれません。これは、テンプレートの考え方が、テンプレートレベルで一括して編集することにあるため、理にかなっています。ただし、個々のホストでアイテムを有効/無効にしたり、更新間隔、履歴の長さ、その他のパラメータを設定したりすることは可能です。

エンティティを完全に編集したい場合は、テンプレートレベルで編集する必要があります（フォーム名にテンプレートレベルのショートカットが表示されます）。これらの変更は、このテンプレートがリンクされているすべてのホストに影響することに注意してください。

Attention:
テンプレートレベルで実装されたエンティティのカスタマイズは、ホストレベルでのエンティティの以前のカスタマイズを上書きします。

テンプレートのリンク解除

ホストからテンプレートのリンクを解除するには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. 必要なホストをクリックし、テンプレートフィールドを探します。
3. テンプレートの横にある リンク解除または リンク解除とクリアをクリックして、テンプレートのリンクを解除します。
4. ホスト属性フォームで 更新をクリックします。

リンク解除オプションを選択すると、テンプレートとの関連付けが解除されますが、そのエンティティはすべてホストに残ります。これにはアイテム、トリガー、グラフ、ローレベルディスカバリールール、Web シナリオが含まれますが、ダッシュボードは含まれません。リンクされたテンプレートから継承された値のマッピングやタグも削除されることに注意してください。

リンク解除とクリアオプションを選択すると、テンプレートとの関連付けとそのすべてのエンティティ（アイテム、トリガーなど）が削除されます。

4 ネスト

概要

ネストは、1 つのテンプレートが 1 つ以上の他のテンプレートを包含する方法です。

さまざまなサービス、アプリケーションなどの個別のテンプレートでエンティティを分離することは理にかなっていますが、最終的に大量のテンプレートが作成され、またそのすべてを大量の数のホストにリンクする必要になる可能性があります。単純化するために、複数のテンプレートを 1 つのテンプレートにリンクすることができます。

ネストの利点は、ホストにリンクするテンプレートが 1 つだけで済み、ホストはその 1 つのテンプレートにリンクされているテンプレートからすべてのエンティティを自動的に継承することです。たとえば、テンプレート *T1* と *T2* を *T3* にリンクすると、*T1* と *T2* のエンティティが *T3* に追加されますが、その逆はありません。*T1* を *T2* と *T3* にリンクすると、*T1* のエンティティが *T2* と *T3* に追加されます。

ネストされたテンプレートの設定

テンプレートをリンクするには、既存のテンプレートを取得するか、新しいテンプレートを作成する必要があります。そして次に:

1. テンプレート設定フォームを開きます。
2. テンプレートフィールドを見つけます。
3. 選択をクリックして、テンプレートポップアップウィンドウを開きます。
4. ポップアップウィンドウで必要なテンプレートを選択し、選択をクリックしてテンプレートをリストに追加します。
5. テンプレート設定フォームで追加または更新をクリックします。

したがって、設定されたテンプレートのすべてのエンティティと、リンクされたテンプレートのすべてのエンティティがテンプレート設定に表示されます。これには、アイテム、トリガー、グラフ、ローレベルディスカバリールール、Web シナリオが含まれますが、ダッシュボードは含まれません。ただし、リンクされたテンプレートダッシュボードはホストによって継承されます。

リンクされているテンプレートのリンクを解除するには、テンプレート設定フォームでリンクを削除またはリンクと保存データを削除をクリックし、更新をクリックします。

リンクを削除オプションは、リンクされたテンプレートとの関連付けを削除しますが、そのエンティティ (アイテム、トリガーなど) は削除されません。

リンクと保存データを削除オプションは、リンクされたテンプレートとの関連付けと、そのすべてのエンティティ (アイテム、トリガーなど) の両方が削除されます。

5 一括アップデート

概要

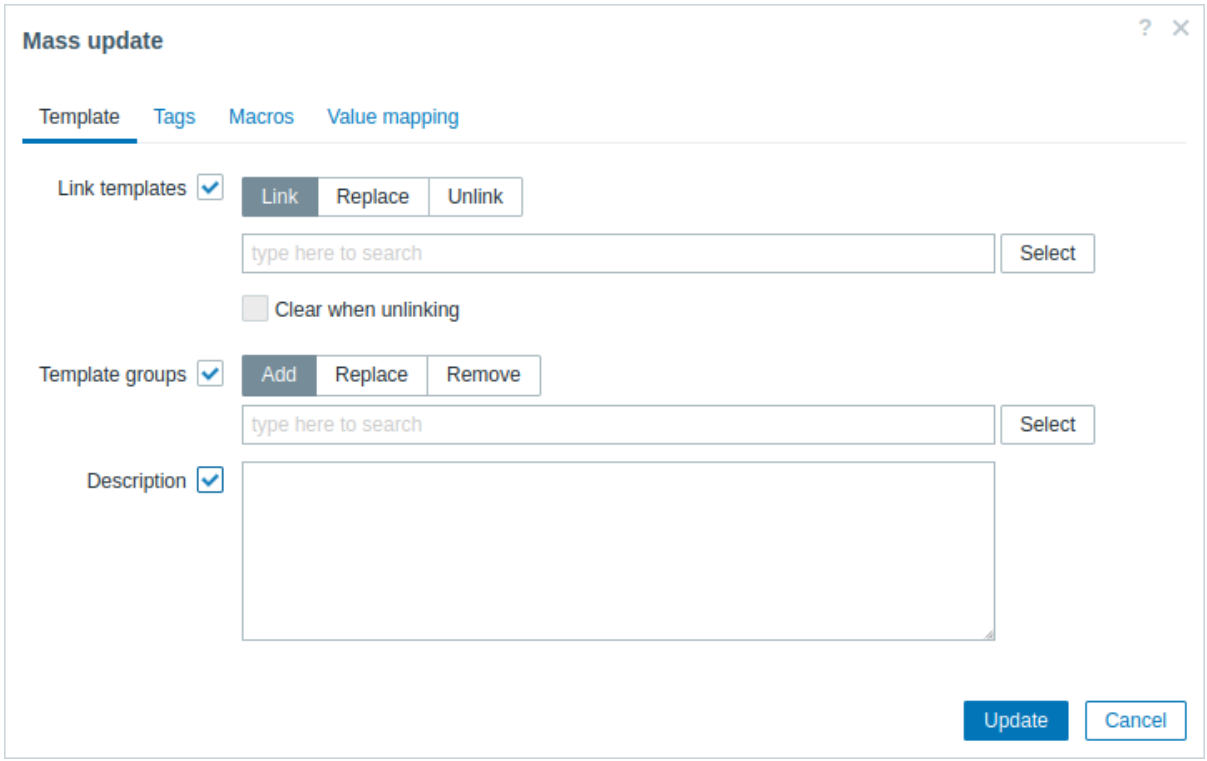
いくつかのテンプレートの属性を一度に変更したい場合があります。個々のテンプレートを開いて編集する代わりに、一括更新機能を使用できます。

一括更新の使用

いくつかのテンプレートを一括で更新するには、次の手順を実行します。

1. テンプレートリストで、更新したいテンプレートの前のチェックボックスをオンにします。
2. リストの下にある一括更新をクリックします。
3. 必要な属性のタブ（テンプレート、タグ、マクロ、値のマッピング）に移動します。
4. 更新する属性のチェックボックスをオンにし、新しい値を入力します。

テンプレートタブには、テンプレートの一括更新の一般的なオプションが含まれています。



リンクテンプレートの更新でそれぞれのボタンを選択した場合、以下のオプションが利用できます。

- リンク - 追加でリンクするテンプレートを指定します。
- 置換 - リンクするテンプレートを指定し、同時に以前にリンクされていたテンプレートのリンクを解除します。
- リンク解除 - リンクを解除するテンプレートを指定します。

リンク/リンク解除するテンプレートを指定するには、オートコンプリートフィールドにテンプレート名を入力し始め、ドロップダウンで一致するテンプレートが表示されるまで入力します。必要なテンプレートを選択するには、スクロールダウンしてください。

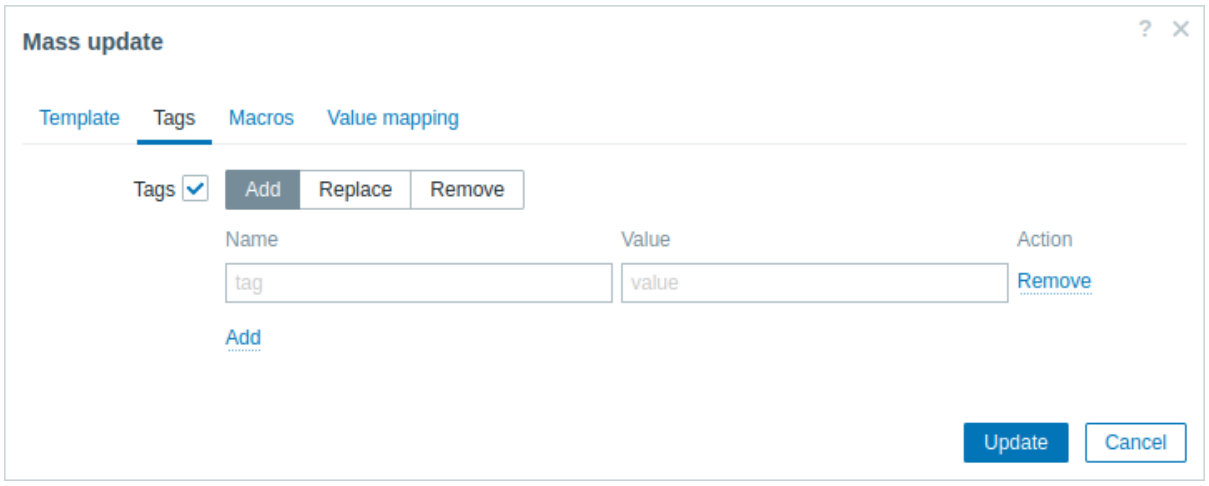
リンク解除時にクリアオプションを選択すると、以前にリンクされていたテンプレートのリンクを解除し、それらから継承されたすべての要素（アイテム、トリガー、グラフなど）も削除できます。

テンプレートグループの更新でそれぞれのボタンを選択した場合、以下のオプションが利用できます。

- 追加 - 既存のテンプレートグループから追加で指定するか、テンプレート用にまったく新しいテンプレートグループを入力できます。
- 置換 - テンプレートを既存のテンプレートグループから削除し、このフィールドで指定したテンプレートグループ（既存または新規）に置き換えます。
- 削除 - テンプレートから特定のテンプレートグループを削除します。

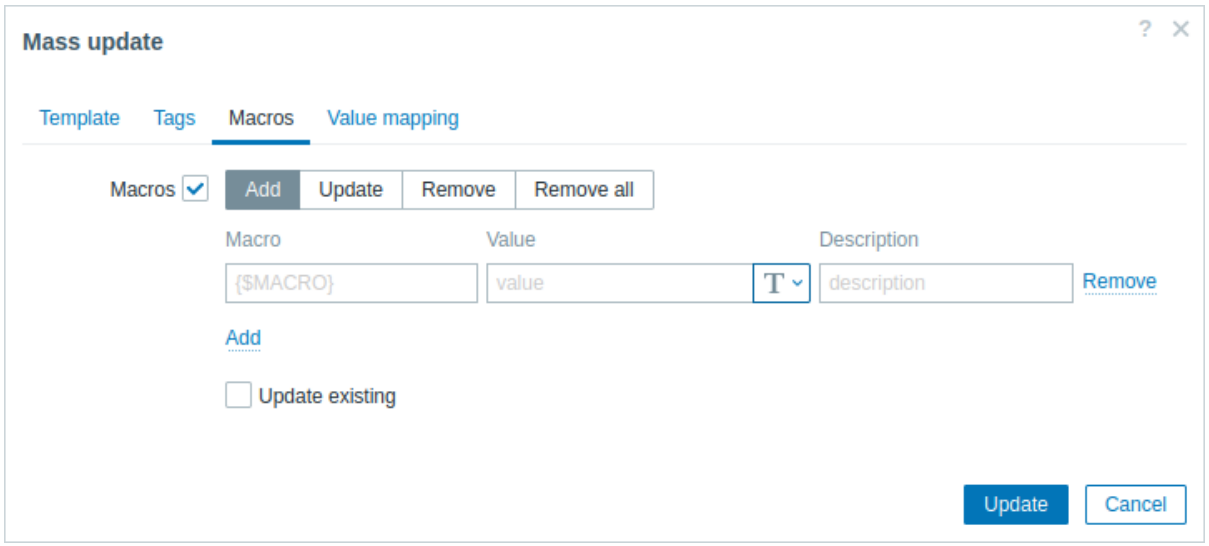
これらのフィールドはオートコンプリートです。入力を開始すると、一致するテンプレートグループのドロップダウンが表示されます。テンプレートグループが新しい場合は、ドロップダウンにも表示され、文字列の後に *(new)* と表示されます。スクロールダウンして選択してください。

タグタブでは、テンプレートレベルのタグを一括更新できます。



タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。同じ名前でも異なる値のタグは「重複」とみなされず、同じテンプレートに追加できます。

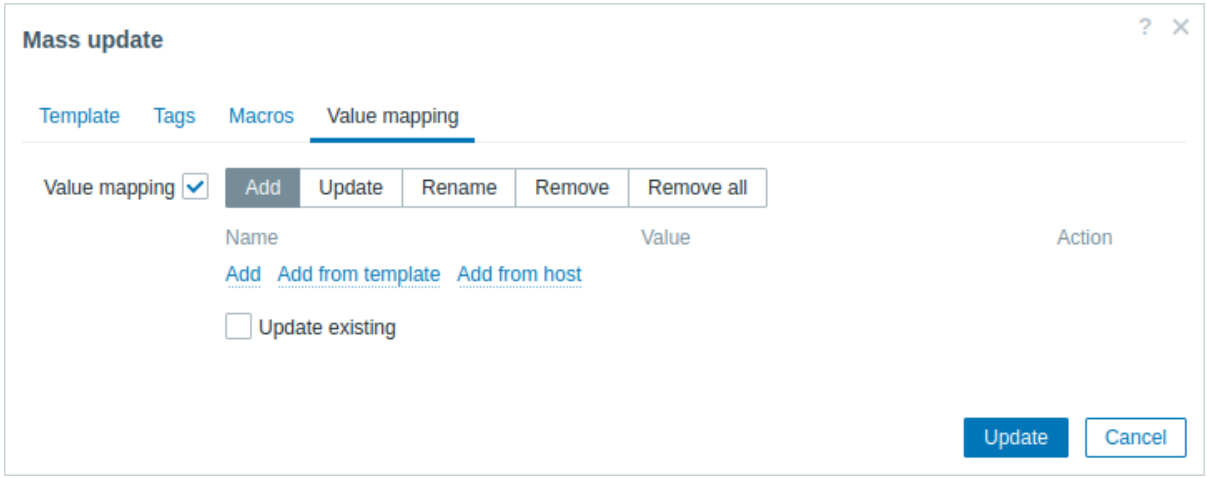
マクロタブでは、テンプレートレベルのマクロを一括更新できます。



マクロの更新でそれぞれのボタンを選択した場合、以下のオプションが利用できます:

- 追加 - テンプレートに追加のユーザーマクロを指定できます。既存のものを更新チェックボックスがオンの場合、指定したマクロ名の値、タイプ、説明が更新されます。オフの場合、テンプレートに同じ名前のマクロが既に存在していても更新されません。
- 更新 - このリストで指定したマクロの値、タイプ、説明を置き換えます。不足分を追加チェックボックスがオンの場合、テンプレートに存在しなかったマクロが新しいマクロとして追加されます。オフの場合、テンプレートに既に存在するマクロのみが更新されます。
- 削除 - 指定したマクロをテンプレートから削除します。選択したもの以外ボックスがオンの場合、リストで指定したもの以外のすべてのマクロが削除されます。オフの場合、リストで指定したマクロのみが削除されます。
- すべて削除 - テンプレートからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認しますチェックボックスがオンでない場合、すべてのマクロの削除を確認する新しいポップアップウィンドウが開きます。

値のマッピングタブでは、値のマッピングを一括更新できます。



値のマッピングの更新には、次のオプションのボタンが利用できます。

- 追加 - テンプレートに値のマッピングを追加します。既存のものを更新をマークすると、この名前の値のマッピングのすべてのプロパティが更新されます。そうでない場合、この名前の値のマッピングがすでに存在する場合は更新されません。
- 更新 - 既存の値のマッピングを更新します。不足しているものを追加をマークすると、テンプレートに以前存在しなかった値のマッピングが新しい値のマッピングとして追加されます。そうでない場合、テンプレートにすでに存在する値のマッピングのみが更新されます。
- 名前の変更 - 既存の値のマッピングに新しい名前を付けます。
- 削除 - 指定した値のマッピングをテンプレートから削除します。選択したものを除くをマークすると、指定したもの以外のすべての値のマッピングが削除されます。
- すべて削除 - テンプレートからすべての値のマッピングを削除します。すべての値のマッピングを削除することを確認しますチェックボックスがマークされていない場合は、削除の確認を求める新しいポップアップウィンドウが開きます。

値のマッピングの追加/更新操作には、テンプレートから追加およびホストから追加オプションが利用できます。
それぞれテンプレートまたはホストから値のマッピングを選択できます。

必要な変更がすべて完了したら、更新をクリックします。選択したすべてのテンプレートの属性がそれに応じて更新されます。

1 ホストとホストグループ

「ホスト」とは？

Zabbix において、「ホスト」とは、物理または仮想デバイス、アプリケーション、サービス、または監視対象パラメータの論理的に関連するコレクションを指します。

ホストの作成は、Zabbix での最初の監視タスクの 1 つです。たとえば、サーバー「x」のいくつかのパラメータを監視したい場合、まず「Server X」と呼ばれるホストを作成し、その後で監視アイテムを追加することができます。

ホストはホストグループに編成されます。

次の手順に進みます:

- ホストウィザード - 新規または既存のホストを作成・設定するためのガイド付きステップバイステップインターフェース
- ホストの設定 - ホストの作成と設定のための従来のアプローチ

1 ホストウィザード

概要

ホストウィザードは、Zabbix で監視対象（デバイス、アプリケーション、サービスなど）をセットアップするための、ガイド付きのステップバイステップインターフェースです。以下の手順で設定を進めます:

- テンプレートの選択
- ホストの作成または選択
- Zabbix エージェントまたはエージェント 2 のインストール
- ホストインターフェースの追加
- 監視対象または Zabbix ホストへの追加設定の適用（テンプレートで必要な場合）

Zabbix フロントエンドでホストウィザードにアクセスするには、以下の手順を実行します:

- データ収集 > ホスト に移動

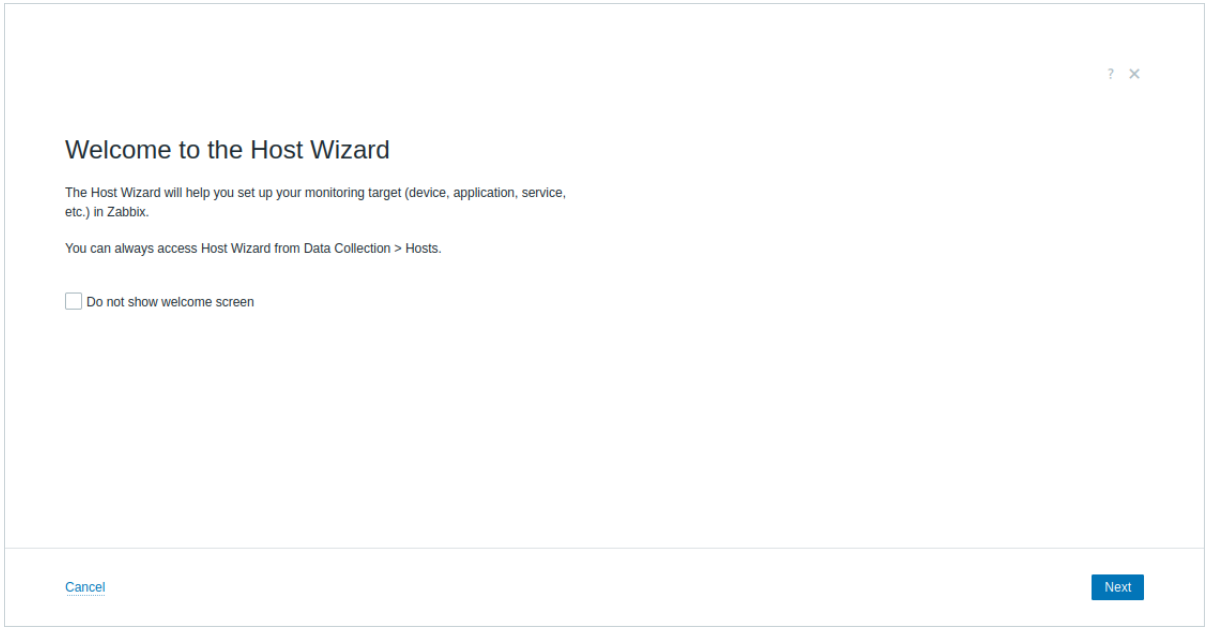
- 画面右上の ホストウィザードをクリック
- ウィザードの指示に従う

また、ホストメニューから既存のホスト名（またはその横の三点アイコン）をクリックして、さまざまなフロントエンドセクションでホストウィザードにアクセスし、既存のホストを編集することもできます。既存のホストを編集する場合、選択したテンプレートで必要な設定がすでに含まれている場合は、一部の手順がスキップされることに注意してください。

または、従来の方法でホストの作成と設定を行うこともできます。

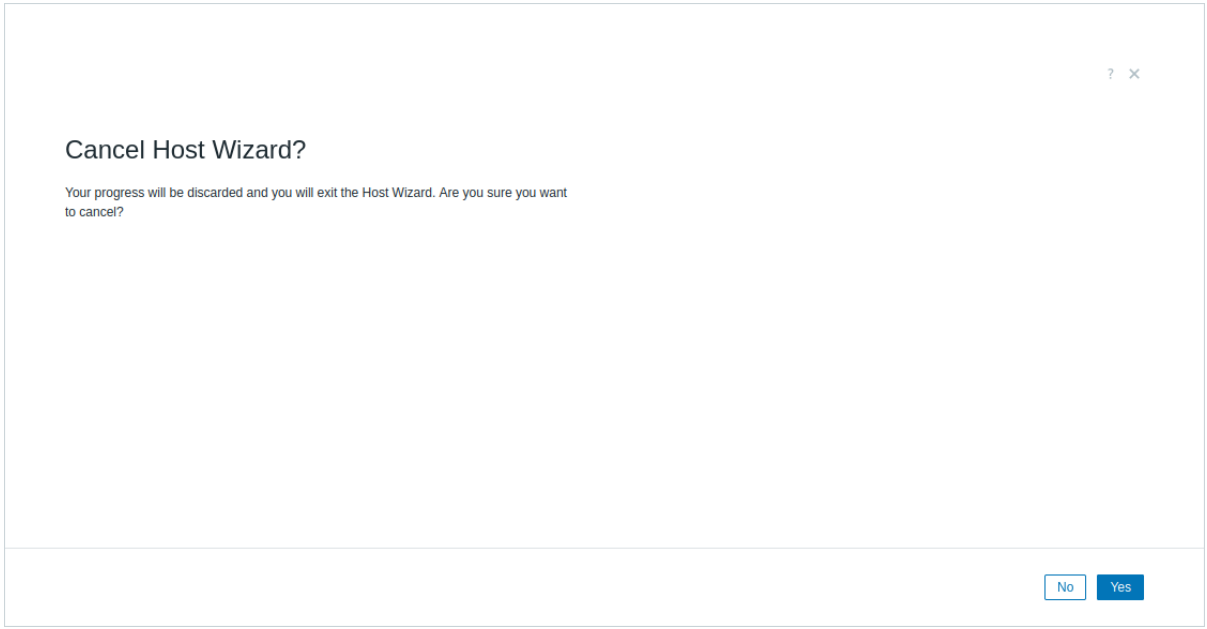
ホストウィザードへようこそ

ホストウィザードを開始すると、ウェルカム画面が表示されます。



今後この画面を表示しない場合は、ウェルカム画面を表示しないチェックボックスをオンにして、次へをクリックします。

次のホストウィザードのステップで設定を開始し、閉じるアイコンをクリックするか ESC キーを押してウィザードを閉じようとすると、確認画面が表示されます。



はいをクリックすると、進行状況を保存せずにホストウィザードを終了します。いいえ、ESC、または閉じるアイコンをクリックすると、最後のステップに戻ります。

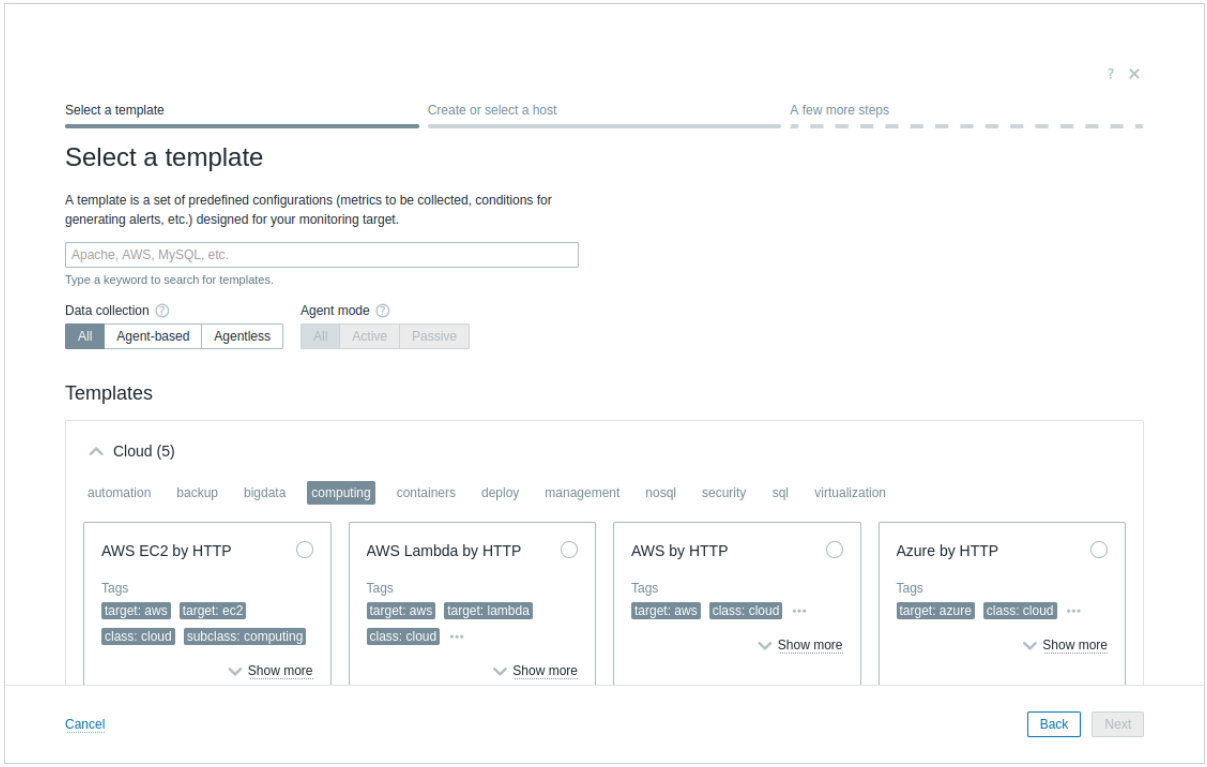
テンプレートの選択

監視対象の設定の最初のステップは、テンプレート（収集するメトリクス、アラート生成条件など、監視対象向けに設計された事前定義済みの設定セット）を選択することです。

Zabbix の標準テンプレートは、オペレーティングシステム、アプリケーション、データベース、ネットワークデバイス、サービスなど、さまざまな監視設定を提供します。

このステップでは、以下のことができます。

- テンプレートのタグに基づいて、クラス（Cloud、Database、Network など）やサブクラス（例：automation、backup）でテンプレートを閲覧できます。
- テンプレート名やテンプレートタグ名・値のキーワードでテンプレートを検索できます。
- データ収集方法（エージェントベースまたはエージェントレス）でテンプレートをフィルタリングできます。エージェントベースのテンプレートには、少なくとも 1 つのZabbix エージェントアイテムが含まれます。
- エージェントモード（アクティブまたはパッシブ）でテンプレートをフィルタリングできます。アクティブテンプレートには少なくとも 1 つの Zabbix エージェント（アクティブ）アイテムが含まれ、パッシブテンプレートには少なくとも 1 つのパッシブアイテムが含まれます。詳細はパッシブおよびアクティブエージェントチェックを参照してください。
- 既存のホストをホストウィザードで設定する場合、すでにホストにリンクされているテンプレートでフィルタリングすることもできます。



Attention:
テンプレートを選択すると、一部のテンプレート *(n)* はホストウィザードと互換性がありません。アップデート方法をご覧ください。カスタムテンプレートはサポートされていません。というメッセージが表示されます。これは、一部の標準テンプレートがまだホストウィザードと互換性がなく、アップグレードが必要であることを示しています。アップグレード手順を参照してください。

選択したテンプレートによって、ホストウィザードの次のステップが決まります。たとえば、データ収集に Zabbix エージェントを使用する *MySQL by Zabbix agent* テンプレートを選択した場合、ウィザードはエージェントのインストール手順を案内します。

ホストの作成または選択

テンプレートは、Zabbix で監視対象を表すホストにリンクする必要があります。リンクされると、ホストは監視に必要なすべてのテンプレートエンティティ（収集するアイテム（メトリック）やトリガー（アラート発生条件）など）を受け取ります。

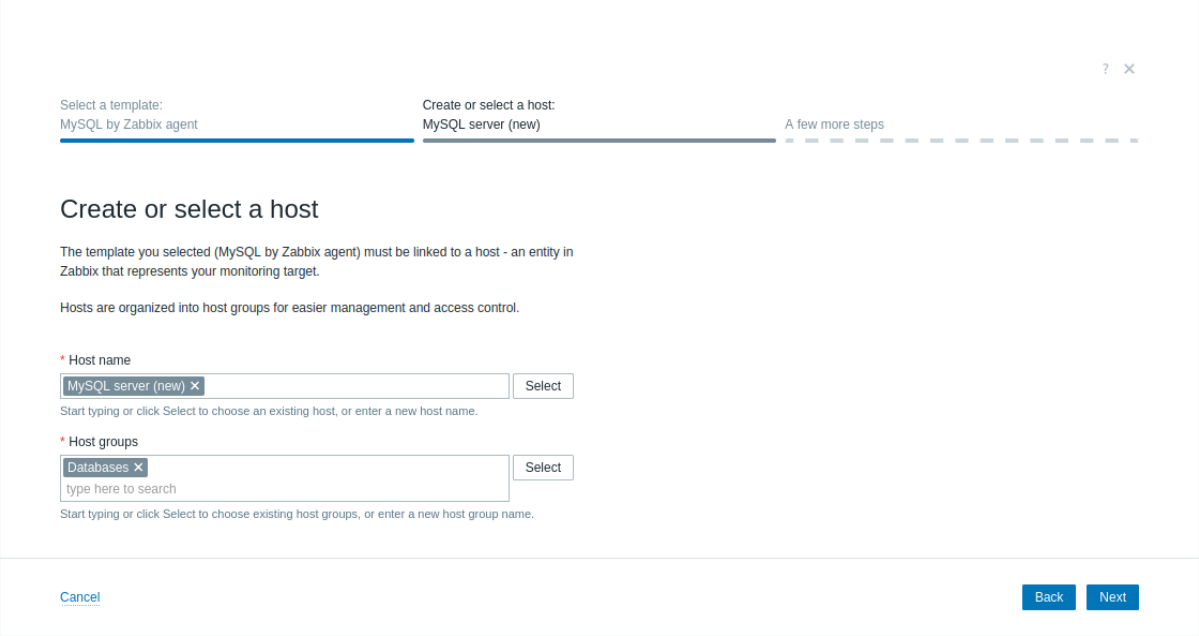
各ホストは、ホストの整理やユーザー権限の割り当てに使用されるホストグループにも、少なくとも 1 つ所属している必要があります。

このステップでは、以下のことができます。

- ホスト名とホストグループ名を入力して、新しいホストとホストグループを作成する
- ホスト名を入力して新しいホストを作成し、既存のホストグループを選択して新しいホストに割り当てる
- 既存のホストを選択する（この場合、現在のグループから削除せずに追加のホストグループに割り当てることも可能）

Note:
既存のホストを選択しても、後続のステップで明示的に指定しない限り、現在の設定が削除または上書きされることはありません。検出されたホストは選択できません。

たとえば、監視したいローカルにインストールされた MySQL サーバーを表す *MySQL server* という新しいホストを作成できます。先ほど選択した *MySQL by Zabbix agent* テンプレートは、このホストにリンクされます。さらに、既存の *Databases* ホストグループ（または新たに *MySQL servers* などを作成）を選択して、他のデータベース関連のホストとまとめて整理し、後で権限管理を簡素化することもできます。



Zabbix エージェントのインストール

エージェントベースのテンプレート（*MySQL by Zabbix agent* など）を使用するには、Zabbix のエージェントまたはエージェント2をインストールする必要があります。これは、監視対象のマシン上でローカルリソースやアプリケーションを積極的に監視するためのプロセスです（テンプレートで必要な場合）。

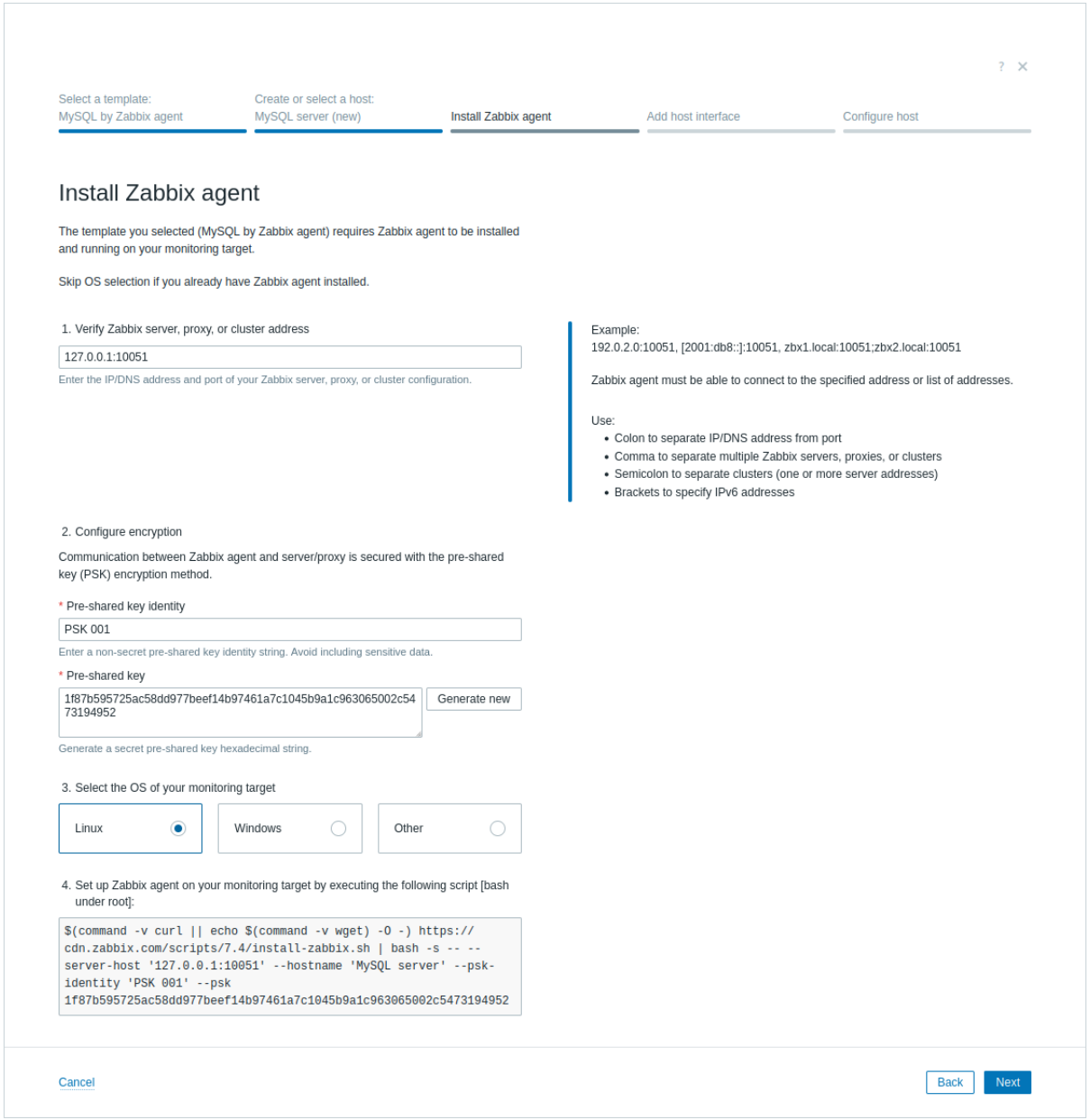
このステップでは:

- Zabbix サーバー、プロキシ、またはクラスタのアドレス（例: 192.0.2.0:10051）を確認します。この値は、Zabbix エージェントのServerおよびServerActiveパラメータの設定に使用されます。
- 新しい一意で秘密ではない事前共有鍵 ID（例: *PSK 001* または *mysql-agent-psk1*）を入力し、事前共有鍵を生成します。これらの値は、Zabbix ホスト（例: *MySQL server*）および Zabbix エージェントの事前共有鍵（PSK）暗号化の設定に使用されます。

Attention:
既存のホストを編集する場合、PSK の設定はホスト上の既存の暗号化設定をすべて上書きします。

- 監視対象をホストしているマシンのオペレーティングシステムを選択します。
- 提供されたスクリプトを実行するか、インストール手順に従って、そのシステムに Zabbix エージェントをインストールします。

例えば、*Pre-shared key identity* に *PSK 001* を設定し、新しい *Pre-shared key* を生成できます。次に、オペレーティングシステムとして *Linux* を選択し、そのシステムで提供されたスクリプトを実行できます。Zabbix エージェントのインストールが完了したら、ホストウィザードに戻ります。



ホストインターフェースの追加

ホストインターフェースは、Zabbix サーバーがネットワーク経由で監視対象に接続する方法を定義します。選択したテンプレートで必要なデータ収集方法に応じて、IP アドレス、DNS 名、ポート、インターフェースタイプ（Agent、SNMP、JMX、または IPMI）などの接続パラメータを定義します。

このステップでは、選択したテンプレートで必要なホストインターフェースの詳細を入力します。

たとえば、Zabbix サーバー、Zabbix エージェント、および MySQL サーバーがすべて同じマシン上でローカルに実行されている場合は、デフォルトのエージェントアドレス（127.0.0.1）とポート（10050）を使用できます。

ホストの設定

一部のテンプレートでは、ホストを作成する前に追加の設定が必要です。これには以下が含まれる場合があります:

- 監視対象の手動設定 (例: 特定サービスの有効化、サービスユーザーの作成、権限の付与):
- ホストマクロの定義—アイテムの動作、接続設定、認証情報、その他のパラメータを制御する変数:

必要な設定手順がすべて完了したら、作成をクリックしてホストを Zabbix に追加します (または更新をクリックして既存ホストの設定を更新します):

設定完了

この時点で、Zabbix はすでにターゲットの監視を開始しています。

新しいホストの場合は、完了をクリックして最新データ セクションに移動し、ホストの最新データを表示します。既存のホストの場合は、完了をクリックしてホストウィザードを閉じ、ウィザードを開いた画面に戻ります。

概要

Zabbix フロントエンドでホストを設定するには、以下の手順を実行します。

- データ収集 > ホストまたは 監視 > ホストに移動します
- 画面右上の ホストの作成をクリックします（または既存のホストを編集する場合はホスト名をクリックします）
- フォームにホストのパラメータを入力します

既存のホストの設定フォームで クローンボタンを使用して、新しいホストを作成することもできます。このホストは、リンクされたテンプレート、テンプレートからのエンティティ（アイテム、トリガーなど）、および既存のホストに直接関連付けられているエンティティを含む、既存のホストのすべてのプロパティを持ちます。

ホストをクローンすると、すべてのテンプレートエンティティはテンプレート上の元の状態のまま保持されることに注意してください。既存のホストレベルで行われたそれらのエンティティへの変更（アイテムの間隔の変更、正規表現の修正、低レベルディスカバリールールへのプロトタイプの追加など）は、新しいホストにはクローンされません。代わりにテンプレート上の状態になります。

また、ホストウィザードを使用して、ガイド付きのステップバイステップインターフェースでホストを設定することもできます。

設定

ホストタブには、一般的なホスト属性が含まれています。

すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
ホスト名	一意のホスト名を入力します。英数字、スペース、ドット、ダッシュ、アンダースコアが使用できます。ただし、先頭および末尾のスペースは使用できません。 注意: 設定しているホストで Zabbix エージェントが動作している場合、エージェントの設定ファイルパラメータ *Hostname* は、ここで入力したホスト名と同じ値でなければなりません。このパラメータの名前は、アクティブチェックの処理に必要です。
表示名	ホストの一意の表示名を入力します。この名前を設定すると、リストやマップなどで技術的なホスト名の代わりに表示されます。この属性は UTF-8 をサポートしています。
テンプレート	テンプレートをホストにリンクします。すべてのエンティティ（アイテム、トリガー、その他）はテンプレートから継承されます。 新しいテンプレートをリンクするには、テキスト入力フィールドにテンプレート名を入力し始めます。一致するテンプレートのリストが表示されるので、スクロールして選択します。あるいは、フィールドの横にある選択をクリックして、ポップアップウィンドウのリストからテンプレートを選択することもできます。すべての選択したテンプレートは、ホスト設定フォームを保存または更新するとホストにリンクされます。 テンプレートのリンクを解除するには、リンクされたテンプレートブロックの 2 つのオプションのいずれかを使用します。 リンク解除 - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガー、その他）は保持します。 リンク解除およびクリア - テンプレートのリンクを解除し、そのエンティティ（アイテム、トリガー、その他）をすべて削除します。 リストされたテンプレート名はクリック可能なリンクで、テンプレート設定フォームに移動します。
ホストグループ	ホストが属するホストグループを選択します。ホストは少なくとも 1 つのホストグループに属している必要があります。存在しないグループ名を追加することで、新しいグループを作成してホストにリンクできます。

パラメータ		説明
インターフェース		ホストには複数のホストインターフェースタイプがサポートされています：エージェント、*SNMP*、*JMX*、*IPMI*。 デフォルトではインターフェースは定義されていません。新しいインターフェースを追加するには、インターフェースブロックの追加をクリックし、インターフェースタイプを選択して*IP/DNS*、接続先、ポート情報を入力します。 注意*:* アイテムで使用されているインターフェースは削除できず、削除リンクはグレーアウトされます。 SNMP インターフェースの「IP」または「DNS」は、SNMP トラップにも使用されます。 マッチング時には、ホストインターフェースで選択された「IP」または「DNS」のみが使用されます。 SNMP インターフェース（v1、v2、v3）の追加設定については、SNMP 監視の設定を参照してください。
	IP アドレス	ホストの IP アドレス（オプション）。
	DNS 名	ホストの DNS 名（オプション）。
	接続先	それぞれのボタンをクリックすると、Zabbix サーバーがエージェントからデータを取得する方法を指定します： **IP** - ホストの IP アドレスに接続（推奨） **DNS** - ホストの DNS 名に接続
	ポート	TCP/UDP ポート番号。デフォルト値は、Zabbix エージェントは 10050、SNMP エージェントは 161、JMX は 12345、IPMI は 623 です。
	デフォルト	ラジオボタンをチェックしてデフォルトインターフェースを設定します。
説明		ホストの説明を入力します。
監視方式プロキシ		ホストがどの方法で監視されるかを選択します： サーバー - ホストは Zabbix サーバーによって監視されます。 プロキシ - ホストはスタンドアロンプロキシによって監視されます。 プロキシグループ - ホストはプロキシグループによって監視されます。 割り当てられたプロキシ名が表示されます（Zabbix サーバーが選択したプロキシグループから割り当てた場合のみ）。 このフィールドは、ホストがプロキシグループによって監視されている場合のみ表示されます。
有効		チェックボックスがオンの場合、ホストは有効で監視可能です。 チェックボックスがオフの場合、ホストは無効で監視されません： Zabbix サーバー/プロキシによって開始されるパッシブデータリクエスト（例：Zabbix エージェント、SNMP エージェント、シンプルチェック）の場合、設定の同期後に監視が無効になります。ホストにリンクされたトリガーやアクションも、設定が再読み込みされた後にのみ無効になります。 Zabbix エージェントのアクティブチェックの場合、ホストが無効になったことを Zabbix エージェントが受信するまでの時間枠（約 5 秒）内で監視が停止します。この短い間、ホストはアクティブチェックのデータをローカルで収集し、サーバー/プロキシに送信しようとしますが、ホストが無効としてマークされているため、サーバー/プロキシはデータを拒否します。 ホストを無効にすると、そのアイテムは履歴キャッシュから即座に削除されます（ただし、最後の値はログ用に保持されます）。

IPMI タブには、IPMI 管理属性が含まれています。

パラメータ	説明
認証アルゴリズム	認証アルゴリズムを選択します。
特権レベル	特権レベルを選択します。
ユーザー名	認証用のユーザー名。ユーザーマクロが使用できます。
パスワード	認証用のパスワード。ユーザーマクロが使用できます。

タグタブでは、ホストレベルのタグを定義できます。このホストのすべての問題は、ここで入力した値でタグ付けされます。

タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。

マクロタブでは、ホストレベルのユーザーマクロを名前と値のペアとして定義できます。マクロ値はプレーンテキスト、シークレットテキスト、Vault シークレットとして保持できます。説明の追加もサポートされています。

継承およびホストマクロオプションを選択すると、テンプレートレベルおよびグローバルユーザーマクロもここで表示できます。ここでは、ホストに定義されたすべてのユーザーマクロが、解決される値とその由来とともに表示されます。

利便性のために、該当するテンプレートやグローバルマクロ設定へのリンクが提供されています。また、ホストレベルでテンプレート/グローバルマクロを編集して、ホスト上にマクロのコピーを作成することも可能です。

インベントリタブでは、ホストのインベントリ情報を手動で入力できます。また、自動インベントリの自動入力を有効にするか、このホストのインベントリ入力を無効にすることもできます。

インベントリが有効（手動または自動）な場合、タブ名の横に緑色のドットが表示されます。

暗号化

暗号化タブでは、ホストと暗号化された接続をするように要求できます。

パラメーター	説明
ホストへの接続	Zabbix サーバーまたはプロキシがホスト上の Zabbix エージェントに接続する方法: 暗号化なし (デフォルト)、PSK(事前共有キー) または証明書を使用。
ホストからの接続	ホスト (Zabbix エージェントおよび Zabbix sender) から許可される接続の種類を選択します。複数の接続タイプを同時に選択できます (テストや他の接続タイプへの切り替えに便利です)。デフォルトは” 暗号化なし” です。
発行者	許可された証明書の発行者。証明書は最初に CA (証明機関) で検証されます。証明書が有効で、CA によって署名されている場合は、発行者フィールドを使用して、許可される CA をさらに制限できます。このフィールドは、Zabbix インストールで複数の CA からの証明書を使用する場合に使用することを目的としています。このフィールドが空の場合、どの CA でも受け入れられます。
サブジェクト	許可された証明書の件名。証明書は最初に CA で検証されます。証明書が有効で、CA によって署名されている場合、サブジェクトフィールドを使用してサブジェクト文字列の値を 1 つだけ許可できます。このフィールドが空の場合、構成された CA によって署名された有効な証明書はすべて受け入れられます。
PSK アイデンティティ	事前共有キー ID 文字列。 PSK ID には機密情報を入れないでください。これは、受信者にどの PSK を使用するかを通知するために、ネットワーク経由で暗号化されずに送信されます。
PSK	事前共有キー (16 進文字列)。最大長:Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 桁の 16 進数 (256 バイトの PSK)、Zabbix が mbed TLS(PolarSSL) ライブラリを使用する場合は 64 桁の 16 進数 (32 バイトの PSK)。例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

値のマッピング

値のマッピングタブでは、値のマッピングを使用してアイテムの値を分かりやすい表現で表示できるよう設定できます。

3 ホストグループの設定

概要

ホストグループは、ホストを論理的にグループ化し、ユーザー権限を割り当てるために使用されます。

各ホストには、少なくとも 1 つのホストグループを割り当てる必要があります。1 つのホストは複数のホストグループに属することができ、各ホストグループには複数のホストを含めることができます。

Zabbix では、すべての権限はグループ（ユーザーグループ、ホストグループ、テンプレートグループ）に基づいていることに注意してください。したがって、1 人のユーザーが 1 つのホストへのアクセスを必要とする場合でも、そのホストを含むホストグループへのアクセス権を持つユーザーグループにユーザーを追加することで権限が付与されます。

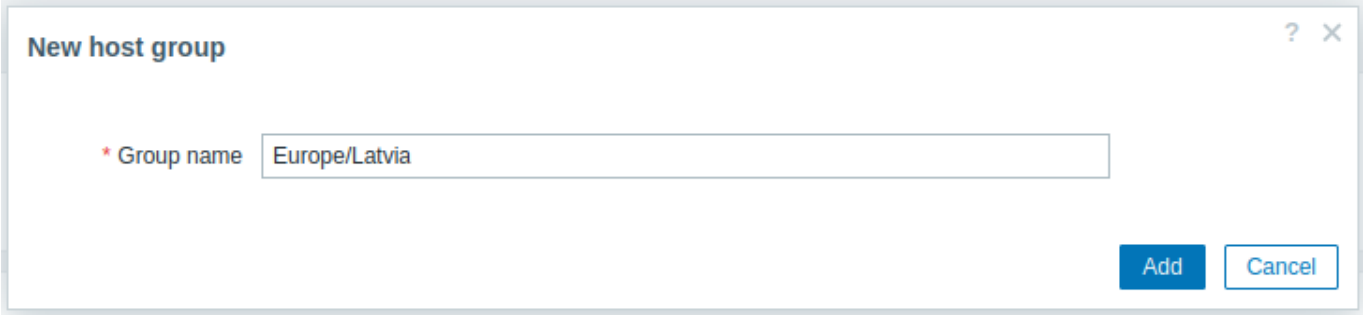
設定

> **Attention:**
> ホストグループの作成は、スーパーユーザーのみが可能です。

Zabbix フロントエンドでホストグループを作成する方法は 2 つあります。

方法 **1**:

- データ収集 → ホストグループに移動します
- 画面右上の ホストグループの作成をクリックします
- フォームにグループ名を入力します



方法 **2**: ホストを設定する際に、ホストグループ入力欄に存在しないグループ名を入力します。

ホストグループが作成されると、データ収集 → ホストグループのリストでグループ名をクリックして、グループ名の編集、グループのクローン作成、またはグループの削除ができます。

ホストグループを削除しても、論理的なグループのみが削除され、グループ内のホストは削除されません。既存のホストにとって唯一のグループであるホストグループは削除できません。

ホストサブグループの作成

ホストサブグループ（またはネストされたホストグループ）は、それを含む親ホストグループの子です。

サブグループは、グループ名入力フィールドでスラッシュ「/」を使用して親グループとの関係を示すことで作成されます。例えば:

- Europe/Latvia と入力すると、親グループ Europe のサブグループ Europe/Latvia が作成されます。
- Europe/Latvia/Riga/Zabbix servers と入力すると、ネストされた親グループ Europe、Europe/Latvia、Europe/Latvia/Riga 内にそれぞれのサブグループが作成されます。

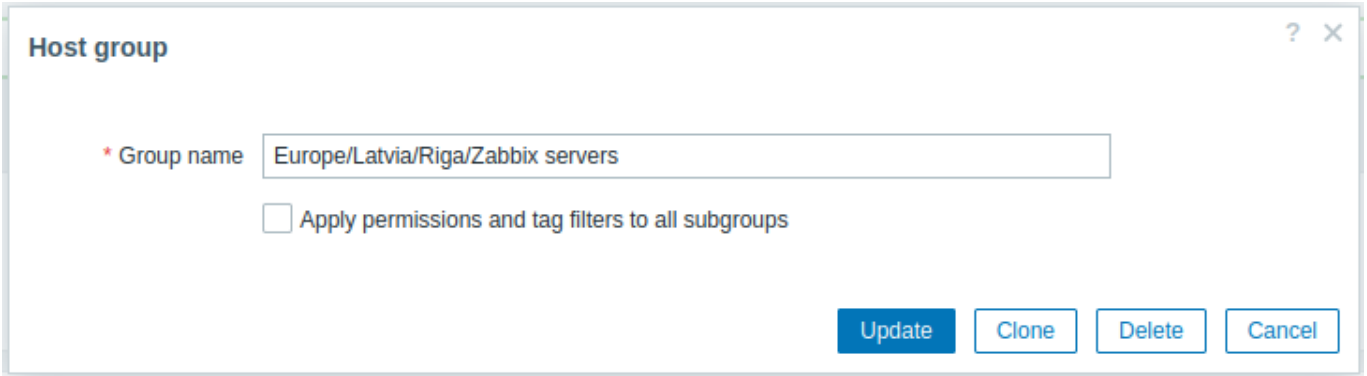
サブグループを作成する際、先頭または末尾のスラッシュや、連続した複数のスラッシュの使用はできません。/のエスケープはサポートされていません。

サブグループを作成する前に親ホストグループを作成する必要はありません。サブグループ（例：Europe/Latvia）から作成を開始するか、親ホストグループ（例：Europe）から作成を開始するかを選択できます。サブグループから作成を開始した場合、親ホストグループは自動的には作成されません。

ホストグループへの権限

- 既存の親ホストグループにサブグループを作成する場合 (例: Europe が既に存在する場合に Europe/Latvia を作成する場合)、ユーザーグループのサブグループへの権限は親から継承されます。
- 既存のサブグループに親ホストグループを作成する場合 (例: Europe/Latvia が既に存在する場合に Europe を作成する場合)、親グループには権限が設定されません。

任意のホストグループを編集する際に、追加オプション「すべてのサブグループに権限とタグフィルターを適用」を設定することもできます。



このチェックボックスをマークして更新をクリックすると、編集中のホストグループのすべての現在および将来のサブグループに同じレベルの権限とタグフィルターが適用されます。

したがって、編集中のホストグループのサブグループに異なる権限がユーザーグループに付与されている場合でも、チェックボックスをマークすると、すべての現在および将来のサブグループに、編集中のグループと同じユーザー権限およびタグベースの権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きすることに注意してください。このオプションによる変更は手動でのみ元に戻すことができます。

4 インベントリ

概要

Zabbix でネットワークデバイスのインベントリを管理することができます。

Zabbix フロントエンドには特別なインベントリメニューがあります。しかし、初期状態ではそこにデータは表示されず、データを入力する場所でもありません。インベントリデータの構築は、ホストを設定するときに手動で行うか、自動配置オプションを使用して自動的に行われます。

インベントリの構築

手動モード

ホストを設定するときに、インベントリタブで、デバイスのタイプ、シリアル番号、場所、担当者、URL などの詳細(インベントリ情報に入力されるデータ) を入力できます。

ホストインベントリ情報に URL が含まれており、その URL が'http' または'https' で始まる場合、インベントリセクションにクリック可能なリンクが表示されます。

自動モード

ホストのインベントリも自動的に入力できます。これを機能させるには、インベントリタブでホストのインベントリモードを設定するときに自動に設定する必要があります。

次に、ホストアイテムを設定して、ホストインベントリフィールドに値を付けるようにアイテム設定のそれぞれの属性を持つ宛先フィールドを示すことができます (呼び出されたアイテムはホストインベントリフィールドに入力されます)。

インベントリデータの自動収集に特に有効なアイテムを収集することができます。

- system.hw.chassis[full|type|vendor|model|serial] - デフォルトは [full] で、ルート権限が必要
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - デフォルトは [all,full]
- system.hw.devices[pci|usb] - デフォルトは [pci]
- system.hw.macaddr[interface,short|full] - デフォルトは [all,full]、インターフェースは正規表現
- system.sw.arch
- system.sw.os[name|short|full] - デフォルトは [name]
- system.sw.packages[package,manager,short|full] - デフォルトは [all,all,full]

インベントリモードの選択

インベントリモードは、ホスト設定フォームで選択することができます。

新しいホストのデフォルトのインベントリモードは、管理 → 一般設定 → その他のデフォルトのホストインベントリモード設定に基づいて選択されます。

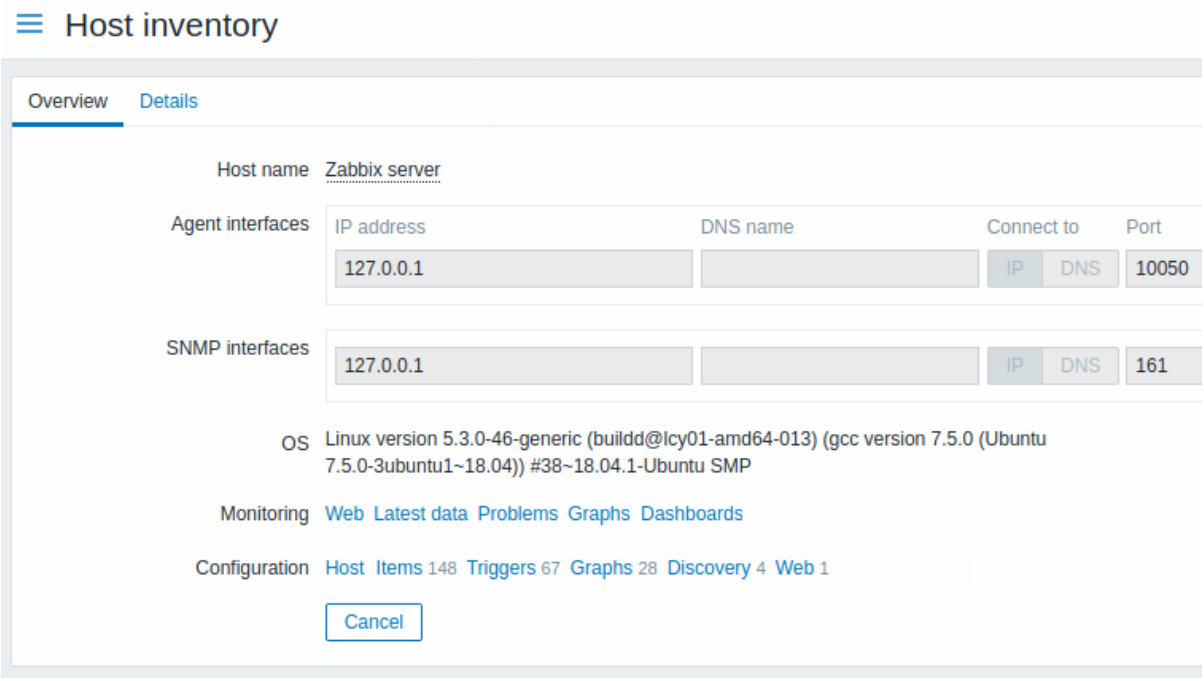
ネットワークディスカバリまたは自動登録アクションによって追加されたホストの場合、ホストインベントリモードの設定操作で手動モードまたは自動モードを選択できます。この操作はデフォルトのホストインベントリモード設定を上書きします。

インベントリ概要

既存のすべてのインベントリデータの詳細は、インベントリメニューに表示されます。

インベントリ → 概要で、インベントリの様々なフィールドによるホスト数を取得できます。

インベントリ → ホストで、インベントリ情報を持つすべてのホストを見ることができます。ホスト名をクリックすると、インベントリの詳細が表示されます。



概要タブは以下を表示します。

パラメーター	説明
ホスト名	ホストの名前。 名前をクリックすると、ホストに定義されているスクリプトを含むメニューが開きます。 ホストがメンテナンス中の場合、ホスト名はオレンジ色のアイコンとともに表示されます。
表示名	ホストの表示名 (定義されている場合)。
ホスト *(エージェント、SNMP、JMX、IPMI) * インターフェース	ホストに設定されたインターフェースの詳細を提供します。
OS	ホストのオペレーティングシステムのインベントリフィールド (定義されている場合)。
ハードウェア	ホストのハードウェアのインベントリフィールド (定義されている場合)。
ソフトウェア	ホストのソフトウェアのインベントリフィールド (定義されている場合)。
説明	ホストの説明。
監視データ	このホストのデータを含む監視セクションへのリンク: *Web*、最新データ、障害、グラフ、ダッシュボード。
設定	このホストの設定セクションへのリンク: ホスト、アイテム、トリガー、グラフ、ディスカバリ、*Web* 設定されたエンティティの数は、各リンクの後に表示されます。

詳細タブには、入力されている (空ではない) すべてのインベントリフィールドが表示されます。

インベントリマクロ

通知用にホストインベントリマクロ {INVENTORY.} が用意されています。例えば、

”{INVENTORY.LOCATION1} のサーバーに障害が発生しました。責任者は {INVENTORY.CONTACT1}、電話番号は {INVENTORY.POC.PRIMARY.PHONE.A1}。”

詳細は、サポートしているマクロページをご覧ください。

5 一括更新

概要

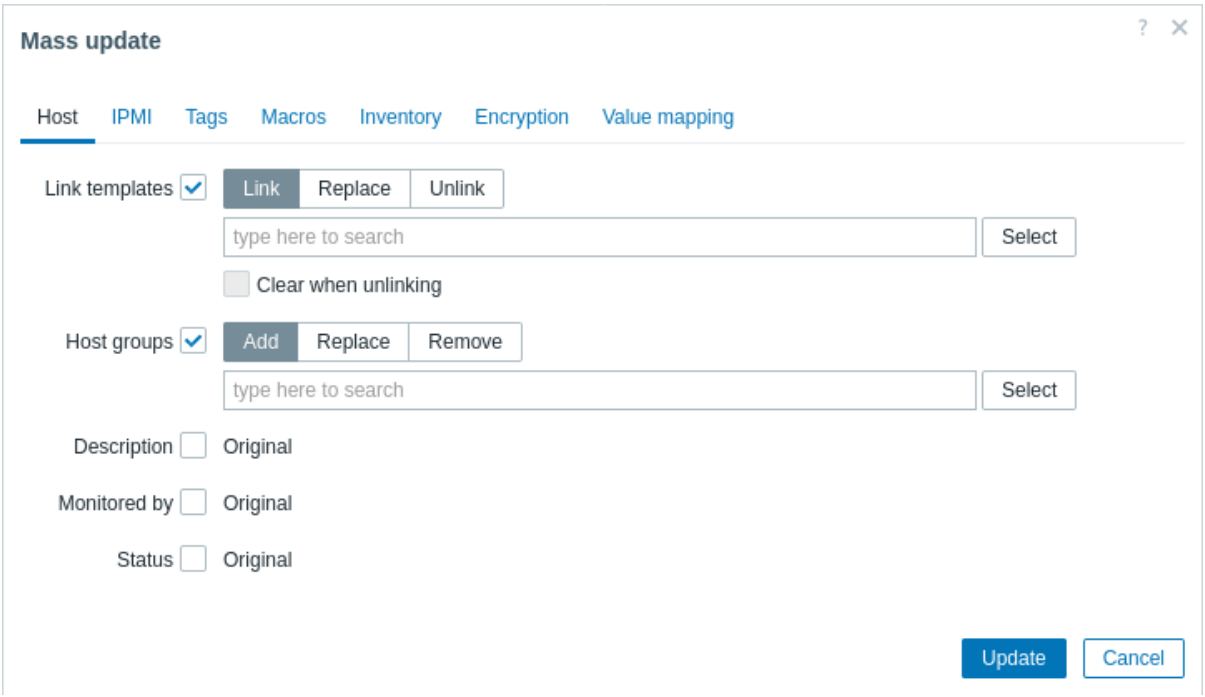
一度に複数のホストの属性を変更したい場合があります。個々のホストを開いて編集する代わりに一括更新機能を使います。

一括アップデート機能の使用

いくつかのホストを一括でアップデートするには、以下のようにします。

- ホストリストの更新したいホストの前にあるチェックボックスにチェックします。
- リストの下にある一括更新をクリックします。

- 必要な属性のタブ (ホスト、*IPMI*、タグ、マクロ、インベントリ、暗号化、値のマッピング) に移動します。
- 更新する属性のチェックボックスをチェックし、新しい値を入力します。



それぞれのボタンを選択すると、次のオプションに基づいてテンプレートのリンクが更新されます。

- リンク - 追加でリンクするテンプレートを指定します。
- 置換 - リンクするテンプレートを指定し、それ以前にホストとリンクしていたテンプレートはリンクを削除します。
- リンクを削除 - リンクを削除するテンプレートを指定します。

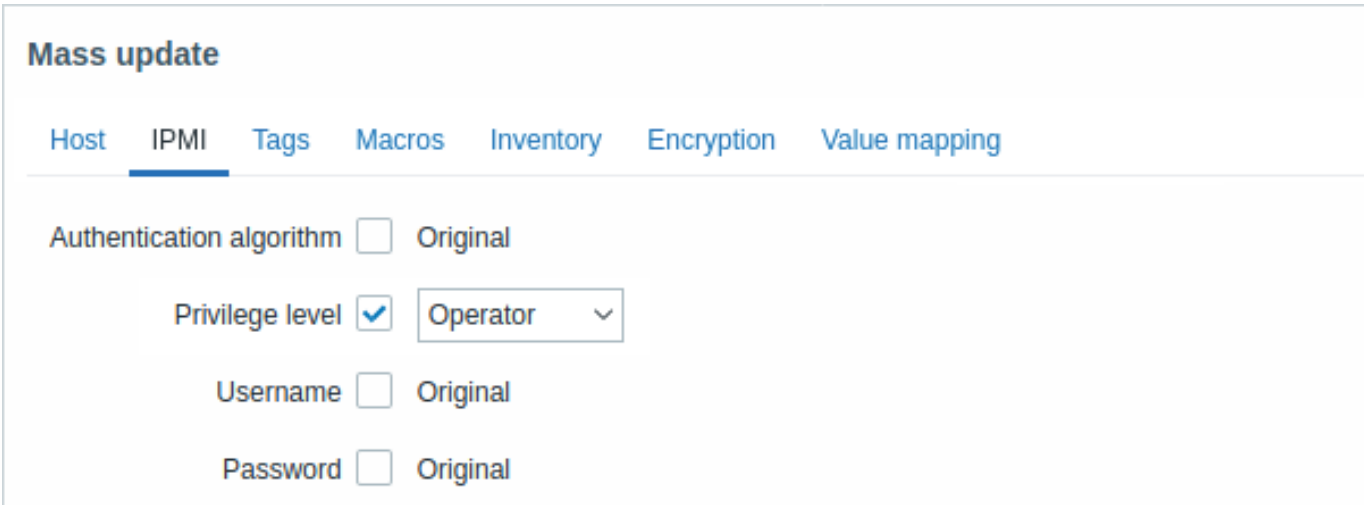
リンク/リンクの削除を行うテンプレートを指定するには、検索文字列を入力にテンプレート名を入力し、一致するテンプレートがドロップダウンが表示されるまで待ちます。下にスクロールして必要なテンプレートを選択します。

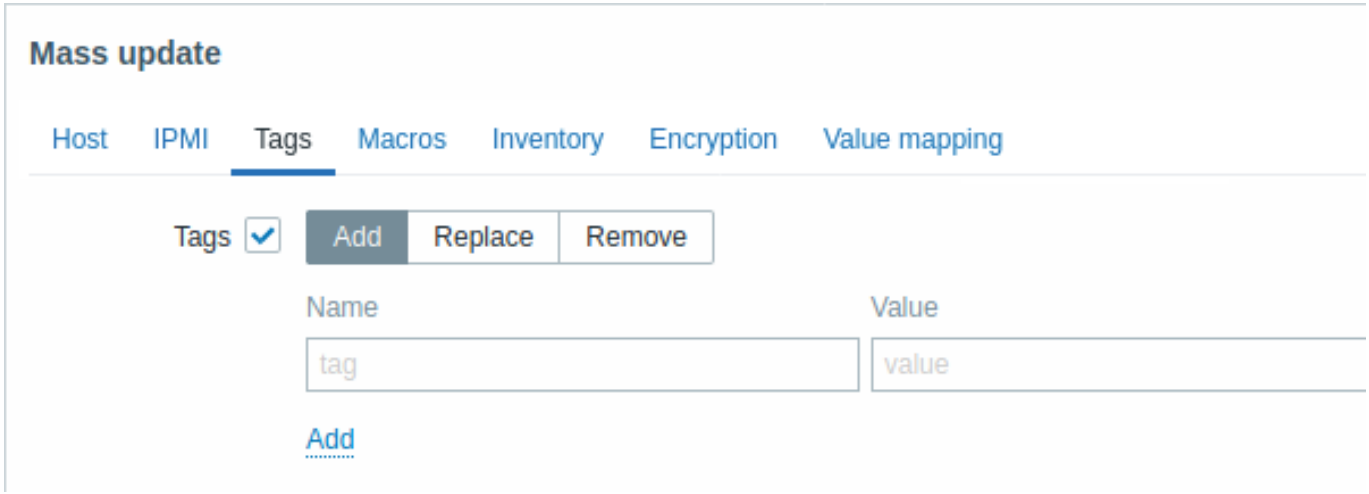
リンク削除時にクリアをチェックすると、以前にリンクされたテンプレートを解除するだけでなく、テンプレートから継承されたすべての要素 (アイテム、トリガーなど) も削除されます。

それぞれのボタンを選択すると、次のオプションに基づいてホストグループが更新されます。

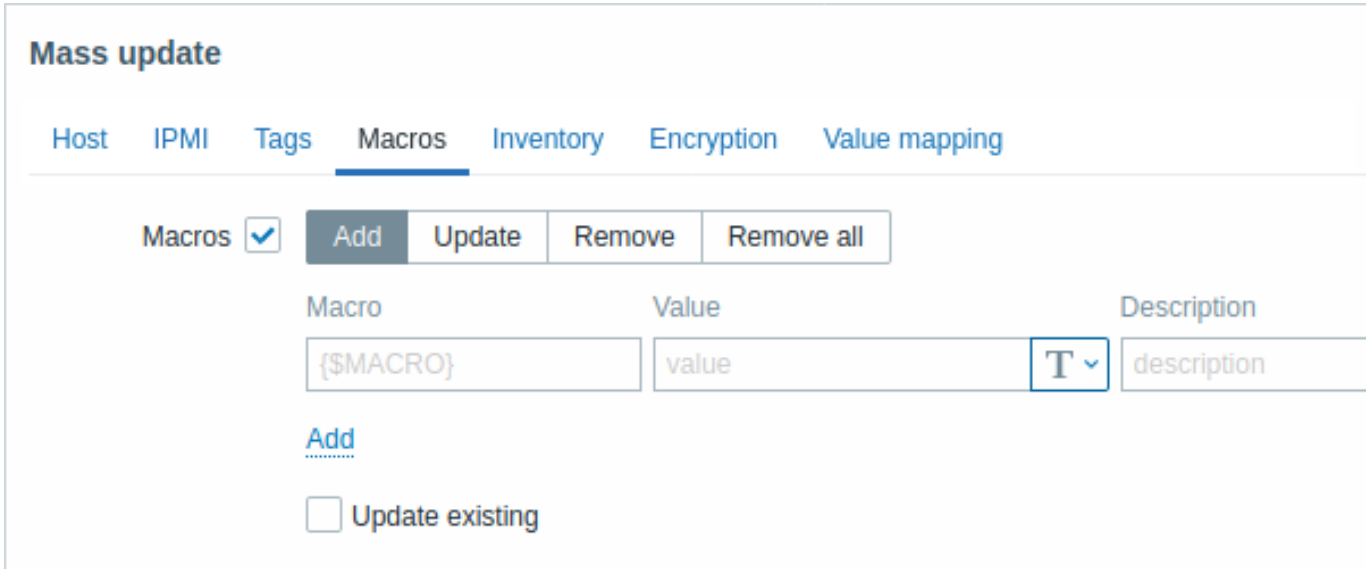
- 追加 - 既存のホストグループから追加するホストグループを指定したり、新しいホストグループを入力することができます。
- 置換 - 既存のホストグループからホストを削除し、指定されたホストグループ (既存または新規のホストグループ) と置き換えます。
- 削除 - 特定のホストグループをホストから削除します。

これらのフィールドはオートコンプリートで、入力を開始すると、一致するホストグループのドロップダウンが表示されます。ホストグループが新しい場合、それはドロップダウンに表示され、文字列の後に *(新規)* が表示されます。選択するには下にスクロールするだけです。



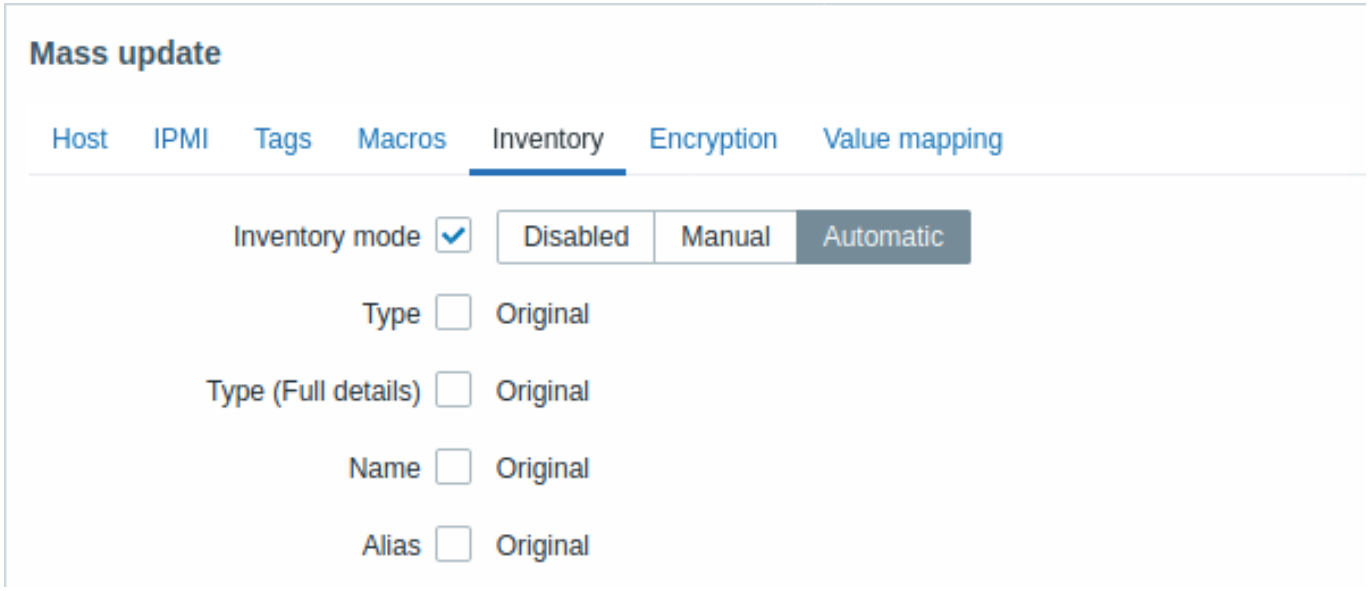


ユーザーマクロ、{INVENTORY. *} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロは、タグでサポートされています。同じ名前で異なる値を持つタグは、' 重複' とはみなされず、同じホストに追加されることに注意してください。

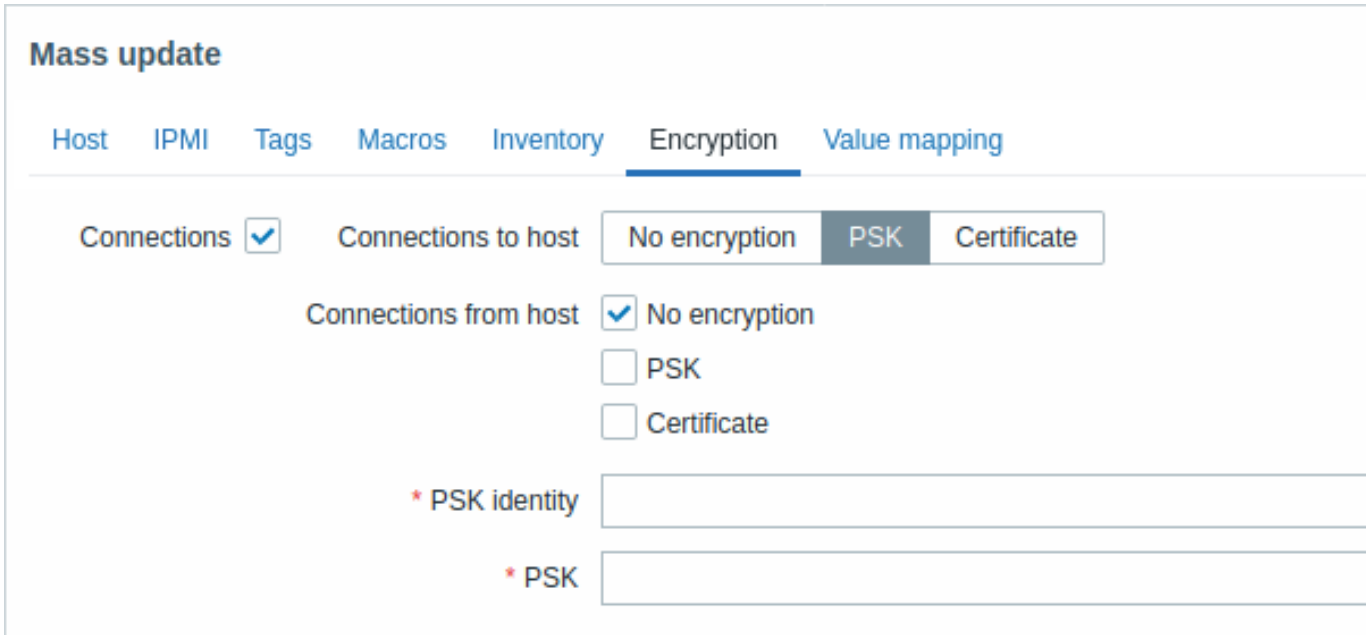


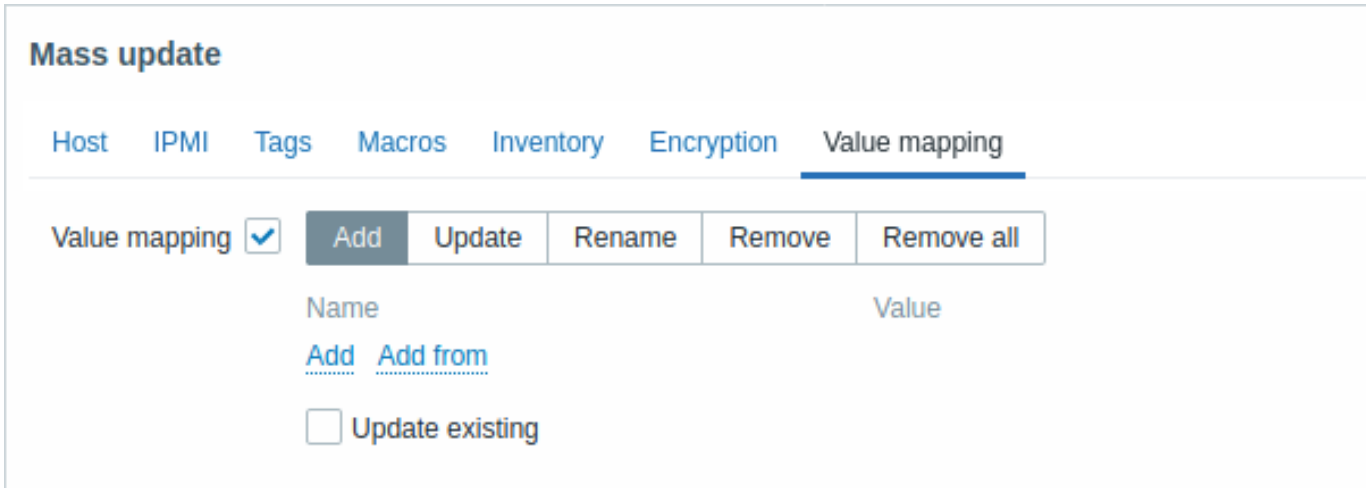
それぞれのボタンを選択すると、次のオプションに基づいてマクロが更新されます。

- 追加 - ホストに追加するユーザマクロを指定することができます。既存の設定を上書チェックボックスにチェックを入れると、指定したマクロの値、タイプ、および説明が更新されます。チェックをはずすと、その名前のマクロがすでにホスト上に存在する場合は、更新されません。
- 更新 - このリストで指定されたマクロの値、タイプ、説明を置き換えます。未設定時に追加チェックボックスにチェックを入れると、ホスト上に存在しなかったマクロを新しいマクロとして追加します。チェックを外すと、既にホスト上に存在するマクロのみが更新されます。
- 削除 - 指定されたマクロをホストから削除します。選択したものを除くチェックボックスにチェックを入れると、リストで指定された以外のすべてのマクロが削除されます。チェックを外すと、リストで指定されたマクロだけが削除されます。
- 全て削除 - ホストからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認チェックボックスのチェックされていない場合、新しいポップアップウィンドウが開き、すべてのマクロの削除を確認します。



インベントリフィールドを一括更新するには、インベントリモードを' マニュアル' または' 自動' に設定する必要があります。





それぞれのボタンを選択すると、次のオプションに基づいて値のマッピングが更新されます。

- 追加 - ホストに値のマッピングを追加します。既存の設定を上書チェックボックスのチェックすると、この名前の値のマッピングのすべてのプロパティが更新されます。それ以外の場合、その名前の値のマッピングが既に存在する場合は、更新されません。
- 更新 - 既存の値のマッピングを更新します。未設定時に追加チェックボックスのチェックすると、ホスト上に以前は存在しなかった値のマッピングが新しい値のマッピングとして追加されます。それ以外の場合は、ホスト上に既に存在する値のマッピ

ングのみが更新されます。
- 名前変更 - 既存の値のマッピングに新しい名前を付けます。
- 削除 - 指定された値のマッピングをホストから削除します。もし選択したものを除くチェックボックスのチェックすると、指定したものを除くすべての値のマッピングが削除されます。
- 全て削除 - ホストからすべての値のマッピングを削除します。もし、すべての値のマッピングを削除することを確認チェックボックスのチェックされていない場合、新しいポップアップウィンドウが開き、削除を確認します。

必要な変更をすべて終えたら、更新をクリックします。選択したすべてのホストの属性は、それに応じて更新されます。

2 アイテム

概要

アイテムは個別の指標です。

アイテムはデータ収集に使用されます。ホストを設定したら、実際のデータを取得するためのアイテムを追加する必要があります。多くのアイテムをすばやく追加する方法は、事前定義されたテンプレートの 1 つをホストにアタッチすることです。ただし、システムのパフォーマンスを最適化するには、必要なだけのアイテムと頻度の監視を行うようにテンプレートを微調整する必要があります。

ホストからどのようなデータを収集するかを指定するには、アイテムキーを使用します。たとえば、キー名 **system.cpu.load** のアイテムはプロセッサ負荷データを収集し、キー名 **net.if.in** のアイテムは受信トラフィック情報を収集します。

キー名の後の角括弧内に追加のパラメーターを指定できます。たとえば、system.cpu.load**[avg5]** は過去 5 分間のプロセッサ負荷平均を返し、net.if.in**[eth0]** はインターフェース”eth0” の受信トラフィックを表示します。

> **Note:**
> サポートされているすべてのアイテムタイプとアイテムキーについては、アイテムタイプの個々のセクションを参照してください。

アイテムの作成と設定に進んでください。

1 アイテムの作成

概要

Zabbix フロントエンドでアイテムを作成するには、以下の手順を実行します。

- データ収集 > ホストに移動します
- ホストの行でアイテムをクリックします
- 画面右上のアイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します

既存のアイテムを開き、クローンボタンを押して別名で保存することで、アイテムを作成することもできます。

設定

アイテムタブには、一般的なアイテム属性が含まれています。

すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	アイテム名。 ユーザーマクロがサポートされています。
タイプ	アイテムタイプ。個別のitem typeセクションを参照してください。
キー	アイテムキー（最大 2048 文字）。 サポートされているitem keysは、個別のアイテムタイプセクションで確認できます。 キーは 1 つのホスト内で一意でなければなりません。 キータイプが「Zabbix エージェント」、「Zabbix エージェント（アクティブ）」または「シンプルチェック」の場合、キー値は Zabbix エージェントまたは Zabbix サーバーでサポートされている必要があります。 また、正しいkey formatも参照してください。

パラメータ	説明
情報の型	変換後（必要な場合）にデータベースに保存されるデータの型。 数値（符号なし） - 64 ビット符号なし整数。 整数アイテムに浮動小数点値が受信された場合、小数点以下は切り捨てられます。 数値（浮動小数点） - 64 ビット浮動小数点数。 この型は約 15 桁の精度と、約-1.79E+308 から 1.79E+308 までの範囲を許容します（PostgreSQL 11 以前のバージョンは例外）。 科学的記数法での値の受信もサポートされています。例：1.23E+7、1e308、1.1E-4。 文字列 - 短いテキストデータ。 ログ - ログ関連プロパティ（タイムスタンプ、ソース、重大度、logeventid）をオプションで持つ長いテキストデータ。 テキスト - 長いテキストデータ。テキストデータの制限も参照してください。 バイナリ - バイナリ値（従属アイテムのみサポート）。バイナリ値は最新データでは静的な「binary value」文字列として解決され、{ITEM.VALUE}、{ITEM.LASTVALUE} および式マクロは UNKNOWN に解決されます。 特定のフォーマットでのみデータを返すアイテムキーの場合、対応する情報の型が自動的に選択されます。
ホストインターフェース	ホストインターフェースを選択します。このフィールドはホストレベルでアイテムを編集する場合に利用できます。
単位	単位記号が設定されている場合、Zabbix は受信したアイテム値に後処理を適用し、指定された単位の接尾辞を付けて表示します。 特別なフォーマットを持つサポートされている単位記号（受信値 → 表示値の例）： **B** - バイト（1024 → 1 KB） **Bps** - バイト/秒（1024 → 1 KBps） **s** - 秒、最大 3 つの最大非ゼロ時間単位で表示（881764 → 10d 4h 56m） **uptime** - 経過時間を hh:mm:ss または N 日, hh:mm:ss で表示（881764 → 10 days, 04:56:04） **unixtime** - Unix タイムスタンプを yyyy.mm.dd hh:mm:ss でフォーマット（881764 → 1970-01-11 04:56:04 AM）；正しくフォーマットするには受信値が数値（符号なし）である必要があります。 その他の単位（Hz、W など）の場合、受信値が 1000 を超えると 1000 で割られ、対応する接頭辞が付いて表示されます（5000 → 5 KHz、881764 → 881.76 KW）。 単位記号の前に**!** を付けると、単位変換と接頭辞付与が無効になります（1024 !B → 1024 B、61 !s → 61 s）。 単位記号と単位変換の詳細や例については、アイテム値のサフィックスを参照してください。
更新間隔	このアイテムの新しい値を N 秒ごとに取得します。最大許容更新間隔は 86400 秒（1 日）です。 時間のサフィックスがサポートされています（例：30s、1m、2h、1d）。 ユーザーマクロがサポートされています。 1 つのマクロでフィールド全体を埋める必要があります。フィールド内の複数のマクロやテキストと混在したマクロはサポートされていません。 注: カスタム間隔が非ゼロ値で存在する場合のみ、更新間隔を「0」に設定できます。0 に設定し、カスタム間隔（フレキシブルまたはスケジュール）が非ゼロ値で存在する場合、アイテムはカスタム間隔の期間中にポーリングされます。 注: アイテムがアクティブになった後や更新間隔が変更された後の最初のアイテムポーリングは、設定値より早く発生する場合があります。 新しいアイテムは、作成から 60 秒以内にチェックされます（スケジューリングまたはフレキシブル更新間隔があり、更新間隔が 0 の場合を除く）。 既存のパッシブアイテムは、今すぐ実行ボタンを押すことで即座に値を取得できます。
カスタム間隔	アイテムチェックのカスタムルールを作成できます： フレキシブル - 更新間隔の例外（異なる頻度の間隔）を作成します。 スケジューリング - カスタムポーリングスケジュールを作成します。 詳細はカスタム間隔を参照してください。 間隔フィールドでは時間のサフィックスがサポートされています（例：30s、1m、2h、1d）。 ユーザーマクロがサポートされています。 1 つのマクロでフィールド全体を埋める必要があります。フィールド内の複数のマクロやテキストと混在したマクロはサポートされていません。
タイムアウト	アイテムチェックのタイムアウトを設定します（サポートされているアイテムタイプで利用可能）。 タイムアウトオプションを選択： グローバル - プロキシ/グローバルタイムアウトが使用されます（グレーアウトされたタイムアウトフィールドに表示）。 上書き - カスタムタイムアウトが使用されます（タイムアウトフィールドで設定；許容範囲：1 - 600 秒）。時間のサフィックス（例：30s、1m）およびユーザーマクロがサポートされています。 タイムアウトリンクをクリックすると、プロキシタイムアウトやグローバルタイムアウト（プロキシ未使用時）を設定できます。タイムアウトリンクは、管理 > 一般または管理 > プロキシフロントエンドセクションへの権限を持つスーパ管理者タイプのユーザーのみが表示できます。

パラメータ	説明
履歴	次のいずれかを選択： 保存しない - アイテム履歴は保存されません。従属アイテムのみ履歴を保持する必要があるマスターアイテムに便利です。 この設定はグローバルハウスキーパー設定で上書きできません。 最大保存期間 - データベースに詳細な履歴を保持する期間を指定します（1 時間～25 年）。古いデータはハウスキーパーによって削除されます。秒単位で保存されます。 時間のサフィックスがサポートされています（例：2h、1d）。ユーザーマクロがサポートされています。 最大保存期間の値は管理 > ハウスキーピングでグローバルに上書きできます。  グローバル上書き設定が存在する場合、オレンジ色の情報アイコンが表示されます。マウスを重ねると警告メッセージが表示されます（例：グローバルハウスキーパー設定で上書きされています（*1d*））。 データベースの値履歴サイズを小さくするために、記録値はできるだけ短期間保持することを推奨します。長期間の値履歴を保存する代わりに、トレンドデータを長期間保存できます。 履歴とトレンドも参照してください。
トレンド	次のいずれかを選択： 保存しない - トレンドは保存されません。 この設定はグローバルハウスキーパー設定で上書きできません。 最大保存期間 - データベースに集計（1 時間ごとの最小、最大、平均、カウント）履歴を保持する期間を指定します（1 日～25 年）。古いデータはハウスキーパーによって削除されます。秒単位で保存されます。 時間のサフィックスがサポートされています（例：24h、1d）。ユーザーマクロがサポートされています。 最大保存期間の値は管理 > ハウスキーピングでグローバルに上書きできます。  グローバル上書き設定が存在する場合、オレンジ色の情報アイコンが表示されます。マウスを重ねると警告メッセージが表示されます（例：グローバルハウスキーパー設定で上書きされています（*7d*））。 注: トレンドの保持は非数値データ（文字列、ログ、テキスト）には利用できません。 履歴とトレンドも参照してください。
値のマッピング	このアイテムに値のマッピングを適用します。値のマッピングは受信値を変更せず、表示のみを目的としています。 数値（符号なし）、数値（浮動小数点）、文字列アイテムで動作します。 例：「Windows サービスの状態」。
ログ時刻フォーマット	ログタイプのアイテムのみ利用可能。サポートされているプレースホルダー： **y**: 年（*1970-2038*）。 **M**: 月（*01-12*）。 **d**: 日（*01-31*）。 **h**: 時（*00-23*）。 **m**: 分（*00-59*）。 **s**: 秒（*00-59*）。 空白の場合、タイムスタンプは Unix 時間で 0（1970 年 1 月 1 日）に設定されます。 例として、Zabbix エージェントのログファイルの次の行を考えます： ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” この行は PID の 6 文字、日付、時刻、メッセージの残りで始まります。 この行のログ時刻フォーマットは「ppppp:yyyyMMdd:hhmmss」となります。 「p」や「:」はプレースホルダーであり、「yMdhms」以外の任意の文字にできます。
ホストインベントリフィールドに値を反映	アイテムの値を反映させるホストインベントリフィールドを選択できます。ホストで自動インベントリ反映が有効な場合に機能します。 情報の型が「ログ」に設定されている場合、このフィールドは利用できません。
説明	アイテムの説明を入力します。ユーザーマクロがサポートされています。
有効	チェックボックスをオンにするとアイテムが有効になり、処理されます。 アイテムを無効にすると、直ちに履歴キャッシュから削除されます（最後の値はログ用に保持されます）。
最新データ	リンクをクリックするとアイテムの最新データを表示できます。 このリンクは既存のアイテムを編集する場合のみ利用できます。

Note:
アイテムタイプ固有のフィールドは該当ページで説明されています。

Note:
既存のテンプレートレベルのアイテムをホストレベルで編集する場合、多くのフィールドは読み取り専用です。フォームヘッダーのリンクを使用してテンプレートレベルに移動し、そこで編集できますが、テンプレートレベルでの変更は、そのテンプレートがリンクされているすべてのホストのアイテムに影響することに注意してください。

タグタブでは、アイテムレベルのタグを定義できます。

アイテム値の事前処理

事前処理タブでは、受信した値に対する変換ルールを定義できます。

アイテムのテスト

Attention:
アイテムのテストを行うには、サーバーとプロキシのシステム時刻が同期されていることを確認してください。サーバーの時刻が遅れている場合、アイテムのテストで「The task has been expired.」というエラーメッセージが返されることがあります。ただし、サーバーとプロキシで異なるタイムゾーンを設定しても、テスト結果には影響しません。

アイテムをテストし、正しく設定されていれば実際の値を取得することができます。アイテムを保存する前でもテストを行うことができます。

ホストおよびテンプレートのアイテム、アイテムプロトタイプ、低レベルディスカバリールールでテストが可能です。アクティブアイテムではテストできません。

アイテムのテストは、以下のパッシブアイテムタイプで利用できます:

- Zabbix エージェント
- SNMP エージェント (v1, v2, v3)
- IPMI エージェント
- SSH チェック
- Telnet チェック
- JMX エージェント
- シンプルチェック（`icmpping*`、`vmware.*` アイテムを除く）
- Zabbix 内部
- 計算アイテム
- 外部チェック
- データベースモニター
- HTTP エージェント
- スクリプト
- ブラウザ

アイテムをテストするには、アイテム設定フォームの下部にあるテストボタンをクリックします。テストできないアイテム（アクティブチェックや除外されたシンプルチェックなど）では、テストボタンは無効になります。

アイテムのテストフォームには、必要なホストパラメータ（ホストアドレス、ポート、サーバー/プロキシでテスト（プロキシ名））や、アイテム固有の詳細（SNMPv2 コミュニティや SNMPv3 セキュリティ認証情報など）のフィールドがあります。これらのフィールドはコンテキストに応じて変化します:

- エージェントを必要とするアイテムの場合、ホストの選択されたエージェントインターフェースから情報を取得して値が自動入力されます。
- テンプレートアイテムの場合は、値を手動で入力する必要があります。
- プレーンテキストのマクロ値は解決されます。
- 値（または値の一部）がシークレットまたは Vault マクロであるフィールドは空になり、手動で入力する必要があります。アイテムパラメータにシークレットマクロ値が含まれている場合、次の警告メッセージが表示されます:「アイテムにシークレット値を持つユーザー定義マクロが含まれています。これらのマクロの値は手動で入力する必要があります。」
- アイテムタイプのコンテキストで不要な場合はフィールドが無効になります（例: 計算アイテムではホストアドレスフィールドやプロキシフィールドが無効）

アイテムをテストするには、値の取得をクリックします。値が正常に取得されると、値フィールドに表示され、現在の値（存在する場合）は前回の値フィールドに移動し、前回との差フィールド（2 つの値（クリック）の時間差）も計算されます。また、EOL シーケンスを検出し、取得した値に”\n\r” が含まれている場合は CRLF に切り替えようとします。

値の取得とテストをクリックすると、前処理のテストも行います。

ホストから取得した値やテスト結果は、フロントエンドに送信される際に最大 512KB までに切り詰められます。結果が切り詰められた場合は、警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータでも、Zabbix サーバーでは完全に処理されることに注意してください。

設定が正しくない場合は、考えられる原因を説明するエラーメッセージが表示されます。

ホストから正常に取得した値は、前処理ステップのテストにも利用できます。

フォームのボタン

フォームの下部にあるボタンで、いくつかの操作を行うことができます。

Add	アイテムの追加。このボタンは新規のアイテムの時のみ使用できます。
Update	アイテムのプロパティの更新。
Clone	現在のアイテムプロパティをベースにして別のアイテムを作成。
Execute now	アイテムの値を取得。アイテムのタイプがパッシブなものだけ対応しています。(詳細参照) 注意値の取得時には設定キャッシュは更新されません。つまり直近のアイテムの設定変更が反映されません。
Test	値を取得できるかのテスト実行。
Delete and clear	アイテムのヒストリとトレンドを削除。
Delete	アイテムの削除。
Cancel	アイテムのプロパティ変更をキャンセル。

テキストデータの制限

テキストデータの制限は、データベースのバックエンドによって異なります。テキスト値は、データベースの値の型の制限に合わせて切り捨てられてからデータベースに保存されます:

データベース	情報の種類		
	Character	**Log**	**Text**
MySQL	255 文字	65536 バイト	65536 バイト
PostgreSQL	255 文字	65536 文字	65536 文字
SQLite (Zabbix プロキシのみ)	255 文字	65536 文字	65536 文字

カスタムスクリプトの制限

使用可能なカスタムスクリプトの長さは、使用するデータベースによって異なります。

データベース	文字数の上限	バイト数の上限
MySQL	65535	65535
PostgreSQL	65535	制限なし
SQLite (Zabbix プロキシのみ)	65535	制限なし

アイテムタイムアウト

アイテムタイムアウトは、Zabbix がチェックを失敗として中止するまでの待機時間を指定します。

タイムアウトに達すると、データ取得が完了していなくてもチェックは中止されます。データが部分的に受信された場合、アイテムはサポート対象外となり、エラーメッセージが記録されます（例えば、SNMP チェックで複数の OID のうち 1 つだけが正常に収集された場合など）。

多くのアイテムタイプでは、柔軟なアイテムタイムアウトを設定できます：

- 単一アイテムごと
- アイテムタイプごと（プロキシレベル）
- アイテムタイプごと（グローバルレベル）

単一アイテムごとのカスタムタイムアウトは、特定のアイテムに対してより長いタイムアウトを指定し、他のアイテムのタイムアウトを低く保ちたい場合に便利です。

タイムアウトの優先順位

1. 単一アイテムのタイムアウトが他のタイムアウトより優先されます。
2. プロキシレベルのタイムアウトがグローバルタイムアウトより優先されます。

Zabbixサーバーまたはプロキシの設定からのタイムアウトは、柔軟なタイムアウトが設定されているチェックには影響しません。

柔軟なタイムアウトのサポート

以下のアイテムタイプで柔軟なタイムアウトがサポートされています:

- Zabbix エージェント (パッシブおよびアクティブチェックの両方)
- SNMP エージェント (レガシー SNMP チェック[1] を除く)
- シンプルチェック (icmpping* および VMware アイテム[2] を除く)
- SSH エージェント
- Telnet エージェント
- 外部チェック
- データベース監視
- スクリプト
- HTTP エージェント
- ブラウザ

[1] レガシー SNMP チェック (単一 OID ポーリング) の場合、サーバーまたはプロキシのタイムアウト設定が適用されます。
 [2] icmpping* アイテムの場合、タイムアウト値はアイテムキーで直接指定します。VMware 監視アイテムは、サーバーまたはプロキシの VMwareTimeout パラメータを使用します。

サポートされていないアイテム

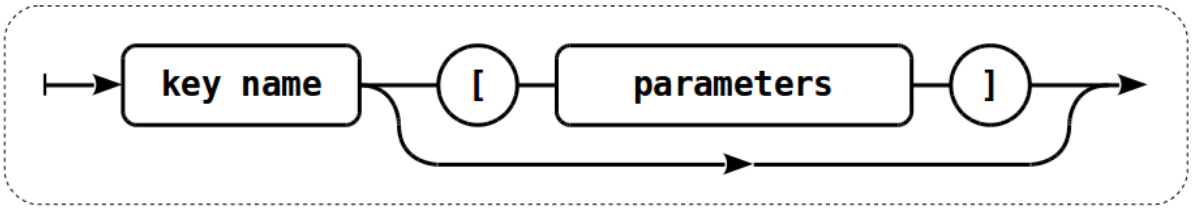
サポートされていないアイテムはサポートされていませんのステータスとして報告され、標準の更新間隔で再チェックされます。

アイテムの値が何らかの理由で取得できない場合（例：接続エラー、アイテムを処理するポーラーが設定されていない場合）、アイテムはサポートされていない状態になります。さらに、履歴データを受信しないアイテムはサポートされていませんのステータスのままです。これは想定された動作であり、アイテムは新しい有効な履歴データを受信したときにのみ通常のステータスに移行します。

問題と同様に、サポートされていないアイテムは新しいデータが受信されたときにのみ再評価されます。たとえそのアイテムに履歴データがもう利用できなくてもです。言い換えれば、アイテムとトリガーは新しいデータを受信したときにのみ状態を変更します。ただし、トリガー式に日付と時刻やnodata()関数が含まれている場合、トリガーは 30 秒ごとに再計算されます（詳細はトリガーの計算時間を参照してください）。

1 アイテムキーのフォーマット

キーパラメーターを含むアイテムキーのフォーマットは、構文規則に従う必要があります。次の図は、サポートされている構文を示しています。各ポイントで許可される要素と文字は、矢印に従って決定できます。矢印の先にあるブロックに到達できる場合は許可され、到達できない場合は許可されません。



有効なアイテムキーを作成するには、キー名を指定し、次にパラメーターの有無を選択します。

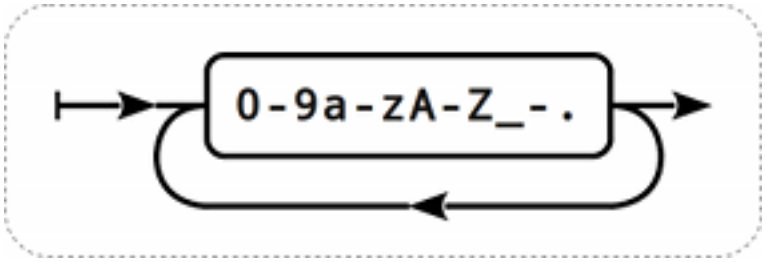
キー名

キー名自体には、使用できる文字の範囲が限られており、その文字が相互に続くだけです。許容される文字は、

0-9a-zA-Z_-.

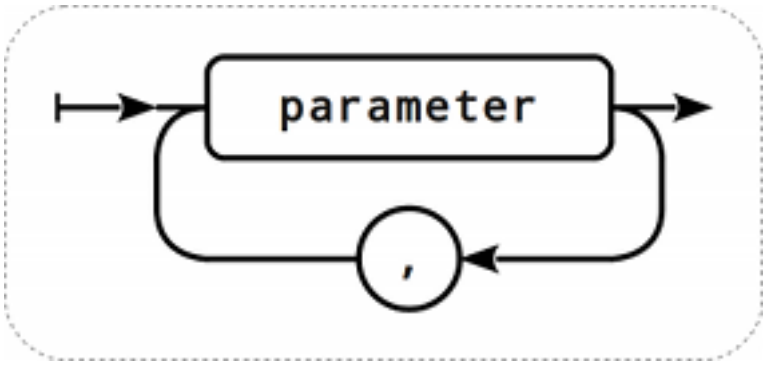
つまり、以下の文字種を指します。

- すべての数字
- すべての小文字
- すべての大文字
- アンダースコア
- ダッシュ
- ドット

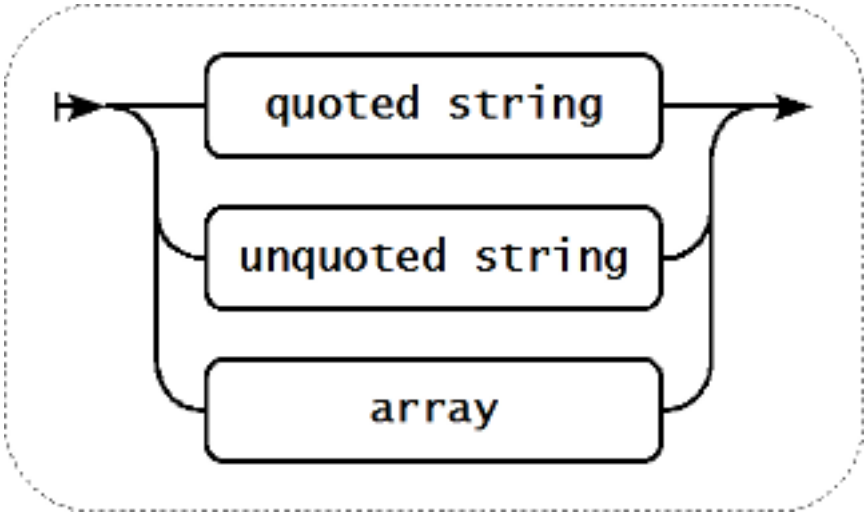


キーパラメーター

アイテムキーは、カンマ区切りで複数のパラメーターを持つことができます。



各キーパラメーターは、引用符で囲まれた文字列、引用符で囲まれていない文字列、または配列のいずれかになります。

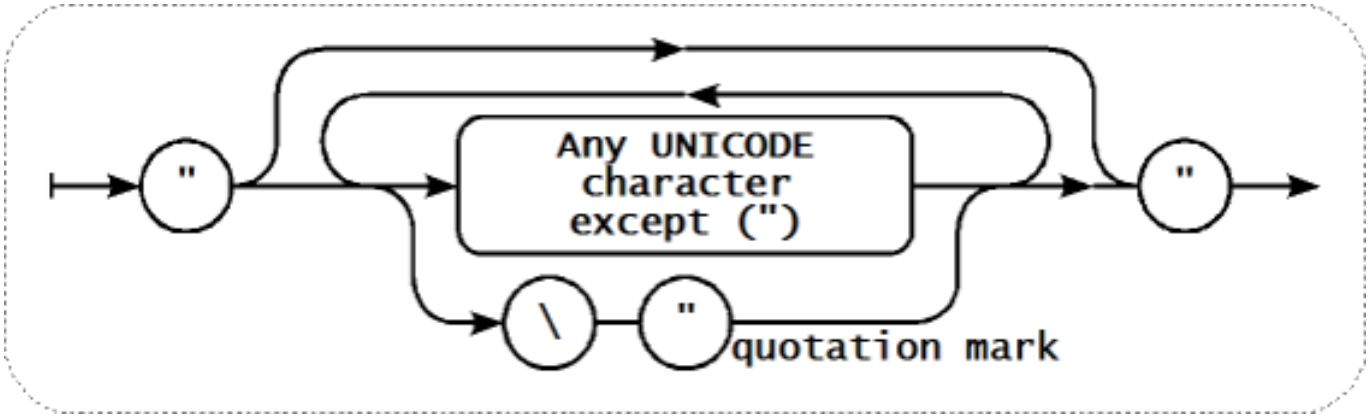


また、パラメーターを空にして、デフォルト値を使用することも可能です。その場合、さらにパラメーターを指定する場合は、適切な数のカンマを追加する必要があります。例えば、アイテムキー **icmpping[„200„500]** は、個々の ping の間隔が 200 ミリ秒、ping のタイムアウトが 500 ミリ秒、その他のパラメーターはデフォルトのままです。

パラメーターにマクロを含めることができます。マクロはユーザーマクロまたは組み込みマクロの一部です。アイテムキーパラメーターでサポートされている特定の組み込みマクロを確認するには、サポートされているマクロページで” アイテムキーパラメーター” を検索してください。

パラメーター - 二重引用符 (”) で囲まれた文字列

キーパラメーターが二重引用符で囲まれた文字列の場合、任意の Unicode 文字が使用可能です。キーパラメーターが二重引用符を含む場合、このパラメーターを二重引用符で囲み、パラメーター文字列の一部である各二重引用符はバックスラッシュ (\) でエスケープする必要があります。キーパラメーターがカンマを含む場合、このパラメーターは二重引用符で囲む必要があります。

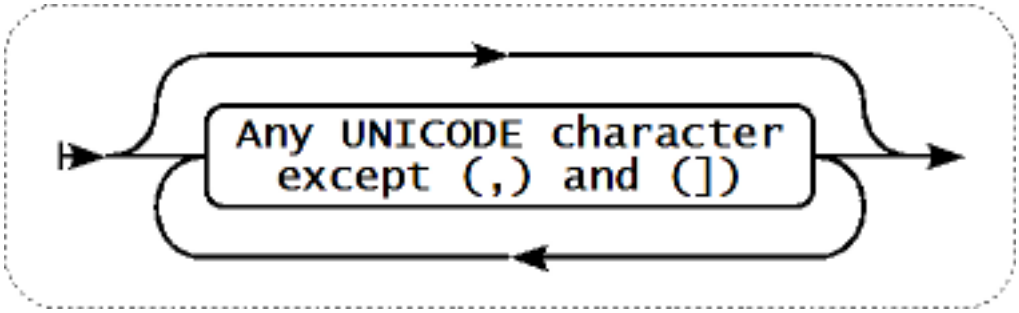


Warning:
アイテムのキーパラメーターを引用符で囲むには、二重引用符 (”) のみを使用してください。一重引用符 (’) はサポートされていません。

Attention:
`[a,[b,[c,d]],e]` のようなマルチレベルパラメーター配列は許可されていません。

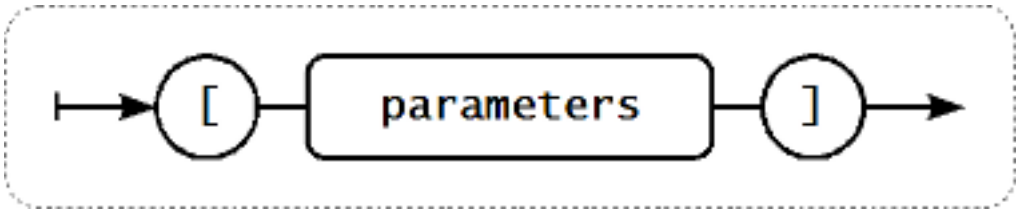
パラメーター - 二重引用符 (”) がない文字列

キーパラメーターに二重引用符がない文字列の場合、カンマと右大括弧 (]) を除く全ての Unicode 文字が使用可能です。二重引用符で囲まれていないパラメータで、左大括弧 ([) で始まる文字列は使用できません。



パラメーター - 配列

キーパラメーターが配列の場合、これも角括弧で囲まれ、個々のパラメーターは複数のパラメーターを指定する規則と構文に従います。



2 監視間隔のカスタマイズ

概要

アイテムがチェックされる時間に関するカスタムルールを作成することができます。そのための 2 つの方法は、デフォルトの更新間隔を再定義できるフレキシブル間隔と、アイテムチェックを特定の時刻または一連の時刻に実行できるスケジューリングです。

柔軟な間隔

柔軟な間隔を使用すると、特定の期間にデフォルトの更新間隔を再定義できます。柔軟な間隔は、間隔と期間で定義されます。

- 間隔 – 指定した期間の更新間隔。時間のサフィックスがサポートされています（例：30s、1m、2h、1d）。
- 期間 – 柔軟な間隔が有効な期間（期間のフォーマットの詳細については時間の期間を参照してください）

複数の柔軟な間隔が重複している場合、重複期間には最小の間隔値が使用されます。重複する柔軟な間隔の最小値が「0」の場合、ポーリングは行われません。柔軟な間隔の外では、デフォルトの更新間隔が使用されます。

柔軟な間隔が期間の長さと等しい場合、アイテムは正確に 1 回チェックされます。柔軟な間隔が期間よりも大きい場合、アイテムは 1 回チェックされるか、まったくチェックされない場合があります（このような設定は推奨されません）。柔軟な間隔が期間よりも短い場合、アイテムは少なくとも 1 回チェックされます。

柔軟な間隔が「0」に設定されている場合、柔軟な間隔の期間中はアイテムはポーリングされず、期間が終了するとデフォルトの更新間隔に従ってポーリングが再開されます。例：

間隔	期間	説明
10	1-5,09:00-18:00	アイテムは営業時間中、10 秒ごとにチェックされます。
0	1-7,00:00-7:00	アイテムは夜間はチェックされません。
0	7-7,00:00-24:00	アイテムは日曜日はチェックされません。
60	1-7,12:00-12:01	アイテムは毎日 12:00 にチェックされます。これはスケジュールされたチェックの回避策として使用されていましたが、このようなチェックにはスケジューリング間隔を使用することを推奨します。

スケジューリング間隔

スケジューリング間隔は、特定の時刻にアイテムをチェックするために使用されます。柔軟な間隔がデフォルトのアイテム更新間隔を再定義するために設計されているのに対し、スケジューリング間隔は独立したチェックスケジュールを指定するために使用され、並行して実行されます。

スケジューリング間隔は、`md<filter>wd<filter>h<filter>m<filter>s<filter>` の形式で定義されます。ここで:

- **md** - 月の日
- **wd** - 週の日
- **h** - 時間
- **m** - 分
- **s** – 秒

`<filter>` は、その接頭辞 (日、時間、分、秒) に対する値を指定するために使用され、`[<from>[-<to>]][/<step>][,<filter>]` の形式で定義されます。ここで:

- `<from>` と `<to>` は一致する値の範囲（含む）を定義します。`<to>` が省略された場合、フィルタは `<from> - <from>` の範囲に一致します。`<from>` も省略された場合、フィルタはすべての可能な値に一致します。
- `<step>` は、範囲内で値をスキップする間隔を定義します。デフォルトでは `<step>` の値は 1 で、定義された範囲のすべての値に一致することを意味します。

フィルタの定義は省略可能ですが、少なくとも 1 つのフィルタを使用する必要があります。フィルタには、範囲または *<step>* の値のいずれかを定義する必要があります。

下位レベルのフィルタが定義されていない場合、空のフィルタは'0' に一致し、そうでない場合はすべての可能な値に一致します。たとえば、時間フィルタが省略された場合、分および秒フィルタも省略されていれば'0' 時間のみが一致し、そうでなければ空の時間フィルタはすべての時間値に一致します。

それぞれのフィルタ接頭辞に対する有効な `<from>` および `<to>` の値は次のとおりです:

Prefix	Description	*<from>*	*<to>*
md	月の日	1-31	1-31
wd	週の日	1-7	1-7
h	時間	0-23	0-23
m	分	0-59	0-59
s	秒	0-59	0-59

`<from>` の値は `<to>` の値以下でなければなりません。`<step>` の値は 1 以上、`<to> - <from>` 以下でなければなりません。

1 桁の月の日、時間、分、秒の値には 0 を付けることができます。たとえば、`md01-31` や `h/02` は有効な間隔ですが、`md01-031` や `wd01-07` は無効です。

Zabbix フロントエンドでは、複数のスケジューリング間隔は別々の行に入力します。Zabbix API では、セミコロン ; で区切って 1 つの文字列に連結されます。

時刻が複数の間隔に一致した場合でも、1 回だけ実行されます。たとえば、`wd1h9;h9` は月曜日の午前 9 時に 1 回だけ実行されます。

例:

Interval	Will be executed
m0-59	毎分
h9-17/2	9:00 から 2 時間ごと（9:00、11:00 ...）
m0,30 または m/30	毎時 hh:00 および hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 または m/5	5 分ごと
wd1-5h9	月曜日から金曜日の 9:00
wd1-5h9-18	月曜日から金曜日の 9:00,10:00,...,18:00
h9,10,11 または h9-11	毎日 9:00、10:00、11:00
md1h9m30	毎月 1 日の 9:30
md1wd1h9m30	毎月 1 日が月曜日の場合の 9:30
h9m/30	毎日 9:00、9:30
h9m0-59/30	毎日 9:00、9:30
h9,10m/30	毎日 9:00、9:30、10:00、10:30
h9-10m30	毎日 9:30、10:30
h9m10-40/30	毎日 9:10、9:40
h9,10m10-40/30	毎日 9:10、9:40、10:10、10:40
h9-10m10-40/30	毎日 9:10、9:40、10:10、10:40
h9m10-40	毎日 9:10、9:11、9:12、... 9:40
h9m10-40/1	毎日 9:10、9:11、9:12、... 9:40
h9-12,15	毎日 9:00、10:00、11:00、12:00、15:00
h9-12,15m0	毎日 9:00、10:00、11:00、12:00、15:00
h9-12,15m0s30	毎日 9:00:30、10:00:30、11:00:30、12:00:30、15:00:30
h9-12s30	毎日 9:00:30、9:01:30、9:02:30 ... 12:58:30、12:59:30
h9m/30;h10 (*API* 固有の構文)	毎日 9:00、9:30、10:00
h9m/30 h10 (フロントエンドで別の行として追加)	毎日 9:00、9:30、10:00

プロキシとエージェントのタイムゾーンの調整

Zabbix プロキシとエージェントは、スケジューリング間隔を処理する際にローカルのタイムゾーンを使用することに注意してください。

このため、Zabbix プロキシまたはアクティブエージェントによって監視されるアイテムにスケジューリング間隔を適用する場合は、キューがアイテムの遅延を正しく報告しない可能性があるため、該当するプロキシまたはエージェントのタイムゾーンを Zabbix サーバーと同じに設定することを推奨します。

Zabbix プロキシまたはエージェントのタイムゾーンは、systemd のユニットファイルで環境変数 TZ を使用して設定できます:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2 アイテムの値の保存前処理

概要

前処理を使用すると、受信したアイテム値をデータベースに保存する前に変換を適用できます。これらの変換/前処理ステップは、Zabbix サーバーまたはプロキシ（アイテムがプロキシによって監視されている場合）によって実行されます。

この機能は、以下のようなさまざまなユースケースをサポートします。

- バイトをビットに変換する（例：ネットワークトラフィック値に「8」を掛ける）
- 増分値の毎秒統計を計算する
- 正規表現を適用して値を抽出または変更する
- 値に対してカスタムスクリプトを実行する
- 変更されていない値を破棄してデータベースストレージを最適化する

1 つまたは複数の前処理ステップをアイテムに設定できます。これらのステップは、設定された順序で実行されます。

Attention:
前処理ステップが失敗した場合、アイテムはサポート対象外になります。これは、失敗時のカスタムエラー処理（ほとんどの変換で利用可能）によって回避でき、値を破棄したりカスタム値を設定したりできます。

 ログアイテムの場合、ログメタデータ（値なし）は常にアイテムのサポート対象外状態をリセットし、再びサポート対象にします。これは、最初のエラーがエージェントからログ値を受信した後に発生した場合でも発生します。

前処理に渡されるすべての値は、最初は文字列として扱われます。希望する値の型への変換（アイテム設定で定義）は、前処理パイプラインの最後で実行されます。ただし、特定の前処理ステップでは、必要に応じて早期に変換がトリガーされる場合があります。詳細な技術情報については、前処理の詳細を参照してください。

前処理設定が期待どおりに機能することを確認するには、テストできます。

参考: 前処理の例

設定

前処理ステップは、アイテム設定フォームの前処理タブで定義します。

追加をクリックして、サポートされている変換を選択します。

前処理ステップが少なくとも 1 つ定義されている場合、タブの下部に情報の型フィールドが表示されます。必要に応じて、前処理タブを離れることなく情報の型を変更することができます。詳細なパラメータの説明については、アイテムの作成を参照してください。

サポートされている変換

サポートされているすべての変換は以下にリストされています。変換名をクリックすると、その詳細が表示されます。

名前	説明	タイプ
正規表現	値を正規表現にマッチさせ、必要な出力に置き換えます。	テキスト
置換	検索文字列を見つけて、別の文字列（または何もない）に置き換えます。	
トリム	値の先頭と末尾から指定した文字を削除します。	
右トリム	値の末尾から指定した文字を削除します。	
左トリム	値の先頭から指定した文字を削除します。	
XML XPath	XPath 機能を使用して XML データから値またはフラグメントを抽出します。	構造化データ
JSON Path	JSONPath 機能を使用して JSON データから値またはフラグメントを抽出します。	
CSV から JSON	CSV ファイルデータを JSON 形式に変換します。	
XML から JSON	XML 形式のデータを JSON に変換します。	
SNMP walk 値	指定した OID/MIB 名で値を抽出し、フォーマットオプションを適用します。	SNMP
SNMP walk から JSON	SNMP 値を JSON に変換します。	
SNMP get 値	SNMP get 値にフォーマットオプションを適用します。	
カスタム乗数	値に指定した整数または浮動小数点値を掛けます。	算術
単純変化量	現在値と前回値の差を計算します。	変化量
秒間変化量	現在値と前回値の差分の 1 秒あたりの変化速度を計算します。	
ブール値から 10 進数	値をブール形式から 10 進数に変換します。	数値システム
8 進数から 10 進数	値を 8 進数形式から 10 進数に変換します。	
16 進数から 10 進数	値を 16 進数形式から 10 進数に変換します。	
JavaScript	JavaScript コードを入力します。	カスタムスクリプ
範囲内	値が収まるべき範囲を定義します。	バリデーション
正規表現に一致	値が一致しなければならない正規表現を指定します。	
正規表現に一致しない	値が一致してはならない正規表現を指定します。	
JSON でエラーをチェック	JSONPath にあるアプリケーションレベルのエラーメッセージをチェックします。	
XML でエラーをチェック	XPath にあるアプリケーションレベルのエラーメッセージをチェックします。	
正規表現でエラーをチェック	正規表現を使用してアプリケーションレベルのエラーメッセージをチェックします。	
サポートされていない値をチェック	アイテム値が取得できなかったかどうかをチェックします。	
変更なしを破棄	値が変更されていない場合は破棄します。	スロットリング
ハートビート付き変更なしを破棄	定義された期間内に値が変更されていない場合は破棄します。	
Prometheus パターン	以下のクエリを使用して Prometheus メトリクスから必要なデータを抽出します。	Prometheus
Prometheus から JSON	必要な Prometheus メトリクスを JSON に変換します。	

変化量およびスロットリングの前処理ステップでは、Zabbix は新しい値を計算/比較するために最後の値を記憶しておく必要があることに注意してください。これらの前回値は前処理マネージャによって管理されます。Zabbix サーバーまたはプロキシが再起動された場合や、前処理ステップに変更があった場合、対応するアイテムの最後の値がリセットされ、以下のようになります。

- 単純変化量、秒間変化量ステップの場合 - 前回値がないため、次の値は無視されます。
- 変更なしを破棄、ハートビート付き変更なしを破棄ステップの場合 - 破棄ルールにより本来破棄されるべきであっても、次の値は決して破棄されません。

正規表現

値を正規表現に一致させ、必要な出力に置き換えます。

パラメータ:

- **pattern** - 正規表現;

- **output** - 出力フォーマットテンプレート。\N (N=1...9) エスケープシーケンスは、N 番目の一致グループに置き換えられます。\0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- 入力値が一致しない場合、アイテムはサポート対象外になります;
- 正規表現は、\N シーケンスで最大 10 個のキャプチャグループの抽出をサポートします;
- *Custom on fail* チェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムはサポート対象外にはなりません。
- 既存の例については、正規表現セクションを参照してください。

{#replace} の置換

検索文字列を見つけて、別の文字列 (または何もない) に置き換えます。

パラメータ:

- 検索文字列 - 見つけて置き換える文字列、大文字と小文字を区別します (必須);
- 置換文字列 - 検索文字列を置き換える文字列。置換文字列を空にすることもでき、見つかった検索文字列を削除することができます。

コメント:

- 検索文字列のすべての出現箇所が置換されます;
- エスケープシーケンスを使用して、改行、復帰、タブ、スペース”\n \r \t \s” を検索または置換できます; バックスラッシュは”\\” でエスケープでき、エスケープシーケンスは”\\n” でエスケープできます;
- 改行、復帰、タブのエスケープは、低レベルディスカバリ中に自動的に行われます。

Trim

値の先頭と末尾から指定した文字を削除します。

右側トリム

値の末尾から指定した文字を削除します。

左トリム

値の先頭から指定した文字を削除します。

XML XPath

XPath 機能を使用して、XML データから値またはフラグメントを抽出します。

コメント:

- このオプションを機能させるには、Zabbix サーバー（または Zabbix プロキシ）が libxml サポートでコンパイルされている必要があります。
- 名前空間はサポートされていません。
- 失敗時にカスタム値を使用チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することができます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

例:

```
number(/document/item/value) #<document><item><value>10</value></item></document> から '10' を抽出します
number(/document/item/@attribute) #<document><item attribute="10"></item></document> から '10' を抽出します
/document/item #<document><item><value>10</value></item></document> から '<item><value>10</value></item>' を
```

JSON Path

JSONPath 機能を使用して、JSON データから値またはフラグメントを抽出します。

失敗時にカスタムチェックボックスをオンにすると、カスタムエラー処理オプションを指定できます。値を破棄するか、指定した値を設定するか、指定したエラーメッセージを設定するかを選択できます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

CSV to JSON

CSV ファイルデータを JSON 形式に変換します。

詳細については、CSV to JSON の事前処理を参照してください。

XML to JSON

XML 形式のデータを JSON に変換します。

詳細については、シリアライズルールを参照してください。

失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションを選択した場合は、アイテムが未サポートにはなりません。

SNMP walk 値

指定された OID/MIB 名で値を抽出し、フォーマットオプションを適用します:

- 変更しない - 16 進数文字列をエスケープされていない 16 進数文字列として返します（注：表示ヒントは引き続き適用されます）;
- **16** 進数文字列から **UTF-8** - 16 進数文字列を UTF-8 文字列に変換します;
- **16** 進数文字列から **MAC** - 16 進数文字列を MAC アドレスとして検証し、正しい MAC アドレス文字列（' 'が':'に置き換えられます）を返します;
- **BITS** から整数 - 16 進数文字（例：”1A 2B 3C 4D”）のシーケンスとして表現されたビット文字列の最初の 8 バイトを 64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列の場合、以降のバイトは無視されます。

失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

SNMP walk to JSON

SNMP 値を JSON に変換します。

JSON 内のフィールド名と対応する SNMP OID パスを指定します。フィールド値は、指定された SNMP OID パスの値で埋められます。

コメント:

- *SNMP walk value* ステップと同様の値のフォーマットオプションが利用可能です。
- この前処理ステップは、SNMP OID ディスカバリにも使用できます。
- *Custom on fail* チェックボックスをマークすると、カスタムエラー処理オプションを指定できます。値を破棄するか、指定した値を設定するか、指定したエラーメッセージを設定するかを選択できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

SNMP get value

SNMP get 値に書式設定オプションを適用します:

- **UTF-8 from Hex-STRING** - 16 進文字列を UTF-8 文字列に変換します;
- **MAC from Hex-STRING** - 16 進文字列を MAC アドレスとして検証し、適切な MAC アドレス文字列を返します（' 'は':'に置換されます）;
- **Integer from BITS** - 16 進文字列で表現されたビット文字列の最初の 8 バイト（例: ”1A 2B 3C 4D”）を 64 ビットの符号なし整数に変換します。8 バイトを超えるビット文字列の場合、以降のバイトは無視されます。

Custom on fail チェックボックスをオンにすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

カスタム乗数

値を指定した整数または浮動小数点値で乗算します。

コメント:

- このオプションを使用して、KB、MBps などで受信した値を B、Bps に変換します。そうしないと、Zabbix は接頭辞（K、M、G など）を正しく設定できません。
- アイテムの情報の型が数値（整数）の場合、受信値に小数部分があると、カスタム乗数が適用される前に切り捨てられます（例：’0.9’ は’0’ になります）;
- カスタム乗数を使用するか、情報の型が数値（整数）に設定されたアイテムに対して *1* 秒あたりの変化量として値を保存し、計算された値が実際には浮動小数点数である場合でも、小数部分を切り捨てて値を整数として保存することで、計算された値は正しいものとして受け入れられます ;
- サポートされているもの：科学的記数法（例：1e+70）、ユーザーマクロおよび LLD マクロ、マクロを含む文字列（例：`{#MACRO}e+10`、`{$MACRO1}e+{$MACRO2}`）。マクロは整数または浮動小数点数に解決される必要があります。
- 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

単純な変化

現在の値と前回の値の差分を計算します。

コメント:

- このステップは、常に増加する値を測定するのに役立ちます。
- **value-prev_value** として評価されます。ここで、*value* は現在の値、*prev_value* は前回受信した値です。
- アイテムごとに 1 つの変化操作（「単純な変化」または「1 秒あたりの変化」）のみが許可されます。
- 現在の値が前回の値より小さい場合、Zabbix はその差分を破棄（何も保存しない）し、次の値を待ちます。
- 失敗時にカスタム値を使用チェックボックスをオンにすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは「取得不可」にはなりません。

秒あたりの変化量

値の変化量（現在値と前回値の差分）の秒あたりの速度を計算します。

コメント:

- このステップは、常に増加する値の秒あたりの速度を計算するのに便利です。
- この計算は浮動小数点数を生成する場合があるため、受信する生の値が整数であっても「情報の型」を数値（浮動小数点）に設定することを推奨します。特に小さな数値で小数点以下の値が重要な場合に該当します。浮動小数点値が大きくなり、「float」フィールドの長さを超える可能性がある場合（この場合、値全体が失われる可能性があります）、数値（整数）を使用して小数点以下のみを切り捨てることを推奨します。
- (**value-prev_value**)/(**time-prev_time**) として評価されます。ここで、*value* は現在値、*prev_value* は前回受信値、*time* は現在のタイムスタンプ、*prev_time* は前回値のタイムスタンプです。
- アイテムごとに許可される変化演算は 1 つのみです（「単純な変化」または「秒あたりの変化量」）;
- 現在値が前回値より小さい場合、Zabbix はその差分を破棄（何も保存しない）し、次の値を待ちます。これにより、たとえば 32 ビット SNMP カウンタのラップアラウンド（オーバーフロー）などに正しく対応できます。
- 失敗時にカスタム値を使用チェックボックスをオンにすると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

ブール値から 10 進数への変換

値をブール形式から 10 進数に変換します。

コメント:

- テキスト表現は 0 または 1 のいずれかに変換されます。したがって、'TRUE' は 1 として、'FALSE' は 0 として保存されます。すべての値は大文字と小文字を区別せずに一致します。現在認識されている値は、*TRUE* の場合は true, t, yes, y, on, up, running, enabled, available, ok, master、*FALSE* の場合は false, f, no, n, off, down, unused, disabled, unavailable, err, slave です。さらに、ゼロ以外の数値は TRUE と見なされ、ゼロは FALSE と見なされます。
- 失敗時にカスタム値を使用チェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

8 進数から 10 進数

値を 8 進数形式から 10 進数に変換します。

失敗時にカスタムチェックボックスをマークすると、カスタムエラー処理オプションを指定できます。値を破棄するか、指定した値を設定するか、指定したエラーメッセージを設定するかを選択できます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムはサポート対象外にはなりません。

16 進数から 10 進数

値を 16 進数形式から 10 進数に変換します。

失敗時にカスタムチェックボックスをオンにすると、カスタムエラー処理オプションを指定できます。値を破棄するか、指定した値を設定するか、指定したエラーメッセージを設定するかを選択できます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

JavaScript

パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。

> **Attention:**
> 前処理の JavaScript で未宣言の代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

コメント:

- 利用可能な JavaScript の長さは使用するデータベースによって異なります。
- 詳細については、Javascript 前処理を参照してください。

範囲内

最小値/最大値（両端を含む）を指定して、値が含まれるべき範囲を定義します。

コメント:

- 数値が受け入れられます（任意の桁数、小数部の有無、指数部の有無、負の値を含む）；
- 最小値は最大値より小さくなければなりません；
- 少なくとも 1 つの値が必要です；
- ユーザーマクロおよびローレベルディスカバリーマクロが使用できます；
- 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

正規表現 {#matchesregexp} に一致

値が一致する必要がある正規表現を指定します。

Custom on fail チェックボックスをマークすると、カスタムエラー処理オプションを指定できます。値を破棄するか、指定した値を設定するか、指定したエラーメッセージを設定するかを選択できます。前処理ステップが失敗した場合、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムはサポート対象外にはなりません。

正規表現 {#doesnotmatchregexp} に一致しない

値が一致してはいけない正規表現を指定します。

失敗時にカスタムチェックボックスをマークすると、カスタムエラー処理オプション（値の破棄、指定値の設定、指定エラーメッセージの設定）が可能です。前処理ステップが失敗した場合、値の破棄または指定値の設定オプションが選択されていれば、アイテムは「利用不可」にはなりません。

JSON のエラーをチェック

JSONPath にあるアプリケーションレベルのエラーメッセージをチェックします。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前の前処理ステップの値で処理を続行します。

コメント:

- これらの外部サービスのエラーは、前処理ステップ情報を追加せずに、そのままユーザーに報告されます。
- 無効な JSON の解析に失敗した場合は、エラーは報告されません。
- 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

XML のエラーをチェック

XPath にあるアプリケーションレベルのエラーメッセージをチェックします。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前の前処理ステップの値で処理を続行します。

コメント:

- これらの外部サービスエラーは、前処理ステップ情報を追加せずに、そのままユーザーに報告されます。
- 無効な XML の解析に失敗した場合は、エラーは報告されません。
- 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか、指定した値を設定するオプションを選択した場合、アイテムは未サポートにはなりません。

正規表現 {#errorbyregexp} を使用したエラーのチェック

正規表現を使用してアプリケーションレベルのエラーメッセージをチェックします。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。

パラメータ:

- **pattern** - 正規表現;
- **output** - 出力フォーマットテンプレート。\N (N=1...9) エスケープシーケンスは、N 番目の一致グループに置き換えられます。\0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- これらの外部サービスエラーは、前処理ステップ情報を追加せずに、そのままユーザーに報告されます;
- *Custom on fail* チェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

サポートされていない値のチェック

アイテム値が取得できない場合にチェックします。返されたエラーメッセージを調査して、障害をどのように処理するかを指定します。

パラメータ:

- **scope** - エラー処理のスコープを選択します:
any error - すべてのエラー;
error matches - pattern で指定した正規表現に一致するエラーのみ;
error does not match - pattern で指定した正規表現に一致しないエラーのみ
- **pattern** - エラーと一致させる正規表現。scope パラメータで *any error* が選択されている場合、このフィールドは表示されません。表示されている場合、このフィールドは必須です。

コメント:

- 通常、値が取得できない/失敗した場合、アイテムはサポートされていない状態になります。この前処理ステップでは、この動作を変更できます。*Custom on fail* チェックボックスをマークすると（この前処理ステップでは常にマークされグレーアウトされています）、カスタムのエラー処理オプション（値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する）を指定できます。前処理ステップが失敗した場合でも、値を破棄するか指定した値を設定するオプションが選択されていれば、アイテムはサポートされていない状態にはなりません。
- この前処理ステップは、アイテム値が取得できない場合のみをチェックします。例えば、取得した値の型（例: 文字列）がアイテムの情報の型（例: 数値）と一致するかどうかはチェックしません。詳細は前処理の例を参照してください。型が一致しない場合、すべての前処理ステップが実行された後でもアイテムがサポートされていない状態になることがあります。型の不一致をチェックするには、例えば *Custom multiplier* 前処理ステップを使用できます。前処理の例を参照してください。
- *Set value to* または *Set error to* フィールドでは、正規表現グループのキャプチャがサポートされています。\N（N=1...9）を使用して N 番目の一致グループを取得できます。\0 を使用して一致したテキストを取得できます。
- これらのステップは常に最初の前処理ステップとして実行され、アイテムへの変更を保存した後、他のすべての前処理ステップの上に配置されます。
- 複数のサポートされていない値のチェックステップが、指定した順序でサポートされています。*any error* のステップは、このグループの最後のステップとして自動的に配置されます。

変更されていない値を破棄する

値が変更されていない場合は破棄します。

コメント:

- 値が破棄された場合、データベースには保存されず、Zabbix サーバーはこの値が受信されたことを認識しません。トリガー式は評価されず、その結果、関連するトリガーの問題は作成/解決されません。関数は、実際にデータベースに保存されたデータに基づいてのみ動作します。トレンドはデータベース内のデータに基づいて構築されるため、1 時間に値が保存されていない場合、その時間のトレンドデータもありません。
- アイテムごとに指定できるスロットリングオプションは 1 つだけです。

ハートビートで変更されていない値を破棄

定義された期間（秒単位）内に値が変更されていない場合、その値を破棄します。

コメント:

- 秒数を指定するために正の整数値がサポートされています（最小値 - 1 秒）；
- 時間のサフィックスを使用できます（例：30s、1m、2h、1d）；
- ユーザーマクロおよびローレベルディスカバリーマクロが使用できます；
- 値が破棄された場合、その値はデータベースに保存されず、Zabbix サーバーはその値が受信されたことを認識しません。トリガー式は評価されず、その結果、関連するトリガーの問題は作成/解決されません。関数は、実際にデータベースに保存されたデータに基づいてのみ動作します。トレンドはデータベース内のデータに基づいて構築されるため、1 時間値が保存されていない場合、その時間のトレンドデータもありません。
- 各アイテムごとに指定できるスロットリングオプションは 1 つだけです。

Prometheus パターン

Prometheus メトリクスから必要なデータを抽出するには、次のクエリを使用します。

詳細はPrometheus チェックを参照してください。

Prometheus to JSON

必要な Prometheus メトリクスを JSON に変換します。

詳細はPrometheus チェックを参照してください。

マクロのサポート

ユーザーマクロおよびコンテキスト付きユーザーマクロは、以下でサポートされています。

- 前処理ステップのパラメータ（JavaScript コードを含む）

- カスタムエラーハンドリングパラメータ（値を設定およびエラーを設定フィールド）

Note:
マクロが値に置き換えられるとき、マクロのコンテキストは無視されます。マクロ値はそのままコードに挿入され、JavaScript コードに値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることにご注意ください。

テスト

事前処理のテストを参照してください。

1 保存前処理のテスト

テスト

複雑な前処理パイプラインが期待通りの結果をもたらすことを確認するために、前処理ステップのテストは有用です。アイテム値が受信されて前処理されるのを待つ必要はありません。

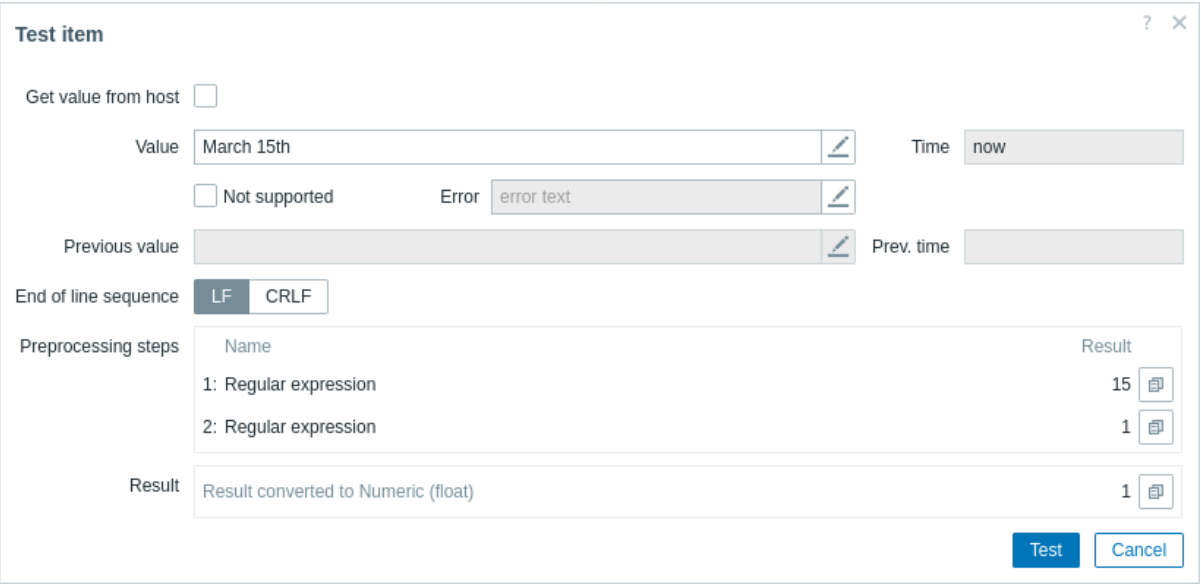


テスト可能な内容:

- 仮想値に対して
- ホストからの実際の値に対して

各前処理ステップは個別にテストできるほか、すべてのステップをまとめてテストすることもできます。アクションブロックでそれぞれテストまたはすべてのステップをテストボタンをクリックすると、テストウィンドウが開きます。

仮想値のテスト



パラメータ	説明
ホストから値を取得	仮想値をテストしたい場合は、このチェックボックスを未選択のままにします。 関連: 実際の値のテスト
値	テストする入力値を入力します。 パラメータフィールドをクリックするか、表示/編集ボタン をクリックすると、値やコードブロックを入力するためのテキストエリアウィンドウが開きます。
未サポート	未サポートの値をテストする場合は、このチェックボックスを選択します。 このオプションは、未サポート値のチェックの前処理ステップをテストするのに便利です。

パラメータ	説明
エラー	エラーテキストを入力します。 このフィールドは、ホストから値を取得が未選択で、未サポートが選択されている場合に有効になります。 ホストから値を取得が選択されている場合、このフィールドにはホストからの実際のエラーメッセージが入力されます (読み取り専用)。
時刻	入力値の時刻が表示されます: `now` (読み取り専用)。
前回の値	比較するための前回の入力値を入力します。 変化およびスロットリングの前処理ステップのみ。
前回の時刻	比較するための前回の入力値の時刻を入力します。 変化およびスロットリングの前処理ステップのみ。 デフォルト値は、アイテムの「更新間隔」フィールド値に基づきます (「1m」の場合、このフィールドには `now-1m` が入力されます)。何も指定されていない場合やユーザーがホストにアクセスできない場合、デフォルトは `now-30s` です。
マクロ	マクロが使用されている場合、それらが値とともに一覧表示されます。値はテスト目的で編集可能ですが、変更はテストの範囲内でのみ保存されます。
改行シーケンス	複数行の入力値に対する改行シーケンスを選択します: **LF** - LF(ラインフィード) シーケンス **CRLF** - CRLF(キャリッジリターン + ラインフィード) シーケンス。
前処理ステップ	前処理ステップが一覧表示され、テストボタンをクリックした後に各ステップのテスト結果が表示されます。 テスト結果は、フロントエンドに送信される際に最大 512KB までに切り捨てられます。テスト結果はコピーできます (切り捨てられた 512KB まで)。結果が切り捨てられた場合は、警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも Zabbix サーバーでは完全に処理されることに注意してください。 ステップがテストで失敗した場合、エラーアイコンが表示されます。エラーの説明はマウスオーバーで表示されます。 ステップに「失敗時のカスタム」が指定され、そのアクションが実行された場合、前処理テストステップ行の直後に新しい行が表示され、どのアクションが実行され、どのような結果 (エラーまたは値) が得られたかが示されます。
結果	前処理ステップをすべて一緒にテストした場合 (「すべてのステップをテスト」ボタンをクリックした場合) に、テストの最終結果が表示されます。 アイテムの値タイプへの変換タイプも表示されます (例: `Result converted to Numeric (unsigned)`)。 テスト結果は、フロントエンドに送信される際に最大 512KB までに切り捨てられます。最終結果もコピーできます (切り捨てられた 512KB まで)。結果が切り捨てられた場合は、警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも Zabbix サーバーでは完全に処理されることに注意してください。

テストをクリックすると、各前処理ステップの後の結果が表示されます。

テスト値は、個々のステップまたはすべてのステップのテストセッション間で保存されるため、ユーザーは前処理ステップやアイテムの設定を変更した後でも、テストウィンドウに戻って情報を再入力する必要がありません。ただし、ページをリフレッシュすると値は失われます。

テストは Zabbix サーバーによって実行されます。フロントエンドは対応するリクエストをサーバーに送信し、結果を待ちます。リクエストには、入力値と前処理ステップ (展開されたユーザーマクロ付き) が含まれます。変化およびスロットリングステップの場合、オプションで前回の値と時刻を指定できます。サーバーは各前処理ステップの結果を返します。

すべての技術的エラーや入力検証エラーは、テストウィンドウの上部にあるエラーボックスに表示されます。

実際の値でのテスト

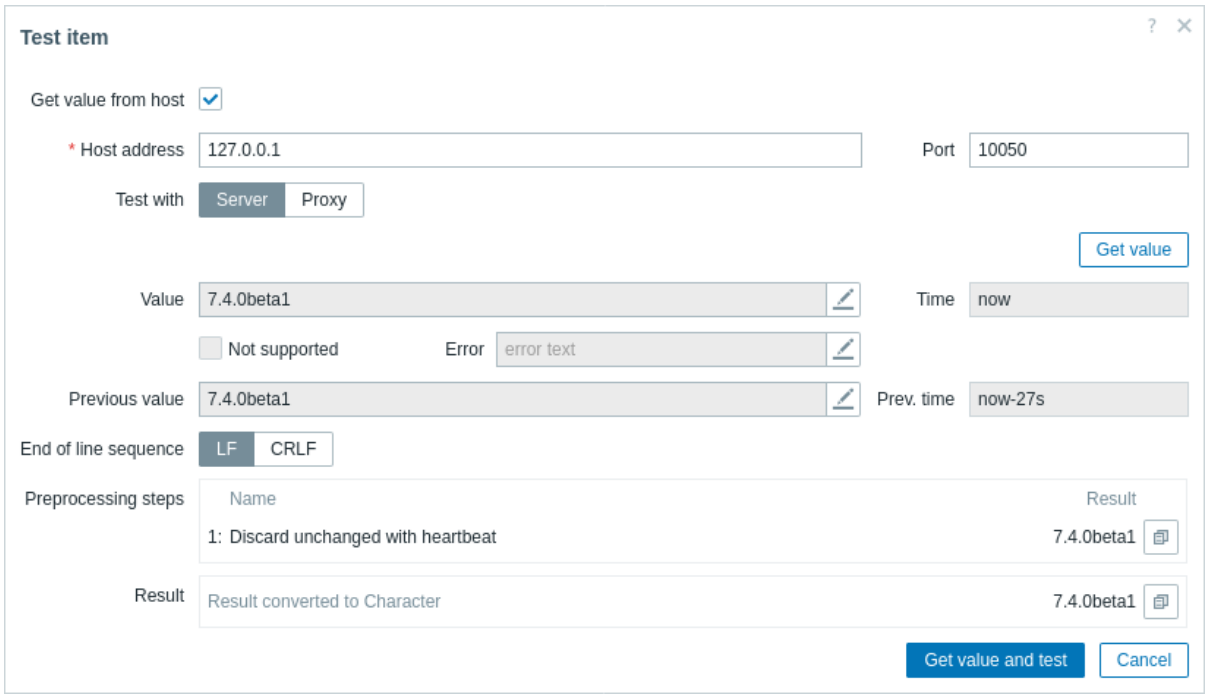
実際の値に対して前処理をテストするには:

- ホストから値を取得チェックボックスをオンにします
- ホストパラメータ（ホストアドレス、ポート、プロキシ名/プロキシなし）およびアイテム固有の詳細（SNMPv2 コミュニティや SNMPv3 セキュリティ認証情報など）を入力または確認します。これらのフィールドはコンテキストに応じて変化します:
  - 可能な場合は値が自動入力されます。たとえば、エージェントを必要とするアイテムの場合、ホストの選択されたエージェントインターフェースから情報を取得します
  - テンプレートアイテムの場合は値を手動で入力する必要があります
  - プレーンテキストのマクロ値は解決されます
  - フィールド値（または値の一部）がシークレットまたは Vault マクロの場合、フィールドは空になり、手動で入力する必要があります。アイテムパラメータにシークレットマクロ値が含まれている場合、次の警告メッセージが表示されます:

”アイテムにシークレット値を持つユーザー定義マクロが含まれています。これらのマクロの値は手動で入力する必要があります。”

- アイテムタイプのコンテキストで不要な場合はフィールドが無効になります（例: 計算アイテムの場合、ホストアドレスとプロキシフィールドは無効になります）

- 値を取得してテストをクリックして前処理をテストします



アイテム設定フォームで値のマッピング（「値の表示」フィールド）を指定している場合、アイテムテストダイアログには最終結果の後に「値のマッピング適用後の結果」という行が表示されます。

ホストから実際の値を取得するための特有のパラメータ:

パラメータ	説明
ホストから値を取得	ホストから実際の値を取得するにはこのチェックボックスをオンにします。
ホストアドレス	ホストアドレスを入力します。 このフィールドはアイテムのホストインターフェースのアドレスで自動入力されます。
ポート	ホストのポートを入力します。 このフィールドはアイテムのホストインターフェースのポートで自動入力されます。
SNMP インターフェース用の追加フィールド * (SNMP* バージョン、*SNMP* コミュニティ、コンテキスト名など*)*	SNMP インターフェース（v1、v2、v3）の設定に関する追加の詳細はSNMP 監視の設定を参照してください。 これらのフィールドはアイテムのホストインターフェースから自動入力されます。
プロキシ	ホストがプロキシによって監視されている場合はプロキシを指定します。 このフィールドはホストのプロキシ（存在する場合）で自動入力されます。
値	ホストから取得した値。 パラメータフィールドをクリックするか、表示/編集ボタン をクリックすると、値またはコードブロックのテキストエリアウィンドウが開きます。 値は最大 512KB までフロントエンドでのみ切り捨てられます。結果が切り捨てられた場合は警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも Zabbix サーバーによって完全に処理されることに注意してください。

その他のパラメータについては、上記の仮想値のテストを参照してください。

2 保存前処理の詳細

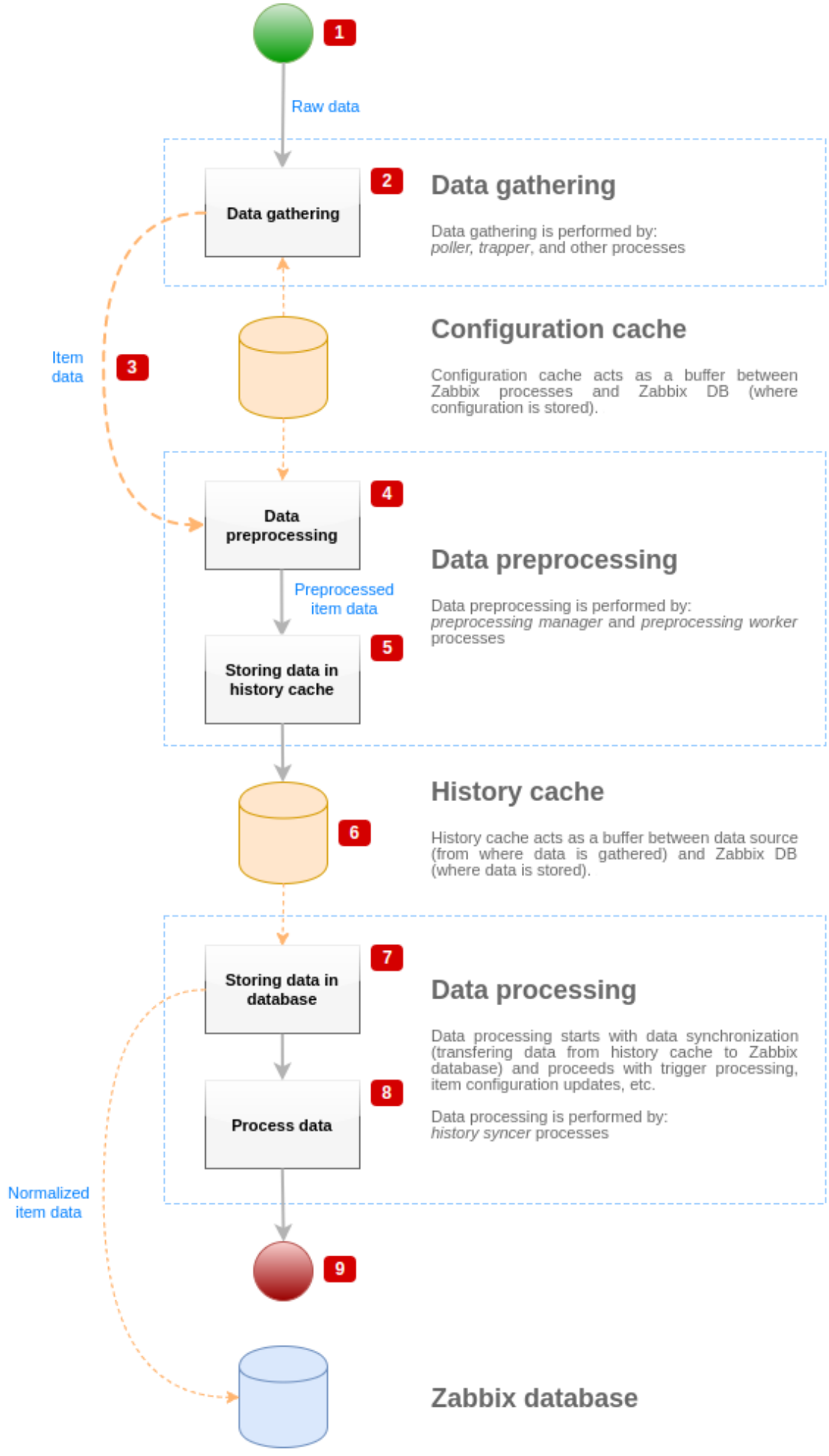
概要

このセクションでは、アイテム値の前処理の詳細について説明します。アイテム値の前処理を使用すると、受信したアイテム値に対して変換ルールを定義および実行できます。

前処理は、前処理マネージャープロセスと、前処理ステップを実行する前処理ワーカーによって管理されます。前処理付きのすべての値は、異なるデータ収集プロセスから受信され、ヒストリキャッシュに追加される前に前処理マネージャーを通過します。データ収集プロセス（ポーラー、トラッパーなど）と前処理プロセス間の通信には、ソケットベースの IPC 通信が使用されます。前処理ステップは、Zabbix サーバーまたは Zabbix プロキシ（プロキシで監視されているアイテムの場合）が実行します。

アイテム値の処理

データソースから Zabbix データベースまでのデータフローを可視化するために、以下の簡略化した図を使用できます。


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

上記の図は、簡略化された形でアイテム値の処理に関連するプロセス、オブジェクト、アクションのみを示しています。条件による方向変更、エラー処理、ループは示されていません。また、前処理マネージャのローカルデータキャッシュも、データフローに直接影響しないため表示されていません。この図の目的は、アイテム値の処理に関与するプロセスと、それらがどのように相互作用するかを示すことです。

- データ収集は、データソースからの生データで開始されます。この時点では、データには ID、タイムスタンプ、値（複数値の場合もあり）しか含まれていません。
- どのタイプのデータ収集プロセスが使用される場合でも、アクティブチェックやパッシブチェック、トラッパーアイテムなど、考え方は同じです。違いはデータフォーマットと通信の開始者（データ収集プロセスが接続とデータを待つか、データ収集プロセスが通信を開始してデータを要求するか）だけです。生データは検証され、アイテム設定が設定キャッシュから取得されます（データは設定データで拡張されます）。
- ソケットベースの IPC メカニズムを使用して、データ収集プロセスから前処理マネージャにデータを渡します。この時点で、データ収集プロセスは前処理マネージャからの応答を待たずにデータ収集を続行します。
- データの前処理が実行されます。これには、前処理ステップの実行や従属アイテムの処理が含まれます。

Note:
前処理の実行中に、いずれかの前処理ステップが失敗した場合、アイテムは NOT SUPPORTED 状態に変更されることがあります。

- 前処理マネージャのローカルデータキャッシュからヒストリキャッシュにヒストリデータがフラッシュされます。
- この時点で、次回のヒストリキャッシュの同期（history syncer プロセスがデータ同期を実行するタイミング）までデータフローは停止します。
- 同期プロセスは、Zabbix データベースにデータを保存する前にデータの正規化から始まります。データの正規化では、アイテム設定で定義されたタイプへの変換や、タイプごとに許可されている事前定義サイズに基づくテキストデータの切り捨て（文字列の場合は HISTORY_STR_VALUE_LEN、テキストの場合は HISTORY_TEXT_VALUE_LEN、ログ値の場合は HISTORY_LOG_VALUE_LEN）が行われます。正規化が完了した後、データは Zabbix データベースに送信されます。

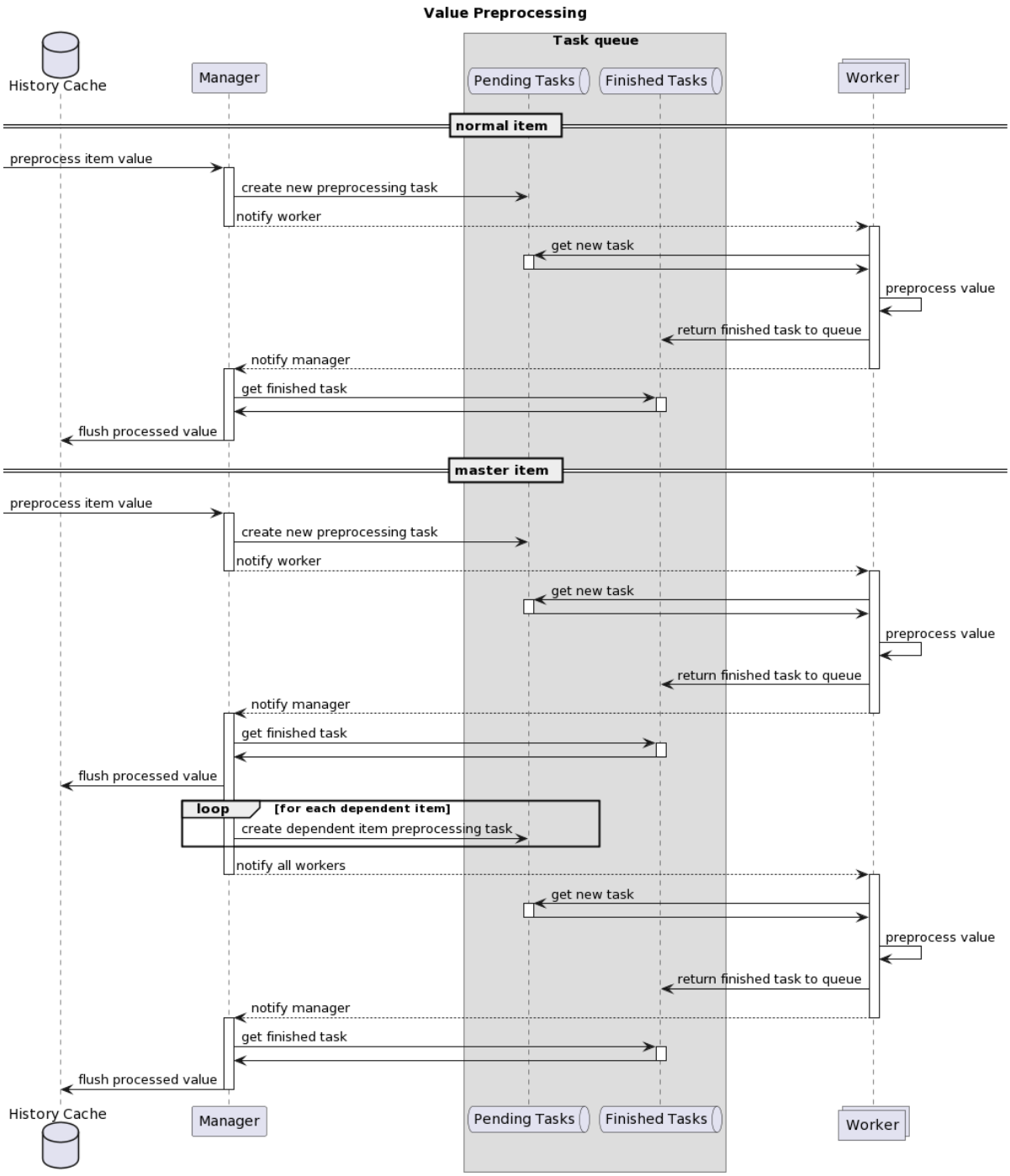
Note:
データの正規化に失敗した場合（例：テキスト値を数値に変換できない場合）、アイテムは NOT SUPPORTED 状態に変更されることがあります。

- 収集されたデータが処理されます。トリガーのチェックや、アイテムが NOT SUPPORTED になった場合のアイテム設定の更新などが行われます。
- これは、アイテム値の処理の観点から見たデータフローの終了と見なされます。

アイテム値の事前処理

データの事前処理は以下の手順で実行されます。

- アイテムに事前処理も従属アイテムもない場合、その値はヒストリキャッシュに追加されるか、LLD マネージャに送信されます。そうでない場合、アイテム値は UNIX ソケットベースの IPC メカニズムを使用して事前処理マネージャに渡されます。
- 事前処理タスクが作成され、キューに追加され、事前処理ワーカーに新しいタスクが通知されます。
- この時点で、少なくとも 1 つの空いている（すなわち、タスクを実行していない）事前処理ワーカーが現れるまでデータフローは停止します。
- 事前処理ワーカーが利用可能になると、キューから次のタスクを取得します。
- 事前処理が完了すると（事前処理ステップの失敗・成功の両方の場合）、事前処理済みの値が完了タスクキューに追加され、マネージャに新しい完了タスクが通知されます。
- 事前処理マネージャは、結果を希望するフォーマット（アイテム値タイプで定義）に変換し、ヒストリキャッシュに追加するか、LLD マネージャに送信します。
- 処理されたアイテムに従属アイテムがある場合、従属アイテムは事前処理済みのマスターアイテム値で事前処理キューに追加されます。従属アイテムは通常の値の事前処理リクエストをバイパスしてキューに追加されますが、値が設定されていて NOT SUPPORTED 状態でないマスターアイテムに対してのみ行われます。



図では、事前処理のキャッシュを省略することで、マスターアイテムの事前処理がやや簡略化されていることに注意してください。

前処理キュー

前処理キューは次のように構成されています:

- 保留中タスクのリスト:
  - 受信順に値の前処理リクエストから直接作成されたタスク
- 即時タスクのリスト (保留中タスクより先に処理される):
  - テストタスク (フロントエンドからのアイテム/前処理テストリクエストに応じて作成)
  - 従属アイテムのタスク
  - シーケンスタスク (厳密な順序で実行する必要があるタスク):
    * 最終値を使用する前処理ステップを持つもの:
      · 差分
      · スロットリング
      · JavaScript (バイトコードキャッシュ)
    * 従属アイテムの前処理キャッシュ
- 完了したタスクのリスト

前処理キャッシュ

前処理キャッシュは、同様の前処理ステップを持つ複数の依存アイテム (これは一般的な LLD の結果です) の前処理パフォーマンスを向上させるために導入されました。

キャッシュは、1 つの依存アイテムを前処理し、内部の前処理データの一部を残りの依存アイテムで再利用することで行われます。前処理キャッシュは、以下のタイプの最初の前処理ステップのみサポートされています:

- Prometheus パターン (メトリクスで入力をインデックス化)
- JSONPath(データをオブジェクトツリーに解析し、最初の式 `[?(@.path == "value")]` をインデックス化)

前処理ワーカー

Zabbix サーバーの設定ファイルでは、前処理ワーカースレッドの数を設定できます。StartPreprocessors 設定パラメータを使用して、前処理ワーカーの事前起動インスタンス数を設定します。この数は、少なくとも利用可能な CPU コア数と同じである必要があります。

前処理タスクが CPU に依存せず、頻繁にネットワークリクエストを伴う場合は、追加のワーカーを設定することを推奨します。最適な前処理ワーカー数は、「前処理可能」なアイテム数（前処理ステップの実行が必要なアイテム）、データ収集プロセス数、アイテム前処理の平均ステップ数など、多くの要因によって決まります。ワーカーが不足すると、メモリ使用量が高くなる可能性があります。Zabbix インストールでの過剰なメモリ使用量のトラブルシューティングについては、tcmalloc による過剰なメモリ使用量のプロファイリング を参照してください。

ただし、大きな XML/JSON チャンクの解析などの重い前処理操作がない場合、前処理ワーカーの数はデータ収集プロセスの合計数と一致させることができます。これにより、(データ収集プロセスからデータが一括で送信される場合を除き）収集されたデータに対して少なくとも 1 つの空いている前処理ワーカーが存在することになります。

Warning:
データ収集プロセス（ポーラー、到達不能ポーラー、ODBC ポーラー、HTTP ポーラー、Java ポーラー、ピンガー、トラッパー、プロキシポーラー）と IPMI マネージャー、SNMP トラッパー、前処理ワーカーが多すぎると、前処理マネージャーのプロセスごとのファイルディスクリプタ制限を使い果たす可能性があります。

 プロセスごとのファイルディスクリプタ制限を使い果たすと、Zabbix サーバーは停止します。通常は起動直後ですが、場合によってはそれよりも長くかかることもあります。このような問題を回避するために、Zabbix サーバーの設定ファイル を見直し、同時チェック数やプロセス数を最適化してください。さらに必要に応じて、システムの制限を確認・調整し、ファイルディスクリプタの制限が十分に高く設定されていることを確認してください。

値処理パイプライン

アイテムの値の処理は、複数のプロセスによって複数のステップ（またはフェーズ）で実行されます。これにより、以下のようなことが発生する可能性があります。

- 従属アイテムは値を受け取ることができますが、マスター値は受け取れません。これは、次のユースケースを使用することで実現できます。
  - マスターアイテムの値の型は `UINT`（トラッパーアイテムを使用可能)、従属アイテムの値の型は `TEXT`。
  - マスターアイテムと従属アイテムの両方に前処理ステップは不要です。
  - テキスト値（例: ”abc”）をマスターアイテムに渡す必要があります。
  - 実行する前処理ステップがないため、前処理マネージャはマスターアイテムが「サポートされていない」状態でないことと値が設定されていることを確認し（両方とも真)、従属アイテムをマスターアイテムと同じ値でキューに入れます（前処理ステップがないため)。
  - マスターアイテムと従属アイテムの両方が履歴同期フェーズに到達すると、マスターアイテムは値変換エラー（テキストデータを符号なし整数に変換できないため）により「サポートされていない」状態になります。

その結果、従属アイテムは値を受け取りますが、マスターアイテムは「サポートされていない」状態に変わります。

- 従属アイテムが、マスターアイテムの履歴に存在しない値を受け取る場合があります。ユースケースは前述のものと非常に似ていますが、マスターアイテムの型が異なります。たとえば、マスターアイテムに `CHAR` 型を使用した場合、マスターアイテムの値は履歴同期フェーズで切り捨てられますが、従属アイテムはマスターアイテムの初期値（切り捨てられていない値）から値を受け取ります。

3 保存前処理の例

概要

このセクションでは、保存前処理のステップを使用していくつかの実用的なタスクを実行する例を紹介します。

VMware イベントログレコードのフィルタリング

この例では、正規表現に一致の前処理ステップを使用して、VMware イベントログから不要なイベントをフィルタリングします。

1. 稼働中の VMware Hypervisor ホストで、vmware.eventlogアイテムが存在し、正常に動作していることを確認します。
 イベントログアイテムは、ホスト作成時にVMwareテンプレートがリンクされている場合、すでにハイパーバイザー上に存在している可能性があります。

2. VMware Hypervisor ホストで、*Log* タイプの従属アイテムを作成し、イベントログアイテムをマスターアイテムとして設定します。

3. 従属アイテムの前処理タブで、追加をクリックして前処理ステップを作成し、ドロップダウンから正規表現に一致を選択します。その後、以下のいずれかのパターンを指定します。

- すべてのログイベントをフィルタリングする場合:

```
.* logged in .*
```

- "User" の後にユーザー名を含む行をフィルタリングする場合:

```
\bUser\s+\K\S+
```

Attention:
正規表現に一致しない場合、従属アイテムは該当するエラーメッセージとともにサポート対象外になります。
 これを回避するには、失敗時にカスタムチェックボックスをマークし、値を破棄するかカスタム値を設定するなどのオプションを選択してください。
 破棄された値はデータベースに保存されないため、トリガーの評価やトレンドデータの生成は行われませんのでご注意ください。

また、正規表現の前処理ステップを使用して、一致するグループを抽出し、出力を制御することもできます。

- "logged in" を含むログイベント全体を抽出して出力する場合、以下のパラメータを指定します:

```
パターン: .*logged in.*
出力: \0
```

- "User" の後に続くユーザー名を抽出して出力する場合:

```
パターン: User (.*?)(?=\ )
出力: \1
```

取得した値の型のチェック

この例では、カスタム乗数の前処理ステップを使用して、取得したアイテム値の型が数値であるかどうかをチェックします。

アイテムの前処理タブで、カスタム乗数の前処理ステップを選択し、次のパラメータを指定します（取得した値に 1 を掛けます）:

```
1
```

Attention:
前処理が失敗した場合（例：入力が数値でない場合）、アイテムは対応するエラーメッセージとともにサポート対象外になります。これを回避するには、失敗時にカスタムのチェックボックスをマークし、値を破棄する、またはカスタム値を設定するなどのオプションを選択してください。破棄された値はデータベースに保存されないため、トリガーの評価やトレンドデータの生成は行われませんのでご注意ください。

サポートされていない値のチェック

この例では、サポートされていない値のチェックの前処理ステップを使用して、アイテム値が取得できなかったかどうかを確認します。

Zabbix サーバー/プロキシのポーラープロセスがアイテム値の収集を試みると、次のいずれかが発生します。

- 有効な結果を返す。
- 一見有効な結果を返すが、後でサポートされなくなる場合がある（例：前処理後に値の型が一致しない場合）。
- 値の収集エラーを返し、アイテムがサポートされなくなる。一般的な原因は次のとおりです。
  - 不明なアイテムキー（Zabbix エージェント、シンプルチェック、Zabbix 内部アイテムの場合）
  - 不明な OID（SNMP エージェント）、不明なセンサー（IPMI エージェント）、JMX メトリックなし（JMX エージェント）
  - トラップファイルが読み取れない（SNMP トラップ）
  - スクリプトが見つからない（外部チェック）
  - そのような URL が存在しない（HTTP エージェント、ブラウザー）
  - ログイン失敗（SSH エージェント、TELNET エージェント）
  - 無効な式の構文（計算アイテム）、JavaScript 構文エラー（スクリプト）、無効な SQL（データベース監視）

アイテム値の収集エラーを検出して処理するには、サポートされていない値のチェックの前処理ステップを使用できます。このステップは常に最初に実行され、前処理が開始される前に発生したエラーのみを検出することに注意してください。

アイテムの前処理タブで、サポートされていない値のチェックの前処理ステップを選択し、次のいずれかのパラメーターを指定します。

- すべてのエラーの場合:

```
パラメーター: any error
```

- "cannot connect" を含むエラーの場合:

```
パラメーター: error matches
パターン: (?i)cannot connect
```

次に、失敗時のカスタムオプションを使用して、値（この場合はエラー）を破棄したり、カスタム値を設定したり、カスタムエラーメッセージを返したりできます。破棄された値はデータベースに保存されないため、トリガーは評価されず、トレンドデータも生成されないことに注意してください。

4 JSONPath 機能

概要

このセクションでは、アイテム値の事前処理ステップでサポートされている JSONPath の機能について説明します。

JSONPath は、ドットで区切られたセグメントで構成されます。セグメントは、JSON 値名を表す単純な単語、ワイルドカード文字 (*)、または角括弧で囲まれたより複雑な構造のいずれかの形式を取ることができます。角括弧で囲まれたセグメントの前のドットは省略可能で、省略することができます。

JSONPath の例	説明
`$.object.name`	`object.name` の内容を返します。
`$.object['name']`	`object.name` の内容を返します。
`$.object.['name']`	`object.name` の内容を返します。
`$["object"]['name']`	`object.name` の内容を返します。
`$.['object'].["name"]`	`object.name` の内容を返します。
`$.object.history.length()`	`object.history` 配列の要素数を返します。
`$[?(@.name == 'Object')].price.first()`	名前が"Object" の最初のオブジェクトの `price` プロパティの値を返します。
`$[?(@.name == 'Object')].history.first().length()`	名前が"Object" の最初のオブジェクトの history 配列の要素数を返します。
`$[?(@.price > 10)].length()`	価格が 10 より大きいオブジェクトの数を返します。

参考: JSONPath での LLD マクロ値の特殊文字のエスケープ

サポートされているセグメント

セグメント	説明
`<name>`	名前でオブジェクトプロパティに一致します。
`*`	すべてのオブジェクトプロパティに一致します。
`['<name>']`	名前でオブジェクトプロパティに一致します。
`['<name>', '<name>', ...]`	リストされた名前のいずれかに一致するオブジェクトプロパティに一致します。
`[<index>]`	インデックスで配列要素に一致します。
`[<number>, <number>, ...]`	リストされたインデックスのいずれかに一致する配列要素に一致します。
`[*]`	すべてのオブジェクトプロパティまたは配列要素に一致します。
`[<start>:<end>]`	定義された範囲で配列要素に一致します: **<start>** - 一致する最初のインデックス (含む); 指定しない場合は配列の先頭からすべての配列要素に一致; 負の場合は配列の末尾からの開始オフセットを指定; **<end>** - 一致する最後のインデックス (含まない); 指定しない場合は配列の末尾までのすべての配列要素に一致; 負の場合は配列の末尾からの開始オフセットを指定。
`[?(<expression>)]`	フィルター式を適用してオブジェクト/配列要素に一致します。

祖先を無視して一致するセグメント (切り離されたセグメント) を見つけるには、2 つのドット (`..`) をプレフィックスとして付ける必要があります。たとえば、`$..name` や `$..['name']` は、すべての `name` プロパティの値を返します。

一致した要素名は、JSONPath にチルダ (~) サフィックスを追加することで抽出できます。これにより、一致したオブジェクトの名前または一致した配列アイテムのインデックス (文字列形式) が返されます。出力形式は他の JSONPath クエリと同じルールに従います - 明確なパスの結果は「そのまま」返され、不明確なパスの結果は配列で返されます。ただし、明確なパスに一致する要素の名前を抽出することには最小限の価値しかありません。なぜなら、それはすでに分かっているからです。

フィルター式

フィルター式は、中置記法の算術式です。

サポートされているオペランド:

オペランド	説明
"<text>" '<text>'	テキスト定数。 例: 'value: \\'1\\'' "value: '1'"
<number>	科学表記法をサポートする数値定数。 例: 123
<jsonpath starting with $>	入力ドキュメントのルートノードからの JSONPath で参照される値。明確なパスのみサポート。 例: $.object.name
<jsonpath starting with @>	現在のオブジェクト/要素からの JSONPath で参照される値。明確なパスのみサポート。 例: @.name

サポートされている演算子:

演算子	タイプ	説明	結果
-	二項	減算	数値
+	二項	加算	数値
/	二項	除算	数値
*	二項	乗算	数値
==	二項	等価	ブール値 (1/0)
!=	二項	非等価	ブール値 (1/0)
	二項	より小さい	ブール値 (1/0)
<=	二項	以下	ブール値 (1/0)
>	二項	より大きい	ブール値 (1/0)
>=	二項	以上	ブール値 (1/0)
=~	二項	正規表現に一致	ブール値 (1/0)
!	単項	ブール NOT	ブール値 (1/0)
||	二項	ブール OR	ブール値 (1/0)
&&	二項	ブール AND	ブール値 (1/0)

関数

関数は JSONPath の末尾で使用できます。複数の関数を連結することができ、前の関数が後続の関数で受け入れられる値を返す場合に有効です。

サポートされている関数:

関数	説明	入力	出力
avg	入力配列内の数値の平均値	数値の配列	数値
min	入力配列内の数値の最小値	数値の配列	数値
max	入力配列内の数値の最大値	数値の配列	数値
sum	入力配列内の数値の合計	数値の配列	数値
length	入力配列内の要素数	配列	数値
first	配列の最初の要素	配列	入力配列の内容に応じた JSON 構造（オブジェクト、配列、値）

JSONPath の集約関数は、引用符付きの数値も受け入れます。これらの値は、集約が必要な場合に自動的に文字列から数値型に変換されます。互換性のない入力は、関数がエラーを生成する原因となります。

出力値

JSONPath は、確定パスと不確定パスに分けることができます。確定パスは、null または単一の一致のみを返すことができます。不確定パスは、複数の一致を返すことができます。これは、分離された複数の名前/インデックスリスト、配列スライス、または式セグメントを持つ JSONPath です。ただし、関数が使用される場合、関数は常に単一の値を出力するため、JSONPath は確定パスになります。

確定パスは、参照しているオブジェクト/配列/値を返します。対照的に、不確定パスは一致したオブジェクト/配列/値の配列を返します。

> **Attention:**
> 内部の最適化手法により、JSONPath クエリ結果のプロパティ順序が元の JSON のプロパティ順序と一致しない場合があります。たとえば、JSONPath `$.books[1]["author", "title"]` は `["title", "author"]` を返す場合があります。元のプロパティ順序を保持することが重要な場合は、クエリ後の処理方法を検討してください。

パスの書式設定ルール

ブラケット表記のセグメントや式には、空白（スペース、タブ文字）を使用できます。例：`$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`。

文字列はシングルクォート（'）またはダブルクォート（"）で囲む必要があります。文字列内では、シングルクォートまたはダブルクォート（どちらで囲むかによる）およびバックスラッシュ（\）は、バックスラッシュ（\）文字でエスケープされます。

例

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Next day delivery in local town",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Printing and assembling book in A5 format",
```

```
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Various restoration methods",
      "active": false,
      "methods": [
        {
          "description": "Chemical cleaning",
          "price": 46
        },
        {
          "description": "Pressing pages damaged by moisture",
          "price": 24.5
        },
        {
          "description": "Rebinding torn book",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Store is closed",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	結果
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]

JSONPath	Type	結果
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] 注: このクエリは、クエリで算術演算が使用できることを示しています。`$.books[?(@.id == 2)].title` に簡略化できます。
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Printing and assembling book in A5 format”, ”Rebinding torn book”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour

JSONPath	Type	結果
`$..tags.first().length()`	definite	5 注: `$..tags` は不定パスなので、マッチした要素の配列、すなわち `[["a", "b", "c", "d", "e" ]]` を返します。`first()` は最初の要素、すなわち `["a", "b", "c", "d", "e"]` を返します。`length()` は要素の長さ、すなわち 5 を計算します。
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	注: マッチしないクエリは、定義済み・不定パスともに NULL を返します。
`$.services[?(@.active=="true")].servicegroup`	indefinite	`[1001,1000]` 注: ブール値の比較にはテキスト定数を使用する必要があります。
`$.services[?(@.active=="false")].servicegroup`	indefinite	`[1002]` 注: ブール値の比較にはテキスト定数を使用する必要があります。
`$.services[?(@.servicegroup==1002)]~.first()`	definite	`"f002"`

1 JSONPath の LLD マクロ値からの特殊文字のエスケープ

ローレベルディスカバリマクロが JSONPath の保存前処理で使用され、その値が解決される場合、次の特殊文字のエスケープ規則が適用されます。

- バックスラッシュ (\) と二重引用符 (") 文字のみがエスケープ対象として考慮されます。
- 解決されたマクロ値にこれらの文字が含まれている場合、それぞれの文字はバックスラッシュでエスケープされます。
- すでにバックスラッシュでエスケープされている場合は、エスケープとみなされず、バックスラッシュとそれに続く特殊文字の両方がさらにエスケープされます。

例:

JSONPath	LLD マクロの値	置換後
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`
	c:\temp	`$.[?(@.value == "c:\\temp")]`
	a\\b	`$.[?(@.value == "a\\\\b")]`

式で使用する場合、特殊文字を含む可能性のあるマクロを二重引用符で囲む必要があります。

JSONPath	LLD マクロの値	置換後	結果
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`	OK
`$.[?(@.value == {#MACRO})]`		`$.[?(@.value == special \"value\")]`	不正な **JSONPath** の式

特殊文字を含むマクロをパスで使用する場合は、角括弧と二重引用符で囲む必要があります。

JSONPath	LLD マクロの値	置換後	結果
`$.["{#MACRO}"].value`	c:\temp	`$.["c:\\temp"].value`	OK
`$.{#MACRO}.value`		`$.c:\\temp.value`	不正な **JSONPath** の式

5 JavaScript の保存前処理

概要

このセクションでは、JavaScript による前処理の詳細について説明します。

> **Attention:**
> 前処理の JavaScript で宣言されていない代入を使用しないでください。ローカル変数を宣言するには var を使用してください。

JavaScript の事前処理

JavaScript の事前処理は、単一のパラメータ'value' とユーザーが指定した関数本体を持つ JavaScript 関数を呼び出すことで行われます。事前処理ステップの結果は、この関数が返す値です。例えば、華氏から摂氏への変換を行うには、次のように入力します。

```
return (value - 32) * 5 / 9
```

この内容を JavaScript の事前処理パラメータに入力すると、サーバーによって次のような JavaScript 関数にラップされます。

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

入力パラメータ'value' は常に文字列として渡されます。戻り値は自動的に toString() メソッドで文字列に変換されます（失敗した場合はエラーが文字列値として返されます）。ただし、いくつか例外があります。

- undefined 値を返すとエラーになります。
- null 値を返すと、入力値が破棄されます。これは「失敗時にカスタム処理」で「値を破棄」と同様の動作です。

エラーは値やオブジェクト（通常は文字列または Error オブジェクト）を throw することで返すことができます。

例：

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

各スクリプトの実行タイムアウトは 10 秒です（スクリプトによってはタイムアウトが発生するまでにさらに時間がかかる場合があります）。これを超えるとエラーが返されます。512 メガバイトのヒープ制限が適用されます。

JavaScript の事前処理ステップのバイトコードはキャッシュされ、次回ステップが適用される際に再利用されます。アイテムの事前処理ステップに変更があると、キャッシュされたスクリプトはリセットされ、後で再コンパイルされます。

実行時エラーが連続して発生した場合（3 回連続）、1 つのスクリプトが次のスクリプトの実行環境を壊す可能性を軽減するためにエンジンが再初期化されます（このアクションは DebugLevel 4 以上でログに記録されます）。

JavaScript の事前処理はDuktape JavaScript エンジンで実装されています。

参考: 追加の JavaScript オブジェクトとグローバル関数

スクリプトでのマクロの使用

JavaScript コードでユーザーマクロを使用することができます。スクリプトにユーザーマクロが含まれている場合、これらのマクロは特定の事前処理ステップを実行する前にサーバー/プロキシによって解決されます。フロントエンドで事前処理ステップをテストする際には、マクロ値は取得されず、手動で入力する必要があることに注意してください。

> **Note:**
> マクロが値に置き換えられる際、コンテキストは無視されます。マクロ値はそのままコードに挿入され、JavaScript コードに値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることにご注意ください。

以下の例では、受信した値が {$THRESHOLD} マクロ値を超えた場合、しきい値（存在する場合）が返されます:

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

例

次の例は、JavaScript の保存前処理の使用方法を示しています。各例には、簡単な説明、JavaScript 保存前処理パラメーターの関数本体、および保存前処理のステップの結果 (関数によって返される値) が含まれています。

例 1: 数値の変換 (科学的記数法から整数へ)

数値"2.62128e+07" を科学的記数法から整数に変換します。

```
return (Number(value))
```

関数によって返される値: 26212800

例 2: 数値の変換 (2 進数から 10 進数へ)

2 進数"11010010" を 10 進数に変換します。

```
return(parseInt(value,2))
```

関数によって返される値: 210

例 3: 数値を丸める

数値"18.2345" を 2 桁に丸めます。

```
return(Math.round(value* 100) / 100)
```

関数によって返される値: 18.23

例 4: 文字列内の文字数を数える

文字列"Zabbix" 内の文字数を数えます。

```
return (value.length)
```

関数によって返される値: 6

例 5: 残り時間を取得する

証明書の有効期限 (Feb 12 12:33:56 2022 GMT) までの残り時間 (秒単位) を取得します。

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();

return parseInt((Date.parse(ISOdate) - now) / 1000);
```

関数によって返される値: 44380233

例 6: JSON プロパティの削除

キー"data_size" または"index_size" を持つプロパティを削除して、JSON データ構造を変更します。

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

関数によって受け入れられる値:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

関数によって返される値:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

例 7: Apache ステータスを JSON に変換

web.page.get Zabbix エージェントアイテム (例: web.page.get[http://127.0.0.1:80/server-status?auto]) から受信した値を JSON オブジェクトに変換します。

```
// Apache ステータスを JSON に変換

// 値をサブストリングに分割し、これらのサブストリングを配列に格納
var lines = value.split('\n');

// 空のオブジェクト "output" を作成
var output = {};

// 事前定義されたプロパティを持つ "workers" オブジェクトを作成
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// "lines" 配列から "output" オブジェクトにプロパティ（キーと値のペア）として追加
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);

    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

// 複数バージョンのメトリクス
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// "Scoreboard" プロパティを解析してワーカー数を取得
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// ワーカーデータを "output" オブジェクトに追加
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// JSON 文字列を返す
return JSON.stringify(output);
```

関数で受け入れられる値:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 11:08:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 405
Content-Type: text/plain; charset=ISO-8859-1

127.0.0.1
ServerVersion: Apache/2.4.52 (Ubuntu)
ServerMPM: prefork
Server Built: 2023-03-08T17:32:01
CurrentTime: Monday, 27-Mar-2023 14:08:39 EEST
RestartTime: Monday, 27-Mar-2023 12:19:59 EEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 6520
ServerUptime: 1 hour 48 minutes 40 seconds
Load1: 0.56
Load5: 0.33
Load15: 0.28
Total Accesses: 2476
Total kBytes: 8370
Total Duration: 52718
CPUUser: 8.16
CPUSystem: 3.44
CPUChildrenUser: 0
CPUChildrenSystem: 0
CPULoad: .177914
Uptime: 6520
ReqPerSec: .379755
BytesPerSec: 3461.58
BytesPerReq: 3461.58
DurationPerReq: 21.2916
BusyWorkers: 2
IdleWorkers: 6
Scoreboard: ____KW__.....................................................................................
```

関数から返される値:

```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
```

```
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................
    "Workers": {
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
        "cleanup": 0,
        "slot": 142
    }
}
```

1 追加の JavaScript オブジェクト

概要

このセクションでは、Duktape で実装された JavaScript 言語への Zabbix の追加機能と、サポートされているグローバル JavaScript 関数について説明します。

> **Attention:**
> プリプロセスの JavaScript では、宣言されていない代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

組込オブジェクト

Zabbix

Zabbix オブジェクトは、Zabbix の内部機能との連携を提供します。

メソッド	説明
`log(loglevel, message)`	<loglevel> のログレベル（設定ファイルの DebugLevel パラメータを参照）で <message> を Zabbix ログに書き込みます。

例:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

以下のエイリアスを使用できます:

エイリアス	エイリアス先
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

Attention:
スクリプト実行ごとに記録されるすべてのメッセージの合計サイズは 8MB に制限されています。

メソッド	説明
sleep(delay)	JavaScript の実行を delay ミリ秒遅延させます。

例（15 秒遅延）：

```
Zabbix.sleep(15000)
```

HttpRequest

このオブジェクトは cURL ハンドルをカプセル化し、シンプルな HTTP リクエストを実行できるようにします。エラーは例外としてスローされます。

Attention:
複数の HttpRequest オブジェクトの初期化は、スクリプト実行ごとに最大 10 個に制限されています。

メソッド	説明
addHeader(value)	HTTP ヘッダーフィールドを追加します。このフィールドは、clearHeader() メソッドでクリアされるまで、以降のすべてのリクエストで使用されます。 1 つの HttpRequest オブジェクトに追加できるヘッダーフィールドの合計長は 128K バイトに制限されています（特殊文字およびヘッダー名を含む）。
clearHeader()	HTTP ヘッダーをクリアします。ヘッダーフィールドが設定されていない場合、HttpRequest は、投稿されるデータが JSON 形式の場合は Content-Type を application/json に、それ以外の場合は text/plain に設定します。
connect(url)	URL に HTTP CONNECT リクエストを送信し、レスポンスを返します。
customRequest(method, url, data)	最初のパラメータで任意の HTTP メソッドを指定できます。URL にメソッドリクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
delete(url, data)	URL に HTTP DELETE リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
getHeaders(<asArray>)	受信した HTTP ヘッダーフィールドのオブジェクトを返します。 asArray パラメータは”true”（例：getHeaders(true)）、”false”、または未定義に設定できます。”true” に設定すると、受信した HTTP ヘッダーフィールド値は配列として返されます。これは、同じ名前の複数のヘッダーのフィールド値を取得する場合に使用します。 未設定または”false” に設定した場合、受信した HTTP ヘッダーフィールド値は文字列として返されます。
get(url, data)	URL に HTTP GET リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
head(url)	URL に HTTP HEAD リクエストを送信し、レスポンスを返します。
options(url)	URL に HTTP OPTIONS リクエストを送信し、レスポンスを返します。
patch(url, data)	URL に HTTP PATCH リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
put(url, data)	URL に HTTP PUT リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
post(url, data)	URL に HTTP POST リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。
getStatus()	直前の HTTP リクエストのステータスコードを返します。
setProxy(proxy)	HTTP プロキシを”proxy” 値に設定します。このパラメータが空の場合、プロキシは使用されません。
setHttpAuth(bitmask, username, password)	有効な HTTP 認証方式（HTTPAUTH_BASIC、HTTPAUTH_DIGEST、HTTPAUTH_NEGOTIATE、HTTPAUTH_NTLM、HTTPAUTH_NONE）を’bitmask’ パラメータで設定します。 HTTPAUTH_NONE フラグは HTTP 認証を無効にします。 例: request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password) request.setHttpAuth(HTTPAUTH_NONE)
trace(url, data)	URL に HTTP TRACE リクエストをオプションの *data* ペイロードとともに送信し、レスポンスを返します。

例:

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

XML オブジェクトは、アイテムおよびローレベルディスカバリの前処理や Webhook で XML データの処理を可能にします。

Attention:
XML オブジェクトを使用するには、サーバー/プロキシを libxml2 サポート付きでコンパイルする必要があります。

メソッド	説明
XML.query(data, expression)	XPath を使用してノードの内容を取得します。ノードが見つからない場合は null を返します。 **expression** - XPath 式; **data** - 文字列としての XML データ。
XML.toJson(data)	XML 形式のデータを JSON に変換します。
XML.fromJson(object)	JSON 形式のデータを XML に変換します。

例:

入力:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
```

```
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

出力:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

シリアライズルール

XML から JSON への変換は、以下のルールに従って処理されます（JSON から XML への変換では逆のルールが適用されます）。

1. XML 属性は、名前の前に「@」が付いたキーに変換されます。

例:

入力:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

出力:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. セルフクローズ要素（<foo/>）は「null」値として変換されます。

例:

入力:

```
<xml>
  <foo/>
</xml>
```

出力:

```
{
  "xml": {
    "foo": null
  }
}
```

3. 空の属性（値が""）は空文字列（''）値として変換されます。

例:

入力:

```xml
<xml>
  <foo bar="" />
</xml>
```

出力:

```json
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. 同じ要素名の複数の子ノードは、値の配列を持つ単一のキーに変換されます。

例:

入力:

```xml
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. テキスト要素に属性や子要素がない場合、文字列として変換されます。

例:

入力:

```xml
<xml>
    <foo>BAZ</foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. テキスト要素に子要素はないが属性がある場合、テキスト内容は「#text」キーと値を持つ要素に変換され、属性はシリアライズルール 1 で説明したように変換されます。

例:

入力:

```xml
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
```

```
    }
  }
}
```

グローバル JavaScript 関数

Duktape で追加のグローバル JavaScript 関数が実装されています:

- btoa(data) - データを base64 文字列にエンコードします。
- atob(base64_string) - base64 文字列を Uint8Array バッファとしてデコードします。

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // デコードロジックはバッファのデータフォーマットに依存することに注意してください。
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- md5(data) - データの MD5 ハッシュを計算します
- sha256(data) - データの SHA256 ハッシュを計算します
- hmac('<hash type>',key,data) - HMAC ハッシュを 16 進数形式の文字列で返します。MD5 および SHA256 ハッシュタイプがサポートされています。key および data パラメータはバイナリデータをサポートします。例:
  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`
- sign(hash,key,data) - 計算された署名（SHA-256 による RSA 署名）を文字列として返します。ここで:
 hash - 'sha256' のみ許可され、それ以外はエラーがスローされます。
 key - 秘密鍵。PKCS#1 または PKCS#8 標準に準拠している必要があります。鍵はさまざまな形式で指定できます:
  - 改行の代わりにスペースを使用;
  - 改行の代わりにエスケープされたまたは非エスケープされた`\n` を使用;
  - 改行なしの 1 行の文字列として;
  - JSON 形式の文字列として。

  鍵はユーザーマクロ/シークレットマクロ/ボールトからロードすることもできます。

  data - 署名されるデータ。文字列（バイナリデータもサポート）またはバッファ（Uint8Array/ArrayBuffer）で指定できます。
 署名の計算には OpenSSL または GnuTLS が使用されます。これらの暗号化ライブラリのいずれも組み込まれていない場合、エラーがスローされます（'missing OpenSSL or GnuTLS library'）。

2 ブラウザアイテムの JavaScript オブジェクト

概要

このセクションでは、ブラウザーアイテムスクリプトで使用するために Duktape で実装された JavaScript 言語への Zabbix の追加機能について説明します。これらの追加機能は、追加の JavaScript オブジェクトページで説明されている JavaScript オブジェクトを補完するものです。

Browser

`Browser` オブジェクトは WebDriver セッションを管理し、作成時にセッションを初期化し、破棄時にセッションを終了します。1 つのスクリプトで最大 4 つの `Browser` オブジェクトをサポートできます。

`Browser` オブジェクトを構築するには、`new Browser(options)` 構文を使用します。`options` (*JSON* オブジェクト）パラメータはブラウザオプションを指定し、通常は WebDriver のオプションメソッドの結果（例：`Browser.chromeOptions()`）です。

`Browser` オブジェクトでサポートされているメソッドは以下の通りです。

メソッド	説明
navigate(url)	指定した URL に移動します。 パラメータ: url - (string) 移動する URL。
getUrl()	開いているページの URL の文字列を返します。
getPageSource()	開いているページのソースの文字列を返します。
findElement(strategy, selector)	開いているページ内の 1 つの要素を持つElementオブジェクトを返します (strategy と selector に一致する要素がない場合は null を返します)。 パラメータ: strategy - (string, CSS セレクタ/リンクテキスト/部分リンクテキスト/タグ名/Xpath) ロケーションストラテジー; selector - (string) 指定したロケーションストラテジーを使用した要素セレクタ。
findElements(strategy, target)	開いているページ内の複数の要素を持つElementオブジェクトの配列を返します（ロケーションストラテジーとターゲットに一致する要素がない場合は空配列を返します)。 パラメータ: strategy - (string, CSS セレクタ/リンクテキスト/部分リンクテキスト/タグ名/Xpath) ロケーションストラテジー; target - (string) 指定したロケーションストラテジーを使用した要素セレクタ。
getCookies()	Cookieオブジェクトの配列を返します。
addCookie(cookie)	Cookie を設定します。 パラメータ: cookie - (Cookieオブジェクト) 設定する Cookie。
getScreenshot()	ブラウザのビューポート画像を表す base64 エンコード文字列を返します。
setScreenSize(x,y)	ブラウザのビューポートサイズを設定します。 パラメータ: x - (string) ピクセル単位の高さ; y - (string) ピクセル単位の幅。
setScriptTimeout(timeout)	スクリプトの読み込みタイムアウトを設定します。 パラメータ: timeout - (integer) タイムアウト値（ミリ秒)。
setSessionTimeout(timeout)	セッション（ページロード）のタイムアウトを設定します。 パラメータ: timeout - (integer) タイムアウト値（ミリ秒)。
setElementWaitTimeout(timeout)	要素ロケーションストラテジー（暗黙的）タイムアウトを設定します。 パラメータ: timeout - (integer) タイムアウト値（ミリ秒)。
collectPerfEntries(mark)	getResult() メソッドで取得するためのパフォーマンスエントリを収集します。 パラメータ: mark - (string, オプション) パフォーマンススナップショットマーク。
getRawPerfEntries()	パフォーマンスエントリオブジェクトの配列を返します。
getResult()	ブラウザセッションの統計情報（エラー情報、パフォーマンススナップショットなど）を持つResultオブジェクトを返します。
getError()	ブラウザエラーを持つ BrowserError オブジェクトを返します（ブラウザエラーがない場合は null を返します)。
setError(message)	Resultオブジェクトに含めるカスタムエラーメッセージを設定します。 パラメータ: message - (string) エラーメッセージ。
discardError()	Resultオブジェクトで返すエラーを破棄します。
getAlert()	ブラウザアラートを持つAlertオブジェクトを返します（ブラウザアラートがない場合は null を返します)。
chromeOptions()	事前定義された Chrome ブラウザオプションを持つ chromeOptions オブジェクトを返します。

メソッド	説明
`firefoxOptions()`	事前定義された Firefox ブラウザオプションを持つ `firefoxOptions` オブジェクトを返します。
`safariOptions()`	事前定義された Safari ブラウザオプションを持つ `safariOptions` オブジェクトを返します。
`edgeOptions()`	事前定義された Edge ブラウザオプションを持つ `edgeOptions` オブジェクトを返します。
`switchFrame(target)`	指定したフレームに切り替えます。 パラメータ: `target` - (ブラウザ要素または整数, オプション) 対象フレーム。要素でフレームを選択する場合は要素を渡します。インデックスでフレームを選択する場合は数値を渡します。空の場合は最上位のブラウジングコンテキストに切り替えます。

すべての Browser メソッドは以下のエラーをスローする可能性があります:

- `BrowserError` - `Browser` コンストラクタが失敗した場合にスローされる `Error` オブジェクトから派生したもの; この `BrowserError` をスローした `Browser` オブジェクトを持つ追加の `browser` プロパティを含みます。
- `WebdriverError` - `BrowserError` から派生したもの; `BrowserError` オブジェクトと同じプロパティを含み、エラーが WebDriver レスポンスのエラーに応答して生成されたかどうかを示します。

Element

`Element` オブジェクトは、`Browser` オブジェクトの `findElement()`/`findElements()` メソッドによって返され、直接構築することはできません。

`Element` オブジェクトは、Web ページ内の要素を表し、それに対して操作を行うためのメソッドを提供します。

`Element` オブジェクトでサポートされているメソッドは以下の通りです。

メソッド	説明
`getAttribute(name)`	要素属性の属性値文字列を返します（指定された属性が見つからなかった場合は `null` を返します）。 パラメータ: `name` - (string) 属性名。
`getProperty(name)`	要素プロパティのプロパティ値文字列を返します（指定されたプロパティが見つからなかった場合は `null` を返します）。 パラメータ: `name` - (string) プロパティ名。
`getText()`	要素テキストのテキスト値文字列を返します。
`click()`	要素をクリックします。
`clear()`	編集可能な要素の内容をクリアします。
`sendKeys(keys)`	キーを送信します。 パラメータ: `keys` - (string) 送信するキー。

Cookie

`Cookie` オブジェクトは、`Browser`オブジェクトの `getCookies()` メソッドによって返され、`addCookie()` メソッドに渡されます。

`Cookie` オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

プロパティ	型	説明
name	string	Cookie の名前
value	string	Cookie の値
path	string	Cookie が有効なパス Cookie 追加時に省略した場合は"/" がデフォルト
domain	string	Cookie が表示されるドメイン Cookie 追加時に省略した場合は、セッションの現在のブラウジングコンテキストのアクティブドキュメントの URL ドメインがデフォルト

プロパティ	型	説明
secure	boolean	Cookie がセキュアかどうか Cookie 追加時に省略した場合は `false` がデフォルト
httpOnly	boolean	Cookie が HTTP 専用かどうか Cookie 追加時に省略した場合は `false` がデフォルト
expiry	integer	Cookie の有効期限（Unix エポックからの秒数） Cookie 追加時に省略した場合は設定不可
sameSite	string	Cookie の `sameSite` 属性。Cookie をファーストパーティまたは同一サイトコンテキストに制限するかどうかを制御します。 `"Lax"` または`"Strict"` のいずれかを設定可能 Cookie 追加時に省略した場合は`"None"` がデフォルト

アラート

`Alert` オブジェクトはウェブページのアラートを表し、`Browser` オブジェクトの `getAlert()` メソッドによって返され、直接構築することはできません。

`Alert` オブジェクトには、アラートのテキスト（またはアラートがない場合は `null`）を持つ `text` プロパティが含まれています。

`Alert` オブジェクトでは、以下のメソッドがサポートされています。

メソッド	説明
`accept()`	アラートを受け入れます。
`dismiss()`	アラートを閉じます。

Result

`Result` オブジェクトにはセッション統計が含まれており、`Browser` オブジェクトの `getResult()` メソッドによって返されます。

通常、`Result` オブジェクトは文字列化されてスクリプトから返され、前処理を通じて依存アイテム値にパースされます。

`Result` オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

プロパティ			型	説明
duration			string	セッション作成から結果取得までのセッション継続時間。
error			object	エラー情報。
	http_status		integer	WebDriver から返された HTTP ステータス（WebDriver エラーがない場合
	error_code		string	WebDriver から返されたエラー（WebDriver エラーがない場合は空文字列
	message		string	WebDriver エラーメッセージ（WebDriver エラーがない場合は空文字列）。
performance_data			object	パフォーマンス統計。
	summary		object	パフォーマンスサマリー。
		navigation	object	ナビゲーションサマリー。
		resource	object	リソースサマリー。
	details		array of objects	ナビゲーションにつながる可能性のある各操作後のパフォーマンス統計。
		mark	string	*(オプション)* `collectPerfEntries()` メソッドで指定されたパフォーマ
		navigation	object	ナビゲーション統計。
		resource	object	このステップのリソースサマリー。
		user	array of objects	mark/measure タイプの統計の配列。
	marks		array of objects	マークされたパフォーマンススナップショットインデックス。
		name	string	パフォーマンススナップショットマーク名。
		index	integer	details 配列内のパフォーマンススナップショットインデックス。

6 CSV から **JSON** への保存前処理

概要

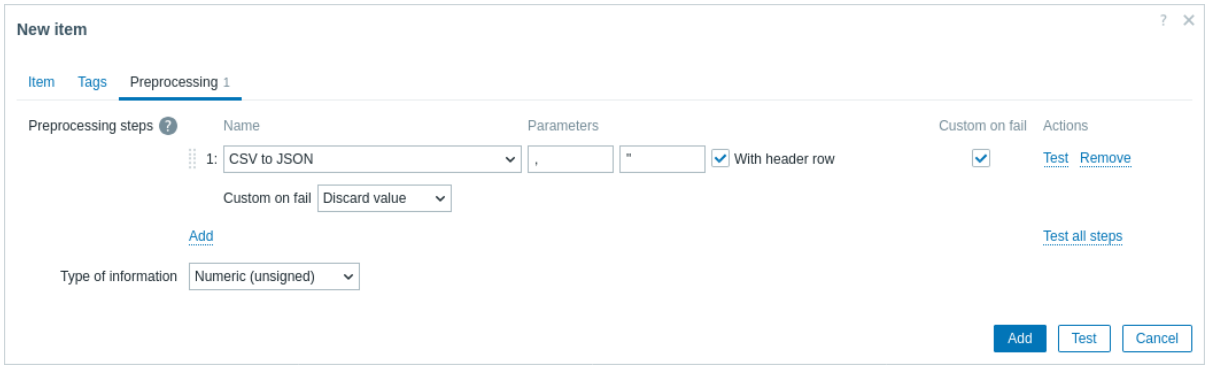
この保存前処理で、CSV ファイルデータを JSON 形式に変換することが可能です。サポートしているのは、

- アイテム (アイテムのプロトタイプ)
- ローレベルディスカバリルール

設定

CSV から JSON への保存前処理のステップを設定する場合:

- アイテム/ディスカバリルールの保存前処理タブにアクセスします。
- 追加をクリックします。
- *CSV から JSON* オプションを選択します。



1 番目のパラメーターカスタムデリミタを設定することができます。CSV 入力の最初の行が”Sep=” で始まり、UTF-8 文字が 1 つ続く場合、最初のパラメーターが設定されていない場合、その文字がデリミタとして使用されることに注意ください。1 番目のパラメーターが設定されておらず、かつ”Sep=” 行からデリミタが取得できない場合は、カンマがセパレーターとして使用されます。

オプションの 2 番目のパラメーターには引用記号を設定することができます。

ヘッダ行を含むのチェックボックスがチェックされている場合、ヘッダー行の値は列名として解釈されます (詳しくはヘッダー処理を参照してください)。

失敗時のカスタマイズのチェックボックスがチェックされている場合、保存前処理のステップが失敗してもそのアイテムがサポートされなくなることはありません。さらに、カスタムエラー処理オプションを設定することができます: 値を破棄、値を設定、エラーを設定。

ヘッダー処理

CSV ファイルのヘッダー行は、2 種類の方法で処理することができます。

- ヘッダ行を含むチェックボックスがチェックされている場合 - ヘッダー行の値はカラム名として解釈されます。この場合、カラム名は一意でなければならず、データ行はヘッダー行より多くのカラムを含んではなりません。
- ヘッダ行を含むチェックボックスがチェックされていない場合 - ヘッダー行はデータとして解釈されます。カラム名は自動的に生成されます (1,2,3,4...)。

CSV ファイルの例:

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:
入力の引用フィールド内の引用文字は、その前に別の引用文字を置くことでエスケープしなければいけません。

ヘッダー行の処理

ヘッダー行が検出される場合の JSON 出力:

```
[
   {
      "Nr":"1",
      "Item name":"active agent item",
      "Key":"agent.hostname",
      "Qty":"33"
   },
   {
      "Nr":"2",
      "Item name":"passive agent item",
      "Key":"agent.version",
      "Qty":"44"
   },
   {
      "Nr":"3",
      "Item name":"active,passive agent items",
```

```
        "Key":"agent.ping",
        "Qty":"55"
    }
]
```

ヘッダー行の処理なし

ヘッダー行が検出できない場合の JSON 出力:

```
[
    {
        "1":"Nr",
        "2":"Item name",
        "3":"Key"
        "4":"Qty"
    },
    {
        "1":"1",
        "2":"active agent item",
        "3":"agent.hostname"
        "4":"33"
    },
    {
        "1":"2",
        "2":"passive agent item",
        "3":"agent.version"
        "4":"44"
    },
    {
        "1":"3",
        "2":"active,passive agent items",
        "3":"agent.ping"
        "4":"55"
    }
]
```

3 アイテムのタイプ

概要

アイテムタイプは、システムからデータを取得するさまざまな方法をカバーしています。各アイテムタイプには、サポートされているアイテムキーと必要なパラメータのセットがあります。

現在、Zabbix で提供されているアイテムタイプは以下の通りです。

- Zabbix エージェントチェック
- SNMP エージェントチェック
- SNMP トラップ
- IPMI チェック
- シンプルチェック
    - VMware 監視
- ログファイル監視
- 計算アイテム
    - 集約計算
- Zabbix 内部チェック
- SSH チェック
- Telnet チェック
- 外部チェック
- トラッパーアイテム
- JMX 監視
- ODBC チェック
- 従属アイテム
- HTTP チェック
- Prometheus チェック
- スクリプトアイテム

- ブラウザーアイテム

すべてのアイテムタイプの詳細は、このセクションのサブページに記載されています。アイテムタイプはデータ収集のための多くのオプションを提供していますが、ユーザーパラメータやロード可能モジュールによってさらに多くのオプションが利用可能です。

一部のチェックは Zabbix サーバー単体で実行されます（エージェントレス監視として）、他のチェックは Zabbix エージェントや Zabbix Java ゲートウェイ（JMX 監視時）が必要です。

Attention:
特定のアイテムタイプが特定のインターフェースを必要とする場合（たとえば、IPMI チェックにはホストに IPMI インターフェースが必要）、そのインターフェースはホスト定義に存在している必要があります。

ホスト定義には複数のインターフェース（Zabbix エージェント、SNMP エージェント、JMX、IPMI）を設定できます。アイテムが複数のインターフェースを使用できる場合、利用可能なホストインターフェース（順序：エージェント →SNMP→JMX→IPMI）を検索し、最初に適切なものにリンクします。

テキスト（文字列、ログ、テキスト型情報）を返すすべてのアイテムは、空白のみを返すこともできます（該当する場合）、戻り値を空文字列に設定します（2.0 以降でサポート）。

1 Zabbix エージェント

概要

このセクションでは、データ収集のために Zabbix エージェントとの通信を使用するアイテムキーの詳細について説明します。

パッシブチェックとアクティブチェックがあります。アイテムを設定する際に、必要なタイプを選択できます:

- *Zabbix* エージェント - パッシブチェック用
- *Zabbix* エージェント（アクティブ） - アクティブチェック用

Zabbix エージェントでサポートされているすべてのアイテムキーは、新世代の Zabbix エージェント 2 でもサポートされていることに注意してください。エージェント 2 でのみ使用できる追加のアイテムキーも参照してください。

サポートされているアイテムキー

Zabbix エージェントで使用できるアイテムキーを以下に示します。

アイテムキーはパラメータや追加情報なしでリストされています。
 アイテムキーをクリックすると、詳細が表示されます。

アイテムキー	説明	アイテムグループ
kernel.maxfiles	OS がサポートする最大オープンファイル数。	カーネル
kernel.maxproc	OS がサポートする最大プロセス数。	
kernel.openfiles	現在開いているファイルディスクリプタ数。	
log	ログファイルの監視。	ログ監視
log.count	監視対象のログファイルで一致した行数。	
logrt	ローテートされるログファイルの監視。	
logrt.count	ローテートされる監視対象のログファイルで一致した行数。	
modbus.get	Modbus データの読み取り。	Modbus
net.dns	DNS サービスの状態をチェック。	ネットワーク
net.dns.perf	DNS サービスのパフォーマンスをチェック。	
net.dns.record	DNS クエリを実行。	
net.if.collisions	ウィンドウ外コリジョン数。	
net.if.discovery	ネットワークインターフェースのリスト。	
net.if.in	ネットワークインターフェースの受信トラフィック統計。	
net.if.out	ネットワークインターフェースの送信トラフィック統計。	
net.if.total	ネットワークインターフェースの受信・送信トラフィック統計の合計。	
net.tcp.listen	この TCP ポートが LISTEN 状態かどうかをチェック。	
net.tcp.port	指定したポートに TCP 接続できるかどうかをチェック。	
net.tcp.service	サービスが稼働し、TCP 接続を受け付けているかどうかをチェック。	
net.tcp.service.perf	TCP サービスのパフォーマンスをチェック。	
net.tcp.socket.count	パラメータに一致する TCP ソケット数を返す。	
net.udp.listen	この UDP ポートが LISTEN 状態かどうかをチェック。	
net.udp.service	サービスが稼働し、UDP リクエストに応答しているかどうかをチェック。	
net.udp.service.perf	UDP サービスのパフォーマンスをチェック。	
net.udp.socket.count	パラメータに一致する UDP ソケット数を返す。	

アイテムキー	説明	アイテムグループ
proc.cpu.util	プロセスの CPU 使用率（%）。	プロセス
proc.get	OS プロセスとそのパラメータのリスト。	
proc.mem	プロセスが使用するメモリ（バイト単位）。	
proc.num	プロセス数。	
sensor	ハードウェアセンサーの値。	センサー
system.boottime	システムの起動時刻。	システム
system.cpu.discovery	検出された CPU/CPU コアのリスト。	
system.cpu.intr	デバイス割り込み数。	
system.cpu.load	CPU 負荷。	
system.cpu.num	CPU 数。	
system.cpu.switches	コンテキストスイッチ数。	
system.cpu.util	CPU 使用率（%）。	
system.hostname	システムのホスト名。	
system.hw.chassis	シャーシ情報。	
system.hw.cpu	CPU 情報。	
system.hw.devices	PCI または USB デバイスのリスト。	
system.hw.macaddr	MAC アドレスのリスト。	
system.localtime	システム時刻。	
system.run	ホスト上で指定したコマンドを実行。	
system.stat	システム統計情報。	
system.sw.arch	ソフトウェアアーキテクチャ情報。	
system.sw.os	オペレーティングシステム情報。	
system.sw.os.get	オペレーティングシステムの詳細情報（バージョン、タイプ、ディストリビューション名、メジャー・マイナーバージョンなど）。	
system.sw.packages	インストール済みパッケージのリスト。	
system.sw.packages.get	インストール済みパッケージの詳細リスト。	
system.swap.in	スワップイン（デバイス → メモリ）の統計。	
system.swap.out	スワップアウト（メモリ → デバイス）の統計。	
system.swap.size	スワップ領域のサイズ（バイト単位または全体に対する割合）。	
system.uname	システムの識別。	
system.uptime	システムの稼働時間（秒単位）。	
system.users.num	ログインしているユーザー数。	
vfs.dev.discovery	ブロックデバイスとそのタイプのリスト。	仮想ファイルシステム
vfs.dev.read	ディスクリード統計。	
vfs.dev.write	ディスクライト統計。	
vfs.dir.count	ディレクトリエントリ数。	
vfs.dir.get	ディレクトリエントリのリスト。	
vfs.dir.size	ディレクトリサイズ。	
vfs.file.cksum	UNIX cksum アルゴリズムで計算したファイルのチェックサム。	
vfs.file.contents	ファイルの内容の取得。	
vfs.file.exists	ファイルの存在チェック。	
vfs.file.get	ファイルに関する情報を返す。	
vfs.file.md5sum	ファイルの MD5 チェックサム。	
vfs.file.owner	ファイルの所有者を取得。	
vfs.file.permissions	UNIX パーミッションの 8 進数を含む 4 桁の文字列を返す。	
vfs.file.regexp	ファイル内の文字列を取得。	
vfs.file.regmatch	ファイル内の文字列を検索。	
vfs.file.size	ファイルサイズ。	
vfs.file.time	ファイルの時刻情報。	
vfs.fs.discovery	タイプやマウントオプション付きのマウント済みファイルシステムのリスト。	
vfs.fs.get	タイプ、空きディスク容量、inode 統計、マウントオプション付きのマウント済みファイルシステムのリスト。	
vfs.fs.inode	inode 数または割合。	
vfs.fs.size	ディスク容量（バイト単位または全体に対する割合）。	
vm.memory.size	メモリ容量（バイト単位または全体に対する割合）。	仮想メモリ
web.page.get	Web ページの内容を取得。	Web 監視
web.page.perf	Web ページ全体のロード時間。	
web.page.regexp	Web ページ内の文字列検索。	
agent.hostmetadata	エージェントのホストメタデータ。	Zabbix
agent.hostname	エージェントのホスト名。	

アイテムキー	説明	アイテムグループ
agent.ping	エージェントの可用性チェック。	
agent.variant	Zabbix エージェントのバリアント（Zabbix agent または Zabbix agent 2）。	
agent.version	Zabbix エージェントのバージョン。	
zabbix.stats	Zabbix サーバーまたはプロキシの内部メトリクスをリモートで返す。	
zabbix.stats	Zabbix サーバーまたはプロキシで遅延しているキュー内の監視アイテム数をリモートで返す。	

サポートされているプラットフォーム

アイテムの詳細で特に指定がない限り、エージェントアイテム（およびすべてのパラメータ）は以下でサポートされています:

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

多くのエージェントアイテムは **Windows** でもサポートされています。詳細はWindows エージェントアイテムのページを参照してください。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

kernel.maxfiles

 OS がサポートする最大オープンファイル数。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD。

kernel.maxproc

 OS がサポートする最大プロセス数。
 戻り値: 整数。
 対応プラットフォーム: Linux 2.6 以降、FreeBSD、Solaris、MacOS X、OpenBSD、NetBSD。

kernel.openfiles

 現在開いているファイルディスクリプタの数。
 戻り値：整数。
 対応プラットフォーム：Linux（他の UNIX 系プラットフォームでも動作する場合があります）。

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ログファイルの監視。
 戻り値: *Log*。
 サポートされているプラットフォームを参照。

パラメータ:

- **file** - ログファイルのフルパスとファイル名;
- **regexp** - 必要なパターンを記述する正規表現;
- **encoding** - コードページの識別子;
- **maxlines** - エージェントが Zabbix サーバーまたはプロキシに 1 秒あたり送信する新しい行の最大数。このパラメータはzabbix_agentd.confの'MaxLinesPerSecond' の値を上書きします;
- **mode** - 可能な値: *all* (デフォルト) または *skip* - 古いデータの処理をスキップ（新しく作成されたアイテムのみに影響）;
- **output** - オプションの出力フォーマットテンプレート。**\0** エスケープシーケンスは一致したテキスト部分（マッチが始まる最初の文字からマッチが終わる文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループ（または N がキャプチャグループ数を超える場合は空文字列）に置き換えられます;
- **maxdelay** - 最大遅延秒数。型: float。値: 0 - （デフォルト）ログファイル行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行が分析されるように、古い行を無視する。使用前にmaxdelayの注意事項をお読みください!
- **options** - 追加オプション:
mtime-noreread - 一意でないレコード、ファイルサイズが変更された場合のみ再読み込み（変更時刻の変更は無視）。（このパラメータは 5.0.2 以降非推奨です。現在は mtime が無視されるためです。）
- **persistent dir**（Unix システムの zabbix_agentd のみ; Zabbix agent 2 ではサポートされていません）- 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。

コメント:

- アイテムはアクティブチェックとして設定する必要があります;
- ファイルが存在しない場合やアクセス権がない場合、アイテムはサポート対象外になります;

- output が空の場合、一致したテキストを含む行全体が返されます。'Result is TRUE' 以外のすべてのグローバル正規表現タイプは常に一致した行全体を返し、output パラメータは無視されることに注意してください。
- output パラメータを使用した内容の抽出はエージェント側で行われます。

例:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

ログレコードから数値を抽出するために output パラメータを使用する例:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイテムはロ
```

サーバーに送信する前にログレコードを書き換えるために output パラメータを使用する例:

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

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 監視対象のログファイル内で一致した行数をカウントします。
 戻り値: 整数。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ログファイルのフルパスとファイル名;
- **regexp** - 必要なパターンを記述する正規表現;
- **encoding** - コードページの識別子;
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数 (10000 を超えることはできません)。デフォルト値はzabbix_agentd.confの 10*'MaxLinesPerSecond' です。
- **mode** - 可能な値: *all* (デフォルト) または *skip* - 古いデータの処理をスキップ (新しく作成されたアイテムにのみ影響します)。
- **maxdelay** - 秒単位の最大遅延。型: float。値: 0 - (デフォルト) ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析するために古い行を無視します。使用前にmaxdelayの注意事項をお読みください!
- **options** - 追加オプション:
mtime-noreread - 一意でないレコード、ファイルサイズが変更された場合のみ再読み込み (変更時刻の変更は無視)。(このパラメータは 5.0.2 以降非推奨です。現在は mtime が無視されるためです。)
- **persistent dir** (Unix システムの zabbix_agentd のみ; Zabbix agent 2 ではサポートされていません) - 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。

コメント:

- アイテムはアクティブチェックとして設定する必要があります;
- 一致した行は、エージェントによる前回のログチェック以降の新しい行でカウントされるため、アイテムの更新間隔に依存します;
- ファイルが存在しない場合や権限がなくアクセスできない場合、アイテムはサポート対象外になります。

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされるログファイルの監視。
 戻り値: *Log*。
 サポートされているプラットフォームを参照。

パラメータ:

- **file regexp** - ファイル名を正規表現で指定した絶対パス。正規表現はファイル名のみに適用され、ファイル名全体に一致する必要はありません（例: /path/to/agent は zabbix_agentd.log に一致します)。
- **regexp** - 必要な内容パターンを記述する正規表現。
- **encoding** - コードページ識別子。
- **maxlines** - エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新しい行の最大数。このパラメータはzabbix_agentd.confの'MaxLinesPerSecond' の値を上書きします。
- **mode** - 可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップ（新しく作成されたアイテムのみに影響)。
- **output** - オプションの出力フォーマットテンプレート。**\0** エスケープシーケンスは一致したテキスト部分（マッチが始まる最初の文字からマッチが終わる文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループ（または N がキャプチャグループ数を超える場合は空文字列）に置き換えられます。
- **maxdelay** - 最大遅延秒数。型: float。値: 0 -（デフォルト）ログファイル行を無視しない; > 0.0 - "maxdelay" 秒以内に最新行を分析するために古い行を無視する。使用前にmaxdelayの注意事項をお読みください！
- **options** - ログファイルのローテーションタイプおよびその他のオプション。可能な値:
rotate（デフォルト）、
copytruncate - *copytruncate* は *maxdelay* と同時に使用できません。この場合 *maxdelay* は 0 または未指定でなければなりません。copytruncateの注意事項を参照、
mtime-reread - 非ユニークレコード、修正時刻またはサイズが変化した場合に再読込（デフォルト)、
mtime-noreread - 非ユニークレコード、サイズが変化した場合のみ再読込（修正時刻の変化は無視)。

- **persistent dir**（Unix システムの zabbix_agentd のみ; Zabbix agent 2 ではサポートされていません）- 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。

コメント:

- アイテムはアクティブチェックとして設定する必要があります。
- ログローテーションはファイルの最終更新時刻に基づいています。
- logrt は現在アクティブなログファイル 1 つと、ローテーションされた複数の非アクティブファイルで動作するよう設計されています。たとえば、ディレクトリに複数のアクティブなログファイルがある場合、それぞれに個別の logrt アイテムを作成する必要があります。1 つの logrt アイテムが多くのファイルを取得すると、メモリ不足や監視のクラッシュにつながる可能性があります。
- output を空にした場合、一致したテキストを含む行全体が返されます。'Result is TRUE' 以外のすべてのグローバル正規表現タイプは常に一致した行全体を返し、output パラメータは無視されます。
- output パラメータによる内容抽出はエージェント側で行われます。
- file regexp パラメータでは、ログディレクトリパスとログファイル正規表現は正しいディレクトリセパレータで区切る必要があります:
  - Windows では、セパレータはバックスラッシュ（\）でなければなりません。スラッシュ（/）は他の位置では許容されますが、ログディレクトリパスとログファイル正規表現を区切る位置では許容されません（下記の例を参照）。
  - その他のシステムでは、スラッシュ（/）でなければなりません。

Windows の例:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、"c:/dir1/dir2/dir3" 内でファイ
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ネットワーク共
```

その他のシステムの例:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] # このアイテムは"logfile1" のようなファイルに一致します（".lo
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは"logfile_abc_1" や
```

ログレコードから数値を抽出するために output パラメータを使用する例:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイ
```

サーバーに送信する前にログレコードを書き換えるために output パラメータを使用する例:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされる監視対象のログファイル内で一致した行数
 戻り値: 整数
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file regexp** - ファイル名を正規表現で指定した絶対パス。正規表現はファイル名のみに適用され、ファイル名全体と一致する必要はありません（例: /path/to/agent は zabbix_agentd.log に一致します）。
- **regexp** - 必要なパターンを記述する正規表現。
- **encoding** - コードページの識別子。
- **maxproclines** - エージェントが解析する 1 秒あたりの新しい行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の 10*'MaxLinesPerSecond' です。
- **mode** - 可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップ（新しく作成されたアイテムのみに影響）。
- **maxdelay** - 最大遅延秒数。型: float。値: 0 - （デフォルト）ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析するために古い行を無視する。使用前にmaxdelay の注意事項をお読みください！
- **options** - ログファイルのローテーションタイプおよびその他のオプション。可能な値:
rotate（デフォルト）、
copytruncate - *copytruncate* は *maxdelay* と同時に使用できません。この場合、*maxdelay* は 0 または未指定でなければなりません。copytruncate の注意事項を参照、
mtime-reread - 一意でないレコード、変更時刻またはサイズが変更された場合に再読み込み（デフォルト）、
mtime-noreread - 一意でないレコード、サイズが変更された場合のみ再読み込み（変更時刻の変更は無視）。
- **persistent dir**（Unix システムの zabbix_agentd のみ; Zabbix agent 2 ではサポートされていません）- 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。

コメント:

- アイテムはアクティブチェックとして設定する必要があります。
- 一致する行は、エージェントによる最後のログチェック以降の新しい行でカウントされるため、アイテムの更新間隔に依存します。
- ログのローテーションはファイルの最終更新時刻に基づいています。
- file regexp パラメータでは、ログディレクトリのパスとログファイルの正規表現は正しいディレクトリセパレータで区切る必要があります。

- Windows では、セパレータはバックスラッシュ（\）でなければなりません。スラッシュは他の位置では許容される場合がありますが、ログディレクトリのパスとログファイルの正規表現を区切る位置では許容されません（下記の例を参照）。
- その他のシステムでは、スラッシュ（/）でなければなりません。

Windows の例:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは "c:/dir1/dir2/dir3" にあ
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムはネットワ
```

その他のシステムの例:

```
logrt.count["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] # このアイテムは "/home/zabbix/logs" ディレクトリ内の
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは "/home/use
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Modbus データを読み取ります。
 戻り値：*JSON* 文字列。
 対応プラットフォーム：Linux。

パラメータ:

- **endpoint** - protocol://connection_string として定義されたエンドポイント;
- **slave id** - スレーブ ID;
- **function** - Modbus 関数;
- **address** - 最初のレジストリ、コイルまたは入力のアドレス;
- **count** - 読み取るレコード数;
- **type** - データのタイプ;
- **endianness** - エンディアン設定;
- **offset** - 'address' から始まるレジスタ数で、結果が破棄されるもの。

パラメータの詳細な説明を参照してください。

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスの状態をチェックします。
 戻り値：0 - DNS 解決に失敗（DNS サーバーが応答しない、またはエラーを返した場合）；1 - DNS 解決に成功。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip**（Zabbix agent 2 を使用していない Windows では無視されます）- DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空欄）；
- **name** - 問い合わせる DNS 名；
- **type** - 問い合わせるレコードタイプ（デフォルトは *SOA*）；
- **timeout**（Zabbix agent 2 を使用していない Windows では無視されます）- リクエストのタイムアウト（秒、デフォルトは 1 秒）；
- **count**（Zabbix agent 2 を使用していない Windows では無視されます）- リクエストの試行回数（デフォルトは 2）；
- **protocol** - DNS クエリに使用するプロトコル：*udp*（デフォルト）または *tcp*。

コメント:

- type の指定可能な値は、*ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS*（Windows の Zabbix agent、すべての OS の Zabbix agent 2 ではサポートされません）、*HINFO*, *MINFO*, *TXT*, *SRV* です。
- 逆引き DNS ルックアップ（type を *PTR* に設定した場合）では、DNS 名を逆順・非逆順のどちらの形式でも指定できます（下記の例を参照）。PTR レコードを要求する場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。

例:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスのパフォーマンスをチェックします。
 戻り値: *Float* (0 - サービスがダウンしている場合; 秒数 - サービスからの応答を待った秒数)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** (Windows では Zabbix agent 2 を使用しない限り無視されます) - DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空欄）；
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ（デフォルトは *SOA*）；
- **timeout** (Windows では Zabbix agent 2 を使用しない限り無視されます) - リクエストのタイムアウト（秒）(デフォルトは 1 秒)；
- **count** (Windows では Zabbix agent 2 を使用しない限り無視されます) - リクエストの試行回数（デフォルトは 2）；
- **protocol** - DNS クエリを実行するためのプロトコル: *udp*（デフォルト）または *tcp*。

コメント:

- type の可能な値は次のとおりです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS*（Windows 上の Zabbix agent、すべての OS 上の Zabbix agent 2 ではサポートされていません）, *HINFO*, *MINFO*, *TXT*, *SRV*
- 逆引き DNS ルックアップ（type が *PTR* に設定されている場合）では、DNS 名を逆順形式と非逆順形式の両方で指定できます (下記の例を参照)。PTR レコードが要求される場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。
- DNS サーバーがエラーコード（たとえば、`NXDOMAIN` や `SERVFAIL`）で応答した場合、アイテムは 0 ではなく応答時間を返します。

例:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS クエリを実行します。

Zabbix agent 2 は、net.dns.getも提供しており、より多くのレコードタイプや DNS 監視の制御強化などの追加機能を備えています。
戻り値: 必要なタイプの情報を含む文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** (Zabbix agent 2 を使用しない限り Windows では無視されます) - DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空白）；
- **name** - クエリを実行する DNS 名;
- **type** - クエリを実行するレコードタイプ（デフォルトは *SOA*）；
- **timeout** (Zabbix agent 2 を使用しない限り Windows では無視されます) - リクエストのタイムアウト（秒、デフォルトは 1 秒）；
- **count** (Zabbix agent 2 を使用しない限り Windows では無視されます) - リクエストの試行回数（デフォルトは 2）；
- **protocol** - DNS クエリを実行する際のプロトコル: *udp*（デフォルト）または *tcp*。

コメント:

- type の可能な値は以下の通りです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS*（Windows 上の Zabbix agent、全 OS 上の Zabbix agent 2 ではサポートされません）、*HINFO*, *MINFO*, *TXT*, *SRV*
- 逆引き DNS ルックアップ（type を *PTR* に設定した場合）では、DNS 名を逆順または非逆順の形式で指定できます（下記の例を参照）。PTR レコードが要求される場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。

例:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 ウィンドウ外の衝突回数。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名

net.if.discovery

 ネットワークインターフェースのリスト。ローレベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD。

net.if.in[if,<mode>]

 ネットワークインターフェースの受信トラフィック統計情報。
 戻り値: 整数。
 対応プラットフォーム: Linux、FreeBSD、Solaris[5]、HP-UX、AIX、MacOS X、OpenBSD、NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス; もしくは波括弧で囲んだ場合はネットワークインターフェースの GUID(Windows);
- **mode** - 可能な値:
bytes - バイト数 (デフォルト)
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
frame - パケットフレームエラー数
compressed - デバイスドライバによって受信された圧縮パケット数
multicast - デバイスドライバによって受信されたマルチキャストフレーム数

コメント:

- このキーは *Change per second* の前処理ステップと組み合わせて、バイト/秒の統計情報を取得するために使用できます。
- *dropped* モードは Linux、FreeBSD、HP-UX、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *overruns*、*frame*、*compressed*、*multicast* モードは Linux でのみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 ネットワークインターフェースの送信トラフィック統計情報。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名（Unix の場合）; ネットワークインターフェースの完全な説明または IPv4 アドレス；または、中括弧で囲まれた場合はネットワークインターフェースの GUID（Windows の場合）;
- **mode** - 指定可能な値：
bytes - バイト数（デフォルト）
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
collisions (colls) - インターフェースで検出された衝突数
carrier - デバイスドライバによって検出されたキャリアロス数
compressed - デバイスドライバによって送信された圧縮パケット数

コメント:

- このキーは、*Change per second* の前処理ステップと組み合わせて使用することで、1 秒あたりのバイト数の統計を取得できます。
- *dropped* モードは Linux、HP-UX のみサポートされています。
- *overruns*、*collision*、*carrier*、*compressed* モードは Linux のみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース（例: lo0）の詳細を提供しません。

例:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 ネットワークインターフェースの受信および送信トラフィック統計の合計。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス; もしくは、中括弧で囲んだ場合はネットワークインターフェースの GUID (Windows);
- **mode** - 指定可能な値:
bytes - バイト数 (デフォルト)
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
collisions (colls) - インターフェースで検出された衝突数
compressed - デバイスドライバによって送受信された圧縮パケット数

コメント:

- このキーは、*1* 秒あたりの変化の前処理ステップと組み合わせて使用することで、バイト/秒の統計を取得できます。
- *dropped* モードは Linux、HP-UX のみサポートされています。ドロップされたパケットは、プラットフォームで `net.if.in` と `net.if.out` の両方がドロップされたパケットに対応している場合のみサポートされます。
- *overruns*、*collision*、*compressed* モードは Linux のみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 この TCP ポートが LISTEN 状態かどうかをチェックします。
 戻り値: 0 - LISTEN 状態ではありません; 1 - LISTEN 状態です。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X。

パラメータ:

- **port** - TCP ポート番号

Linux カーネル 2.6.14 以降では、可能な場合はカーネルの NETLINK インターフェースからリッスンしている TCP ソケットの情報を取得します。それ以外の場合は、/proc/net/tcp および/roc/net/tcp6 ファイルから情報を取得します。

例:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 指定したポートへの TCP 接続が可能かどうかをチェックします。
 戻り値: 0 - 接続できない; 1 - 接続できる。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号。

コメント:

- シンプルな TCP パフォーマンステストには `net.tcp.service.perf[tcp,<ip>,<port>]` を使用してください;
- これらのチェックは、システムデーモンのログファイルに追加のメッセージを生成する場合があります (通常、SMTP や SSH セッションが記録されます)。

例:

```
net.tcp.port[,80] # このアイテムは、ポート 80 で稼働している Web サーバの可用性テストに使用できます
```

net.tcp.service[service,<ip>,<port>]

 サービスが稼働しており、TCP 接続を受け付けているかどうかをチェックします。
 戻り値: 0 - サービスがダウン; 1 - サービスが稼働中。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, または *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- これらのチェックは、システムデーモンのログファイルに追加のメッセージを生成する場合があります (通常、SMTP や SSH セッションが記録されます);
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。このようなチェックには、`net.tcp.port[]` を使用してください。
- Windows での LDAP および HTTPS のチェックは、Zabbix agent 2 でのみサポートされています;
- telnet チェックはログインプロンプト (末尾の':' 記号) を探します。

例:

```
net.tcp.service[ftp,,45] # このアイテムは、TCP ポート 45 での FTP サーバの可用性をテストするために使用できます
```

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスをチェックします。
 戻り値: *Float* (0 - サービスがダウン; 秒数 - サービスからの応答を待つのに費やした秒数)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ssh*、*ldap*、*smtp*、*ftp*、*http*、*pop*、*nntp*、*imap*、*tcp*、*https*、または *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。このようなチェックには、`net.tcp.service.perf[tcp,<ip>,<port>]` を回避策として使用してください。
- telnet チェックはログインプロンプト (末尾の':') を探します。

例:

```
net.tcp.service.perf[ssh] # このアイテムは SSH サーバーからの初期応答の速度をテストするために使用できます
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する TCP ソケットの数を返します。
 戻り値：整数。
 対応プラットフォーム：Linux。

パラメータ:

- **laddr** - ローカル IPv4/6 アドレスまたは CIDR サブネット
- **lport** - ローカルポート番号またはサービス名
- **raddr** - リモート IPv4/6 アドレスまたは CIDR サブネット
- **rport** - リモートポート番号またはサービス名
- **state** - 接続状態 (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*)

例:

```
net.tcp.socket.count[,80,,,established] # 確立状態のローカル TCP ポート 80 への接続数
```

net.udp.listen[port]

 この UDP ポートが LISTEN 状態かどうかをチェックします。
 戻り値: 0 - LISTEN 状態ではありません; 1 - LISTEN 状態です。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X。

パラメータ:

- **port** - UDP ポート番号

例:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 サービスが実行されており、UDP リクエストに応答しているかどうかをチェックします。
 戻り値: 0 - サービスがダウンしている; 1 - サービスが実行中。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

例:

```
net.udp.service[ntp,,45] # このアイテムは UDP ポート 45 での NTP サービスの可用性をテストするために使用できます
```

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスをチェックします。
 戻り値: *Float* (0 - サービスがダウンしている場合; 秒数 - サービスからの応答を待つのに費やした秒数)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

例:

```
net.udp.service.perf[ntp] # このアイテムは NTP サービスからの応答時間のテストに使用できます
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する UDP ソケットの数を返します。
 戻り値: 整数。
 対応プラットフォーム: Linux。

パラメータ:

- **laddr** - ローカル IPv4/6 アドレスまたは CIDR サブネット
- **lport** - ローカルポート番号またはサービス名
- **raddr** - リモート IPv4/6 アドレスまたは CIDR サブネット
- **rport** - リモートポート番号またはサービス名
- **state** - 接続状態 (*established*, *unconn*)

例:

```
net.udp.socket.count[,,,,established] # 接続状態の UDP ソケット数を返す
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 プロセスの CPU 使用率 (%)。
 戻り値: *Float*。
 対応プラットフォーム: Linux, Solaris[6]。

パラメータ:

- **name** - プロセス名 (デフォルトはすべてのプロセス);
- **user** - ユーザー名 (デフォルトはすべてのユーザー);
- **type** - CPU 使用率のタイプ: *total*(デフォルト)、*user*、または *system*;
- **cmdline** - コマンドラインでフィルタリング (正規表現);
- **mode** - データ収集モード: *avg1*(デフォルト)、*avg5*、または *avg15*;
- **zone** - 対象ゾーン: *current*(デフォルト) または *all*。このパラメータは Solaris のみでサポートされています。

コメント:

- 返される値は、単一の CPU コアの使用率に基づきます。たとえば、2 つのコアをフルに使用しているプロセスの CPU 使用率は 200% です。
- プロセスの CPU 使用率データはコレクターによって収集され、名前、ユーザー、コマンドラインごとに最大 1024 個のユニークなクエリをサポートします。過去 24 時間アクセスされなかったクエリはコレクターから削除されます。
- zone パラメータを *current*(またはデフォルト) に設定し、エージェントがゾーンをサポートしない Solaris でコンパイルされているが、ゾーンがサポートされている新しい Solaris で実行されている場合、エージェントは NOTSUPPORTED を返します (エージェントは結果を現在のゾーンのみに制限できません)。ただし、この場合 *all* はサポートされます。

例:

```
proc.cpu.util[,root] # "root" ユーザーで実行されているすべてのプロセスの CPU 使用率
proc.cpu.util[zabbix_server,zabbix] # zabbix ユーザーで実行されているすべての zabbix_server プロセスの CPU 使用率
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 OS プロセスとそのパラメータのリスト。低レベルディスカバリに使用できます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, FreeBSD, Windows, OpenBSD, NetBSD。

パラメータ:

- **name** - プロセス名 (デフォルトはすべてのプロセス);
- **user** - ユーザー名 (デフォルトはすべてのユーザー);
- **cmdline** - コマンドラインでフィルタリング (正規表現)。このパラメータは Windows ではサポートされていません。他のプラットフォームでも mode が'summary' に設定されている場合はサポートされていません。
- **mode** - 可能な値:
process (デフォルト), *thread* (NetBSD ではサポートされていません), *summary*。各 mode および OS で返されるプロセスパラメータのリストを参照してください。

コメント:

- たとえば、エラー (プロセスがすでに終了している、権限がない、システムコールの失敗など) のために値を取得できない場合は、-1 が返されます。
- name および cmdline パラメータによるプロセスの選択については、注意事項を参照してください (Linux 固有)。

例:

```
proc.get[zabbix_server,zabbix,,process] #zabbix ユーザーで実行されているすべての zabbix_server プロセスのリスト、Pl
proc.get[java,,,thread] # すべての Java プロセスのリスト、スレッドごとに 1 つのエントリを返す
proc.get[,zabbix,,summary] #zabbix ユーザーで実行されている各タイプのプロセスの集計データ、プロセス名ごとに 1 つのエント
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 プロセスが使用しているメモリ量 (バイト単位)。
 戻り値: *Integer* - mode が *max*、*min*、*sum* の場合; *Float* - mode が *avg* の場合
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD。

パラメータ:

- **name** - プロセス名 (デフォルトはすべてのプロセス);
- **user** - ユーザー名 (デフォルトはすべてのユーザー);
- **mode** - 許可される値: *avg*、*max*、*min*、または *sum* (デフォルト);
- **cmdline** - コマンドラインでフィルタ (正規表現);
- **memtype** - プロセスが使用するメモリの種類

コメント:

- memtype パラメータは Linux、FreeBSD、Solaris[6]、AIX でのみサポートされています;
- 複数のプロセスが共有メモリを使用している場合、プロセスが使用するメモリの合計は大きく非現実的な値になることがあります。

name および cmdline パラメータによるプロセス選択の注意事項 (Linux 固有) を参照してください。

 このアイテムをコマンドラインから呼び出し、コマンドラインパラメータを含めた場合 (例: エージェントのテストモードで zabbix_agentd -t proc.mem[,,,apache2] を使用)、エージェント自身もカウントされるため、1 つ余分にプロセスがカウントされます。

例:

```
proc.mem[,root] # "root" ユーザーで実行されているすべてのプロセスが使用するメモリ
proc.mem[zabbix_server,zabbix] # zabbix ユーザーで実行されているすべての zabbix_server プロセスが使用するメモリ
proc.mem[,oracle,max,oracleZABBIX] # Oracle ユーザーで実行されているプロセスのうち、コマンドラインに oracleZABBIX を
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 プロセス数を返します。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD。

パラメータ:

- **name** - プロセス名 (デフォルトはすべてのプロセス);
- **user** - ユーザー名 (デフォルトはすべてのユーザー);
- **state** - 指定可能な値:
all (デフォルト),
disk - 割り込み不可のスリープ,
run - 実行中,
sleep - 割り込み可能なスリープ,
trace - 停止中,
zomb - ゾンビ;
- **cmdline** - コマンドラインによるフィルタ (正規表現を使用);
- **zone** - 対象ゾーン: *current* (デフォルト) または *all*。このパラメータは Solaris のみサポートされます。

コメント:

- *disk* および *trace* の state パラメータは Linux、FreeBSD、OpenBSD、NetBSD のみサポートされます。
- このアイテムをコマンドラインから呼び出し、コマンドラインパラメータを含む場合 (例: エージェントのテストモードで zabbix_agentd -t proc.num[,,,apache2] を使用)、エージェント自身もカウントされるため、1 つ多くプロセスがカウントされます。
- エージェントがゾーンサポートなしで Solaris 上でコンパイルされ、ゾーンがサポートされている新しい Solaris 上で実行されている場合、zone パラメータを *current* (またはデフォルト) に設定すると、エージェントは NOTSUPPORTED を返します (エージェントは現在のゾーンのみに結果を制限できません)。ただし、この場合 *all* はサポートされます。
- name および cmdline パラメータによるプロセスの選択については、注意事項を参照してください (Linux 固有)。

例:

```
proc.num[,mysql] #mysql ユーザーで実行されているプロセス数
proc.num[apache2,www-data] #www-data ユーザーで実行されている apache2 プロセス数
proc.num[,oracle,sleep,oracleZABBIX] # コマンドラインに oracleZABBIX を含む、sleep 状態の Oracle プロセス数
```

sensor[device,sensor,<mode>]

 ハードウェアセンサーの読み取り値。
 戻り値: *Float*。
 対応プラットフォーム: Linux, OpenBSD。

パラメータ:

- **device** - デバイス名。mode が省略された場合は正規表現も可。
- **sensor** - センサー名。mode が省略された場合は正規表現も可。
- **mode** - *avg*、*max*、*min* のいずれか (このパラメータが省略された場合、device と sensor はそのまま扱われる)。

コメント:

- Linux 2.4 では/proc/sys/dev/sensors を読み取る。
- Linux 2.6+ では/sys/class/hwmon を読み取る。Linux でのsensorアイテムの詳細な説明を参照。

- OpenBSD では *hw.sensors* MIB を読み取る。

例:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #1 つの CPU の温度
sensor["cpu[0-2]$",temp,avg] # 最初の 3 つの CPU の平均温度
```

system.boottime

 システムの起動時刻。
 戻り値: 整数 *(Unix* タイムスタンプ*)*。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD。

system.cpu.discovery

 検出された CPU/CPU コアのリスト。低レベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

system.cpu.intr

 デバイス割り込み。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD。

system.cpu.load[<cpu>,<mode>]

 CPU ロード。
 戻り値：*Float*。
 サポートされているプラットフォームを参照。

パラメータ:

- **cpu** - 可能な値：*all*（デフォルト）または *percpu*（合計ロードをオンライン CPU 数で割った値）；
- **mode** - 可能な値：*avg1*（1 分間の平均、デフォルト）、*avg5*、または *avg15*。

例:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 CPU の数。
 戻り値：整数。
 対応プラットフォーム：Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD。

パラメータ:

- **type** - 可能な値：*online*（デフォルト）または *max*

max タイプのパラメータは Linux、FreeBSD、Solaris、MacOS X でのみサポートされています。

例:

```
system.cpu.num
```

system.cpu.switches

 コンテキストスイッチの回数。
 戻り値: 整数。
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、NetBSD。

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 CPU 使用率（パーセンテージ）
 戻り値：*Float*
 対応プラットフォーム：Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD

パラメータ:

- **cpu** - *<CPU* 番号 > または *all*（デフォルト）
- **type** - 指定可能な値：*user*（デフォルト）、*idle*、*nice*、*system*、*iowait*、*interrupt*、*softirq*、*steal*（OpenBSD のみ）、*guest*（Linux カーネル 2.6.24 以降）、*guest_nice*（Linux カーネル 2.6.33 以降）
- **mode** - 指定可能な値：*avg1*（1 分平均、デフォルト）、*avg5*、*avg15*
- **logical or physical** - 指定可能な値：*logical*（デフォルト）または *physical*。このパラメータは AIX のみ対応。

コメント:

- *nice* タイプパラメータは Linux、FreeBSD、HP-UX、OpenBSD、NetBSD のみ対応。
- *iowait* タイプパラメータは Linux 2.6 以降、Solaris、AIX のみ対応。
- *interrupt* タイプパラメータは Linux 2.6 以降、FreeBSD、OpenBSD のみ対応。
- *softirq*、*steal*、*guest*、*guest_nice* タイプパラメータは Linux 2.6 以降のみ対応。
- *avg5* および *avg15* モードパラメータは Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD で対応。

例:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 システムのホスト名。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **type** - 可能な値: *netbios* (Windows のデフォルト)、*host* (Linux のデフォルト)、*shorthost* (最初のドットの前のホスト名の部分、ドットのない名前の場合は完全な文字列を返す)、*fqdn* (完全修飾ドメイン名を返す);
- **transform** - 可能な値: *none* (デフォルト) または *lower* (小文字に変換)。

値は uname() システム API の出力から `nodename` を取得して取得されます。

返される値の例:

```
system.hostname → linux-w7x1
system.hostname → example.com
system.hostname[shorthost] → example
system.hostname → WIN-SERV2008-I6
system.hostname[host] → Win-Serv2008-I6LonG
system.hostname[host,lower] → win-serv2008-i6long
system.hostname[fqdn,lower] → blog.zabbix.com
```

system.hw.chassis[<info>]

 シャーシ情報。
 戻り値：文字列。
 対応プラットフォーム：Linux。

パラメータ:

- **info** - 可能な値：*full*（デフォルト）、*model*、*serial*、*type*、*vendor*

コメント:

- このアイテムキーはSMBIOSテーブルの利用可能性に依存します。
- sysfs から DMI テーブルの読み取りを試み、sysfs へのアクセスに失敗した場合はメモリから直接読み取ります。
- 値は sysfs またはメモリから読み取るため、**root** 権限が必要です。

例:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 CPU 情報。
 戻り値: *String* または *Integer*。
 対応プラットフォーム: Linux。

パラメータ:

- **cpu** - <*CPU* 番号 > または *all* (デフォルト);
- **info** - 指定可能な値: *full* (デフォルト), *curfreq*, *maxfreq*, *model* または *vendor*。

コメント:

- `/proc/cpuinfo` および `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq` から情報を取得します;
- CPU 番号と *curfreq* または *maxfreq* が指定された場合、数値 (Hz) が返されます。

例:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 PCI または USB デバイスの一覧。
 戻り値: テキスト。
 対応プラットフォーム: Linux。

パラメータ:

- **type** - *pci* (デフォルト) または *usb*

lspci または lsusb ユーティリティ (パラメータなしで実行) の出力を返します。

例:

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

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

 MAC アドレスの一覧。
 戻り値：*String*。
 対応プラットフォーム：Linux。

パラメータ:

- **interface** - *all* (デフォルト) または正規表現
- **format** - *full* (デフォルト) または *short*

コメント:

- 名前が指定された `interface` の正規表現に一致するインターフェースの MAC アドレスを一覧表示します（*all* はすべてのインターフェースを一覧表示します）
- `format` が *short* に指定されている場合、インターフェース名や同一の MAC アドレスは一覧表示されません。

例:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 システム時刻。
 戻り値: *Integer* - `type` が *utc* の場合; *String* - `type` が *local* の場合。
 サポートされているプラットフォームを参照。

パラメータ:

- **type** - 可能な値: *utc* - (デフォルト) エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数、または *local* - 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm' 形式の時刻

Zabbix エージェントではパッシブチェックとして使用する必要があります。Zabbix agent 2 ではアクティブチェックとして設定することもできます。

例:

```
system.localtime[local] # このキーを使用してアイテムを作成し、*Clock* ダッシュボードウィジェットでホスト時刻を表示するため
```

system.run[command,<mode>]

 指定したコマンドをホスト上で実行します。
 戻り値: コマンドのテキスト結果、または `mode` が *nowait* の場合は 1（コマンド結果に関係なく）。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **command** - 実行するコマンド;
- **mode** - 可能な値: *wait* - 実行終了まで待機（デフォルト）、または *nowait* - 待機しない。

コメント:

- このアイテムはデフォルトで無効になっています。有効化方法を参照してください;
- アイテムの戻り値は、コマンドによって生成された標準出力と標準エラーです。終了コードのチェックは行われません;
- 正しく処理するには、コマンドの戻り値は `text` データ型である必要があります。空の結果も許可されます;
- 戻り値は 16MB に制限されています（末尾の空白も含み、切り捨てられます）。データベースの制限も適用されます;
- 参考: コマンド実行。

例:

```
system.run[ls -l /] # ルートディレクトリの詳細なファイルリストを返す
```

system.stat[resource,<type>]

 システム統計情報。
 戻り値: 整数または *float*。
 対応プラットフォーム: AIX。

パラメータ:

- **ent** - このパーティションが受け取ることができるプロセッサユニット数 (float)
- **kthr,<type>** - カーネルスレッド状態に関する情報:
r - 実行可能なカーネルスレッドの平均数 (float)
b - 仮想メモリマネージャの待機キューに配置されたカーネルスレッドの平均数 (float)
- **memory,<type>** - 仮想メモリおよび実メモリの使用状況に関する情報:
avm - アクティブな仮想ページ数 (整数)
fre - フリーリストのサイズ (整数)
- **page,<type>** - ページフォールトおよびページングアクティビティに関する情報:
fi - 1 秒あたりのファイルページイン数 (float)
fo - 1 秒あたりのファイルページアウト数 (float)
pi - ページングスペースからページインされたページ数 (float)
po - ページングスペースにページアウトされたページ数 (float)
fr - 解放されたページ数 (ページ置換) (float)
sr - ページ置換アルゴリズムによってスキャンされたページ数 (float)
- **faults,<type>** - トラップおよび割り込み率:
in - デバイス割り込み (float)
sy - システムコール (float)
cs - カーネルスレッドのコンテキストスイッチ数 (float)

- **cpu,<type>** - プロセッサ時間の使用率の内訳:
us - ユーザー時間 (float)
sy - システム時間 (float)
id - アイドル時間 (float)
wa - ディスク/NFS I/O リクエストが未処理の間のアイドル時間 (float)
pc - 消費された物理プロセッサ数 (float)
ec - 消費されたエンタイトルドキャパシティの割合 (float)
lbusy - ユーザーおよびシステムレベルで実行中に発生した論理プロセッサの使用率 (float)
app - 共有プール内で利用可能な物理プロセッサ数 (float)
- **disk,<type>** - ディスク統計情報:
bps - ドライブに転送されたデータ量 (読み取りまたは書き込み) を 1 秒あたりのバイト数で示す (整数)
tps - 物理ディスク/テープに発行された 1 秒あたりの転送数 (float)

コメント:

- これらのアイテムには以下の制限があることに注意してください:
 `system.stat[cpu,app]` - AIX LPAR のタイプ「Shared」でのみサポート
 `system.stat[cpu,ec]` - AIX LPAR のタイプ「Shared」と「Dedicated」でサポート (「Dedicated」は常に 100（パーセント）を返す）
 `system.stat[cpu,lbusy]` - AIX LPAR のタイプ「Shared」でのみサポート
 `system.stat[cpu,pc]` - AIX LPAR のタイプ「Shared」と「Dedicated」でサポート
 `system.stat[ent]` - AIX LPAR のタイプ「Shared」と「Dedicated」でサポート

system.sw.arch

 ソフトウェアアーキテクチャ情報。
 戻り値: *String*。
 サポートされているプラットフォームを参照。

情報は `uname()` 関数から取得されます。

例:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 オペレーティングシステム情報。
 戻り値: *String*。
 対応プラットフォーム: Linux, Windows。

パラメータ:

- **info** - 可能な値: *full* (デフォルト), *short*, *name*

情報は以下から取得されます（すべてのディストリビューションで全てのファイルやオプションが存在するわけではありません）:

- Linux では `/proc/version` (*full*) ;
- Linux では `/proc/version_signature` (*short*) ;
- Linux システムでサポートされている場合は `/etc/os-release` の PRETTY_NAME パラメータ、または `/etc/issue.net` (*name*) ;
- Windows では `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` レジストリキー。

例:

```
system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11
system.sw.os[full] → [s|Windows 10 Enterprise 22621.1.asd64fre.ni_release.220506-1250 Build 22621.963]
```

system.sw.os.get

 オペレーティングシステムに関する詳細情報（バージョン、タイプ、ディストリビューション名、マイナーおよびメジャーバージョンなど）。
 戻り値 : *JSON* 文字列。
 対応プラットフォーム : Linux、Windows。

system.sw.packages[<regexp>,<manager>,<format>]

 インストールされているパッケージの一覧。
 戻り値: テキスト。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all* (デフォルト) または正規表現
- **manager** - *all* (デフォルト) またはパッケージマネージャ
- **format** - *full* (デフォルト) または *short*

コメント:

- 名前が指定した正規表現に一致するインストール済みパッケージを (アルファベット順で) 一覧表示します (*all* はすべてを表示)
- サポートされているパッケージマネージャ (実行コマンド):
dpkg (dpkg --get-selections)
pkgtool (ls /var/log/packages)
rpm (rpm -qa)
pacman (pacman -Q)
portage
- `format` が *full* の場合、パッケージはパッケージマネージャごとにグループ化されます (各マネージャは角括弧で囲まれた名前で始まる別々の行)
- `format` が *short* の場合、パッケージはグループ化されず、1 行で一覧表示されます

例:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 インストールされているパッケージの詳細リスト。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all* (デフォルト) または正規表現
- **manager** - *all* (デフォルト) またはパッケージマネージャー (指定可能な値: *rpm*, *dpkg*, *pkgtool*, *pacman*, *portage*)

コメント:

- 指定した正規表現に一致する名前のインストール済みパッケージを、フォーマットされていない JSON で返します。
- 出力は、name、manager、version、size、architecture、buildtime、installtime の各キーを含むオブジェクトの配列です (詳細はこちら)。

system.swap.in[<device>,<type>]

 スワップイン（デバイスからメモリへの）統計情報。
 戻り値：整数。
 対応プラットフォーム：Linux、FreeBSD、OpenBSD。

パラメータ:

- **device** - スワップに使用するデバイス（Linux のみ）または *all*（デフォルト）を指定します。
- **type** - 可能な値：*count*（スワップインの回数、非 Linux プラットフォームのデフォルト）、*sectors*（スワップインしたセクタ数）、または *pages*（スワップインしたページ数、Linux のデフォルト）。

コメント:

- この情報のソースは以下の通りです：
/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)
/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)
- *pages* は device が指定されていない場合のみ機能します。
- *sectors* タイプのパラメータは Linux でのみサポートされています。

例:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 スワップアウト（メモリからデバイスへの移動）の統計情報。
 戻り値：整数。
 対応プラットフォーム：Linux、FreeBSD、OpenBSD。

パラメータ:

- **device** - スワップに使用するデバイスを指定（Linux のみ）または *all*（デフォルト）；
- **type** - 指定可能な値：*count*（スワップアウト回数、非 Linux プラットフォームのデフォルト）、*sectors*（スワップアウトしたセクタ数）、*pages*（スワップアウトしたページ数、Linux のデフォルト）。

コメント:

- この情報のソースは以下の通り：
`/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4)
`/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6)
- *pages* は device が指定されていない場合のみ機能することに注意;
- *sectors* タイプパラメータは Linux のみサポート。

例:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 スワップ領域のサイズ (バイト単位または合計に対する割合) を返します。
 戻り値: バイトの場合は整数、割合の場合は浮動小数点数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD。

パラメータ:

- **device** - スワップに使用するデバイスを指定します (FreeBSD のみ)。または *all*(デフォルト);
- **type** - *free*(空きスワップ領域、デフォルト)、*pfree*(空きスワップ領域、割合)、*pused*(使用中スワップ領域、割合)、*total*(スワップ領域合計)、*used*(使用中スワップ領域) のいずれか。

コメント:

- *pfree*、*pused* は、スワップサイズが 0 の場合、Windows ではサポートされません。
- device が指定されていない場合、Zabbix エージェントはスワップデバイス (ファイル) のみを考慮し、物理メモリは無視します。例えば、Solaris システムでは `swap -s` コマンドは物理メモリの一部とスワップデバイスを含みます (`swap -l` とは異なります)。

例:

```
system.swap.size[,pfree] → 空きスワップ領域の割合
```

system.uname

 システムの識別。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

コメント:

- UNIX では、このアイテムの値は uname() システムコールで取得されます。
- Windows では OS アーキテクチャを返し、UNIX では CPU アーキテクチャを返します。

例:

```
system.uname → FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
system.uname → Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
```

system.uptime

 システムの稼働時間 (秒単位)。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD。

アイテムの設定では、**s** または **uptime** の単位を使用して可読値を取得します。

system.users.num

 ログインしているユーザー数。
 戻り値: 整数。
 サポートされているプラットフォームを参照してください。

値を取得するために、エージェント側で **who** コマンドが使用されます。

vfs.dev.discovery

 ブロックデバイスとそのタイプのリスト。低レベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux。

vfs.dev.read[<device>,<type>,<mode>]

 ディスクの読み取り統計情報。
 戻り値：*Integer* - type が *sectors*、*operations*、*bytes* の場合；*Float* - type が *sps*、*ops*、*bps* の場合。
 対応プラットフォーム：Linux、FreeBSD、Solaris、AIX、OpenBSD。

パラメータ:

- **device** - ディスクデバイス（デフォルトは *all* [3]）；
- **type** - 指定可能な値：*sectors*、*operations*、*bytes*、*sps*、*ops*、*bps*（*sps*、*ops*、*bps* はそれぞれ、1 秒あたりのセクタ数、1 秒あたりの操作数、1 秒あたりのバイト数）；
- **mode** - 指定可能な値：*avg1*（1 分間の平均、デフォルト）、*avg5*、*avg15*。このパラメータは、type が sps、ops、bps の場合のみサポートされます。

コメント:

- 更新間隔が 3 時間以上[2] の場合、このアイテムは常に'0' を返します;
- *sectors* および *sps* タイプのパラメータは Linux のみサポートされます;
- *ops* タイプのパラメータは Linux および FreeBSD のみサポートされます;
- *bps* タイプのパラメータは FreeBSD のみサポートされます;
- *bytes* タイプのパラメータは FreeBSD、Solaris、AIX、OpenBSD のみサポートされます;
- mode パラメータは Linux、FreeBSD のみサポートされます;
- 相対デバイス名（例：sda）およびオプションの/dev/プレフィックス（例：/dev/sda）の両方を使用できます;
- LVM 論理ボリュームがサポートされています;
- OS ごとの'type' パラメータのデフォルト値：
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- サポートされているプラットフォームでの *sps*、*ops*、*bps* は、1024 デバイス（個別に 1023 台と *all* 用に 1 台）に制限されています。

例:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 ディスク書き込み統計情報。
 戻り値: *Integer* - type が *sectors*、*operations*、*bytes* の場合; *Float* - type が *sps*、*ops*、*bps* の場合。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD。

パラメータ:

- **device** - ディスクデバイス (デフォルトは *all* [3]);

- **type** - 指定可能な値: *sectors*、*operations*、*bytes*、*sps*、*ops*、*bps* (*sps*、*ops*、*bps* はそれぞれ、1 秒あたりのセクタ数、オペレーション数、バイト数を表します);
- **mode** - 指定可能な値: *avg1* (1 分間の平均、デフォルト)、*avg5*、*avg15*。このパラメータは、type が sps、ops、bps の場合のみサポートされます。

コメント:

- 更新間隔が 3 時間以上[2] の場合、このアイテムは常に'0' を返します;
- *sectors* および *sps* の type パラメータは Linux のみサポートされます;
- *ops* の type パラメータは Linux および FreeBSD のみサポートされます;
- *bps* の type パラメータは FreeBSD のみサポートされます;
- *bytes* の type パラメータは FreeBSD、Solaris、AIX、OpenBSD のみサポートされます;
- mode パラメータは Linux、FreeBSD のみサポートされます;
- 相対デバイス名 (例: sda) または/dev/プレフィックス付き (例: /dev/sda) のいずれも使用できます;
- LVM 論理ボリュームがサポートされています;
- OS ごとの'type' パラメータのデフォルト値:
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- サポートされているプラットフォームでの *sps*、*ops*、*bps* は、1024 デバイス (個別 1023 台 +*all* 用 1 台) に制限されます。

例:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリの数を返します。
 戻り値: 整数。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **dir** - ディレクトリへの絶対パス
- **regex incl** - 含めるエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合はすべて含める（デフォルト値）。
- **regex excl** - 除外するエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合は除外しない（デフォルト値）。
- **types incl** - カウントするディレクトリエントリのタイプ。指定可能な値: *file* - 通常ファイル、*dir* - サブディレクトリ、*sym* - シンボリックリンク、*sock* - ソケット、*bdev* - ブロックデバイス、*cdev* - キャラクタデバイス、*fifo* - FIFO、*dev* - "bdev,cdev" と同義、*all* - すべてのタイプ（デフォルト）、すなわち"file,dir,sym,sock,bdev,cdev,fifo"。複数のタイプはカンマ区切りで引用符で囲む必要があります。
- **types excl** - カウントしないディレクトリエントリのタイプ（types incl を参照）。types incl と types excl の両方に含まれるエントリタイプはカウントされません。
- **max depth** - サブディレクトリを再帰的に探索する最大深度:
-1（デフォルト）- 無制限,
0 - サブディレクトリを探索しない。
- **min size** - カウント対象とするファイルの最小サイズ（バイト単位）。これより小さいファイルはカウントされません。メモリサフィックスが使用できます。
- **max size** - カウント対象とするファイルの最大サイズ（バイト単位）。これより大きいファイルはカウントされません。メモリサフィックスが使用できます。
- **min age** - カウント対象とするディレクトリエントリの最小経過時間（秒）。これより新しいエントリはカウントされません。時間サフィックスが使用できます。
- **max age** - カウント対象とするディレクトリエントリの最大経過時間（秒）。これより古いエントリはカウントされません（最終更新時刻）。時間サフィックスが使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。ディレクトリの内容全体が除外されます (regex_excl とは異なります)。

コメント:

- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません。
- 疑似ディレクトリ"." および".." はカウントされません。
- ディレクトリ探索時にシンボリックリンクはたどりません。
- エントリ数の計算時、regex incl と regex excl はファイルとディレクトリの両方に適用されますが、サブディレクトリの探索時には無視されます（regex incl が "(?i)^.+\.zip$" で max depth が設定されていない場合、すべてのサブディレクトリが探索されますが、zip ファイルのみがカウントされます）。
- 実行時間はエージェントの設定の timeout 値（3 秒）で制限されます。大きなディレクトリの探索にはそれ以上かかる場合があるため、データが返されずアイテムは未サポートになります。部分的なカウントは返されません。
- サイズでフィルタリングする場合、有効なサイズを持つのは通常ファイルのみです。Linux や BSD ではディレクトリにも非ゼロのサイズ（通常数 KB）が付きます。デバイスはサイズがゼロです。たとえば**/dev/sda1** のサイズは実際のパーティションサイズを反映しません。したがって、<min_size> や <max_size> を使用する場合は、予期しない結果を避けるために <types_incl> を"*file*" に指定することを推奨します。

例:

```
vfs.dir.count[/dev] #/dev 内のデバイス数を監視（Linux）
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリ内のファイル数を監視
```

vfs.dir.get[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリのリスト。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照。

パラメータ:

- **dir** - ディレクトリへの絶対パス
- **regex incl** - 含めるエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する正規表現。空の場合はすべて含める (デフォルト値)
- **regex excl** - 除外するエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する正規表現。空の場合は除外しない (デフォルト値)
- **types incl** - リストするディレクトリエントリのタイプ。指定可能な値: *file* - 通常ファイル, *dir* - サブディレクトリ, *sym* - シンボリックリンク, *sock* - ソケット, *bdev* - ブロックデバイス, *cdev* - キャラクタデバイス, *fifo* - FIFO, *dev* - "bdev,cdev" と同義, *all* - すべてのタイプ (デフォルト)、すなわち"file,dir,sym,sock,bdev,cdev,fifo"。複数のタイプはカンマ区切りで引用符で囲む必要があります。
- **types excl** - リストしないディレクトリエントリタイプ (`types incl` を参照)。`types incl` と `types excl` の両方に含まれるエントリタイプはリストされません。
- **max depth** - サブディレクトリをたどる最大深度:
-1 (デフォルト) - 無制限,
0 - サブディレクトリに降りない。
- **min size** - リストするファイルの最小サイズ (バイト単位)。これより小さいファイルはリストされません。メモリサフィックスが使用できます。
- **max size** - リストするファイルの最大サイズ (バイト単位)。これより大きいファイルはリストされません。メモリサフィックスが使用できます。
- **min age** - リストするディレクトリエントリの最小経過時間 (秒)。これより新しいエントリはリストされません。時間サフィックスが使用できます。
- **max age** - リストするディレクトリエントリの最大経過時間 (秒)。これより古いエントリはリストされません (最終更新時刻)。時間サフィックスが使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。ディレクトリのすべての内容が除外されます (`regex excl` とは異なります)

コメント:

- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません
- 疑似ディレクトリ"." と".." は決してリストされません
- ディレクトリの走査時にシンボリックリンクはたどりません
- エントリリストの生成時には `regex incl` と `regex excl` の両方がファイルとディレクトリに適用されますが、走査するサブディレクトリの選択時には無視されます (例えば `regex incl` が "(?i)^.+\.zip$" で `max depth` が設定されていない場合、すべてのサブディレクトリが走査されますが、zip タイプのファイルのみがカウントされます)。
- 実行時間はエージェントの設定の timeout 値で制限されます。大きなディレクトリの走査にはそれ以上の時間がかかる場合があるため、データは返されずアイテムはサポート外になります。部分的なリストは返されません。
- サイズでフィルタリングする場合、有効なサイズを持つのは通常ファイルのみです。Linux や BSD ではディレクトリにも非ゼロのサイズ (通常数 KB) があります。デバイスはサイズがゼロです。例えば**/dev/sda1** のサイズは対応するパーティションサイズを反映しません。したがって、`min size` や `max size` を使用する場合は、`types incl` を"*file*" に指定することを推奨します。

例:

```
vfs.dir.get[/dev] #/dev( Linux ) のデバイスリストを取得
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリのファイルリストを取得
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 ディレクトリサイズ (バイト単位)。
 戻り値: 整数。
 対応プラットフォーム: Linux, Windows。他の UNIX 系プラットフォームでも動作する場合があります。

パラメータ:

- **dir** - ディレクトリの絶対パス
- **regex incl** - 含めるエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する正規表現。空の場合はすべて含める (デフォルト値)
- **regex excl** - 除外するエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する正規表現。空の場合は除外しない (デフォルト値)
- **mode** - *apparent*(デフォルト): ディスク使用量ではなく見かけ上のファイルサイズを取得 (du -sb dir と同等)、*disk*: ディスク使用量を取得 (du -s -B1 dir と同等)。du コマンドと異なり、vfs.dir.size アイテムはディレクトリサイズ計算時に隠しファイル

も考慮します (dir 内で `du -sb .[^.]* *` と同等)。
- **max depth** - サブディレクトリをたどる最大深度: **-1**(デフォルト): 無制限、**0**: サブディレクトリをたどらない
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。ディレクトリの内容すべてが除外されます (`regex excl` とは異なります)

コメント:

- *zabbix* ユーザーに少なくとも読み取り権限があるディレクトリのみが計算されます。読み取り権限のみの場合はディレクトリ自体のサイズのみ計算されます。読み取り・実行権限がある場合は内容も含めて計算されます。
- 大きなディレクトリや遅いドライブの場合、agentやserver/proxyの設定ファイルの Timeout 設定によりタイムアウトする場合があります。必要に応じてタイムアウト値を増やしてください。
- ファイルサイズの上限は大容量ファイルサポートに依存します。

例:

```
vfs.dir.size[/tmp,log] # 名前に'log'を含む/tmp 内のすべてのファイルのサイズを計算
vfs.dir.size[/tmp,log,^.+\.old$] # 名前に'log'を含み、かつ名前が'.old'で終わるファイルを除外した/tmp 内のすべてのファ
```

vfs.file.cksum[file,<mode>]

 UNIX の cksum アルゴリズムで計算されたファイルのチェックサム。
 戻り値: *crc32* の場合は整数、*md5* または *sha256* の場合は文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス
- **mode** - *crc32* (デフォルト)、*md5*、または *sha256*

ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.cksum[/etc/passwd]
```

返される値の例 (それぞれ crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 ファイルの内容を取得します[7]。
 戻り値: テキスト。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス
- **encoding** - コードページの識別子

コメント:

- 戻り値は 16MB に制限されます（末尾の空白も含み、切り捨てられます)。データベースの制限も適用されます。
- ファイルが空、または LF/CR 文字のみの場合は空文字列が返されます。
- バイトオーダーマーク（BOM）は出力から除外されます。

例:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 ファイルの存在を確認します。
 戻り値: 0 - 見つからない; 1 - 指定したタイプのファイルが存在する。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス
- **types incl** - 含めるファイルタイプのリスト。指定可能な値: *file* (通常のファイル、types_excl が設定されていない場合のデフォルト)、*dir* (ディレクトリ)、*sym* (シンボリックリンク)、*sock* (ソケット)、*bdev* (ブロックデバイス)、*cdev* (キャラクタデバイス)、*fifo* (FIFO)、*dev* (”bdev,cdev” と同義)、*all* (上記すべてのタイプ、types_excl が設定されている場合のデフォルト)
- **types excl** - 除外するファイルタイプのリスト。指定可能な値は types_incl を参照 (デフォルトでは除外されるタイプはありません)

コメント:

- 複数のタイプはカンマで区切り、全体をダブルクォーテーション”” で囲みます

- <types_incl> と <types_excl> の両方に同じタイプがある場合、そのタイプのファイルは除外されます
- ファイルサイズの制限は大きなファイルのサポートに依存します

例:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 ファイルに関する情報を返します。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス

コメント:

- UNIX 系システムでサポートされているファイルタイプ: 通常ファイル、ディレクトリ、シンボリックリンク、ソケット、ブロックデバイス、キャラクタデバイス、FIFO。
- ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.get[/etc/passwd] #/etc/passwd ファイルの情報 (type, user, permissions, SID, uid など) を含む JSON を返
```

vfs.file.md5sum[file]

 ファイルの MD5 チェックサム。
 戻り値: 文字列 (ファイルの MD5 ハッシュ)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス

ファイルサイズの制限は、大きなファイルのサポートに依存します。

例:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

返される値の例:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 ファイルの所有者を取得します。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス
- **ownertype** - *user* (デフォルト) または *group* (Unix のみ)
- **resulttype** - *name* (デフォルト) または *id*; id の場合は Unix では uid/gid、Windows では SID を返します。

ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.owner[/tmp/zabbix_server.log] #/tmp/zabbix_server.log のファイル所有者を返す
vfs.file.owner[/tmp/zabbix_server.log,,id] #/tmp/zabbix_server.log のファイル所有者 ID を返す
```

vfs.file.permissions[file]

 UNIX パーミッションを 8 進数で表した 4 桁の文字列を返します。
 戻り値: 文字列。
 対応プラットフォーム: Linux。他の UNIX 系プラットフォームでも動作する場合があります。

パラメータ:

- **file** - ファイルのフルパス

ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.permissions[/etc/passwd] #/etc/passwd のパーミッションを返します。例えば'0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 ファイル内の文字列を取得します[7]。
 戻り値: 一致した文字列を含む行、または省略可能な output パラメータで指定された値。
 対応プラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス
- **regexp** - 必要なパターンを記述した正規表現
- **encoding** - コードページ識別子
- **start line** - 検索を開始する最初の行番号 (デフォルトはファイルの最初の行)
- **end line** - 検索を終了する最後の行番号 (デフォルトはファイルの最後の行)
- **output** - 省略可能な出力フォーマットテンプレート **\0** エスケープシーケンスは一致したテキスト部分 (一致が始まる最初の文字から一致が終わる文字まで) に置き換えられ、**\N** (N=1...9) エスケープシーケンスは N 番目に一致したグループ (キャプチャグループ数を超える場合は空文字列) に置き換えられます。

コメント:

- ファイルサイズの制限は大きなファイルのサポートに依存します。
- 最初に一致した行のみが返されます。
- 式に一致する行がない場合は空文字列が返されます。
- バイトオーダーマーク (BOM) は出力から除外されます。
- output パラメータによる内容抽出はエージェント側で行われます。

例:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] → ユーザー *zabbix* の ID を取得
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 ファイル内で文字列を検索します[7]。
 戻り値: 0 - 一致しない; 1 - 一致。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス
- **regexp** - 必要なパターンを記述する正規表現
- **encoding** - コードページの識別子
- **start line** - 検索を開始する最初の行番号 (デフォルトはファイルの最初の行)
- **end line** - 検索を終了する最後の行番号 (デフォルトはファイルの最後の行)

コメント:

- ファイルサイズの制限は大きなファイルのサポートに依存します。
- バイトオーダーマーク (BOM) は無視されます。

例:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 ファイルサイズ（バイト単位）を返します。
 戻り値：整数。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス
- **mode** - *bytes*（デフォルト）または *lines*（空行もカウントされます）

コメント:

- ファイルには *zabbix* ユーザーの読み取り権限が必要です
- ファイルサイズの上限は大きなファイルのサポートに依存します

例:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 ファイルの時刻情報。
 戻り値: 整数 (Unix タイムスタンプ)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルのフルパス
- **mode** - 可能な値:
modify (デフォルト) - ファイル内容が最後に変更された時刻
access - ファイルが最後に読み込まれた時刻
change - ファイルプロパティが最後に変更された時刻

ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 ファイルシステムのタイプとマウントオプションを含む、マウントされたファイルシステムのリスト。ローレベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD。

vfs.fs.get

 マウントされているファイルシステムのリストと、そのタイプ、利用可能なディスク容量、inode 統計、マウントオプションを取得します。低レベルディスカバリに使用できます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD。

コメント:

- inode 数がゼロのファイルシステム（btrfs などの動的 inode を持つファイルシステムの場合）も報告されます。
- 参考: マウントされたファイルシステムのディスカバリ。

vfs.fs.inode[fs,<mode>]

 inode の数またはパーセンテージ。
 戻り値: *Integer* - 数値の場合; *Float* - パーセンテージの場合。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム;
- **mode** - 可能な値: *total* (デフォルト), *free*, *used*, *pfree* (free, パーセンテージ), *pused* (used, パーセンテージ)。

inode 数がゼロの場合（例えば btrfs のような動的 inode を持つファイルシステム）、pfree/pused の値はそれぞれ”100” と”0” として報告されます。

例:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 ディスク容量をバイト単位または全体に対するパーセンテージで取得します。
 戻り値: バイトの場合は *Integer*、パーセンテージの場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム;
- **mode** - *total* (デフォルト), *free*, *used*, *pfree* (空き容量のパーセンテージ), *pused* (使用容量のパーセンテージ) のいずれか。

コメント:

- ファイルシステムがマウントされていない場合、マウントポイントが存在するローカルファイルシステムのサイズを返します;
- ファイルシステムの予約領域は考慮され、*free* モードを使用した場合は含まれません。

例:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 バイト単位または合計に対するパーセンテージでのメモリサイズ。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。
 サポートされているプラットフォームを参照。

パラメータ:

- **mode** - 指定可能な値: *total* (デフォルト), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (使用中, パーセンテージ), *available*, *pavailable* (利用可能, パーセンテージ)。

コメント:

- このアイテムは 3 つのカテゴリのパラメータを受け付けます:
1) *total* - メモリの合計量
2) プラットフォーム固有のメモリタイプ: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*
3) 使用中および利用可能なメモリのユーザーレベルの推定値: *used*, *pused*, *available*, *pavailable*
- *active* モードパラメータは FreeBSD、HP-UX、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *anon*、*exec*、*file* モードパラメータは NetBSD でのみサポートされています。
- *buffers* モードパラメータは Linux、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- *cached* モードパラメータは Linux、FreeBSD、AIX、OpenBSD、NetBSD でのみサポートされています。
- *inactive*、*wired* モードパラメータは FreeBSD、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *pinned* モードパラメータは AIX でのみサポートされています。
- *shared* モードパラメータは Linux 2.4、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- このアイテムの追加詳細も参照してください。

例:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>]

 Web ページの内容を取得します。
 戻り値: テキストとしての Web ページのソース (ヘッダーを含む)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL(`scheme://host:port/path`、*host* のみ必須)。許可される URL スキーム: *http*、*https*[4]。スキームが省略された場合は *http* として扱われます。URL が指定された場合、`path` と `port` は空でなければなりません。認証が必要なサーバーに接続する際にユーザー名/パスワードを指定すること (例: `http://user:password@www.example.com`) は、cURL サポートがある場合のみ可能です[4]。ホスト名でPunycodeがサポートされています。
- **path** - HTML ドキュメントへのパス (デフォルトは/)。
- **port** - ポート番号 (HTTP の場合はデフォルトで 80)

コメント:

- `host` で指定されたリソースが存在しないか利用できない場合、このアイテムはサポート外になります。
- `host` はホスト名、ドメイン名、IPv4 または IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合は、Zabbix エージェントが IPv6 サポートを有効にしてコンパイルされている必要があります。

例:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
```

web.page.perf[host,<path>,<port>]

 Web ページ全体の読み込み時間（秒単位）。
 戻り値：*Float*。
 サポートされているプラットフォームを参照。

パラメータ:

- **host** - ホスト名または URL(`scheme://host:port/path` の形式、*host* のみ必須)。許可される URL スキーム：*http*、*https*[4]。スキームが省略された場合は *http* として扱われます。URL を指定する場合、`path` と `port` は空でなければなりません。認証が必要なサーバーに接続する際にユーザー名/パスワードを指定するには、例えば `http://user:password@www.example.com` のように、cURL サポートが必要です[4]。ホスト名で Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは/）；
- **port** - ポート番号（HTTP のデフォルトは 80）

コメント:

- `host` で指定されたリソースが存在しない、または利用できない場合、このアイテムは未サポートになります。
- `host` にはホスト名、ドメイン名、IPv4 または IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合は、Zabbix エージェントを IPv6 サポート有効でコンパイルする必要があります。

例:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Web ページ上の文字列を検索します。
 戻り値：一致した文字列、または省略可能な `output` パラメータで指定された値。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（`scheme://host:port/path` の形式、*host* のみ必須）。許可される URL スキーム：*http*、*https*[4]。スキームが省略された場合は *http* として扱われます。URL を指定する場合、`path` と `port` は空にする必要があります。認証が必要なサーバーに接続する際にユーザー名/パスワードを指定するには、例えば `http://user:password@www.example.com` のように指定しますが、これは cURL サポートが有効な場合のみ可能です[4]。ホスト名で Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは/）；
- **port** - ポート番号（HTTP のデフォルトは 80）
- **regexp** - 必要なパターンを記述する正規表現;
- **length** - 返す最大文字数;
- **output** - 省略可能な出力フォーマットテンプレート。**\0** エスケープシーケンスは一致したテキスト部分（マッチが始まる最初の文字からマッチが終わる文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループ（または N がキャプチャグループ数を超える場合は空文字列）に置き換えられます。

コメント:

- `host` で指定されたリソースが存在しないか利用できない場合、このアイテムは未サポートになります。
- `host` にはホスト名、ドメイン名、IPv4 または IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合は Zabbix エージェントが IPv6 サポートを有効にしてコンパイルされている必要があります。
- `output` パラメータを使用したコンテンツ抽出はエージェント側で行われます。

例:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 エージェントのホストメタデータ。
 戻り値: 文字列。
 サポートされているプラットフォームを参照。

HostMetadataまたはHostMetadataItemパラメータの値を返します。どちらも定義されていない場合は空文字列を返します。

agent.hostname

 エージェントのホスト名。
 戻り値: *String*。
 サポートされているプラットフォームを参照。

戻り値:

- パッシブチェックの場合 - エージェント設定ファイルのHostnameパラメータにリストされている最初のホスト名
- アクティブチェックの場合 - 現在のホスト名

agent.ping

 エージェントの可用性チェック。
 戻り値: なし - 利用不可; 1 - 利用可能。
 サポートされているプラットフォームを参照してください。

ホストの利用不可を確認するには、**nodata()** トリガー関数を使用します。

agent.variant

 Zabbix エージェントのバリアント（Zabbix agent または Zabbix agent 2）。
 戻り値: 1 - Zabbix agent; 2 - Zabbix agent 2。
 サポートされているプラットフォームを参照。

agent.version

 Zabbix エージェントのバージョン。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

返される値の例:

```
6.0.3
```

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

 Zabbix サーバーまたはプロキシの内部メトリクスのセットをリモートで返します。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** - リモートでクエリを実行するサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）；
- **port** - リモートでクエリを実行するサーバー/プロキシのポート（デフォルトは 10051）

コメント:

- このアイテムによって選択された内部メトリクスのセットが返されます。詳細については、Zabbix 統計情報のリモート監視を参照してください;
- stats リクエストは、ターゲットインスタンスの'StatsAllowedIP' server/proxyパラメータにリストされているアドレスからのみ受け付けられることに注意してください。

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

 Zabbix サーバーまたはプロキシで遅延しているキュー内の監視項目数をリモートで返します。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** - リモートでクエリするサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）
- **port** - リモートでクエリするサーバー/プロキシのポート（デフォルトは 10051）
- **queue** - 定数（そのまま使用）
- **from** - 少なくともこの秒数遅延（デフォルトは 6 秒）
- **to** - 最大でこの秒数遅延（デフォルトは無限大）

stats リクエストは、ターゲットインスタンスのserver/proxyパラメータの'StatsAllowedIP' にリストされているアドレスからのみ受け付けられることに注意してください。

脚注

[1]Linux 固有の注意事項です。Zabbix エージェントはファイルシステム*/proc* への読み取り専用アクセス権を持っている必要があります。www.grsecurity.org のカーネルパッチは、非特権ユーザーのアクセス権を制限します。

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: アイテム値が 3 時間以上アクセスされない場合、Zabbix エージェントは「古い」デバイス接続を終了します。これは、システムに動的にパスが変更されるデバイスがある場合や、デバイスが手動で削除された場合に発生する可能性があります。また、これらのアイテムが 3 時間以上の更新間隔を使用している場合、常に「0」を返すことにも注意してください。

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: 最初のパラメータにデフォルトの *all* を使用した場合、キーは sda、sdb などのすべてのブロックデバイスと、それらのパーティション（sda1、sda2、sdb3...）、およびそれらのブロックデバイス/パーティションに基づく複数デバイス（MD raid）、論理ボリューム（LVM）を含む集計統計を返します。このような場合、返される値は絶対値ではなく、相対値（時間とともに変動する値）としてのみ考慮する必要があります。

[4] SSL（HTTPS）は、エージェントが cURL サポート付きでコンパイルされている場合のみサポートされます。それ以外の場合、アイテムは未サポートになります。

[5] Solaris 10 6/06 までの Solaris システムでは、ループバックインターフェースの `bytes` および `errors` 値はサポートされていません。これは、バイト、エラー、および使用率の統計がカーネルによって保存または報告されていないためです。ただし、net-snmp 経由で Solaris システムを監視している場合、値が返されることがあります。これは、net-snmp が 1997 年まで遡る cmu-snmp のレガシーコードを保持しており、インターフェース統計からバイト値の読み取りに失敗した場合、パケットカウンタ（ループバックインターフェースには存在する）に任意の値 308 を掛けたものを返すためです。これは、パケットの平均長が 308 オクテットであると仮定していますが、Solaris システムのループバックインターフェースの MTU 制限は 8892 バイトであるため、非常に大まかな推定値です。これらの値が正しい、または正確であると仮定すべきではありません。あくまで推測値です。Zabbix エージェントは推測を行いませんが、net-snmp はこれらのフィールドに値を返します。

[6] Solaris で/proc/pid/psinfo から取得されるコマンドラインは 80 バイトに制限されており、プロセスが開始されたときのコマンドラインが含まれます。

[7] `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` アイテムはファイルの内容を取得するために使用できます。機密情報を含む特定のファイルへのアクセスを制限したい場合は、これらのファイルの閲覧権限を持たないユーザーで Zabbix エージェントを実行してください。

コマンドラインユーティリティでの使用

zabbix_agentd や zabbix_get でアイテムキーをコマンドラインからテストまたは使用する場合、シェルの構文も考慮する必要があります。

たとえば、キーの特定のパラメータをダブルクォーテーションで囲む必要がある場合、ダブルクォーテーションを明示的にエスケープする必要があります。そうしないと、シェルによって特殊文字としてトリムされ、Zabbix ユーティリティに渡されません。

例:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

エンコーディング設定

取得したデータが破損しないように、チェック（例：'vfs.file.contents'）の処理に正しいエンコーディングを `encoding` パラメータで指定することができます。サポートされているエンコーディング（コードページ識別子）の一覧は、libiconv（GNU プロジェクト）

のドキュメントや、Microsoft Windows SDK の" コードページ識別子"のドキュメントで確認できます。Microsoft は一部のコードページを「マネージアプリケーションでのみ利用可能」としている場合があります。これは、.NET ランタイム経由でのみ公開され、ネイティブ Win32 API では利用できないことを意味します。Zabbix エージェントは独自のエンコーディング変換ロジックを実装しているため、ネイティブ Windows 関数が提供しない場合でも、これらのコードページは Zabbix エージェントでサポートされています。

encoding パラメータでエンコーディングが指定されていない場合、以下の解決戦略が適用されます：

- エンコーディングが指定されていない（または空文字列）の場合、UTF-8 とみなされ、データは「そのまま」処理されます。
- BOM 解析 - アイテム'vfs.file.contents'、'vfs.file.regexp'、'vfs.file.regmatch' に適用されます。ファイルの先頭にあるバイトオーダーマーク（BOM）を使用して正しいエンコーディングを判定しようとします。BOM が存在しない場合は、標準の解決方法（上記参照）が適用されます。

エージェントアイテムのトラブルシューティング

パッシブチェックの場合、サーバーからエージェントへのリクエストが先にタイムアウトしてアイテムが値を取得できなくなるのを防ぐために、以下の点に注意してください。

- エージェントのバージョンがサーバーのバージョンより古い場合、アイテムの設定の *Timeout* 値（またはグローバルタイムアウト）を、エージェントの設定ファイルの `Timeout` 値より大きくする必要があります。
- エージェントのバージョンがサーバーのバージョンより新しい場合、サーバーの設定ファイルの `Timeout` 値を、エージェントの設定ファイルの `Timeout` 値より大きくする必要があります。

1 Zabbix エージェント 2

Zabbix agent 2 は、UnixおよびWindowsの Zabbix agent でサポートされているすべてのアイテムキーをサポートしています。
 このページでは、Zabbix agent 2 でのみ使用できる追加のアイテムキーについて、プラグインごとにグループ化して説明します。

アイテムキーはパラメータや追加情報なしで一覧表示されています。
 アイテムキーをクリックすると、詳細が表示されます。

アイテムキー	説明	プラグイン
ceph.df.details	クラスタのデータ使用量とプール間の分布。	Ceph
ceph.osd.stats	集約および OSD ごとの統計。	
ceph.osd.discovery	検出された OSD のリスト。	
ceph.osd.dump	OSD の使用しきい値とステータス。	
ceph.ping	Ceph への接続が確立できるかどうかをテストします。	
ceph.pool.discovery	検出されたプールのリスト。	
ceph.status	クラスタ全体のステータス。	
docker.container_info	コンテナに関する低レベル情報。	Docker
docker.container_stats	コンテナのリソース使用統計。	
docker.containers	コンテナのリストを返します。	
docker.containers.discovery	コンテナのリストを返します。低レベルディスカバリ用。	
docker.data.usage	現在のデータ使用量に関する情報。	
docker.images	イメージのリストを返します。	
docker.images.discovery	イメージのリストを返します。低レベルディスカバリ用。	
docker.info	システム情報。	
docker.ping	Docker デーモンが生きているかどうかをテストします。	
ember.get	要求されたデバイスの結果を返します。	Ember+
memcached.ping	接続が生きているかどうかをテストします。	Memcached
memcached.stats	STATS コマンドの出力を取得します。	
mongodb.collection.stats	指定されたコレクションのさまざまなストレージ統計を返します。	MongoDB
mongodb.collections.discovery	検出されたコレクションのリストを返します。	
mongodb.collections.usage	コレクションの使用統計を返します。	
mongodb.connpool.stats	現在のデータベースインスタンスからシャードクラスタまたはレプリカセットの他のメンバーへのオープンな発信接続に関する情報を返します。	
mongodb.db.stats	指定されたデータベースシステムの状態を反映する統計を返します。	
mongodb.db.discovery	検出されたデータベースのリストを返します。	
mongodb.jumbo_chunks.count	ジャンボチャンクの数を返します。	
mongodb.oplog.stats	oplog から取得したデータを使用して、レプリカセットのステータスを返します。	
mongodb.ping	接続が生きているかどうかをテストします。	
mongodb.rs.config	レプリカセットの現在の構成を返します。	
mongodb.rs.status	メソッドが実行されたメンバーの観点からレプリカセットのステータスを返します。	
mongodb.server.status	データベースの状態を返します。	
mongodb.sh.discovery	クラスタ内に存在する検出されたシャードのリストを返します。	
mongodb.version	データベースサーバのバージョンを返します。	

アイテムキー	説明	プラグイン
mqtt.get	指定されたブローカーの特定のトピックまたはワイルドカード付きトピックにサブスクライブし、パブリケーションを待ちます。	MQTT
mssql.availability.group.get	可用性グループを返します。	MSSQL
mssql.custom.query	カスタムクエリの結果を返します。	
mssql.db.get	利用可能なすべての MSSQL データベースを返します。	
mssql.job.status.get	ジョブのステータスを返します。	
mssql.last.backup.get	すべてのデータベースの最終バックアップ時刻を返します。	
mssql.local.db.get	Always On 可用性グループおよびレプリカ（プライマリまたはセカンダリ）に参加し、接続が確立されたサーバーに存在するデータベースを返します。	
mssql.mirroring.get	ミラーリング情報を返します。	
mssql.nonlocal.db.get	Always On 可用性グループおよびレプリカ（プライマリまたはセカンダリ）に参加し、他のサーバーに存在するデータベース（接続が確立された SQL Server インスタンスにローカルでないデータベース）を返します。	
mssql.perfcounter.get	パフォーマンスカウンタを返します。	
mssql.ping	接続が生きているかどうかをテストします。	
mssql.quorum.get	クォーラム情報を返します。	
mssql.quorum.member.get	クォーラムメンバーを返します。	
mssql.replica.get	レプリカを返します。	
mssql.version	MSSQL のバージョンを返します。	
mysql.custom.query	カスタムクエリの結果を返します。	MySQL
mysql.db.discovery	MySQL データベースのリストを返します。	
mysql.db.size	データベースサイズ（バイト単位）。	
mysql.get_status_variables	グローバルステータス変数の値。	
mysql.ping	接続が生きているかどうかをテストします。	
mysql.replication.discovery	MySQL レプリケーションのリストを返します。	
mysql.replication.get_slave_status	レプリケーションステータス。	
mysql.version	MySQL のバージョン。	
net.dns.get	DNS クエリを実行し、詳細な DNS レコード情報を返します。	Network
nvml.device.count	GPU デバイスの数。	NVIDIA GPU
nvml.device.decoder.utilization	GPU デバイスのデコーダ使用率。	
nvml.device.ecc.mode	GPU デバイスの ECC モード。	
nvml.device.encoder.stats.get	GPU デバイスのエンコーダ統計。	
nvml.device.encoder.utilization	GPU デバイスのエンコーダ使用率。	
nvml.device.energy.consumption	GPU デバイスの消費電力。	
nvml.device.errors.memory	GPU デバイスの ECC メモリエラー情報。	
nvml.device.errors.register	GPU デバイスの ECC レジスタエラー情報。	
nvml.device.fan.speed.avg	GPU デバイスのファン速度の平均。	
nvml.device.get	UUID と名前で検出された GPU。	
nvml.device.graphics.frequency	GPU デバイスのグラフィックスクロック速度。	
nvml.device.memory.bar1.get	GPU デバイスの BAR1 メモリ情報。	
nvml.device.memory.fb.get	GPU デバイスのフレームバッファメモリ情報。	
nvml.device.memory.frequency	GPU デバイスのメモリクロック速度。	
nvml.device.pci.utilization	GPU デバイスの PCI 使用率。	
nvml.device.performance.state	GPU デバイスのパフォーマンスステート。	
nvml.device.power.limit	GPU デバイスの電力制限。	
nvml.device.power.usage	GPU デバイスの電力使用量。	
nvml.device.serial	GPU デバイスのシリアル番号。	
nvml.device.sm.frequency	GPU デバイスのストリーミングマルチプロセッサクロック速度。	
nvml.device.temperature	GPU デバイスの温度。	
nvml.device.utilization	GPU デバイスの使用統計。	
nvml.device.video.frequency	GPU デバイスのビデオクロック速度。	
nvml.system.driver.version	NVIDIA ドライバのバージョン。	
nvml.version	NVML ライブラリのバージョン。	
oracle.diskgroups.stats	自動ストレージ管理（ASM）ディスクグループの統計を返します。	Oracle
oracle.diskgroups.discovery	ASM ディスクグループのリストを返します。	
oracle.archive.info	アーカイブログの統計。	
oracle.cdb.info	コンテナデータベース（CDB）の情報。	
oracle.custom.query	カスタムクエリの結果。	
oracle.datafiles.stats	データファイルの統計を返します。	
oracle.db.discovery	データベースのリストを返します。	

アイテムキー	説明	プラグイン
oracle.fra.stats	高速リカバリア（FRA）の統計を返します。	
oracle.instance.info	インスタンスの統計。	
oracle.pdb.info	プラガブルデータベース（PDB）の情報。	
oracle.pdb.discovery	PDB のリストを返します。	
oracle.pga.stats	プログラムグローバルエリア（PGA）の統計を返します。	
oracle.ping	Oracle への接続が確立できるかどうかをテストします。	
oracle.proc.stats	プロセスの統計を返します。	
oracle.redolog.info	制御ファイルからのログファイル情報。	
oracle.sga.stats	システムグローバルエリア（SGA）の統計を返します。	
oracle.sessions.stats	セッションの統計を返します。	
oracle.sys.metrics	システムメトリック値のセットを返します。	
oracle.sys.params	システムパラメータ値のセットを返します。	
oracle.ts.stats	表領域の統計を返します。	
oracle.ts.discovery	表領域のリストを返します。	
oracle.user.info	Oracle ユーザー情報を返します。	
oracle.version	データベースサーバのバージョンを返します。	
pgsql.autovacuum.count	自動バキュームワーカーの数。	PostgreSQL
pgsql.archive	アーカイブされたファイルに関する情報。	
pgsql.bgwriter	チェックポイントタイプごとに分類された、データベースクラスタのチェックポイントの合計数。	
pgsql.cache.hit	PostgreSQL バッファキャッシュヒット率。	
pgsql.connections	タイプ別の接続数を返します。	
pgsql.custom.query	カスタムクエリの結果を返します。	
pgsql.db.age	データベースの最も古い FrozenXID の経過時間。	
pgsql.db.bloating_tables	データベースごとの膨張テーブル数。	
pgsql.db.discovery	PostgreSQL データベースのリスト。	
pgsql.db.size	データベースサイズ（バイト単位）。	
pgsql.dbstat	データベースごとの統計を収集します。	
pgsql.dbstat.sum	クラスタ内のすべてのデータベースの集計データ。	
pgsql.locks	データベースごとに付与されたロックに関する情報。	
pgsql.oldest.xid	最も古い XID の経過時間。	
pgsql.ping	接続が生きているかどうかをテストします。	
pgsql.queries	実行時間ごとのクエリメトリック。	
pgsql.replication.count	スタンバイサーバの数。	
pgsql.replication.process	各送信プロセスごとのフラッシュラグ、書き込みラグ、リプレイラグ。	
pgsql.replication.process.discovery	レプリケーションプロセス名のディスカバリ。	
pgsql.replication.recovery_role	リカバリステータス。	
pgsql.replication.status	レプリケーションのステータス。	
pgsql.replication_lag.b	レプリケーションラグ（バイト単位）。	
pgsql.replication_lag.sec	レプリケーションラグ（秒単位）。	
pgsql.uptime	PostgreSQL の稼働時間（ミリ秒単位）。	
pgsql.version	PostgreSQL のバージョンを返します。	
pgsql.wal.stat	WAL 統計。	
redis.config	パターンに一致する Redis インスタンスの構成パラメータを取得します。	Redis
redis.info	INFO コマンドの出力を取得します。	
redis.ping	接続が生きているかどうかをテストします。	
redis.slowlog.count	Redis 起動以降のスローログエントリ数。	
smart.attribute.discovery	S.M.A.R.T. デバイス属性のリストを返します。	S.M.A.R.T.
smart.disk.discovery	S.M.A.R.T. デバイスのリストを返します。	
smart.disk.get	S.M.A.R.T. デバイスの利用可能なすべてのプロパティを返します。	
systemd.unit.get	systemd ユニットのすべてのプロパティを返します。	Systemd
systemd.unit.info	systemd ユニット情報。	
systemd.unit.discovery	systemd ユニットのリストとその詳細。	
web.certificate.get	証明書を検証し、証明書の詳細を返します。	Web certificates

関連項目:

- 組み込みプラグイン
- ロード可能プラグイン

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

ceph.df.details[connString,<user>,<apikey>]

 クラスタのデータ使用量とプール間の分布。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.osd.stats[connString,<user>,<apikey>]

 集計および OSD ごとの統計情報。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.osd.discovery[connString,<user>,<apikey>]

 検出された OSD のリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.osd.dump[connString,<user>,<apikey>]

 OSD の使用しきい値とステータス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.ping[connString,<user>,<apikey>]

 Ceph への接続が確立できるかどうかをテストします。
 戻り値: *0* - 接続が切断されています（AUTH や設定の問題など、エラーが発生した場合）; *1* - 接続が成功しています。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.pool.discovery[connString,<user>,<apikey>]

 検出されたプールのリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

ceph.status[connString,<user>,<apikey>]

 クラスタ全体のステータス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Ceph のログイン認証情報。

docker.container_info[<ID>,<info>]

 コンテナのローレベル情報。
 戻り値: ContainerInspect API 呼び出しの出力を JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前;
- **info** - 返される情報の量。サポートされる値: *short* (デフォルト) または *full*。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.container_stats[<ID>]

 コンテナのリソース使用状況の統計情報です。
 戻り値: ContainerStats API 呼び出しの出力と CPU 使用率のパーセンテージを JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前

Agent 2 のユーザー ('zabbix') は、十分な権限を得るために'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.containers

 コンテナのリスト。
 戻り値: ContainerList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.containers.discovery[<options>]

 コンテナのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **options** - すべてのコンテナまたは実行中のコンテナのみをディスカバリするかどうかを指定します。サポートされる値: *true* - すべてのコンテナを返す; *false* - 実行中のコンテナのみを返す (デフォルト)。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.data.usage

 現在のデータ使用量に関する情報。
 戻り値: SystemDataUsage API 呼び出しの出力を JSON としてシリアル化したもの。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.images

 イメージのリストを返します。
 戻り値: ImageList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.images.discovery

 イメージのリストを返します。ローレベルディスカバリで使用されます。
 戻り値 : *JSON* オブジェクト

Agent 2 のユーザー（'zabbix'）は、十分な権限を持つために'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.info

 システム情報。
 戻り値: SystemInfo API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.ping

 Docker デーモンが生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

ember.get[<uri>,<path>]

 要求されたデバイスの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - Ember+ デバイスの URI。デフォルト: 127.0.0.1:9998;
- **path** - デバイスの OID パス。デフォルトは空で、ルートコレクションデータを返します。

memcached.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切断されている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Memcached のログイン認証情報。

memcached.stats[connString,<user>,<password>,<type>]

 STATS コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Memcached のログイン認証情報;
- **type** - 返される統計情報のタイプ: *items*、*sizes*、*slabs*、*settings*（デフォルトは空、一般的な統計情報を返します）。

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 指定したコレクションのさまざまなストレージ統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報;
- **database** - データベース名（デフォルト: admin）;
- **collection** - コレクション名。

mongodb.collections.discovery[connString,<user>,<password>]

 検出されたコレクションのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.collections.usage[connString,<user>,<password>]

 コレクションの使用統計を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ :

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.connpool.stats[connString,<user>,<password>]

 現在のデータベースインスタンスからシャードクラスタまたはレプリカセットの他のメンバーへのオープンアウトゴーイング接続に関する情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報;
- **database** - データベース名 (デフォルト: admin);
- **collection** - コレクション名。

mongodb.db.stats[connString,<user>,<password>,<database>]

 指定したデータベースのシステム状態を反映する統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報;
- **database** - データベース名（デフォルト: admin）。

mongodb.db.discovery[connString,<user>,<password>]

 検出されたデータベースのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 ジャンボチャンクの数を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.oplog.stats[connString,<user>,<password>]

 oplog から取得したデータを使用して、レプリカセットのステータスを返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.config[connString,<user>,<password>]

 レプリカセットの現在の構成を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.status[connString,<user>,<password>]

 このメソッドが実行されるメンバーの観点からレプリカセットのステータスを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.server.status[connString,<user>,<password>]

 データベースの状態を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.sh.discovery[connString,<user>,<password>]

 クラスタ内に存在する検出されたシャードのリストを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.version[connString,<user>,<password>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mqtt.get[<broker url>,topic,<user>,<password>]

 指定されたブローカーの特定のトピックまたはワイルドカードを含むトピックにサブスクライブし、パブリケーションを待ちます。
 戻り値: トピックの内容による。ワイルドカードが使用されている場合、トピックの内容を JSON で返します。

パラメータ:

- **broker url** - クエリパラメータなしの `protocol://host:port` 形式の MQTT ブローカー URL（サポートされているプロトコル: `tcp`, `ssl`, `ws`）。値が指定されていない場合、エージェントは `tcp://localhost:1883` を使用します。プロトコルまたはポートが省略された場合、デフォルトのプロトコル（`tcp`）またはポート（1883）が使用されます。
- **topic** - MQTT トピック（必須）。ワイルドカード（+、#）がサポートされています。
- **user, password** - 認証情報（必要な場合）。

コメント:

- アイテムはアクティブチェック（「Zabbix エージェント（アクティブ）」アイテムタイプ）として設定する必要があります。
- TLS 暗号化証明書は、デフォルトの場所（例: Ubuntu の場合は `/etc/ssl/certs/` ディレクトリ）に保存することで使用できます。TLS を使用する場合は、`tls://` スキームを使用してください。

mssql.availability.group.get[URI,<user>,<password>]

 可用性グループを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされているスキーマは `sqlserver://`のみ）。埋め込み認証情報は無視されます。URI の一部としてインスタンス名を指定することができます。例: `sqlserver://localhost/InstanceName`（ポートなし）。ポートが指定されている場合、インスタンス名は無視されます;
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされているスキーマは `sqlserver://`のみ）。埋め込まれた認証情報は無視されます。URI の一部としてインスタンス名を指定することができます。例: `sqlserver://localhost/InstanceName`（ポートなし）。ポートが指定されている場合、インスタンス名は無視されます;
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード;
- **queryName** - `.sql` 拡張子なしで `Plugins.MSSQL.CustomQueriesDir` に設定されたカスタムクエリの名前;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

詳細はMSSQL プラグインの readme を参照してください。

mssql.db.get

 利用可能なすべての MSSQL データベースを返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.job.status.get

 ジョブのステータスを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.last.backup.get

 すべてのデータベースの最終バックアップ時刻を返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.local.db.get

 Always On 可用性グループおよびレプリカ (プライマリまたはセカンダリ) に参加しており、接続が確立されたサーバーに存在するデータベースを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグイン の readme を参照してください。

mssql.mirroring.get

 ミラーリング情報を返します。
 戻り値: *JSON* オブジェクト。

詳細は MSSQL プラグイン の readme を参照してください。

mssql.nonlocal.db.get

 Always On 可用性グループに参加しているデータベースおよび他のサーバーにあるレプリカ (プライマリまたはセカンダリ) を返します (接続が確立された SQL Server インスタンスにローカルではないデータベース)。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグイン の readme を参照してください。

mssql.perfcounter.get

 パフォーマンスカウンターを返します。
 戻り値: *JSON* オブジェクト。

詳細は MSSQL プラグイン の readme を参照してください。

mssql.ping

 データベースに ping を送信します。接続が正しく設定されているかテストします。
 戻り値: *1* - 生存, *0* - 非生存。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.quorum.get

 クォーラム情報を返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.quorum.member.get

 クォーラムメンバーを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.replica.get

 レプリカを返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.version

 MSSQL のバージョンを返します。
 戻り値: 文字列。

詳細については、MSSQL プラグインの readme を参照してください。

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **queryName** - カスタムクエリの名前。SQL ファイル名 (拡張子なし) と一致する必要があります;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

詳細はMySQL プラグインの readme を参照してください。

mysql.db.discovery[connString,<user>,<password>]

 MySQL データベースのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: LLD JSON 形式の”show databases” SQL クエリの結果。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.db.size[connString,<user>,<password>,<database name>]

 バイト単位のデータベースサイズ。
 戻り値: バイト単位で特定のデータベースに対して”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” SQL クエリの結果。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **database name** - データベース名。

mysql.get_status_variables[connString,<user>,<password>]

 グローバルステータス変数の値。
 戻り値: ”show global status” SQL クエリの結果を JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切断されている（AUTH や設定の問題を含むエラーが発生した場合）。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.replication.discovery[connString,<user>,<password>]

 MySQL レプリケーションのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: `SHOW SLAVE STATUS` または `SHOW REPLICA STATUS` SQL クエリの結果を LLD JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 レプリケーションのステータス。
 戻り値: `SHOW SLAVE STATUS` または `SHOW REPLICA STATUS` SQL クエリの結果を JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **master host** - レプリケーションマスターホスト名。見つからない場合はエラーを返します。このパラメータが指定されていない場合は、すべてのホストが返されます。

mysql.version[connString,<user>,<password>]

 MySQL のバージョン。
 戻り値: *String* (MySQL インスタンスのバージョン)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

DNS クエリを実行し、詳細な DNS レコード情報を返します。
 このアイテムは、より多くのレコードタイプとカスタマイズ可能なフラグをサポートする、`net.dns.record` Zabbix エージェントアイテムの拡張バージョンです。
 戻り値: *JSON* オブジェクト

パラメータ:

- **ip** - DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空欄）;
- **name** - クエリを実行する DNS 名;
- **type** - クエリを実行するレコードタイプ（デフォルトは *SOA*）;
- **timeout** - リクエストのタイムアウト（秒、デフォルトは 1 秒）;
- **count** - リクエストの試行回数（デフォルトは 2）;
- **protocol** - DNS クエリを実行するプロトコル: *udp*（デフォルト）または *tcp*;
- **flags** - クエリに渡す 1 つ以上のカンマ区切りの引数。

コメント:

- `type` の可能な値は、*A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*, *TALINK*, *CDS*, *CDNSKEY*, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY* です。値は大文字のみでなければならないことに注意してください。小文字や混在した値はサポートされていません。
- 逆引き DNS ルックアップ（`type` が *PTR* の場合）では、DNS 名を逆順および非逆順の両方の形式で指定できます（下記の例を参照）。PTR レコードが要求される場合、DNS 名は実際には IP アドレスであることに注意してください。
- `flags` の可能な値は、*cdflag* または *nocdflag*（デフォルト）、*rdflag*（デフォルト）または *nordflag*、*dnssec* または *nodnssec*（デフォルト）、*nsid* または *nonsid*（デフォルト）、*edns0*（デフォルト）または *noedns0*、*aaflag* または *noaaflag*（デフォルト）、*adflag* または *noadflag*（デフォルト）です。`flags` の *dnssec* および *nsid* は *noedns0* と同時に使用できません。両方とも *edns0* を必要とします。値は小文字のみでなければならないことに注意してください。大文字や混在した値はサポートされていません。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。
- 出力は、指定されたパラメータに基づく DNS レコード情報を含むオブジェクトです（詳細はこちらを参照）。

例:

```
net.dns.get[192.0.2.0,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]
```

```
net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 GPU デバイスの数。
 戻り値: 整数。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのデコーダー使用率 (%)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの ECC モード情報 (現在、保留中)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのエンコーダ統計情報。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスエンコーダの使用率 (%)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 ドライバが最後にリロードされてからの、ミリジュール (mJ) 単位での GPU デバイスの総エネルギー消費量。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの README を参照してください。

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

 GPU デバイスの ECC メモリエラー情報 (修正済み、未修正)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの ECC レジスタエラー情報（修正済み、未修正）。
 戻り値 : *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの最大速度に対する平均ファン速度の割合。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.get

 UUID と名前で検出された GPU。
 戻り値 : *JSON*。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのグラフィックスクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのフレームバッファメモリ統計情報 (total, reserved, free, used)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの BAR1 メモリ統計情報 (total, free, used)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのメモリクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの PCI 使用率（送信/受信スループット、KBps 単位）。
 戻り値 : *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのパフォーマンス状態。
 戻り値：整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 ミリワット単位の GPU デバイスの電力制限。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの現在の消費電力 (ミリワット単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.serial[<deviceUUID>]

 GPU デバイスのシリアル番号。
 戻り値：文字列。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのストリーミングマルチプロセッサのクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.temperature[<deviceUUID>]

 GPU デバイスの温度 (摂氏)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.utilization[<deviceUUID>]

 GPU デバイスの使用率統計（GPU/メモリの使用率、パーセンテージ）。
 戻り値：*JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのビデオクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.system.driver.version

 NVIDIA ドライバーのバージョン。
 戻り値: 文字列。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

nvml.version

 NVML ライブラリのバージョン。
 戻り値: 文字列。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 ASM（Automatic Storage Management）ディスクグループの統計情報を返します。
 戻り値：*JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります。

    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例：1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合。
    - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。
- **diskgroup** - 問い合わせる ASM ディスクグループの名前。

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 ASM ディスクグループのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
    - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件がいずれも該当しない場合;
    - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 アーカイブログの統計情報。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます。
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例:1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
    - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件がいずれも該当しない場合;
    - TNS 値 - 先頭が開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **destination** - 問い合わせる宛先の名前。

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

 コンテナデータベース（CDB）の情報。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります：
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例：1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合。
    - TNS 値 - 先頭が開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。

- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。
- **destination** - クエリを実行するデータベース名。

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 カスタムクエリの結果。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ (例: ”tcp://”)、ポート (例: 1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されていて、上記の条件がいずれも該当しない場合;
  - TNS 値 - 先頭が開き括弧 “(“で始まる場合 (先頭のスペースは無視)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加可能 (ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **queryName** - カスタムクエリの名前。SQL ファイル名 (拡張子なし) と一致する必要があります;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

コメント: - 返されるデータは自動的に JSON に変換されます。- クエリから直接 JSON を返すことは避けてください。プラグインが再度 JSON に変換しようとすると破損します。

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 データファイルの統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されていて、上記の条件がいずれも当てはまらない場合;
  - TNS 値 - 先頭が開き括弧 “(“（先頭のスペースは無視）で始まる場合。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **diskgroup** - 問い合わせる ASM ディスクグループの名前。

oracle.db.discovery[connString,<user>,<password>,<service>]

 データベースのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字

小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.fra.stats[connString,<user>,<password>,<service>]

 Fast Recovery Area (FRA) の統計情報を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります。
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例 : "tcp://")、ポート（例:1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されていて、上記のいずれにも該当しない場合。
  - TNS 値 - 開き括弧 "(" で始まる場合（先頭のスペースは無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

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

 インスタンスの統計情報。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます。
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例 : "tcp://")、ポート（例:1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件がいずれも該当しない場合;
  - TNS 値 - 先頭が開き括弧 "("で始まる場合（先頭のスペースは無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 Pluggable Databases (PDBs) の情報。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://")、ポート（例: 1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 "(" で始まる場合（先頭のスペースは無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **destination** - クエリを実行するデータベース名。

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 PDB のリストを返します。ローレベルディスカバリで使用されます。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります。
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例：1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルにそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されていて、上記の条件がいずれも該当しない場合;
  - TNS 値 - 先頭が開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.pga.stats[connString,<user>,<password>,<service>]

 プログラム・グローバル・エリア（PGA）の統計情報を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります。
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例：1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルでその名前が見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されていて、上記の条件がいずれも該当しない場合。
  - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

oracle.ping[connString,<user>,<password>,<service>]

 Oracle への接続が確立できるかどうかをテストします。
 戻り値: *1* - 接続に成功した場合; *0* - 接続が失敗した場合（認証や設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 先頭が開き括弧 “(“（先頭のスペースは無視）で始まる場合。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

例:

```
oracle.ping[tcp://127.0.0.1:1521,ZABBIX_MON,zabbix,xe]
oracle.ping[localhost,ZABBIX_MON,zabbix,xe]
oracle.ping[zbx_tns_example,ZABBIX_MON,zabbix,xe]
oracle.ping["(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=x
```

oracle.proc.stats[connString,<user>,<password>,<service>]

 プロセスの統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:

    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、上記のいずれの条件にも該当しない場合;
    - TNS 値 - 先頭が開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 制御ファイルからのログファイル情報。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ (例: ”tcp://”)、ポート (例: 1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
    - TNS 値 - 先頭に開き括弧 “(“(先頭のスペースは無視) がある場合。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます (ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

oracle.sga.stats[connString,<user>,<password>,<service>]

 システムグローバル領域（SGA）の統計情報を返します。
 戻り値 : *JSON* オブジェクト

パラメータ:

- **connString** - 以下のいずれかになります。
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例 : ”tcp://”）、ポート（例:1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件がいずれも該当しない場合。
    - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 セッション統計を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります。
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例 : ”tcp://”）、ポート（例 : 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
    - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合。
    - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。

- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加することができます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **lockMaxTime** - セッションを長時間ロックされたと見なす最大ロック時間（秒）。デフォルト：600 秒。

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 システムメトリック値のセットを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ (例: ”tcp://”)、ポート (例: 1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合 (先頭のスペースは無視)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます (ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **duration** - システムメトリック値の取得間隔 (秒)。可能な値: *60* — 長い間隔 (デフォルト)、*15* — 短い間隔。

oracle.sys.params[connString,<user>,<password>,<service>]

 システムパラメータ値のセットを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれか:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 表領域の統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかになります:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **tablespace** - 問い合わせる表領域の名前。デフォルト（空欄で `type` が設定されている場合）:
  - ”TEMP”（`type` が”TEMPORARY” の場合）;
  - ”USERS”（`type` が”PERMANENT” の場合）。

- **type** - 問い合わせる表領域のタイプ。デフォルト（tablespace が設定されている場合）：”PERMANENT”。
- **conname** - 情報が必要なコンテナの名前。

tablespace、type、conname のいずれかが省略された場合、アイテムは一致するすべてのコンテナ（PDB および CDB を含む）の表領域統計情報を返します。

oracle.ts.discovery[connString,<user>,<password>,<service>]

 表領域のリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ (例: ”tcp://”)、ポート (例: 1521)、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます;
  - セッション名 - プラグイン設定ファイルでその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合 (先頭のスペースは無視)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、as sysrac のいずれかを user as sysdba の形式で追加できます (ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません)。
- **password** - Oracle のパスワード;
- **service** - Oracle サービス名。

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

 Oracle ユーザー情報を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます：
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例：”tcp://”）、ポート（例：1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます；
  - セッション名 - プラグイン設定ファイルでそのような名前が見つかった場合；
  - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件がいずれも該当しない場合；
  - TNS 値 - 先頭が開き括弧 “(“で始まる場合（先頭のスペースは無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、as sysrac のいずれかを user as sysdba の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード ；
- **service** - Oracle サービス名；
- **username** - 情報が必要なユーザー名。小文字のユーザー名はサポートされていません。デフォルト：現在のユーザー。

oracle.version[connString,<user>,<password>,<service>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - 以下のいずれか:
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、URI として扱われます。
  - セッション名 - プラグイン設定ファイルにその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定され、上記の条件がいずれも該当しない場合;
  - TNS 値 - 開き括弧 “(“で始まる場合 (先頭のスペースは無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、as sysrac のいずれかを user as sysdba の形式で追加することができます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 autovacuum ワーカーの数。
 戻り値：整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 アーカイブされたファイルに関する情報。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 チェックポイントの種類ごとに分類された、データベースクラスタのチェックポイントの合計数。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 PostgreSQL バッファキャッシュのヒット率。
 戻り値：*Float*。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 タイプ別の接続数を返します。
 戻り値：*JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

pgsql.custom.query[uri,<username>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **queryName** - カスタムクエリの名前。SQL ファイル名 (拡張子なし) と一致する必要があります;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

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

 データベースの最も古い FrozenXID の年齢。
 戻り値: 整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 データベースごとの膨張テーブルの数。
 戻り値: 整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 PostgreSQL データベースのリスト。ローレベルディスカバリに使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 データベースのサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 データベースごとの統計情報を収集します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 クラスタ内のすべてのデータベースの集計データ。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 データベースごとに付与されたロックに関する情報です。ローレベルディスカバリに使用されます。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 最も古い XID の経過時間。
 戻り値 : 整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 実行時間によるクエリのメトリクス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名;
- **timePeriod** - 遅いクエリのカウントのための実行時間の閾値 (正の整数でなければなりません)。

pgsql.replication.count[uri,<username>,<password>]

 スタンバイサーバーの数。
 戻り値：整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process[uri,<username>,<password>]

 各送信プロセスごとのフラッシュラグ、書き込みラグ、リプレイラグ。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process.discovery[uri,<username>,<password>]

 レプリケーションプロセス名のディスカバリ。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.recovery_role[uri,<username>,<password>]

 リカバリーステータス。
 戻り値: *0* - マスターモード; *1* - リカバリーがまだ進行中 (スタンバイモード)。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.status[uri,<username>,<password>]

 レプリケーションのステータス。
 戻り値: *0* - ストリーミングがダウン; *1* - ストリーミングがアップ; *2* - マスターモード。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication_lag.b[uri,<username>,<password>]

 レプリケーションラグ（バイト単位）
 戻り値：整数

パラメータ:

- **uri** - URI またはセッション名
- **username, password** - PostgreSQL の認証情報

pgsql.replication_lag.sec[uri,<username>,<password>]

 レプリケーションラグ（秒単位）
 戻り値：整数

パラメータ:

- **uri** - URI またはセッション名
- **username, password** - PostgreSQL の認証情報

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

 PostgreSQL の稼働時間 (ミリ秒単位)。
 戻り値: *Float*。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 PostgreSQL のバージョンを返します。
 戻り値: 文字列。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

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

 WAL の統計情報。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名。

redis.config[connString,<password>,<pattern>]

 パターンに一致する Redis インスタンスの設定パラメータを取得します。
 戻り値: *JSON* - グロブスタイルのパターンが使用された場合; パターンにワイルドカード文字が含まれていない場合は単一値。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード;
- **pattern** - グロブスタイルのパターン (デフォルトは **)。

redis.info[connString,<password>,<section>]

 INFO コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード;
- **section** - 情報のセクション（デフォルトは *default*）。

redis.ping[connString,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード。

redis.slowlog.count[connString,<password>]

 Redis が起動してからのスローログエントリの数。
 戻り値: 整数。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード。

smart.attribute.discovery

 S.M.A.R.T. デバイス属性のリストを返します。
 戻り値: *JSON* オブジェクト。

コメント:

- 次のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- HDD、SSD、NVME ドライブタイプがサポートされています。ドライブは単独または RAID で組み合わせることができます。RAID の場合、{#NAME} にはアドオンが付きます。例: {”{#NAME}”: ”/dev/sda cciss,2”}。

smart.disk.discovery[<type>]

 S.M.A.R.T. デバイスのリストを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **type** - ディスクをスキャンする値を指定します。指定可能な値: *id*, *name* (デフォルト)。*id* は Windows ではサポートされておらず、選択した場合はエラーを返します

コメント:

- 以下のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAID-TYPE};
- HDD、SSD、NVME ドライブタイプがサポートされています。ドライブが RAID に属していない場合、{#RAIDTYPE} は空になります。RAID の場合、{#NAME} にはアドオンが付きます。例: {"{#NAME}": "/dev/sda cciss,2"}。

smart.disk.get[<path>,<raid type>]

 S.M.A.R.T. デバイスの利用可能なすべてのプロパティを返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **path** - ディスクパス。値として {#PATH} マクロを使用できます。
- **raid_type** - RAID タイプ。値として {#RAID} マクロを使用できます。

コメント:

- HDD、SSD、NVME ドライブタイプがサポートされています。ドライブは単独または RAID で組み合わせることができます。
- データには smartctl のバージョンと呼び出し引数、および追加フィールドが含まれます：
disk_name - RAID ディスカバリ用の必要なアドオンを含む名前を保持します。例: {"disk_name": "/dev/sda cciss,2"}
disk_type - ディスクタイプ（HDD、SSD、NVME）を保持します。例: {"disk_type": "ssd"};
- パラメータが指定されていない場合、アイテムはすべてのディスクに関する情報を返します。

systemd.unit.get[unit name,<interface>]

 systemd ユニットのすべてのプロパティを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **unit name** - ユニット名（アイテムプロトタイプで {#UNIT.NAME} マクロを使用して名前を検出することもできます）;
- **interface** - ユニットインターフェースのタイプ。可能な値: *Unit*（デフォルト）、*Service*、*Socket*、*Device*、*Mount*、*Automount*、*Swap*、*Target*、*Path*。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされています;
- Unit インターフェースの LoadState、ActiveState、UnitFileState はテキストと整数の両方で返されます: `"ActiveState":{"state":1,"`

systemd.unit.info[unit name,<property>,<interface>]

 systemd ユニット情報。
 戻り値：文字列。

パラメータ:

- **unit name** - ユニット名（アイテムプロトタイプで {#UNIT.NAME} マクロを使用して名前を検出することができます）;
- **property** - ユニットプロパティ（例：ActiveState（デフォルト）、LoadState、Description）;
- **interface** - ユニットインターフェースタイプ（例：Unit（デフォルト）、Socket、Service）。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされています;
- このアイテムは、dbus APIで説明されているように、特定のインターフェースタイプから特定のプロパティを取得できます。

例:

```
systemd.unit.info["{#UNIT.NAME}"] # 検出された systemd ユニットのアクティブ状態（active、reloading、inactive、fai
systemd.unit.info["{#UNIT.NAME}",LoadState] # 検出された systemd ユニットのロード状態情報を収集
systemd.unit.info[mysqld.service,Id] # サービスの技術名（mysqld.service）を取得
systemd.unit.info[mysqld.service,Description] # サービスの説明（MySQL Server）を取得
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] # サービスが最後にアクティブ状態になった時刻（1562565036283
systemd.unit.info[dbus.socket,NConnections,Socket] # このソケットユニットからの接続数を収集
```

systemd.unit.discovery[<type>]

 systemd ユニットとその詳細のリスト。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **type** - 可能な値: *all*, *automount*, *device*, *mount*, *path*, *service* (デフォルト), *socket*, *swap*, *target*。

このアイテムは Linux プラットフォームでのみサポートされています。

web.certificate.get[hostname,<port>,<address>]

 証明書を検証し、証明書の詳細を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **hostname** - IP または DNS のいずれかを指定できます。
URL スキーム（*https* のみ）、パス（無視されます）、ポートを含めることができます。
 最初と 2 番目のパラメータの両方でポートが指定されている場合、それらの値は一致している必要があります。
3 番目のパラメータ（address）が指定されている場合、hostname は SNI およびホスト名検証のみに使用されます;
- **port** - ポート番号（HTTPS のデフォルトは 443）;
- **address** - IP または DNS のいずれかを指定できます。指定した場合、接続にはこの値が使用され、hostname（1 番目のパラメータ）は SNI およびホスト検証に使用されます。1 番目のパラメータが IP で 3 番目のパラメータが DNS の場合、1 番目のパラメータが接続に使用され、3 番目のパラメータが SNI およびホスト検証に使用されます。

コメント:

- ホスト設定で指定された宛先が存在しない、利用できない、または TLS ハンドシェイクが無効な証明書以外のエラーで失敗した場合、このアイテムはサポート外になります;
- 現在、AIA（Authority Information Access）X.509 拡張、CRL および OCSP（OCSP スタップリングを含む）、Certificate Transparency はサポートされていません;
- JSON レスポンスのフィールド:
  - *x509*: X.509 証明書の詳細を含みます。
    - *version*: X.509 バージョン（例: 3）。
    - *serial_number*: 証明書のシリアル番号。
    - *signature_algorithm*: 証明書の署名に使用されたアルゴリズム（例: SHA256-RSA）。
    - *issuer*: 証明書の発行者。
    - *not_before*: 証明書の有効開始日。
    - *not_after*: 証明書の有効期限。
    - *subject*: 証明書のサブジェクト。
    - *public_key_algorithm*: 公開鍵に使用されたアルゴリズム（例: RSA）。
    - *alternative_names*: サブジェクトの代替名（存在する場合）、存在しない場合は `null`。
  - result: 検証結果を含みます。
    - *value*: 検証ステータス（下記の値を参照）。
    - *message*: 詳細な検証メッセージ（例: `"certificate verified successfully"`）。
  - *sha1_fingerprint*: 証明書の SHA-1 フィンガープリント。
  - *sha256_fingerprint*: 証明書の SHA-256 フィンガープリント。
- `$.result.value` フィールドは証明書の検証結果を示します。可能な値は以下の通りです:
  - *valid* - 証明書は有効で信頼されています。
  - *valid-but-self-signed* - 証明書は有効ですが自己署名されています（サブジェクトと発行者が一致）。
  - *invalid* - 証明書が期限切れ、不正なホスト名、未知の署名機関などの理由で無効です。

例:

```
web.certificate.get[example.com,443]
```

JSON レスポンス:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
```

```
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


概要

Windows Zabbix エージェントのアイテムは、2 つのリストで表示されます。

- 共通アイテム - UNIX Zabbix エージェントと共通のアイテムキー
- Windows 固有のアイテム - **Windows** のみでサポートされているアイテムキー

Windows 上の Zabbix エージェントでサポートされているすべてのアイテムキーは、新世代の Zabbix エージェント 2 でもサポートされていることに注意してください。エージェント 2 でのみ使用できる追加のアイテムキーも参照してください。

参考: Windows アイテムの最小権限

共有アイテム

以下の表は、Windows でサポートされており、UNIX Zabbix エージェントと共有されている Zabbix エージェントアイテムを一覧にしたものです。

- アイテムキーは UNIX Zabbix エージェントアイテムの詳細へのリンクです
- Windows に関連するアイテムコメントが含まれています

アイテムキー	説明	アイテムグループ
log	ログファイルの監視。このアイテムは Windows イベントログには対応していません。 `persistent_dir` パラメータは Windows ではサポートされていません。	ログ監視
log.count	監視対象のログファイル内で一致した行数。このアイテムは Windows イベントログには対応していません。 `persistent_dir` パラメータは Windows ではサポートされていません。	
logrt	ローテートされるログファイルの監視。このアイテムは Windows イベントログには対応していません。 `persistent_dir` パラメータは Windows ではサポートされていません。	
logrt.count	ローテートされる監視対象のログファイル内で一致した行数。このアイテムは Windows イベントログには対応していません。 `persistent_dir` パラメータは Windows ではサポートされていません。	
modbus.get	Modbus データの読み取り。	Modbus
net.dns	DNS サービスが稼働しているかどうかをチェックします。 Zabbix agent 2 を使用しない限り、Windows では `ip`、`timeout`、`count` パラメータは無視されます。	ネットワーク
net.dns.perf	DNS サービスのパフォーマンスをチェックします。 Zabbix agent 2 を使用しない限り、Windows では `ip`、`timeout`、`count` パラメータは無視されます。	
net.dns.record	DNS クエリを実行します。 Zabbix agent 2 を使用しない限り、Windows では `ip`、`timeout`、`count` パラメータは無視されます。	
net.if.discovery	ネットワークインターフェースの一覧。 一部の Windows バージョン（例：Server 2008）では、インターフェース名で非 ASCII 文字をサポートするために最新の更新プログラムが必要な場合があります。	
net.if.in	ネットワークインターフェースの受信トラフィック統計。 Windows では、利用可能な場合は 64 ビットカウンタから値を取得します。64 ビットインターフェース統計カウンタは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンタが利用できない場合、エージェントは 32 ビットカウンタを使用します。 Windows ではマルチバイトインターフェース名がサポートされています。 Windows でネットワークインターフェースの説明を取得するには、net.if.discovery または net.if.list アイテムを使用できます。	

アイテムキー	説明	アイテムグループ
net.if.out	ネットワークインターフェースの送信トラフィック統計。 Windows では、利用可能な場合は 64 ビットカウンタから値を取得します。64 ビットインターフェース統計カウンタは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンタが利用できない場合、エージェントは 32 ビットカウンタを使用します。 Windows ではマルチバイトインターフェース名がサポートされています。 Windows でネットワークインターフェースの説明を取得するには、net.if.discovery または net.if.list アイテムを使用できます。	
net.if.total	ネットワークインターフェースの受信および送信トラフィック統計の合計。 Windows では、利用可能な場合は 64 ビットカウンタから値を取得します。64 ビットインターフェース統計カウンタは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンタが利用できない場合、エージェントは 32 ビットカウンタを使用します。 Windows でネットワークインターフェースの説明を取得するには、net.if.discovery または net.if.list アイテムを使用できます。	
net.tcp.listen	この TCP ポートが LISTEN 状態かどうかをチェックします。	
net.tcp.port	指定したポートへの TCP 接続が可能かどうかをチェックします。	
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかどうかをチェックします。 Windows での LDAP および HTTPS のチェックは Zabbix agent 2 のみサポートされています。	
net.tcp.service.perf	TCP サービスのパフォーマンスをチェックします。 Windows での LDAP および HTTPS のチェックは Zabbix agent 2 のみサポートされています。	
net.tcp.socket.count	パラメータに一致する TCP ソケットの数を返します。 このアイテムは Linux では Zabbix エージェントでサポートされていますが、 Windows では 64 ビット Windows 上のZabbix agent 2のみサポートされています。	
net.udp.service	サービスが稼働しており、UDP リクエストに応答しているかどうかをチェックします。	
net.udp.service.perf	UDP サービスのパフォーマンスをチェックします。	
net.udp.socket.count	パラメータに一致する UDP ソケットの数を返します。 このアイテムは Linux では Zabbix エージェントでサポートされていますが、 Windows では 64 ビット Windows 上のZabbix agent 2のみサポートされています。	
proc.get	OS プロセスとそのパラメータの一覧。 Windows では `cmdline` パラメータはサポートされていません。	プロセス
proc.num	プロセス数。 Windows では `name` および `user` パラメータのみサポートされています。	
system.cpu.discovery	検出された CPU/CPU コアの一覧。	システム
system.cpu.load	CPU 負荷。 Zabbix エージェントでコレクタープロセスが開始されると、次のパフォーマンスカウンタが初期化され、後でこのアイテムで使用されます：`\System\Processor Queue Length`	
system.cpu.num	CPU 数。	
system.cpu.util	CPU 使用率（パーセント）。 値は *Processor Time* パフォーマンスカウンタを使用して取得されます。Windows 8 以降のタスクマネージャーは *Processor Utility* パフォーマンスカウンタに基づいて CPU 使用率を表示しますが、以前のバージョンでは *Processor Time* カウンタでした (詳細はこちらを参照)。 Windows でサポートされている `type` パラメータは *system* のみです。	
system.hostname	システムホスト名。 Windows では、値は GetComputerName()（**netbios** の場合）、GetComputerNameExA()（**fqdn** の場合）、または gethostname()（**host** の場合）関数のいずれかで取得されます。 詳細な説明も参照してください。	
system.localtime	システム時刻。	
system.run	ホスト上で指定したコマンドを実行します。	
system.sw.arch	ソフトウェアアーキテクチャ情報。	

アイテムキー	説明	アイテムグループ
system.swap.size	スワップ領域のサイズ（バイトまたは合計に対するパーセント）。 pused タイプパラメータは Linux では Zabbix エージェントでサポートされていますが、Windows ではZabbix agent 2のみサポートされています。 このキーは仮想化（VMware ESXi、VirtualBox）Windows プラットフォーム上でスワップ領域のサイズ/パーセントを正しく報告しない場合があります。この場合、perf_counter[\700(_Total)\702] キーを使用して正しいスワップ領域のパーセントを取得できます。	
system.uname	システムの識別。 Windows では、このアイテムの値は Win32_OperatingSystem および Win32_Processor WMI クラスから取得されます。OS 名（エディションを含む）はユーザーの表示言語に翻訳される場合があります。一部の Windows バージョンでは、商標記号や余分なスペースが含まれる場合があります。	
system.uptime	システムの稼働時間（秒）。	
vfs.dir.count	ディレクトリエントリ数。 Windows では、ディレクトリシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	仮想ファイルシステム
vfs.dir.get	ディレクトリエントリリスト。 Windows では、ディレクトリシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	
vfs.dir.size	ディレクトリサイズ。 Windows では、シンボリックリンクはすべてスキップされ、ハードリンクは 1 回だけカウントされます。	
vfs.file.cksum	UNIX cksum アルゴリズムで計算されたファイルのチェックサム。	
vfs.file.contents	ファイルの内容の取得。	
vfs.file.exists	ファイルの存在をチェックします。 Windows では、zabbix_get.exe または agent2 をコマンドラインユーティリティで呼び出す場合、ダブルクォートはバックスラッシュ'\' でエスケープし、アイテムキー全体をダブルクォートで囲む必要があります。 Windows では、存在しないディレクトリ内でディレクトリを検索した場合（例：vfs.file.exists[C:\no\dir,dir]（'no' が存在しない場合））、アイテムが未サポートになる場合があります。	
vfs.file.get	ファイルに関する情報を返します。 Windows でサポートされているファイルタイプ：通常ファイル、ディレクトリ、シンボリックリンク	
vfs.file.md5sum	ファイルの MD5 チェックサム。	
vfs.file.owner	ファイルの所有者を取得します。	
vfs.file.regexp	ファイル内の文字列を取得します。	
vfs.file.regmatch	ファイル内の文字列を検索します。	
vfs.file.size	ファイルサイズ。	
vfs.file.time	ファイルの時刻情報。 Windows XP では vfs.file.time[file,change] は vfs.file.time[file,access] と等しい場合があります。	
vfs.fs.discovery	タイプとマウントオプション付きのマウントされたファイルシステムの一覧。 Windows では {#FSLABEL} マクロがサポートされています。	
vfs.fs.get	タイプ、利用可能なディスク容量、inode 統計、マウントオプション付きのマウントされたファイルシステムの一覧。 Windows では {#FSLABEL} マクロがサポートされています。	
vfs.fs.size	ディスク容量（バイトまたは合計に対するパーセント）。	
vm.memory.size	メモリサイズ（バイトまたは合計に対するパーセント）。	仮想メモリ
web.page.get	Web ページの内容を取得します。	Web 監視
web.page.perf	Web ページ全体の読み込み時間。	
web.page.regexp	Web ページ内の文字列を検索します。	
agent.hostmetadata	エージェントホストのメタデータ。	Zabbix
agent.hostname	エージェントホスト名。	
agent.ping	エージェントの可用性チェック。	
agent.variant	Zabbix エージェントのバリアント（Zabbix agent または Zabbix agent 2）。	
agent.version	Zabbix エージェントのバージョン。	
zabbix.stats	Zabbix サーバーまたはプロキシの内部メトリクスセットをリモートで返します。	
zabbix.stats	Zabbix サーバーまたはプロキシで遅延しているキュー内の監視アイテム数をリモートで返します。	

Windows 固有のアイテム

この表は、Windows Zabbix エージェントでのみサポートされているアイテムキーの詳細を示しています。

Windows 固有のアイテムは、同様のエージェントアイテムの概算的な対応物である場合があります。たとえば、Windows でサポートされている `proc_info` は、Windows でサポートされていない `proc.mem` アイテムに大まかに対応します。

アイテムキーは、完全なアイテムキーの詳細へのリンクです。

アイテムキー	説明	アイテムグループ
eventlog	Windows イベントログの監視。	ログ監視
eventlog.count	Windows イベントログ内の行数。	
net.if.list	ネットワークインターフェースのリスト（インターフェースタイプ、ステータス、IPv4 アドレス、説明を含む）。	ネットワーク
perf_counter	任意の Windows パフォーマンスカウンターの値。	パフォーマンスカウンター
perf_counter_en	英語での任意の Windows パフォーマンスカウンターの値。	
perf_instance.discovery	Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。	
perf_instance_en.discovery	英語のオブジェクト名を使用して検出された Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。	
proc_info	特定のプロセスに関するさまざまな情報。	プロセス
registry.data	Windows レジストリキー内の指定された値名のデータを返します。	レジストリ
registry.get	指定されたキーにある Windows レジストリの値またはキーのリスト。	
service.discovery	Windows サービスのリスト。	サービス
service.info	サービスに関する情報。	
services	サービスのリスト。	
vm.vmemory.size	バイト単位または合計に対するパーセンテージでの仮想メモリサイズ。	仮想メモリ
wmi.get	WMI クエリを実行し、最初に選択されたオブジェクトを返します。	WMI
wmi.getall	WMI クエリを実行し、全レスポンスを返します。	

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 イベントログの監視。
 戻り値：*Log*。

パラメータ:

- **name** - イベントログチャネルの名前（イベントビューア GUI のログ名）;
- **regexp** - 必要なパターンを記述する正規表現（大文字小文字を区別）;
- **severity** - 重大度を記述する正規表現（大文字小文字を区別しない）。このパラメータは、次の値に基づく正規表現を受け入れます：”Information”、”Warning”、”Error”、”Critical”、”Verbose”（Windows Vista 以降で実行時）。
- **source** - ソース識別子を記述する正規表現（大文字小文字を区別しない）;
- **eventid** - イベント識別子を記述する正規表現（大文字小文字を区別）;
- **maxlines** - エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新しい行の最大数。このパラメータは、zabbix_agentd.confの’MaxLinesPerSecond’ の値を上書きします。
- **mode** - 可能な値：*all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新しく作成されたアイテムにのみ影響します）。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります;
- エージェントは”Forwarded events” ログからイベントを送信できません;
- Windows Eventing 6.0 がサポートされています;
- このアイテムに対して情報のタイプで Log 以外を選択すると、ローカルタイムスタンプ、ログの重大度、ソース情報が失われます;
- ログ監視に関する追加情報も参照してください。

例:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
```

```
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] # ここでは、カスタム正規表現`TWOSHORT`（*Result is TRUE* タイプとして定義され、式自体は
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Windows イベントログ内の行数を返します。
 戻り値：整数。

パラメータ:

- **name** - イベントログチャネルの名前（イベントビューアー GUI のログ名）;
- **regexp** - 必要なパターンを記述する正規表現（大文字小文字を区別）;
- **severity** - 重大度を記述する正規表現（大文字小文字を区別しない）。このパラメータは、次の値に基づく正規表現を受け入れます：”Information”、”Warning”、”Error”、”Critical”、”Verbose”（Windows Vista 以降で実行時）。
- **source** - ソース識別子を記述する正規表現（大文字小文字を区別しない）;
- **eventid** - イベント識別子を記述する正規表現（大文字小文字を区別）;
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.confの 10*’MaxLinesPerSecond’ です。
- **mode** - 可能な値：*all*（デフォルト）または *skip* - 古いデータの処理をスキップ（新しく作成されたアイテムにのみ影響）。

コメント:

- アイテムはアクティブチェックとして設定する必要があります。
- エージェントは”Forwarded events” ログからイベントを送信できません。
- Windows Eventing 6.0 をサポートしています。
- このアイテムに対して非ログ情報タイプを選択すると、ローカルタイムスタンプ、ログ重大度、ソース情報が失われます。
- ログ監視に関する追加情報も参照してください。

例:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 ネットワークインターフェースのリスト (インターフェースの種類、状態、IPv4 アドレス、説明を含む)。
 戻り値: テキスト。

コメント:

- マルチバイトのインターフェース名をサポート;
- 無効なインターフェースはリストされません;
- 一部のコンポーネントの有効化/無効化により、Windows のインターフェース名の順序が変更される場合があります;
- 一部の Windows バージョン (例: Server 2008) では、インターフェース名で非 ASCII 文字をサポートするために最新の更新プログラムが必要な場合があります。

perf_counter[counter,<interval>]

 任意の Windows パフォーマンスカウンターの値。
 戻り値: 整数、浮動小数点、文字列またはテキスト (リクエストによる)。

パラメータ:

- **counter** - カウンターへのパス;
- **interval** - 平均値を保存するための直近 N 秒。`interval` は 1 から 900 秒 (含む) の間で指定し、デフォルト値は 1 です。

コメント:

- `interval` は複数のサンプルが必要なカウンター (CPU 使用率など) に使用され、チェックごとに直近”interval” 秒の平均値を返します;
- 利用可能なカウンターのリストを取得するには Performance Monitor を使用できます。
- 参考: Windows パフォーマンスカウンター。

perf_counter_en[counter,<interval>]

 英語での任意の Windows パフォーマンスカウンターの値。
 戻り値：整数、浮動小数点、文字列またはテキスト（リクエストによる）。

パラメータ:

- **counter** - 英語でのカウンターへのパス;
- **interval** - 平均値を保存するための直近 N 秒間。`interval` は 1 から 900 秒（含む）の間でなければならず、デフォルト値は 1 です。

コメント:

- interval は複数のサンプルが必要なカウンター（CPU 使用率など）に使用されるため、チェックは毎回直近の”interval” 秒間の平均値を返します;
- このアイテムは **Windows Server 2008/Vista** 以降でのみサポートされています;
- 英語の文字列のリストは､次のレジストリキーを表示することで確認できます：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009。

perf_instance.discovery[object]

 Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **object** - オブジェクト名（ローカライズ済み）。

perf_instance_en.discovery[object]

 英語のオブジェクト名を使用して検出された Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。ローレベルディスカバリで使用されます。
 戻り値：*JSON* オブジェクト。

パラメータ:

- **object** - オブジェクト名（英語）。

proc_info[process,<attribute>,<type>]

 特定のプロセスに関するさまざまな情報。
 戻り値: *Float*。

パラメータ:

- **process** - プロセス名;
- **attribute** - 取得するプロセス属性;
- **type** - 表示タイプ (同じ名前のプロセスが複数存在する場合に有効)

コメント:

- サポートされている attribute は以下の通り:
vmsize (デフォルト) - プロセスの仮想メモリサイズ (Kbyte 単位)
wkset - プロセスのワーキングセットサイズ (プロセスが使用している物理メモリ量、Kbyte 単位)
pf - ページフォールト数
ktime - プロセスのカーネル時間 (ミリ秒単位)
utime - プロセスのユーザー時間 (ミリ秒単位)
io_read_b - プロセスが I/O 操作中に読み取ったバイト数
io_read_op - プロセスが実行した読み取り操作数
io_write_b - プロセスが I/O 操作中に書き込んだバイト数
io_write_op - プロセスが実行した書き込み操作数
io_other_b - プロセスが読み書き以外の操作で転送したバイト数
io_other_op - プロセスが読み書き以外で実行した I/O 操作数
gdiobj - プロセスが使用している GDI オブジェクト数
userobj - プロセスが使用している USER オブジェクト数;
- 有効な type は以下の通り:
avg (デフォルト) - <process> という名前のすべてのプロセスの平均値
min - <process> という名前のすべてのプロセスの最小値
max - <process> という名前のすべてのプロセスの最大値
sum - <process> という名前のすべてのプロセスの値の合計;
- 64 ビットシステムでは、このアイテムが正しく動作するために 64 ビットの Zabbix エージェントが必要です。

例:

```
proc_info[iexplore.exe,wkset,sum] # すべての Internet Explorer プロセスが消費している物理メモリ量を取得
proc_info[iexplore.exe,pf,avg] #Internet Explorer プロセスの平均ページフォールト数を取得
```

registry.data[key,<value name>]

 Windows レジストリキー内の指定された値名のデータを返します。
 戻り値: 整数、文字列またはテキスト（値の型による）

パラメータ:

- **key** - ルートキーを含むレジストリキー; ルートの省略形（例: HKLM）が使用可能;
- **value name** - キー内のレジストリ値名（デフォルトは空文字列””）。値名が指定されていない場合は、デフォルト値が返されます。

コメント:

- サポートされているルートの省略形:
HKCR - HKEY_CLASSES_ROOT
HKCC - HKEY_CURRENT_CONFIG
HKCU - HKEY_CURRENT_USER
HKCULS - HKEY_CURRENT_USER_LOCAL_SETTINGS
HKLM - HKEY_LOCAL_MACHINE
HKPD - HKEY_PERFORMANCE_DATA
HKPN - HKEY_PERFORMANCE_NLSTEXT
HKPT - HKEY_PERFORMANCE_TEXT
HKU - HKEY_USERS
- スペースを含むキーは二重引用符で囲む必要があります。

例:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] # このキーのデフォルト
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] # このキーの"Enable Zip
```

registry.get[key,<mode>,<name regexp>]

 指定したキーにある Windows レジストリの値またはキーのリスト。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **key** - ルートキーを含むレジストリキー; ルートの省略形（例: HKLM）が使用可能（省略形の全リストは registry.data[] のコメントを参照）;
- **mode** - 許可される値:
values（デフォルト）または *keys*;
- **name regexp** - 名前が正規表現に一致する値のみをディスカバリ（デフォルト - すべての値をディスカバリ）。mode が *values* の場合のみ許可。

スペースを含むキーはダブルクォートで囲む必要があります。

例:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON にはキー、最後のサブキー、値の名前、値の型、値のデータの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] # このキー内のすべての値のデータを
JSON にはキー、最後のサブキー、値の名前、値の型、値のデータの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] # このキーのすべてのサブキーを返しま
JSON にはキーと最後のサブキーの詳細が含まれます。
```

service.discovery

 Windows サービスのリスト。
 ローレベルディスカバリで使用します。
 戻り値: *JSON* オブジェクト。

service.info[service,<param>]

 サービスに関する情報。
 戻り値: *Integer* - `param` が *state*、*startup* の場合; *String* - `param` が *displayname*、*path*、*user* の場合; *Text* - `param` が *description* の場合
state の場合の詳細: 0 - 実行中、1 - 一時停止中、2 - 開始保留中、3 - 一時停止保留中、4 - 継続保留中、5 - 停止保留中、6 - 停止、7 - 不明、255 - そのようなサービスは存在しない
startup の場合の詳細: 0 - 自動、1 - 自動 (遅延開始)、2 - 手動、3 - 無効、4 - 不明、5 - 自動トリガースタート、6 - 自動遅延トリガースタート、7 - 手動トリガースタート

パラメータ:

- **service** - MMC サービススナップインで表示される実際のサービス名または表示名
- **param** - *state* (デフォルト)、*displayname*、*path*、*user*、*startup*、または *description*

コメント:

- `service.info[service,state]` や `service.info[service]` のようなアイテムは同じ情報を返します
- `param` が *state* の場合のみ、存在しないサービスに対して値 (255) を返します

例:

```
service.info[SNMPTRAP] - SNMPTRAP サービスの状態
service.info[SNMP Trap] - 表示名を指定した同じサービスの状態
service.info[EventLog,startup] - EventLog サービスの起動タイプ
```

services[<type>,<state>,<exclude>]

 サービスの一覧。
 戻り値: *0* - 空の場合; *Text* - サービスのリスト (改行で区切られる)。

パラメータ:

- **type** - *all* (デフォルト), *automatic*, *manual*, または *disabled*;
- **state** - *all* (デフォルト), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending*, または *paused*;
- **exclude** - 結果から除外するサービス。除外するサービスはダブルクォーテーションで囲み、カンマで区切り、スペースは入れません。

例:

```
services[,started] # 起動しているサービスのリストを返します;
services[automatic, stopped] # 起動しているべきなのに停止しているサービスのリストを返します;
services[automatic, stopped, "service1,service2,service3"] # 起動しているべきなのに停止しているサービスのリストを返
```

vm.vmemory.size[<type>]

 バイト単位または合計からのパーセンテージでの仮想メモリサイズ。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **type** - 指定可能な値: *available* (利用可能な仮想メモリ), *pavailable* (利用可能な仮想メモリ、パーセンテージ), *pused* (使用中の仮想メモリ、パーセンテージ), *total* (合計仮想メモリ、デフォルト), *used* (使用中の仮想メモリ)

コメント:

- 仮想メモリ統計の監視は以下に基づきます:
  - Windows 上の合計仮想メモリ (物理メモリ合計 + ページファイルサイズ);
  - Zabbix エージェントがコミットできる最大メモリ量;
  - システムまたは Zabbix エージェントの現在のコミット済みメモリ制限 (小さい方)。

例:

```
vm.vmemory.size[pavailable] # 利用可能な仮想メモリをパーセンテージで返す
```

wmi.get[<namespace>,<query>]

 WMI クエリを実行し、最初に選択されたオブジェクトを返します。
 戻り値: 整数、浮動小数点数、文字列またはテキスト (リクエストによる)。

パラメータ:

- **namespace** - WMI 名前空間;
- **query** - 単一オブジェクトを返す WMI クエリ。

WMI クエリはWQLで実行されます。

例:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] # 最初の物理ディスク
```

wmi.getall[<namespace>,<query>]

 WMI クエリを実行し、応答全体を返します。ローレベルディスカバリに使用できます。
 戻り値: *JSON* オブジェクト

パラメータ:

- **namespace** - WMI 名前空間;
- **query** - WMI クエリ。

コメント:

- WMI クエリはWQLで実行されます。
- 返された JSON 内のより特定の値を指すために、JSONPath の事前処理を使用できます。

例:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] # 物理ディスクのステー
```

Windows サービスの監視

このチュートリアルでは、Windows サービスの監視を設定する手順を説明します。Zabbix サーバーとエージェントが設定され、稼働していることを前提としています。

ステップ 1

サービス名を取得します。

MMC サービススナップインに移動し、サービスのプロパティを表示することで、サービス名を取得できます。全般タブに「サービス名」というフィールドが表示されます。その後に続く値が、監視用のアイテムを設定する際に使用する名前です。たとえば、「workstation」サービスを監視したい場合、サービス名は **lanmanworkstation** となります。

ステップ 2

サービスの監視のためにアイテムを設定します。

アイテム `service.info[service,<param>]` は、特定のサービスに関する情報を取得します。必要な情報に応じて、`param` オプションを指定します。`param` には *displayname*、*state*、*path*、*user*、*startup*、*description* のいずれかの値を指定できます。`param` を指定しない場合 (`service.info[service]`)、デフォルト値は *state* です。

返される値の型は、選択した `param` によって異なります。*state* および *startup* の場合は整数、*displayname*、*path*、*user* の場合は文字列、*description* の場合はテキストです。

例:

- キー: `service.info[lanmanworkstation]`
- 情報の型: 数値（整数）

アイテム `service.info[lanmanworkstation]` は、サービスの状態を数値として取得します。フロントエンドで数値をテキスト表現にマッピングするには（例: "0" を"Running"、"1" を"Paused" など）、アイテムが設定されているホストで値のマッピングを設定できます。これを行うには、*Windows services by Zabbix agent* または *Windows services by Zabbix agent active* テンプレートをホストにリンクするか、または該当テンプレートで設定されている *Windows service state* の値のマッピングを基に、ホストで新しい値のマッピングを設定します。

なお、上記の両テンプレートには、サービスを自動的に検出するディスカバリルールが設定されています。これを望まない場合は、テンプレートをホストにリンクした後、ホストレベルでディスカバリルールを無効化できます。

Windows サービスのディスカバリ

ローレベルディスカバリは、コンピュータ上のさまざまなエンティティのアイテム、トリガー、グラフを自動的に作成する方法を提供します。Zabbix は、サービスの正確な名前を知る必要や、各サービスごとに手動でアイテムを作成することなく、マシン上の Windows サービスの監視を自動的に開始できます。フィルタを使用して、関心のあるサービスのみの実際のアイテム、トリガー、グラフを生成することができます。

2 SNMP エージェント

概要

プリンター、ネットワークスイッチ、ルーター、UPS など、通常 SNMP 対応であり、完全なオペレーティングシステムや Zabbix エージェントのセットアップが非現実的なデバイスで SNMP 監視を使用したい場合があります。

これらのデバイス上の SNMP エージェントが提供するデータを取得できるようにするには、Zabbix サーバーを SNMP サポート付きで初期設定し、`--with-net-snmp` フラグを指定する必要があります。アイテム値が正しい形式で表示されるように、MIB ファイルのインストールも推奨されます。MIB ファイルがない場合、値が UTF-8 の代わりに HEX で表示される、またはその逆など、フォーマットの問題が発生する可能性があります。

SNMP チェックは UDP プロトコルのみで実行されます。

Zabbix サーバーおよびプロキシデーモンは、不正な SNMP 応答を受信した場合、次のようなログ行を記録します。

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

これらはすべての問題ケースを網羅しているわけではありませんが、複合リクエストを無効にすべき個々の SNMP デバイスを特定するのに役立ちます。

Zabbix サーバー/プロキシは、SNMP の `walk` および `get` アイテムに対して最大 5 回まで再試行します。再試行メカニズムは DNS 解決の失敗には適用されません。

レガシー SNMP チェック（単一の OID 番号または文字列）の場合、Zabbix サーバー/プロキシは、クエリが失敗した場合に少なくとも 1 回は再試行します。これは、SNMP ライブラリの再試行メカニズム、または内部の複合処理メカニズムによって行われます。

Warning:
SNMPv3 デバイスを監視する場合、msgAuthoritativeEngineID（snmpEngineID または「Engine ID」とも呼ばれる）が 2 つのデバイスで共有されないようにしてください。RFC 2571（セクション 3.1.1.1）によると、これは各デバイスごとに一意でなければなりません。

Warning:
RFC3414 は、SNMPv3 デバイスが engineBoots を永続化することを要求しています。一部のデバイスはこれを行わず、再起動後に SNMP メッセージが古いものとして破棄されます。このような場合、サーバー/プロキシ上で SNMP キャッシュを手動でクリアする（-R snmp_cache_reloadを使用）か、サーバー/プロキシを再起動する必要があります。

Note:
Zabbix は SNMPv3 の EngineID→IP マッピングをキャッシュし、毎回プローブを送信する代わりにキャッシュされた EngineID を後続のチェックで再利用することで、ネットワークトラフィックを削減します。EngineID が再利用できない場合は、新しい EngineID を検出するためにプローブによる再試行が行われます。

SNMP 監視の設定

SNMP を介してデバイスの監視を開始するには、次の手順を実行する必要があります。

ステップ 1

監視したいアイテムの SNMP 文字列（または OID）を調べます。

SNMP 文字列の一覧を取得するには、**snmpwalk** コマンド（net-snmpソフトウェアの一部で、Zabbix インストール時にインストールされているはずです）または同等のツールを使用します。

```
snmpwalk -v 2c -c public <host IP> .
```

ここで'2c' は SNMP バージョンを表しているため、デバイスの SNMP バージョン 1 を指定する場合は'1' に置き換えることもできます。

これにより、SNMP 文字列とその最新値のリストが表示されます。表示されない場合は、SNMP の「コミュニティ」が標準の「public」と異なる可能性があるため、何であるかを調べる必要があります。

その後、リストを確認して監視したい文字列を見つけます。例えば、スイッチのポート 3 に入ってくるバイト数を監視したい場合は、次の行の IF-MIB::ifHCInOctets.3 文字列を使用します。

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

次に、**snmpget** コマンドを使用して「IF-MIB::ifHCInOctets.3」の数値 OID を調べます。

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

文字列の最後の数字が監視対象のポート番号であることに注意してください。参考: 動的インデックス。

これにより、次のような出力が得られます。

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

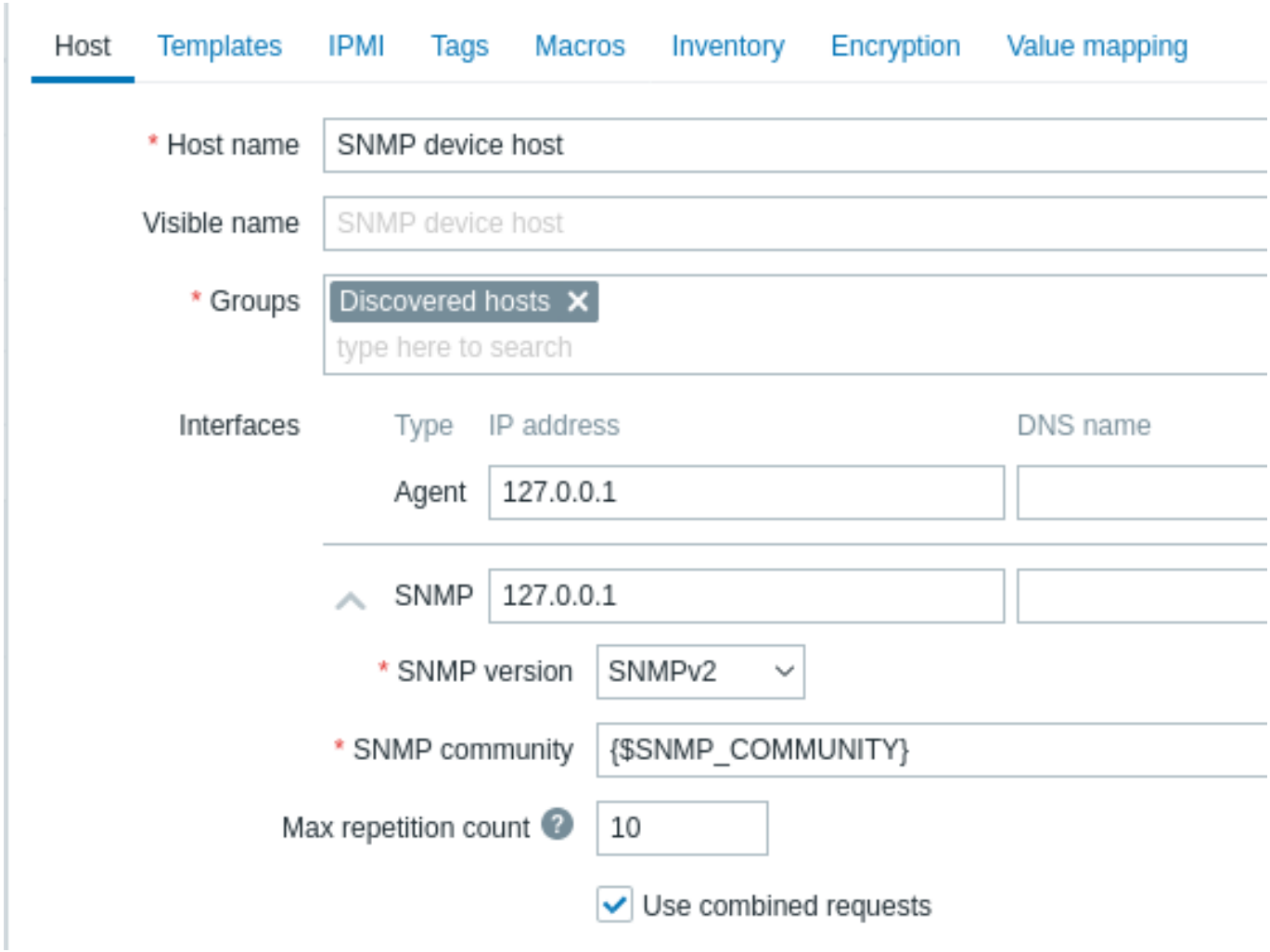
ここでも、OID の最後の数字がポート番号です。

> **Note:**
> 最もよく使われる SNMP OID の一部は、Zabbix によって自動的に数値表現に変換されます。

上記の最後の例では、値の型は「Counter64」であり、内部的には ASN_COUNTER64 型に対応します。サポートされている型の全リストは、ASN_COUNTER、ASN_COUNTER64、ASN_UINTEGER、ASN_UNSIGNED64、ASN_INTEGER、ASN_INTEGER64、ASN_FLOAT、ASN_DOUBLE、ASN_TIMETICKS、ASN_GAUGE、ASN_IPADDRESS、ASN_OCTET_STR、ASN_OBJECT_ID です。これらの型は、**snmpget** の出力では「Counter32」、「Counter64」、「UInteger32」、「INTEGER」、「Float」、「Double」、「Timeticks」、「Gauge32」、「IpAddress」、「OCTET STRING」、「OBJECT IDENTIFIER」に大まかに対応しますが、表示ヒントの有無によっては「STRING」、「Hex-STRING」、「OID」などとして表示される場合もあります。

ステップ 2

デバイスに対応するホストを作成します。



ホストに SNMP インターフェースを追加します:

- IP アドレス/DNS 名とポート番号を入力します。
- ドロップダウンから *SNMP* バージョンを選択します。
- 選択した SNMP バージョンに応じてインターフェースの認証情報を追加します:
  - SNMPv1、v2 ではコミュニティ (通常は'public') のみが必要です。
  - SNMPv3 では、より詳細なオプションが必要です (下記参照)。
- ネイティブ SNMP バルクリクエスト(GetBulkRequest-PDUs) の最大繰り返し値 (デフォルト:10) を指定します。これは SNMPv2 および v3 の `discovery[]` および `walk[]` アイテムにのみ適用されます。この値を高く設定しすぎると、SNMP エージェントのチェックタイムアウトが発生する場合がありますのでご注意ください。
- *Use combined requests* チェックボックスをマークすると、SNMP リクエストの複合処理が許可されます (ネイティブ SNMP バルクリクエスト"walk" および"get" とは無関係です)。

SNMPv3 パラメータ	説明
Context name	SNMP サブネット上のアイテムを識別するためのコンテキスト名を入力します。 このフィールドではユーザーマクロが解決されます。
Security name	セキュリティ名を入力します。 このフィールドではユーザーマクロが解決されます。
Security level	セキュリティレベルを選択します: **noAuthNoPriv** - 認証およびプライバシープロトコルは使用しません **AuthNoPriv** - 認証プロトコルのみ使用し、プライバシープロトコルは使用しません **AuthPriv** - 認証およびプライバシープロトコルの両方を使用します
Authentication protocol	認証プロトコルを選択します - *MD5*、*SHA1*; net-snmp 5.8 以降では *SHA224*、*SHA256*、*SHA384* または *SHA512* も選択可能です。
Authentication passphrase	認証パスフレーズを入力します。 このフィールドではユーザーマクロが解決されます。
Privacy protocol	プライバシープロトコルを選択します - *DES*、*AES128*、*AES192*、*AES256*、*AES192C*(Cisco) または *AES256C*(Cisco)。 プライバシープロトコルのサポートに関する注意事項を参照してください。

SNMPv3 パラメータ	説明
Privacy passphrase	プライバシーパスフレーズを入力します。 このフィールドではユーザーマクロが解決されます。

SNMPv3 の認証情報 (セキュリティ名、認証プロトコル/パスフレーズ、プライバシープロトコル) が間違っている場合:

- *Privacy passphrase* が間違っている場合を除き、Zabbix は net-snmp から ERROR を受信します。*Privacy passphrase* が間違っている場合、Zabbix は net-snmp から TIMEOUT エラーを受信します。
- SNMP インターフェースの可用性は赤 (利用不可) に切り替わります。

Note:
Authentication protocol、*Authentication passphrase*、*Privacy protocol* または *Privacy passphrase* を *Security name* を変更せずに変更した場合、対応する SNMPv3 インターフェースが Zabbix で更新されると通常自動的に適用されます。*Security name* も変更した場合は、すべてのパラメータが即座に更新されます。

自動的に一連のアイテムを追加する提供済みの SNMP テンプレートのいずれかを使用できます。テンプレートを使用する前に、ホストと互換性があることを確認してください。

追加をクリックしてホストを保存します。

プライバシープロトコルのサポート

オペレーティングシステムや net-snmp の設定によっては、一部のプライバシープロトコルが利用できない場合があります。

- 一部の新しいオペレーティングシステム（例：RHEL9）では、net-snmp パッケージで DES のサポートが削除されています。
- AES192 以上の暗号化プロトコルは、RHEL 8、CentOS 8、Oracle Linux 8、Debian 12、Ubuntu LTS 22.04、openSUSE Leap 15.5 より古いオペレーティングシステムでは標準でサポートされていません。

net-snmp ライブラリが AES192+ をサポートしているかどうかを確認するには、以下のいずれかの方法を使用します。

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

出力に`--enable-blumenthal-aes` が含まれていれば、AES192+ がサポートされています。

なお、net-snmp-config は SNMP の開発パッケージ（Debian/Ubuntu では libsnmp-dev、CentOS/RHEL/OL/SUSE では net-snmp-devel）の一部であり、デフォルトではインストールされていない場合があります。

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

出力に `Invalid privacy protocol specified after -3x flag: AES-256` が含まれていれば、AES192+ はサポートされていません。出力に `No hostname specified.` が含まれていれば、AES192+ はサポートされています。

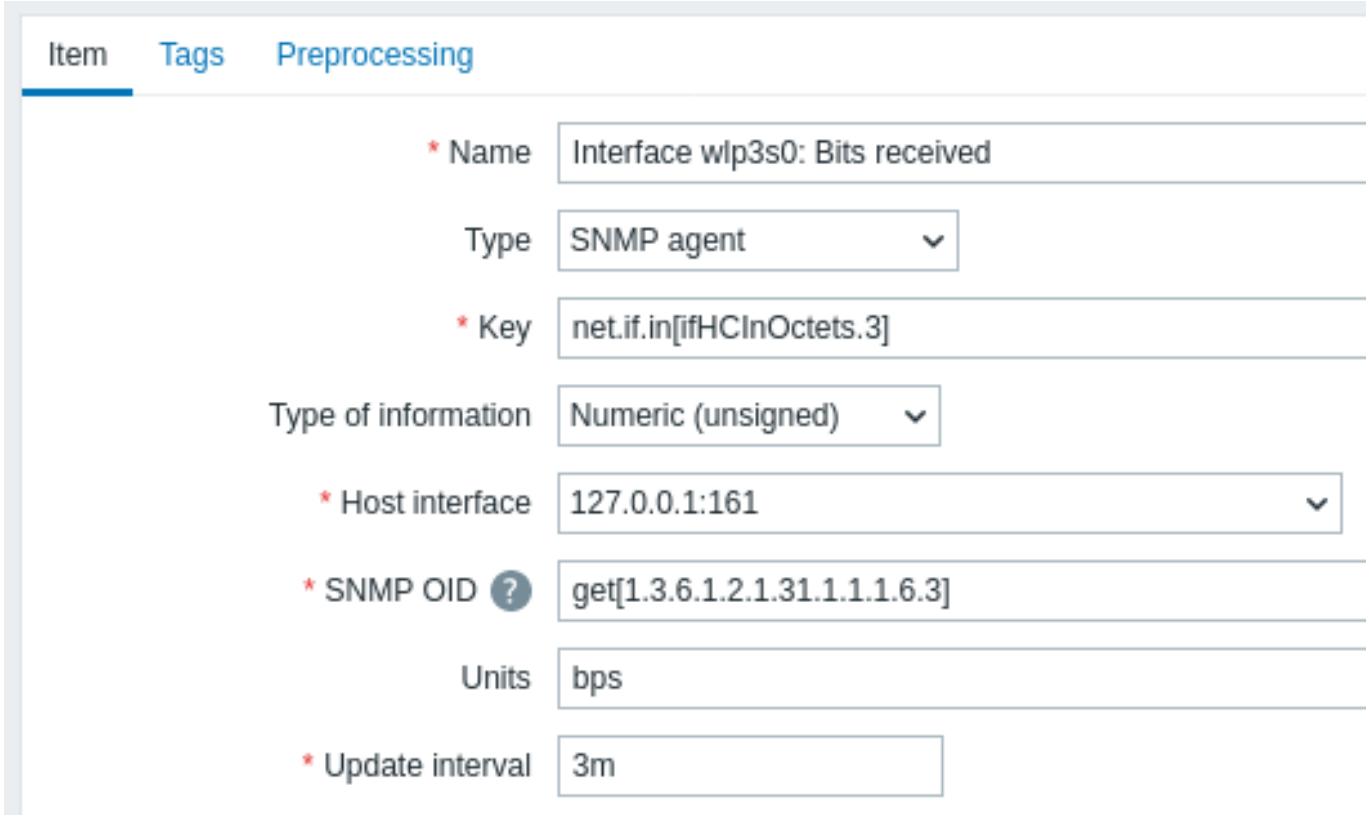
net-snmp ライブラリが AES192 以上のプロトコルをサポートしていない場合は、`--enable-blumenthal-aes` オプションを指定して net-snmp を再コンパイルし、その後`--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config` オプションを指定して Zabbix サーバーを再コンパイルしてください。

ステップ 3

監視用のアイテムを作成します。

それでは、Zabbix に戻り、先ほど作成した SNMP ホストのアイテムをクリックします。ホスト作成時にテンプレートを使用したかどうかによって、ホストに関連付けられた SNMP アイテムのリストが表示されるか、空のリストが表示されます。ここでは、snmpwalk や snmpget で収集した情報を使って自分でアイテムを作成することを前提としますので、アイテムの作成をクリックしてください。

新しいアイテムフォームに必要なパラメータを入力します。



パラメータ	説明
名前	アイテム名を入力します。
タイプ	ここでは **SNMP** エージェントを選択します。
キー	意味のあるキーを入力します。
ホストインターフェース	スイッチやルーターなどの SNMP インターフェースを選択してください。

パラメータ	説明
SNMP OID	OID 値を入力するには、サポートされているいずれかの形式を使用します。 **walk[OID1,OID2,...]** - 値のサブツリーを取得します。 例: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]` このオプションは、ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDUs）を非同期で使用します。 このアイテムのタイムアウト設定は、アイテム設定フォームで設定できます。デバイスに到達できない場合に長い遅延を避けるため、低いタイムアウト値を設定することを検討してください。タイムアウトや失敗が発生した場合、最大 5 回までリトライされます（例：3 秒のタイムアウトの場合、最大 15 秒待機することになります）。 このアイテムをマスターアイテムとして使用し、依存アイテムで前処理を使ってデータを抽出することができます。 1 回の snmpwalk で複数の OID を指定することが可能です。例えば `walk[OID1,OID2,...]` のように指定すると、1 つずつ非同期で処理されます。 バルクリクエストで結果が返らない場合は、バルクリクエストなしで単一レコードの取得が試みられます。 MIB 名もパラメータとしてサポートされているため、`walk[1.3.6.1.2.1.2.2.1.2]` と `walk[ifDescr]` は同じ出力を返します。 複数の OID/MIB を指定した場合（例：`walk[ifDescr,ifType,ifPhysAddress]`）、出力は連結リストとなります。 GetBulk リクエストは SNMPv2 および v3 インターフェースで使用され、SNMPv1 インターフェースでは GetNext が使用されます。バルクリクエストの最大繰り返し数はインターフェースレベルで設定します。 最大繰り返し数パラメータは、1 回のバルク応答で返される OID の最大数を決定します。 値が大きいほどバルク応答が大きくなり、必要な通信回数が減ります。ただし、すべてのデバイスが非常に大きな値をサポートしているわけではなく、問題が発生する可能性があります。 このアイテムは、snmpwalk ユーティリティの-Oe -Ot -On パラメータ付きの出力を返します。 このアイテムをSNMP ディスカバリのマスターアイテムとして使用できます。 **get[OID]** - 単一の値を非同期で取得します。 例: `get[1.3.6.1.2.1.31.1.1.1.6.3]` このアイテムのタイムアウト設定は、アイテム設定フォームで設定できます。デバイスに到達できない場合に長い遅延を避けるため、低いタイムアウト値を設定することを検討してください。タイムアウトや失敗が発生した場合、最大 5 回までリトライされます（例：3 秒のタイムアウトの場合、最大 15 秒待機することになります）。 **OID** - （レガシー）単一のテキストまたは数値 OID を入力して、1 つの値を同期的に取得します。他の値と組み合わせることも可能です。 例: `1.3.6.1.2.1.31.1.1.1.6.3` このオプションの場合、アイテムチェックのタイムアウトはサーバー設定ファイルで設定された値と同じになります。 パフォーマンス向上のため、`walk[OID]` および `get[OID]` アイテムの使用を推奨します。すべての `walk[OID]` および `get[OID]` アイテムは非同期で実行されます。1 つのリクエストの応答を受信する前に他のチェックを開始する必要はありません。DNS 解決も非同期です。 非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPollerで定義）。非同期 SNMP ポーラーの数はStartSNMPPollersパラメータで定義されます。 ネットワークトラフィック統計（いずれの方法でも返される）については、前処理タブで *1* 秒あたりの変化ステップを追加する必要があることに注意してください。そうしないと、SNMP デバイスから累積値が返され、最新の変化量が取得できません。

必須入力フィールドは赤いアスタリスクでマークされています。

アイテムを保存し、監視データ > 最新データで SNMP データを確認してください。

例 1

一般的な例:

パラメータ	説明
OID	1.2.3.45.6.7.8.0 (または.1.2.3.45.6.7.8.0)
キー	< トリガーの参照として使用する一意の文字列 > 例: ”my_param”。

OID は数値または文字列のいずれかの形式で指定できます。ただし、場合によっては文字列 OID を数値表現に変換する必要があります。この目的には snmpget ユーティリティを使用できます:

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

例 2

稼働時間の監視:

パラメータ	説明
OID	MIB::sysUpTime.0
キー	router.uptime
値の型	浮動小数点
単位	uptime
前処理ステップ**:** カスタム乗数	0.01

ネイティブ SNMP バルクリクエスト

walk[OID1,OID2,...] アイテムは、SNMP バージョン 2/3 で利用可能なバルクリクエスト（GetBulkRequest-PDUs）のネイティブ SNMP 機能を使用できます。

SNMP の GetBulk リクエストは、複数の GetNext リクエストを実行し、その結果を 1 つのレスポンスで返します。これにより、通常の SNMP アイテムや SNMP ディスカバリでネットワークの往復回数を最小限に抑えることができます。

SNMP の **walk[OID1,OID2,...]** アイテムは、1 回のリクエストでデータを収集するマスターアイテムとして使用でき、依存アイテムがプリプロセスを使用して必要に応じてレスポンスを解析します。

ネイティブ SNMP バルクリクエストの使用は、SNMP リクエストの結合オプションとは関係ありません。これは、複数の SNMP リクエストを結合する Zabbix 独自の方法です（次のセクションを参照）。

SNMP バルクアイテムでは、パケットの 1 つが失われた場合の失敗を回避するために最大 5 回のリトライが行われます。`get` および `walk` の SNMP アイテムのタイムアウト（アイテムの設定フォームで設定）は、セッション全体に設定されます。タイムアウトは、データが完全に取得されたかどうかに関係なく適用されます。データが部分的に受信された場合（たとえば、複数の OID のうち 1 つだけが正常に収集された場合）、アイテムは「部分的なデータのみ受信」とメッセージが表示され、サポート対象外になります。タイムアウトに達した場合はリトライが行われ、タイムアウトがリセットされ、最後のリクエストが再送信されます。これにより、単一のパケットが失われたり遅れて到着した場合でも、最後のリクエストからセッションを継続できます。デバイスに到達できない場合に長い遅延を避けるため、タイムアウト値を低く設定することを検討してください。タイムアウトや失敗が発生した場合、最大 5 回のリトライが行われるためです（例：3 秒のタイムアウトの場合、最大 15 秒待機する可能性があります）。

複合処理の内部動作

Zabbix サーバーおよびプロキシは、1 回のリクエストで複数の値を SNMP デバイスに問い合わせることができます。これは、いくつかの種類の SNMP アイテムに影響します:

- 通常の SNMP アイテム
- 動的インデックスを持つ SNMP アイテム
- SNMP のローレベルディスカバリルール

同じインターフェース上のすべての SNMP アイテムでパラメータが同一の場合、同時に問い合わせるようにスケジューリングされます。最初の 2 種類のアイテムは、ポーラーによって最大 128 アイテムのバッチで取得されますが、ローレベルディスカバリルールは従来通り個別に処理されます。

下位レベルでは、値を問い合わせるために 2 種類の操作が行われます: 複数の指定オブジェクトの取得と OID ツリーのウォークです。

「取得」には、最大 128 個の変数バインディングを持つ GetRequest-PDU が使用されます。「ウォーク」には、SNMPv1 では GetNextRequest-PDU が、SNMPv2 および SNMPv3 では「max-repetitions」フィールドが最大 128 の GetBulkRequest が使用されます。

したがって、各 SNMP アイテムタイプに対する複合処理の利点は以下の通りです:

- 通常の SNMP アイテムは「取得」の改善の恩恵を受けます。
- 動的インデックスを持つ SNMP アイテムは、「取得」と「ウォーク」の両方の改善の恩恵を受けます。「取得」はインデックスの検証に、「ウォーク」はキャッシュの構築に使用されます。

- SNMP ローレベルディスカバリルールは「ウォーク」の改善の恩恵を受けます。

ただし、すべてのデバイスが 1 回のリクエストで 128 個の値を返すことができるわけではないという技術的な問題があります。常に適切な応答を返すものもあれば、潜在的な応答がある制限を超えると「tooBig(1)」エラーで応答したり、まったく応答しなくなるものもあります。

特定のデバイスに対して最適な問い合わせオブジェクト数を見つけるために、Zabbix は以下の戦略を使用します。最初は慎重に、1 回のリクエストで 1 つの値を問い合わせます。それが成功した場合、1 回のリクエストで 2 つの値を問い合わせます。それが再び成功した場合、1 回のリクエストで 3 つの値を問い合わせ、以降は問い合わせるオブジェクト数を 1.5 倍にしていきます。これにより、リクエストサイズのシーケンスは次のようになります: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128。

ただし、デバイスが適切な応答を返さなくなった場合 (例えば、42 個の変数で)、Zabbix は 2 つのことを行います。

まず、現在のアイテムバッチに対して、1 回のリクエストでのオブジェクト数を半分にして 21 個の変数を問い合わせます。デバイスが生きていれば、この問い合わせはほとんどの場合機能するはずです。なぜなら、28 個の変数で動作することが分かっており、21 個はそれよりもかなり少ないからです。それでも失敗した場合、Zabbix は値を 1 つずつ問い合わせる方法に戻ります。この時点でも失敗する場合は、デバイスが確実に応答していないため、リクエストサイズが問題ではありません。

Zabbix が次のアイテムバッチに対して行う 2 つ目のことは、最後に成功した変数数 (この例では 28) から始めて、リクエストサイズを 1 ずつ増やしながら制限に達するまで続けることです。例えば、最大応答サイズが 32 変数だとすると、次のリクエストは 29, 30, 31, 32, 33 個のサイズになります。最後のリクエストは失敗し、Zabbix は今後 33 個のサイズのリクエストを発行しません。それ以降、Zabbix はこのデバイスに対して最大 32 個の変数を問い合わせます。

この変数数で大きなクエリが失敗した場合、2 つの可能性があります。デバイスが応答サイズを制限する正確な基準は分かりませんが、変数数を使ってそれを近似しようとします。したがって、1 つ目の可能性は、この変数数が一般的なケースでデバイスの実際の応答サイズ制限付近であることです: 時には応答が制限より少なく、時にはそれより多い場合があります。2 つ目の可能性は、どちらかの方向で UDP パケットが単純に失われたことです。これらの理由から、Zabbix が失敗したクエリを受け取った場合、最大変数数を減らしてデバイスの快適な範囲にさらに近づけようとしますが、最大 2 回までしか行いません。

上記の例では、32 個の変数でのクエリがたまたま失敗した場合、Zabbix はカウントを 31 に減らします。それも失敗した場合、Zabbix はカウントを 30 に減らします。ただし、Zabbix はカウントを 30 未満には減らしません。なぜなら、それ以降の失敗はデバイスの制限ではなく、UDP パケットの損失によるものと見なすからです。

ただし、デバイスが他の理由で複合リクエストを適切に処理できず、上記のヒューリスティックが機能しない場合は、各インターフェースに「複合リクエストを使用」設定があり、そのデバイスに対して複合リクエストを無効にすることができます。

複合リクエストが部分的または不正な応答を引き起こし、その結果、1 秒あたり (デルタ) の計算が不正確になる場合 (例えば、インターフェースカウンタの見かけ上のスパイクなど)、該当するインターフェースの複合リクエストを使用を無効にして、アイテムごとに個別のクエリを強制してください。これにより、偽のスパイクが防止されることがよくあります。あるいは、非同期で実行され、インターフェースごとの複合リクエストを使用のバッチ処理の対象外となる非同期 `get[]` または `walk[]` アイテムの使用を検討してください。これらは、複合リクエスト関連の問題を回避するために従来の同期 OID チェックの代わりに使用できます。影響を受けるデバイスを特定するには、概要セクションに示されているものと同様のサーバー/プロキシのログエントリを探してください。

さらに、インターフェースが頻繁に利用不可になる場合は、リクエストの頻度を減らすためにZabbix サーバーまたはZabbix プロキシの設定ファイルで `UnavailableDelay` パラメータを増やす必要があるかもしれません。ディスカバリや OID ウォーク中に部分的なデータが受信された場合、アイテムがサポート対象外になることがあります。

1 ダイナミックインデックス

概要

必要なインデックス番号（例えば、ネットワーク・インターフェース）を SNMP OID の中から見つけることができるかもしれませんが
 インデックス番号が常に同じであるとは限りません。

インデックス番号は動的であり、時間の経過とともに変化し、その結果、アイテムが動作しなくなる可能性があります。

このようなシナリオを回避するために、インデックス番号が変化する可能性を考慮した OID を定義することが可能です。

たとえば、次のようなインデックス値を取得する必要があるとします。
 Cisco のデバイスの **GigabitEthernet0/1** インターフェイスに対応する **ifInOctets** に対応する
 インデックス値を取得する必要がある場合、次の OID を使用します。

```
ifInOctets["index", "ifDescr", "GigabitEthernet0/1"]
```

文法

OID の特別な構文が使用されます。

<OID of data>[”index”,”<base OID of index>”,”<string to search for>”]

パラメーター	説明
OID of data	アイテムのデータ検索に使用するメイン OID。
index	処理方法。現在、1 つの方法がサポートされています: **index** - インデックスを検索し、データ OID に追記する。
base OID of index	この OID を検索して、文字列に対応するインデックス値を取得する。
string to search for	検索時に値と完全一致するために使用する文字列。大文字と小文字を区別する。

例

apache プロセスのメモリ使用量を取得します。

この OID 構文を使用する場合、以下のようになります。

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index", "HOST-RESOURCES-MIB::hrSWRunPath","/usr/sbin/apache2"]
```

インデックス番号はここで検索されます。

```
...
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
...
```

これで、5388 というインデックスができました。
 このインデックスは、データ OID に追加され、目的の値を受け取ります。

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

index lookup のキャッシュ

動的インデックスアイテムが要求された場合、Zabbix はインデックスのベース OID の下にある
 SNMP テーブル全体を取得し、キャッシュします。
 これは、別のアイテムが同じベース OID を後で参照する場合に備えて行われます。
 Zabbix は監視ホストに再度問い合わせる代わりに、キャッシュからインデックスを検索します。
 各ポーラープロセスは別々のキャッシュを使用していることに注意してください。

その後のすべての値取得操作では、見つかったインデックスのみが検証されます。
 インデックスが変更されていない場合、値が要求されます。
 変更されたインデックスに遭遇した各ポーラは、インデックス SNMP テーブルを再度検索し、キャッシュを再構築します。

2 特殊な OID

最もよく使われる SNMP OID のいくつかは、Zabbix によって自動的に数値表現に変換されます。例えば、**ifIndex** は **1.3.6.1.2.1.2.2.1.1** に、**ifIndex.0** は **1.3.6.1.2.1.2.2.1.1.0** に変換されます。

この表は、特殊な OID のリストです。

特殊 OID	識別子	説明
ifIndex	1.3.6.1.2.1.2.2.1.1	A unique value for each interface.
ifDescr	1.3.6.1.2.1.2.2.1.2	A textual string containing information about the interface.This string should include the name of the manufacturer, the product name and the version of the hardware interface.
ifType	1.3.6.1.2.1.2.2.1.3	The type of interface, distinguished according to the physical/link protocol(s) immediately 'below' the network layer in the protocol stack.
ifMtu	1.3.6.1.2.1.2.2.1.4	The size of the largest datagram which can be sent / received on the interface, specified in octets.
ifSpeed	1.3.6.1.2.1.2.2.1.5	An estimate of the interface's current bandwidth in bits per second.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	The interface's address at the protocol layer immediately 'below' the network layer in the protocol stack.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	The current administrative state of the interface.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	The current operational state of the interface.
ifInOctets	1.3.6.1.2.1.2.2.1.10	The total number of octets received on the interface, including framing characters.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	The number of subnetwork-unicast packets delivered to a higher-layer protocol.

特殊 OID	識別子	説明
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	The number of non-unicast (i.e., subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
ifInErrors	1.3.6.1.2.1.2.2.1.14	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	The total number of octets transmitted out of the interface, including framing characters.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	The number of outbound packets that could not be transmitted because of errors.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	The length of the output packet queue (in packets).

3 MIB ファイル

はじめに

MIB は Management Information Base（管理情報ベース）の略です。MIB ファイルを使用すると、OID（オブジェクト識別子）のテキスト表現を利用できます。Zabbix で SNMP デバイスを監視する際に生の OID を使用することも可能ですが、テキスト表現の方が使いやすい場合は、MIB ファイルをインストールする必要があります。

例えば、

`ifHCOutOctets`

は OID のテキスト表現であり、

`1.3.6.1.2.1.31.1.1.1.10`

に対応します。

MIB ファイルのインストール

Debian 系システムの場合:

```
apt install snmp-mibs-downloader
download-mibs
```

RedHat 系システムの場合:

```
dnf install net-snmp-libs
```

MIB ファイルの有効化

RedHat 系のシステムでは、MIB ファイルはデフォルトで有効になっています。Debian 系のシステムでは、`/etc/snmp/snmp.conf` ファイルを編集し、`mibs :`と記載されている行をコメントアウトする必要があります。

```
# snmpパッケージはライセンス上の理由によりMIBファイルなしで提供されるため、
# デフォルトではMIBのロードは無効になっています。MIBを追加した場合は、
# 以下の行をコメントアウトすることでロードを再度有効にできます。
mibs :
```

MIB ファイルのテスト

SNMP MIB のテストは、snmpwalk ユーティリティを使用して行うことができます。インストールされていない場合は、以下の手順を使用してください。

Debian 系システムの場合:

```
apt install snmp
```

RedHat 系システムの場合:

```
dnf install net-snmp-utils
```

その後、ネットワークデバイスにクエリを実行したときに、次のコマンドでエラーが発生しないことを確認してください:

```
$ snmpwalk -v 2c -c public <NETWORK DEVICE IP> ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Zabbix での MIB の使用

最も重要な点は、Zabbix のプロセスは MIB ファイルに加えられた変更を認識しないということです。そのため、MIB ファイルを変更した後は、必ず Zabbix サーバーまたはプロキシを再起動する必要があります。例：

```
systemctl restart zabbix-server
```

これで、MIB ファイルに加えた変更が有効になります。

カスタム MIB ファイルの使用

すべての GNU/Linux ディストリビューションには標準の MIB ファイルが付属しています。しかし、一部のデバイスベンダーは独自の MIB ファイルを提供しています。

たとえば、CISCO-SMI MIB ファイルを使用したいとします。以下の手順でダウンロードおよびインストールできます。

```
wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
mkdir -p /usr/local/share/snmp/mibs
grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/loca
cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

これで使用できるようになります。MIB ファイルからオブジェクト *ciscoProducts* の名前を OID に変換してみましょう。

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

OID の代わりにエラーが表示される場合は、前述のコマンドがすべてエラーなく実行されたことを確認してください。

オブジェクト名の変換が成功した場合、カスタム MIB ファイルを使用する準備ができています。クエリで使用されている MIB 名のプレフィックス（*CISCO-SMI::*）に注意してください。これはコマンドラインツールや Zabbix で使用する際にも必要になります。

この MIB ファイルを Zabbix で使用する前に、Zabbix サーバー/プロキシを再起動することを忘れないでください。

> **Attention:**
> MIB ファイルには依存関係がある場合があることに注意してください。つまり、ある MIB が別の MIB を必要とする場合があります。これらの依存関係を満たすためには、影響を受けるすべての MIB ファイルをインストールする必要があります。

3 SNMP トラップ

概要

SNMP トラップの受信は、SNMP 対応デバイスへのクエリとは逆です。

この場合、情報は SNMP 対応デバイスから snmptrapd へ送信され、ファイルから Zabbix サーバーまたは Zabbix プロキシによって収集または” トラップ” されます。

通常、何らかの条件が変化するとトラップが送信され、エージェントはポート 162 でサーバーに接続します (クエリに使用されるエージェント側のポート 161 とは対照的です)。トラップを使用すると、クエリ間隔中に発生し、クエリデータによって見逃される可能性のあるいくつかの短い障害を検出できます。

Zabbix での SNMP トラップの受信は、**snmptrapd** と、トラップを Zabbix に渡すためのメカニズムの 1 つ (Bash または Perl スクリプト、SNMPTT) で動作するように設計されています。

Note:
Zabbix の構成後にトラップ監視をセットアップする最も簡単な方法は、Bash スクリプトソリューションを使用することです。これは、最近のディストリビューションでは Perl と SNMPTT が欠落していることが多く、より複雑な構成が必要になるためです。ただし、このソリューションでは、`traphandle` として構成されたスクリプトが使用されます。本番システムでのパフォーマンスを向上させるには、組み込みの Perl ソリューション (`do perl` オプション付きのスクリプトまたは SNMPTT) を使用してください。

トラップを受信するワークフロー:

1. `snmptrapd` がトラップを受信
2. `snmptrapd` はトラップをレシーバースクリプト (Bash、Perl) または SNMPTT に渡す
3. 受信者はトラップを解析、フォーマットし、ファイルに書き込む
4. Zabbix SNMP トラッパーは、トラップファイルを読み取り、解析
5. Zabbix はトラップごとに、受信したトラップアドレスと一致するホストインターフェースを持つすべての”SNMP トラッパー” アイテムを検索します。マッチングでは、ホストインターフェースで選択した”IP” または”DNS” のみが使用されることに注意してください。
6. 見つかったアイテムごとに、トラップは `snmptrap[regexp]` の正規表現と比較されます。トラップは一致したすべてのアイテムの値として設定されます。一致するアイテムが見つからず、snmptrap.fallback アイテムが存在する場合、トラップはその値として設定されます。
7. トラップがアイテムの値として設定されていない場合、Zabbix はデフォルトで一致しないトラップをログに記録します。(これは、管理 > 一般設定 > その他の” マッチしない SNMP トラップをログに記録” で設定されます。)

HA フェイルオーバーに関する注意事項

高可用性 (HA) ノードの切り替え中、Zabbix は最後の ISO 8601 タイムスタンプ内の最後のレコードの後に処理を続行します。同じレコードが見つからない場合は、タイムスタンプのみを使用して最後の位置を識別します。

SNMP トラップの設定

このアイテムタイプには、次のフロントエンド設定が必要です。

1. ホスト用の SNMP インターフェースを作成

- データ収集 > ホストで、ホストを作成/編集し、インターフェースフィールドに、インターフェースタイプ”SNMP” を追加、IP または DNS アドレスを指定します。

 受信した各トラップのアドレスは、すべての SNMP インターフェイスの IP および DNS アドレスと比較され、対応するホストを見つけます。

2. アイテムの設定

- データ収集 > ホストで、必要なアイテムを作成/編集します。
- キーフィールドで、SNMP トラップキーの 1 つを使用します。

キー		
説明	戻り値	コメント
snmptrap[regexp] **regexp** で指定された正規表現に一致するすべての SNMP トラップをキャッチします。regexp が指定されていない場合、すべてのトラップをキャッチします。	SNMP トラップ	このアイテムは、SNMP インターフェースに対してのみ設定できます。 このアイテムキーのパラメーターでは、ユーザーマクロとグローバル正規表現がサポートされています。
snmptrap.fallback そのインターフェースの snmptrap[] アイテムのいずれによってもキャッチされなかったすべての SNMP トラップをキャッチします。	SNMP トラップ	このアイテムは、SNMP インターフェースに対してのみ設定できます。

Note:
現時点では、複数行の正規表現一致はサポートされていません。

- データ型をタイムスタンプが解析されるように” ログ” に設定します。” 数値” などの他の形式も受け入れられますが、カスタムトラップハンドラーが必要になる場合があります。

SNMP トラップ監視の設定

Zabbix サーバー/プロキシの設定

トラップを読み取るには、Zabbix サーバーまたはプロキシが SNMP トラッパープロセスを開始し、SNMPTT または Bash/Perl トラップレシーバーによって書き込まれているトラップファイルを指すように構成されている必要があります。これを行うには設定ファイル (zabbix_server.confまたはzabbix_proxy.conf) を編集します。

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
systemd パラメーター**PrivateTmp**が使用されている場合、このファイルは*/tmp* では機能しません。

Bash トラップレシーバーの設定

要件: snmptrapd のみ。

Bash トラップレシーバースクリプトを使用して、trapper ファイルを使用して snmptrapd から Zabbix サーバーにトラップを渡すことができます。設定するには、snmptrapd 設定ファイル (`snmptrapd.conf`) に `traphandle` オプションを追加します。例を参照してください。

Note:
snmptrapd は設定の変更を取得するために再起動する必要がある場合があります。

Perl トラップレシーバーの設定

要件: Perl、--enable-embedded-perl でコンパイルされた Net-SNMP（Net-SNMP 5.4 以降はデフォルトで有効）

Perl トラップレシーバー（misc/snmptrap/zabbix_trap_receiver.pl を参照）は、snmptrapd から直接 Zabbix サーバーにトラップを渡すために使用できます。設定方法は以下の通りです。

- Perl スクリプトを snmptrapd の設定ファイル（snmptrapd.conf）に追加します。例:

```
perl do "[PERL レシーバースクリプトへのフルパス]";
```

- レシーバーを設定します。例:

```
$SNMPTrapperFile = '[TRAP ファイル]';
$DateTimeFormat = '[日付時刻フォーマット]';
```

Note:
snmptrapd は、設定の変更を反映するために再起動が必要な場合があります。

Note:
スクリプト名が引用符で囲まれていない場合、snmptrapd は以下のようなメッセージとともに起動を拒否します。


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

SNMPTT の設定

まず、snmptrapd を SNMPTT を使用するように設定する必要があります。

Note:
最高のパフォーマンスを得るために、SNMPTT は **snmptthandler-embedded** を使用してデーモンとして設定し、トラップを渡すようにしてください。SNMPTT の設定の手順を参照してください。

SNMPTT がトラップを受信するように設定されたら、`snmptt.ini` を設定します:

1. NET-SNMP パッケージの Perl モジュールの使用を有効にします:

```
net_snmp_perl_enable = 1
```

2. Zabbix が読み取るトラップファイルにトラップを記録します:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. 日時フォーマットを設定します:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

> **Warning:**
> "net-snmp-perl" パッケージは RHEL 8.0-8.2 で削除され、RHEL 8.3 で再追加されました。詳細は既知の問題を参照してください。

次に、Zabbix がトラップを認識できるようにトラップのフォーマットを設定します（snmptt.conf を編集）:

1. 各 FORMAT 文は"ZBXTRAP [address]" で始める必要があります。ここで、[address] は Zabbix の SNMP インターフェースの IP および DNS アドレスと比較されます。例:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. SNMP トラップのフォーマットについては、以下を参照してください。

> **Attention:**
> 不明なトラップは使用しないでください。Zabbix はそれらを認識できません。
 不明なトラップは、snmptt.conf で一般的なイベントを定義することで処理できます:

> `EVENT general .* "General event" Normal`

SNMP トラップのフォーマット

すべてのカスタマイズされた Perl トラップレシーバーおよび SNMPTT トラップの設定は、以下のようにトラップをフォーマットする必要があります:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

ここで

- [timestamp] - "%Y-%m-%dT%H:%M:%S%z" 形式のタイムスタンプ
- ZBXTRAP - この行で新しいトラップが始まることを示すヘッダー
- [address] - このトラップのホストを見つけるために使用される IP アドレス

"ZBXTRAP" と"[address]" は、処理中にメッセージから切り取られることに注意してください。トラップが他の形式でフォーマットされている場合、Zabbix はトラップを予期せず解析する可能性があります。

トラップの例:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

これは、IP=192.168.1.1 の SNMP インターフェースに対して、次のようなトラップになります:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

システム要件

> **Note:**
> アイテム値が正しい形式で表示されるように、MIB ファイルをインストールすることを推奨します。MIB ファイルがない場合、値が UTF-8 ではなく HEX で表示される、またはその逆など、フォーマットの問題が発生する可能性があります。

ラージファイルサポート

Zabbix は、SNMP トラッパーのラージファイルをサポートしています。Zabbix が読み取れる最大ファイルサイズは 2^63 (8 EiB) です。ファイルシステムによってはファイルサイズに下限があることに注意してください。

ログローテーション

Zabbix はログローテーションシステムを提供していません。これはユーザーが管理する必要があります。ログローテーションは、トラップが失われないように、まず古いファイルの名前を変更し、その後で削除する必要があります。

1. Zabbix は最後に記録された位置でトラップファイルを開き、ステップ 3 に進みます。

2. Zabbix は、現在開いているファイルの inode 番号を定義されたトラップファイルの inode 番号と比較して、現在開いているファイルがローテーションされたかどうかを確認します。開いているファイルがない場合、Zabbix は最後の位置をリセットし、ステップ 1 に戻ります。
3. Zabbix は現在開いているファイルからデータを読み取り、新しい位置を設定します。
4. 新しいデータが解析されます。これがローテーションされたファイルであった場合、ファイルを閉じてステップ 2 に戻ります。
5. 新しいデータがなかった場合、Zabbix は 1 秒間スリープし、ステップ 2 に戻ります。

ファイルシステム

Trap ファイルの実装のため、Zabbix はファイルを区別するために inode をサポートする
 ファイルシステムを必要とします (この情報は stat() 呼出で取得します)。

異なる SNMP プロトコルバージョンを使用したセットアップ例

この例では、snmptrapd と Bash 受信スクリプトを使用して、トラップを Zabbix サーバーに渡します。

セットアップ手順:

1. Zabbix で SNMP トラッパーを起動し、トラップファイルを設定します。zabbix_server.conf に以下を追加します:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Bash スクリプトを/usr/sbin/zabbix_trap_handler.sh にダウンロードします:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/Dock
```

必要に応じて、スクリプト内の ZABBIX_TRAPS_FILE 変数を調整してください。デフォルト値を使用する場合は、親ディレクトリを先に作成してください:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. snmtrapd.conf に以下を追加します（動作する例を参照）

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> snmptrapd は、設定の変更を反映させるために再起動が必要な場合があります。

4. SNMP アイテム TEST を作成します（初期の設定要件に注意）:

タイプ: SNMP トラップ
情報のタイプ: ログホストインターフェース: SNMP 127.0.0.1
キー: snmptrap["linkUp"]
ログ時刻フォーマット: yyyyMMdd.hhmmss

ISO 8601 の日付と時刻のフォーマットが使用されていることに注意してください。

5. 次に、選択した SNMP プロトコルバージョン用に snmptrapd を設定し、snmptrap ユーティリティを使用してテストトラップを送信します。

SNMPv1, SNMPv2

SNMPv1 および SNMPv2 プロトコルは、「コミュニティストリング」認証に依存しています。以下の例では「secret」をコミュニティストリングとして使用します。SNMP トラップ送信側でも同じ値に設定する必要があります。

SNMPv2 は現在も多くの本番環境で使用されていますが、暗号化や実際の送信者認証を提供しません。データは平文で送信されるため、これらのプロトコルバージョンはプライベートネットワークなどの安全な環境でのみ使用し、パブリックネットワークや第三者ネットワークでは絶対に使用しないでください。

SNMP バージョン 1 は 64 ビットカウンタをサポートしておらず、レガシープロトコルと見なされているため、現在はほとんど使用されていません。

SNMPv1 または SNMPv2 トラップの受信を有効にするには、snmptrapd.conf に以下の行を追加します。「secret」は SNMP トラップ送信側で設定したコミュニティストリングに置き換えてください:

```
authCommunity log,execute,net secret
```

次に、snmptrap を使用してテストトラップを送信できます。この例では一般的な「リンクアップ」OID を使用します:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 は SNMPv1/v2 のセキュリティ問題に対応し、認証と暗号化を提供します。MD5 または複数の SHA 認証方式、および DES/複数の AES 暗号を使用できます。

SNMPv3 の受信を有効にするには、snmptrapd.conf に以下の行を追加します:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

Attention:
このユーザーセキュリティモデルでスクリプトの実行を許可する「execute」キーワードに注意してください。

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

Warning:
AES192 や AES256 などの強力な暗号化方式を使用したい場合は、バージョン 5.8 以降の net-snmp を使用してください。configure オプション: --enable-blumenthal-aes で再コンパイルが必要な場合があります。古いバージョンの net-snmp は AES192/AES256 をサポートしていません。参考: Strong Authentication or Encryption。

検証

どちらの例でも、/var/lib/zabbix/snmptraps/snmptraps.log に以下のような行が表示されます:

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Zabbix のアイテム値は次のようになります:

```
2024-01-30 10:04:23 2024-01-30 10:04:21

2024-01-30T10:04:21+0200 UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Perl を使った例:

```
2024-01-30T11:42:54+0200 ZBXTRAP 127.0.0.1
PDU INFO:
  receivedfrom                   UDP: [127.0.0.1]:58649->[127.0.0.1]:162
  notificationtype               TRAP
  version                        1
  community                      public
  errorstatus                    0
  transactionid                  1
  requestid                      2101882550
  messageid                      0
  errorindex                     0
VARBINDS:
  DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (457671) 1:16:16.71
  SNMPv2-MIB::snmpTrapOID.0      type=6  value=OID: IF-MIB::linkUp.0
```

参照

- Zabbix blog article on SNMP traps
- Configuring snmptrapd (net-snmp の公式ドキュメント)
- Configuring snmptrapd to receive SNMPv3 notifications (net-snmp の公式ドキュメント)

4 IPMI チェック

概要

Zabbix では、Intelligent Platform Management Interface（IPMI）デバイスのヘルスと可用性を監視できます。IPMI チェックを実行するには、Zabbix サーバーが最初に IPMI サポート付きで構成されている必要があります。

IPMI は、コンピュータシステムのリモート「ライトアウト」または「アウトオブバンド」管理のための標準化されたインターフェースです。これにより、オペレーティングシステムやマシンの電源が入っているかどうかに関係なく、いわゆる「アウトオブバンド」管理カードからハードウェアの状態を直接監視できます。

Zabbix の IPMI 監視は、IPMI サポートを持つデバイス（HP iLO、DELL DRAC、IBM RSA、Sun SSP など）でのみ動作します。

IPMI マネージャープロセスは、IPMI ポーラーによって IPMI チェックをスケジューリングします。ホストは常に 1 つの IPMI ポーラーによってのみポーリングされるため、BMC コントローラーへのオープン接続数が削減されます。そのため、BMC コントローラーの過負荷を心配することなく、IPMI ポーラーの数を増やすことができます。IPMI マネージャープロセスは、少なくとも 1 つの IPMI ポーラーが開始されると自動的に開始されます。

IPMI チェックの既知の問題も参照してください。

設定

ホストの設定

IPMI チェックを処理するために、ホストを設定する必要があります。
 IPMI インターフェイスをを追加し、それぞれの IP アドレスおよびポート番号を指定し、IPMI 認証パラメーターを定義する必要があります。

詳細は、configuration of hosts を参照してください。

Server の設定

デフォルトでは、Zabbix サーバは IPMI ポーラーを起動するように設定されていないため、IPMI アイテムを追加しても動作しません。
 これを変更するには、Zabbix サーバ設定ファイル (zabbix_server.conf) を root ユーザで開き、
 次の行を探します。

```
# StartIPMIPollers=0
```

この行をアンコメントし、ポーラー数を 3 に設定すると、次のようになります。

```
StartIPMIPollers=3
```

ファイルを保存し、その後 zabbix_server を再起動します。

item の設定

ホストレベルでconfiguring an itemを行う場合:

- *Type* として'IPMI エージェント' を選択します。
- ホスト内で一意である item key (例えば、ipmi.fan.rpm) を入力します。
- *Host interface* には、関連する IPMI インターフェース (IP アドレスおよびポート番号) を選択します。IPMI インターフェースはホスト上に存在する必要があることに注意してください。
- IPMI センサー * (Dell Poweredge の'FAN MOD 1A RPM' など) からメトリックを取得するように指定します。デフォルトでは、センサー ID を指定する必要があります。接頭辞を使用することも可能です。
  - `id:` - センサーの ID を指定します。
  - `name:` - センサーのフルネームを指定します。これは、フルネームを指定することによってのみセンサを区別できる場合に役立ちます。
- それぞれの情報の種類を選択します (この場合は、'Numeric (float)' ディスクリートセンサーの場合は'Numeric (unsigned)')、単位（多くの場合'rpm')、およびその他の必要な項目属性を選択します。

サポートされているチェック

IPMI エージェントは、IPMI センサー関連の情報を返す組み込みアイテム **ipmi.get** をサポートしており、IPMI センサーのディスカバリに使用できます。
戻り値 : *JSON* オブジェクト

タイムアウトとセッションの終了

IPMI メッセージのタイムアウトとリトライ回数は、OpenIPMI ライブラリで定義されています。OpenIPMI の現在の設計上、これらの値を Zabbix で設定可能にすることはできません。インターフェースレベルでもアイテムレベルでも同様です。

LAN 用の IPMI セッション非アクティブタイムアウトは 60±3 秒です。現在のところ、OpenIPMI で Activate Session コマンドの定期的な送信を実装することはできません。Zabbix から特定の BMC への IPMI アイテムチェックが、BMC で設定されたセッションタイムアウトよりも長い間行われない場合、タイムアウト後の次の IPMI チェックは、個々のメッセージタイムアウト、リトライ、または受信エラーによりタイムアウトします。その後、新しいセッションが開かれ、BMC のフルリスキャンが開始されます。新しいセッションを管理するために新しい UDP ポートが開かれる場合があります。非アクティブとは、送信リクエストと受信レスポンスの両方がない状態を指します。不要な BMC のリスキャンを避けたい場合は、BMC で設定された IPMI セッション非アクティブタイムアウトよりも短い IPMI アイテムのポーリング間隔を設定することを推奨します。

IPMI 離散センサーに関する注意事項

ホスト上のセンサーを見つけるには、**DebugLevel=4** を有効にして Zabbix サーバーを起動します。数分待って、Zabbix サーバーログファイルでセンサー検出の記録を探します:

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

IPMI センサーのタイプや状態をデコードするために、IPMI 2.0 仕様書のコピーが利用可能です（IPMI 仕様書の今後の更新は予定されていませんのでご注意ください）。

最初に確認するパラメータは「reading_type」です。仕様書の「Table 42-1, Event/Reading Type Code Ranges」を使用して「reading_type」コードをデコードします。例のほとんどのセンサーは「reading_type:0x1」で、これは「threshold」センサーを意味します。「Table 42-3, Sensor Type Codes」によると、「type:0x1」は温度センサー、「type:0x2」は電圧センサー、「type:0x4」はファンなどを意味します。しきい値センサーは、温度、電圧、回転数などの連続パラメータを測定するため、時々「アナログ」センサーと呼ばれます。

別の例として、「reading_type:0x3」のセンサーがあります。「Table 42-1, Event/Reading Type Code Ranges」によると、reading type コード 02h-0Ch は「Generic Discrete」センサーを意味します。離散センサーは最大 15 の状態（言い換えれば最大 15 の意味のあるビット）を持つことができます。例えば、「type:0x7」の「CATERR」センサーの場合、「Table 42-3, Sensor Type Codes」によると、このタイプは「Processor」を意味し、個々のビットの意味は次の通りです: 00h（最下位ビット）- IERR、01h - Thermal Trip など。

例には「reading_type:0x6f」のセンサーもいくつかあります。これらのセンサーについては、「Table 42-1, Event/Reading Type Code Ranges」により、「Table 42-3, Sensor Type Codes」を使用してビットの意味をデコードすることが推奨されています。例えば、「Power Unit Stat」センサーは「type:0x9」で、これは「Power Unit」を意味します。オフセット 00h は「PowerOff/Power Down」を意味します。つまり、最下位ビットが 1 の場合、サーバーの電源がオフになっています。このビットをテストするには、マスク「1」を使ってbitand関数を使用できます。トリガー式は次のようになります。

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

これにより、サーバーの電源オフを警告できます。

OpenIPMI-2.0.16, 2.0.17, 2.0.18, 2.0.19 のディスクリートセンサーの名前に関する注意事項

OpenIPMI-2.0.16, 2.0.17, 2.0.18 のディスクリートセンサーの名前には”0” (または他の数字や文字) を
 末尾に付加していることがよくあります。
 例えば、ipmitool と OpenIPMI-2.0.19 では、センサーの名前は”PhysicalSecurity” や”CATERR” のように表示されます。
 OpenIPMI-2.0.16, 2.0.17, 2.0.18 では”PhysicalSecurity0” または”CATERR0” と表示されます。

OpenIPMI-2.0.16、2.0.17、2.0.18 を使用して Zabbix サーバで IPMI item を設定する場合、
 IPMI エージェントアイテムの *IPMI sensor* フィールドにこれらの名前を 0 として使用してください。
 Zabbix サーバを OpenIPMI-2.0.19(またはそれ以降) を使用する新しい Linux ディストリビューションにアップグレードした場合、
 これらの IPMI センサーのアイテムは「NOT SUPPORTED」になります。そのため
 IPMI センサー名を変更し（末尾の’0’ を削除)、再び”Enabled ” になる前に、しばらく待つ必要があります。

しきい値センサーと離散センサーの同時利用に関する注意事項

一部の IPMI エージェントは、同じ名前でしきい値センサーと離散センサーの両方を提供します。常にしきい値センサーが優先されます。

接続終了時の注意事項

IPMI チェックが行われなかった場合 (何らかの理由: ホストの IPMI 項目が全て無効/サポートされていない、
 ホストが無効/削除された、ホストがメンテナンス中であるなど)
 Zabbix server / proxy を起動した時間に応じて 3～4 時間後に Zabbix server / proxy から IPMI 接続が切断されます。

5 シンプルチェック

概要

シンプルチェックは、サービスのリモートエージェントレスチェックに通常使用されます。

シンプルチェックには Zabbix エージェントは必要ありません。Zabbix サーバー/プロキシがシンプルチェックの処理 (外部接続など) を担当します。

シンプルチェックの使用例:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
シンプルチェックアイテムの設定にあるユーザー名とパスワードのフィールド (255 文字まで) は、VMware 監視アイテムで使用されます。それ以外の場合は無視されます。

サポートされているチェック

アイテムキーは、オプションパラメータや追加情報なしでリストされています。アイテムキーをクリックすると、詳細が表示されます。

VMware 監視のアイテムキーも参照してください。

アイテムキー	説明
icmpping	ICMP ping によるホストの到達性。
icmppingloss	パケット損失率 (%)。
icmppingretry	リトライ付き ICMP ping によるホストの到達性。
icmppingsec	ICMP ping の応答時間。
net.tcp.service	サービスが稼働し、TCP 接続を受け付けているかをチェックします。
net.tcp.service.perf	TCP サービスのパフォーマンスをチェックします。
net.udp.service	サービスが稼働し、UDP リクエストに応答しているかをチェックします。
net.udp.service.perf	UDP サービスのパフォーマンスをチェックします。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 ICMP ping によるホストの到達性。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメータ:

- **target** - ホストの IP または DNS 名
- **packets** - パケット数
- **interval** - 連続するパケット間の時間 (ミリ秒)
- **size** - パケットサイズ (バイト)
- **timeout** - タイムアウト (ミリ秒)
- **options** - リダイレクトを許可する場合に使用: 空 (デフォルト値) の場合、リダイレクトされた応答はターゲットホストがダウンしているとみなされる; *allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストがアップしているとみなされる

デフォルト値の表も参照してください。

例:

```
icmpping[,4] #4つのうち少なくとも1つのパケットが返された場合、アイテムは1を返します。
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 失われたパケットの割合。
 戻り値: *Float*。

パラメータ:

- **target** - ホストの IP または DNS 名
- **packets** - パケット数
- **interval** - 連続するパケット間の時間 (ミリ秒)
- **size** - パケットサイズ (バイト)

- **timeout** - タイムアウト (ミリ秒)
- **options** - リダイレクトを許可する場合に使用: 空の場合 (デフォルト値)、リダイレクトされた応答はターゲットホストがダウンしているとみなされる; *allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストが稼働しているとみなされる

デフォルト値の表も参照してください。

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

 ICMP ping によるホストの到達性をリトライ付きでチェックします。最初のパケットが成功した場合は停止し、失敗した場合はリトライします。このアイテムはネットワーク上で送信されるパケット数を減らすのに役立ちます。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメータ:

- **target** - ホストの IP または DNS 名
- **retries** - ターゲットへの ping を試行する回数（最初の試行は含まない、0 以上、デフォルト 1）
- **backoff** - 各リクエストごとに待機時間を乗算する数値（1.0-5.0 の範囲、デフォルト 1.0）
- **size** - パケットサイズ（バイト単位）
- **timeout** - タイムアウト（ミリ秒単位）
- **options** - リダイレクトを許可する場合に使用: 空（デフォルト値）の場合、リダイレクトされた応答はターゲットホストがダウンしていると見なされます。*allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストがアップしていると見なされます。

デフォルト値の表も参照してください。

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 ICMP ping の応答時間（秒単位）。
 戻り値：*Float*。

パラメータ:

- **target** - ホストの IP または DNS 名
- **packets** - パケット数
- **interval** - 連続するパケット間の時間（ミリ秒単位）
- **size** - パケットサイズ（バイト単位）
- **timeout** - タイムアウト（ミリ秒単位）
- **mode** - 可能な値：*min*、*max*、または *avg*（デフォルト）
- **options** - リダイレクトを許可する場合に使用：空（デフォルト値）の場合、リダイレクトされた応答はターゲットホストがダウンしていると見なされる。*allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストがアップしていると見なされる。

コメント:

- ロストまたはタイムアウトしたパケットは計算に使用されません
- ホストが利用できない場合（タイムアウトに達した場合）、アイテムは 0 を返します
- 戻り値が 0.0001 秒未満の場合、値は 0.0001 秒に設定されます
- デフォルト値の表も参照してください

net.tcp.service[service,<ip>,<port>]

 サービスが稼働しており、TCP 接続を受け付けているかどうかをチェックします。
 戻り値: *0* - サービスは停止しています; *1* - サービスは稼働しています。

パラメータ:

- **service** - 使用可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- *tcp* サービスの場合、ポートの指定は必須です;
- これらのチェックは、システムデーモンのログファイルに追加のメッセージを生成する場合があります (通常、SMTP や SSH セッションが記録されます);
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

`net.tcp.service[ftp,,45]` #このアイテムは、TCPポート45でFTPサーバーの可用性をテストするために使用できます。

Attention:
SELinux が強制モードで実行されている場合、カスタム TCP/UDP シンプルチェックはポリシーによってブロックされることがあります。新しい送信接続を確認して許可するには、`grep denied /var/log/audit/audit.log` で監査拒否を確認してください。

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスがダウンしている; 秒 - サービスへの接続にかかった秒数。

パラメータ:

- **service** - 使用可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- *tcp* サービスの場合、ポートの指定は必須です;
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

`net.tcp.service.perf[ssh]` #このアイテムはSSHサーバーからの初期応答速度のテストに使用できます。

net.udp.service[service,<ip>,<port>]

 サービスが稼働しており、UDP リクエストに応答しているかどうかをチェックします。
 戻り値: *0* - サービスはダウンしています; *1* - サービスは稼働しています。

パラメータ:

- **service** - 可能な値: *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

例:

`net.udp.service[ntp,,45]` #このアイテムはUDPポート45でNTPサービスの可用性をテストするために使用できます。

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスがダウンしている; 秒 - サービスからの応答を待った秒数。

パラメータ:

- **service** - 可能な値: *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

例:

`net.udp.service.perf[ntp]` #このアイテムはNTPサービスからの応答時間をテストするために使用できます。

Attention:
LDAP シンプルチェック（例：`net.tcp.service[ldap]`）で SourceIP をサポートするには、OpenLDAP バージョン 2.6.1 以上が必要です。

タイムアウト処理

柔軟なアイテムタイムアウトは、シンプルチェックでサポートされていますが、`icmpping*` および VMware アイテムには適用されません。

ICMP ping

Zabbix は、ICMP ping（**icmpping**、**icmppingloss**、**icmppingretry**、**icmppingsec**）を処理するために外部ユーティリティ**fping**を使用します。

インストール

fping は Zabbix には含まれておらず、別途インストールする必要があります。

- さまざまな Unix 系プラットフォームでは、fping パッケージがデフォルトのリポジトリにありますが、事前にインストールされていません。この場合、パッケージマネージャーを使用して fping をインストールできます。
- Zabbix は RHEL およびその派生ディストリビューション向けにfping パッケージを提供しています。これらのパッケージは公式サポートなしで提供されていることにご注意ください。
- fping はソースからコンパイルすることもできます。

設定

Zabbix サーバー/プロキシの設定ファイルの*FpingLocation*パラメータで fping の場所を指定します (または IPv6 アドレスを使用する場合は*Fping6Location*パラメータ)。

fping は Zabbix サーバー/プロキシが実行されるユーザーによって実行可能であり、このユーザーには十分な権限が必要です。

また、fping バージョン 3.10 未満でのシンプルチェック処理に関する既知の問題も参照してください。

デフォルト値

ICMP チェックパラメータのデフォルト、制限、および値の説明:

パラメータ	単位	説明	Fping のフラグ	デフォルト設定元		Zabbix による許容範囲	
				fping	**Zabbix**	**min**	**max**
packets	数値	ターゲットに送信するリクエストパケット数	-C		3	1	10000
interval	ミリ秒	個々のターゲットへの連続パケット間の待機時間	-p	1000		20	無制限
size	バイト	パケットサイズ（バイト単位） x86 では 56 バイト、x86_64 では 68 バイト	-b	56 または 68		24	65507
timeout	ミリ秒	**fping v3.x** - 最後のパケット送信後の待機タイムアウト、*-C* フラグの影響を受ける **fping v4.x** - 各パケットごとの個別タイムアウト	-t	**fping v3.x** - 500 **fping v4.x** 以降 - *-p* フラグから継承。ただし 2000 を超えない		50	無制限
retries	数値	ターゲットへの ping 試行回数（最初の試行は含まない）	-r	3	1	0	無制限
backoff factor	数値	各リクエストごとに待機時間を乗算する値	-B	1.5	1.0	1.0	5.0

デフォルト値は、プラットフォームやバージョンによって若干異なる場合があります。

さらに、Zabbix は fping オプション*-i interval ms*（上記の表で説明したアイテムパラメータ *interval* と混同しないこと。これは fping オプション*-p* に対応）および*-S source IP address*（古い fping バージョンでは*-I*）を使用します。これらのオプションは、異なるオプションの組み合わせでチェックを実行することで自動検出されます。Zabbix は、*-i* で fping が使用を許可する最小値（ミリ秒単位）を、0、1、10 の 3 つの値で試行して検出します。最初に成功した値が、その後の ICMP チェックで使用されます。このプロセスは、各ICMP pingerプロセスごとに個別に行われます。

自動検出された fping オプションは 1 時間ごとに無効化され、次回の ICMP チェック実行時に再検出されます。このプロセスの詳細をサーバーまたはプロキシのログファイルで確認するには、DebugLevel>=4 を設定してください。

Zabbix は、*icmpping** キーのいずれかでチェックされる IP アドレスを一時ファイルに書き込み、そのファイルを fping に渡します。アイテムのキーのパラメータが異なる場合は、同一のキーのパラメータを持つもののみが 1 つのファイルに書き込まれます。1 つのファイルに書き込まれたすべての IP アドレスは、fping によって並列にチェックされるため、Zabbix ICMP pinger プロセスは、ファイル内の IP アドレス数に関係なく一定時間で処理を完了します。

1 VMware 監視アイテムキー

VMware 監視のitem キーのリストは、VMware 監視セクションに移動されました。

6 ログファイル監視

概要

Zabbix は、ログローテーションの有無にかかわらず、ログファイルの集中監視および分析に使用できます。

通知機能を利用して、ログファイルに特定の文字列やパターンが含まれている場合にユーザーに警告することができます。

ログファイルを監視するには、以下が必要です。

- 監視対象ホストで Zabbix エージェントが稼働していること
- ログ監視アイテムが設定されていること

Attention:
監視対象のログファイルのサイズ制限は、大容量ファイルサポートに依存します。

設定

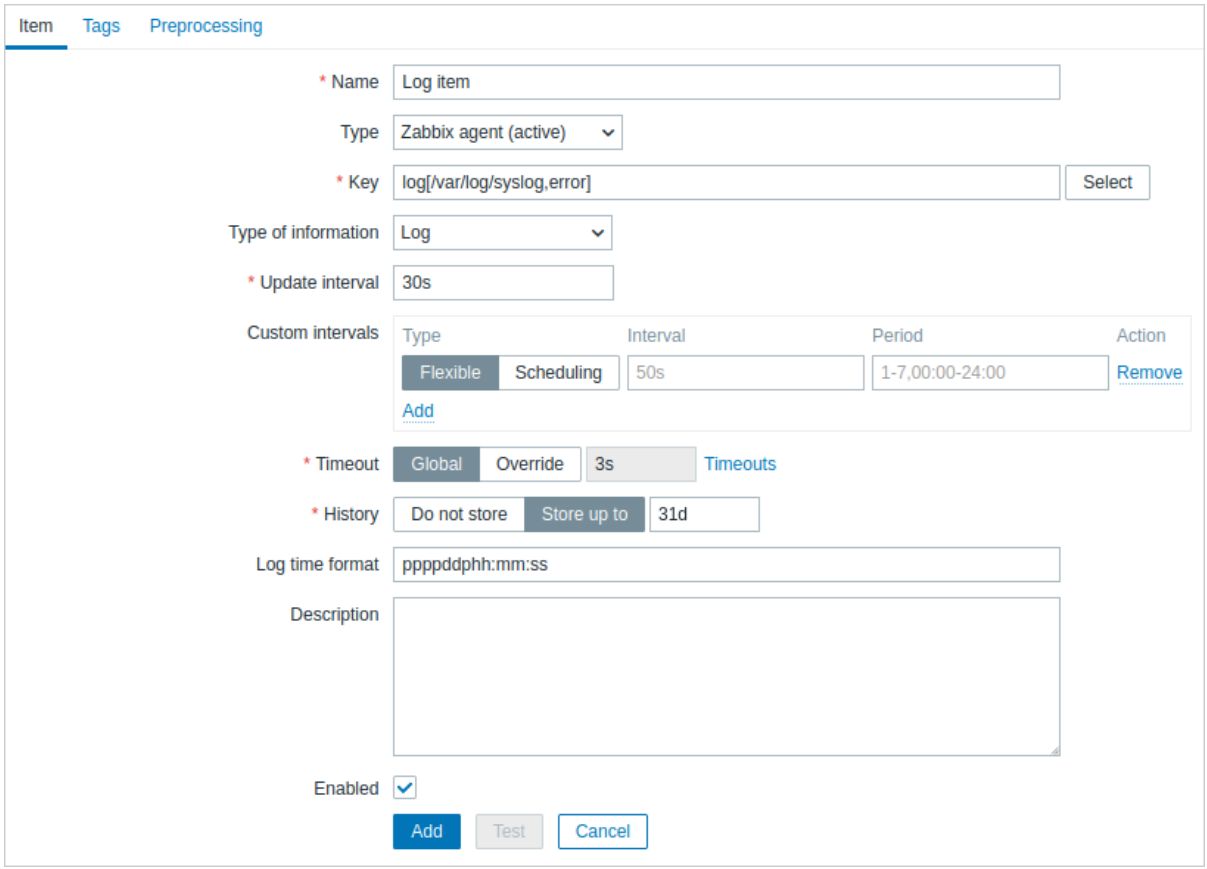
エージェントパラメータの確認

エージェントの設定ファイルで、以下を確認してください。

- `Hostname` パラメータがフロントエンドのホスト名と一致していること。
- `ServerActive` パラメータにアクティブチェックを処理するサーバーが指定されていること。

アイテムの設定

ログ監視用のアイテムを設定します。



必須入力フィールドには赤いアスタリスクが付いています。

特にログ監視アイテムの場合は、以下を入力します。

タイプ	ここでは **Zabbix** エージェント（アクティブ）を選択します。

キー	次のいずれかのアイテムキーを使用します。 **log[]** または **logrt[]**: これら 2 つのアイテムキーは、ログを監視し、存在する場合は内容の正規表現でログエントリをフィルタリングできます。 例: `log[/var/log/syslog,error]`。ファイルに「zabbix」ユーザーの読み取り権限があることを確認してください。そうでない場合、アイテムのステータスは「サポートされていません」に設定されます。 **log.count[]** または **logrt.count[]**: これら 2 つのアイテムキーは、一致する行数のみを返します。 これらのアイテムキーとそのパラメータの使用方法については、サポートされているZabbix エージェントアイテムキーのセクションを参照してください。
情報のタイプ	自動的に入力されます。 log[] または logrt[] アイテムの場合 - ログ; log.count[] または logrt.count[] アイテムの場合 - 数値（整数）。 オプションで `output` パラメータを使用する場合は、「ログ」以外の適切な情報のタイプを手動で選択できます。 「ログ」以外の情報のタイプを選択すると、ローカルタイムスタンプが失われることに注意してください。
更新間隔（秒）	このパラメータは、Zabbix エージェントがログファイルの変更をどのくらいの頻度でチェックするかを定義します。1 秒に設定すると、新しいレコードをできるだけ早く取得できます。
ログ時刻フォーマット	このフィールドでは、オプションでログ行のタイムスタンプを解析するためのパターンを指定できます。サポートされているプレースホルダー: * **y**: 年 *(1970-2038)* * **M**: 月 *(01-12)* * **d**: 日 *(01-31)* * **h**: 時 *(00-23)* * **m**: 分 *(00-59)* * **s**: 秒 *(00-59)* 空白のままにすると、タイムスタンプは Unix 時間で 0（1970 年 1 月 1 日）に設定されます。 例えば、Zabbix エージェントのログファイルから次の行を考えてみましょう。 ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” この行は、PID の 6 文字分、日付、時刻、メッセージの残りの部分で始まります。 この行のログ時刻フォーマットは「pppppp:yyyyMMdd:hhmmss」となります。 「p」と「:」の文字はプレースホルダーであり、「yMdhms」以外の任意の文字にできます。

重要な注意事項

- サーバーとエージェントは、監視対象のログのサイズと最終更新時刻（logrt の場合）を 2 つのカウンターで管理します。さらに：
  - エージェントは内部的に、ログファイルが切り詰められたりローテーションされた場合の判断を改善するために、inode 番号（UNIX/GNU/Linux の場合）、ファイルインデックス（Microsoft Windows の場合）、および最初の 512 バイトの MD5 サムを使用します。
  - UNIX/GNU/Linux システムでは、ログファイルが保存されているファイルシステムが inode 番号を報告し、それを使ってファイルを追跡できることが前提となっています。
  - Microsoft Windows では、Zabbix エージェントはログファイルが存在するファイルシステムの種類を判別し、以下を使用します：
    - NTFS ファイルシステムでは 64 ビットのファイルインデックス。
    - ReFS ファイルシステム（Microsoft Windows Server 2012 以降のみ）では 128 ビットのファイル ID。
    - ファイルインデックスが変更されるファイルシステム（例：FAT32、exFAT）では、ログファイルのローテーションにより同じ最終更新時刻を持つ複数のログファイルが発生した場合、不確実な状況下で合理的なアプローチを取るフォールバックアルゴリズムが使用されます。
  - inode 番号、ファイルインデックス、MD5 サムは Zabbix エージェントによって内部的に収集されます。これらは Zabbix サーバーには送信されず、Zabbix エージェントが停止すると失われます。
  - ログファイルの最終更新時刻を変更しないでください（例：`touch` コマンドなど）。また、監視対象のログファイルを元の名前にファイルをコピーして置き換えないでください（これにより新しい inode が作成されます）。いずれの場合も、Zabbix はファイルを別のファイルとして扱い、最初から再読込する可能性があり、重複したアラートが発生することがあります。
  - `logrt[]` アイテムに複数の一致するログファイルがあり、Zabbix エージェントがその中で最新のものを追跡していて、その最新のログファイルが削除された場合、警告メッセージ`"there are no files matching "<regexp mask>" in "<directory>"` が記録されます。Zabbix エージェントは、`logrt[]` アイテムでエージェントが確認した最新の最終更新時刻よりも古い更新時刻のログファイルは無視します。
- エージェントは、前回停止したポイントからログファイルの読み取りを開始します。

- すでに解析されたバイト数（サイズカウンター）と最終更新時刻（時刻カウンター）は Zabbix データベースに保存され、エージェントに送信されます。これにより、エージェントが再起動された場合や、以前は無効または未サポートだったアイテムを受信した場合でも、エージェントがこのポイントからログファイルの読み取りを開始できるようになります。ただし、エージェントがサーバーから 0 以外のサイズカウンターを受信したが、logrt[] または logrt.count[] アイテムが一致するファイルを見つけられない場合、サイズカウンターは 0 にリセットされ、ファイルが後で現れた場合は最初から解析されます。
- ログファイルがエージェントが認識しているログサイズカウンターよりも小さくなった場合、カウンターはゼロにリセットされ、エージェントは時刻カウンターを考慮してログファイルの先頭から読み取りを開始します。
- ディレクトリ内に同じ最終更新時刻の一致するファイルが複数ある場合、エージェントは同じ更新時刻のすべてのログファイルを正しく解析し、データのスキップや重複解析を避けるようにしますが、すべての状況で保証されるわけではありません。エージェントは特定のログファイルローテーション方式を前提とせず、またそれを判別しません。同じ最終更新時刻の複数のログファイルがある場合、エージェントはそれらを辞書順で降順に処理します。したがって、ローテーション方式によっては、ログファイルは元の順序で解析・報告されます。他のローテーション方式では元のログファイルの順序が維持されず、一致したログファイルレコードが順序を変更して報告されることがあります（ログファイルの最終更新時刻が異なる場合はこの問題は発生しません）。
- Zabbix エージェントは、更新間隔秒ごとにログファイルの新しいレコードを処理します。
- Zabbix エージェントは、1 秒あたり **maxlines** を超えるログファイルのレコードを送信しません。この制限はネットワークや CPU リソースの過負荷を防ぎ、エージェント設定ファイルの **MaxLinesPerSecond** パラメータで指定されたデフォルト値を上書きします。
- 必要な文字列を見つけるために、Zabbix は MaxLinesPerSecond で設定された値の 10 倍の新しい行を処理します。したがって、例えば `log[]` または `logrt[]` アイテムの更新間隔が 1 秒の場合、デフォルトでエージェントは最大 200 件のログファイルレコードを解析し、1 回のチェックで最大 20 件の一致するレコードを Zabbix サーバーに送信します。エージェント設定ファイルの **MaxLinesPerSecond** を増やすか、アイテムキーの **maxlines** パラメータを設定することで、1 回のチェックで最大 10000 件の解析済みログファイルレコードと 1000 件の一致レコードを Zabbix サーバーに送信するように制限を引き上げることができます。更新間隔が 2 秒に設定されている場合、1 回のチェックの制限は更新間隔が 1 秒の場合の 2 倍になります。
- さらに、log および log.count の値は、非ログ値がバッファに存在しない場合でも、常にエージェント送信バッファサイズの 50% に制限されます。したがって、**maxlines** の値を 1 回の接続で送信する（複数回の接続ではなく）には、エージェントのBufferSizeパラメータを maxlines x 2 以上にする必要があります。Zabbix エージェントはログ収集中にデータをアップロードしてバッファを解放できますが、Zabbix agent 2 はデータがアップロードされバッファが解放されるまでログ収集を停止します（これは非同期で実行されます）。
- ログアイテムがない場合、すべてのエージェントバッファサイズは非ログ値に使用されます。ログ値が入ってくると、必要に応じて古い非ログ値を置き換え、指定された 50% まで使用します。
- 256kB を超えるログファイルレコードの場合、最初の 256kB のみが正規表現と照合され、残りのレコードは無視されます。ただし、Zabbix エージェントが長いレコードを処理中に停止した場合、エージェントの内部状態が失われ、再起動後に長いレコードが再度、かつ異なる方法で解析される可能性があります。
- "\" パスセパレータに関する特記事項：file_format が"file\.log" の場合、"file" ディレクトリを作成しないでください。なぜなら、"." がエスケープされているのか、ファイル名の最初の文字なのかを明確に定義できないためです。
- `logrt` の正規表現はファイル名のみでサポートされており、ディレクトリの正規表現マッチングはサポートされていません。
- UNIX プラットフォームでは、ログファイルが見つかるはずのディレクトリが存在しない場合、`logrt[]` アイテムは NOTSUPPORTED になります。
- Microsoft Windows では、ディレクトリが存在しない場合でもアイテムは NOTSUPPORTED になりません（例：アイテムキーでディレクトリ名を誤記した場合など）。
- `logrt[]` アイテムにログファイルが存在しない場合でも NOTSUPPORTED にはなりません。`logrt[]` アイテムのログファイル読み取りエラーは、Zabbix エージェントのログファイルに警告として記録されますが、アイテムを NOTSUPPORTED にはしません。
- `log[]` または `logrt[]` アイテムが NOTSUPPORTED になった理由を調べるには、Zabbix エージェントのログファイルが役立ちます。Zabbix は、DebugLevel=4 または DebugLevel=5 の場合を除き、エージェントのログファイル自体も監視できます。
- 正規表現を使って疑問符を検索する場合（例：\?）、テキストファイルに NUL 記号が含まれていると誤検出が発生する可能性があります。これは、Zabbix が NUL 記号を"?" に置き換えて、改行文字まで行の処理を継続するためです。

正規表現の一致部分の抽出

正規表現の一致が見つかった場合に、対象ファイルの全行を返すのではなく、興味のある値だけを抽出したい場合があります。

ログアイテムには、一致した行から目的の値を抽出する機能があります。これは、`log` および `logrt` アイテムの追加パラメータ **output** によって実現されます。

'output' パラメータを使用すると、関心のある一致の「キャプチャグループ」を指定できます。

たとえば、次のようにします。

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

これは、次の内容に見られるエントリ数を返すことができます。

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

\1 は最初で唯一のキャプチャグループ **([0-9]+)** を参照しているため、数値だけが返されます。

また、数値を抽出して返すことができるため、その値を使用してトリガーを定義できます。

maxdelay パラメータの使用

ログアイテムの `maxdelay` パラメータを使用すると、`maxdelay` 秒以内に最新の行が分析されるように、ログファイルの古い行を無視することができます。

> **Warning:**
> 'maxdelay' > 0 を指定すると、重要なログファイルの記録が無視され、アラートが見逃される可能性があります。必要な場合のみ、自己責任で慎重に使用してください。

デフォルトでは、ログ監視用のアイテムはログファイルに現れるすべての新しい行を追跡します。しかし、アプリケーションによっては、特定の状況下でログファイルに大量のメッセージを書き込み始めるものがあります。たとえば、データベースや DNS サーバーが利用できない場合、そのようなアプリケーションは通常の動作が回復するまで、ほぼ同一のエラーメッセージでログファイルを埋め尽くします。デフォルトでは、これらすべてのメッセージが忠実に分析され、`log` および `logrt` アイテムで設定された条件に一致する行がサーバーに送信されます。

過負荷に対する組み込みの保護には、設定可能な `maxlines` パラメータ（サーバーが受信する一致するログ行の数を制限）と 10*'maxlines' の制限（1 回のチェックでエージェントによるホストの CPU と I/O の過負荷を防止）があります。それでも、組み込みの保護には 2 つの問題があります。第一に、あまり有益でない可能性のある大量のメッセージがサーバーに報告され、データベースのスペースを消費します。第二に、1 秒あたりに分析される行数が制限されているため、エージェントが最新のログ記録に何時間も遅れる可能性があります。おそらく、何時間も古い記録を追いかけるよりも、ログファイルの現在の状況を早く知りたいと思うでしょう。

この 2 つの問題の解決策が `maxdelay` パラメータの使用です。`maxdelay` > 0 が指定されている場合、各チェック時に処理されたバイト数、残りのバイト数、および処理時間が測定されます。これらの数値から、エージェントはログファイル内のすべての残りの記録を分析するのに何秒かかるかという推定遅延を計算します。

遅延が `maxdelay` を超えない場合、エージェントは通常どおりログファイルの分析を続行します。

遅延が `maxdelay` を超える場合、エージェントはログファイルの一部を「ジャンプ」して無視し、残りの行が `maxdelay` 秒以内に分析できるように新しい推定位置に移動します。

エージェントは無視された行をバッファに読み込むことすらせず、ファイル内でジャンプするおおよその位置を計算することに注意してください。

ログファイルの行をスキップした事実は、エージェントのログファイルに次のように記録されます。

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]"
logfile:"/home/zabbix32/test1.log" skipping 679858 bytes
(from byte 75653115 to byte 76332973) to meet maxdelay
```

"to byte" の数値はおおよそであり、「ジャンプ」後にエージェントがファイル内のログ行の先頭に位置を調整するため、さらに先または前になる場合があります。

ログファイルの成長速度と分析速度の比較によって、「ジャンプ」が発生しない場合、まれにまたは頻繁に「ジャンプ」する場合、大きなまたは小さな「ジャンプ」になる場合、あるいはすべてのチェックで小さな「ジャンプ」になる場合があります。システム負荷やネットワーク遅延の変動も遅延の計算に影響し、そのため「maxdelay」パラメータに追いつくために先に「ジャンプ」することがあります。

`maxdelay` < 更新間隔に設定することは推奨されません（頻繁に小さな「ジャンプ」が発生する可能性があります）。

'copytruncate' ログファイルローテーションの取り扱いに関する注意事項

`logrt` に `copytruncate` オプションを指定した場合、異なるログファイルには異なるレコード（少なくともタイムスタンプが異なる）が含まれていると仮定します。そのため、初期ブロック（最初の 512 バイトまで）の MD5 サムも異なります。初期ブロックの MD5 サムが同じ 2 つのファイルがある場合、一方がオリジナルで、もう一方がコピーであることを意味します。

`logrt` に `copytruncate` オプションを指定した場合、重複を報告せずにログファイルのコピーを正しく処理するように努めます。ただし、同じタイムスタンプで複数のログファイルコピーを作成したり、logrt[] アイテムの更新間隔よりも頻繁にログファイルをローテーションしたり、エージェントを頻繁に再起動したりすることは推奨されません。エージェントはこれらの状況すべてを適切に処理しようとしますが、すべての状況で良好な結果が保証されるわけではありません。

log*[] item の persistent files に関する注意事項

永続ファイルの目的

Zabbix エージェントが起動されると、Zabbix サーバーまたはプロキシからアクティブチェックのリストを受信します。log*[] メトリックの場合、処理済みのログサイズと変更時刻を受信し、どこからログファイルの監視を開始するかを判断します。ファイルシステムから報告された実際のログファイルサイズと変更時刻に応じて、エージェントは処理済みのログサイズからログファイルの監視を継続するか、ログファイルの先頭から再解析するかを決定します。

実行中のエージェントは、チェック間で監視されているすべてのログファイルを追跡するための属性のより大きなセットを保持します。このメモリ内の状態は、エージェントが停止すると失われます。

新しいオプションパラメータ **persistent_dir** は、log[]、log.count[]、logrt[]、または logrt.count[] アイテムの状態をファイルに保存するディレクトリを指定します。Zabbix エージェントが再起動された後、ログアイテムの状態は永続ファイルから復元されます。

主なユースケースは、ミラー化されたファイルシステム上にあるログファイルの監視です。ある時点までは、ログファイルは両方のミラーに書き込まれます。その後、ミラーが分割されます。アクティブなコピーでは、ログファイルは新しいレコードを取得しながら成長し続けます。Zabbix エージェントはそれを分析し、処理済みのログサイズと変更時刻をサーバーに送信します。パッシブコピーでは、ログファイルは同じままで、アクティブコピーよりもかなり遅れています。後で、オペレーティングシステムと Zabbix エージェントがパッシブコピーから再起動されます。Zabbix エージェントがサーバーから受信する処理済みのログサイズと変更時刻は、パッシブコピーの状況には有効でない場合があります。ファイルシステムのミラー分割時点でエージェントが中断した場所からログファイルの監視を継続するために、エージェントは永続ファイルから状態を復元します。

永続ファイルを使用したエージェントの動作

起動時、Zabbix エージェントは永続ファイルについて何も知りません。Zabbix サーバー（プロキシ）からアクティブチェックのリストを受信した後にのみ、エージェントは一部のログアイテムが指定されたディレクトリの下に永続ファイルでバックアップされるべきであることを認識します。

エージェントの動作中、永続ファイルは書き込み用にオープンされ（fopen(filename, ”w”)）、最新のデータで上書きされます。上書きとファイルシステムのミラー分割が同時に発生した場合に永続ファイルのデータが失われる可能性は非常に低く、特別な処理は行いません。永続ファイルへの書き込み後にストレージメディアへの強制同期（fsync() の呼び出し）は行いません。

最新データでの上書きは、該当するログファイルレコードまたはメタデータ（処理済みログサイズおよび変更時刻）が Zabbix サーバーに正常に報告された後に行われます。これは、ログファイルが変更され続ける場合、アイテムチェックごとに発生する可能性があります。

エージェントのシャットダウン時に特別な処理はありません。

アクティブチェックのリストを受信した後、エージェントは不要になった永続ファイルを削除対象としてマークします。永続ファイルが不要になるのは、1) 対応するログアイテムが監視されなくなった場合、2) ログアイテムが以前とは異なる **persistent_dir** の場所で再設定された場合です。

削除は 24 時間の遅延で行われます。なぜなら、NOTSUPPORTED 状態のログファイルはアクティブチェックのリストに含まれませんが、後で SUPPORTED になる可能性があり、その場合永続ファイルが役立つためです。

エージェントが 24 時間経過する前に停止された場合、エージェントは Zabbix サーバーから永続ファイルの場所に関する情報を取得できなくなるため、不要なファイルは削除されません。

> **Warning:**
> エージェントが停止している間に、ユーザーが古い永続ファイルを削除せずにログアイテムの **persistent_dir** を元の **persistent_dir** の場所に再設定すると、古い永続ファイルからエージェントの状態が復元され、メッセージの取りこぼしや誤ったアラートが発生する可能性があります。

永続ファイルの命名と場所

Zabbix エージェントは、キーによってアクティブチェックを区別します。たとえば、logrt[/home/zabbix/test.log] と logrt[/home/zabbix/test.log,] は異なるアイテムです。フロントエンドでアイテム logrt[/home/zabbix/test.log„,10] を logrt[/home/zabbix/test.log„,20] に変更すると、エージェントのアクティブチェックリストから logrt[/home/zabbix/test.log„,10] アイテムが削除され、logrt[/home/zabbix/test.log„,20] アイテムが作成されます（いくつかの属性はフロントエンド/サーバーでの変更時に引き継がれますが、エージェントでは引き継がれません）。

ファイル名は、アイテムキーの MD5 サムにアイテムキーの長さを付加して構成され、衝突の可能性を減らします。たとえば、logrt[/home/zabbix50/test.log„„„„/home/zabbix50/agent_private] アイテムの状態は、永続ファイル c963ade4008054813bbc0a650bb8e09266 に保持されます。

複数のログアイテムで同じ **persistent_dir** 値を使用できます。

persistent_dir は、特定のファイルシステムレイアウト、マウントポイントとマウントオプション、ストレージミラーリング構成を考慮して指定します。永続ファイルは、監視対象のログファイルと同じミラーリングされたファイルシステム上にある必要があります。

persistent_dir ディレクトリが作成できない、または存在しない、または Zabbix エージェントのアクセス権がファイルの作成/書き込み/読み取り/削除を許可しない場合、ログアイテムは NOTSUPPORTED になります。

エージェントの動作中に永続ストレージファイルへのアクセス権が削除された場合や、その他のエラー（例：ディスクの空き容量不足）が発生した場合、エラーはエージェントのログファイルに記録されますが、ログアイテムは NOTSUPPORTED にはなりません。

I/O の負荷

アイテムの永続ファイルは、サーバーへの各データバッチ（アイテムのデータを含む）の送信が成功した後に更新されます。例えば、デフォルトの BufferSize は 100 です。ログアイテムが 70 件の一致するレコードを見つけた場合、最初の 50 件のレコードは 1 つのバッチで送信され、永続ファイルが更新され、残りの 20 件のレコードは（より多くのデータが蓄積されたときに遅延して）2 番目のバッチで送信され、永続ファイルが再度更新されます。

エージェントとサーバー間の通信が失敗した場合の動作

log[] および logrt[] アイテムの一致した各行、および各 log.count[] および logrt.count[] アイテムチェックの結果は、エージェント送信バッファの指定された 50% 領域内の空きスロットを必要とします。バッファ要素は定期的にサーバー（またはプロキシ）に送信され、バッファスロットは再び空きます。

エージェント送信バッファの指定されたログ領域に空きスロットがあり、エージェントとサーバー（またはプロキシ）間の通信が失敗している間は、ログ監視結果は送信バッファに蓄積されます。これにより、短期間の通信障害を緩和できます。

長期間の通信障害が発生すると、すべてのログスロットが占有され、次のアクションが実行されます。

- log[] および logrt[] アイテムチェックが停止されます。通信が回復し、バッファに空きスロットができると、チェックは前回の位置から再開されます。一致した行は失われず、後で報告されるだけです。
- log.count[] および logrt.count[] チェックは、maxdelay = 0（デフォルト）の場合に停止されます。動作は上記の log[] および logrt[] アイテムと同様です。このことは log.count[] および logrt.count[] の結果に影響を与える可能性があることに注意してください。たとえば、あるチェックでログファイル内の一致した行が 100 件カウントされますが、バッファに空きスロットがないためチェックが停止されます。通信が回復すると、エージェントは同じ 100 件の一致した行と新たに一致した 70 件の行をカウントします。エージェントは、1 回のチェックで見つかったかのように count = 170 を送信します。
- maxdelay > 0 の log.count[] および logrt.count[] チェック：チェック中に「ジャンプ」がなかった場合、動作は上記と同様です。ログファイル行の「ジャンプ」が発生した場合は、「ジャンプ」後の位置が保持され、カウントされた結果は破棄されます。したがって、エージェントは通信障害が発生しても増加するログファイルに追従しようとします。

正規表現のコンパイルエラーおよび実行時エラーの処理

log[]、logrt[]、log.count[]、logrt.count[] アイテムで使用されている正規表現が PCRE または PCRE2 ライブラリでコンパイルできない場合、アイテムは NOTSUPPORTED 状態となり、エラーメッセージが表示されます。ログアイテムの監視を継続するには、正規表現を修正する必要があります。

正規表現が正常にコンパイルされても、実行時に失敗した場合（いくつかまたはすべてのログレコードで）、ログアイテムはサポートされたままで監視が継続されます。実行時エラーは、Zabbix エージェントのログファイルに記録されます（ログファイルのレコードは含まれません）。

Zabbix エージェントが自身のログファイルを監視できるように、ログの記録頻度は 1 チェックにつき 1 回の実行時エラーに制限されています。たとえば、10 件のレコードを分析し、3 件のレコードで正規表現の実行時エラーが発生した場合、エージェントログには 1 件のレコードのみが出力されます。

例外：MaxLinesPerSecond=1 かつ更新間隔 =1（1 チェックにつき 1 レコードのみ分析可能）の場合、正規表現の実行時エラーは記録されません。

zabbix_agentd は実行時エラーが発生した場合にアイテムキーをログに記録し、zabbix_agent2 はアイテム ID を記録して、どのログアイテムで実行時エラーが発生したかを特定できるようにします。実行時エラーが発生した場合は、正規表現の設計を見直すことを推奨します。

7 計算

概要

計算アイテムを使用すると、既存のアイテムの値に基づいて計算を作成できます。たとえば、あるアイテム値の 1 時間ごとの平均を計算したり、アイテムグループの合計値を計算したりすることができます。これが計算アイテムの目的です。

計算では、以下の両方を使用できます。

- 個々のアイテムの単一値
- 集計のために複数のアイテムを選択するための複雑なフィルター（詳細は集計計算を参照）

計算アイテムは、仮想データソースを作成する方法です。すべての計算は Zabbix サーバーのみで実行されます。値は、使用される算術式に基づいて定期的に計算されます。

結果のデータは、他のアイテムと同様に Zabbix データベースに保存されます。履歴値とトレンド値の両方が保存され、グラフを生成できます。

Note:
計算結果が浮動小数点値の場合、計算アイテムの情報の型が数値（整数）の場合は整数に切り捨てられます。
また、キャッシュに最近のデータがなく、関数で問い合わせ期間が定義されていない場合、Zabbix はデフォルトで過去 1 週間までさかのぼってデータベースから履歴値を取得します。

計算アイテムは、トリガー式と構文を共有します。計算アイテムでは文字列との比較も許可されています。計算アイテムは、他のアイテムタイプと同様にマクロや他のエンティティから参照できます。

計算アイテムを使用するには、アイテムタイプ計算を選択します。

設定可能なフィールド

キーは、ホストごとに一意のアイテム識別子です。サポートされている記号を使用して任意のキー名を作成できます。

計算式は式フィールドに入力します。式とキーの間に関連性はありません。キーのパラメータは式内で一切使用されません。

単純な式の構文は次のとおりです:

`function(/host/key,<parameter1>,<parameter2>,...)`

ここで:

function	サポートされている関数のいずれか: last, min, max, avg, count など
host	計算に使用されるアイテムのホスト。 現在のホストは省略可能です（例: `function(//key,parameter,...)` のように）。
key	計算に使用されるアイテムのキー。
parameter(s)	関数のパラメータ（必要な場合）。 時間のサフィックスおよびメモリサイズのサフィックスがサポートされています。

Attention:
式内のユーザーマクロは、関数パラメータ、アイテムフィルタパラメータ、または定数を参照する場合に展開されます。関数名、ホスト名、アイテムキー、アイテムキーパラメータ、または演算子を参照する場合は展開されません。

より複雑な式では、関数、演算子、括弧を組み合わせて使用できます。トリガー式でサポートされているすべての関数と演算子を使用できます。論理や演算子の優先順位もまったく同じです。

トリガー式とは異なり、Zabbix は新しい値を受信したときではなく、アイテムの更新間隔に従って計算アイテムを処理します。

計算アイテムの式で履歴関数によって参照されるすべてのアイテムは、存在し、データを収集している必要があります。また、参照されるアイテムのアイテムキーを変更した場合は、そのキーを使用しているすべての式を手動で更新する必要があります。

計算アイテムがサポート対象外になる場合はいくつかあります:

- 参照アイテムが
  - 見つからない
  - 無効になっている
  - 無効なホストに属している
  - サポート対象外である（ただし、nodata() 関数および未知の値を持つ演算子を除く）
- 関数を計算するためのデータがない
- ゼロによる除算
- 不正な構文の使用

使用例

例 1

'/' のディスク空き容量のパーセンテージを計算します。

Use of function **last**:

`100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])`

Zabbix はフリーディスクスペースとトータルディスクスペースの最新値を取得し、指定された計算式に従ってパーセンテージを計算します。

例 2

Zabbix で処理された値の 10 分間の平均値を計算します。

Use of function **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

長い時間の計算 item を多用すると、Zabbix server のパフォーマンスに影響を与える可能性があることに注意してください。

例 3

eth0 の総帯域幅を計算する。

2 つの関数の合計:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

例 4

受信トラフィックのパーセンテージを計算する。

より複雑な表現:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Examples of aggregate calculationsを参照してください。

1 集計計算

概要

集計計算は、Zabbix サーバーによって複数のアイテムから情報を収集し、使用する集計関数に応じて集計を計算できる計算アイテムタイプです。

集計計算では、監視対象のホストでエージェントを実行する必要はありません。

文法

集計を取得するには、サポートされている集計関数のいずれかを使用します: avg、max、min、sum など。次に **foreach** 関数を唯一のパラメーターとして使用し、そのアイテムフィルターを使用して必要なアイテムを選択します

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

foreach 関数 (例: *avg_foreach*、*count_foreach* など) は、選択されたアイテムごとに 1 つの集計値を返します。アイテムは、アイテムフィルター (`/host/key?[group="host group"]`) を使用してアイテムヒストリから選択されます。詳細については、foreach関数を参照してください。

一部のアイテムに要求された期間のデータがない場合、計算では無視されます。どのアイテムにもデータがない場合、関数はエラーを返します。

または、集計のパラメーターとして複数のアイテムをリストすることもできます:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

ここで `function` はヒストリ/トレンド関数でなければならないことに注意してください。

> **Note:**
> 集計の結果が浮動小数点値になった場合、集計されたアイテムの情報の種類が数値 (符号なし) であれば、整数に切り詰められます。

ユーザーマクロとローレベルディスカバリマクロは、次の場合にサポートされます:

- アイテムキーパラメーター
- 関数パラメーター
- フィルター条件 (ホストグループ名とタグ名)
- 式定数

集計計算は、次の場合にサポートされなくなる可能性があります:

- 参照されたアイテムが見つからない (アイテムキーが正しくない、アイテムが存在しない、含まれているすべてのグループが正しくない、といった場合に発生する可能性があります)
- 関数を計算するためのデータがない

使用例

集計のためのキーの例です。

例 1

ホストグループ'MySQL Servers' の総ディスク容量

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

例 2

そのホストで net.if.in[*] に一致するすべての item の最新値の合計

```
sum(last_foreach(/host/net.if.in[*]))
```

例 3

ホストグループ'MySQL Servers' の平均プロセッサ負荷

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

例 4

ホストグループ'MySQL Servers' の 1 秒あたりのクエリ数の 5 分平均値

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

例 5

特定のタグを持つ複数のホストグループに属する全ホストの CPU 負荷の平均値

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

例 6

ホストグループ全体の最新の item 値合計で計算します

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

例 7

ホストグループ'Zabbix servers' でサポートされていないアイテムの総数。

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

正しい/間違った構文の例

式 (関数呼び出しを含む) は、ヒストリー、トレンド、または foreach 関数 パラメーターとして使用できません。ただし、これらの関数自体は、他の (非ヒストリ) 関数パラメーターで使用できます。

式	例
有効	`avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)` `max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))`

I 無効	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

次のような表現に注意してください。

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

式の両方の部分が常に同じ値のセットになるとは限りません。式の一部が評価されている間に、要求された期間の新しい値が到着し、式の他の部分が異なる値のセットを持つことがあります。

8 内部チェック

概要

内部チェックは、Zabbix の内部プロセスを監視することを可能にします。言い換えれば、Zabbix サーバーや Zabbix プロキシで何が起こっているかを監視できます。

内部チェックは以下のように計算されます:

- サーバーによって監視されているホストの場合は Zabbix サーバーで
- プロキシによって監視されているホストの場合は Zabbix プロキシで

内部チェックは、ホストのメンテナンス状態に関係なく、サーバーまたはプロキシによって処理されます。

このアイテムを使用するには、**Zabbix** 内部アイテムタイプを選択します。

> **Note:**
> 内部チェックは Zabbix ポーラーによって処理されます。

パフォーマンス

一部の内部アイテムはパフォーマンスに悪影響を及ぼす可能性があります。これらのアイテムは以下の通りです:

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue,,]
- zabbix[requiredperformance]
- zabbix[stats,,,queue,,]
- zabbix[triggers]

システム情報およびキューのフロントエンドセクションも影響を受けます。

サポートされているチェック

アイテムキーはカスタマイズ可能なパラメータや追加情報なしでリストされています。アイテムキーをクリックすると、詳細が表示されます。

アイテムキー	説明
zabbix[boottime]	Zabbix サーバーまたは Zabbix プロキシプロセスの起動時間（秒単位）。
zabbix[cluster,discovery,nodes]	有効な高可用性クラスタノードを検出します。
zabbix[connector_queue]	コネクタキューにエンキューされた値の数。
zabbix[discovery_queue]	ディスカバリキューにエンキューされたネットワークチェックの数。
zabbix[host,,items]	ホスト上で有効なアイテム（サポートされているものとされていないもの）の数。
zabbix[host,,items_unsupported]	ホスト上で有効なサポートされていないアイテムの数。
zabbix[host,maintenance]	ホストの現在のメンテナンスステータス。
zabbix[host,active_agent,available]	ホスト上のアクティブエージェントチェックの可用性。
zabbix[host,discovery,interfaces]	Zabbix フロントエンドで設定されたホストのすべてのインターフェースの詳細。
zabbix[host,,available]	ホスト上の特定のタイプのチェックのメインインターフェースの可用性。
zabbix[hosts]	監視対象ホストの数。
zabbix[items]	有効なアイテム（サポートされているものとされていないもの）の数。
zabbix[items_unsupported]	サポートされていないアイテムの数。
zabbix[java,,]	Zabbix Java ゲートウェイに関する情報。
zabbix[lld_queue]	低レベルディスカバリ処理キューにエンキューされた値の数。
zabbix[preprocessing]	プリプロセッシングマネージャーが受信した値の統計。
zabbix[preprocessing_queue]	プリプロセッシングキューにエンキューされた値の数。
zabbix[process,,,]	特定の Zabbix プロセスまたはプロセスグループ（<type> と <mode> で識別）が <state> で費やした時間の割合。
zabbix[proxy,,]	Zabbix プロキシに関する情報。
zabbix[proxy,discovery]	Zabbix プロキシのリスト。
zabbix[proxy group,,available]	プロキシグループ内のオンラインプロキシの数。
zabbix[proxy group,,pavailable]	プロキシグループ内のオンラインプロキシの割合。
zabbix[proxy group,,proxies]	プロキシグループ内の Zabbix プロキシのリスト。
zabbix[proxy group,,state]	プロキシグループの状態。
zabbix[proxy group,discovery]	設定データとリアルタイムデータを持つプロキシグループのリストを返します。
zabbix[proxy_buffer,buffer,]	プロキシメモリバッファの使用状況統計を返します。
zabbix[proxy_buffer,state,changes]	起動以降のディスク/メモリバッファモード間の状態変化の回数を返します。
zabbix[proxy_buffer,state,current]	新しいデータが保存されている現在の動作状態を返します。
zabbix[proxy_history]	サーバーに送信待ちのプロキシ履歴テーブル内の値の数。
zabbix[queue,,]	<from> 秒以上、<to> 秒未満遅延しているキュー内の監視アイテムの数。

アイテムキー	説明
zabbix[rcache,]	Zabbix 設定キャッシュの可用性統計。
zabbix[requiredperformance]	Zabbix サーバーまたは Zabbix プロキシの必要なパフォーマンス（1 秒あたりの新しい値）。
zabbix[stats,]	リモート Zabbix サーバーまたはプロキシの内部メトリクス。
zabbix[stats,,queue,]	リモート Zabbix サーバーまたはプロキシの内部キューメトリクス。
zabbix[tcache,]	Zabbix トレンド関数キャッシュの有効性統計。
zabbix[triggers]	Zabbix データベース内で有効なトリガーの数（有効なホスト上で全アイテムが有効なもの）。
zabbix[uptime]	Zabbix サーバーまたはプロキシプロセスの稼働時間（秒単位）。
zabbix[vcache,buffer,]	Zabbix 値キャッシュの可用性統計。
zabbix[vcache,cache,]	Zabbix 値キャッシュの有効性統計。
zabbix[version]	Zabbix サーバーまたはプロキシのバージョン。
zabbix[vmware,buffer,]	Zabbix vmware キャッシュの可用性統計。
zabbix[vps,written]	データベースに書き込まれた履歴値の合計数。
zabbix[wcache,]	Zabbix 書き込みキャッシュの統計と可用性。

アイテムキーの詳細

- 山括弧のないパラメータは必須であり、そのまま使用する必要があります（例：zabbix[host,<type>,available]の”host” および”available”）。
- 山括弧 **< >** のあるパラメータは、有効な値に置き換える必要があります。パラメータにデフォルト値がある場合、省略できます。
- 「プロキシではサポートされていません」とラベル付けされたアイテムやアイテムパラメータの値は、ホストがサーバーによって監視されている場合にのみ取得できます。逆に、「サーバーではサポートされていません」とラベル付けされた値は、ホストがプロキシによって監視されている場合にのみ取得できます。

zabbix[boottime]

 Zabbix サーバーまたは Zabbix プロキシプロセスの起動時刻 (秒単位)。
 戻り値: 整数。

zabbix[cluster,discovery,nodes]

 高可用性クラスタのノードを検出します。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムは、ローレベルディスカバリで使用できます。

zabbix[connector_queue]

 コネクタキューにエンキューされた値の数。
 戻り値: 整数。

zabbix[discovery_queue]

 ディスカバリキューに入っているネットワークチェックの数。
 戻り値: 整数。

zabbix[host,,items]

 ホスト上で有効なアイテム数（サポートされているものとサポートされていないものの両方）。
 戻り値: 整数。

zabbix[host,,items_unsupported]

 ホストで有効なサポートされていないアイテムの数。
 戻り値: 整数。

zabbix[host,,maintenance]

 ホストの現在のメンテナンスステータス。
 返り値: *0* - 通常状態; *1* - データ収集ありのメンテナンス; *2* - データ収集なしのメンテナンス。

コメント:

- このアイテムは、ホストの場所（サーバーまたはプロキシ）に関係なく、常に Zabbix サーバーによって処理されます。プロキシはこのアイテムを設定データとともに受信しません。
- 2 番目のパラメータは空でなければならず、将来の使用のために予約されています。

zabbix[host,active_agent,available]

 ホスト上のアクティブエージェントチェックの可用性。
 戻り値: *0* - 不明; *1* - 利用可能; *2* - 利用不可。

zabbix[host,discovery,interfaces]

 Zabbix フロントエンドで設定されたホストのすべてのインターフェースの詳細。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムはローレベルディスカバリで使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[host,<type>,available]

 ホスト上の特定のタイプのチェックのメインインターフェースの可用性。
 戻り値: *0* - 利用不可; *1* - 利用可能; *2* - 不明。

パラメータ:

- **type** - *agent*、*snmp*、*ipmi*、または *jmx*。

コメント:

- アイテム値は、ホストの到達不能/利用不可に関する設定パラメータに従って計算されます。

zabbix[hosts]

 監視対象ホストの数。
 戻り値: 整数。

zabbix[items]

 有効なアイテム数（サポートされているものとサポートされていないもの）。
 戻り値: 整数。

zabbix[items_unsupported]

 サポートされていないアイテムの数。
 戻り値: 整数。

zabbix[java,,<param>]

 Zabbix Java ゲートウェイに関する情報です。
 戻り値: *1* - <param> が *ping* の場合; *Java* ゲートウェイのバージョン - <param> が *version* の場合 (例: "8.0.0")。

パラメータ:

- **param** - *ping* または *version*。

コメント:

- このアイテムは、nodata() トリガー関数を使用して Java ゲートウェイの可用性をチェックするために使用できます。
- 2 番目のパラメータは空にする必要があり、将来の使用のために予約されています。

zabbix[lld_queue]

 低レベルディスカバリ処理キューにエンキューされた値の数。
 戻り値: 整数。

コメント:

- このアイテムは、低レベルディスカバリ処理キューの長さを監視するために使用できます。

zabbix[preprocessing]

 前処理マネージャーが受信した値の統計情報:
 - *queued* - 前処理が必要なキューイングされた値の数とサイズ (カウンター)
 - *direct* - 前処理が不要なキューイングされた値の数とサイズ (カウンター)
 - *queue* - 前処理キューにキューイングされた値の数 (zabbix[preprocessing_queue] と同じ)
 戻り値: *JSON*。

戻り値の例:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 前処理キューに入れられた値の数。
 戻り値: 整数。

コメント:

- このアイテムは前処理キューの長さを監視するために使用できます。

zabbix[process,<type>,<mode>,<state>]

 特定の Zabbix プロセスまたはプロセスグループ（<type> と <mode> で識別）が <state> で費やした時間の割合。直近 1 分間のみ計算されます。
 戻り値：*Float*。

パラメータ:

- **type** - サーバープロセスの場合：*agent poller*、*alert manager*、*alert syncer*、*alerter*、*availability manager*、*browser poller*、*configuration syncer*、*configuration syncer worker*、*connector manager*、*connector worker*、*discovery manager*、*discovery worker*、*escalator*、*ha manager*、*history poller*、*history syncer*、*housekeeper*、*http agent poller*、*http poller*、*icmp pinger*、*internal poller*、*ipmi manager*、*ipmi poller*、*java poller*、*lld manager*、*lld worker*、*odbc poller*、*poller*、*preprocessing manager*、*preprocessing worker*、*proxy group manager*、*proxy poller*、*self-monitoring*、*service manager*、*snmp poller*、*snmp trapper*、*task manager*、*timer*、*trapper*、*trigger housekeeper*、*unreachable poller*、*vmware collector*;
プロキシプロセスの場合：*agent poller*、*availability manager*、*browser poller*、*configuration syncer*、*data sender*、*discovery manager*、*discovery worker*、*history syncer*、*housekeeper*、*http agent poller*、*http poller*、*icmp pinger*、*internal poller*、*ipmi manager*、*ipmi poller*、*java poller*、*odbc poller*、*poller*、*preprocessing manager*、*preprocessing worker*、*self-monitoring*、*snmp poller*、*snmp trapper*、*task manager*、*trapper*、*unreachable poller*、*vmware collector*;
- **mode** - *avg* - 指定したタイプのすべてのプロセスの平均値（デフォルト）；
count - 指定したプロセスタイプのフォーク数を返します。<state> は指定しないでください；
max - 最大値；
min - 最小値；
<*process number*> - プロセス番号（1 から事前フォークされたインスタンス数までの間；例えば、4 つの trapper が実行されている場合、値は 1 から 4 の間）；
- **state** - *busy* - プロセスがビジー状態、例えばリクエスト処理中（デフォルト）；
idle - プロセスがアイドル状態で何もしていない。

コメント:

- <mode> が実行されていない Zabbix プロセス番号の場合（例えば、5 つの poller が実行されているのに <mode> が 6 の場合）、そのアイテムはサポート対象外になります。
- 最小値と最大値は、単一プロセスの使用率を指します。したがって、3 つの poller グループで各プロセスの使用率が 2、18、66 だった場合、min は 2、max は 66 を返します。
- プロセスは共有メモリで自分の状態を報告し、self-monitoring プロセスがそのデータを毎秒集計します。状態の変化（busy/idle）は変化時に登録されます。したがって、プロセスが busy になった場合、その状態として登録され、idle になるまで状態は変更または更新されません。これにより、完全にハングしたプロセスでも 100% busy として正しく登録されます。
- 現在、「busy」は「スリープしていない」ことを意味しますが、将来的には追加の状態（ロック待ち、データベースクエリの実行など）が導入される可能性があります。非同期 poller は、server/proxyの MaxConcurrentChecksPerPoller 設定パラメータで設定された上限に達した場合、busy と見なされることに注意してください。
- Linux およびほとんどの他のシステムでは、分解能は 1/100 秒です。

例:

```
zabbix[process,poller,avg,busy] #直近1分間にpollerプロセスが何かをしていた平均時間
zabbix[process,"icmp pinger",max,busy] #直近1分間に任意のICMP pingerプロセスが何かをしていた最大時間
zabbix[process,"history syncer",2,busy] #直近1分間にhistory syncerの2番目が何かをしていた時間
zabbix[process,trapper,count] #現在実行中のtrapperプロセスの数
```

zabbix[proxy,<name>,<param>]

 Zabbix プロキシに関する情報。
 戻り値: 整数。

パラメータ:

- **name** - プロキシ名
- **param** - *lastaccess* - プロキシから受信した最後のハートビートメッセージのタイムスタンプ
delay - 収集された値が未送信のままになっている時間。計算式は「プロキシ遅延」+（「現在のサーバー時刻」-「プロキシの lastaccess」）で、「プロキシ遅延」はプロキシの現在時刻とプロキシ上の最も古い未送信値のタイムスタンプの差。

コメント:

- このアイテムは、ホストの場所（サーバーまたはプロキシ上）に関係なく、常に Zabbix サーバーによって処理されます。
- プロキシの可用性を確認するために、`fuzzytime()`関数を使用できます。

例:

```
zabbix[proxy,"Germany",lastaccess] # "Germany"プロキシから受信した最後のハートビートメッセージのタイムス
```

zabbix[proxy,discovery]

 名前、モード、暗号化、圧縮、バージョン、最終確認時刻、ホスト数、アイテム数、必要な値/秒（vps）、バージョンステータス（current/outdated/unsupported）、アイテムタイプごとのタイムアウト、プロキシグループ名（プロキシがグループに属している場合）、状態（unknown/offline/online）を含む Zabbix プロキシのリスト。
 戻り値：*JSON* オブジェクト。

zabbix[proxy group,<name>,available]

 プロキシグループ内のオンラインプロキシの数。
 戻り値: 整数。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,pavailable]

 プロキシグループ内のオンラインプロキシの割合。
 戻り値: *Float*。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,proxies]

 名前、モード、暗号化、圧縮、バージョン、最終確認、ホスト数、アイテム数、必要な値/秒（vps）、バージョンステータス（current/outdated/unsupported）、タイムアウト、プロキシグループ名、状態（unknown/offline/online）を持つプロキシグループ内の Zabbix プロキシのリスト。
 戻り値 : *JSON*。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,state]

 プロキシグループの状態。
 戻り値: *0* - 不明; *1* - オフライン; *2* - 復旧中; *3* - オンライン; *4* - 劣化中。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,discovery]

 設定データとリアルタイムデータを持つプロキシグループのリストを返します。設定データには、プロキシグループ名、フェイルオーバー遅延、必要なオンラインプロキシの最小数が含まれます。リアルタイムデータには、プロキシグループの状態（詳細はコメント参照）、オンラインプロキシの数、オンラインプロキシの割合が含まれます。
 戻り値 : *JSON*。

コメント:

- このアイテムはグループに属さないプロキシは返しません。
- ”failover_delay” または”min_online” に無効な値がある場合は、特別な値*-1* が報告されます。無効な値は、設定にマクロが使用されていて、そのマクロが有効な値に展開できない場合に発生することがあります。
- プロキシグループの状態は整数で報告されます : *0* - 不明; *1* - オフライン; *2* - 復旧中; *3* - オンライン; *4* - 劣化中。

戻り値の例 :

```
{
    "groups": [
       { "name": "Riga", "failover_delay": 60, "min_online": 1 },
       { "name": "Tokyo", "failover_delay": 60, "min_online": 2 },
       { "name": "Porto Alegre", "failover_delay": 60, "min_online": 3 }
    ],
    "details": {
        "Riga": { "state": 3, "available": 10, "pavailable": 20 },
        "Tokyo": { "state": 3, "available": 10, "pavailable": 20 },
        "Porto Alegre": { "state": 1, "available": 0, "pavailable": 0 }
    }
}
```

zabbix[proxy_buffer,buffer,<mode>]

 プロキシのメモリバッファ使用統計。
 戻り値: *Integer* (サイズの場合); *Float* (パーセンテージの場合)。

パラメータ:

- **mode**: *total* - バッファの合計サイズ (メモリバッファが有効かどうかの確認に使用可能);
free - 空きバッファのサイズ;
pfree - 空きバッファのパーセンテージ;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファのパーセンテージ。

コメント:

- メモリバッファが無効な場合は「Proxy memory buffer is disabled」エラーを返します;
- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[proxy_buffer,state,changes]

 起動後、ディスク/メモリバッファモード間の状態変更回数を返します。
 戻り値: 整数; *0* - メモリバッファが無効です。

コメント:

- 状態変更が頻繁に発生する場合は、メモリバッファのサイズまたは有効期間を増やす必要があります。
- メモリバッファの状態を監視する頻度が低い場合 (例えば 1 分に 1 回) は、バッファが状態を切り替えても記録されないことがあります。

zabbix[proxy_buffer,state,current]

 新しいデータが保存されている現在の作業状態を返します。
 戻り値: *0* - ディスク; *1* - メモリ。

コメント:

- メモリバッファが無効な場合も”0” が返されます。

zabbix[proxy_history]

 サーバーに送信されるのを待っているプロキシ履歴テーブル内の値の数。
 戻り値: 整数。

コメント:

- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[queue,<from>,<to>]

 監視対象アイテムのうち、<from> 秒以上、<to> 秒未満遅延しているキュー内のアイテム数を返します。
 戻り値: 整数。

パラメータ:

- **from** - 少なくともこの秒数だけ遅延している（デフォルトは 6 秒）;
- **to** - 最大でこの秒数だけ遅延している（デフォルトは無限大）。

コメント:

- パラメータには時間のサフィックス（s,m,h,d,w）が使用できます。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

zabbix[rcache,<cache>,<mode>]

 Zabbix 設定キャッシュの可用性統計情報。
 戻り値: サイズの場合は整数; パーセンテージの場合は浮動小数点数。

パラメータ:

- **cache** - *buffer*;
- **mode** - *total* - バッファの合計サイズ;
free - 空きバッファのサイズ;
pfree - 空きバッファのパーセンテージ;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファのパーセンテージ。

zabbix[requiredperformance]

 Zabbix サーバーまたは Zabbix プロキシの必要なパフォーマンス（1 秒あたりの新しい値の期待値）。
 戻り値: *Float*。

コメント:

- レポート → システム情報 の「必要なサーバーパフォーマンス（1 秒あたりの新しい値）」とほぼ相関します。

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

 リモートの Zabbix サーバーまたはプロキシの内部メトリクス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **ip** - リモートでクエリを実行するサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）
- **port** - リモートでクエリを実行するサーバー/プロキシのポート（デフォルトは 10051）

コメント:

- stats リクエストは、ターゲットインスタンスのserver/proxyパラメータの’StatsAllowedIP’ にリストされているアドレスからのみ受け付けられます。
- このアイテムによって返されるのは、内部メトリクスの選択されたセットです。詳細はZabbix stats のリモート監視を参照してください。

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

 リモート Zabbix サーバーまたはプロキシの内部キューメトリクス（`zabbix[queue,<from>,<to>]` を参照）。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **ip** - リモートでクエリするサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）;
- **port** - リモートでクエリするサーバー/プロキシのポート（デフォルトは 10051）;
- **from** - 少なくともこの秒数だけ遅延（デフォルトは 6 秒）;
- **to** - 最大でこの秒数だけ遅延（デフォルトは無限大）。

コメント:

- stats リクエストは、ターゲットインスタンスのserver/proxyパラメータの'StatsAllowedIP' にリストされているアドレスからのみ受け付けられます。
- このアイテムによって、選択された内部メトリクスのセットが返されます。詳細はZabbix stats のリモート監視を参照してください。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

zabbix[tcache,<cache>,<parameter>]

 Zabbix トレンド関数キャッシュの有効性統計。
 戻り値: *Integer* (サイズの場合); *Float* (パーセンテージの場合)。

パラメータ:

- **cache** - *buffer*;
- **parameter** - *all* - キャッシュリクエストの合計 (デフォルト);
hits - キャッシュヒット数;
phits - キャッシュヒットのパーセンテージ;
misses - キャッシュミス数;
pmisses - キャッシュミスのパーセンテージ;
items - キャッシュされたアイテム数;
requests - キャッシュされたリクエスト数;
pitems - キャッシュされたアイテム数/(キャッシュされたアイテム数 + リクエスト数) のパーセンテージ。パーセンテージが低い場合、キャッシュサイズを減らすことができます。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[triggers]

 有効なホスト上のすべてのアイテムが有効な Zabbix データベース内の有効なトリガーの数。
 戻り値: 整数。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[uptime]

 Zabbix サーバーまたはプロキシプロセスの稼働時間 (秒単位)。
 戻り値: 整数。

zabbix[vcache,buffer,<mode>]

 Zabbix 値キャッシュの可用性統計情報。
 戻り値: サイズの場合は整数; パーセンテージの場合は浮動小数点数。

パラメータ:

- **mode** - *total* - バッファの合計サイズ;
free - 空きバッファのサイズ;
pfree - 空きバッファのパーセンテージ;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファのパーセンテージ。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[vcache,cache,<parameter>]

 Zabbix 値キャッシュの有効性統計。
 戻り値: 整数。*mode* パラメータを指定した場合の戻り値: 0 - 通常モード; 1 - 低メモリモード。

パラメータ:

- **parameter** - *requests* - リクエストの総数;
hits - キャッシュヒット数（キャッシュから取得した履歴値）;
misses - キャッシュミス数（データベースから取得した履歴値）;
mode - 値キャッシュの動作モード。

コメント:

- 低メモリモードが有効になると、このモードを引き起こした問題が早期に解決しても、値キャッシュは 24 時間この状態を維持します。

- このキーは、秒間の変化の前処理ステップと組み合わせて、1 秒あたりの統計値を取得するために使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbixversion

 Zabbix サーバーまたはプロキシのバージョン。
 戻り値: 文字列。例: 8.0.0。

zabbix[vmware,buffer,<mode>]

 Zabbix vmware キャッシュの可用性統計情報。
 戻り値：*Integer*（サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ：

- **mode** - *total* - バッファの合計サイズ；
free - 空きバッファのサイズ；
pfree - 空きバッファのパーセンテージ；
used - 使用済みバッファのサイズ；
pused - 使用済みバッファのパーセンテージ。

zabbix[vps,written]

 データベースに書き込まれた履歴値の合計数。
 戻り値: 整数。

zabbix[wcache,<cache>,<mode>]

 Zabbix 書き込みキャッシュの統計と可用性。
 戻り値: 整数（数値/サイズの場合）、浮動小数点数（パーセンテージの場合）。

パラメータ:

- **cache** - *values*、*history*、*index*、または *trend*;
- **mode** -（*values* の場合）*all*（デフォルト）- Zabbix サーバー/プロキシによって処理された値の合計数（未サポートアイテムを除く）（カウンタ）；
float - 処理された浮動小数点値の数（カウンタ）；
uint - 処理された符号なし整数値の数（カウンタ）；
str - 処理された文字列値の数（カウンタ）；
log - 処理されたログ値の数（カウンタ）；
text - 処理されたテキスト値の数（カウンタ）；
not supported - アイテム処理がアイテムを未サポート状態にした、またはその状態を維持した回数（カウンタ）；
（*history*、*index*、*trend* キャッシュの場合）*pfree*（デフォルト）- 空きバッファのパーセンテージ；
total - バッファの合計サイズ；
free - 空きバッファのサイズ；
used - 使用中バッファのサイズ；
pused - 使用中バッファのパーセンテージ。

コメント:

- <cache> の指定は必須です。`trend` キャッシュパラメータは Zabbix プロキシではサポートされていません。
- ヒストリキャッシュはアイテム値を保存するために使用されます。値が小さい場合は、データベース側のパフォーマンス問題を示します。
- ヒストリインデックスキャッシュは、ヒストリキャッシュに保存された値をインデックス化するために使用されます。
- ヒストリキャッシュが満杯になり、その後クリアされた場合でも、ヒストリインデックスキャッシュは一部のデータを保持し続けます。この動作は想定されており、メモリサイズを常に変更するために必要な追加処理を回避することで、システムをより効率的に動作させます。
- トレンドキャッシュは、データを受信するすべてのアイテムの現在の 1 時間分の集計を保存します。
- zabbix[wcache,values] キーを *1* 秒あたりの変化量の前処理ステップと組み合わせて使用することで、1 秒あたりの値の統計を取得できます。

9 SSH チェック

概要

SSH チェックは、エージェントレス監視として実行されます。SSH チェックには Zabbix エージェントは必要ありません。

SSH チェックを実行するには、Zabbix サーバーを最初に SSH2 サポート（libssh または libssh2）で構成する必要があります。詳細は要件も参照してください。

> **Attention:**
> RHEL 8 以降では、libssh のみがサポートされています。他のディストリビューションでは、libssh2 よりも libssh の使用を推奨します。

設定

パスフレーズ認証

SSH チェックは、ユーザー/パスワードのペアとキーファイルベースの 2 つの認証方法を提供します。

キーを使用しない場合は、ソースからビルドする場合に libssh または libssh2 を Zabbix にリンクする以外に、追加の設定は必要ありません。

キーファイル認証

SSH アイテムでキーベースの認証を使用するには、サーバーの設定にいくつかの変更が必要です。

Zabbix サーバーの設定ファイル (*zabbix_server.conf*) を `root` として開き、次の行を探します。

```
##### SSHKeyLocation=
```

コメントを外し、公開鍵と秘密鍵が配置されるフォルダへのフルパスを設定します。

```
SSHKeyLocation=/home/zabbix/.ssh
```

ファイルを保存し、その後 Zabbix サーバーを再起動します。

ここでの*/home/zabbix* は *zabbix* ユーザーアカウントのホームディレクトリであり、*.ssh* はデフォルトで公開鍵と秘密鍵がssh-keygenコマンドによってホームディレクトリ内に生成されるディレクトリです。

通常、さまざまな OS ディストリビューションからの Zabbix サーバーのインストールパッケージは、*zabbix* ユーザーアカウントを別の場所（たとえば、システムアカウント用の*/var/lib/zabbix*）にホームディレクトリを作成します。

鍵を生成する前に、ホームディレクトリを*/home/zabbix* に再割り当てして、上記の Zabbix サーバー設定パラメータ `SSHKeyLocation` と一致させることができます。

> **Note:**
> インストールセクション](/manual/installation/install/sources#create_user_account) に従って *zabbix* アカウントが手動で追加されている場合、以下の手順はスキップできます。その場合、*zabbix* アカウントのホームディレクトリはすでに*/home/zabbix* である可能性が高いです。

zabbix ユーザーアカウントのホームディレクトリを変更するには、それを使用しているすべてのワーキングプロセスを停止する必要があります。

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

ホームディレクトリの場所を変更し、存在する場合は移動を試みるには、次のコマンドを実行します。

```
usermod -m -d /home/zabbix zabbix
```

古い場所にホームディレクトリが存在しなかった場合は、新しい場所に作成する必要があります。安全な試みとしては次のとおりです。

```
test -d /home/zabbix || mkdir /home/zabbix
```

すべてが安全であることを確認するために、ホームディレクトリのパーミッションを設定する追加のコマンドを実行できます。

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

以前に停止したプロセスを再度開始できます。

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

次のコマンドで公開鍵と秘密鍵を生成する手順を実行できます（可読性を高めるため、コマンドプロンプトはコメントアウトされています）。

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |       .         |
##### |      o          |
##### | .     o         |
```

```
##### |+      . S         |
##### |.+    o =         |
##### |E .    * =        |
##### |=o . ..* .        |
##### |... oo.o+         |
##### +-----------------+
```

Note:
公開鍵と秘密鍵（*id_rsa.pub* および *id_rsa*）は、デフォルトで*/home/zabbix/.ssh* ディレクトリに生成されており、これは Zabbix サーバーの SSHKeyLocation 設定パラメータに対応しています。

Attention:
”rsa” 以外の鍵タイプは ssh-keygen ツールや SSH サーバーでサポートされている場合がありますが、Zabbix で使用されている libssh2 ではサポートされていない場合があります。

シェル設定フォーム

この手順は、SSH チェックで監視される各ホストごとに 1 回だけ実行する必要があります。

以下のコマンドを使用して、公開鍵ファイルをリモートホスト *10.10.10.10* にインストールできます。これにより、*root* アカウントで SSH チェックを実行できるようになります（可読性を高めるため、コマンドプロンプトはコメントアウトされています）。

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

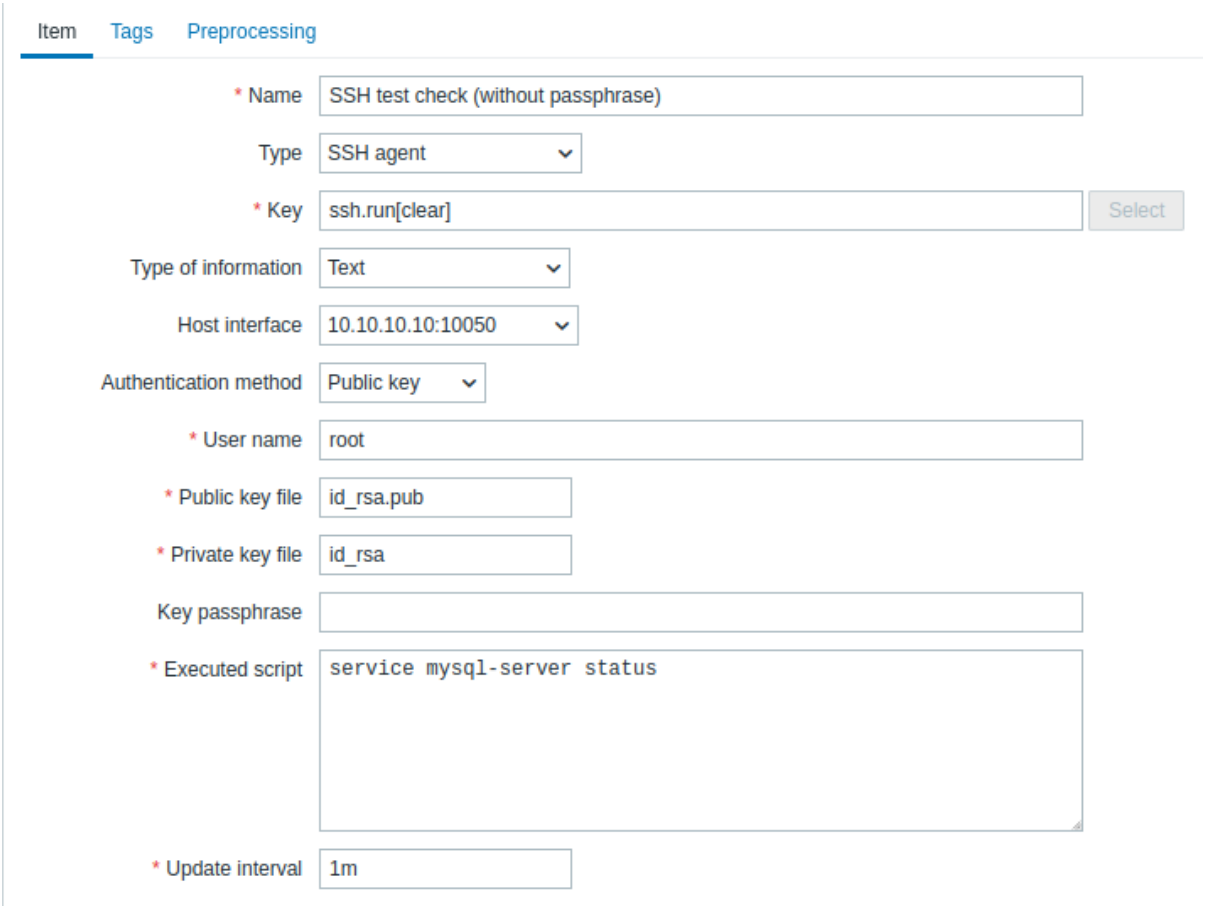
これで、*zabbix* ユーザーアカウントのデフォルトの秘密鍵（*/home/zabbix/.ssh/id_rsa*）を使用して SSH ログインを確認できるようになります。

```
sudo -u zabbix ssh root@10.10.10.10
```

ログインが成功した場合、シェルでの設定は完了し、リモート SSH セッションを閉じることができます。

アイテムの設定

実行するコマンドは、アイテム設定の実行スクリプトフィールドに入力する必要があります。複数のコマンドを改行して記述することで、順番に実行できます。この場合、返される値も複数行でフォーマットされます。



必須入力フィールドは赤いアスタリスクでマークされています。

SSH アイテムに固有の情報が必要なフィールドは以下の通りです。

パラメータ	説明	コメント
タイプ	ここで **SSH** エージェントを選択します。	

パラメータ	説明	コメント

キー	**ssh.run[**一意の短い説明**,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]** 形式のホストごとに一意なアイテムキー	一意の短い説明は必須で、ホストごとに各 SSH アイテムで一意である必要があります。
		
		デフォルトのポートは 22 であり、このアイテムが割り当てられているインターフェースで指定されたポートではありません。
		
		ssh options は、*key1=value1;key2=value2,value3* の形式で追加の SSH オプションを渡すことができます。1 つのキーに複数の値をカンマ区切りで渡すことができ（この場合、パラメータは引用符で囲む必要があります）、複数のオプションキーはセミコロンで区切って渡すことができます。
		
		サポートされているオプションキーは次の通りです：`KexAlgorithms`、`HostkeyAlgorithms`、`Ciphers`、`MACs`、`PubkeyAcceptedKeyTypes`。オプションキーと値のサポートは SSH ライブラリによって異なります（例：`PubkeyAcceptedKeyTypes` は libssh でのみサポートされます）。サポートされていないオプションが指定された場合、エラーが返され、アイテムは未サポートとなります。
		
		暗号設定の追加のための”+” 記号や、特定の暗号設定を無効にするための”!” 記号（GnuTLS や OpenSSL のような）はサポートされていませんのでご注意ください。
		
		例：
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		subsystem は SSH サブシステムを渡すためのもので、SSH 接続をサブシステムで許可された特定の操作（例：SFTP によるファイル転送や NETCONF によるネットワーク機器管理）に限定します。サブシステムを使用する場合、実行スクリプトパラメータで特定のスクリプト構文を使用する必要がある場合がありますのでご注意ください。
		
		例：
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

認証方法	” パスワード” または” 公開鍵” のいずれか。	
ユーザー名	リモートホストで認証するためのユーザー名（最大 255 文字）。必須。	
公開鍵ファイル	認証方法が” 公開鍵” の場合の公開鍵ファイル名。必須。	例：*id_rsa.pub* - コマンドssh-keygenで生成されるデフォルトの公開鍵ファイル名。
秘密鍵ファイル	認証方法が” 公開鍵” の場合の秘密鍵ファイル名。必須。	例：*id_rsa* - デフォルトの秘密鍵ファイル名。
パスワードまたは 鍵パスフレーズ	認証用のパスワード（最大 255 文字）または秘密鍵にパスフレーズを使用した場合のパスフレーズ。	パスフレーズを使用しなかった場合は鍵パスフレーズフィールドを空欄にしてください。 パスフレーズ使用に関する既知の問題も参照してください。

パラメータ	説明	コメント

実行スクリプト	SSH リモートセッションで実行されるシェルコマンド。	実行されたシェルコマンドの戻り値は 16MB に制限されます（末尾の空白も含み、切り捨てられます）。データベースの制限も適用されます。
		
		libssh2 ライブラリでは、実行可能なスクリプトが約 32kB に切り捨てられる場合があることに注意してください。
		
		例：
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		例（NETCONF サブシステム用）：
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

10 Telnet チェック

概要

Telnet チェックはエージェントレス監視として実行されます。Zabbix エージェントは必要ありません。

設定可能なフィールド

実行する実際のコマンドは、アイテム設定の実行するスクリプトフィールドに記載する必要があります。
複数のコマンドを記載すると続けて実行できます。この場合、戻り値も複数行としてフォーマットされます。

シェルプロンプトの最後に使用できる文字:

- $
- #
- >
- %

Note:
これらの文字のいずれかで終わる Telnet プロンプト行は戻り値から削除されますが、ただしそれはコマンドリストの最初のコマンド、つまり Telnet セッションの開始時のみです。

キー	説明
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Telnet 接続を使用してリモートデバイス上でコマンドを実行する

Attention:
Telnet チェックで非 ASCII 文字および非 UTF8 エンコーディングの値を返す場合は、キーの *<encoding>* パラメーターを適切に指定する必要があります。詳細は戻り値のエンコードページを参照してください。

11 外部チェック

概要

外部チェックは、Zabbix サーバーがシェルスクリプトまたはバイナリを実行することで実行されるチェックです。ただし、ホストが Zabbix プロキシによって監視されている場合、外部チェックはプロキシによって実行されます。

外部チェックは、監視対象ホストでエージェントを実行する必要がありません。

アイテムキーの構文は以下の通りです。

`script[<parameter1>,<parameter2>,...]`

ここで:

引数	定義
script	シェルスクリプトまたはバイナリの名前
parameter(s)	コマンドラインパラメータ（省略可能）

スクリプトにパラメータを渡したくない場合は、以下のように指定できます。

`script[]` または
`script`

Zabbix サーバーまたはプロキシは、外部スクリプト用に指定されたディレクトリを検索し、コマンドを実行します（Zabbixサーバー/プロキシの設定ファイルの `ExternalScripts` パラメータを参照）。コマンドは Zabbix サーバー/プロキシと同じユーザーで実行されるため、必要に応じてラッパースクリプト内でアクセス権や環境変数を処理する必要があります。コマンドのパーミッションも、そのユーザーが実行できるように設定する必要があります。指定されたディレクトリ内のコマンドのみが実行可能です。

Warning:
外部チェックの多用は避けてください。各スクリプトの実行には Zabbix サーバー/プロキシによる fork プロセスの開始が必要となり、多数のスクリプトを実行すると Zabbix のパフォーマンスが大幅に低下する可能性があります。

使用例

最初のパラメータ'-h' でスクリプト **check_oracle.sh** を実行します。2 番目のパラメータは、ホストプロパティでの選択に応じて、IP アドレスまたは DNS 名に置き換えられます。

`check_oracle.sh["-h","{HOST.CONN}"]`

ホストが IP アドレスを使用するように設定されていると仮定すると、Zabbix サーバー/プロキシは次のように実行します。

`check_oracle.sh '-h' '192.168.1.4'`

外部チェックの結果

外部チェックの戻り値は、チェックによって生成された標準出力と標準エラーです。

Attention:
テキスト (文字列、ログ、またはテキスト型の情報) を返すアイテムは、標準エラー出力があっても未サポートにはなりません。

戻り値は 16MB に制限されています (末尾の空白も含み、切り捨てられます)。データベースの制限も適用されます。

要求されたスクリプトが見つからない場合や、Zabbix サーバー/プロキシに実行権限がない場合、アイテムは未サポートとなり、対応するエラーメッセージが表示されます。

タイムアウトの場合、アイテムは未サポートとなり、対応するエラーメッセージが表示され、スクリプト用に fork されたプロセスは終了します。

12 トラッパーアイテム

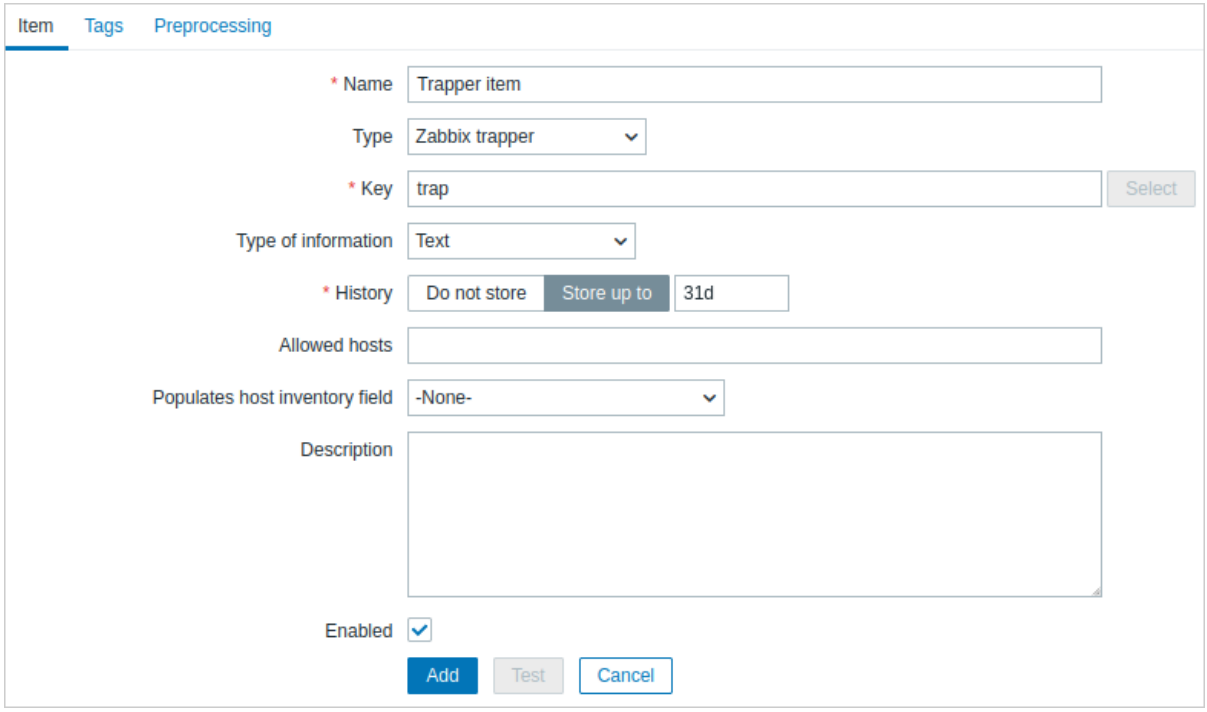
概要

トラッパーアイテムは、データを問い合わせる代わりに受信データを受け付けます。これは、Zabbix に送信したい任意のデータに便利です。

設定

トラッパーアイテムを設定するには:

1. データ収集 → ホストに移動します。
2. ホストの行で アイテムをクリックします。
3. アイテムの作成をクリックします。
4. 設定フォームにアイテムのパラメータを入力します。



必須入力フィールドは赤いアスタリスクでマークされています。

トラッパーアイテムに特有の情報が必要なフィールドは次のとおりです。

タイプ	「Zabbix トラッパー」を選択します。
キー	Zabbix サーバーにデータを送信する際にアイテムを識別するために使用されるキーを入力します。
情報のタイプ	送信されるデータのフォーマットに対応する情報のタイプを選択します。
許可されたホスト	カンマ区切りの IP アドレス（オプションで CIDR 表記）または DNS 名のリスト。 指定した場合、ここにリストされたホストからの着信接続のみが受け入れられます。 IPv6 サポートが有効な場合、「127.0.0.1」、「::127.0.0.1」、「::ffff:127.0.0.1」は同等に扱われ、「::/0」は任意の IPv4 または IPv6 アドレスを許可します。「0.0.0.0/0」は任意の IPv4 アドレスを許可するために使用できます。 「IPv4 互換 IPv6 アドレス」（0000::/96 プレフィックス）はサポートされていますが、RFC4291で非推奨となっています。 例: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} スペース、ユーザーマクロ、およびホストマクロ {HOST.HOST}、{HOST.NAME}、{HOST.IP}、{HOST.DNS}、{HOST.CONN} がサポートされています。

Note:
値を送信する前に、アイテムを保存してから Zabbix サーバーが設定キャッシュの更新で変更を取得するまで最大 60 秒待つ必要がある場合があります。

データの送信

Zabbix senderユーティリティまたは Zabbix senderプロトコルを使用して、Zabbix サーバーまたはプロキシにデータを送信することができます。`history.push`API メソッドを使用して、Zabbix サーバーにデータを送信することも可能です。

Zabbix sender

Zabbix sender ユーティリティを使用して Zabbix サーバーまたはプロキシにデータを送信するには、次のコマンドを実行して「test value」を送信できます。

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

「test value」を送信するために、次のコマンドオプションが使用されます。

- -z Zabbix サーバーの IP アドレスを指定

- -p Zabbix サーバーのポート番号を指定（デフォルトは 10051）
- -s ホストを指定（表示名ではなく、必ず技術名を使用してください）
- -k トラッパーアイテムで設定されたアイテムキーを指定
- -o 送信する値を指定

Attention:
Zabbix トラッパープロセスは、アイテムキーに使用されているマクロを展開して、対象ホストの対応するアイテムキーの存在を確認しません。

Zabbix sender と Zabbix サーバーまたはプロキシ間の通信の詳細については、*Zabbix sender プロトコル*を参照してください。

history.push

history.push API メソッドを使用して Zabbix サーバーにデータを送信するには、以下のようなテスト値を含む HTTP POST リクエストを作成できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

リクエストが正しい場合、API から返されるレスポンスは以下のようになります。

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

レスポンスデータ内のエラーは、特定のアイテムへのデータ送信が Zabbix サーバーによる検証に失敗したことを示します。これは以下の理由で発生する可能性があります。

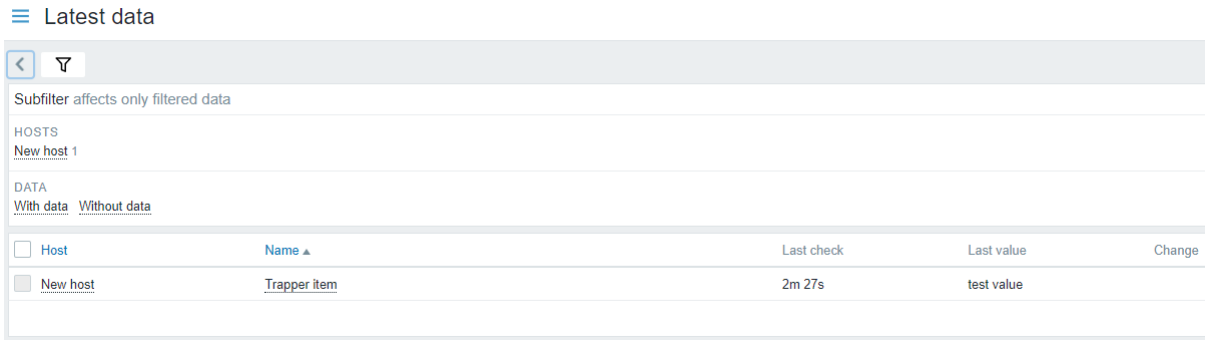
- データを送信するユーザーがアイテムのホストに対する読み取り権限を持っていない
- ホストが無効化されている、またはデータ収集なしのメンテナンス中である
- アイテムが存在しない、またはまだサーバーの設定キャッシュに含まれていない
- アイテムが無効化されている、またはタイプが Zabbix トラッパーまたはHTTP エージェント（トラッピング有効）以外である
- ユーザーの IP または DNS がアイテムの許可ホストリストに設定されていない
- 他のアイテムがナノ秒レベルでタイムスタンプが重複する値を持っている

エラーがない場合は、送信された値が処理のために受け入れられたことを示します。これには（存在する場合）前処理、トリガー処理、データベースへの保存が含まれます。受け入れられた値の処理も失敗する場合があることに注意してください（たとえば、前処理中など）。その場合、値は破棄されます。

Zabbix API の操作方法の詳細については、APIを参照してください。

データの表示

データが送信されたら、監視データ → 最新データに移動して結果を確認できます。



Note:
単一の数値が送信された場合、データのグラフは、その値の時刻の左右に水平線を表示します。

13 JMX 監視

概要

JMX 監視は、Java アプリケーションの JMX カウンタを監視するために使用できます。

JMX 監視は、"Zabbix Java gateway" と呼ばれる Zabbix デーモンの形で Zabbix にネイティブサポートされています。

ホスト上の特定の JMX カウンタの値を取得するために、Zabbix サーバーは Zabbix **Java gateway** にクエリを送り、Java gateway はJMX 管理 APIを使用して、対象のアプリケーションにリモートでクエリを送ります。

詳細およびセットアップについては、Zabbix Java gatewayのセクションを参照してください。

Warning:
Java gateway と監視対象の JMX アプリケーション間の通信は、ファイアウォールで遮断しないでください。

Java アプリケーションのリモート JMX 監視の有効化

Java アプリケーションには追加のソフトウェアをインストールする必要はありませんが、リモート JMX 監視をサポートするために、以下に指定するコマンドラインオプションを付けて起動する必要があります。

最低限、ローカルホスト上のシンプルな Java アプリケーションをセキュリティなしで監視したい場合は、次のオプションで起動します。

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.registry.ssl=false \
-jar /path/to/your/application.jar
```

これにより、Java はローカルホストからのみポート 12345 で JMX 接続を受け付け、認証や SSL を要求しないようになります。

別のインターフェースで接続を許可したい場合は、-Djava.rmi.server.hostname パラメータにそのインターフェースの IP を指定します。

セキュリティをより厳格にしたい場合は、他にも多くの Java オプションが利用できます。例えば、次の例では、より柔軟なオプションセットでアプリケーションを起動し、ローカルホストだけでなく広いネットワークに公開します。

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
```

```
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

これらの設定のほとんど（すべてではないにしても）は、`$JRE/lib/management/management.properties`（またはシステム上のそのファイルがある場所）に指定できます。

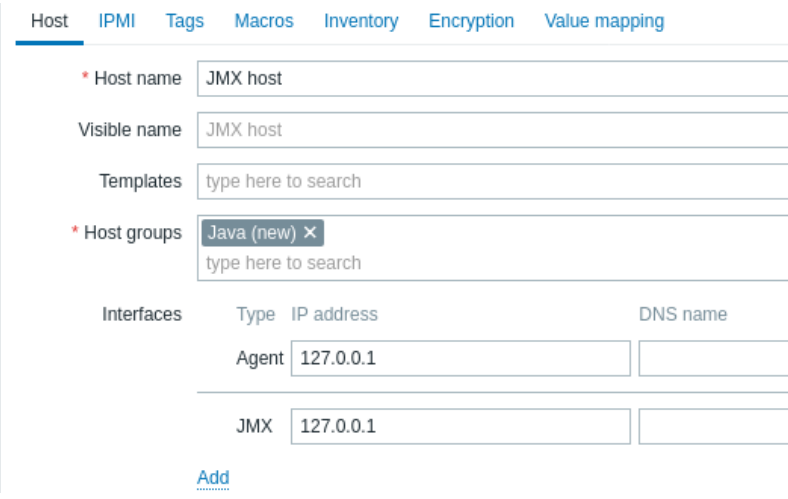
SSL を使用する場合は、Java ゲートウェイがキーストアやトラストストアの場所を認識できるように、`-Djavax.net.ssl.*` オプションを追加して startup.sh スクリプトを修正する必要があることに注意してください。

詳細については、Monitoring and Management Using JMXを参照してください。

Zabbix フロントエンドでの JMX インターフェースと item の設定

Java gateway が動作し、serveer が Java gateway の場所を認識し、リモート JMX 監視をサポートする Java アプリケーションが
 起動したら、Zabbix GUI でインタフェースと item を設定する必要があります。

JMX インターフェースの設定



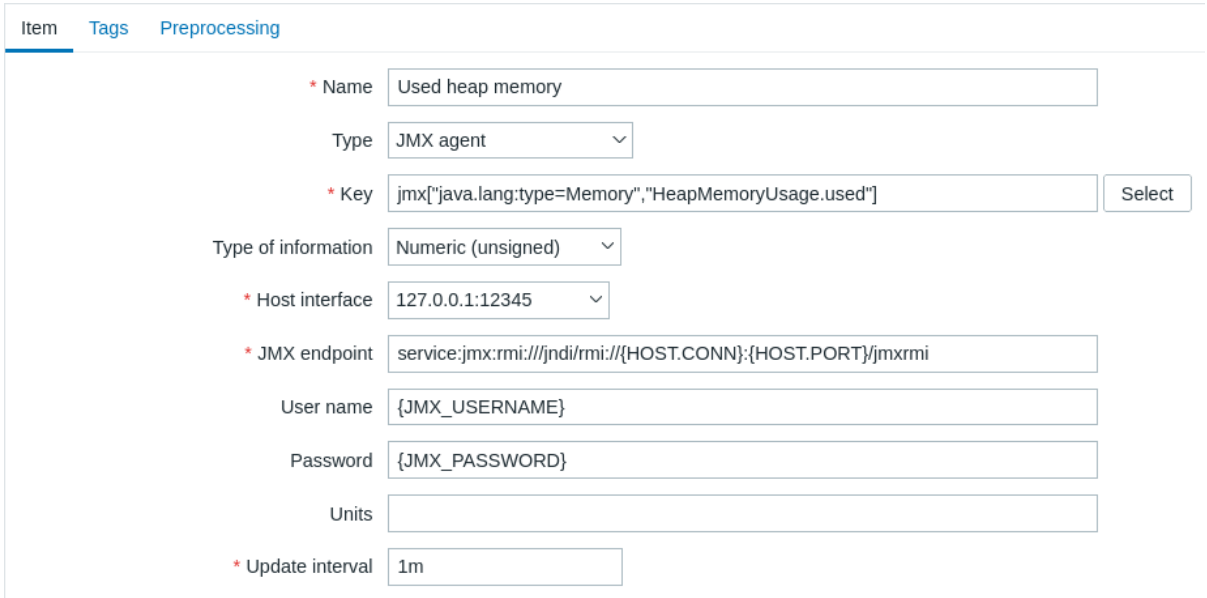
まず､対象のホストでJMX タイプのインターフェイスを作成します。

必須入力項目にはすべて赤いアスタリスクが付けられています。

JMX エージェントアイテムの追加

監視したい各 JMX カウンターについて、そのインターフェースに **JMX** エージェントアイテムを追加します。

下のスクリーンショットのキーは `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]` です。



必須入力フィールドには赤いアスタリスクが付いています。

JMX アイテムに特有の情報が必要なフィールドは以下の通りです。

タイプ	ここで **JMX** エージェントを設定します。

キー	jmx[] アイテムキーには 3 つのパラメータがあります: オブジェクト名 - MBean のオブジェクト名 属性名 - MBean 属性名。複合データフィールド名はドットで区切って指定可能 一意の短い説明 - ホスト上で同じオブジェクト名と属性名を持つ複数の JMX アイテムを許可する一意の説明（省略可能） JMX アイテムキーの詳細は下記を参照してください。 jmx.discovery[] 低レベルディスカバリアイテムを使用して、MBean や MBean 属性を発見することもできます。
JMX エンドポイント	カスタム JMX エンドポイントを指定できます。JMX エンドポイントの接続パラメータが JMX インターフェースと一致していることを確認してください。これは、デフォルトの JMX エンドポイントで行われているように {HOST.*} マクロを使用することで実現できます。 {HOST.*} マクロおよびユーザーマクロがサポートされています。
ユーザー名	Java アプリケーションで認証を設定している場合は、ユーザー名（最大 255 文字）を指定します。 ユーザーマクロがサポートされています。
パスワード	Java アプリケーションで認証を設定している場合は、パスワード（最大 255 文字）を指定します。 ユーザーマクロがサポートされています。

”true” または”false” のいずれかとなる Boolean カウンターを監視したい場合は、情報の型を「数値（整数）」に指定し、[事前処理] タブで「Boolean to decimal」事前処理ステップを選択します。サーバーは Boolean 値をそれぞれ 1 または 0 として保存します。

JMX item キーの詳細

シンプルな属性

MBean オブジェクト名は、Java アプリケーションで定義する文字列にすぎません。一方、属性名はより複雑になる場合があります。属性がプリミティブデータ型（整数、文字列など）を返す場合は、特に心配する必要はありません。キーは次のようになります。

```
jmx[com.example:Type=Hello,weight]
```

この例では、オブジェクト名は”com.example:Type=Hello”、属性名は”weight” で、返される値の型はおそらく ” 数値（浮動小数点）” であるべきです。

複合データを返す属性

例えば、属性名が”apple” で、”weight” 、”color” などのパラメータを表すハッシュを返す場合、属性はより複雑になります。キーは次のようになります。

```
jmx[com.example:Type=Hello,apple.weight]
```

このように、属性名とハッシュ・キーは、ドット記号で区切られます。同じように、属性がネストした複合データを返す場合は、ドットで区切ります。

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

表形式データを返す属性

表形式データ属性は、1 つまたは複数の複合属性から構成されます。属性名パラメータにそのような
 属性が指定された場合、この item 値はその属性の完全な構造を JSON 形式で返します。
 表形式データ属性の中の個々の要素値は、プリプロセスによって取得することができます。

表形式データ属性の例:

Item 値:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

ドットに関する問題

ここまでは順調です。しかし、属性名やハッシュキーにドット記号が含まれている場合はどうでしょうか？以下はその例です。

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

これは問題です。属性名が"all.fruits" であることを、Zabbix にどのように伝えればよいのでしょうか？属性名とハッシュキーを区切るドットと、名前の一部であるドットをどのように区別すればよいのでしょうか？

これは可能です。名前の一部であるドットをバックスラッシュでエスケープするだけです。

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

同様に、ハッシュキーにドットが含まれている場合もエスケープします。

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

その他の問題

属性名にバックスラッシュ文字が含まれる場合は、エスケープする必要があります。

```
jmx[com.example:type=Hello,c:\\documents]
```

JMX アイテムキー内のその他の特殊文字の扱いについては、アイテムキーのフォーマットセクションを参照してください。

実際、これがすべてです。JMX 監視をお楽しみください!

非プリミティブデータ型

toString() メソッドをオーバーライドする非プリミティブデータ型を返すカスタム MBean を使用することができます。

JBoss EAP 6.4 でカスタムエンドポイントを使用する

カスタムエンドポイントを使用すると、デフォルトの RMI 以外のさまざまなトランスポートプロトコルで動作させることができます。

この機能を例示するために、JBoss EAP 6.4 の監視を設定してみましょう。まず、いくつかの前提条件を設定します。

- すでに Zabbix Java ゲートウェイをインストールしていること。まだの場合は、ドキュメントに従ってインストールしてください。
- Zabbix サーバーと Java ゲートウェイは /usr/local/ プレフィックスでインストールされている
- JBoss はすでに /opt/jboss-eap-6.4/ にインストールされており、スタンドアロンモードで動作している
- これらすべてのコンポーネントが同じホスト上で動作していると仮定する
- ファイアウォールと SELinux は無効化されている（または適切に設定されている）

zabbix_server.conf に簡単な設定を行います。

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

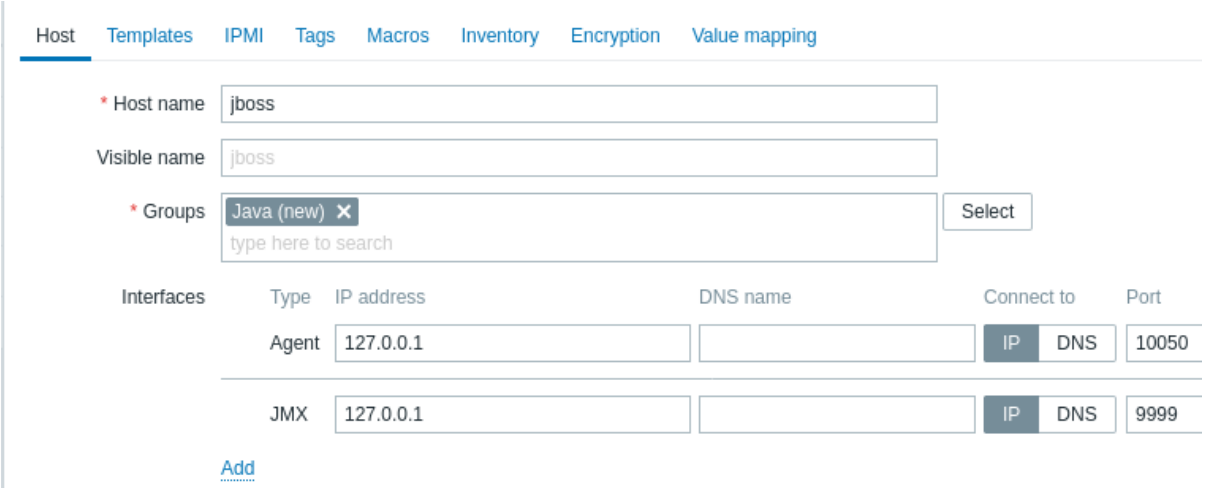
zabbix_java/settings.sh 設定ファイル（または zabbix_java_gateway.conf）にも設定します。

```
START_POLLERS=5
```

JBoss が標準の管理ポートでリッスンしていることを確認します。

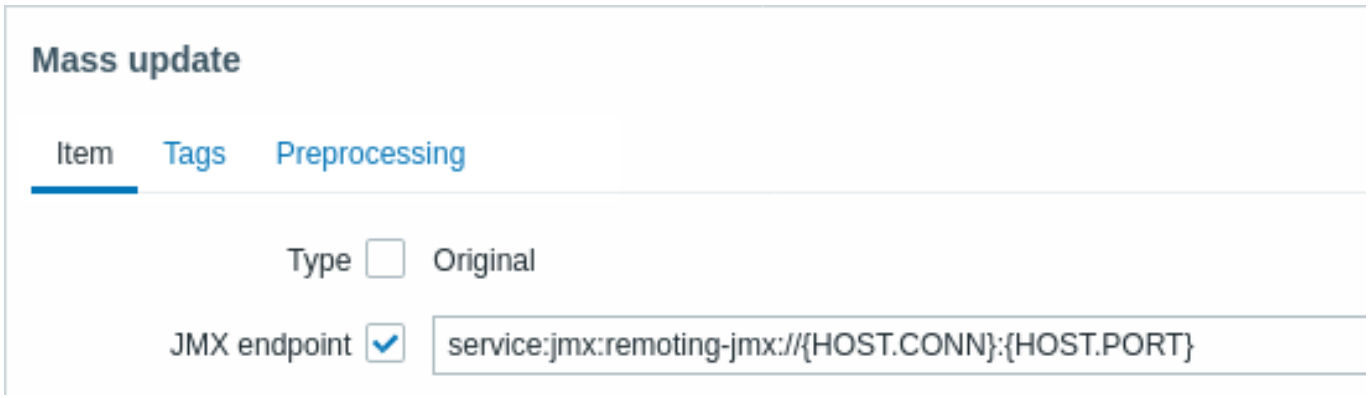
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Zabbix で JMX インターフェース 127.0.0.1:9999 のホストを作成します。



このバージョンの JBoss が RMI の代わりに JBoss Remoting プロトコルを使用していることが分かっているので、JMX テンプレート内のアイテムの JMX エンドポイントパラメータを一括更新します。

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



設定キャッシュを更新します。

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

最初にエラーが発生する場合がありますのでご注意ください。

```
3. mc [root@centos7-dev]:/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN  com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted
```

「Unsupported protocol: remoting-jmx」は、Java ゲートウェイが指定されたプロトコルで動作する方法を知らないことを意味します。これは、次の内容の ~/needed_modules.txt ファイルを作成することで修正できます。

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

次のコマンドを実行します。

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

これにより、Java ゲートウェイは jmx-remoting で動作するために必要なすべてのモジュールを持つことになります。あとは Java ゲートウェイを再起動し、しばらく待って、すべてが正しく設定されていれば、JMX 監視データが Zabbix に届き始めるはずです（参

考：最新データ)。

14 ODBC 監視

概要

ODBC 監視は、Zabbix フロントエンドの **Database monitor** アイテムタイプに対応しています。

ODBC は、データベース管理システム（DBMS）にアクセスするための C プログラミング言語のミドルウェア API です。ODBC のコンセプトは Microsoft によって開発され、その後他のプラットフォームにも移植されました。

Zabbix は、ODBC でサポートされている任意のデータベースにクエリを実行できます。そのために、Zabbix はデータベースに直接接続するのではなく、ODBC で設定された ODBC インターフェースとドライバーを使用します。これにより、さまざまな目的（たとえば、特定のデータベースキューのチェック、使用統計の取得など）で複数のデータベースをより効率的に監視できます。

Zabbix は、最も一般的に使用されているオープンソースの ODBC API 実装の 1 つである unixODBC をサポートしています。

Attention:
ODBC チェックの既知の問題も参照してください。

unixODBC のインストール

unixODBC のインストールには、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用することを推奨します。最も一般的な Linux ディストリビューションでは、unixODBC はデフォルトでパッケージリポジトリに含まれています。パッケージが利用できない場合は、unixODBC のホームページからソースファイルを取得できます: http://www.unixodbc.org/download.html。

unixODBC をインストールするには、使用しているシステムのパッケージマネージャーを使用します:

```
##### Ubuntu/Debian システムの場合:
apt install unixodbc unixodbc-dev

##### RedHat/Fedora ベースのシステムの場合:
dnf install unixODBC unixODBC-devel

##### SUSE ベースのシステムの場合:
zypper in unixODBC-devel
```

Attention:
unixodbc-dev または unixODBC-devel パッケージは、unixODBC サポート付きで Zabbix をコンパイルするために必要です。ODBC サポートを有効にするには、Zabbix を以下の設定オプションでコンパイルする必要があります:

--with-unixodbc[=ARG] # unixODBCパッケージに対してODBCドライバを使用します。

unixODBC ドライバーのインストール

監視対象のデータベースに対して unixODBC データベースドライバーをインストールする必要があります。サポートされているデータベースとドライバーの一覧については、unixODBC のホームページを参照してください: http://www.unixodbc.org/drivers.html。

Note:
一部の Linux ディストリビューションでは、データベースドライバーがパッケージリポジトリに含まれています。

MySQL

MySQL unixODBC データベースドライバをインストールするには、お使いのシステムに応じたパッケージマネージャを使用してください:

```
##### Ubuntu/Debian システムの場合:
apt install odbc-mariadb

##### RedHat/Fedora ベースのシステムの場合:
dnf install mariadb-connector-odbc

##### SUSE ベースのシステムの場合:
zypper install mariadb-connector-odbc
```

パッケージマネージャを使用せずにデータベースドライバをインストールするには、mysql-connector-odbc については MySQL ドキュメント、mariadb-connector-odbc については MariaDB ドキュメントを参照してください。

PostgreSQL

PostgreSQL unixODBC データベースドライバーをインストールするには、お使いのシステムのパッケージマネージャーを使用します:

```
##### Ubuntu/Debian システムの場合:
apt install odbc-postgresql

##### RedHat/Fedora ベースのシステムの場合:
dnf install postgresql-odbc

##### SUSE ベースのシステムの場合:
zypper install psqlODBC
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールするには、PostgreSQL ドキュメントを参照してください。

Oracle

unixODBC データベースドライバーのインストールについては、Oracle ドキュメントを参照してください。

MSSQL

Ubuntu/Debian システムに MSSQL unixODBC データベースドライバーをインストールするには、選択したシステムのパッケージマネージャーを使用します:

```
##### Ubuntu/Debian システムの場合:
apt install tdsodbc

##### RedHat/Fedora ベースのシステムの場合 (EPEL パッケージ: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### SUSE ベースのシステムの場合:
zypper install libtdsodbc0
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールするには、FreeTDS ユーザーガイド を参照してください。

unixODBC の設定

unixODBC を設定するには、`odbcinst.ini` と `odbc.ini` ファイルを編集する必要があります。これらのファイルの場所は、次のコマンドを実行して確認できます。

```
odbcinst -j
```

コマンドの結果は、次のような情報が含まれているはずです。

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
```

odbcinst.ini

`odbcinst.ini` ファイルには、インストールされている ODBC データベースドライバーが一覧表示されます。`odbcinst.ini` が存在しない場合は、手動で作成する必要があります。

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

パラメータ	説明
TEST_MYSQL	データベースドライバー名
Description	データベースドライバーの説明
Driver	データベースドライバーライブラリの場所
FileUsage	データベースドライバーがローカルファイルへのアクセスをサポートせずにデータベースサーバーへの接続をサポートするか (0)、ファイルからのデータの読み取りをサポートするか (1)、ファイルへのデータの書き込みをサポートするか (2) を決定します。

パラメータ	説明
Threading	スレッドの直列化レベル。PostgreSQL でサポートされています。 1.6 以降、ドライバーマネージャーがスレッドサポート付きでビルドされている場合は、別のドライバーエントリを追加できます。

odbc.ini

`odbc.ini` ファイルはデータソースの設定に使用されます。サポートされているパラメータの一覧はデータベースドライバによって異なることに注意してください（例えば、Oracle データベースでは Server の代わりに ServerName を使用する場合があります）。

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

パラメータ	説明
TEST_MYSQL	データソース名（DSN）。
Description	データソースの説明。
Driver	データベースドライバ名（`odbcinst.ini` で指定されたもの）。
Server	データベースサーバの IP/DNS。
User	接続用のデータベースユーザー。
Password	データベースユーザーのパスワード。
Port	データベース接続ポート。
Socket	データベース接続ソケット。
Database	データベース名。

その他の設定パラメータのオプションについては、MySQL ドキュメントを参照してください。

PostgreSQL 用の `odbc.ini` ファイルには、追加のパラメータを含めることができます:

```
[TEST_PSQL]
Description=PostgreSQL Test Database
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

パラメータ	説明
ReadOnly	データベース接続が読み取り操作（SELECT クエリ）のみを許可し、変更操作（INSERT、UPDATE、DELETE ステートメント）を制限するかどうかを指定します。データを変更せずに保持する必要があるシナリオで便利です。
Protocol	PostgreSQL バックエンドプロトコルバージョン（SSL 接続を使用している場合は無視されます）。
ShowOidColumn	SQLColumns に Object ID（OID）を含めるかどうかを指定します。
FakeOidIndex	OID に偽の一意インデックスを作成するかどうかを指定します。

パラメータ	説明
RowVersioning	他のユーザーによってデータが変更された場合に、行を更新しようとしたときにアプリケーションがそれを検出できるようにするかどうかを指定します。このパラメータを有効にすると、WHERE 句で全てのカラムを指定する必要がなくなるため、更新処理が高速化される場合があります。
ShowSystemTables	データベースドライバがシステムテーブルを SQLTables で通常のテーブルとして扱うかどうかを指定します。システムテーブルへのアクセス性を高めるために有用です。
Fetch	ドライバが SELECT ステートメントを自動的に declare cursor/fetch で処理し、100 行のキャッシュを維持するかどうかを指定します。
BoolsAsChar	Boolean 型のマッピングを制御します。 「Yes」に設定すると、Bool は SQL_CHAR にマッピングされ、それ以外の場合は SQL_BIT にマッピングされます。
SSLmode	接続の SSL モードを指定します。
ConnSettings	接続時にバックエンドに送信される追加設定。

ODBC 接続のテスト

ODBC 接続が正常に動作しているかどうかをテストするには、isql ユーティリティ（unixODBC パッケージに含まれています）を使用できます。

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Zabbix フロントエンドでのアイテムの設定

データベース監視のアイテムを設定します。

必須入力フィールドには赤いアスタリスクが付いています。

データベース監視アイテムの場合、以下を指定する必要があります。

タイプ	ここで「データベース監視」を選択します。
キー	サポートされているアイテムキーのいずれかを入力します: **db.odbc.select[]** - このアイテムは SQL クエリ結果の最初の行の最初の列の値を返します; **db.odbc.get[]** - このアイテムは複数の行/列を JSON 形式で返します; **db.odbc.discovery[]** - このアイテムはローレベルディスカバリーデータを返します。
ユーザー名	データベースのユーザー名（255 文字まで）を入力します。 このパラメータは、odbc.ini ファイルでデータベースのユーザー名が指定されている場合は省略可能です。 接続文字列が使用されていて、ユーザー名フィールドが空でない場合は、UID=<user> として接続文字列に追加されます。
パスワード	データベースのユーザーパスワード（255 文字まで）を入力します。 このパラメータは、odbc.ini ファイルでパスワードが指定されている場合は省略可能です。 接続文字列が使用されていて、パスワードフィールドが空でない場合は、PWD=<password> として接続文字列に追加されます。 このフィールドでは特殊文字もサポートされています。 パスワードは、たとえば UID=<username>;PWD=P?;)*word のように、ユーザー名の後に接続文字列へ追加されます。 結果の文字列をテストするには、次のコマンドを実行できます: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL クエリ	SQL クエリを入力します。 db.odbc.select[] の場合、クエリは 1 つの値のみを返す必要があることに注意してください。
情報の型	ここでクエリによって返される情報の型を選択します。 情報の型が正しく選択されていない場合、アイテムはサポート対象外になります。

重要な注意事項

- サーバーまたはプロキシの設定で *odbc poller* プロセスが開始されていない場合、データベース監視アイテムはサポートされなくなります。ODBC ポーラーを有効にするには、Zabbix サーバーの設定ファイル、またはプロキシによって実行されるチェックの場合は Zabbix プロキシの設定ファイルで `StartODBCPollers` パラメータを設定します。
- アイテム設定フォームの *Timeout* パラメータ値は、ODBC ログインタイムアウトおよびクエリ実行タイムアウトとして使用されます。インストールされている ODBC ドライバーがこれらのタイムアウト設定をサポートしていない場合、これらのタイムアウト設定が無視されることがあるので注意してください。
- SQL コマンドは、`select` ステートメントを使用するクエリのように、結果セットを返す必要があります。クエリの構文は、それらを処理する RDBMS によって異なります。ストアドプロシージャへのリクエストの構文は、`call` キーワードで始める必要があります。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

db.odbc.select[<unique short description>,<dsn>,<connection string>]

 SQL クエリ結果の最初の行の最初の列、つまり 1 つの値を返します。
 戻り値: SQL クエリに依存します。

パラメータ:

- **unique short description** - アイテムを識別するための一意の短い説明 (トリガーなどで使用);
- **dsn** - データソース名 (`odbc.ini` で指定されたもの);
- **connection string** - 接続文字列 (ドライバー固有の引数を含めることができます)。

コメント:

- `dsn` と `connection string` は省略可能なパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合は、`dsn` は無視されます。
- クエリが複数の列を返す場合、最初の列のみが読み取られます。クエリが複数の行を返す場合、最初の行のみが読み取られます。

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 SQL クエリの結果を JSON 配列に変換します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **unique short description** - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（`odbc.ini` で指定）;
- **connection string** - 接続文字列（ドライバ固有の引数を含む場合があります)。

コメント:

- `dsn` と `connection string` はどちらも省略可能なパラメータですが、少なくともどちらか一方が必要です。両方が定義されている場合は、`dsn` は無視されます。
- JSON 形式で複数の行/列が返される場合があります。このアイテムは、1 回のシステムコールですべてのデータを収集するマスターアイテムとして使用でき、依存アイテムでは JSONPath の前処理を使用して個々の値を抽出できます。返される形式の例については、低レベルディスカバリで使用されるものを参照してください。

例:

```
# MySQL ODBCドライバ5用の接続:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<unique short description>,<dsn>,<connection string>]

 SQL クエリの結果を JSON 配列に変換し、ローレベルディスカバリで使用します。クエリ結果のカラム名は、ディスカバリされたフィールド値とペアになったローレベルディスカバリマクロ名に変換されます。これらのマクロは、アイテム、トリガーなどのプロトタイプ作成時に使用できます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **unique short description** - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（`odbc.ini` で指定）;
- **connection string** - 接続文字列（ドライバ固有の引数を含めることができます)。

コメント:

- `dsn` と `connection string` はどちらも省略可能なパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合は、`dsn` は無視されます。

エラーメッセージ

ODBC エラーメッセージは、詳細な情報を提供するためにフィールドに構造化されています。例えば、エラーメッセージは次のようになります。

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "Cannot execute ODBC query" - Zabbix メッセージ
- "[SQL_ERROR]" - ODBC リターンコード
- "[42601]" - SQLState
- "[7]" - ネイティブエラーコード
- "[ERROR: syntax error at or near ";"; Error while executing the query]" - ネイティブエラーメッセージ

エラーメッセージの長さは 2048 バイトに制限されているため、メッセージが切り捨てられる場合があることに注意してください。ODBC 診断レコードが複数ある場合、Zabbix は長さ制限の範囲内でそれらを（| で区切って）連結しようとします。

15 依存アイテム

概要

1 つのアイテムが一度に複数のメトリックを収集する場合や、関連するメトリックを同時に収集した方が理にかなっている場合があります。例えば:

- 個々のコアの CPU 使用率
- 受信/送信/合計ネットワークトラフィック

一括メトリック収集と複数の関連アイテムでの同時利用を可能にするために、Zabbix は従属アイテムをサポートしています。従属アイテムは、データを同時に 1 つのクエリで収集するマスターアイテムに依存します。マスターアイテムに新しい値が入ると、自動的に従属アイテムの値も設定されます。従属アイテムは、マスターアイテムと異なる更新間隔を持つことはできません。

Zabbix の前処理オプションを使用して、マスターアイテムのデータから従属アイテムに必要な部分を抽出できます。

前処理は `preprocessing manager` プロセスによって管理され、ワーカースレッドが前処理ステップを実行します。前処理付きのすべての値は、異なるデータ収集プロセスから受信され、ヒストリキャッシュに追加される前に前処理マネージャを通過します。データ収集プロセス（ポーラー、トラッパーなど）と前処理プロセス間の通信には、ソケットベースの IPC 通信が使用されます。

Zabbix サーバーまたは Zabbix プロキシ（ホストがプロキシで監視されている場合）が前処理ステップと従属アイテムの処理を実行します。

どのタイプのアイテムでも、従属アイテムであっても、マスターアイテムとして設定できます。既存の従属アイテムの値からさらに小さな部分を抽出するために、追加の従属アイテム階層を使用することもできます。

制限事項

- 同じホスト（テンプレート）間の依存関係のみ許可されます
- アイテムプロトタイプは、同じホストの他のアイテムプロトタイプまたは通常のアイテムに依存できます
- テンプレートからのマスターアイテムを持つホスト上の従属アイテムは、XML にエクスポートされません

アイテムの設定

従属アイテムは、データを取得するためにマスターアイテムに依存します。そのため、マスターアイテムを最初に設定（または作成）する必要があります。

- データ収集 → ホストに移動します
- ホストの行で アイテムをクリックします
- アイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します

必須入力フィールドは赤いアスタリスクでマークされています。

追加をクリックしてマスターアイテムを保存します。

その後、従属アイテムを設定できます。

必須入力フィールドは赤いアスタリスクでマークされています。

従属アイテムに特有の情報が必要なフィールドは次のとおりです。

タイプ	ここで従属アイテムを選択します。
キー	アイテムを識別するために使用されるキーを入力します。
マスターアイテム	マスターアイテムを選択します。マスターアイテムの値が従属アイテムの値を設定するために使用されます。
情報のタイプ	保存されるデータの形式に対応する情報のタイプを選択します。

アイテム値の事前処理を使用して、マスターアイテム値から必要な部分を抽出できます。

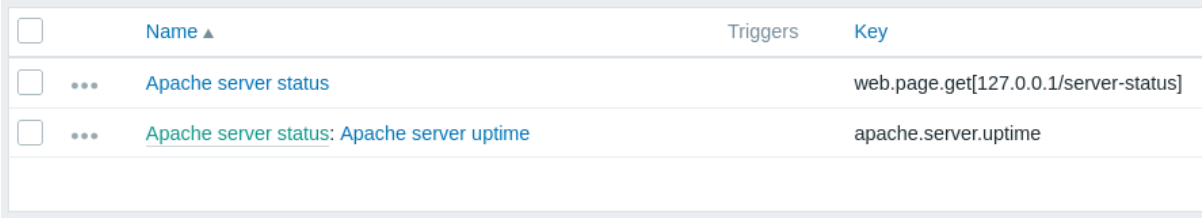
事前処理を行わない場合、従属アイテムの値はマスターアイテムの値とまったく同じになります。

追加をクリックして従属アイテムを保存します。

従属アイテムをより迅速に作成するためのショートカットは、アイテムリストでボタンをクリックし、従属アイテムの作成を選択することで利用できます。

表示

アイテムリストでは、依存アイテムはマスターアイテム名をプレフィックスとして表示されます。



マスターアイテムが削除された場合、そのアイテムに依存する依存アイテムもすべて削除されます。

16 HTTP エージェント

概要

このアイテムタイプでは、HTTP/HTTPS プロトコルを使用したデータのポーリングが可能です。トラッピングも、Zabbix senderユーティリティや Zabbix senderプロトコル（Zabbix サーバーまたはプロキシへのデータ送信）、または`history.push`API メソッド（Zabbix サーバーへのデータ送信）を使用して可能です。

HTTP アイテムチェックは Zabbix サーバーによって実行されます。ただし、ホストが Zabbix プロキシによって監視されている場合、HTTP アイテムチェックはプロキシによって実行されます。

HTTP アイテムチェックは、監視対象ホストでエージェントを実行する必要はありません。

HTTP エージェントは、HTTP と HTTPS の両方をサポートします。Zabbix は、リダイレクトをオプションで追跡します（下記のリダイレクトを追跡オプションを参照）。リダイレクトの最大数は 10 にハードコードされています（cURL オプション CURLOPT_MAXREDIRS を使用）。

> **Attention:**
> Zabbix サーバー/プロキシは、最初に cURL（libcurl）サポートで構成する必要があります。

HTTP チェックは非同期で実行されます。1 つのリクエストの応答を受信する前に他のチェックを開始する必要はありません。DNS 解決も非同期です。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPollerで定義）。

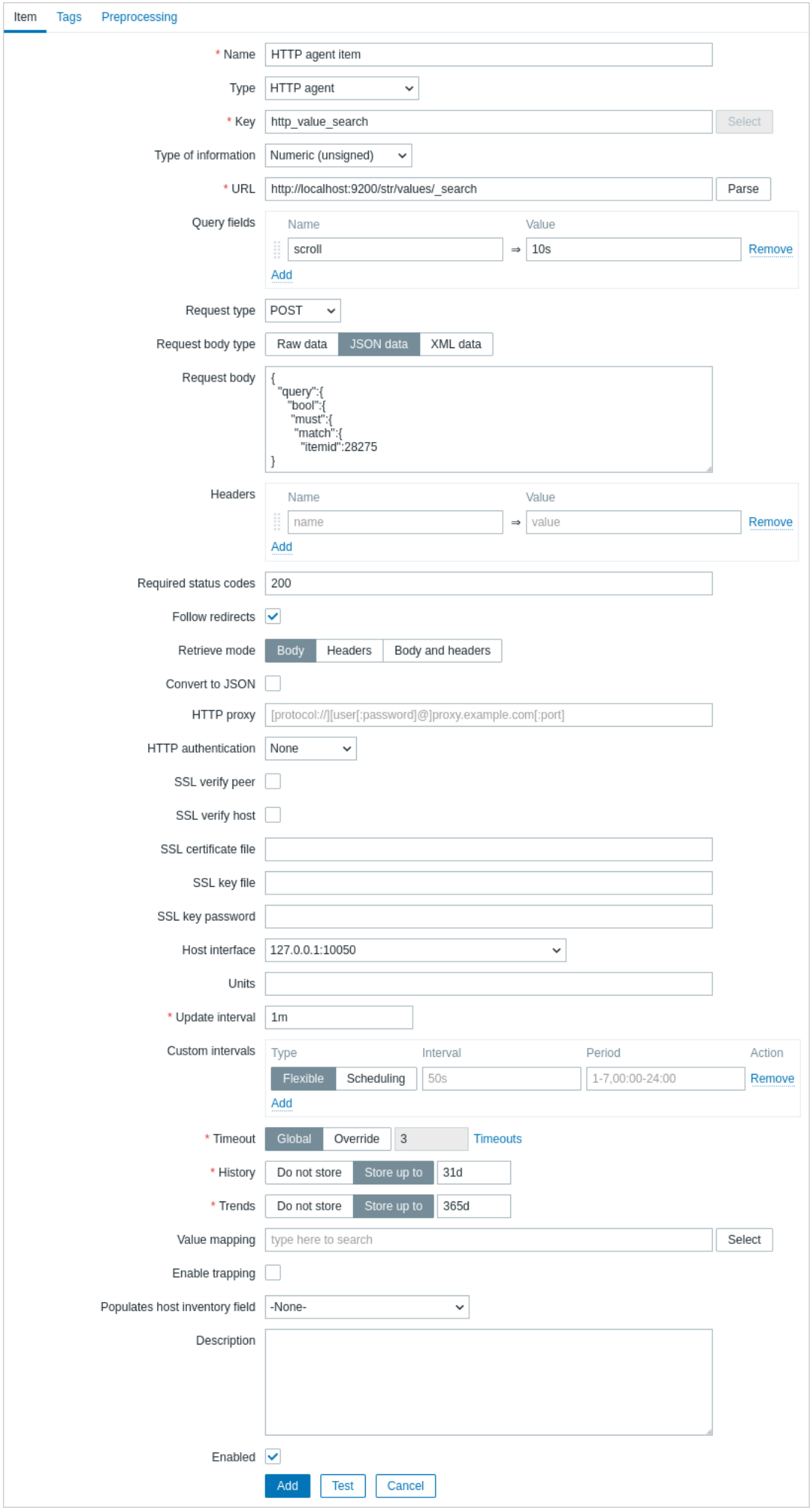
非同期 HTTP エージェントポーラーの数は、StartHTTPAgentPollersパラメータで定義されます。

cURL の永続的な接続機能は、Zabbix 7.0 以降の HTTP エージェントチェックに追加されました。

設定

HTTP アイテムを設定するには:

- データ収集 → ホストに移動します
- ホストの行で アイテムをクリックします
- アイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します


Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
⇒
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid":28275
}
Headers
Name
Value
name
⇒
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Action
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

すべての必須入力フィールドには赤いアスタリスクが付いています。

HTTP アイテムに特有の情報が必要なフィールドは次のとおりです:

パラメータ	説明
タイプ	ここで **HTTP** エージェントを選択します。
キー	一意のアイテムキーを入力します。
URL	接続してデータを取得するための URL。例: https://www.example.com http://www.example.com/download ドメイン名は Unicode 文字で指定できます。HTTP チェックを実行する際に自動的に punycode で ASCII に変換されます。 解析ボタンを使用すると、URL からオプションのクエリフィールド（例: ?name=Admin&password=mypassword）を分離し、属性と値をクエリフィールドに移動して自動的に URL エンコードできます。 2048 文字まで。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_URL cURL オプションを設定します。
クエリフィールド	URL の変数（上記参照）。 属性と値のペアとして指定します。 値は自動的に URL エンコードされます。マクロからの値は解決され、その後自動的に URL エンコードされます。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_URL cURL オプションを設定します。
リクエストタイプ	リクエストメソッドタイプを選択します: *GET*, *POST*, *PUT* または *HEAD*
リクエストボディタイプ	リクエストボディタイプを選択します: 生データ - カスタム HTTP リクエストボディ、マクロは置換されますがエンコードは行われません **JSON** データ - JSON 形式の HTTP リクエストボディ。マクロは文字列、数値、true、false として使用できます。文字列として使用する場合はダブルクォートで囲む必要があります。マクロからの値は解決され、その後自動的にエスケープされます。ヘッダーで”Content-Type” が指定されていない場合は、デフォルトで”Content-Type: application/json” になります。 **XML** データ - XML 形式の HTTP リクエストボディ。マクロはテキストノード、属性、または CDATA セクションとして使用できます。マクロからの値はテキストノードと属性で解決され、その後自動的にエスケープされます。ヘッダーで”Content-Type” が指定されていない場合は、デフォルトで”Content-Type: application/xml” になります。 注: *XML* データを選択するには libxml2 が必要です。
リクエストボディ	リクエストボディを入力します。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。
ヘッダー	リクエストを実行する際に送信されるカスタム HTTP ヘッダー。 属性と値のペアとして指定します。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_HTTPHEADER cURL オプションを設定します。
必要なステータスコード	期待される HTTP ステータスコードのリスト。Zabbix がリストにないコードを取得した場合、アイテムはサポート対象外になります。空の場合はチェックされません。 例: 200,201,210-299 リスト内でサポートされているマクロ: ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLINFO_RESPONSE_CODE cURL オプションを使用します。
リダイレクトを追跡	HTTP リダイレクトを追跡するにはチェックボックスをオンにします。 これはCURLOPT_FOLLOWLOCATION cURL オプションを設定します。
取得モード	取得するレスポンスの部分を選択します: ボディ - ボディのみ ヘッダー - ヘッダーのみ ボディとヘッダー - ボディとヘッダー

パラメータ	説明

JSON に変換	取得モードがボディ以外の場合、ヘッダーは”header” キーの下で属性と値のペアとして保存されます。
	’Content-Type: application/json’ が検出された場合、ボディはオブジェクトとして保存され、それ以外の場合は文字列として保存されます。例:
	`{`
	`  "header": {`
	`  "<key>": "<value>",`
	`  "<key2>": "<value>"`
	`  },`
	`  "body": <body>`
	`}`

HTTP プロキシ	使用する HTTP プロキシを `[protocol://][username[:password]@]proxy.example.com[:port]` の形式で指定できます。 オプションの `protocol://`プレフィックスを使用して、代替プロキシプロトコル（例: https, socks4, socks5; ドキュメント参照; プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました）を指定できます。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。間違ったプロトコルを指定すると、接続に失敗し、アイテムはサポート対象外になります。 デフォルトでは、1080 ポートが使用されます。 指定した場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連の環境変数を上書きします。指定しない場合、プロキシはプロキシ関連の環境変数を上書きしません。入力された値はそのまま渡され、整合性チェックは行われません。 注: HTTP プロキシでサポートされるのはシンプル認証のみです。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_PROXY cURL オプションを設定します。
HTTP 認証	認証オプションを選択します: なし - 認証を使用しません; **Basic** - Basic 認証を使用します; **NTLM** - NTLM (Windows NT LAN Manager) 認証を使用します; **Kerberos** - Kerberos 認証を使用します（詳細はKerberos の Zabbix での設定も参照してください）; **Digest** - Digest 認証を使用します。 これはCURLOPT_HTTPAUTH cURL オプションを設定します。
ユーザー名	ユーザー名を入力します（255 文字まで）。 *HTTP* 認証が Basic、NTLM、Kerberos、Digest に設定されている場合に利用できます。ユーザーマクロとローレベルディスカバリーマクロがサポートされています。
パスワード	ユーザーパスワードを入力します（255 文字まで）。 *HTTP* 認証が Basic、NTLM、Kerberos、Digest に設定されている場合に利用できます。ユーザーマクロとローレベルディスカバリーマクロがサポートされています。
SSL ピア検証	Web サーバーの SSL 証明書を検証するにはチェックボックスをオンにします。サーバー証明書はシステム全体の証明書認証局（CA）ロケーションから自動的に取得されます。CA ファイルの場所は Zabbix サーバーまたはプロキシの設定パラメータ SSLCALocation で上書きできます。 これはCURLOPT_SSL_VERIFYPEER cURL オプションを設定します。
SSL ホスト検証	Web サーバー証明書の Common Name フィールドまたは Subject Alternate Name フィールドが一致するかどうかを検証するにはチェックボックスをオンにします。 これはCURLOPT_SSL_VERIFYHOST cURL オプションを設定します。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイル名。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、SSL キーのファイルフィールドを空にします。キーが暗号化されている場合は、SSL キーパスワードフィールドにパスワードを指定します。 このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLCertLocation で指定します。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_SSLCERT cURL オプションを設定します。

パラメータ	説明
SSL キーファイル	クライアント認証に使用する SSL 秘密鍵ファイル名。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLKeyLocation で指定します。 サポートされているマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_SSLKEY cURL オプションを設定します。
SSL キーパスワード	SSL 秘密鍵ファイルのパスワード。 サポートされているマクロ: ユーザーマクロ, ローレベルディスカバリーマクロ。 これはCURLOPT_KEYPASSWD cURL オプションを設定します。
タイムアウト	Zabbix は URL の処理に設定された時間（1-600 秒）以上は費やしません。実際には、このパラメータは URL への接続の最大時間と HTTP リクエストの実行の最大時間を定義します。したがって、Zabbix は 1 回のチェックで 2 x タイムアウト秒以上は費やしません。 これはCURLOPT_TIMEOUT cURL オプションを設定します。 タイムアウトパラメータの詳細については、一般的なアイテム属性を参照してください。
トラッピングを有効化	このチェックボックスをオンにすると、アイテムはトラッパーアイテムとしても機能し、Zabbix senderユーティリティまたは Zabbix senderプロトコルを使用して Zabbix サーバーまたはプロキシに送信されたデータ、または`history.push`API メソッドを使用して Zabbix サーバーに送信されたデータを受け入れます。データ送信の詳細については、トラッパーアイテムを参照してください。
許可されたホスト	トラッピングを有効化チェックボックスがオンの場合のみ表示されます。 カンマ区切りの IP アドレス（オプションで CIDR 表記）、または DNS 名のリスト。 指定した場合、ここにリストされたホストからの着信接続のみが受け入れられます。 IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。 "IPv4 互換 IPv6 アドレス"（0000::/96 プレフィックス）はサポートされていますが、RFC4291で非推奨とされています。 例: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} このフィールドではスペースとユーザーマクロが使用できます。 このフィールドではホストマクロ: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} が使用できます。

Note:
HTTP プロキシフィールドを空のままにした場合、HTTP プロキシを使用するもう 1 つの方法は、プロキシ関連の環境変数を設定することです。
HTTP の場合 - Zabbix サーバーユーザーの `http_proxy` 環境変数を設定します。例:
`http_proxy=http://proxy_ip:proxy_port`
HTTPS の場合 - `HTTPS_PROXY` 環境変数を設定します。例:
`HTTPS_PROXY=http://proxy_ip:proxy_port`。詳細はシェルコマンド: *# man curl* を実行して確認できます。

Attention:
[1] Zabbix は PEM 形式のみの証明書および秘密鍵ファイルをサポートします。証明書および秘密鍵データが PKCS #12 形式ファイル（通常は *.p12 または *.pfx 拡張子）にある場合は、次のコマンドを使用して PEM ファイルを生成できます:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

例

例 1

Elasticsearch のようなサービスからデータを取得するために、単純な GET リクエストを送信します。

- URL が `localhost:9200/?pretty` の GET アイテムを作成します
- レスポンスを確認します:

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
```

```
        "number" : "6.1.3",
        "build_hash" : "af51318",
        "build_date" : "2018-01-26T18:22:55.523Z",
        "build_snapshot" : false,
        "lucene_version" : "7.1.0",
        "minimum_wire_compatibility_version" : "5.6.0",
        "minimum_index_compatibility_version" : "5.0.0"
      },
      "tagline" : "You know, for search"
    }
```

- 次に、JSONPath の前処理ステップを使用してバージョン番号を抽出します: `$.version.number`

例 2

Elasticsearch などのサービスからデータを取得するために、シンプルな POST リクエストを送信します:

- URL に対して POST アイテムを作成します: `http://localhost:9200/str/values/_search?scroll=10s`
- プロセッサの負荷 (1 分平均/コア) を取得するために、以下の POST ボディを設定します

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- 受信データ:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
    "_shards": {
        "total": 5,
        "successful": 5,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 1,
        "max_score": 1.0,
        "hits": [{
            "_index": "dbl",
            "_type": "values",
            "_id": "dqX9VWEBV6sEKSMyk6sw",
            "_score": 1.0,
            "_source": {
                "itemid": 28275,
                "value": "0.138750",
                "clock": 1517566136,
                "ns": 25388713,
                "ttl": 604800
```

```
                }
            }]
        }
    }
```

- JSONPath の事前処理ステップを使用してアイテム値を取得します: `$.hits.hits[0]._source.value`

例 3

Zabbix API が生きているかどうかを確認するには、apiinfo.versionを使用します。

- アイテムの設定:

POST メソッドで JSON データを使用し、リクエストヘッダーを設定し、ヘッダーのみを返すように指定している点に注意してください。

- HTTP コードを取得するための正規表現によるアイテム値の事前処理:
- 最新データでの結果の確認:

例 4

Openweathermap のパブリックサービスに接続して天気情報を取得します。

- 1 つの JSON で一括データ収集のためのマスターアイテムを設定します。

クエリフィールドでのマクロの使用に注意してください。これらの入力方法についてはOpenweathermap APIを参照してください。

HTTP エージェントの応答で返されるサンプル JSON:


```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
```

```
    }
}
```

次の作業は、JSON からデータを抽出する従属アイテムを設定することです。

- 湿度のサンプル従属アイテムを設定します。

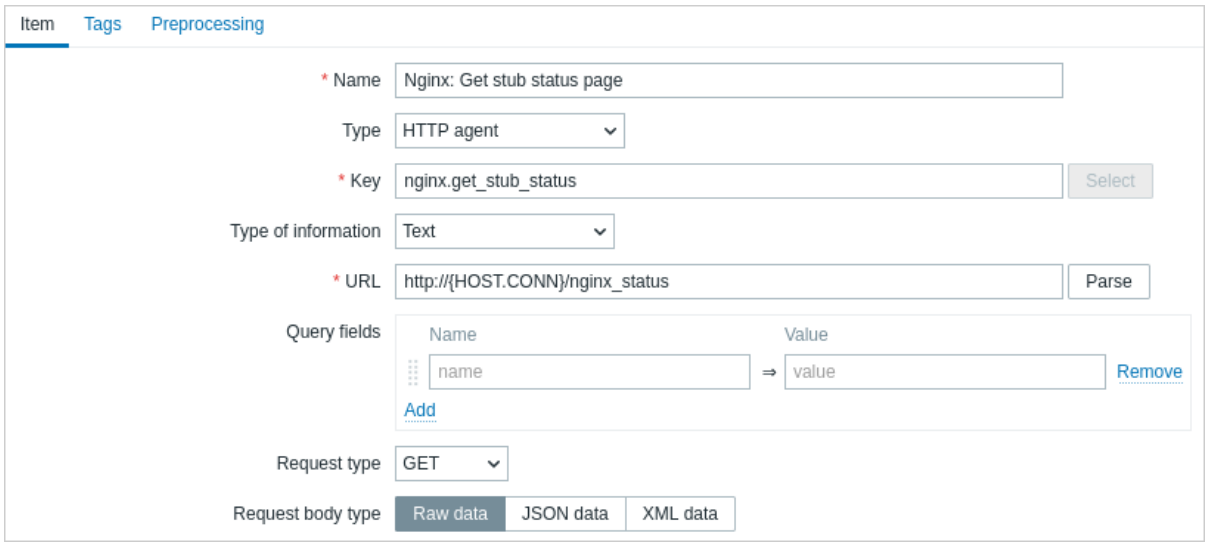
「温度」などの他の天気指標も同様に追加します。

- JSONPath によるサンプル従属アイテムの値の前処理:
- 最新データで天気データの結果を確認します。

例 5

Nginx のステータスページに接続し、そのメトリクスを一括で取得します。

- 公式ガイドに従って Nginx を設定します。
- 一括データ収集用のマスターアイテムを設定します:

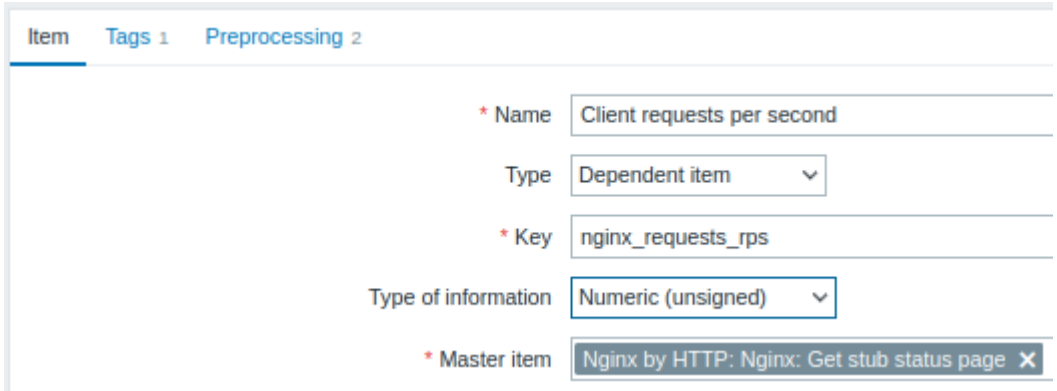


Nginx stub status の出力例:

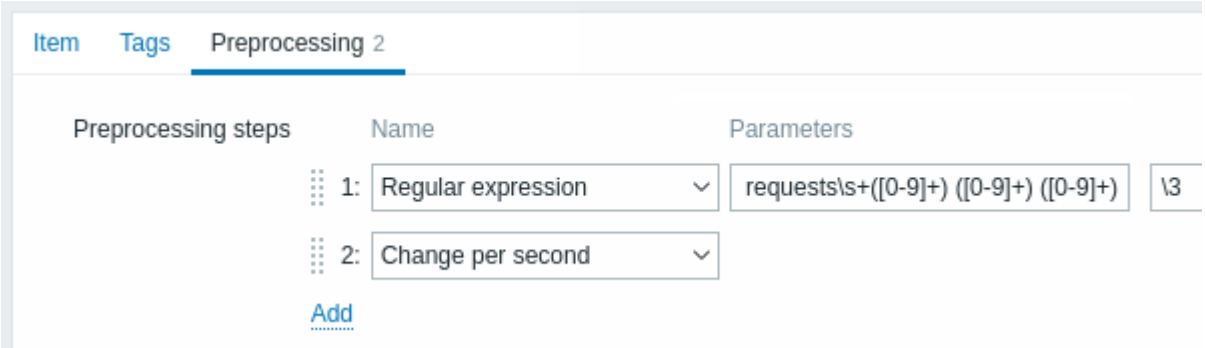
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

次の作業は、データを抽出する従属アイテムを設定することです。

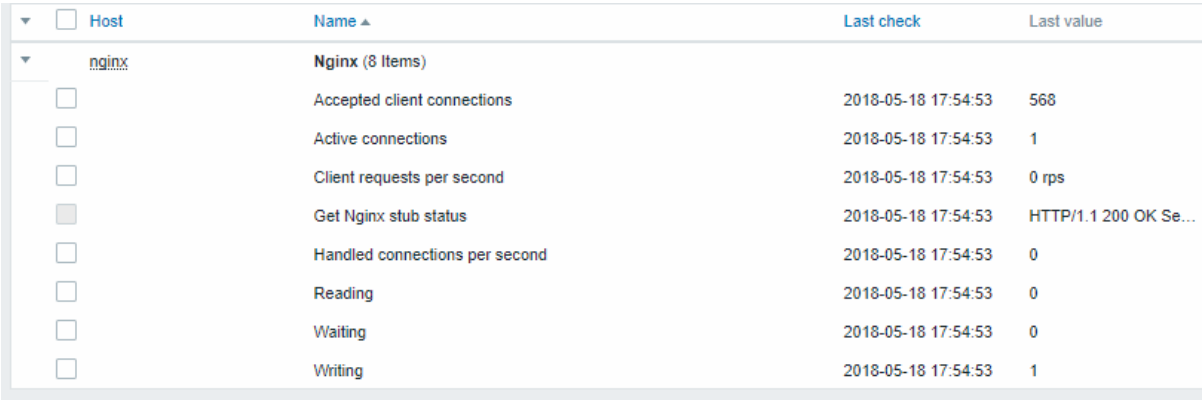
- 秒あたりのリクエスト数のサンプル従属アイテムを設定します:



- 正規表現 `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)` を使用した従属アイテム値の前処理例:



- 最新データで stub モジュールからの完全な結果を確認します:



17 Prometheus チェック

概要

Zabbix は、Prometheus のラインフォーマットで公開されたメトリクスを取得できます。

Prometheus データの収集を開始するには、2 つのステップが必要です。

- 適切なデータエンドポイント (例: `https://<prometheus host>/metrics`) を指すHTTP マスターアイテム
- マスターアイテムで収集したメトリクスから必要なデータを取得するための Prometheus 前処理オプションを使用する従属アイテム

Prometheus データの前処理オプションは 2 つあります。

- *Prometheus* パターン - 通常のアイテムで Prometheus データを取得するために使用
- *Prometheus to JSON* - 通常のアイテムおよびローレベルディスカバリで使用。この場合、取得した Prometheus データは JSON 形式で返されます。

バルクプロセッシング

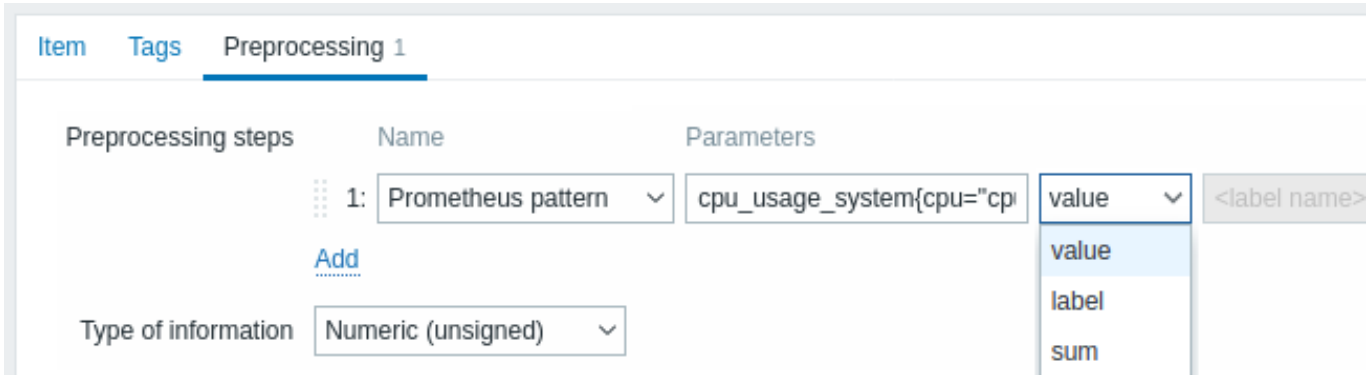
依存 item はバルクプロセッシングに対応しています。
 キャッシュとインデックスを有効にするには、*Prometheus pattern* のプリプロセスを 最初のプリプロセスステップにする必要があります。
 Prometheus pattern が最初のプリプロセスステップである場合、パースされた Prometheus データは、
 Prometheus pattern プリプロセスステップの最初の `<label>==<value>` 条件によってキャッシュされ、インデックスが作成されます。
 このキャッシュは、このバッチ内の他の依存 item を処理する際に再利用されます。
 最適なパフォーマンスを得るために、最初のラベルは最も異なる値を持つものにすべきです。

最初のステップの前に他のプリプロセスがある場合は、マスター item か、従属 item のマスター item として使用される
 新しい従属 item のどちらかに移動させる必要があります。

設定

HTTP マスターアイテムが設定されている場合は、Prometheus の前処理ステップを使用する従属アイテムを作成する必要があります。

- 設定フォームで一般的な従属アイテムのパラメータを入力します
- 前処理タブに移動します
- Prometheus の前処理オプション（*Prometheus* パターンまたは *Prometheus to JSON*）を選択します



Prometheus パターン前処理オプションに固有のパラメータは以下の通りです。

パラメータ	説明	例
パターン	必要なデータパターンを定義するために、Prometheus クエリ言語に似たクエリ言語を使用できます（比較表を参照）、例： <metric name> - メトリック名で選択 {__name__=”<metric name>”} - メトリック名で選択 {__name__=~”<regex>”} - 正規表現に一致するメトリック名で選択 {<label name>=”<label value>”,...} - ラベル名で選択 {<label name>=~”<regex>”,...} - 正規表現に一致するラベル名で選択 {__name__=~”.*”}==<value> - メトリック値で選択 または上記の組み合わせ： <metric name>{<label1 name>=”<label1 value>”,<label2 name>=~”<regex>”,...}==<value> ラベル値は任意の UTF-8 文字列ですが、バックスラッシュ、ダブルクォート、改行文字はそれぞれ\\、\"、\n としてエスケープする必要があります。他の文字はエスケープしないでください。	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu=”cpu-total”}* *cpu_usage_system{cpu=~”.*”}* *cpu_usage_system{cpu=”cpu-total”,host=~”.*”}* *wmi_service_state{name=”dhcp”}==1* *wmi_os_timezone{timezone=~”.*”}==1*
結果処理	値を返すか、ラベルを返すか、または適切な関数を適用するか（パターンが複数行に一致し、結果を集約する必要がある場合）： **value** - メトリック値を返す（複数行に一致した場合はエラー） **label** - ラベルフィールドで指定したラベルの値を返す（複数のメトリックに一致した場合はエラー） **sum** - 値の合計を返す **min** - 最小値を返す **max** - 最大値を返す **avg** - 平均値を返す **count** - 値の数を返す このフィールドは *Prometheus* パターンオプションでのみ利用可能です。	下記のパラメータ使用例も参照してください。
出力	ラベル名を定義します（オプション）。この場合、ラベル名に対応する値が返されます。 このフィールドは、結果処理フィールドで「ラベル」が選択されている場合にのみ、*Prometheus* パターンオプションで利用可能です。	

パラメータ使用例

1. 最も一般的なユースケースは値を返すことです。以下から/var/db の値を返すには：

node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09
 node_disk_usage_bytes{path="/var/db"} 20480
 node_disk_usage_bytes{path="/var/dpkg"} 8192
 node_disk_usage_bytes{path="/var/empty"} 4096

次のパラメータを使用します：

- パターン - node_disk_usage_bytes{path="/var/db"}
- 結果処理 - 「値」を選択

2. すべての node_disk_usage_bytes パラメータの平均値に興味がある場合：

- パターン - node_disk_usage_bytes
- 結果処理 - 「平均」を選択

3. Prometheus は数値データのみをサポートしていますが、関連するテキストの説明も返すための回避策が一般的に使用されています。これはフィルタとラベルの指定で実現できます。したがって、以下から「color」ラベルの値を返すには

elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1
 elasticsearch_cluster 0
 elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0

次のパラメータを使用します：

- パターン - elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1
- 結果処理 - 「ラベル」を選択
- ラベル - 「color」と指定

フィルタ（数値「1」に基づく）は適切な行に一致し、ラベルはヘルスステータスの説明（現在は「green」、ただし将来的には「red」や「yellow」もあり得る）を返します。

Prometheus から JSON へ

Prometheus からのデータローレベルディスカバリーに使用することができます。この場合、JSON 形式のデータが必要で、
 Prometheus to JSON プリプロセッシングオプションはまさにそのデータを返します。

詳しくは、Discovery using Prometheus dataを参照してください。

クエリ言語の比較

次の表は、PromQL と Zabbix Prometheus 前処理クエリ言語の違いと類似点を示しています。

PromQL インスタントベクターセレクター		Zabbix Prometheus 前処理
違い		
クエリターゲット	Prometheus サーバー	Prometheus エクスポート形式のプレーンテキスト
戻り値	インスタントベクター	メトリックまたはラベル値（Prometheus パターン） JSON での単一値のメトリック配列（Prometheus to JSON）
ラベルマッチング演算子	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
ラベルまたはメトリック名のマッチングに使用される正規表現	RE2	PCRE
比較演算子	リストを参照	値フィルタリングには **==**（等しい）のみサポート
類似点		
文字列と等しいメトリック名での選択	<metric name> または {__name__="<metric name>"}	<metric name> または {__name__="<metric name>"}
正規表現に一致するメトリック名での選択	{__name__=~"<regex>"}	{__name__=~"<regex>"}
文字列と等しい <label name> 値での選択	{<label name>="<label value>",...}	{<label name>="<label value>",...}
正規表現に一致する <label name> 値での選択	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
値が文字列と等しい場合の選択	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

18 スクリプト

概要

スクリプトアイテムは、ユーザー定義の JavaScript コードを実行してデータを収集するために使用でき、HTTP/HTTPS 経由でデータを取得することも可能です。スクリプトに加えて、オプションでパラメータのリスト（名前と値のペア）やタイムアウトを指定できます。

このアイテムタイプは、複数のステップや複雑なロジックを必要とするデータ収集シナリオで役立ちます。例えば、スクリプトアイテムを設定して HTTP コールを実行し、最初のステップで受信したデータを何らかの方法で処理し、変換した値を 2 回目の HTTP コールに渡すことができます。

スクリプトアイテムは、Zabbix サーバーまたはプロキシのポーラーによって処理されます。

設定

アイテム設定フォームのタイプフィールドで「スクリプト」を選択し、必要なフィールドに入力します。

script_item.png

すべての必須入力フィールドには赤いアスタリスクが付いています。

スクリプトアイテムに固有の情報が必要なフィールドは次のとおりです。

フィールド	説明
キー	アイテムを識別するために使用される一意のキーを入力します。
パラメータ	スクリプトに渡す変数を属性と値のペアで指定します。 ユーザーマクロがサポートされています。サポートされている組み込みマクロを確認するには、サポートされているマクロテーブルで「Script-type item」を検索してください。
スクリプト	パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。このコードはメトリック値を返すためのロジックを提供する必要があります。 このコードは、すべてのパラメータおよび Zabbix によって追加された追加の JavaScript オブジェクトにアクセスできます。 参考: JavaScript ガイド。
タイムアウト	JavaScript の実行タイムアウト（1～600 秒；これを超えるとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでに時間がかかる場合があります。 タイムアウトパラメータの詳細については、一般的なアイテム属性を参照してください。

例

シンプルなデータ収集

https://www.example.com/release_notes の内容を収集します:

- タイプが「スクリプト」のアイテムを作成します。
- スクリプトフィールドに以下を入力します:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

パラメータを使用したデータ収集

特定のページの内容を収集し、パラメータを利用します:

- タイプ「スクリプト」と 2 つのパラメータを持つアイテムを作成します:
  - **url : {$DOMAIN}**（ユーザーマクロ {$DOMAIN} は、できればホストレベルで定義してください）
  - **subpage : /release_notes**
- スクリプトフィールドに以下を入力します:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

複数の HTTP リクエスト

https://www.example.com と *https://www.example.com/release_notes* の両方の内容を収集します。

- タイプが「スクリプト」のアイテムを作成します。
- スクリプトフィールドに以下を入力します。

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

ロギング

Zabbix server ログに”Log test” エントリを追加し、アイテム値”1” を返信します。

- タイプ”Script” の item を作成します。
- *Script* フィールドに、以下のコードを入力します。

```
Zabbix.log(3, 'Log test');
return 1;
```

19 ブラウザアイテム

概要

ブラウザーアイテムを使用すると、ブラウザーを使用して複雑な Web サイトや Web アプリケーションを監視できます。

> **Attention:**
> ブラウザーアイテムのサポートは現在実験的なものです。

ブラウザーアイテムは、ユーザー定義の JavaScript コードを実行し、HTTP/HTTPS 経由でデータを取得することでデータを収集します。このアイテムは、クリック、テキスト入力、Web ページのナビゲーション、その他の Web サイトや Web アプリケーションでのユーザー操作など、ブラウザー関連のアクションをシミュレートできます。

スクリプトに加えて、オプションでパラメーターのリスト（名前と値のペア）やタイムアウトを指定できます。

> **Attention:**
> このアイテムは、W3C WebDriver 標準を部分的に実装しており、Selenium Server またはプレーンな WebDriver（例：ChromeDriver）を Web テストエンドポイントとして使用します。アイテムを動作させるには、Zabbix サーバー/プロキシの設定ファイルで `WebDriverURL` パラメーターにエンドポイントを設定してください（ChromeDriver を使用する場合は、セキュリティに関する考慮事項を参照）。パフォーマンス向上のため、Web テスト環境用に専用サーバーの使用を検討してください。

ブラウザーアイテムのチェックは、Zabbix サーバーまたはプロキシのブラウザーポーラーによって実行および処理されます。必要に応じて、Zabbix サーバー/プロキシの設定ファイルで `StartBrowserPollers` パラメーターを使用して、ブラウザーポーラーの事前フォークインスタンス数を調整できます。

複雑な Web サイトや Web アプリケーションの監視には、Website by Browserテンプレートが標準テンプレートとして利用できます。

設定

アイテム設定フォームのタイプフィールドで「ブラウザー」を選択し、必要なフィールドに入力します。

すべての必須入力フィールドには赤いアスタリスクが付いています。

ブラウザーアイテムに固有の情報が必要なフィールドは次のとおりです。

フィールド	説明
キー	アイテムを識別するために使用される一意のキーを入力します。
パラメーター	スクリプトに渡す変数を属性と値のペアで指定します。 ユーザーマクロがサポートされています。サポートされている組み込みマクロを確認するには、サポートされているマクロテーブルで「ブラウザータイプのアイテム」を検索してください。
スクリプト	パラメーターフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディターに JavaScript コードを入力します。このコードはメトリック値を返すロジックを提供する必要があります。 コードは、すべてのパラメーター、すべての追加 JavaScript オブジェクト、および Zabbix によって追加されたブラウザーアイテム JavaScript オブジェクトにアクセスできます。 参考: JavaScript ガイド。
タイムアウト	JavaScript 実行のタイムアウト（1～600 秒；これを超えるとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでに時間がかかる場合があります。 タイムアウトパラメーターの詳細については、一般的なアイテム属性を参照してください。

例

Website by Browser テンプレートを使用して Web サイトを監視するための Zabbix のセットアップ方法の例については、Browser アイテムで Web サイトを監視するを参照してください。

デフォルトスクリプト

以下のスクリプトは、次の処理を行います。

1. ブラウザーセッションを初期化します。
2. 指定された URL に移動します。
3. パフォーマンスエントリとセッション統計を収集し、JSON 文字列として返します。

スクリプトフィールドに以下を入力します。

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

カスタム機能でブラウザを初期化する

以下のスクリプトは:

1. スクリプト内で指定された順序で最初に一致するブラウザに基づいて、利用可能なブラウザのセッションを初期化します。
2. ページロード戦略や、Chrome、Firefox、Microsoft Edge ブラウザのヘッドレスモードなど、各ブラウザ固有のオプションを含むブラウザ機能を定義します。

スクリプトフィールドに以下を入力します:

```
var browser = new Browser({
    "capabilities":{
        "firstMatch":[
            {
                "browserName":"chrome",
                "pageLoadStrategy":"normal",
                "goog:chromeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"firefox",
                "pageLoadStrategy":"normal",
                "moz:firefoxOptions":{
                    "args":[
                        "--headless"
                    ]
                }
            },
            {
                "browserName":"MicrosoftEdge",
                "pageLoadStrategy":"normal",
                "ms:edgeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"safari",
                "pageLoadStrategy":"normal"
            }
        ]
```

```
    }
});
```

GUI でブラウザを初期化する

デフォルトでは、ブラウザセッション（Safari を除く）はヘッドレスモードで初期化されます。つまり、ブラウザのグラフィカルユーザーインターフェース（GUI）は表示されません。

以下のスクリプトは、GUI を有効にしたブラウザセッションを初期化します。

WebDriver がブラウザのバイナリを見つけられない場合は、パスを手動で指定できます。

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// GUI で Firefox セッションを初期化するには、以下の行のコメントを外してください。
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// GUI で Microsoft Edge セッションを初期化するには、以下の行のコメントを外してください。
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

Note:
テストがリモートサーバーやコンテナで実行されている場合は、VNC（Virtual Network Computing）クライアントを使用してマシンの VNC サーバーに接続できます。これにより、ブラウザの GUI をリモートで表示および操作できます。

スクリーンショットを撮る

以下のスクリプトは:

1. ブラウザセッションを初期化します。
2. ブラウザのビューポートサイズを設定してスクリーンショットのサイズを決定します（パラメータで指定、下記参照）。
3. URL に移動します（パラメータで指定、下記参照）。
4. セッション統計情報を収集し、スクリーンショットを撮影して収集した統計情報に追加します。
5. エラーが発生した場合は、エラーメッセージとスクリーンショットを取得します。
6. 収集した結果を JSON 文字列として返します。

このスクリプトは、アイテム設定フォームからのパラメータも使用します:

- webURL - http://example.com
- width - 1920
- height - 1080

スクリプトフィールドに以下を入力します:

```
var browser, result;

var browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Zabbix から渡されたパラメータを含む JSON 文字列を解析します。

    browser.setScreenSize(Number(params.width), Number(params.height))

    browser.navigate(params.webURL);

    result = browser.getResult();
    result.screenshot = browser.getScreenshot();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
```

```
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Zabbix ログインのチェック

以下のスクリプトは、

1. ブラウザセッションを初期化します。
2. ページに移動します（パラメータで指定、下記参照）。
3. ユーザー名とパスワードを入力します（パラメータで指定、下記参照）。
4. ログインボタンを見つけてクリックします。
5. ログアウトボタンを見つけてクリックします。
6. ログイン前後およびログアウト後のパフォーマンスデータを収集します。
7. エラーが発生した場合、エラーメッセージとスクリーンショットを取得します。
8. 収集した結果を JSON 文字列として返します。

また、アイテム設定フォームのパラメータも使用します。

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

スクリプトフィールドには、以下を入力します。

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.navigate(params.webURL);
    browser.collectPerfEntries("open page");

    var el = browser.findElement("xpath", "//input[@id='name']");
    if (el === null) {
        throw Error("cannot find name input field");
    }
    el.sendKeys(params.username);

    el = browser.findElement("xpath", "//input[@id='password']");
    if (el === null) {
        throw Error("cannot find password input field");
    }
    el.sendKeys(params.password);

    el = browser.findElement("xpath", "//button[@id='enter']");
    if (el === null) {
        throw Error("cannot find login button");
    }
    el.click();

    browser.collectPerfEntries("login");

    el = browser.findElement("link text", "Sign out");
    if (el === null) {
        throw Error("cannot find logout button");
    }
    el.click();

    browser.collectPerfEntries("logout");
```

```
    result = browser.getResult();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

リンクの検索

以下のスクリプトは:

1. ブラウザセッションを初期化します。
2. 配列から重複要素を削除する関数を定義します（ステップ 5 を参照）。
3. ページに移動します（パラメータで指定、下記参照）。
4. ページ上のリンクを検索します。
5. リンクの重複を削除し、一意にします。
6. "http" で始まるリンクのみを抽出します。
7. 抽出したリンクを特定の構造にフォーマットします。
8. エラーが発生した場合、エラーメッセージとスクリーンショットを取得します。
9. 収集した結果を JSON 文字列として返します。

また、アイテム設定フォームのパラメータも使用します:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

スクリプトフィールドに以下を入力します:

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value);  // Zabbix から渡されたパラメータを含む JSON 文字列を解析します。

    function uniq(a) {
        return a.sort().filter(function (item, pos, ary) {
            return !pos || item != ary[pos - 1];
        });
    }

    browser.navigate(params.scheme + '://' + params.domain + params.path);

    var el = browser.findElements("link text", "");
    var links = [];
    for (var n = 0; n < el.length; n++) {
        links.push(el[n].getAttribute('href'));
    }

    links = uniq(links);

    result = [];
    for (i = 0; i < links.length; i++) {
        if (links[i].match(/^http.*/)) {
            var row = {};
            row["{#URL}"] = links[i];
            result.push(row);
        }
```

```
    }
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

4 ヒストリとトレンド

概要

Zabbix では、収集したデータを保存する方法として、ヒストリとトレンドの 2 つがあります。

ヒストリは収集した各値を保持するのに対し、トレンドは 1 時間単位で平均化された情報を保持するため、リソースをあまり消費しません。

ヒストリの保持

ヒストリを何日間残すかを設定することができます。

- アイテムの設定画面
- アイテムの一括更新時
- ハウスキーパータスクの設定を行う時

古いデータは、ハウスキーパープロセスによって削除されます。

一般的なアドバイスとして、ヒストリはできるだけ少ない日数で保存し、多くのヒストリの値でデータベースに負荷をかけないようにすることが重要です。

例えば、ヒストリを 14 日間、トレンドを 5 年間保存することができます。

ヒストリとトレンドのデータでどれくらいの容量が必要かは、データベースのサイジングを参照してください。

グラフは古いデータを表示するためにトレンドの値を使用するので、短いヒストリの保存期間であっても、グラフで古いデータを確認することができます。

Attention:
ヒストリの保存期間を'0' に設定すると、アイテムが更新されるのは依存アイテムとインベントリのみを更新します。トリガー条件式の評価はヒストリの値にのみ基づいて行われるため、トリガー関数は評価されません。

Note:
ヒストリを保存する別の方法として、ローダブルモジュールのヒストリエクスポート機能を使うことを検討してください。

トレンドの保持

トレンドは、数値データタイプの 1 時間あたりの最小値、最大値、平均値、および合計値を保存するために組み込まれたヒストリデータ削減メカニズムです。

トレンドは何日間保存されるかを設定することができます。

- アイテムの設定画面で設定できます。
- アイテムの一括更新時
- データの保存期間を設定するとき

トレンドは通常、ヒストリよりもはるかに長い期間保持することができます。古いデータは housekeeper プロセスによって削除されます。

Zabbix サーバーは、データが流入するにつれて、実行時にトレンドデータをトレンドキャッシュに蓄積します。サーバーは、以下のような場合、すべてのアイテムの前の **1** 時間のトレンドをデータベース (Web インターフェースが検索可能な場所) にフラッシュします：

- サーバーがそのアイテムの最初の現在時刻の値を受信したとき
- 現在の時間が残り 5 分以下で、まだそのアイテムの現在の時間の値がないとき

- サーバーが停止するとき

グラフでトレンドを見るには、少なくとも次の時間の始まりまで（アイテムが頻繁に更新される場合）、最大でも次の時間の終わりまで（アイテムがほとんど更新されない場合）、つまり最大 2 時間待つ必要があります。

サーバーがトレンドキャッシュをフラッシュし、データベースにその時間のトレンドが既にある場合
 （例えば、サーバーが時間の途中で再起動した場合）、サーバーは単純な挿入の代わりに更新ステートメントを使用する必要があります。
 したがって、大規模なインストールで再起動が必要な場合は、トレンドデータの重複を避けるために、ある時間の終わりにサーバーを停止し、次の時間の初めに開始することが望ましいです。

ヒストリテーブルはトレンドの生成には一切関与しません。

Attention:
トレンドの保存期間が'0' に設定されている場合、Zabbix サーバーはトレンドの計算や保存を全く行いません。

Note:
トレンドは元の値と同じデータ型で計算され、保存されます。その結果、符号なしデータ型の値の平均値の計算は丸められ、値の間隔が狭いほど結果の精度は低くなります。例えば、項目が 0 と 1 の値を持つ場合、平均値は 0.5 ではなく、0 になります。
また、サーバーを再起動すると、符号なしデータ型の平均値の計算の精度が落ちることがあります。

5 ユーザーパラメーター

概要

Zabbix にあらかじめ定義されていないエージェントチェックを実行したい場合があります。そのような場合にユーザーパラメータが役立ちます。

必要なデータを取得するコマンドを作成し、それをエージェントの設定ファイルのユーザーパラメータ（'UserParameter' 設定パラメータ）に含めることができます。

ユーザーパラメータの構文は以下の通りです。

```
UserParameter=<キー>,<コマンド>
```

ご覧の通り、ユーザーパラメータにもキーが含まれています。このキーはアイテムを設定する際に必要となります。参照しやすい任意のキーを入力してください（ホスト内で一意である必要があります）。

エージェントを再起動するか、エージェントのランタイム制御オプションを使用して新しいパラメータを反映させます。例：

```
zabbix_agentd -R userparameter_reload
```

その後、アイテムを設定する際に、実行したいユーザーパラメータのコマンドを参照するためのキーを入力します。

ユーザーパラメータは Zabbix エージェントによって実行されるコマンドです。アイテム値の事前処理ステップの前に、最大 16MB のデータを返すことができる点に注意してください。

UNIX 系 OS では**/bin/sh** がコマンドラインインタプリタとして使用されます。ユーザーパラメータはエージェントチェックのタイムアウトに従います。タイムアウトに達した場合、フォークされたユーザーパラメータプロセスは終了されます。

関連情報:

- ユーザーパラメータを活用するためのステップバイステップチュートリアル
- コマンド実行

単純なユーザーパラメータの例

単純なコマンド:

```
UserParameter=ping,echo 1
```

agent は、'ping' キーを持つ item に対して常に'1' を返します。

より複雑な例:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

agent は、MySQL サーバが生きている場合は'1' を、そうでない場合は'0' を返します。

柔軟なユーザーパラメータ

柔軟なユーザーパラメータは、キーにパラメータを受け入れます。この方法で、柔軟なユーザーパラメータは複数のアイテムを作成するための基礎となります。

柔軟なユーザーパラメータは次の構文を持ちます:

```
UserParameter=key[*],command
```

パラメータ	説明
Key	一意のアイテムキー。[*] は、このキーがブラケット内のパラメータを受け入れることを定義します。パラメータはアイテムの設定時に指定されます。
Command	キーの値を評価するために実行されるコマンド。 柔軟なユーザーパラメータのみ: コマンド内で位置参照 $1...$9 を使用して、アイテムキー内のそれぞれのパラメータを参照できます。 Zabbix はアイテムキーの [ ] で囲まれたパラメータを解析し、コマンド内の $1,...,$9 をそれに応じて置換します。 $0 は、実行される元のコマンド（$0,...,$9 の展開前）に置換されます。 位置参照は、ダブルクォート (") またはシングルクォート (') で囲まれているかどうかに関係なく解釈されます。 位置参照を変更せずに使用するには、ダブルドル記号を指定します（例: awk '{print $$2}'）。この場合、$$2 は実際にコマンド実行時に $2 に変換されます。

Attention:
$ 記号による位置参照は、柔軟なユーザーパラメータに対してのみ Zabbix エージェントによって検索および置換されます。単純なユーザーパラメータの場合、そのような参照処理はスキップされるため、$ 記号のエスケープは必要ありません。

Attention:
特定の記号は、デフォルトでユーザーパラメータで許可されていません。完全なリストについては、UnsafeUserParametersのドキュメントを参照してください。

例 1

とてもシンプルなもの:

```
UserParameter=ping[*],echo $1
```

ping[something] というフォーマットを持つ監視項目は無制限に定義することができる。

- ping[0] - 常に'0' を戻す
- ping[aaa] - 常に'aaa' を戻す

例 2

もっとセンスを磨こう!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

このパラメータは、MySQL データベースの可用性を監視するために使用することができます。
 ユーザー名とパスワードを渡すことができます:

```
mysql.ping[zabbix,our_password]
```

例 3

正規表現にマッチする行はファイル中に何行あるか？

```
UserParameter=wc[*],grep -c "$2" $1
```

このパラメータは、ファイル内の行数を計算するために使用することができます。

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

コマンドの結果

コマンドの戻り値は、コマンドによって生成された標準出力と標準エラー出力の両方です。

Attention:
テキストを返すアイテム (文字列、ログ、テキスト型の情報) は、標準エラー出力があっても未サポートにはなりません。

戻り値は 16MB に制限されています (切り捨てられる末尾の空白も含む); データベースの制限も適用されます。

テキストを返すユーザーパラメータ (文字列、ログ、テキスト型の情報) も空白を返すことができます。無効な結果の場合、アイテムは未サポートになります。

1 Zabbix エージェントの機能拡張

このチュートリアルでは、ユーザーパラメータを使用して Zabbix エージェントの機能を拡張する方法を段階的に説明します。

ステップ 1

必要なパラメータを取得するためのスクリプトまたはコマンドラインを記述します。

例えば、MySQL サーバで実行されたクエリの総数を取得するために、以下のようなコマンドを記述することができます。

```
mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

このコマンドを実行すると、SQL クエリの総数が返されます。

ステップ 2

zabbix_agentd.conf にコマンドを追加します。

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

mysql.questions は一意な識別子です。それは任意の有効なキーであることができます。例えば、*queries* などです。

このパラメータをテストするには、Zabbix agent を「-t」フラグで使用します (root 権限で実行する場合)。ただし、デーモンとして起動した場合、agent のパーミッションが異なる可能性があります。

```
zabbix_agentd -t mysql.questions
```

ステップ 3

以下のコマンドを実行して、設定ファイルからユーザパラメータを再ロードします。

```
zabbix_agentd -R userparameter_reload
```

実行時制御コマンドの代わりに、agent を再起動することもできます。

zabbix_get ユーティリティを使用して、パラメータをテストします。

ステップ 4

監視するホストに Key=mysql.questions の新しい item を追加します。
 item のタイプは、Zabbix agent または Zabbix agent (active) のいずれかにする必要があります。

Zabbix server で返される値のタイプが正しく設定されている必要があることに注意してください。
 そうでない場合、Zabbix は値を受け付けません。

6 Windows パフォーマンスカウンタ

概要

Windows のパフォーマンスカウンタは、*perf_counter[]* キーを使用して効率的に監視することができます。

例えば:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

あるいは

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

このキーまたは英語版のみの同等物 perf_counter_en の使用に関する詳細については、[Windows-specific item keys]((/manual/config/items/it
を参照してください。

監視可能なパフォーマンスカウンタの全リストを取得するには、次のコマンドを実行します。

```
typeperf -qx
```

また、ローレベルディスカバリ機能を使用して、複数のobject instancesを検出し、複数のインスタンスオブジェクトに対する <br> perf_counter item の作成を自動化することも可能です。

数値表現

Windows は、オブジェクト名やパフォーマンスカウンター名に対して数値表現（インデックス）を保持しています。Zabbix は、perf_counter、perf_counter_en アイテムキーや PerfCounter、PerfCounterEn 構成パラメータのパラメータとして、これらの数値表現をサポートしています。

ただし、特定のホストで数値インデックスが正しい文字列にマッピングされることを保証できない限り、これらを使用することは推奨されません。さまざまなローカライズされた Windows バージョンを持つ異なるホストで動作する移植可能なアイテムを作成する必要がある場合は、システムロケールに関係なく英語名を使用できる `perf_counter_en` キーまたは `PerfCounterEn` 構成パラメータを使用できます。

数値の対応関係を調べるには、**regedit** を実行し、*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009* の *Counter* を探します。

レジストリエントリには次のような情報が含まれています:

```
1
1847
2
System
4
Memory
6
% Processor Time
10
File Read Operations/sec
12
File Write Operations/sec
14
File Control Operations/sec
16
File Read Bytes/sec
18
File Write Bytes/sec
....
```

ここで、'\System\% Processor Time' のようなパフォーマンスカウンターの各文字列部分に対応する番号を見つけることができます:

```
System → 2
% Processor Time → 6
```

その後、これらの番号を使用してパスを数値で表現できます:

```
\2\6
```

パフォーマンスカウンタパラメーター

Windows のパフォーマンスカウンタを監視するために、いくつかの PerfCounter パラメータを導入することができます。

例えば、Zabbix エージェント設定ファイルに以下を追加することができます。

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
```
```
または
```
```
PerfCounter=UserPerfCounter2,"\4\24",30
```

このようなパラメータを設定すると、それぞれの item を作成するためのキーとして *UserPerfCounter1* または *UserPerfCounter2* を
 使用するだけでよくなります。

設定ファイルを変更した後は、Zabbix エージェントを再起動することを忘れないでください。

7 一括アップデート

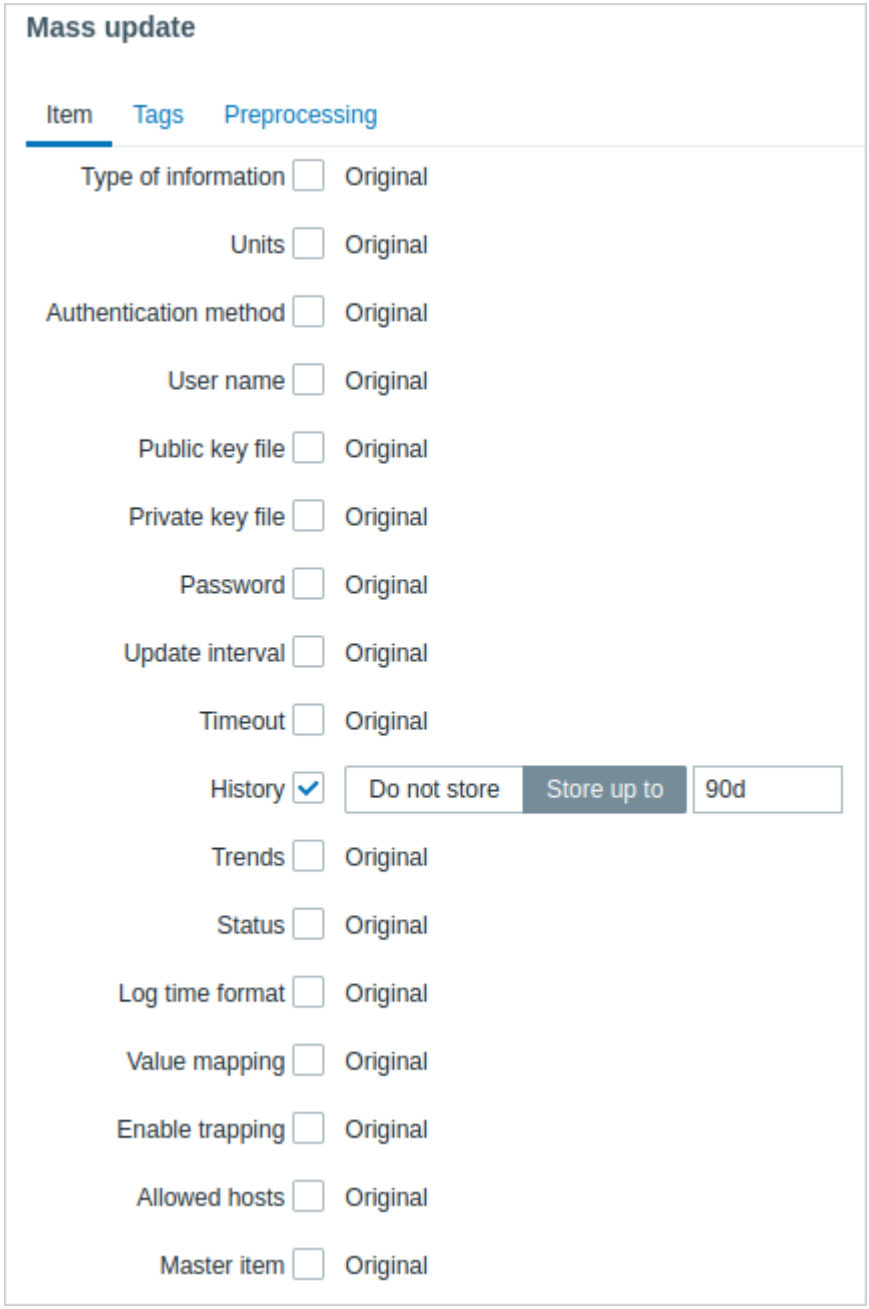
概要

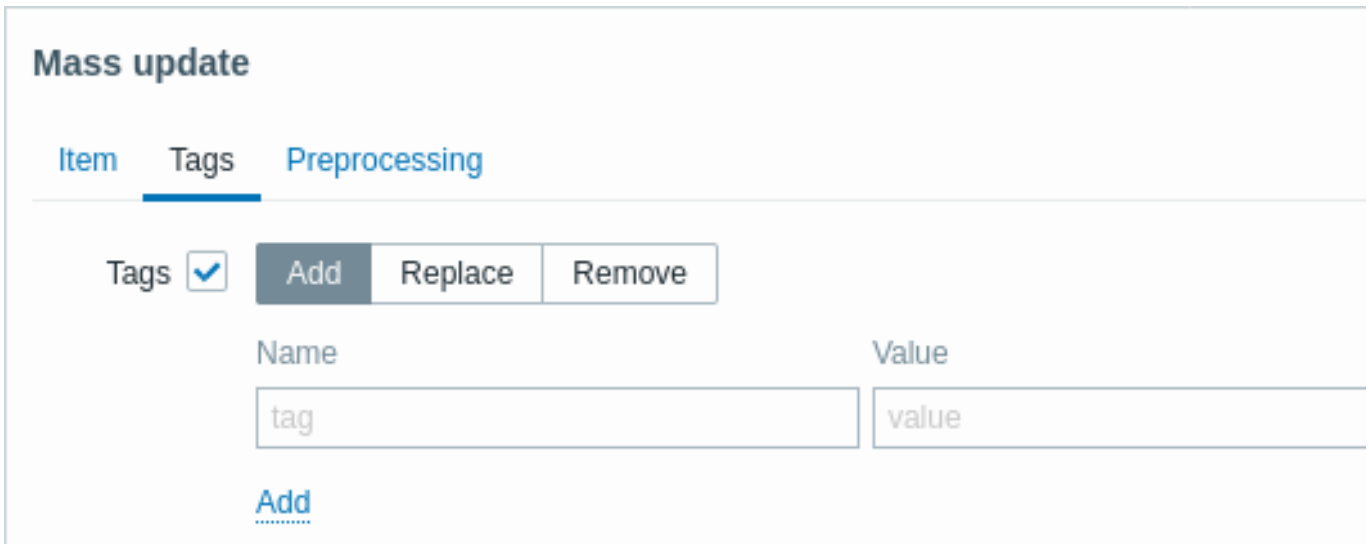
一度に複数のアイテムの属性を変更したい場合があります。個々のアイテムを開いて編集する代わりに一括更新機能を使います。

一括更新の使用

いくつかのアイテムを一括で更新するには、次のようにします。

- リスト内の更新するアイテムのチェックボックスにチェックを付ける
- リストの下にある一括更新をクリックする
- 必要な属性 (アイテム、タグ、保存前処理) のあるタブに移動する
- 更新する属性のチェックボックスにチェックを付ける
- 属性の新しい値を入力する



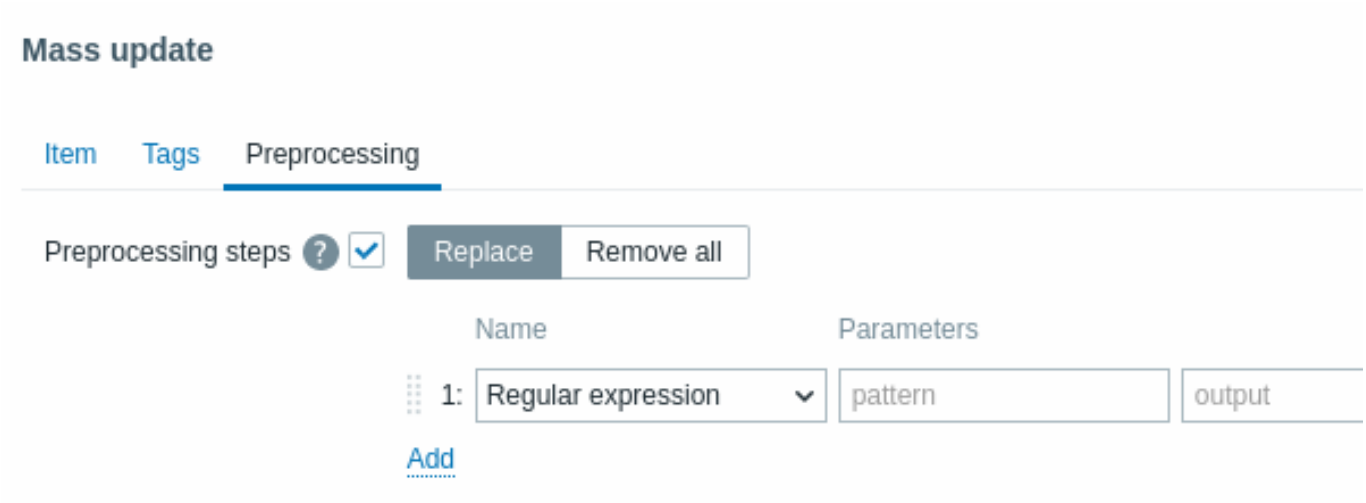


タグオプションでは次の操作が可能です。

- 追加 - 指定したタグをアイテムに追加します (同じ名前でも異なる値のタグは' 重複' とみなされず、同じホストに追加できます)。
- 置換 - 指定されたタグを削除し、新しい値のタグを追加します。

- 削除 - 指定されたタグを項目から削除します。

タグではユーザーマクロ、{INVENTORY. *} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、そして {HOST.ID} マクロがサポートされています。



保存前処理のステップの横にあるチェックボックスをチェックして、保存前処理のステップの一括更新を有効にします。

保存前処理の一括更新では次の操作が可能です。

- *Replace* - アイテムの既存の保存前処理のステップを、以下に指定する保存前処理のステップに置き換えます
- *Remove all* - アイテムからすべての保存前処理ステップを削除します

完了したら、更新をクリックします。

8 値のマッピング

概要

値のマッピングを使用すると、数値/文字列の値と文字列表現の間のマッピングを使用して、受信した値をよりユーザーフレンドリーな表現に設定できます。

たとえば、アイテムの値が”0” または”1” の場合、値のマッピングを使用してこれらの値をよりユーザーフレンドリーな方法で表現できます。

- 0 → 利用不可
- 1 → 利用可能

データバックアップタイプの値のマッピングは、次のように設定できます。

- F → フル
- D → 差分
- I → 増分

電圧の値の範囲は、次のように設定できます。

- <=209 → 低
- 210-230 → OK
- >=231 → 高

値のマッピングは、Zabbix フロントエンドおよびメディアタイプによって送信される通知で使用されます。

Attention:
受信した値を設定された表現に置き換える処理は、Zabbix フロントエンドとサーバーの両方で実行されます。ただし、サーバーが置換を処理するのは、次の場合のみです。

- ホストインベントリを入力する場合
- サポートされているマクロ {ITEM.VALUE}、{ITEM.LASTVALUE}、{EVENT.OPDATA}、{EVENT.CAUSE.OPDATA} を展開する場合

値のマッピングはテンプレートまたはホストで設定します。一度設定すると、該当するテンプレートまたはホスト内のすべてのアイテムで利用できます。アイテムを設定する際に、値のマッピングパラメータで事前に設定した値のマッピング名を指定します。

Note:
値のマッピングの継承はありません。ホストやテンプレートは、リンクされたテンプレートから値のマッピングを継承しません。ホスト上のテンプレートアイテムは、テンプレートで設定された値のマッピングを引き続き使用します。

Note:
値のマッピングは、数値（整数）、数値（浮動小数点）、文字列の情報タイプを持つアイテムで使用できます。

値のマッピングは、該当するテンプレートまたはホストとともにエクスポート/インポートされます。また、ホストおよびテンプレートの一括更新フォームを使用して一括更新することもできます。

設定

値のマッピングを設定するには、次の手順に従います。

1. ホストまたはテンプレートの設定フォームを開きます。

2. 値のマッピングタブで、追加をクリックして新しい値のマッピングを追加するか、既存のマッピング名をクリックして編集します。

値のマッピングのパラメータ:

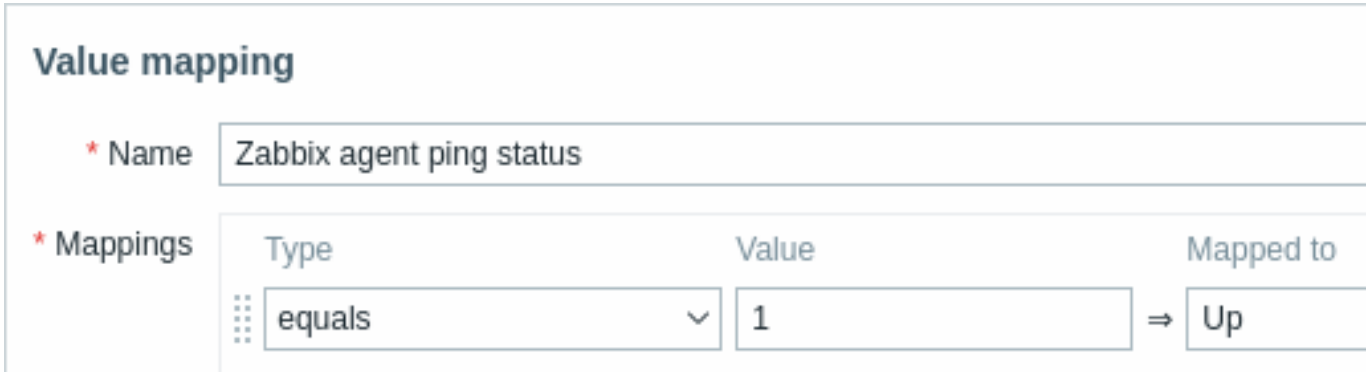
パラメータ		説明
名前		値のマッピングセットの一意の名前。
マッピング		数値/文字列の値を文字列表現にマッピングする個別のルール。 マッピングはルールの順序で適用され、ドラッグして並べ替えることができます。
	タイプ	マッピングタイプ: 等しい - 等しい値がマッピングされます; 以上 - 等しいまたは大きい値がマッピングされます; 以下 - 等しいまたは小さい値がマッピングされます; 範囲内 - 範囲内の値がマッピングされます; 範囲は <number1>-<number2> または <number> で表現されます; 複数の範囲がサポートされます (例: 1-10,101-110,201); 正規表現 - 正規表現に一致する値がマッピングされます (グローバル正規表現はサポートされません); デフォルト - 特定のマッピングがないすべての値がマッピングされます。 範囲のマッピングでは、数値型の値 (以上、以下、範囲内) のみがサポートされます。
	値	入力値 (マッピングタイプに応じて範囲または正規表現を含む場合があります)。
	マッピング先	入力値の文字列表現 (最大 64 文字)。

すべての必須入力フィールドには赤いアスタリスクが付いています。

値のマッピングをリストで表示する場合、最初の 3 つのマッピングのみが表示され、それ以上のマッピングが存在する場合は 3 つのドットが表示されます。

値のマッピング例

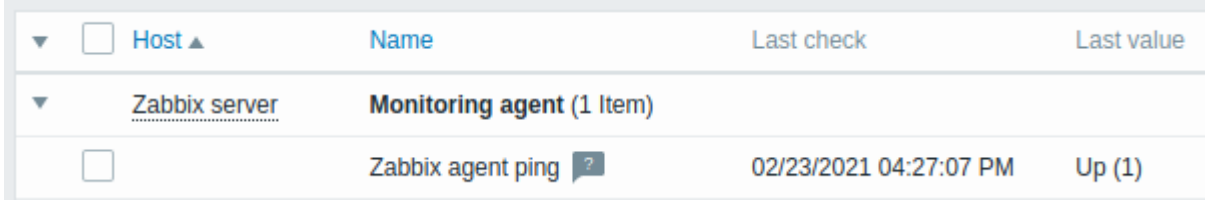
あらかじめ定義されているエージェントアイテム *Zabbix agent ping* は、テンプレートレベルの値のマッピング「Zabbix agent ping status」を使用して、その値を表示します。



アイテムの設定フォームでは、値のマッピングフィールドでこの値のマッピングへの参照を見つけることができます。

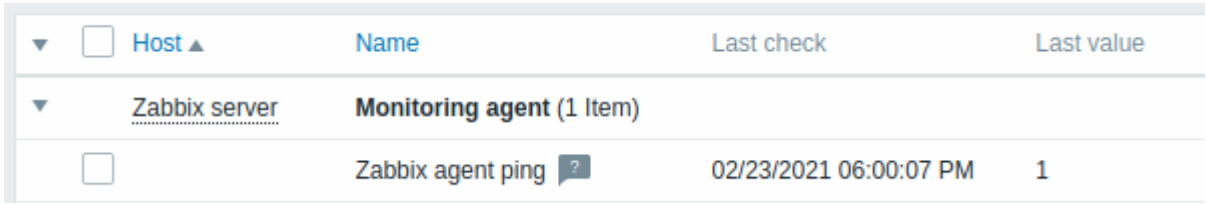


このマッピングは、監視データ → 最新データセクションで「Up」(生の値は括弧内）として表示するために使用されます。



Note:
最新データセクションでは、表示される値は 20 文字に短縮されます。値のマッピングが使用されている場合、この短縮はマッピングされた値には適用されず、生の値（括弧内に表示）にのみ適用されます。

あらかじめ定義された値のマッピングがない場合、「1」だけが表示され、理解しにくい場合があります。



9 キュー

概要

キューには、更新を待っているアイテムが表示されます。キューはデータの論理的な表現にすぎません。Zabbix には IPC キューやその他のキュー機構はありません。

プロキシによって監視されているアイテムもキューに含まれます。これらはプロキシの履歴データ更新期間中、キューに入っているとカウントされます。

キューには、スケジュールされた更新時間を持つアイテムのみが表示されます。つまり、以下のアイテムタイプはキューから除外されます。

- log、logrt、およびイベントログのアクティブ Zabbix エージェントアイテム
- SNMP トラップアイテム
- トラッパーアイテム
- Web 監視アイテム
- 従属アイテム

キューによって表示される統計情報は、Zabbix サーバーのパフォーマンスを示す良い指標です。

キューは、JSON プロトコルを使用して Zabbix サーバーから直接取得されます。この情報は、Zabbix サーバーが稼働している場合にのみ利用できます。

Attention:
接続の問題やエージェントの不具合によりアイテムインターフェースが利用できなくなった場合、アイテムはキューにカウントされません。

キューの読み取り

キューを読み取るには、管理 > キュー > キューの概要に移動します。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

ここに表示されている画像は一般的に「正常」であり、サーバーが正常に動作していると考えられます。

キューには最大 30 秒待機しているアイテムが表示されています。これらがどのアイテムなのかを知ることができれば便利です。

それを確認するには、タイトルのドロップダウンでキューの詳細を選択します。これで遅延しているアイテムのリストが表示されます。

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

これらの詳細情報があれば、なぜこれらのアイテムが遅延しているのかを調べることができるかもしれません。

1 つまたは 2 つのアイテムが遅延している場合は、特に心配する必要はないかもしれません。1 秒後には更新される可能性があります。しかし、多数のアイテムが長時間遅延している場合は、より深刻な問題があるかもしれません。

関連情報: スケジューリング間隔を使用する場合のタイムゾーンの調整についても参照してください。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

キュー item

特別な内部 item **zabbix[queue,<from>,<to>]** を使用すると、Zabbix のキューの状態を監視することができます。
 設定された時間だけ遅延したアイテムの数が返されます。詳細については、Internal itemsを参照してください。

10 バリューキャッシュ

概要

トリガー式、計算アイテム、一部のマクロの計算を高速化するために、Zabbix server ではバリューキャッシュオプションがサポートされています。

このインメモリキャッシュを利用すると、直接データベースへ SQL を発行せずに過去のデータへアクセスすることができます。過去の値がキャッシュに存在しない場合、不足分をデータベースから取得し、キャッシュの更新を行います。

アイテムの値は、次のいずれかの状態になるまでバリューキャッシュに残ります:

- アイテムが削除された場合 (キャッシュされた値は、次の構成同期後に削除されます)
- アイテムの値が、トリガーもしくは計算アイテム式で指定された時間またはカウントの範囲外になった場合 (キャッシュされた値は、新しい値を受信した時に削除されます)
- トリガー/計算アイテム式で指定された時間またはカウント範囲が変更され、計算に必要なデータが少なくなった場合 (不要なキャッシュされた値は 24 時間後に削除されます)

Note:
バリューキャッシュの状態は、サーバーのランタイムコントロール でオプション `diaginfo` (または `diaginfo=valuecache`) を実行し、診断情報のバリューキャッシュセクションを調べることで確認できます。これは、誤って構成されたトリガーまたは計算アイテムを特定する場合に役立ちます。

バリューキャッシュ機能を有効にするために、オプションの **ValueCacheSize** パラメータが Zabbix server の設定ファイル でサポートされています。

バリューキャッシュを監視するには、以下の 2 つの内部チェックアイテムが利用できます。**zabbix[vcache,buffer,<mode>]** **zabbix[vcache,cache,<parameter>]** 詳細は、内部チェックを参照してください。

11 即時実行

概要

今すぐ実行オプションを使用すると、パッシブチェックを即座に実行できます。

Zabbix でのアイテム値の収集は、設定された更新間隔に従います。一部のアイテム（例：ローレベルディスカバリルール）は間隔が長く、すぐに新しい値を取得する必要がある場合があります。たとえば、検出可能なリソースの変更を検出する場合などです。

今すぐ実行オプションは、以下のアイテムタイプでサポートされています。

- Zabbix エージェント（パッシブ）
- シンプルチェック
- SNMP エージェント（v1/v2/v3）
- Zabbix 内部
- 外部チェック
- データベースモニター
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 従属アイテム
- スクリプト
- ブラウザー

アイテムが「従属アイテム」タイプの場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

設定

パッシブチェックは以下から実行できます:

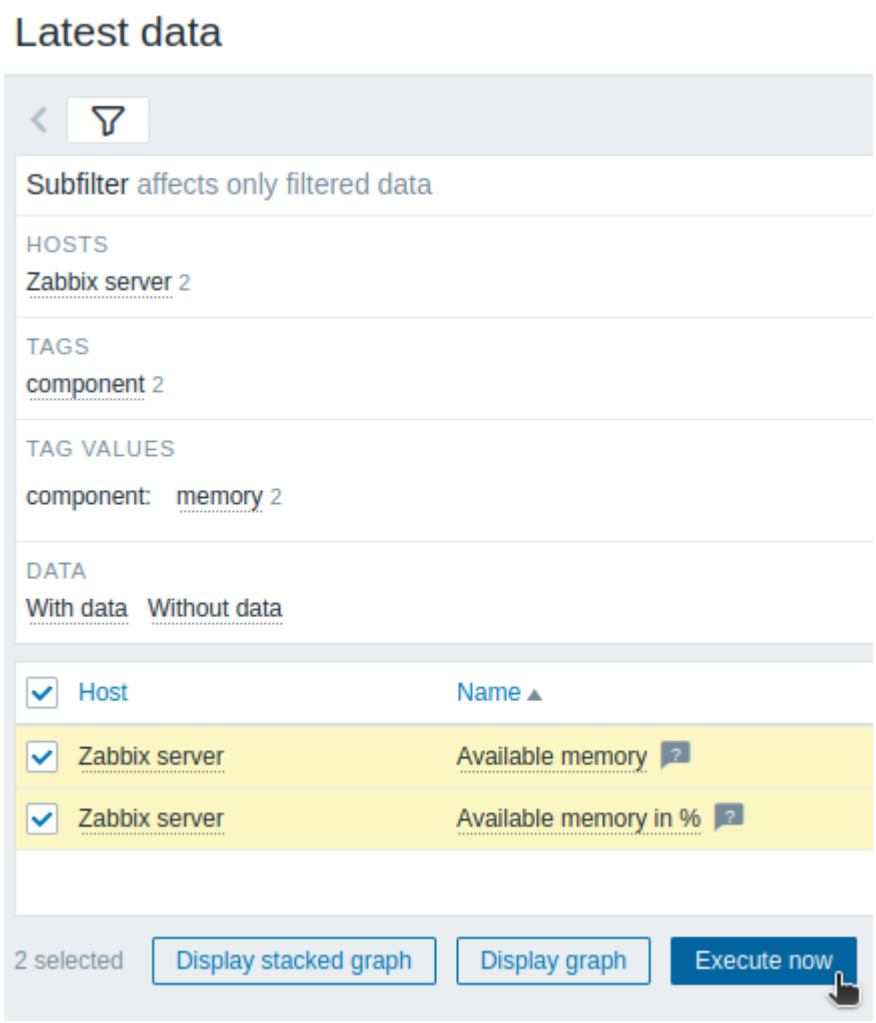
- 監視データ > 最新データ

- データ収集 > ホスト > アイテム またはディスカバリ
- アイテムメニュー
- アイテムまたはローレベルディスカバリルールの編集時（設定フォーム内）

Attention:
チェックを実行するには、設定キャッシュにチェックが存在している必要があります。CacheUpdateFrequencyも参照してください。チェック実行前にキャッシュは更新されません。そのため、アイテムやローレベルディスカバリルールの設定を最近変更した場合は反映されません。作成中または新規作成したアイテムや LLD ルールをテストするには、代わりにテストオプションを使用できます。

例えば、監視データ > 最新データでパッシブチェックを実行するには:

1. アイテムを選択します。複数のアイテムに対して一度にチェックを実行できます。
2. 今すぐ実行をクリックします。



Note:
最新データでは、ユーザーは読み書きの権限を持つホストのアイテム、または読み取り専用の権限かつロールで読み取り専用ホストで「今すぐ実行」を呼び出すアクションが有効な場合に今すぐ実行できます。

12 エージェントチェックの制限

概要

エージェント側でアイテムのブラックリスト、ホワイトリスト、またはホワイトリスト/ブラックリストの組み合わせを作成することで、チェックを制限することができます。

そのためには、2 つのエージェント設定パラメータを組み合わせて使用します:

- `AllowKey=<pattern>` - 許可するチェック; <pattern> はワイルドカード (*) 式で指定
- `DenyKey=<pattern>` - 拒否するチェック; <pattern> はワイルドカード (*) 式で指定

注意点:

- すべての system.run[*] アイテム (リモートコマンド、スクリプト) はデフォルトで無効です。deny key が指定されていない場合でも、DenyKey=system.run[*] が暗黙的に追加されていると考えてください。
- Zabbix 5.0.2 以降、EnableRemoteCommands エージェントパラメータは:
  - Zabbix agent では非推奨
  - Zabbix agent2 ではサポートされていません

したがって、リモートコマンドを許可するには、許可する各コマンドに対して AllowKey=system.run[<command>,*] を指定します。* は wait および nowait モードを表します。また、AllowKey=system.run[*] パラメータを指定して、wait および nowait モードのすべてのコマンドを許可することもできます。特定のリモートコマンドを許可しない場合は、AllowKey=system.run[*] パラメータの前に system.run[] コマンドを含む DenyKey パラメータを追加します。

重要ルール

- deny ルールのないホワイトリストは、system.run[*] アイテムに対してのみ許可されます。その他のアイテムでは、DenyKey パラメータがない場合、AllowKey パラメータを使用することはできません。
- 順番は重要です。指定されたパラメータは、設定ファイル内の表示順に従って 1 つずつチェックされます:
  - item キーが allow / deny ルールに一致すると、item は許可または拒否され、ルールチェックは停止されます。したがって、ある item が allow ルールと deny ルールの両方にマッチした場合、どちらのルールが先に来るかによって結果が異なります。
  - この順序は、EnableRemoteCommands パラメータにも影響します。(使用する場合)
- AllowKey / DenyKey パラメータは無制限に使用できます。
- AllowKey、DenyKey ルールは、HostnameItem、HostMetadataItem、HostInterfaceItem 構成パラメータには影響しません。
- キーパターンは、ワイルドカード (*) 文字が特定の位置にある任意の数の文字と一致するワイルドカード式です。キー名とパラメーターの両方で使用することができます。
- 特定の item キーが agent 構成で許可されていない場合、その item はサポートされていないと報告されます。(理由のヒントは与えられません)。
- Zabbix agent に --print (-p) コマンドラインオプションを指定すると、設定により許可されていないキーは表示されません。
- Zabbix agent の--test(-t) コマンドラインオプションを使用すると、設定で許可されていないキーに対して”Unsupported item key.” というステータスが返されます。
- 拒否されたリモートコマンドは agent ログに記録されません。(LogRemoteCommands=1 の場合)

使用例

特定のチェックを拒否する

- DenyKey パラメータで特定のチェックをブラックリスト化します。一致するキーは許可されません。一致しないキーはすべて許可されます。ただし、system.run[] 項目を除きます。

例えば:

```
# セキュアデータアクセスを拒否する
DenyKey=vfs.file.contents[/etc/passwd,*]
```

Attention:
Zabbix の新しいバージョンでは、既存の設定で明示的に制限されていない新しいキーが使用される可能性があるため、
ブラックリストは良い選択とは言えないかもしれません。これはセキュリティ上の欠陥を引き起こす可能性があります。

特定のコマンドを拒否し、他のコマンドを許可する

- 特定のコマンドをブラックリスト化するには、DenyKey パラメータを使用します。AllowKey パラメータを使用して、他のすべてのコマンドをホワイトリストに登録します。

```
# 特定のコマンドを許可しない
DenyKey=system.run[ls -l /]

# 他のスクリプトを許可する
AllowKey=system.run[*]
```

特定のチェックを許可し、他のチェックを拒否する

- AllowKey パラメータで特定のチェックをホワイトリスト化し、DenyKey=* で他のチェックを拒否する。

例:

```
# Allow reading logs:
AllowKey=vfs.file.*[/var/log/*]

# Allow localtime checks
```

```
AllowKey=system.localtime[*]

# Deny all other keys
DenyKey=*
```

パスや引数にスペースを含むスクリプトの許可

- スペースを含むパスにあるスクリプトをホワイトリストに登録し、引数の受け渡しも許可します。Windows では、パスを指定する際に ^ でスペースをエスケープしてください。

例:

```
# 引数の有無にかかわらず test.bat の実行を許可
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
```

テスト方法:

```
PS C:\Program Files\Zabbix Agent 2> .\zabbix_get.exe -s 127.0.0.1 -k system.run["C:\Program^ Files\Zabbix^
```

この設定により、引数（caret など）の有無にかかわらずスクリプトの実行が許可され、スクリプトのパスにスペースが含まれていても system.run の実行が許可されます。

パターン例

パターン	説明	一致	不一致
*	パラメータの有無にかかわらず、すべてのキーに一致します。	すべて	なし
vfs.file.contents	パラメータなしの vfs.file.contents に一致します。	vfs.file.contents	vfs.file.contents[/etc/passwd]
vfs.file.contents[]	空のパラメータを持つ vfs.file.contents に一致します。	vfs.file.contents[]	vfs.file.contents
vfs.file.contents[*]	任意のパラメータを持つ vfs.file.contents に一致します。角括弧なしの vfs.file.contents には一致しません。	vfs.file.contents[] vfs.file.contents[/path/to/file]	vfs.file.contents
vfs.file.contents[/etc/passwd,]*	最初のパラメータが/etc/passwd に一致し、他のすべてのパラメータが任意の値 (空も含む) である vfs.file.contents に一致します。	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]	vfs.file.contents[/etc/passwd] vfs.file.contents[/var/log/zabbix_s vfs.file.contents[]
*vfs.file.contents[*passwd*]*	最初のパラメータが *passwd* に一致し、他のパラメータがない vfs.file.contents に一致します。	vfs.file.contents[/etc/passwd,]	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd, utf8]
*vfs.file.contents[*passwd*,*]*	最初のパラメータが *passwd* に一致し、以降のすべてのパラメータが任意の値 (空も含む) である vfs.file.contents に一致します。	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd, utf8]	vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/var/log/zabbix_server.log,]*	最初のパラメータが/var/log/zabbix_server.log、3 番目のパラメータが'abc'、2 番目のパラメータが任意 (空も含む) である vfs.file.contents に一致します。	vfs.file.contents[/var/log/zabbix_server.log,utf8,abc]	vfs.file.contents[/var/log/zabbix_s
vfs.file.contents[/etc/passwd,utf8]	最初のパラメータが/etc/passwd、2 番目のパラメータが'utf8' で、他の引数がない vfs.file.contents に一致します。	vfs.file.contents[/etc/passwd,utf8]	vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf1
*vfs.file.**	パラメータなしで vfs.file. で始まる任意のキーに一致します。	vfs.file.contents vfs.file.size	vfs.file.contents[] vfs.file.size[/var/log/zabbix_serve
vfs.file.[*]*	任意のパラメータを持つ vfs.file. で始まる任意のキーに一致します。	vfs.file.size.bytes[] vfs.file.size[/var/log/zabbix_server.log, utf8]	vfs.file.size.bytes
vfs..contents*	vfs. で始まり .contents で終わるパラメータなしの任意のキーに一致します。	vfs.mount.point.file.contents vfs..contents	vfs.contents

system.run と AllowKey

仮想的なスクリプト「myscript.sh」は、Zabbix エージェントを介してホスト上でいくつかの方法で実行できます。

1. パッシブまたはアクティブチェックのアイテムキーとして、例えば:

- system.run[myscript.sh]
- system.run[myscript.sh,wait]
- system.run[myscript.sh,nowait]

ここでユーザーは「wait」や「nowait」を追加したり、第 2 引数を省略して system.run[] のデフォルト値を使用できます。

2. グローバルスクリプトとして（フロントエンドまたは API でユーザーが開始）。

ユーザーはこのスクリプトをアラート → スクリプトで設定し、「実行先: Zabbix agent」を設定し、スクリプトの「コマンド」入力フィールドに「myscript.sh」を入力します。フロントエンドまたは API から呼び出されると、Zabbix サーバーはエージェントに以下を送信します:

- system.run[myscript.sh,wait] - Zabbix 5.0.4 まで
- system.run[myscript.sh] - 5.0.5 以降

ここではユーザーは「wait」/「nowait」パラメータを制御できません。

3. アクションからのリモートコマンドとして。Zabbix サーバーはエージェントに以下を送信します:

- system.run[myscript.sh,nowait]

ここでもユーザーは「wait」/「nowait」パラメータを制御できません。

これは、AllowKey を次のように設定した場合:

```
AllowKey=system.run[myscript.sh]
```

この場合

- system.run[myscript.sh] - 許可される
- system.run[myscript.sh,wait]、system.run[myscript.sh,nowait] は許可されない - アクションのステップとして呼び出された場合、スクリプトは実行されない

すべてのバリエーションを許可するには、次のように追加できます:

```
AllowKey=system.run[myscript.sh,*]
DenyKey=system.run[*]
```

をエージェント/agent2 のパラメータに追加します。

3 トリガー

概要

トリガーは、アイテムによって収集されたデータを「評価」し、現在のシステム状態を表す論理式です。

アイテムはシステムデータを収集するために使用されますが、これらのデータを常に監視し、警告や注意が必要な条件を待つのは非常に非効率的です。データの「評価」の役割は、トリガー式に任せることができます。

トリガー式を使用すると、データの状態が「許容範囲内」であるかどうかの閾値を定義できます。したがって、受信データが許容範囲を超えた場合、トリガーが「発火」し、ステータスが PROBLEM に変わります。

トリガーには、次のステータスがあります。

ステータス	説明
OK	これは通常のトリガーステータスです。
Problem	何かが発生しました。たとえば、プロセッサの負荷が高すぎます。
Unknown	トリガー値を計算できません。Unknown statusを参照してください。

単純なトリガーでは、たとえば CPU 負荷などのデータの 5 分間平均の閾値を設定したい場合があります。これは、次のようなトリガー式を定義することで実現できます。

- 'avg' 関数をアイテムキーで受信した値に適用する
- 評価期間として 5 分間を使用する
- 閾値を「2」に設定する

```
avg(/host/key,5m)>2
```

このトリガーは、5 分間の平均値が 2 を超えた場合に「発火」（PROBLEM になる）します。

より複雑なトリガーでは、式に複数の関数や複数の閾値の組み合わせを含めることができます。詳細はトリガー式も参照してください。

バイナリ値を持つアイテムにはトリガーを作成できません。

Note:
トリガーを有効にした後（設定ステータスを無効から有効に変更）、トリガー式は、その中のアイテムが値を受け取るか、date and timeやnodata()関数を処理する時間が来るとすぐに評価されます。

ほとんどのトリガー関数はアイテムの値のヒストリデータに基づいて評価されますが、一部のトリガー関数は長期的な分析に使用されます。**trendavg()**、**trendcount()** などはトレンドデータを使用します。

計算時間

トリガーは、Zabbix サーバーが式の一部である新しい値を受信するたびに再計算されます。新しい値が受信されると、式に含まれる各関数が再計算されます（新しい値を受信した関数だけでなく）。

さらに、トリガーは新しい値が受信されるたびに、および式に日付と時刻やnodata()関数が使用されている場合は 30 秒ごとに再計算されます。

日付と時刻やnodata()関数は、Zabbix の history syncer プロセスによって 30 秒ごとに再計算されます。

トレンド関数のみを参照するトリガーは、式内の最小の時間間隔ごとに 1 回評価されます。トレンド関数も参照してください。

評価期間

評価期間は、アイテムの履歴を参照する関数で使用されます。関心のある間隔を指定することができます。時間期間（30s、10m、1h）または値の範囲（#5 - 最新の 5 つの値）として指定できます。

評価期間は「現在」まで測定されます。「現在」とは、トリガーの最新の再計算時刻（上記の計算時刻を参照）であり、サーバーの「現在」時刻ではありません。

評価期間は、次のいずれかを指定します。

- 「現在-期間」から「現在」まで（または、時間シフトを使用する場合は「現在-時間シフト-期間」から「現在-時間シフト」まで）のすべての値を考慮する
- 「現在」までの過去の値のうち、num 個以内の値のみを考慮する
  - 指定された期間または num 個の値が 0 の場合、この関数を使用するトリガーや計算アイテムはサポートされなくなります

注意点：

- トリガーで単一の関数（データ履歴を参照）が使用されている場合、「現在」は常に最新の受信値です。たとえば、最後の値が 1 時間前に受信された場合、評価期間は 1 時間前の最新値までと見なされます。
- 新しいトリガーは、最初の値が受信されるとすぐに計算されます（履歴関数）。日付と時刻およびnodata()関数の場合は 30 秒以内に計算されます。したがって、トリガーが作成されてから設定された評価期間（たとえば 1 時間）がまだ経過していなくても、トリガーは計算されます。また、評価範囲がたとえば最新の 10 個の値に設定されていても、最初の値の後にトリガーが計算されます。

不明なステータス

トリガー式に不明なオペランドが表示される場合があります。

- サポートされていないアイテムが使用されている
- サポートされているアイテムの関数評価でエラーが発生した

この場合、トリガーは一般的に「不明」と評価されます（いくつかの例外があります）。詳細については、不明なオペランドを含む式を参照してください。

不明なトリガーについて通知を受け取ることができます。

1 トリガーの設定

概要

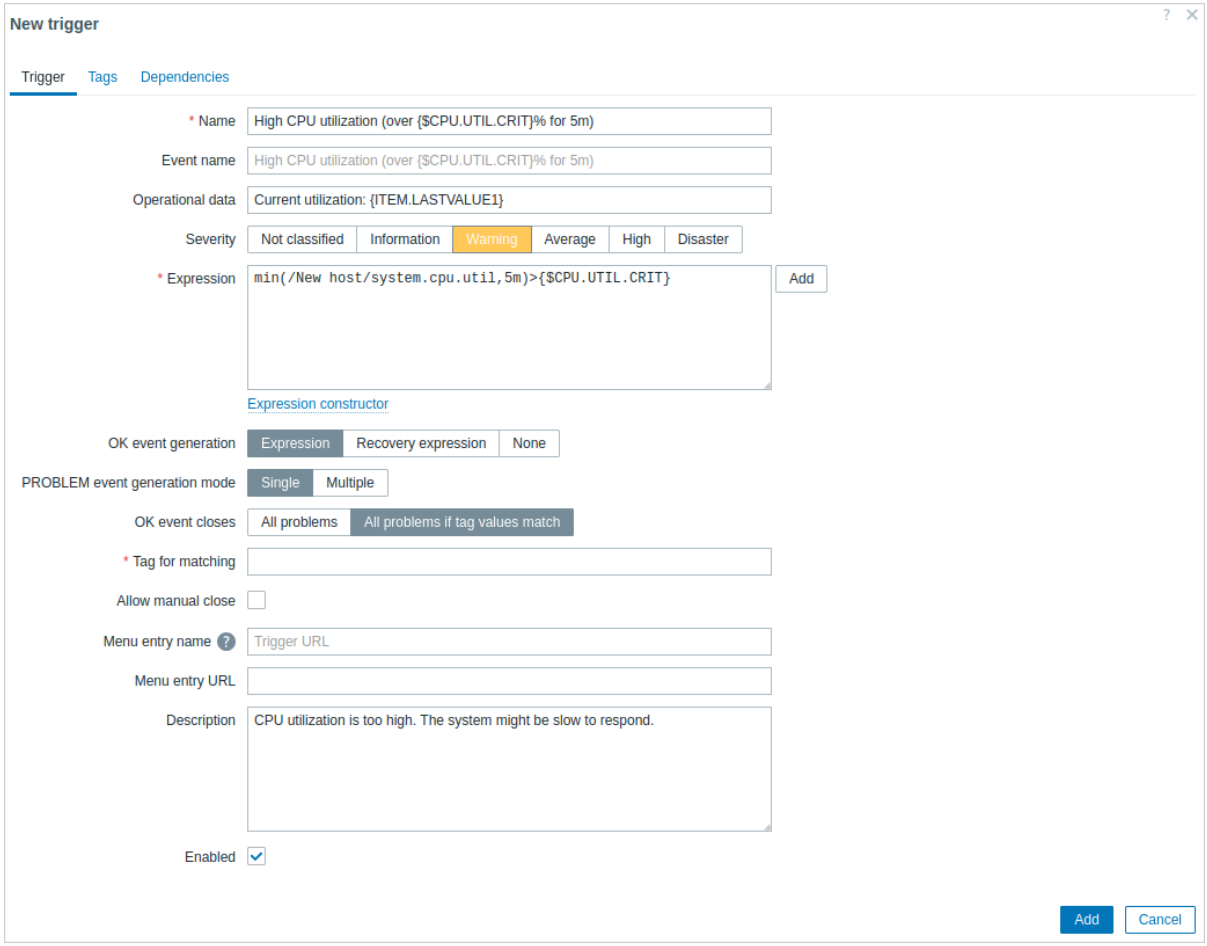
トリガーを設定するには、以下の手順を実行します。

- データ収集 > ホストに移動します
- ホストの行でトリガーをクリックします
- 右側のトリガーの作成をクリックします（または既存のトリガーを編集する場合はトリガー名をクリックします）
- フォームにトリガーのパラメータを入力します

トリガーおよびその計算タイミングの一般情報も参照してください。

設定

トリガータブには、すべての基本的なトリガー属性が含まれています。

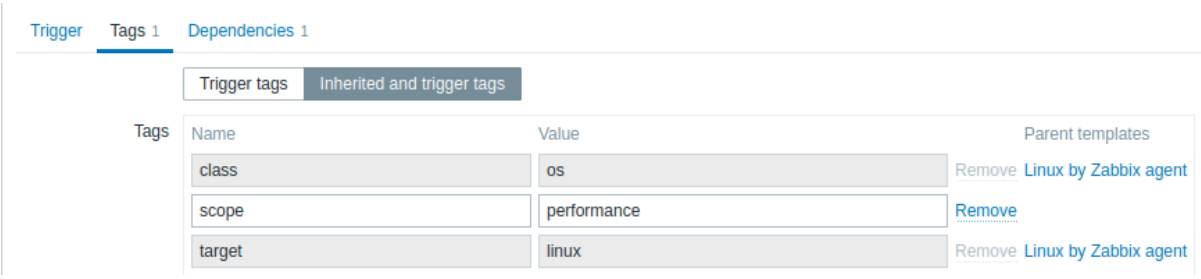


すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	トリガー名。 サポートされているマクロは、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{ITEM.VALUE}、{ITEM.VALUE.AGE}、{ITEM.VALUE.DATE}、{ITEM.VALUE.TIME}、{ITEM.VALUE.TIMESTAMP}、{ITEM.LASTVALUE}、{ITEM.LASTVALUE.AGE}、{ITEM.LASTVALUE.DATE}、{ITEM.LASTVALUE.TIME}、{ITEM.LASTVALUE.TIMESTAMP}、{ITEM.LOG.*}、および {$MACRO} ユーザーマクロです。 **$1, $2...$9** マクロを使用して、式の最初、2 番目...9 番目の定数を参照できます。 注: $1-$9 マクロは、比較的単純で明確な式の定数を参照する場合に正しく解決されます。たとえば、名前「Processor load above $1 on {HOST.NAME}」は、式が last(/New host/system.cpu.load[percpu,avg1])>5 の場合、自動的に「Processor load above 5 on New host」に変更されます。
イベント名	定義されている場合、この名前がトリガー名の代わりに問題イベント名の作成に使用されます。 イベント名は、問題データを含む意味のあるアラートを作成するために使用できます（例を参照）。 トリガー名と同じマクロセットに加え、{TIME}、{TIMESTAMP}、{?EXPRESSION} 式マクロがサポートされています。
運用データ	運用データでは、マクロとともに任意の文字列を定義できます。マクロは監視 > 問題でリアルタイムデータに動的に解決されます。トリガー名のマクロ（上記参照）は、問題発生時の値に解決され、静的な問題名の基礎となりますが、運用データのマクロは、最新情報を動的に表示する機能を維持します。トリガーレベルで運用データが設定されていない場合、式内のすべてのアイテムの最新値が表示されます。 トリガー名と同じマクロセットがサポートされています。
深刻度	ボタンをクリックして必要なトリガー深刻度を設定します。
式	問題の条件を定義するために使用される論理式。 時間サフィックスおよびメモリサイズサフィックスがサポートされています。 式に含まれるすべての条件が満たされた後、つまり式が TRUE と評価された場合に問題が作成されます。追加の復旧条件が復旧式で指定されていない限り、式が FALSE と評価されるとすぐに問題は解決されます。

パラメータ	説明
OK イベント生成	OK イベント生成オプション： 式 - 問題イベントと同じ式に基づいて OK イベントが生成されます； 復旧式 - 問題式が FALSE と評価され、復旧式が TRUE と評価された場合に OK イベントが生成されます； なし - この場合、トリガーは自動的に OK 状態に戻ることはありません。
復旧式	元の問題式がすでに FALSE と評価された後、問題が解決される前に満たす必要がある追加の条件を定義する論理式（オプション）。 復旧式は、トリガーのヒステリシスに便利です。問題式がまだ TRUE の場合、復旧式だけで問題を解決することはできません。 このフィールドは、「OK イベント生成」で「復旧式」が選択されている場合にのみ利用できます。
PROBLEM イベント生成モード	問題イベントを生成するモード： 単一 - トリガーが初めて「問題」状態になったときに単一のイベントが生成されます； 複数 - トリガーのすべての「問題」評価時にイベントが生成されます。
OK イベントでクローズ	OK イベントでクローズするかどうかを選択します： すべての問題 - このトリガーのすべての問題； タグ値が一致するすべての問題 - イベントタグ値が一致するトリガー問題のみ。
一致するタグ	イベントの相関に使用するイベントタグ名を入力します。 このフィールドは、「OK イベントでクローズ」プロパティで「タグ値が一致するすべての問題」が選択されている場合に表示され、この場合は必須です。
手動クローズを許可	このトリガーによって生成された問題イベントの手動クローズを許可する場合はチェックします。 手動クローズは、問題イベントの確認時に可能です。
メニューエントリ名	空でない場合、ここに入力された名前（最大 64 文字）は、メニューエントリ *URL* パラメータで指定されたトリガー URL のラベルとして、いくつかのフロントエンドの場所で使用されます。空の場合、デフォルト名 *Trigger URL* が使用されます。 トリガー名と同じマクロセットに加え、{EVENT.ID}、{HOST.ID}、{TRIGGER.ID} がサポートされています。
メニューエントリ *URL*	空でない場合、ここに入力された URL（最大 2048 文字）は、イベントメニューのリンクとして、たとえば監視 > 問題や問題ダッシュボードウィジェットで問題名をクリックしたときなど、いくつかのフロントエンドの場所で利用できます。 トリガー名と同じマクロセットに加え、{EVENT.ID}、{HOST.ID}、{TRIGGER.ID} がサポートされています。注：シークレット値を持つユーザーマクロは URL で解決されません。
説明	このトリガーに関する詳細情報を提供するためのテキストフィールド。特定の問題の修正手順、担当者の連絡先などを含めることができます。 トリガー名と同じマクロセットがサポートされています。
有効	このボックスのチェックを外すと、必要に応じてトリガーが無効になります。 無効なトリガーの問題はフロントエンドに表示されなくなりますが、削除はされません。

タグタブでは、トリガーレベルのタグを定義できます。このトリガーのすべての問題は、ここで入力した値でタグ付けされます。



さらに、継承およびトリガータグオプションを使用すると、トリガーがそのテンプレートから来ている場合、テンプレートレベルで定義されたタグを表示できます。同じタグを持つ複数のテンプレートがある場合、これらのタグは 1 回だけ表示され、テンプレート名はカンマで区切られます。トリガーはホストレベルのタグを「継承」して表示することはありません。

パラメータ	説明
名前/値	トリガーイベントをマークするためのカスタムタグを設定します。 タグはタグ名と値のペアです。名前だけを使用することも、値とペアにすることもできます。トリガーには同じ名前で異なる値の複数のタグを持たせることができます。 イベントタグでは、ユーザーマクロ、コンテキスト付きユーザーマクロ、ローレベルディスカバリーマクロ、および {{ITEM.VALUE}}、{{ITEM.LASTVALUE}}、ローレベルディスカバリーマクロを使用したマクロ関数がサポートされています。ローレベルディスカバリーマクロはマクロコンテキスト内で使用できます。 {TRIGGER.ID} マクロはトリガータグ値でサポートされています。これは、トリガープロトタイプから作成されたトリガーを識別したり、たとえばメンテナンス中にこれらのトリガーからの問題を抑制したりするのに役立ちます。 展開された値の合計長が 255 を超える場合は、255 文字に切り捨てられます。 イベントタグでサポートされているすべてのマクロを参照してください。 イベントタグは、イベントの相関、アクション条件に使用でき、監視 > 問題または問題ウィジェットでも表示されます。

依存関係タブには、トリガーのすべての依存関係が含まれています。

追加をクリックして新しい依存関係を追加します。

Note:
既存のトリガーを開いてクローンボタンを押し、別の名前で保存することで、トリガーを設定することもできます。

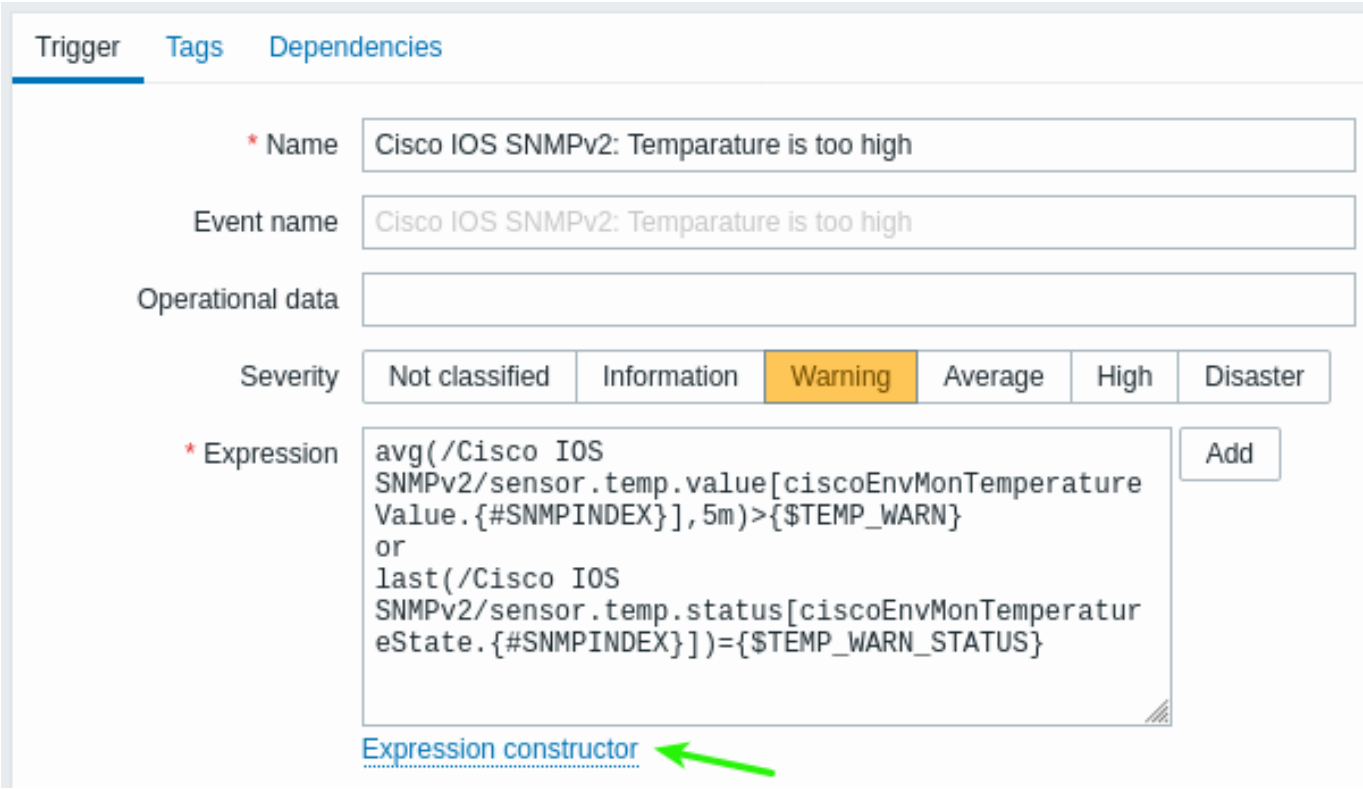
評価式のテスト

設定した trigger 式が、受け取った値によってどのような式結果になるかをテストすることが可能である。

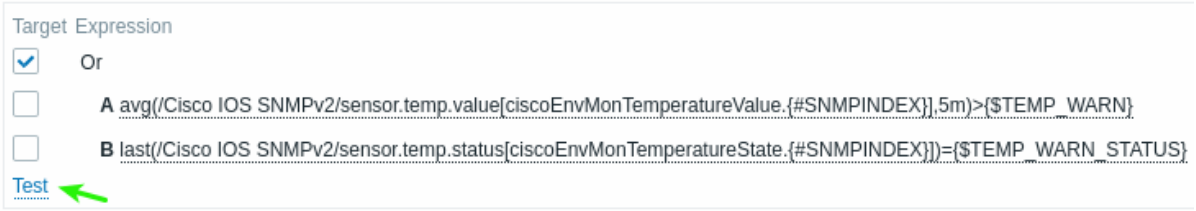
例として、公式テンプレートにある以下の式を挙げる。

```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],5m)>{$TEMP_WARN}
または
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}])={$TEMP_WARN_STATUS}
```

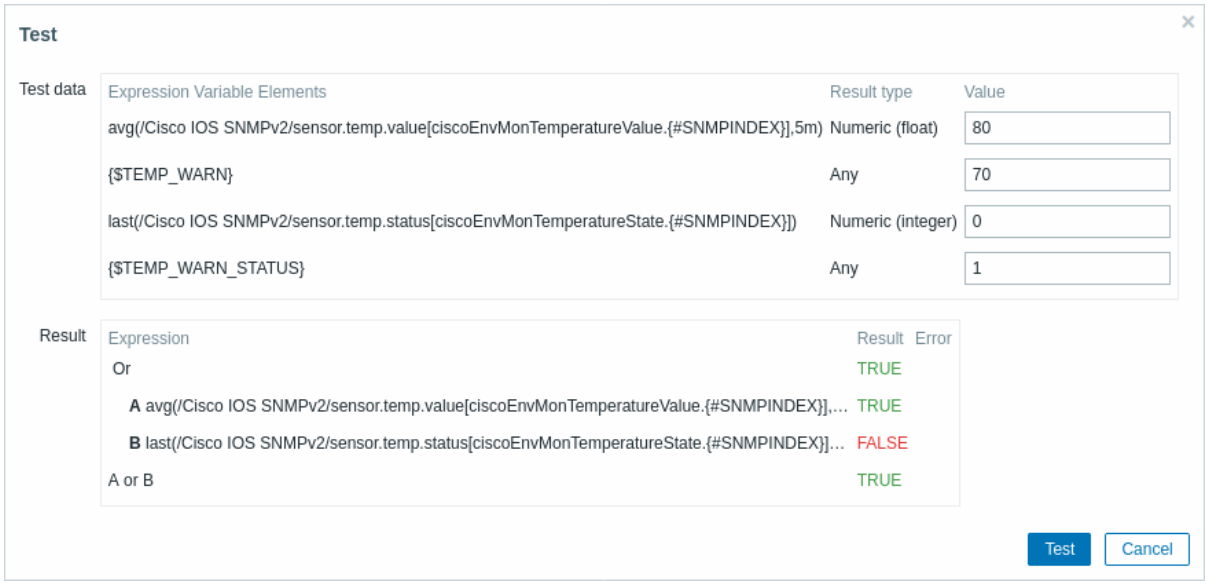
式をテストするには、式フィールドの下にある *Expression constructor* をクリックします。



Expression コンストラクタでは、個々の式がすべてリストアップされます。テスト・ウィンドウを開くには、
 式リストの下にある *Test* をクリックします。



テスト画面では、サンプル値（この例では'80'，'70'，'0'，'1'）を入力し、*Test* ボタンをクリックすると、式の結果を確認することができます。



個々の表現だけでなく、全体の表現の結果も見ることができます。

"TRUE " は、指定された式が正しいことを意味する。このケース A では、"80" は {$TEMP_WARN} の指定値 (この例では"70" です) より大きい。
 予想通り、"TRUE " の結果が表示される。

"FALSE " は、指定された式が正しくないことを意味します。この場合この特定のケース B では、
 {$TEMP_WARN_STATUS} の"1 " が"1 " と等しい必要がある。"1" は、指定された値、この例では"0 " と等しくなければならない。
 予想通り、"FALSE" の結果が表示される。

式のタイプは"OR " である。指定された条件のうち少なくとも 1 つ（この例では A または B）が
 指定された条件（この例では A または B）の少なくとも 1 つが"TRUE" であれば、全体の結果も"TRUE" となります。
 つまり、現在値が警告値を超えており、問題が発生していることを意味します。

2 トリガー条件式

概要

トリガーで使用される式は非常に柔軟です。これらを使用して、監視している統計情報に関する複雑な論理テストを作成できます。

単純な式では、いくつかのパラメータを持つ関数がアイテムに適用されます。関数は、演算子と定数を使用してしきい値と比較される結果を返します。

単純で有用な式の構文は、`function(/host/key,parameter)<operator><constant>` です。

例:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

これは、過去 5 分間に受信したバイト数が常に 100 キロバイトを超えていた場合にトリガーされます。

構文はまったく同じですが、機能的な観点からは、トリガー式には 2 つのタイプがあります:

- 問題式 - 問題の条件を定義します
- 復旧式 (オプション) - 問題解決の追加条件を定義します

問題式のみを定義する場合、この式は問題のしきい値と問題の復旧しきい値の両方として使用されます。問題式が TRUE と評価されると、問題が発生します。問題式が FALSE と評価されると、問題は解決されます。

問題式と補足的な復旧式の両方を定義する場合、問題の解決はより複雑になります: 問題式が FALSE であるだけでなく、復旧式も TRUE でなければなりません。これはヒステリシスを作成し、トリガーのフラッピングを回避するのに役立ちます。

Note:
復旧式で {TRIGGER.VALUE} マクロを使用しても非効率的です。なぜなら、この式はトリガーが「問題」状態のときにのみ評価されるためです。そのため、式を評価している間、{TRIGGER.VALUE} は常に「1」(「問題」状態を示す) になります。

関数

関数を使用すると、収集した値の計算（平均、最小、最大、合計）、文字列の検索、現在時刻やその他の要素の参照が可能です。

サポートされている関数の完全なリストが利用可能です。

通常、関数は比較のために数値を返します。文字列を返す場合は、**=** および **<>** 演算子で比較が可能です（例を参照）。

関数パラメータ

関数パラメータでは、以下を指定できます。

- ホストとアイテムキー（ホストアイテムの履歴のみを参照する関数の場合）
- 関数固有のパラメータ
- 他の式（ホストアイテムの履歴を参照する関数では利用できません。例については他の式を参照してください）

ホストとアイテムキーは /host/key の形式で指定できます。
最初のパラメータでホスト名を省略する（つまり function(//key,parameter,...) のように記述する）ことは、特定のコンテキストでのみサポートされています。

- 計算アイテムの計算式内
- 式マクロ内（以下で使用可能）:
  - イベント名フィールド
  - グラフ名
  - ”ホスト”および”トリガー”のマップ要素のラベル

これらのコンテキストでは、{HOST.HOST}マクロも使用できます。
イベント名フィールドや”トリガー”マップ要素の場合、トリガー式内の特定のアイテムを参照するために {HOST.HOST<1-9>} を使用できます。
これらのコンテキストでホスト名を省略したり、{HOST.HOST} に置き換えたりした場合、参照はトリガー式内の最初のアイテムまたはグラフの最初のアイテムを指します。
これらのサポートされているコンテキスト以外でトリガー式内のホスト名を省略すると、エラーとなります。
イベント名マクロでのダブルスラッシュの使用例については、例 18を参照してください。

参照されるアイテムは、サポートされている状態でなければなりません（**nodata()** 関数のみ、サポートされていないアイテムにも計算されます）。

他のトリガー式を関数パラメータとして使用する場合、トリガー内では履歴関数以外に制限されますが、この制限は計算アイテムには適用されません。

関数固有のパラメータ

関数固有のパラメータはアイテムキーの後に配置され、アイテムキーとはカンマで区切られます。これらのパラメータの完全なリストについては、サポートされている関数を参照してください。

ほとんどの数値関数は、パラメータとして時間を受け取ります。秒または時間のサフィックスを使用して時間を指定できます。ハッシュマーク（#）を前に付けると、パラメータの意味が異なります。

式	説明
sum(/host/key,**10m)**	直近 10 分間の値の合計
sum(/host/key,**#10)**	直近 10 個の値の合計

ハッシュマーク付きのパラメータは、**last** 関数では意味が異なり、N 番目に前の値を示します。たとえば、値が 30, 70, 20, 60, 50（最新から最も古い順）であれば:

- last(/host/key,#2) は’70’ を返します
- last(/host/key,#5) は’50’ を返します

時間シフト

関数パラメータとして時間または値のカウントを指定する場合、オプションで時間シフトがサポートされています。このパラメータを使用すると、過去のある期間のデータを参照できます。

時間シフトは now で始まり（現在時刻を指定）、+N< 時間単位 > または -N< 時間単位 > を続けて N 単位の時間を加算または減算します。

例えば、avg(/host/key,1h:now-1d) は、1 日前の 1 時間の平均値を返します。

Attention:
月（M）や年（y）で指定した時間シフトは、トレンド関数でのみサポートされます。他の関数では、秒（s）、分（m）、時間（h）、日（d）、週（w）がサポートされています。

絶対時間期間での時間シフト

絶対時間期間は、時間シフトパラメータでサポートされています。例えば、1 日の場合は 0 時から 0 時、1 週間の場合は月曜日から日曜日、1 ヶ月の場合は月初から月末までです。

絶対時間期間での時間シフトは now で始まり（現在時刻を指定）、任意の数の時間操作を続けます。/< 時間単位 > は時間単位の開始と終了を定義します（例：1 日の場合は 0 時から 0 時）、+N< 時間単位 > または -N< 時間単位 > は N 単位の時間を加算または減算します。

時間シフトの値は 0 以上である必要があり、時間期間の最小値は 1 であることに注意してください。

パラメータ	説明
1d:now/d	昨日
1d:now/d+1d	今日
2d:now/d+1d	直近 2 日間
1w:now/w	先週
1w:now/w+1w	今週

その他の式

関数のパラメータには、次の構文のように他の式を含めることができます。

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

関数がアイテムの履歴を参照する場合、他の式は使用できないことに注意してください。たとえば、次の構文は許可されていません。

~~min(/host/key,#5*10)~~

演算子

トリガーでサポートされている演算子は次のとおりです **(**実行の優先順位が高い順**)**:

優先度	演算子	定義	不明な値に関する注意	オペランドを float に強制キャスト [1]
1	**-**	単項マイナス	**-**Unknown → Unknown	はい
2	**not**	論理 NOT	**not** Unknown → Unknown	はい
3	*	乗算	0 * Unknown → Unknown (はい、Unknown、0 以外 - 算術演算で Unknown を失わないため) 1.2 * Unknown → Unknown	はい
	/	除算	Unknown / 0 → エラー Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	はい
4	+	算術加算	1.2 + Unknown → Unknown	はい
	-	算術減算	1.2 - Unknown → Unknown	はい
5	<	より小さい。演算子は次のように定義されます: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	はい
	<=	以下。演算子は次のように定義されます: A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	はい

優先度	演算子	定義	不明な値に関する注意	オペランドを float に強制キャスト [1]
	>	より大きい。演算子は次のように定義されます: A>B ⇔ (A>B+0.000001)		はい
	>=	以上。演算子は次のように定義されます: A>=B ⇔ (A≥B-0.000001)		はい
6	**=**	等しい。演算子は次のように定義されます: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		いいえ [1]
	<>	等しくない。演算子は次のように定義されます: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		いいえ [1]
7	**and**	論理 AND	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	はい
8	**or**	論理 OR	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	はい

[1] 文字列オペランドは、以下の場合でも数値にキャストされます:

- 他のオペランドが数値の場合
- **=** または **<>** 以外の演算子がオペランドに使用されている場合

(キャストに失敗した場合 - 数値オペランドは文字列オペランドにキャストされ、両方のオペランドが文字列として比較されます。)

not、**and**、**or** 演算子は大文字と小文字を区別し、小文字でなければなりません。また、スペースまたは括弧で囲む必要があります。

単項 **-** および **not** を除くすべての演算子は、左から右への結合性を持ちます。単項 **-** および **not** は非結合性です（つまり、**--1** や **not not 1** の代わりに **-(-1)** や **not (not 1)** を使用する必要があります）。

評価結果:

- **<**、**<=**、**>**、**>=**、**=**、**<>** 演算子は、指定された関係が真であればトリガー式で'1' を、偽であれば'0' を返します。少なくとも 1 つのオペランドが Unknown の場合、結果は Unknown になります。
- **and** は既知のオペランドの場合、両方のオペランドが'0' と等しくない場合は'1' を返し、それ以外の場合は'0' を返します。不明なオペランドの場合、**and** は 1 つのオペランドが'0' と等しい場合のみ'0' を返し、それ以外の場合は'Unknown' を返します。
- **or** は既知のオペランドの場合、いずれかのオペランドが'0' と等しくない場合は'1' を返し、それ以外の場合は'0' を返します。不明なオペランドの場合、**or** は 1 つのオペランドが'0' と等しくない場合のみ'1' を返し、それ以外の場合は'Unknown' を返します。
- 論理否定演算子 **not** の既知のオペランドに対する結果は、オペランドの値が'0' と等しくない場合は'0'、オペランドの値が'0' と等しい場合は'1' です。不明なオペランドの場合、**not** は'Unknown' を返します。

値のキャッシュ

トリガーの評価に必要な値は、Zabbix サーバーによってキャッシュされます。このため、サーバーの再起動後しばらくの間、トリガーの評価によってデータベースの負荷が高くなります。アイテムの履歴値が削除された場合（手動またはハウスキーパーによる）、値のキャッシュはクリアされません。そのため、サーバーはトリガー関数で定義された期間より古い値になるか、サーバーが再起動されるまでキャッシュされた値を使用します。

Note:
キャッシュに最近のデータがなく、関数でクエリ期間が定義されていない場合、Zabbix はデフォルトで過去 1 週間までさかのぼってデータベースから履歴値を取得します。

trigger の例

例 1

Zabbix サーバーのプロセッサ負荷が高すぎます。

```
last(/Zabbix server/system.cpu.load[all,avg1])>5
```

'last()' 関数を使用することで、最新の値を参照しています。/Zabbix server/system.cpu.load[all,avg1] は、監視パラメータの短縮名を示します。ホストが「Zabbix server」で、監視しているキーが「system.cpu.load[all,avg1]」であることを指定しています。最後に、>5 は、Zabbix サーバーからの最新のプロセッサ負荷測定値が 5 より大きい場合に、トリガーが PROBLEM 状態になることを意味します。

例 2

www.example.com が過負荷です。

```
last(/www.example.com/system.cpu.load[all,avg1])>5 or min(/www.example.com/system.cpu.load[all,avg1],10m)>
```

現在のプロセッサ負荷が 5 を超えているか、直近 10 分間のプロセッサ負荷が 2 を超えていた場合に、この式は真となります。

例 3

/etc/passwd が変更されました。

```
last(/www.example.com/vfs.file.cksum[/etc/passwd],#1)<>last(/www.example.com/vfs.file.cksum[/etc/passwd],#
```

/etc/passwd のチェックサムの前回の値が最新の値と異なる場合、式は真となります。

同様の式は、/etc/passwd、/etc/inetd.conf、/kernel などの重要なファイルの変更を監視するのに役立ちます。

例 4

誰かがインターネットから大きなファイルをダウンロードしています。

min 関数の使用例:

```
min(/www.example.com/net.if.in[eth0,bytes],5m)>100K
```

この式は、eth0 で受信したバイト数が直近 5 分間で 100KB を超えた場合に真となります。

例 5

クラスタ化された SMTP サーバーの両方のノードがダウンしています。

1 つの式で 2 つの異なるホストを使用していることに注意してください:

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

この式は、smtp1.example.com と smtp2.example.com の両方の SMTP サーバーがダウンしている場合に真となります。

例 6

Zabbix エージェントをアップグレードする必要があります。

find() 関数の使用例:

```
find(/example.example.com/agent.version,,"like","beta8")=1
```

Zabbix エージェントのバージョンが beta8 の場合、この式は真となります。

例 7

サーバーに到達できません。

```
count(/example.example.com/icmpping,30m,,"0")>5
```

この式は、ホスト”example.example.com” が過去 30 分間に 5 回以上到達不能であった場合に真となります。

例 8

過去 3 分間にハートビートがありません。

nodata() 関数の使用例:

```
nodata(/example.example.com/tick,3m)=1
```

このトリガーを利用するには、'tick' を Zabbix のトラッパーアイテムとして定義する必要があります。ホストは zabbix_sender を使用して、このアイテムのデータを定期的に送信する必要があります。180 秒以内にデータが受信されない場合、トリガー値は PROBLEM になります。

注意'nodata' は任意のアイテムタイプで使用できます。

例 9

夜間の CPU アクティビティ。

time() 関数の使用例:

```
min(/Zabbix server/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

このトリガーは、夜間 (00:00 - 06:00) のみ障害状態に変化する可能性があります。

例 10

例外を除いて、いつでも CPU アクティビティを監視します。

time() 関数と **not** 演算子の使用例:

```
min(/zabbix/system.cpu.load[all,avg1],5m)>2
and not (dayofweek()=7 and time()>230000)
and not (dayofweek()=1 and time()<010000)
```

このトリガーは、週の切り替わり時の 2 時間 (日曜日 23:00～月曜日 01:00) を除き、いつでも障害状態に変化する可能性があります。

例 11

クライアントのローカル時刻が Zabbix サーバーの時刻と同期しているかどうかを確認します。

fuzzytime() 関数の使用例:

```
fuzzytime(/MySQL_DB/system.localtime,10s)=0
```

MySQL_DB サーバーと Zabbix サーバーのローカル時刻が 10 秒以上ずれている場合、トリガーは障害状態に変わります。'system.localtime' は、Zabbix エージェントでパッシブチェックとして設定する必要があります。Zabbix agent 2 ではアクティブチェックとして設定することもできます。

例 12

今日の平均負荷を昨日の同じ時間の平均負荷と比較します (time shift として now-1d を使用)。

```
avg(/server/system.cpu.load,1h)/avg(/server/system.cpu.load,1h:now-1d)>2
```

このトリガーは、直近 1 時間の平均負荷が昨日の同じ時間の平均負荷の 2 倍を超えた場合に発生します。

例 13

別のアイテムの値を使用してトリガーの閾値を取得する:

```
last(/Template PfSense/hrStorageFree[{#SNMPVALUE}])<last(/Template PfSense/hrStorageSize[{#SNMPVALUE}])*0.
```

空きストレージが 10 パーセント未満になるとトリガーが発生します。

例 14

評価結果を使用して、しきい値を超えたトリガーの数を取得します。

```
(last(/server1/system.cpu.load[all,avg1])>5) + (last(/server2/system.cpu.load[all,avg1])>5) + (last(/serve
```

この式のトリガーのうち少なくとも 2 つが障害状態の場合にトリガーが発生します。

例 15

2 つのアイテムの文字列値を比較します - ここでのオペランドは文字列を返す関数です。

問題: 異なるホストで Ubuntu のバージョンが異なる場合にアラートを作成する

```
last(/NY Zabbix server/vfs.file.contents[/etc/os-release])<>last(/LA Zabbix server/vfs.file.contents[/etc/
```

例 16

2 つの文字列値を比較する - オペランドは以下の通りです。

- 文字列を返す関数
- マクロと文字列の組み合わせ

問題: DNS クエリの変更を検出する

アイテムキーは次のとおりです。

```
net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1]
```

マクロは次のように定義されています。

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

通常は次のように返されます。

```
example.com           MX       0 mail.example.com
```

したがって、DNS クエリの結果が期待値から逸脱したかどうかを検出するトリガー式は次のようになります。

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

2 番目のオペランドの周りに引用符があることに注意してください。

例 17

2 つの文字列値を比較する - オペランドは以下の通りです:

- 文字列を返す関数
- 特殊文字 \ および" を含む文字列定数

問題: /tmp/hello ファイルの内容が以下と等しいか検出する:

```
\" //hello ?\"
```

オプション 1) 文字列を直接記述する:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

文字列を直接比較する場合、\ および" 文字がエスケープされていることに注意してください。

オプション 2) マクロを使用する

```
{$HELLO_MACRO} = \" //hello ?\"
```

式では:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

例 18

長期的な期間を比較する。

問題: Exchange サーバーの負荷が先月より 10% 以上増加した

```
trendavg(/Exchange/system.cpu.load,1M:now/M)>1.1*trendavg(/Exchange/system.cpu.load,1M:now/M-1M)
```

また、トリガー設定のイベント名フィールドを使用して、意味のあるアラートメッセージを作成することもできます。たとえば、次のようなメッセージを受け取るには

"Load of Exchange server increased by 24% in July (0.69) comparing to June (0.56)"

イベント名は次のように定義する必要があります。

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

この種の問題には、トリガー設定で手動クローズを許可することも有用です。

ヒステリシス

単純な閾値ではなく、障害状態と復旧状態の間に間隔が必要な場合があります。たとえば、サーバールームの温度が 20°C を超えたときに障害を報告し、温度が 15°C 未満になるまで障害状態を維持したい場合、単純なトリガー閾値を 20°C に設定するだけでは十分ではありません。

代わりに、まず障害イベントのトリガー式（温度が 20°C を超える）を定義する必要があります。次に、追加の復旧条件（温度が 15°C 未満）を定義する必要があります。これは、トリガーを定義する際に追加の復旧式パラメータを定義することで実現します。

この場合、障害の復旧は 2 段階で行われます。

- まず、障害式（温度が 20°C を超える）が FALSE と評価される必要があります
- 次に、復旧式（温度が 15°C 未満）が TRUE と評価される必要があります

復旧式は、最初に障害イベントが解決された場合にのみ評価されます。

Warning:
復旧式が TRUE であっても、障害式が TRUE のままの場合は障害は解決されません！

例 1

サーバールームの温度が高すぎます。

障害の式:

```
last(/server/temp)>20
```

復旧の式:

```
last(/server/temp)<=15
```

例 2

空きディスク容量が少なすぎます。

障害の条件式: 直近 5 分間で 10GB 未満

```
max(/server/vfs.fs.size[/,free],5m)<10G
```

復旧の条件式: 直近 10 分間で 40GB を超える

```
min(/server/vfs.fs.size[/,free],10m)>40G
```

オペランドが不明な場合の式

一般的に、式の中でオペランドが不明（サポートされていないアイテムなど）な場合、トリガー値は即座に Unknown になります。

ただし、場合によっては不明なオペランド（サポートされていないアイテム、関数エラー）が式の評価に認められることがあります。

- nodata() 関数は、参照されるアイテムがサポートされているかどうかに関係なく評価されます。
- OR および AND を含む論理式は、不明なオペランドに関係なく、2 つの場合に既知の値に評価できます。
  - ケース **1**: "1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ..." は既知の結果（'1' または"Problem"）に評価できます。
  - ケース **2**: "0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ..." は既知の結果（'0' または"OK"）に評価できます。
    Zabbix は、このような論理式をサポートされていないアイテムを不明なオペランドとして評価しようとします。上記 2 つのケースでは既知の値（それぞれ"Problem" または"OK"）が生成されますが、それ以外のすべての場合、トリガーは Unknown に評価されます。
- サポートされているアイテムに対する関数評価がエラーになった場合、関数値は Unknown となり、以降の式評価で不明なオペランドとして扱われます。

不明なオペランドが「消える」可能性があるのは、上記のような論理式の場合のみです。算術式では、不明なオペランドは常に結果を Unknown に導きます（0 による除算を除く）。

Attention:
Unknown となる式は、トリガーの状態（"Problem/OK"）を変更しません。したがって、"Problem" だった場合（ケース 1 参照）、既知の部分が解決されても（'1' が'0' になっても）、式が Unknown に評価されるため、トリガーの状態は変わりません。

複数のサポートされていないアイテムを含むトリガー式が Unknown に評価された場合、フロントエンドのエラーメッセージは最後に評価されたサポートされていないアイテムを参照します。

3 トリガーの依存関係

概要

ホストの可用性が他のホストに依存する場合があります。ルーターの背後にあるサーバーは、ルーターがダウンすると到達不能になります。両方にトリガーが設定されていると、2 つのホストがダウンしたという通知が届くかもしれませんが、実際にはルーターだけが原因だったということになります。

このような場合、ホスト間の依存関係が役立ちます。依存関係を設定すると、依存するホストの通知を保留し、根本的な問題の通知のみを送信できます。

Zabbix はホスト間の依存関係を直接サポートしていませんが、より柔軟な方法であるトリガー依存関係を定義できます。トリガーは、1 つまたは複数の依存するトリガーを持つことができます。

この簡単な例では、サーバーのトリガー設定フォームを開き、ルーターの該当するトリガーに依存するように設定します。このような依存関係がある場合、依存するトリガーが「PROBLEM」状態である限り、サーバートリガーは状態を変更せず、依存するアクションは実行されず、通知も送信されません。

サーバーとルーターの両方がダウンしていて依存関係がある場合、Zabbix は依存するトリガーのアクションを実行しません。

親トリガーが PROBLEM 状態の間は、その依存トリガーが信頼できない値を報告する可能性があります。したがって、親トリガー（上記の例ではルーター）が以下のいずれかの状態になるまで、依存するトリガーは再評価されません。

- 「PROBLEM」から「OK」状態に戻る
- 「PROBLEM」から「UNKNOWN」状態に変わる
- 手動、相関、または日付と時刻やnodata()関数の助けでクローズされる
- 依存するトリガーに関与していないアイテムの値によって解決される
- 無効化されている、無効なアイテムがある、または無効なアイテムホストがある

上記のいずれの場合も、依存するトリガー（サーバー）は新しいメトリックが受信されたときにのみ再評価されます。つまり、依存するトリガーはすぐには更新されない場合があります。

また、

- トリガー依存関係は、循環依存関係にならない限り、任意のホストトリガーから他のホストトリガーに追加できます。
- トリガー依存関係は、あるテンプレートから別のテンプレートに追加できます。テンプレート A のトリガーがテンプレート B のトリガーに依存している場合、テンプレート A はテンプレート B と一緒にのみホスト（または他のテンプレート）にリンクできますが、テンプレート B は単独でホスト（または他のテンプレート）にリンクできます。
- トリガー依存関係は、テンプレートトリガーからホストトリガーに追加できます。この場合、そのようなテンプレートをホストにリンクすると、依存していたトリガーテンプレートと同じトリガーに依存するホストトリガーが作成されます。これにより、たとえば、いくつかのトリガーがルーター（ホスト）トリガーに依存するテンプレートを持つことができます。このテンプレートにリンクされたすべてのホストは、その特定のルーターに依存します。
- トリガー依存関係は、ホストトリガーからテンプレートトリガーには追加できません。
- トリガー依存関係は、トリガープロトタイプから別のトリガープロトタイプ（同じ低レベルディスカバリールール内）または実際のトリガーに追加できます。トリガープロトタイプは、異なる LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーには依存できません。ホストトリガープロトタイプは、テンプレートのトリガーには依存できません。

設定

依存関係を定義するには、トリガーの設定フォームで「依存関係」タブを開きます。「依存関係」ブロックで追加をクリックし、トリガーが依存する 1 つまたは複数のトリガーを選択します。

更新をクリックします。これで、トリガーのリストに依存関係が表示されます。

複数の依存関係の例

例えば、ホストは Router2 の背後にあり、Router2 は Router1 の背後にあります。

```
Zabbix - Router1 - Router2 - Host
```

Router1 がダウンしている場合、ホストと Router2 も明らかに到達不能ですが、ホスト、Router1、Router2 がすべてダウンしているという 3 つの通知を受け取るのは過剰です。

この場合、2 つの依存関係を定義します。

```
「ホストがダウンしている」トリガーは「Router2がダウンしている」トリガーに依存する
「Router2がダウンしている」トリガーは「Router1がダウンしている」トリガーに依存する
```

「ホストがダウンしている」トリガーのステータスを変更する前に、Zabbix は対応するトリガーの依存関係をチェックします。そのような依存関係が見つかり、それらのトリガーの 1 つが「問題」状態の場合、トリガーのステータスは変更されず、アクションは実行されず、通知も送信されません。

Zabbix はこのチェックを再帰的に実行します。Router1 または Router2 が到達不能な場合、ホストのトリガーは更新されません。

4 トリガーの深刻度

トリガーの深刻度は、トリガーの重要度レベルを表します。



Zabbix は、次のデフォルトのトリガー深刻度をサポートしています。

深刻度	色	説明
未分類 (Not classified)	灰色	イベントの深刻度レベルが不明、決定されていない、通常の監視範囲に含まれていない場合などに使用できます。たとえば、初期構成中、将来の評価のためのプレースホルダーとして、または統合プロセスの一部として使用できます。
情報 (Information)	水色	すぐに対処する必要はないが、貴重な洞察を提供できる情報イベントに使用できます。
警告 (Warning)	黄色	調査や対応が必要になる可能性があるが、重大ではない潜在的な問題を示すために使用できます。
軽度の障害 (Average)	オレンジ	さらなる問題を防ぐために比較的早く対処する必要がある重大な問題を示すために使用できます。
重度の障害 (High)	淡い赤	重大な中断を避けるためにすぐに対処する必要がある重大な問題を示すために使用できます。
致命的な障害 (Disaster)	赤	システム停止やデータ損失などを防ぐためにすぐに対処する必要がある重大なインシデントを示すために使用できます。

Note:
トリガーの深刻度名と色はカスタマイズできます。

深刻度は、以下の用途で使用されます:

- トリガーの視覚的表現 - 深刻度に応じて異なる色
- グローバルアラームの音声 - 深刻度に応じて異なる音声
- ユーザーメディア - 深刻度に応じて異なるメディア (通知チャネル) (たとえば、深刻度が重度の障害および致命的な障害のトリガーの場合は SMS、その他の深刻度のトリガーの場合は電子メール)
- トリガーの深刻度に対する条件によるアクションの制限

5 トリガーの深刻度のカスタマイズ

GUI に関連づけられているトリガーの深刻度の名称および色については、管理 → 一般 → トリガー表示オプションで設定することができます。色はすべての GUI テーマで共有されます。

カスタマイズされた深刻度名の翻訳

Attention:
Zabbix フロントエンドの翻訳を使用している場合、カスタマイズした深刻度名は、デフォルトで翻訳された名前より優先されます。

デフォルトのトリガーの深刻度名は、すべてのロケールにおいて翻訳可能です。深刻度名を変更した場合、すべてのロケールにおいてカスタム名が使用されるため、手動での追加翻訳が必要です。

カスタム深刻度名の翻訳手順

- 必要なカスタム深刻度名、例えば'Important' を設定します。
- <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po を編集します。
- 次の 2 行を追加します。

```
msgid "Important"
msgstr "<翻訳する文>"
```

そしてファイルを保存してください。

- <frontend_dir>/locale/README で説明されているように.mo ファイルを作成します。

ここで **msgid** は新しいカスタム深刻度名と一致し、**msgstr** は特定の言語での翻訳でなければなりません。

この手順は深刻度名を変更するたびに実行する必要があります。

6 一括アップデート

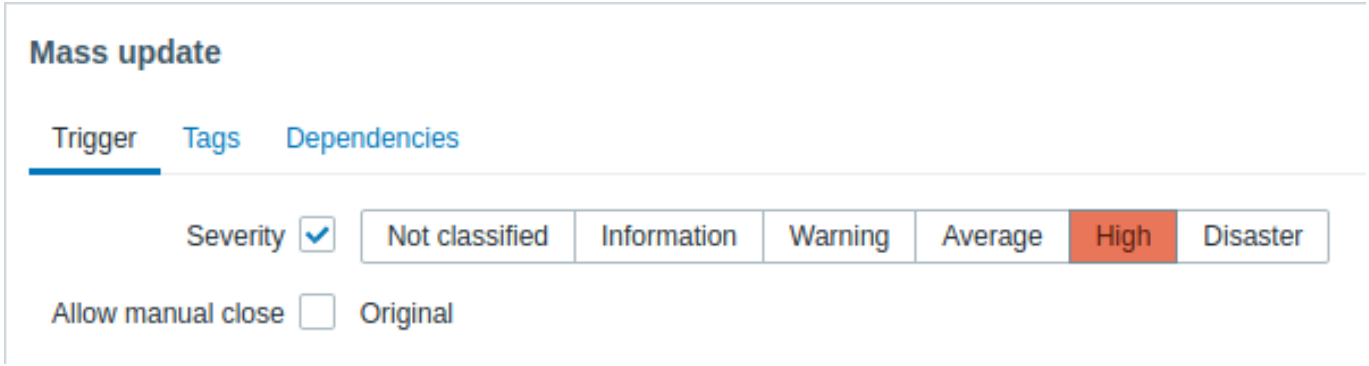
概要

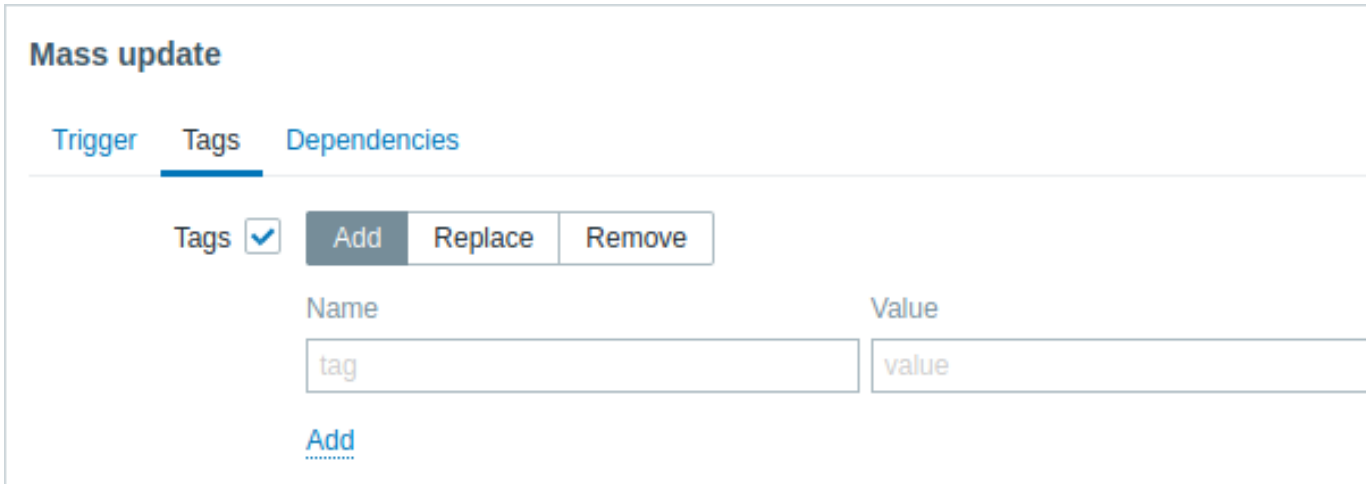
一括アップデートでは、一度に多くの trigger の属性を変更することができ、編集のために個々の trigger を開く手間を省くことができます。

一括アップデートの利用

いくつかの trigger を一括でアップデートするには、以下のようにします。

- リスト内の更新したい trigger のチェックボックスにマークを付けます。
- リストの下にある *Mass update* をクリックします。
- 必要な属性（*Trigger*、*Tags*、*Dependencies*）のあるタブに移動します。
- 更新する属性のチェックボックスにマークを付けます。

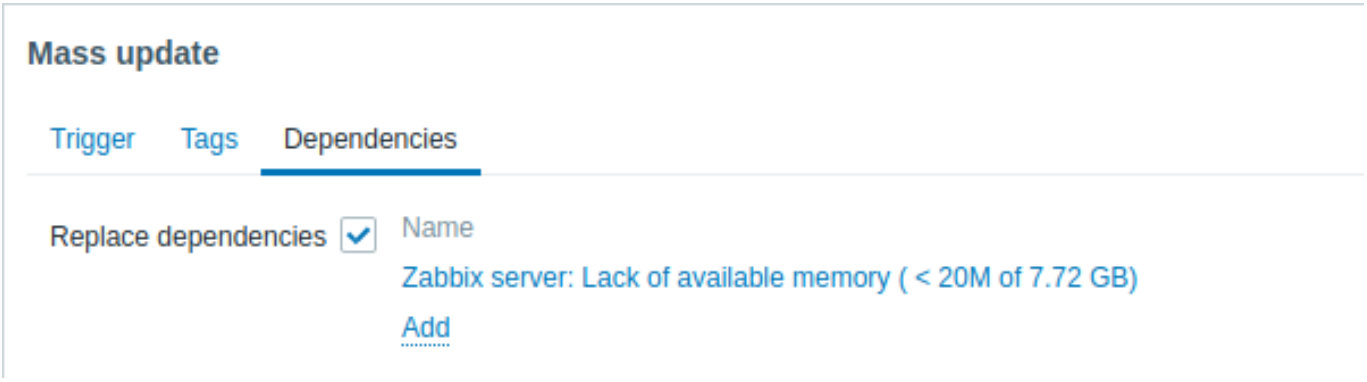




タグ更新の各ボタンを選択すると、以下のオプションが利用できます。

- *Add* - トリガーに新しいタグを追加することができます。
- *Replace* - トリガーから既存のタグを削除し、以下で指定するタグに置き換えます。
- *Remove* - 指定したタグをトリガーから削除します。

同じ名前で異なる値を持つタグは、’ 重複’ とはみなされず、同じトリガーに追加することができることに注意してください。



Replace dependencies - トリガーから既存の依存関係を削除し、指定されたものに置き換えます。

変更を適用するには、*Update* をクリックしてください。

7 予測トリガー関数

概要

時には、問題が発生する予兆があることがあります。これらの兆候を発見することで、事前に対策を講じ、問題の影響を防ぐ、
 または少なくとも最小限に抑えることができます。

Zabbix は history データに基づいて監視システムの将来の挙動を予測するツールを備えています。
 これらのツールは予兆トリガー機能で実現されます。

関数

トリガーを設定する前に、問題状態が何であるか、そしてアクションを実行するまでにどれだけの時間が必要かを定義する必要があります。その後、潜在的な望ましくない状況を通知するトリガーを設定する方法は 2 つあります。1 つ目は、「アクションまでの時間」後にシステムが問題状態になると予想される場合にトリガーを発火させる方法です。2 つ目は、システムが「アクションまでの時間」未満で問題状態に到達しそうな場合にトリガーを発火させる方法です。対応するトリガー関数は **forecast** と **timeleft** です。両関数の基礎となる統計解析は基本的に同一であることに注意してください。どちらの方法でも、同様の結果でトリガーを設定できます。

パラメータ

両方の関数はほぼ同じパラメータセットを使用します。参照用にサポートされている関数のリストを使用してください。

時間間隔

まず、Zabbix が予測を行うために分析すべき履歴期間を指定する必要があります。これは、**avg**、**count**、**delta**、**max**、**min**、**sum** 関数と同様に、`time period` パラメータとオプションのタイムシフトを使用して指定します。

予測の範囲

(**forecast** のみ)
パラメータ `time` は、Zabbix が過去のデータから見つけた依存関係を将来どこまで外挿するかを指定します。`time_shift` を使用するかどうかに関係なく、`time` は常に現在の時点からカウントされます。

到達するしきい値

(**timeleft** のみ)
パラメータ `threshold` は、分析対象のアイテムが到達しなければならない値を指定します。上からでも下からでも違いはありません。f(t)（下記参照）が決定したら、方程式 f(t) = `threshold` を解き、現在に最も近く、かつ現在より右側にある根を返します。該当する根がない場合は 1.7976931348623158E+308 を返します。

Note:
アイテムの値がしきい値に近づき、それを超えた場合、**timeleft** は交点がすでに過去にあるとみなし、`threshold` レベルとの次の交点に切り替えます（存在する場合）。ベストプラクティスとしては、予測は通常の問題診断の補完として使用し、代替として使用しないことが推奨されます。[1]

フィット関数

デフォルトの `fit` は *linear* 関数です。しかし、監視対象のシステムがより複雑な場合は、他のオプションを選択できます。

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$

fit	x = f(t)
exponential	x = a*exp(b*t)
logarithmic	x = a + b*log(t)
power	$x = a*t^b$

モード

(**forecast** のみ)
トリガー関数が評価されるたびに、指定された履歴期間からデータを取得し、指定された関数をデータにフィットさせます。したがって、データがわずかに異なれば、フィットした関数もわずかに異なります。将来の指定した時点でフィットした関数の値を単純に計算しても、今から将来のその時点までに分析対象のアイテムがどのように振る舞うかはわかりません。いくつかの `fit` オプション(*polynomial* など)では、将来の単純な値は誤解を招く可能性があります。

mode	**forecast** の結果
value	f(now + time)
max	$max_{now <= t <= now + time}$ f(t)
min	$min_{now <= t <= now + time}$ f(t)
delta	*max - min*
avg	定義に従った f(t)(now <= t <= now + time)の平均値

詳細

巨大な数値での計算を避けるために、指定された期間の最初の値のタイムスタンプに 1ns を加えたものを新しいゼロ時刻とみなします(現在のエポック時刻は 10^9 のオーダー、エポックの 2 乗は 10^{18}、倍精度は約 10^{-16})。*logarithmic* および *power* フィットでは log(t) の計算が含まれるため、すべての時刻値を正にするために 1ns が加えられます。時刻のシフトは *linear*、*polynomial*、*exponential* には(計算が容易かつより正確になることを除き)影響しませんが、*logarithmic* および *power* 関数の形状は変化します。

起こりうるエラー

関数は、次のような状況で-1 を返します。

- 指定された評価期間にデータが存在しない場合
- 数学的演算の結果が定義されていない場合[3]
- 数値的な問題(残念ながら、入力データの組み合わせによっては、倍精度浮動小数点形式の範囲や精度が不十分になることがあります)[4]

Note:
選択した近似が提供されたデータをうまく表現できていない場合や、正確な予測を行うにはデータが少なすぎる場合でも、警告やエラーは表示されません。

例とエラーへの対処

ホストの空きディスク容量がなくなりそうなときに警告を受け取るには、次のようなトリガー式を使用できます。

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

ただし、エラーコード-1 が発生し、トリガーが障害状態になる場合があります。一般的には、予測が正しく機能していないことを警告してくれるので良いことですが、なぜそうなったのかを詳しく調べる必要があります。しかし、-1 は単に過去 1 時間以内にホストの空きディスク容量のデータが取得できなかったことを意味する場合もあります。誤検知のアラートが多すぎる場合は、より複雑なトリガー式の使用を検討してください[5]。

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

forecast の場合は状況が少し複雑です。まず、-1 がトリガーを障害状態にするかどうかは、`forecast(/host/item,(...))<...` のような式か、`forecast(/host/item,(...))>...` のような式かによって異なります。

さらに、アイテム値が負になるのが通常の場合、-1 は有効な予測値となる場合があります。しかし、実際の運用でこのような状況になる可能性はほとんどありません(演算子 **=** の動作についてはこちらを参照)。したがって、-1 を障害とみなしたい場合は `... or forecast(/host/item,(...))=-1`、みなしたくない場合は `... and forecast(/host/item,(...))<>-1` を追加してください。

脚注

[1] 例えば、`timeleft(/host/item,1h,X) < 1h` のような単純なトリガーは、アイテム値が X に近づくと障害状態になり、値が X に達すると突然復旧する場合があります。問題がアイテム値が X 未満である場合は、`last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` を、アイテム値が X より大きい場合は `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h` を使用してください。

[2] 多項式の次数は 1 から 6 まで指定でき、*polynomial1* は *linear* と同等です。ただし、高次の多項式は注意して使用してください。評価期間内のデータポイント数が多項式係数の決定に必要な数より少ない場合、多項式の次数は自動的に下げられます（例：*polynomial5* が指定されているが、ポイントが 4 つしかない場合は *polynomial3* が適用されます）。

[3] 例えば、*exponential* や *power* 関数のフィッティングでは、アイテム値の log() 計算が必要です。データにゼロや負の値が含まれている場合、log() は正の値にしか定義されていないためエラーになります。

[4] *linear*、*exponential*、*logarithmic*、*power* フィットでは、必要な計算はすべて明示的に記述できます。*polynomial* では *value* のみ追加計算なしで算出できます。*avg* の計算には多項式の不定積分（解析的）が必要です。*max*、*min*、*delta* の計算には多項式の導関数（解析的）とその根の数値的な探索が必要です。f(t) = 0 の解法には多項式の根の数値的な探索が必要です。

[5] ただし、この場合-1によってトリガーが障害状態から復旧する可能性があります。完全に保護したい場合は、`timeleft(/host/vfs.fs.size[ and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)` を使用してください。

4 イベント

概要

Zabbix で生成されるイベントにはいくつかの種類があります。

- トリガーイベント - トリガーのステータスが変わるたびに (*OK*→ 障害 →*OK*)
- サービスイベント - サービスのステータスが変わるたびに (*OK*→ 障害 →*OK*)
- ディスカバリイベント - ホストまたはサービスが検出されたとき
- 自動登録イベント - アクティブなエージェントがサーバーによって自動登録されたとき
- 内部イベント - アイテム/ローレベルディスカバリルールがサポートされなくなったとき、またはトリガーが不明な状態になったとき

イベントにはタイムスタンプが付けられ、通知メールの送信などのアクションの元となることができます。

フロントエンドでイベントの詳細を表示するには、監視 → 障害へ移動してください。そこでイベントの日付と時刻をクリックすると、イベントの詳細が表示されます。

詳細については、以下を参照してください。

- トリガーイベント
- その他のイベントソース

1 トリガーイベントの生成

概要

トリガーステータスの変化は、最も頻繁かつ重要なイベントの発生源です。トリガーステータスが変化するたびに、イベントが生成されます。イベントには、トリガーステータスの変化の詳細 (いつ発生したか、新しいステータスは何か) が含まれます。

トリガーによって、障害と正常の 2 種類のイベントが作成されます。

障害イベント

次のとき、障害イベントが作成されます。

- トリガーが正常ステータスの場合、トリガー式が TRUE と評価されたときに。
- トリガーに対して複数の障害イベントの生成が有効になっている場合、トリガー式が TRUE と評価されるたびに。

正常イベント

正常イベントは関連する障害イベントを閉じます。次の 3 つのコンポーネントによって作成される場合があります。

- トリガー - ’ 正常イベントの生成’ および’ 正常イベントの終了’ 設定に基づく
- イベント相関関係
- タスクマネージャー – イベントが手動でクローズされたとき

トリガー

トリガーには、OK イベントの生成方法を制御する’OK イベント生成’ 設定があります。

- *Expression* - 問題のある状態のトリガーに対して、その式が FALSE と評価されると OK イベントが生成されます。これは、最も単純な設定で、デフォルトで有効になっています。
- *Recovery expression* - OK イベントは、その式が FALSE と評価され、回復式が TRUE と評価されたとき、問題状態のトリガーに対して生成されます。これは、トリガーの回復基準が問題基準と異なる場合に使用できます。

- *None* - OK イベントは、決して生成されません。これは、複数の問題イベント生成と組み合わせて、何かが起こったときに単に通知を送るために使用することができます。

さらに、トリガーには'OK イベントクローズ' の設定があり、どの問題イベントがクローズされるかを制御します。

- *All problems* - OK イベントは、トリガーによって作成されたすべての未解決問題を閉じます。
- *All problems if tag values match* - OK イベントは、トリガーによって作成され、少なくとも 1 つの一致するタグ値を持つ未解決の問題を閉じます。タグは、'Tag for matching' トリガ設定によって定義されます。閉じるべき問題イベントがない場合、OK イベントは生成されません。これは、トリガレベルイベント相関と呼ばれます。

イベント相関

イベント相関（グローバルイベント相関とも呼ばれる）は、カスタムイベント終了（結果的に OK イベントを生成する）ルールを
 設定する方法です。

このルールでは、新しい問題イベントと既存の問題イベントの組み合わせ方法を定義し、対応する OK イベントを生成して、
 新しいイベントまたは一致したイベントを閉じることができます。

しかし、イベント相関は、イベント処理のパフォーマンスに悪影響を及ぼしたり、設定を誤ると意図したよりも多くのイベントを
 クローズしてしまう（最悪の場合、すべての問題イベントをクローズしてしまう）ため、非常に慎重に設定する必要があります。
 設定のヒントをいくつか紹介します。

1. コントロールイベント（古いイベントと対になるイベント）に一意のタグを設定し、' 新しいイベントタグ' 相関条件を使用することで、常に相関範囲を縮小します。
2. 'close old event' オペレーションを使用する際に、古いイベントに基づいた条件を追加することを忘れないでください。
3. 異なる相関設定によって使用される共通のタグ名を使用しないようにします。

タスクマネージャー

トリガーに対して' 手動クローズを許可' 設定が有効になっている場合、トリガーによって生成された障害イベントを手動で閉じることができます。これは、障害を更新する ときに Web インターフェースで行われます。イベントは直接閉じられません。代わりに'close event' タスクが作成され、タスクマネージャーによってすぐに処理されます。タスクマネージャーは対応する正常イベントを生成し、障害イベントは閉じられます。

2 その他のイベントソース

サービスイベント

サービスイベントは、これらのイベントに対するサービスアクションが有効な場合にのみ生成されます。
 この場合、サービスステータスの変更ごとに新しいイベントを生成します。

- 問題イベント - サービスステータスが OK から PROBLEM に変更されたとき
- OK イベント - サービスステータスが PROBLEM から OK に変更されたとき

このイベントには、サービス状態の変化の詳細（いつ起こったか、新しい状態は何か）が含まれています。

ディスカバリイベント

Zabbix は、ネットワークディスカバリルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度は、ルールごとに個別に設定できます。ホストまたはサービスが検出されると、ディスカバリイベント (または複数のイベント) が生成されます。

Zabbix は次のイベントを生成します。

イベント	発生条件
Service Up	Zabbix がアクティブなサービスを検出するたびに
Service Down	Zabbix がサービスを検出できないたびに
Host Up	少なくとも 1 つのサービスが IP に対して稼働している場合
Host Down	すべてのサービスが応答していない場合
Service Discovered	サービスがダウンタイム後に戻った場合、または初めて検出された場合
Service Lost	起動後にサービスが失われた場合
Host Discovered	ホストがダウンタイム後に復帰した場合、または初めて検出された場合
Host Lost	起動後にホストが失われた場合

アクティブエージェントの自動登録イベント

アクティブエージェントの自動登録は、Zabbix でイベントを作成します。

設定されている場合、以前に知られていなかったアクティブエージェントがチェックを要求した場合や、ホストメタデータが変更された場合に、アクティブエージェントの自動登録イベントが作成されます。サーバーは、受信したエージェントの IP アドレスとポートを使用して、新しい自動登録ホストを追加します。

詳細については、アクティブエージェントの自動登録ページを参照してください。

内部イベント

内部イベントは以下の場合に発生します。

- アイテムの状態が「通常」から「サポート外」に変化した場合
- アイテムの状態が「サポート外」から「通常」に変化した場合
- LLD ルールの状態が「通常」から「サポート外」に変化した場合
- LLD ルールの状態が「サポート外」から「通常」に変化した場合
- トリガーの状態が「通常」から「不明」に変化した場合
- トリガーの状態が「不明」から「通常」に変化した場合

内部イベントを導入する目的は、例えばアイテムがサポート外となりデータ収集を停止した場合など、内部イベントが発生した際にユーザーが通知を受け取れるようにすることです。

内部イベントは、これらのイベントに対する内部アクションが有効になっている場合にのみ作成されます。内部イベントの生成を停止するには（例えば、アイテムがサポート外になった場合など）、アラート → アクション → [内部アクション] で内部イベントに対するすべてのアクションを無効にしてください。

> **Note:**
> 内部アクションが無効になっている場合でも、オブジェクトが「サポート外」状態の間は、そのオブジェクトのリカバリエベントは作成されます。
>
> 内部アクションが有効になっている場合、オブジェクトが「サポート外」状態の間は、「問題イベント」が作成されていなくても、そのオブジェクトのリカバリエベントが作成されます。

関連項目：サポート外アイテムの通知を受け取る

3 問題の手動クローズ

概要

一般的に、トリガーのステータスが障害から正常に変わると、障害イベントは自動的に解決されますが、トリガー式によって問題が解決されたかどうかを判断するのが難しい場合があります。そのような場合、問題は手動で解決する必要があります。

例えば、*syslog* が最適なパフォーマンスのためにカーネルパラメータを調整する必要があると報告する場合があります。この場合、問題は Linux 管理者に報告され、管理者が修正し、その後手動で問題をクローズします。

問題は、手動クローズを許可オプションが有効になっているトリガーに対してのみ手動でクローズできます。

問題が「手動でクローズ」されると、Zabbix は Zabbix サーバー用の新しい内部タスクを生成します。その後、タスクマネージャープロセスがこのタスクを実行し、OK イベントを生成することで、問題イベントをクローズします。

生成された *OK* イベントには、そのイベントに対して解決されたイベントタグの完全なセット（テンプレート、ホスト、トリガーから継承されたタグを含む）が含まれます。これらのタグは、通知や`{EVENT.RECOVERY.TAGS}`、`{EVENT.RECOVERY.TAGSJSON}`などのマクロで利用できます。

手動でクローズされた問題は、基礎となるトリガーが再び障害状態にならないことを意味するものではありません。トリガー式は再評価され、以下の場合に問題が発生する可能性があります:

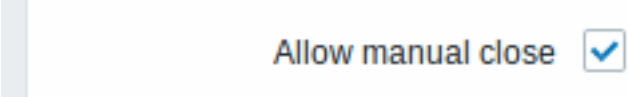
- トリガー式に含まれる任意のアイテムに新しいデータが到着した場合（スロットリングの前処理ステップで破棄された値は受信されたものとは見なされず、トリガー式の再評価を引き起こしません）；
- 式で日付と時刻やnodata()関数が使用されている場合。

設定

手動で問題を閉じるには、2 つのステップが必要です。

トリガー設定

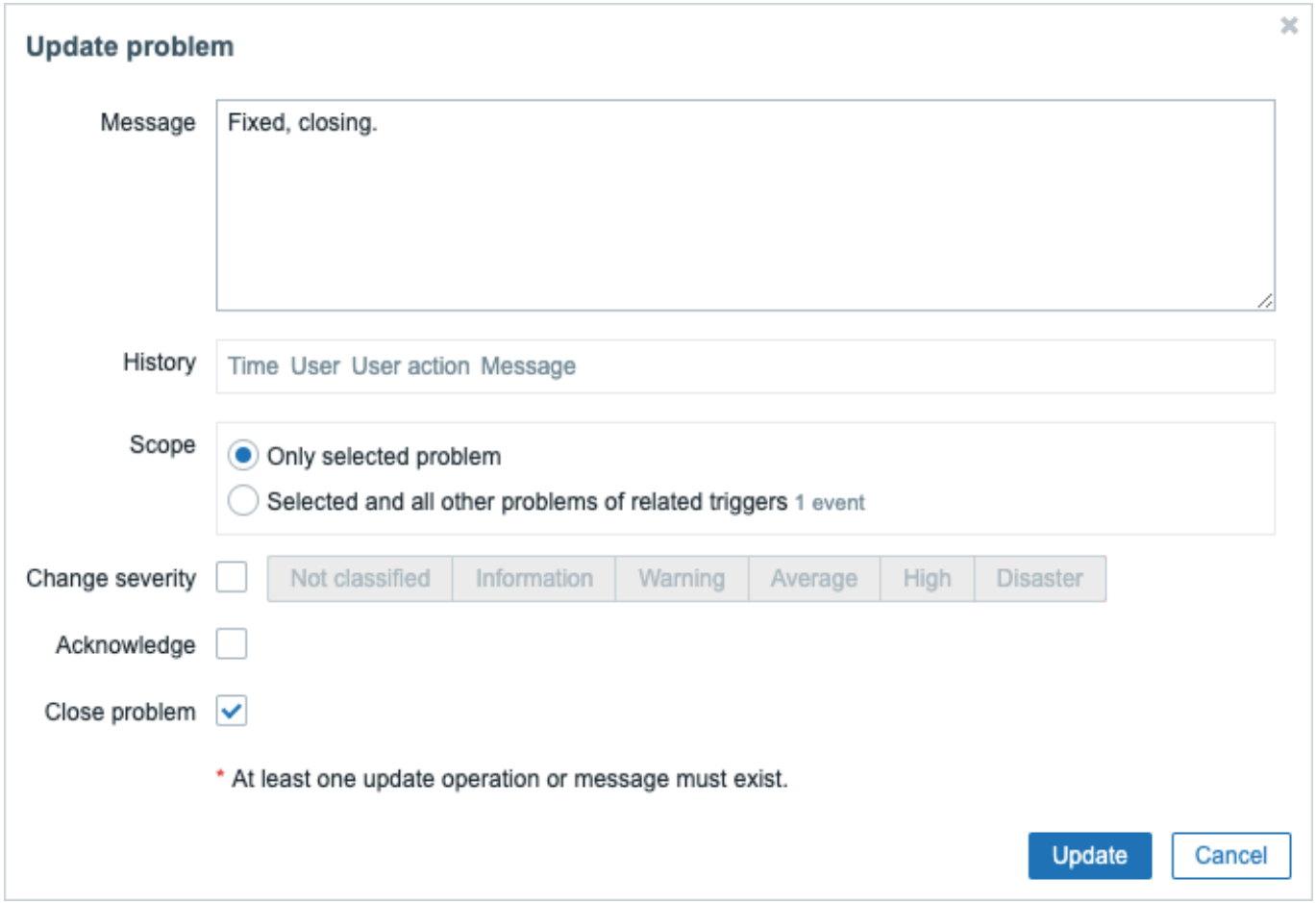
トリガー設定で、手動クローズを許可オプションを有効にします。



問題更新ウィンドウ

手動クローズフラグが設定されたトリガーで問題が発生した場合、その問題の問題更新ポップアップウィンドウを開き、手動で問題をクローズできます。

問題をクローズするには、フォームの問題をクローズオプションにチェックを入れ、更新をクリックします。



すべての必須入力フィールドには赤いアスタリスクが付いています。

リクエストは Zabbix サーバーによって処理されます。通常、問題をクローズするのに数秒かかります。その処理中、監視データ > 障害で問題のステータスとして *CLOSING* が表示されます。

検証

問題が手動でクローズされたことは、以下の方法で確認できます。

- 監視 > 問題から利用できるイベントの詳細で確認できます。
- 通知メッセージで `{EVENT.UPDATE.HISTORY}` マクロを使用することで、この情報を取得できます。

5 イベント相関

概要

イベント相関により、非常に正確かつ柔軟な方法で、障害のイベントをその解決策に関連付けることができます。

イベントの相関関係は次のように定義できます。

- トリガーレベル I - 1 つのトリガーを使用して、個別の障害をその解決策に関連付けることができます
- グローバル - グローバル相関ルールを使用して、別のトリガー/ポーリングメソッドから障害をその解決策に関連付けることができます

1 トリガーによるイベント相関

概要

トリガーベースのイベント相関は、1 つのトリガーによって報告された個別の問題を相関させることができます。

通常、OK イベントは 1 つのトリガーによって作成されたすべての問題イベントをクローズできますが、より詳細なアプローチが必要な場合もあります。たとえば、ログファイルを監視する場合、ログファイル内の特定の問題を検出し、それらを一括ではなく個別にクローズしたい場合があります。

これは、*PROBLEM* イベント生成モードパラメータが複数に設定されているトリガーの場合です。このようなトリガーは通常、ログ監視やトラップ処理などに使用されます。

Zabbix では、タグ付けに基づいて問題イベントを関連付けることができます。タグは値を抽出し、問題イベントの識別を作成するために使用されます。これを利用して、タグの一致に基づいて問題を個別にクローズすることもできます。

つまり、同じトリガーがイベントタグによって識別される個別のイベントを作成できます。したがって、問題イベントは 1 つずつ識別され、イベントタグによる識別に基づいて個別にクローズできます。

仕組み

ログ監視では、次のような行に遭遇することがあります。

```
Line1: Service 1 stopped
Line2: Service 2 stopped
Line3: Service 1 was restarted
Line4: Service 2 was restarted
```

イベント相関のアイデアは、Line1 の問題イベントを Line3 の解決に、Line2 の問題イベントを Line4 の解決にマッチさせ、これらの問題を 1 つずつクローズできるようにすることです。

```
Line1: Service 1 stopped
Line3: Service 1 was restarted #Line 1の問題がクローズ

Line2: Service 2 stopped
Line4: Service 2 was restarted #Line 2の問題がクローズ
```

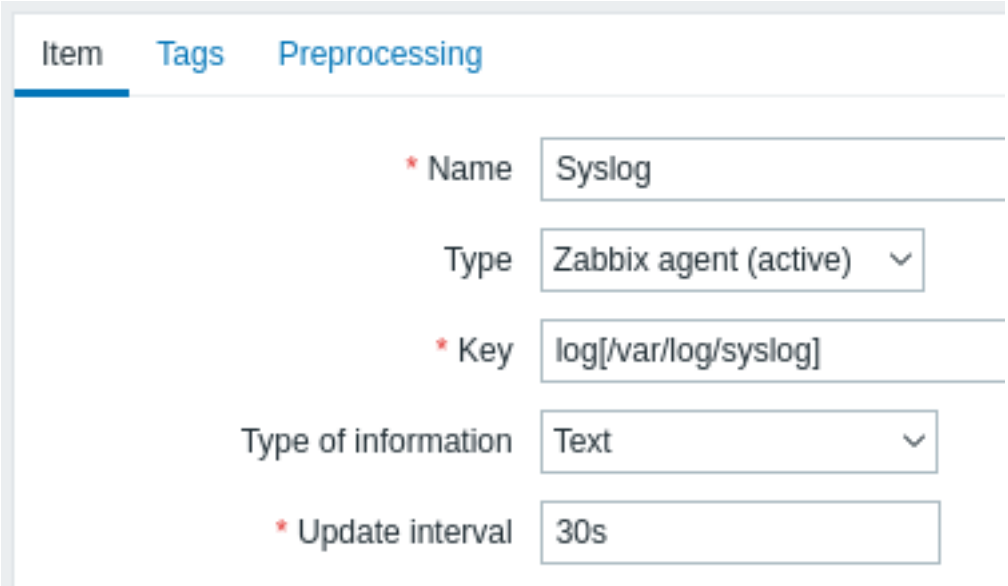
これを行うには、これらの関連するイベントに「Service 1」や「Service 2」などのタグを付ける必要があります。これは、タグ値を抽出するためにログ行に正規表現を適用することで実現できます。イベントが作成されると、それぞれ「Service 1」と「Service 2」でタグ付けされ、問題を解決にマッチさせることができます。

設定

item

始めに、例えばログファイルを監視するような項目を設定するのがよいでしょう:

```
log[/var/log/syslog]
```



item が設定された状態で、設定変更が反映されるまで 1 分ほど待ってから、Latest dataにアクセスして、
 item がデータ収集を開始したことを確認します。

トリガー

アイテムが動作している場合は、トリガーを設定する必要があります。ログファイル内のどのエントリに注意を払うべきかを決定することが重要です。たとえば、次のトリガー式は「Stopping」のような文字列を検索して、潜在的な問題を示します。

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
「Stopping」という文字列を含む各行が問題と見なされるようにするには、トリガー設定の問題イベント生成モードを「複数」に設定してください。

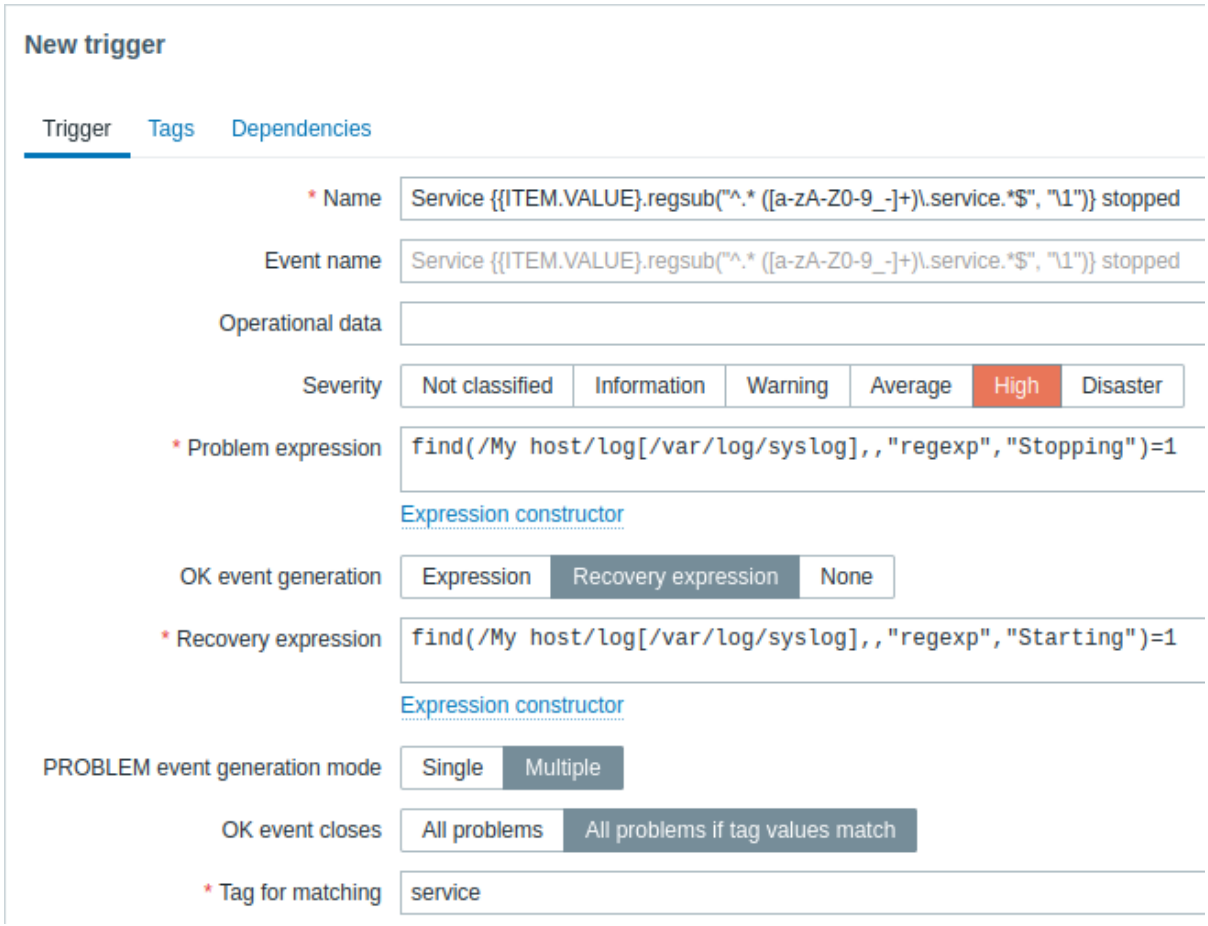
次に、リカバリー式を定義します。次のリカバリー式は、「Starting」という文字列を含むログ行が見つかった場合にすべての問題を解決します。

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

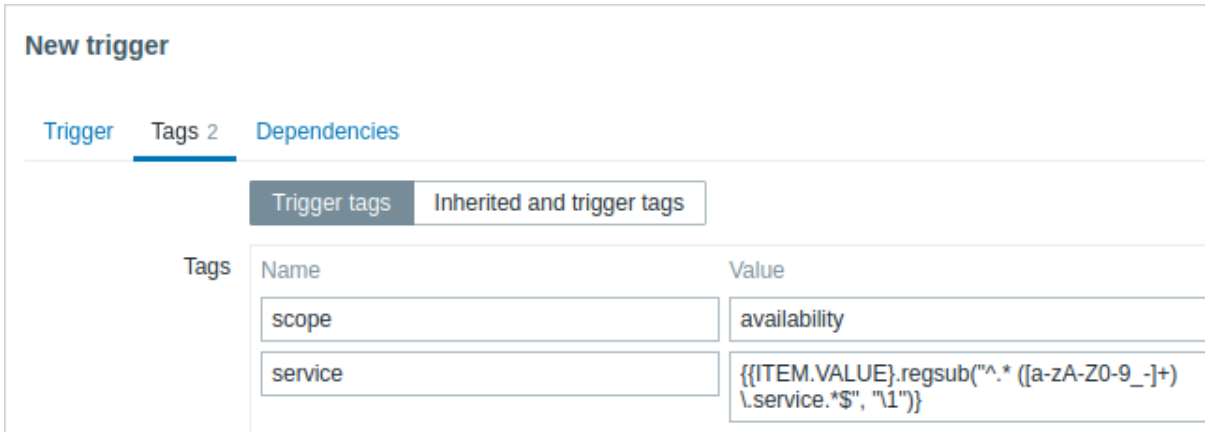
これは望ましくないため、対応する根本的な問題のみがクローズされるように何らかの方法で確実にすることが重要です。ここでタグ付けが役立ちます。

トリガー設定でタグを指定することで、問題と解決を一致させることができます。次の設定を行う必要があります。

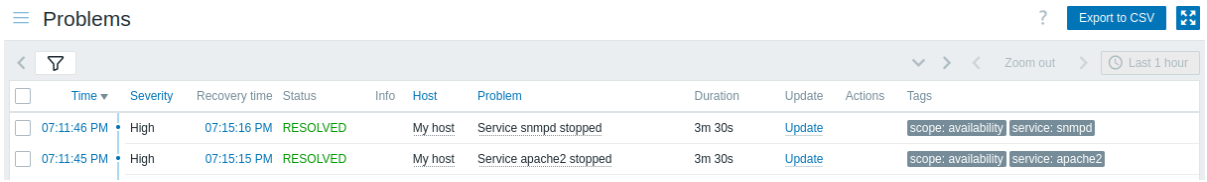
- 問題イベント生成モード: 複数
- *OK* イベントでクローズ: タグ値が一致するすべての問題
- イベントマッチング用のタグ名を入力



- タグを設定して、ログ行からタグ値を抽出します



正常に設定されていれば、監視 → 障害でアプリケーションごとにタグ付けされ、解決にマッチした問題イベントを確認できます。



Warning:
誤った設定により、無関係な問題に対して類似したイベントタグが作成される可能性があるため、以下に示すケースを必ずご確認ください！

- インデックス付きマクロは常にトリガー設定の式フィールドを参照し、リカバリー式は参照しません。たとえば、リカバリーイベントでは、**{ITEM.VALUE1}** はリカバリー時点での問題式の最初のアイテムの最新値に解決されます。リカバリー式が異なるアイテムに基づいており、リカバリー時点までに問題式アイテムの値が変化した場合、イベントには異なるタグが付き、相関されません。

- 2 つのアプリケーションが同じログファイルにエラーとリカバリーメッセージを書き込む場合、ユーザーはタグ値に異なる正規表現を使用して、{ITEM.VALUE} マクロからサービス A やサービス B の名前を抽出し、同じトリガーで 2 つの *service* タグを使用することを決定するかもしれません（たとえば、メッセージフォーマットが異なる場合）。ただし、正規表現に一致しない場合、これは計画通りに動作しない可能性があります。一致しない正規表現は空のタグ値となり、問題イベントと OK イベントの両方でタグ値が空の場合、それらは相関されます。そのため、サービス A からのリカバリーメッセージが、誤ってサービス B からのエラーメッセージをクローズしてしまう可能性があります。
- 実際のタグとタグ値は、トリガーが発生したときにのみ表示されます。使用した正規表現が無効な場合、暗黙的に *UNKNOWN* 文字列に置き換えられます。最初の問題イベントで *UNKNOWN* タグ値が見逃された場合、同じ *UNKNOWN* タグ値を持つ後続の OK イベントが発生し、それらがクローズすべきでない問題イベントをクローズしてしまう可能性があります。
- ユーザーがタグ値としてマクロ関数なしで {ITEM.VALUE} マクロを使用する場合、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字が非特異的な場合、無関係な問題に対しても類似したイベントタグが作成される可能性があります。

2 グローバルイベント相関

概要

グローバルイベント相関は、Zabbix で監視されているすべてのメトリックに対して相関を作成することを可能にします。

まったく異なるトリガーによって作成されたイベントを相関させ、それらすべてに同じ操作を適用することができます。インテリジェントな相関ルールを作成することで、何千もの繰り返し通知から自分自身を救い、問題の根本原因に集中することが実際に可能です！

グローバルイベント相関は、1 つのトリガーに基づく問題と解決のロジックから自分自身を解放する強力なメカニズムです。これまでは、1 つのトリガーによって 1 つの問題イベントが作成され、その問題の解決も同じトリガーに依存していました。1 つのトリガーで作成された問題を別のトリガーで解決することはできませんでした。しかし、イベントタグに基づくイベント相関を使用すれば、それが可能になります。

例えば、ログトリガーがアプリケーションの問題を報告し、ポーリングトリガーがアプリケーションが稼働中であることを報告する場合があります。イベントタグを活用することで、ログトリガーには *status:down*、ポーリングトリガーには *status:up* というタグを付けることができます。そして、グローバル相関ルールでこれらのトリガーを関連付け、この相関に適切な操作（古いイベントをクローズするなど）を割り当てることができます。

別の使い方として、グローバル相関は類似のトリガーを識別し、それらに同じ操作を適用することができます。ネットワークポートの問題ごとに 1 つの問題レポートだけを取得できたらどうでしょうか？すべてを報告する必要はありません。これもグローバルイベント相関で可能です。

グローバルイベント相関は相関ルールで設定します。相関ルールは、新しい問題イベントが既存の問題イベントとどのようにペアリングされるか、および一致した場合に何をするか（新しいイベントをクローズする、一致した古いイベントを対応する OK イベントを生成してクローズするなど）を定義します。問題がグローバル相関によってクローズされた場合、監視 > 問題の 情報列に報告されます。

グローバル相関ルールの設定は、Super Admin レベルのユーザーのみが利用できます。

Attention:
イベント相関は非常に慎重に設定する必要があります。イベント処理のパフォーマンスに悪影響を及ぼす可能性があるほか、誤って設定すると意図した以上のイベントがクローズされる可能性があります（最悪の場合、すべての問題イベントがクローズされることもあります）。

グローバル相関を安全に設定するには、次の重要なヒントを守ってください。

- 相関の範囲を縮小する。新しいイベントに一意のタグを必ず設定し、新しいイベントのタグ名の相関条件を使用する。
- 古いイベントをクローズ操作を使用する場合は、古いイベントに基づく条件を追加する（そうしないと、すべての既存の問題がクローズされる可能性があります）。
- 異なる相関設定で使用される可能性のある一般的なタグ名の使用は避ける。
- 相関ルールの数は本当に必要なものだけに制限する。

参考: 既知の問題

設定

イベント相関ルールをグローバルに設定するには:

- データ収集 > イベント相関に移動します
- 右側の イベント相関の作成をクリックします（または既存のルールを編集する場合は相関名をクリックします）
- フォームに相関ルールのパラメータを入力します

correlation_rule.png

すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	一意の相関ルール名。
計算タイプ	条件の計算方法として以下のオプションが利用可能です: **And** - すべての条件を満たす必要があります **Or** - いずれか 1 つの条件を満たせば十分です **And/Or** - 異なる条件タイプでは AND、同じ条件タイプでは OR カスタム式 - アクション条件を評価するためのユーザー定義の計算式。すべての条件（大文字の A、B、C、... で表される）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字区別）、**or**（大文字小文字区別）、**not**（大文字小文字区別）を含めることができます。
条件	条件のリスト。条件の設定方法については下記を参照してください。
説明	相関ルールの説明。
操作	イベントが相関されたときに実行する操作のチェックボックスをオンにします。利用可能な操作は以下の通りです: 古いイベントをクローズ - 新しいイベントが発生したときに古いイベントをクローズします。古いイベントをクローズ操作を使用する場合は、必ず古いイベントに基づく条件を追加してください。そうしないと、既存のすべての問題がクローズされる可能性があります。 新しいイベントをクローズ - 新しいイベントが発生したときに新しいイベントをクローズします
有効	このチェックボックスをオンにすると、相関ルールが有効になります。

新しい条件の詳細を設定するには、条件ブロックの をクリックします。ポップアップウィンドウが開き、条件の詳細を編集できます。

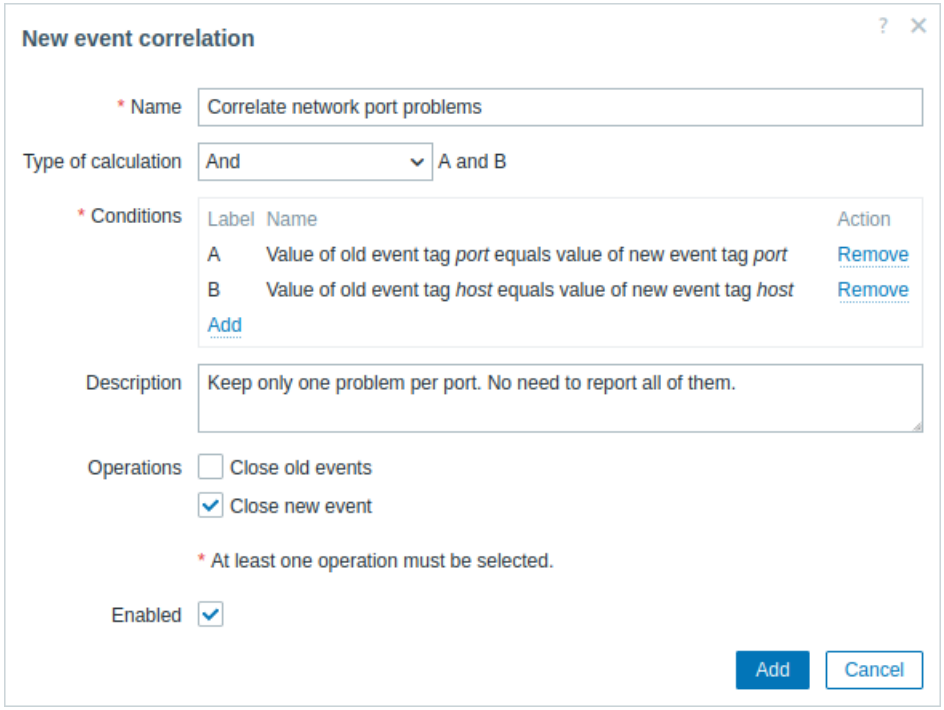
パラメータ	説明
新しい条件	イベントを相関させるための条件を選択します。 注意古いイベント条件が指定されていない場合、すべての古いイベントが一致してクローズされる可能性があります。同様に新しいイベント条件が指定されていない場合、すべての新しいイベントが一致してクローズされる可能性があります。 利用可能な条件は以下の通りです: 古いイベントのタグ名 - 一致させる古いイベントのタグ名を指定します。 新しいイベントのタグ名 - 一致させる新しいイベントのタグ名を指定します。 新しいイベントのホストグループ - 一致させる新しいイベントのホストグループを指定します。 イベントタグペア - 一致させる新しいイベントのタグ名と古いイベントのタグ名を指定します。この場合、両方のイベントのタグの値が一致すれば一致とみなされます。タグ名は一致している必要はありません。 このオプションは、設定時に値がわからない可能性がある実行時の値を一致させる場合に便利です(例も参照)。 古いイベントのタグ値 - 以下の演算子を使用して、一致させる古いイベントのタグ名と値を指定します: 等しい - 古いイベントのタグ値を持つ 等しくない - 古いイベントのタグ値を持たない 含む - 古いイベントのタグ値に文字列を含む 含まない - 古いイベントのタグ値に文字列を含まない 新しいイベントのタグ値 - 以下の演算子を使用して、一致させる新しいイベントのタグ名と値を指定します: 等しい - 新しいイベントのタグ値を持つ 等しくない - 新しいイベントのタグ値を持たない 含む - 新しいイベントのタグ値に文字列を含む 含まない - 新しいイベントのタグ値に文字列を含まない

Warning:
誤った設定により、無関係な問題に対しても同様のイベントタグが作成される可能性があるため、下記のケースを必ずご確認ください！

- 実際のタグおよびタグ値は、トリガーが発火したときにのみ表示されます。使用した正規表現が無効な場合、*UNKNOWN* という文字列に自動的に置き換えられます。最初の問題イベントで *UNKNOWN* タグ値が見逃された場合、同じ *UNKNOWN* タグ値を持つ OK イベントが後から発生し、本来クローズすべきでない問題イベントをクローズしてしまう可能性があります。
- ユーザーがタグ値としてマクロ関数なしの {ITEM.VALUE} マクロを使用した場合、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字が特定できない場合、無関係な問題に対しても同様のイベントタグが作成される可能性があります。

例

同じネットワークポートからの繰り返しの問題イベントを停止します。



このグローバル相関ルールは、*host* および *port* タグ値がトリガーに存在し、元のイベントと新しいイベントで同じである場合に問題を相関させます。

この操作により、同じネットワークポートで新たに発生した問題イベントをクローズし、元の問題のみをオープンのままにします。

6 タグ付け

概要

タグは、タグ名とタグ値で構成されます。エンティティにタグを付ける際は、名前だけを使用することも、値と組み合わせて使用することもできます（例：`mysql`、`jira`、`target:mysql`、`service:jira` など）。

タグはさまざまなエンティティに定義できます：

- テンプレート
- ホスト
- アイテム
- Web シナリオ
- トリガー
- サービス
- テンプレートのアイテムおよびトリガー
- ホスト、アイテム、トリガーのプロトタイプ

Note:
タグの定義に関する一般的な推奨事項や、テンプレート、アイテム、トリガー、低レベルディスカバリルールに関する具体的なガイダンスについては、公式 Zabbix ガイドラインを参照してください。

タグには複数の目的がありますが、特にイベントをマークするために使用されます。エンティティにタグが付けられている場合、タグ付きエンティティに関連する新しいイベントは、そのタグを継承します。例えば：

- タグ付きテンプレートの場合 - テンプレートのトリガーによって作成されたホストの問題は、テンプレートのタグを継承します。
- タグ付きホストの場合 - ホストの問題はホストのタグを継承します。
- タグ付きアイテム/Web シナリオの場合 - アイテム/Web シナリオの問題はアイテム/Web シナリオのタグを継承します。
- タグ付きトリガーの場合 - トリガーによって作成された問題はトリガーのタグを継承します。

問題イベントは、テンプレート、ホスト、アイテム/Web シナリオ、トリガーというエンティティの全チェーンからすべてのタグを継承します。同一の `tag:value` の組み合わせ（マクロ解決後）は 1 つにまとめられ、重複が回避されます。

手動クローズによって生成されたリカバリエベントにも、テンプレート、ホスト、アイテム/Web シナリオ、トリガーから継承された解決済みイベントタグが含まれます。これらのタグは、通知や`{EVENT.RECOVERY.TAGS}`、`{EVENT.RECOVERY.TAGSJSON}`など

のマクロで利用できます。

カスタムイベントタグは、より柔軟な運用を可能にします。例えば：

- イベント相関をイベントタグに基づいて設定できます。
- アクション条件をイベントタグに基づいて設定できます。
- アイテムの問題をイベントタグに基づいてグループ化できます。
- 問題タグを使用して、問題をサービスにマッピングできます。

エンティティには、同じタグ名でも異なるタグ値（例：component:memory と component:storage）でタグを付けることができます。同様に、値のないタグと同じタグ名で値を持つタグ（例：database と database:postgresql）をエンティティに付与することもできます。このようなタグは重複とは見なされません。

ユースケース

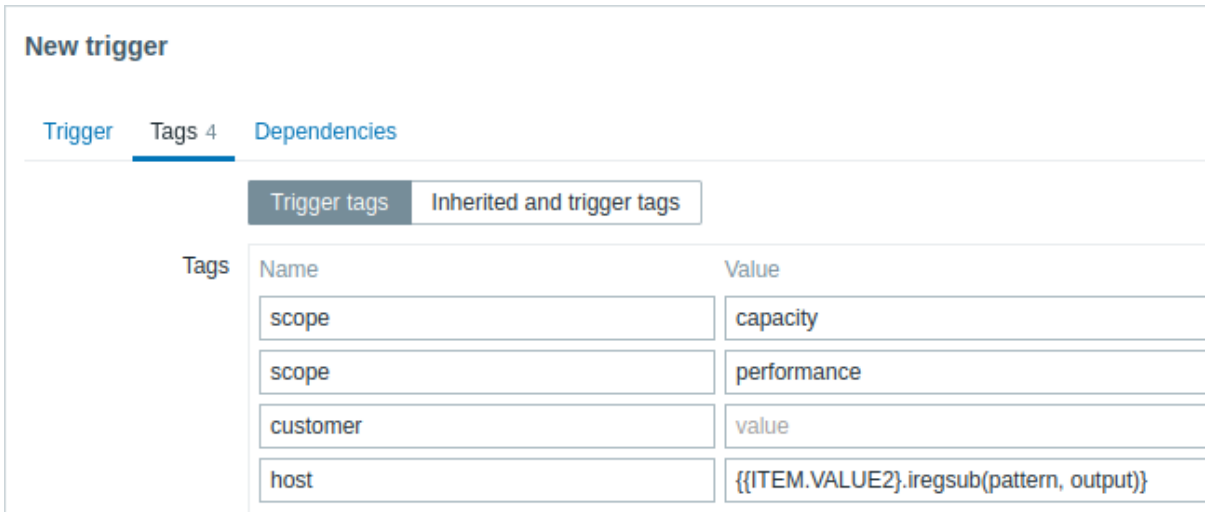
タグ付けの一般的なユースケースは以下の通りです。

1. トリガーイベントのマーク:
   - トリガータグを定義します（例: scope:performance）。
   - このトリガーによって作成された障害にはトリガータグが付きます。
2. テンプレート継承の障害のマーク:
   - テンプレートタグを定義します（例: target:mysql）。
   - このテンプレートのトリガーから作成された障害にはテンプレートタグが付きます。
3. ホスト障害のマーク:
   - ホストタグを定義します（例: service:jira）。
   - このホストのトリガーから作成された障害にはホストタグが付きます。
4. 関連アイテムのフィルタリング:
   - アイテムタグを定義します（例: component:cpu）。
   - 監視 > 最新データで、アイテムは component:cpu タグでフィルタリングできます。
5. アイテム値から抽出した情報をタグ値として使用:
   - タグ値にマクロを使用してタグを定義します（例: tag-name:{{ITEM.VALUE<N>}.regsub()}）。
   - 監視 > 障害で、障害にはアイテム値から抽出したデータに解決されたタグ値が付きます。
6. ログファイル内の障害を個別に特定してクローズ:
   - ログ監視アイテムトリガー用のトリガータグを定義し、マクロを使用してアイテム値から値を抽出します（例: service:{{ITEM.VALUE<N>}.regsub()}）。
   - トリガーの設定でイベント相関を設定します:
     - 障害イベント生成モードを「複数」に設定;
     - *OK* イベントでクローズを「タグ値が一致するすべての障害」に設定;
     - 一致させるタグを設定。
   - ログアイテムトリガーによって作成された障害にはトリガータグが付き、個別にクローズされます。
7. 通知のフィルタリング:
   - トリガータグを定義します（例: trigger1 には scope:security、trigger2 には scope:availability）。
   - アクション条件でタグフィルタリングを使用し、タグデータに一致するイベントのみ通知を受信します。
8. 通知で障害を特定:
   - トリガータグを定義します。
   - 障害通知で{EVENT.TAGS}マクロを使用します。
   - 障害通知にはトリガータグが含まれ、どのアプリケーション/サービスの通知か特定しやすくなります。
9. テンプレートタグを使用して設定作業を簡素化:
   - テンプレートトリガータグを定義します。
   - このテンプレートトリガーから作成されたトリガーにはそのタグが付きます。
10. LLD からタグ付きトリガーを作成:
    - タグ名または値に LLD マクロを含むトリガープロトタイプタグを定義します（例: scope:{#FSNAME}）。
    - トリガープロトタイプから作成されたトリガーにはそのタグが付きます。
11. サービスタグを使用してサービスを一致:
    - サービスタグを定義します。
    - 一致するタグを持つサービス用にサービスアクションを設定します。
    - さらに、サービスタグを使用してサービスをSLAにリンクし、SLA 計算に利用します。
12. サービス障害タグを使用してサービスと障害をリンク:
    - サービス設定で障害タグを定義します（例: target:mysql）。
    - 一致するタグを持つ障害は自動的にサービスに関連付けられ、設定されたサービスステータス計算ルールに基づいてサービスステータスが変更されます。
13. ホストがメンテナンスモードのときに障害を抑制:
    - メンテナンス期間の設定でタグを定義します。
    - 定義したタグを持つ障害は抑制されます。
14. ユーザーグループへのアクセス権付与:
    - ユーザーグループの設定でタグを定義します。

- ユーザーグループのユーザーは、定義したタグを持つ障害のみ閲覧できます。

設定

タグは、トリガーの設定などの専用タブで定義できます。



マクロのサポート

組み込みおよびユーザーマクロは、タグ内でイベント発生時に解決されます。イベントが発生するまでは、これらのマクロは Zabbix フロントエンドで未解決のまま表示されます。

低レベルディスカバリマクロは、ディスカバリプロセス中に解決されます。

トリガータグ名および値で使用できるマクロは以下の通りです。

- {ITEM.VALUE}、{ITEM.VALUE.AGE}、{ITEM.VALUE.DATE}、{ITEM.VALUE.TIME}、{ITEM.VALUE.TIMESTAMP}、{ITEM.LASTVALUE}、{ITEM.LASTVALUE.AGE}、{ITEM.LASTVALUE.DATE}、{ITEM.LASTVALUE.TIME}、{ITEM.LASTVALUE.TIMESTAMP}、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ（トリガー式内で 1 つまたは複数のホストのインベントリ値を参照するため）
- ユーザーマクロおよびコンテキスト付きユーザーマクロ（コンテキストには低レベルディスカバリマクロを含めることができます）
- 低レベルディスカバリマクロ（トリガープロトタイプタグのみ）

テンプレート、ホスト、アイテム/ウェブシナリオのタグ名および値で使用できるマクロは以下の通りです。

- {HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ
- ユーザーマクロ
- 低レベルディスカバリマクロ（ホストおよびアイテムプロトタイプタグのみ）

トリガーベースの通知で使用できるマクロは以下の通りです。

- {EVENT.TAGS} および {EVENT.RECOVERY.TAGS} の組み込みマクロ（これらのマクロはイベントタグまたはリカバリーイベントタグのカンマ区切りリストに解決されます）
- {EVENT.TAGSJSON} および {EVENT.RECOVERY.TAGSJSON} の組み込みマクロ（これらのマクロはイベントタグオブジェクトまたはリカバリーイベントタグオブジェクトを含む JSON 配列に解決されます）

トリガータグでの部分文字列抽出

マクロ関数を使用して、タグ名またはタグ値を入力する際に部分文字列の抽出がサポートされています。この関数は、サポートされているマクロで取得した値に正規表現を適用します。例:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

マクロ解決後にタグ名または値が 255 文字を超える場合は、255 文字に切り詰められます。

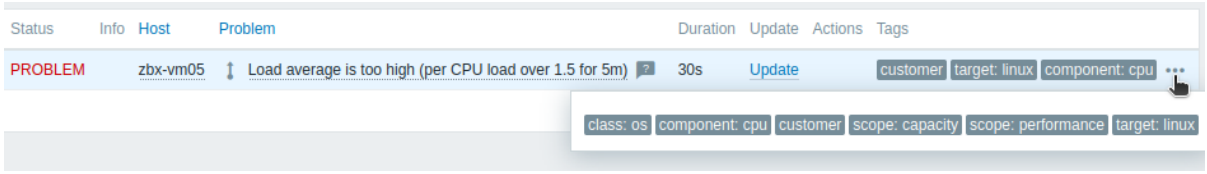
参考: イベントタグ付けのための低レベルディスカバリマクロでのマクロ関数の使用も参照してください。

イベントタグの表示

タグが定義されている場合、新しいイベントで以下の場所でタグを表示できます:

- 監視データ > 障害
- 監視データ > 障害 > イベントの詳細
- ダッシュボード > 障害ウィジェット

表示されるタグの順序と数は、監視データ > 障害または 障害ダッシュボードウィジェットの タグ表示優先度および タグの表示フィルタリングオプションによって決まります。最大 3 つのタグが表示されます。タグが 3 つを超える場合は、3 つのドットにカーソルを合わせると、すべてのタグがポップアップウィンドウで表示されます。



7 視覚化

サイドバーから、参照したい項目を選択してください。

1 グラフ

概要

Zabbix には大量のデータが流入するため、数字だけでなく何が起こっているかを視覚的に表現できれば、ユーザーにとってははるかに簡単になります。

ここでグラフが登場します。グラフを使用すると、データフローを一目で把握したり、問題を関連付けたり、何かがいつ始まったかを発見したり、いつ問題が発生する可能性があるかを提示したりすることができます。

Zabbix はユーザーに以下を提供します。

- 1 アイテムのデータの組み込みシンプルグラフ
- より複雑なものが作れるカスタムグラフ
- 複数のアイテムの比較にアドホックグラフ
- モダンなカスタマイズが可能なベクトルグラフと円グラフ

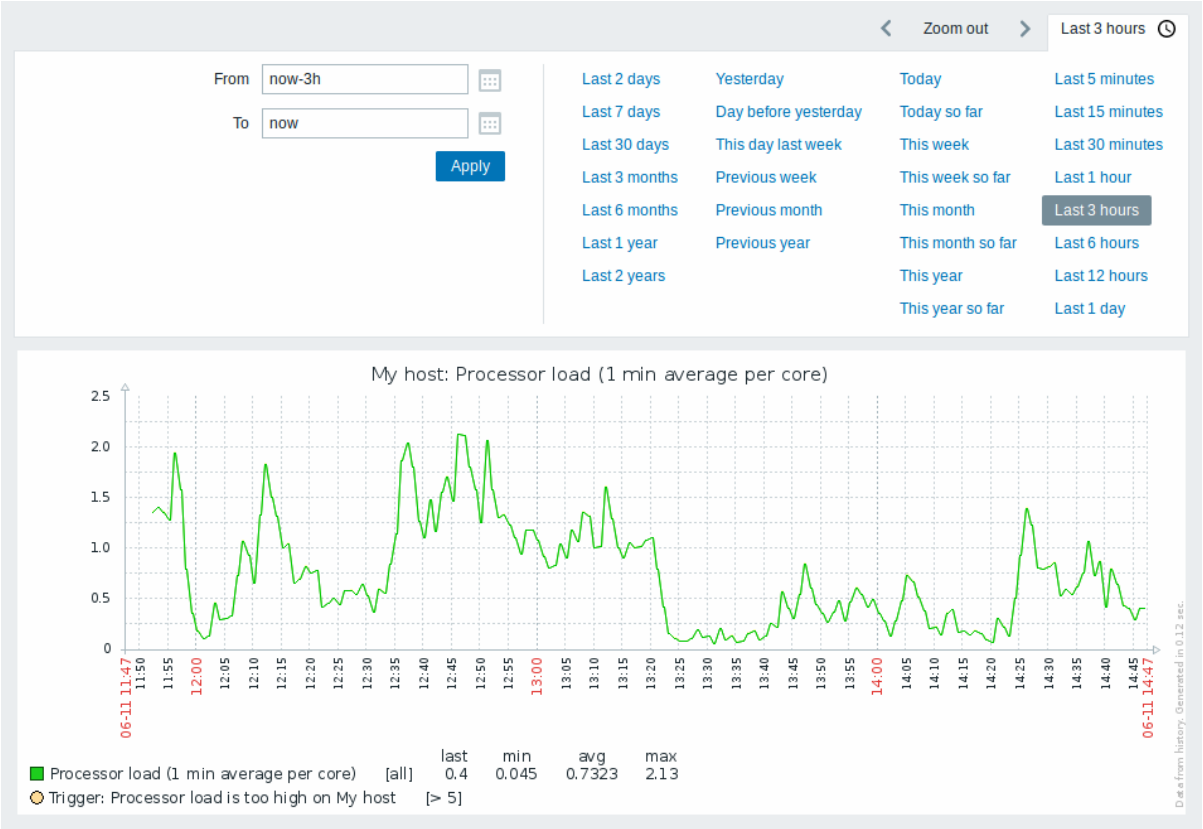
1 シンプルグラフ

概要

アイテムによって収集されたデータの可視化のために、シンプルグラフが提供されています。

シンプルグラフを表示するために、ユーザー側での設定作業は必要ありません。Zabbix によって自由に利用できます。

監視 > 最新データに移動し、該当するアイテムのグラフリンクをクリックするだけで、グラフが表示されます。



Note:
シンプルグラフはすべての数値アイテムに提供されます。テキストアイテムの場合は、監視 > 最新データでヒストリへのリンクが利用できます。

期間セレクター

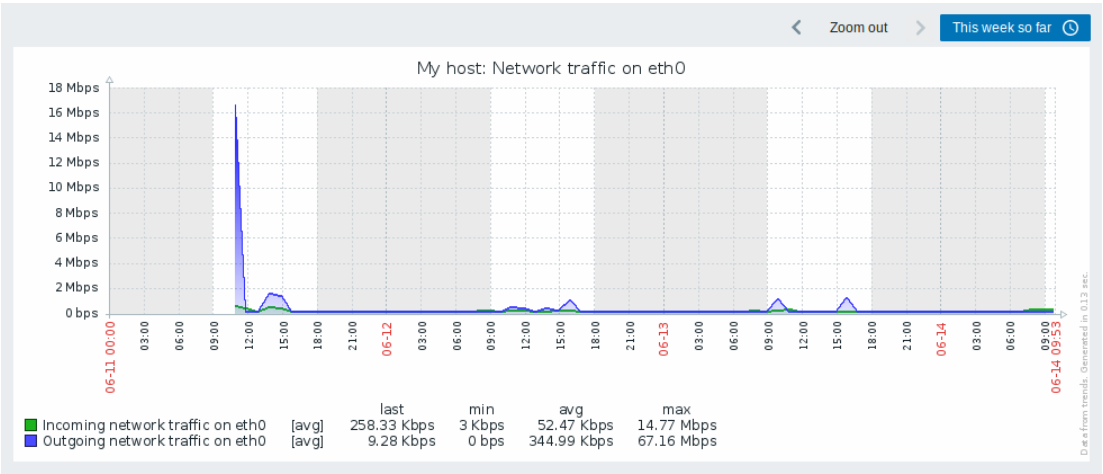
グラフ上部の期間セレクターを使用すると、よく使用する期間をワンクリックで選択できます。
 詳細は、期間およびホストセレクターを参照してください。

最近のデータと長期間のデータ

ごく最近のデータについては、単一の線が各受信値を結んで描画されます。1 つの値に対して少なくとも 1 つの水平ピクセルが利用可能な限り、単一の線が描画されます。

より長い期間のデータについては、**3** 本の線が描画されます。濃い緑色の線は平均値を示し、薄いピンク色と薄い緑色の線はその時点での最大値と最小値を示します。高値と安値の間のスペースは黄色の背景で塗りつぶされます。

作業時間（稼働日）はグラフ上で白い背景で表示され、非作業時間はグレー（デフォルトの *Original blue* フロントエンドテーマの場合）で表示されます。

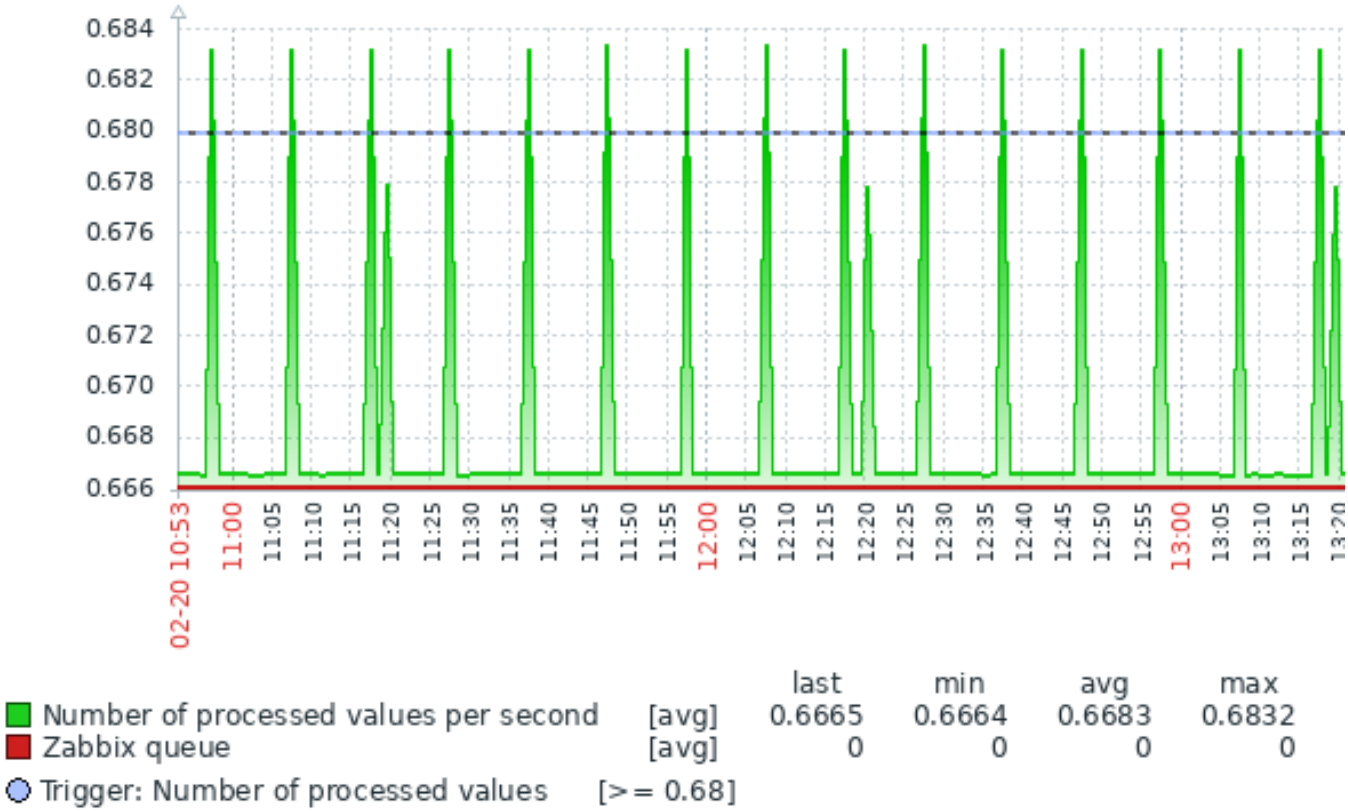


作業時間は常にシンプルグラフで表示されますが、カスタムグラフで表示するかどうかはユーザーの設定によります。

グラフが 3 か月以上を表示している場合、作業時間は表示されません。

トリガーライン

シンプルトリガーは、トリガーの深刻度の色の上に黒い破線で表示されます。グラフ上の青い線と凡例に表示されるトリガー情報に注目してください。グラフには最大 3 つのトリガーラインを表示できます。トリガーがそれ以上ある場合は、深刻度の低いトリガーが優先されます。トリガーは常にシンプルグラフに表示されますが、カスタムグラフでの表示はユーザーの好みによります。



ヒストリ/トレンドからの生成

グラフは、アイテムのヒストリまたはトレンドに基づいて描画できます。

フロントエンドのデバッグモードを有効にしているユーザーには、グラフの右下にグレーの縦書きキャプションが表示され、データの出所が示されます。

ヒストリまたはトレンドが使用されるかどうかには、いくつかの要因が影響します:

- アイテムヒストリの保持期間。例えば、アイテムヒストリを 14 日間保持することができます。この場合、14 日より前のデータはトレンドから取得されます。
- グラフ内のデータの混雑度。横方向のグラフピクセルで表示する秒数が 3600/16 を超える場合、トレンドデータが表示されます（同じ期間のアイテムヒストリがまだ利用可能であっても）。
- トレンドが無効になっている場合、その期間のアイテムヒストリが利用可能であれば、グラフ作成にアイテムヒストリが使用されます。

データの欠如

通常の更新間隔を持つアイテムの場合、アイテムデータが収集されていない場合はグラフに何も表示されません。

ただし、トラッパーアイテムやスケジュールされた更新間隔（通常の更新間隔が 0 に設定されている）を持つアイテムの場合、最初に収集された値までと最後に収集された値からグラフの終わりまで直線が描画されます。この線はそれぞれ最初/最後の値のレベルにあります。

生データへの切り替え

右上のドロップダウンで、シンプルグラフから値/最新 *500* 件の値のリストに切り替えることができます。これは、グラフを構成する数値を確認するのに便利です。

ここで表示される値は生データであり、単位や値の後処理は行われません。ただし、値のマッピングは適用されます。

既知の問題

グラフについては、known issuesを参照してください。

2 カスタムグラフ

概要

カスタムグラフは、その名の通りカスタマイズ機能を提供します。

シンプルグラフは単一のアイテムのデータを表示するのに適していますが、設定機能はありません。

したがって、グラフのスタイルや線の表示方法を変更したい場合や、複数のアイテム (例えば、1 つのグラフで受信トラフィックと送信トラフィックを比較したい場合) を比較したい場合は、カスタムグラフが必要です。

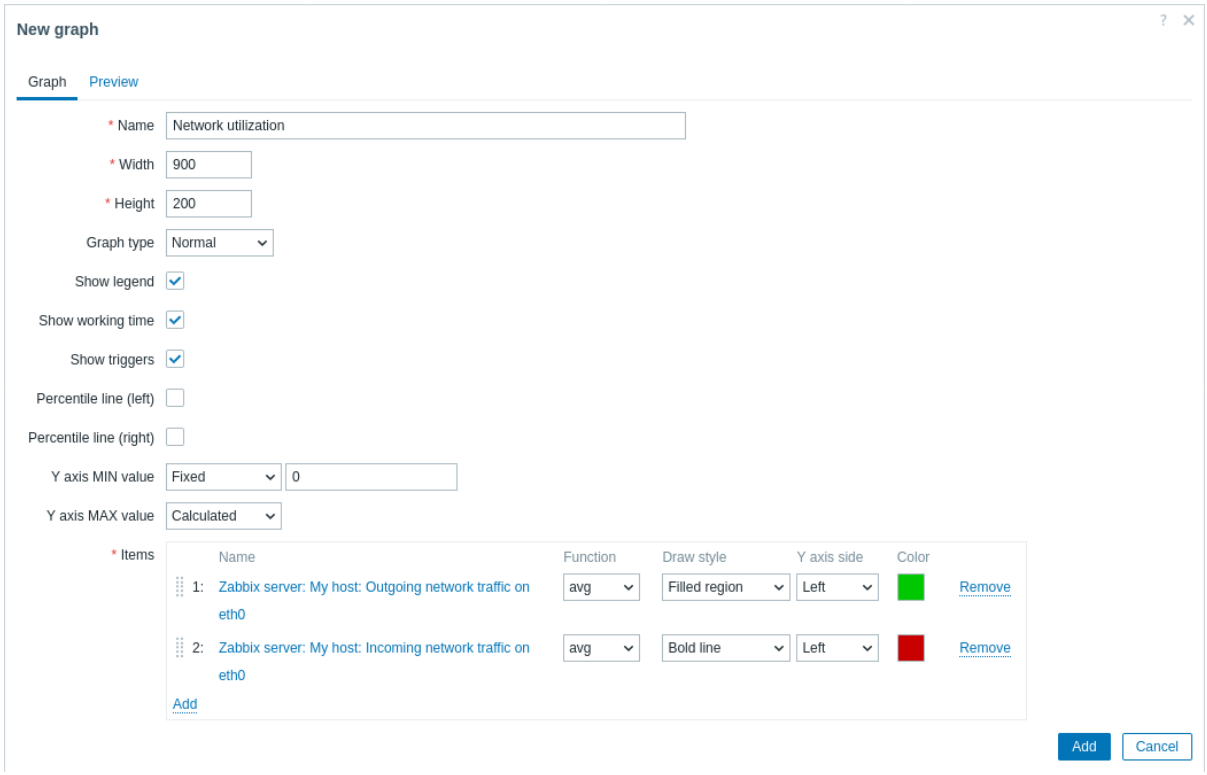
カスタムグラフは手動で設定します。

ホストまたは複数のホスト、または単一のテンプレートに対して作成できます。

カスタムグラフの設定

カスタムグラフを作成するには、以下の手順を実行します。

- データ収集 > ホスト（またはテンプレート）に移動します
- 対象のホストまたはテンプレートの行で グラフをクリックします
- グラフ画面で グラフの作成をクリックします
- グラフ属性を編集します



必須入力フィールドは赤いアスタリスクでマークされています。

グラフ属性:

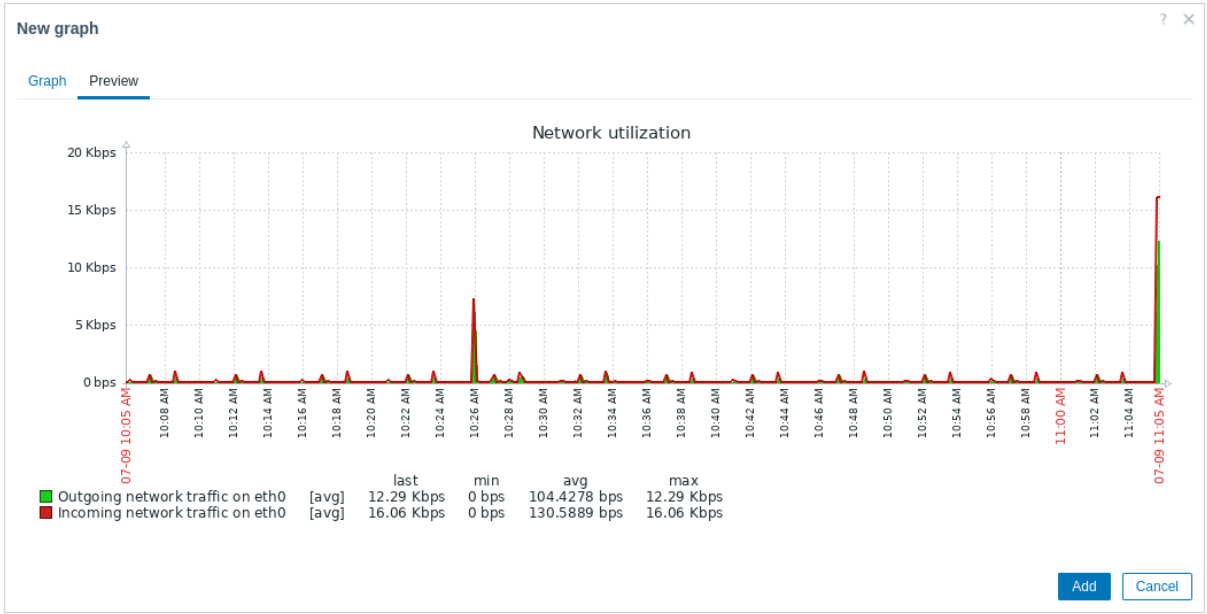
パラメータ	説明
名前	一意のグラフ名。 このフィールドでは、マクロを式として使用できますが、avg、last、min、max 関数のみがサポートされ、パラメータとして時間を指定します（例：{?avg(/host/key,1h)}）。 {HOST.HOST<1-9>} マクロは、このマクロ内で使用する場合にサポートされており、グラフ内の 1 番目、2 番目、3 番目などのホストを参照します（例：{?avg(/{HOST.HOST2}/key,1h)}）。このマクロで 1 番目のホストを参照するのは冗長であり、1 番目のホストは暗黙的に参照できます（例：{?avg(//key,1h)}）。
幅	グラフの幅（ピクセル単位、プレビューおよび円グラフ/爆発円グラフのみ）。
高さ	グラフの高さ（ピクセル単位）。

パラメータ	説明
グラフタイプ	グラフのタイプ： 通常 - 通常のグラフ、値は線で表示 積み上げ - 積み上げグラフ、塗りつぶし領域で表示 円グラフ - 円グラフ 爆発円グラフ - ” 爆発” 円グラフ、部分が円から” 切り出された” ように表示
凡例を表示	このボックスをチェックすると、グラフの凡例が表示されます。
作業時間を表示	選択すると、非作業時間がグレーの背景で表示されます。このパラメータは円グラフおよび爆発円グラフでは利用できません。
トリガーを表示	選択すると、シンプルトリガーがトリガーの深刻度の色の上に黒い点線で表示されます。このパラメータは円グラフおよび爆発円グラフでは利用できません。
パーセンタイル線 (左)	左 Y 軸のパーセンタイルを表示します。例えば、95% パーセンタイルを設定すると、パーセンタイル線は値の 95% がその下に入るレベルに表示されます。明るい緑色の線で表示されます。 通常グラフのみ利用可能です。
パーセンタイル線 (右)	右 Y 軸のパーセンタイルを表示します。例えば、95% パーセンタイルを設定すると、パーセンタイル線は値の 95% がその下に入るレベルに表示されます。明るい赤色の線で表示されます。 通常グラフのみ利用可能です。
Y 軸最小値	Y 軸の最小値： 自動計算 - Y 軸の最小値は自動的に計算されます。 固定 - Y 軸の最小値を固定します。 アイテム - 選択したアイテムの最新値を最小値とします。 このパラメータは円グラフおよび爆発円グラフでは利用できません。
Y 軸最大値	Y 軸の最大値： 自動計算 - Y 軸の最大値は自動的に計算されます。 固定 - Y 軸の最大値を固定します。 アイテム - 選択したアイテムの最新値を最大値とします。 このパラメータは円グラフおよび爆発円グラフでは利用できません。

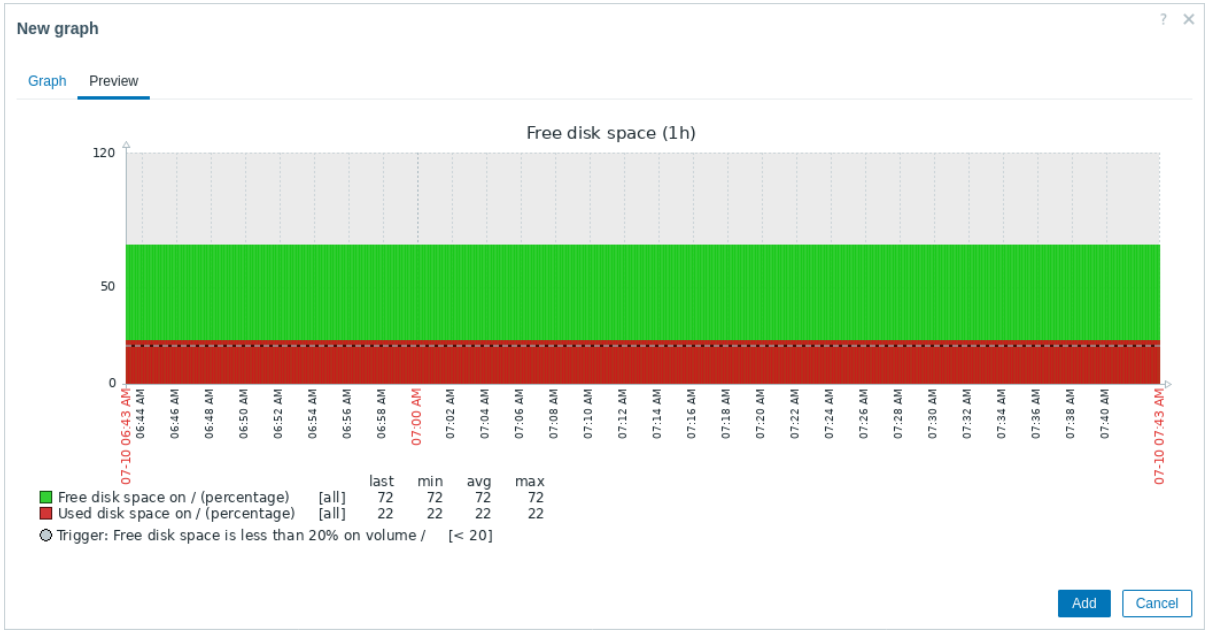
パラメータ		説明
3D ビュー		3D スタイルを有効にします。円グラフおよび爆発円グラフのみ。
アイテム		このグラフに表示するデータのアイテム。追加をクリックしてアイテムを選択します。表示オプション（関数、描画スタイル、左右軸表示、色）も選択できます。
	ソート順（*0→100*）	描画順序。0 が最初に処理されます。線や領域を他の線や領域の後ろ（または前）に描画するために使用できます。 行の先頭のアイコンをドラッグ＆ドロップして、ソート順やどのアイテムが他のアイテムの前に表示されるかを設定できます。
	名前	選択したアイテムの名前がリンクとして表示されます。リンクをクリックすると、他の利用可能なアイテムのリストが開きます。
	タイプ	タイプ（円グラフおよび爆発円グラフのみ）： 単純 - アイテムの値が円グラフ上で比例して表現されます グラフ合計 - アイテムの値が円グラフ全体を表します ”グラフ合計” アイテムの色付けは、”比例” アイテムによって占められていない範囲でのみ表示されることに注意してください。
	関数	アイテムに対して 1 ピクセルの縦方向に複数の値が存在する場合に表示する値を選択します： **all** - グラフ内のすべての可能な値（最小値、最大値、平均値）を表示します。短い期間ではこの設定は効果がありませんが、長い期間でデータが 1 ピクセルに詰まる場合に、’all’ は最小値、最大値、平均値を表示し始めます。この関数は通常グラフタイプでのみ利用可能です。参考：履歴/トレンドからのグラフ生成。 **avg** - 平均値を表示 **last** - 最新値を表示。この関数はグラフタイプとして円グラフ/爆発円グラフが選択されている場合のみ利用可能です。 **max** - 最大値を表示 **min** - 最小値を表示
	描画スタイル	描画スタイルを選択します（通常グラフのみ利用可能。積み上げグラフでは常に塗りつぶし領域が使用されます）。線、太線、塗りつぶし領域、点、破線、グラデーション線から選択できます。
	Y 軸側	アイテムデータを表示する Y 軸側を選択します - 左、右。
	色	アイテムデータに適用する色を選択します。

グラフのプレビュー

プレビュータブでは、グラフのプレビューが表示され、作成している内容をすぐに確認できます。



テンプレートアイテムの場合、プレビューにはデータが表示されないことに注意してください。



この例では、トリガーレベルを表示する破線の太線と、凡例に表示されるトリガー情報に注目してください。

Note:
表示できるトリガーラインは 3 本までです。トリガーが 3 つ以上ある場合は、重大度の低いトリガーが優先して表示されます。

 グラフの高さが 120 ピクセル未満に設定されている場合、凡例にはトリガーが表示されません。

3 アドホックグラフ

概要

シンプルグラフは 1 つのアイテムのデータにアクセスするのに便利であり、カスタムグラフはカスタマイズオプションを提供しますが、どちらも複数のアイテムを簡単に比較するグラフをすばやく作成することはできません。

この問題に対処するために、複数のアイテムのアドホックグラフを非常に迅速に作成することができます。

設定

アドホックグラフを作成するには、以下の手順を実行します。

- 監視データ → 最新データに移動します
- フィルタを使用して表示したいアイテムを表示します
- グラフ化したいアイテムのチェックボックスをマークします
- 積算グラフを表示または グラフを表示ボタンをクリックします

グラフが即座に作成されます。

グラフに表示される線が多くなりすぎないように、各アイテムの平均値のみが表示されます（最小/最大値の線は表示されません）。トリガーやトリガー情報はグラフに表示されません。

作成されたグラフウィンドウでは、期間選択が利用でき、「通常」の折れ線グラフから積算グラフへの切り替え（およびその逆）が可能です。

4 グラフの集計

概要

ダッシュボードのグラフおよび円グラフウィジェットで利用可能な集計関数を使用すると、すべての値の代わりに、選択した間隔（5 分、1 時間、1 日）の集計値を表示できます。

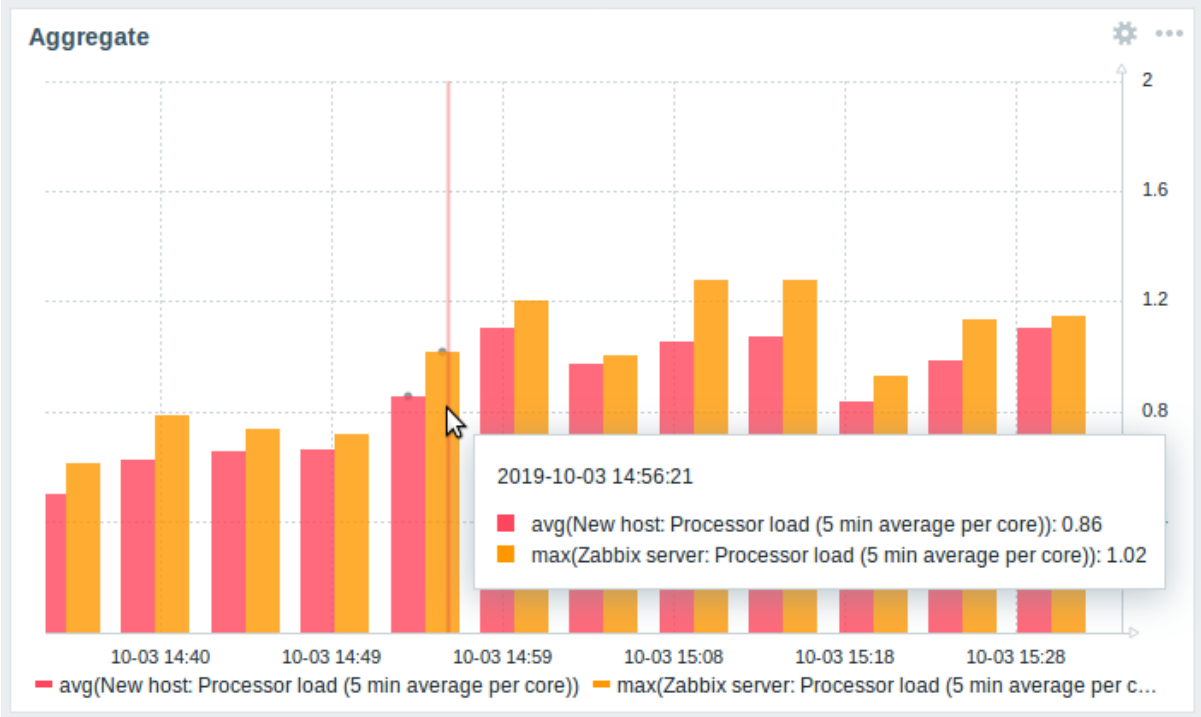
このセクションでは、グラフウィジェットの集計オプションについて詳しく説明します。

集計オプションは次のとおりです。

- min
- max

- avg
- count
- sum
- first（最初に表示される値）
- last（最後に表示される値）

データ集計の最も魅力的な使い方は、ある期間のデータを並べて比較できることです。



グラフ上のある時点にカーソルを合わせると、アイテムとその集計値に加えて、日付と時刻が表示されます。アイテムは括弧内に表示され、使用された集計関数が接頭辞として付けられます。グラフウィジェットにデータセットラベルが設定されている場合、ラベルは使用された集計関数が接頭辞として付けられ、括弧内に表示されます。これは実際の値の日付と時刻ではなく、グラフ上のポイントの日付と時刻であることに注意してください。

設定

集計のオプションは、グラフウィジェットの設定時にデータセット設定で利用できます。

集計関数と時間間隔を選択できます。データセットには複数のアイテムが含まれる場合があるため、各アイテムごとに集計データを個別に表示するか、すべてのデータセットアイテムを 1 つの集計値として表示するかを選択するオプションもあります。

使用例

Nginx サーバーへの平均リクエスト数

Nginx サーバーへの 1 日あたりの平均リクエスト数 (秒) を表示します。

- データセットに秒間リクエスト数の item を追加します。
- 集約関数 avg を選択し、インターバル 1d を指定します。
- 棒グラフが表示され、各棒は一日あたりの平均秒間リクエスト数を表します。

クラスタ間の最小週次ディスク容量

1 週間におけるクラスタ間の最小ディスク容量を表示します。

- データセットに追加します: ホスト `cluster*`, キー”Free disk space on /data”
- 集約関数 `min` を選択し、間隔 `1w` を指定します。
- 棒グラフが表示され、各棒はクラスタの各 /data ボリュームの 1 週間あたりの最小ディスク容量を表します。

2 ネットワークマップ

概要

ネットワークの管理をしている場合、インフラの概要をどこかで把握したいと思うかもしれません。
 そのために、Zabbix ではネットワークや好きなもののマップを作成することができます。

すべてのユーザがネットワークマップを作成することができます。
 マップはパブリック (すべてのユーザが利用可能) またはプライベート (選択されたユーザが利用可能) にすることができます。

configuring a network mapに進みます。

1 ネットワークマップの設定

概要

Zabbix でマップを設定するには、まずマップの一般的なパラメータを定義してマップを作成し、その後、実際のマップに要素やリンクを追加していきます。

マップには、ホスト、ホストグループ、トリガー、イメージ、または別のマップを要素として追加できます。

アイコンはマップ要素を表すために使用されます。アイコンとともに表示される情報を定義したり、最近の問題が特別な方法で表示されるように設定したりできます。アイコン同士をリンクし、リンク上に表示される情報も定義できます。

アイコンをクリックしてアクセスできるカスタム URL を追加することもできます。これにより、ホストアイコンをホストのプロパティにリンクしたり、マップアイコンを別のマップにリンクしたりできます。

マップ内の問題数は、原因となる問題のみが表示されます。

マップは監視 → マップで管理され、ここで設定、管理、表示が可能です。監視ビューでは、アイコンをクリックして一部のスクリプトや URL へのリンクを利用できます。

ネットワークマップはベクターグラフィックス（SVG）に基づいています。

パブリックマップとプライベートマップ

Zabbix のすべてのユーザー（管理者以外のユーザーも含む）は、ネットワークマップを作成できます。マップには所有者（作成したユーザー）がいます。マップはパブリックまたはプライベートにすることができます。

- パブリックマップはすべてのユーザーに表示されますが、表示するにはユーザーが少なくとも 1 つのマップ要素への読み取り権限を持っている必要があります。パブリックマップは、ユーザー/ユーザーグループがこのマップに対して読み書き権限を持ち、かつ対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合に編集できます。
- プライベートマップは、所有者と所有者が共有したユーザー/ユーザーグループにのみ表示されます。通常の（スーパー管理者以外の）ユーザーは、自分が所属するグループやユーザーとしか共有できません。管理者レベルのユーザーは、所有者や共有ユーザーリストに属していなくてもプライベートマップを表示できます。プライベートマップは、マップの所有者と、ユーザー/ユーザーグループがこのマップに対して読み書き権限を持ち、かつ対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合に編集できます。

ユーザーが読み取り権限を持たないマップ要素は、グレーアウトしたアイコンで表示され、要素に関するすべてのテキスト情報は非表示になります。ただし、ユーザーがトリガーへの権限を持っていなくても、トリガーラベルは表示されます。

マップに要素を追加するには、ユーザーがその要素に対して少なくとも読み取り権限を持っている必要があります。

マップの作成

マップを作成するには、以下の手順を実行します。

- 監視データ → マップに移動します
- すべてのマップが表示されるビューに移動します
- マップの作成をクリックします

既存のマップの設定フォームで 複製ボタンを使用して、新しいマップを作成することもできます。このマップは、既存のマップの一般的なレイアウト属性や要素を含む、すべてのプロパティを持ちます。

マップタブには、マップの一般的な属性が含まれています。

必須入力フィールドは赤いアスタリスクでマークされています。

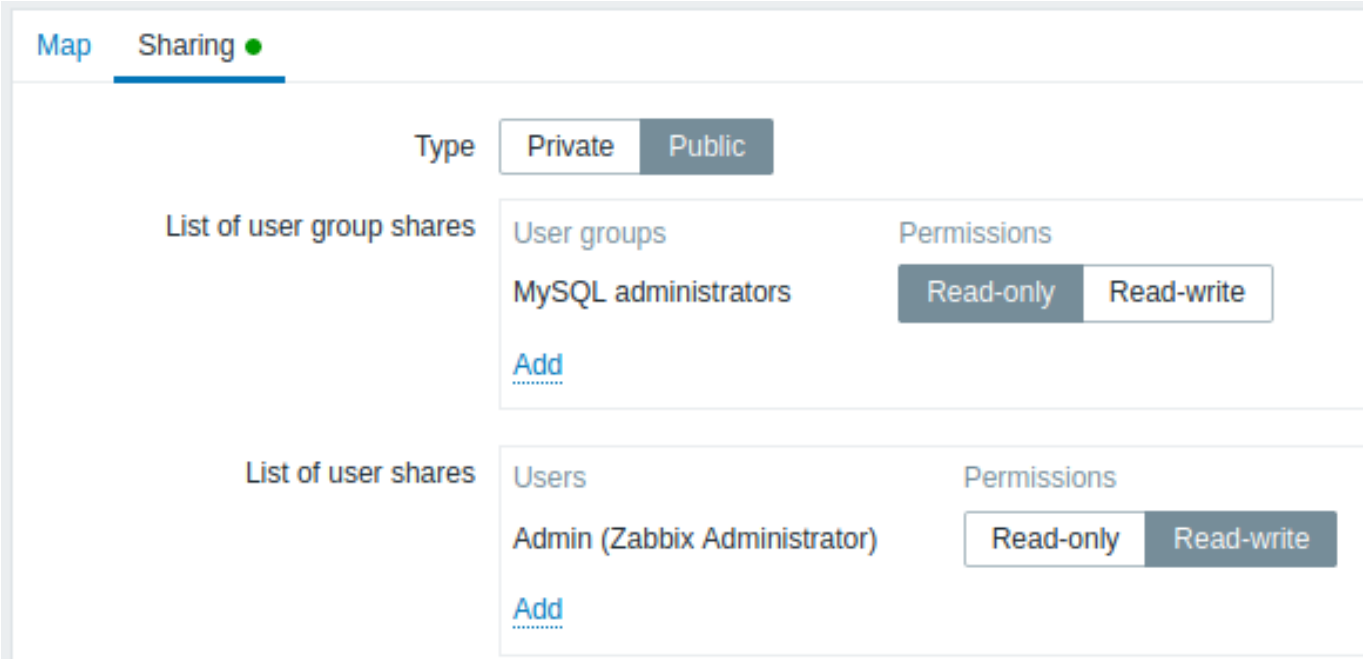
マップの一般的な属性:

パラメータ	説明
所有者	マップの所有者名。
名前	一意のマップ名。
幅	マップの幅（ピクセル単位）。
高さ	マップの高さ（ピクセル単位）。

パラメータ	説明
背景画像	背景画像の使用: 画像なし - 背景画像なし（白背景） 画像 - 選択した画像を背景画像として使用します。地図やその他の画像を使用してマップを強化できます。
背景の拡大縮小	背景画像の拡大縮小の使用: なし - 拡大縮小しない; 比率を維持 - 画像の比率を維持しつつ、マップ全体をカバーします（デフォルト）。
アイコンの自動マッピング	管理 → 一般 → アイコンマッピングで設定された自動アイコンマッピングを使用するように設定できます。アイコンマッピングを使用すると、特定のホストインベントリフィールドに対して特定のアイコンを割り当てることができます。
アイコンのハイライト	このボックスをチェックすると、マップ要素がハイライト表示されます。 アクティブなトリガーを持つ要素は、最も高い深刻度のトリガーと同じ色の円形の背景が表示されます。さらに、すべての問題が確認済みの場合は、円の周りに太い緑色の線が表示されます。 「無効」または「メンテナンス中」のステータスの要素には、それぞれ灰色とオレンジ色の四角い背景が表示されます。 詳細はマップの表示 も参照してください。
トリガーステータス変更時の要素のマーク	トリガーステータスの最近の変更（最近の問題または解決）は、要素アイコンのラベルがない 3 辺に内向きの赤い三角形のマーカーでハイライト表示されます。マーカーは 30 分間表示されます。
問題の表示	マップ要素で問題をどのように表示するかを選択します: 単一の問題を展開 - 問題が 1 つだけの場合は問題名を表示し、それ以外の場合は問題の総数を表示します。 問題数 - 問題の総数を表示します 問題数と最も深刻な問題を展開 - 最も深刻な問題の名前と問題の総数を表示します。 「最も深刻な問題」は、問題の深刻度と、同じ場合は問題イベント ID（ID が大きいまたは発生が遅いものが優先）に基づいて決定されます。トリガーマップ要素の場合は、問題の深刻度と、同じ場合はトリガーリスト内のトリガーの位置に基づきます。同じトリガーに複数の問題がある場合は、最も新しいものが表示されます。
高度なラベル	このボックスをチェックすると、要素タイプごとに個別のラベルタイプを定義できます。
マップ要素ラベルタイプ	マップ要素に使用するラベルタイプ: ラベル - マップ要素のラベル **IP** アドレス - IP アドレス 要素名 - 要素名（例: ホスト名） ステータスのみ - ステータスのみ（OK または PROBLEM） なし - ラベルは表示されません
マップ要素ラベルの位置	マップ要素に対するラベルの位置: 下 - マップ要素の下 左 - 左側 右 - 右側 上 - マップ要素の上
マップ要素ラベルの表示	マップ要素ラベルの表示方法を選択します: 常に表示 - 常に表示（デフォルト）; 自動非表示 - ホバーまたは選択されていない場合はラベルを非表示にします。
リンクラベルの表示	リンクラベルの表示方法を選択します: 常に表示 - 常に表示（デフォルト）; 自動非表示 - ホバーまたは選択されていない場合はラベルを非表示にします。
問題の表示方法	問題数の表示方法: すべて - 問題の総数を表示します 分離 - 未確認の問題数を、問題の総数とは別に表示します 未確認のみ - 未確認の問題数のみを表示します
最小トリガー深刻度	選択した最小深刻度レベル未満の問題はマップに表示されません。 たとえば、警告を選択した場合、情報および 未分類レベルのトリガーによる変更はマップに反映されません。
抑制された問題の表示	ホストのメンテナンスにより抑制（非表示）される問題も表示する場合は、チェックボックスをオンにします。
URL	各要素タイプごとに URL（最大 2048 文字）を定義できます。URL のラベルも定義できます。これらは、マップの表示モードで要素をクリックしたときにリンクとして表示されます。 マップ URL 名と値にはマクロを使用できます。詳細はサポートされているマクロ を参照し、「map URL names and values」で検索してください。

共有

共有タブには、マップの種類や、プライベートマップの共有オプション（ユーザーグループ、ユーザー）が含まれています。



パラメータ	説明
タイプ	マップの種類を選択: プライベート - 選択したユーザーグループとユーザーのみに表示されます パブリック - すべてのユーザーに表示されます
ユーザーグループ共有のリスト	マップにアクセスできるユーザーグループを選択します。 読み取り専用または読み書きアクセスを許可できます。
ユーザー共有のリスト	マップにアクセスできるユーザーを選択します。 読み取り専用または読み書きアクセスを許可できます。

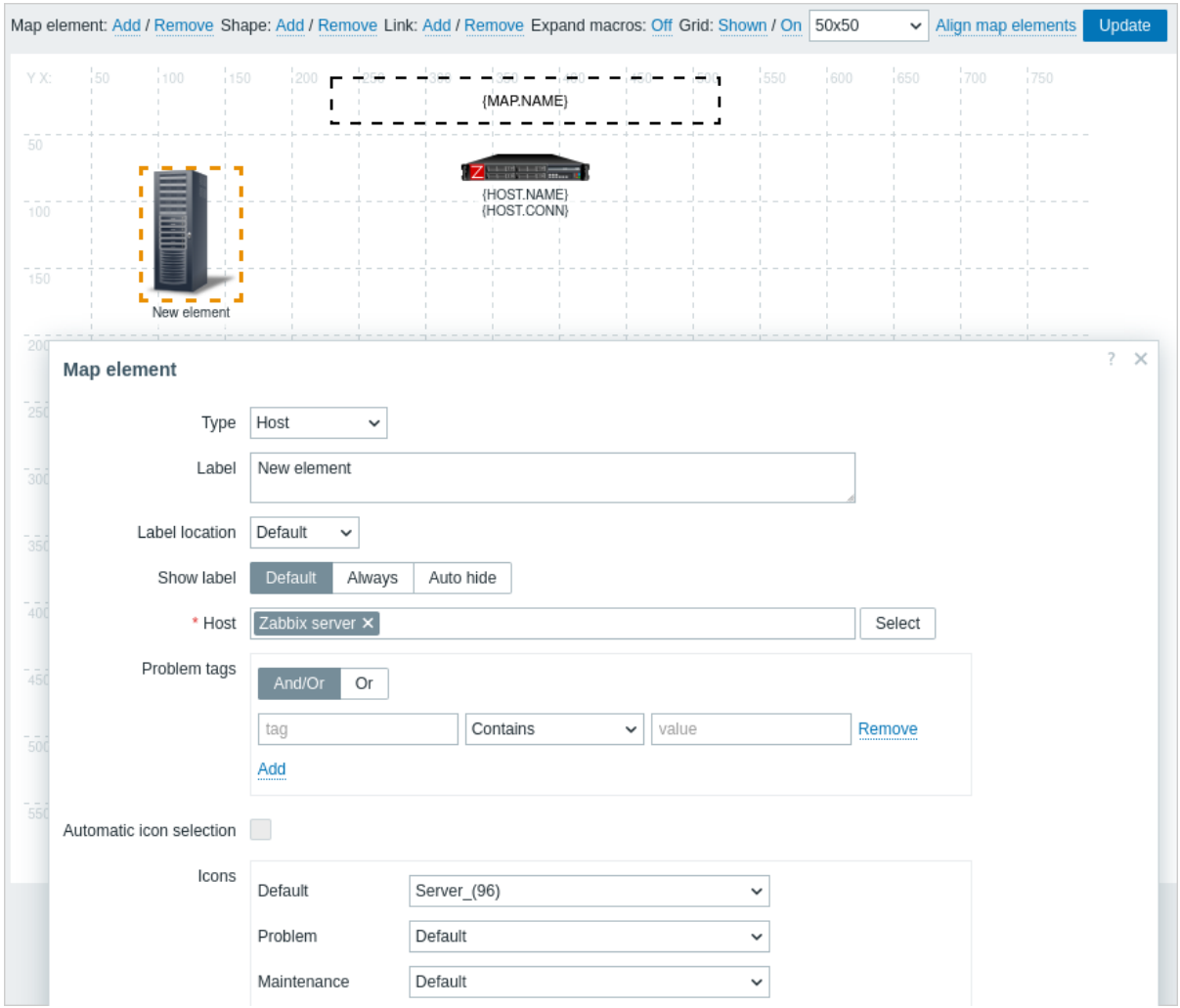
追加をクリックしてこのマップを保存すると、名前、サイズ、およびいくつかの設定を持つ空のマップが作成されます。次に、いくつかの要素を追加する必要があります。そのためには、マップリストで編集をクリックして編集可能な領域を開きます。

要素の追加

要素を追加するには、マップ要素の横にある追加をクリックします。新しい要素はマップの左上隅に表示されます。ドラッグ＆ドロップで好きな場所に移動できます。

グリッドオプションが「オン」の場合、要素は常にグリッドに揃えられます（ドロップダウンからさまざまなグリッドサイズを選択したり、グリッドの表示/非表示を切り替えたりできます）。要素をどこにでも自由に配置したい場合は、オプションを「オフ」にしてください。(マップ要素の整列をクリックすることで、後からランダムな要素をグリッドに揃えることもできます。)

要素をいくつか配置したら、名前を付けるなどして区別したくなるかもしれません。要素をクリックするとフォームが表示され、要素タイプの設定、名前の付与、別のアイコンの選択などができます。



マップ要素の属性:

パラメータ	説明
タイプ	要素のタイプ: ホスト - 選択したホストのすべてのトリガーのステータスを表すアイコン マップ - マップのすべての要素のステータスを表すアイコン トリガー - 1 つまたは複数のトリガーのステータスを表すアイコン ホストグループ - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表すアイコン イメージ - リソースにリンクされていないアイコン
ラベル	アイコンのラベル。任意の文字列。 マクロや複数行の文字列が使用できます。 このフィールドではマクロ式がサポートされており、avg、last、min、max 関数のみ、時間をパラメータとして使用できます（例: {?avg(/host/key,1h)}）。 サポートされているマクロの一覧については、サポートされているマクロを参照し、「map element labels」で検索してください。
ラベルの表示	要素ラベルの表示方法を選択: デフォルト - マップ設定のマップ要素ラベルの表示設定を使用（デフォルト）; 常に - 常に表示（デフォルト）; 自動非表示 - ホバーまたは選択されていない場合はラベルを非表示にする。
ラベルの位置	アイコンに対するラベルの位置: デフォルト - マップのデフォルトのラベル位置 下 - アイコンの下 左 - 左側 右 - 右側 上 - アイコンの上
ホスト	要素タイプが「ホスト」の場合はホストを入力します。このフィールドはオートコンプリートなので、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。スクロールして選択します。選択を削除するには「x」をクリックします。

パラメータ	説明
マップ	要素タイプが「マップ」の場合はマップを選択します。このフィールドはオートコンプリートなので、マップ名の入力を始めると一致するマップのドロップダウンが表示されます。スクロールして選択します。選択を削除するには「x」をクリックします。
トリガー	要素タイプが「トリガー」の場合は、下の新しいトリガーフィールドで 1 つまたは複数のトリガーを選択し、追加をクリックします。 選択したトリガーの順序は、同じ重大度のトリガー内でのみ変更できます。複数のトリガーを選択すると、編集モードと表示モードの両方で {HOST.*} マクロの解決にも影響します。 // 1 編集モード// では、最初に表示される {HOST.*} マクロはリスト内の最初のトリガー（トリガーの重大度に基づく）に応じて解決されます。 // 2 表示モード// では、問題の表示パラメータ（一般マップ属性）が影響します。 * 単一の問題を展開モードが選択されている場合、最初に表示される {HOST.*} マクロは、最新の検出された問題トリガー（重大度に関係なく）または（問題が検出されていない場合は）リスト内の最初のトリガーに応じて解決されます。 * 問題数と最も重大なものを展開モードが選択されている場合、最初に表示される {HOST.*} マクロはトリガーの重大度に応じて解決されます。
ホストグループ	要素タイプが「ホストグループ」の場合はホストグループを入力します。このフィールドはオートコンプリートなので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。スクロールして選択します。選択を削除するには「x」をクリックします。
タグ	ウィジェットに表示される問題の数を制限するためのタグを指定します。特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化される **Or** - いずれか 1 つの条件を満たせば十分 このフィールドはホストおよびホストグループの要素タイプで利用できます。
自動アイコン選択	この場合、アイコンマッピングが使用され、表示するアイコンが決定されます。
アイコン	デフォルト、問題、メンテナンス、無効の各ケースで異なるアイコンを表示することができます。
X 座標	マップ要素の X 座標。
Y 座標	マップ要素の Y 座標。
URL	要素固有の URL（最大 2048 文字）を要素に設定できます。URL のラベルも定義できます。これらは、マップの表示モードでユーザーが要素をクリックしたときにリンクとして表示されます。要素に独自の URL があり、そのタイプに対してマップレベルの URL が定義されている場合、それらは同じメニューで組み合わされます。 マップ要素名や値にはマクロが使用できます。詳細はサポートされているマクロを参照し、「map URL names and values」で検索してください。

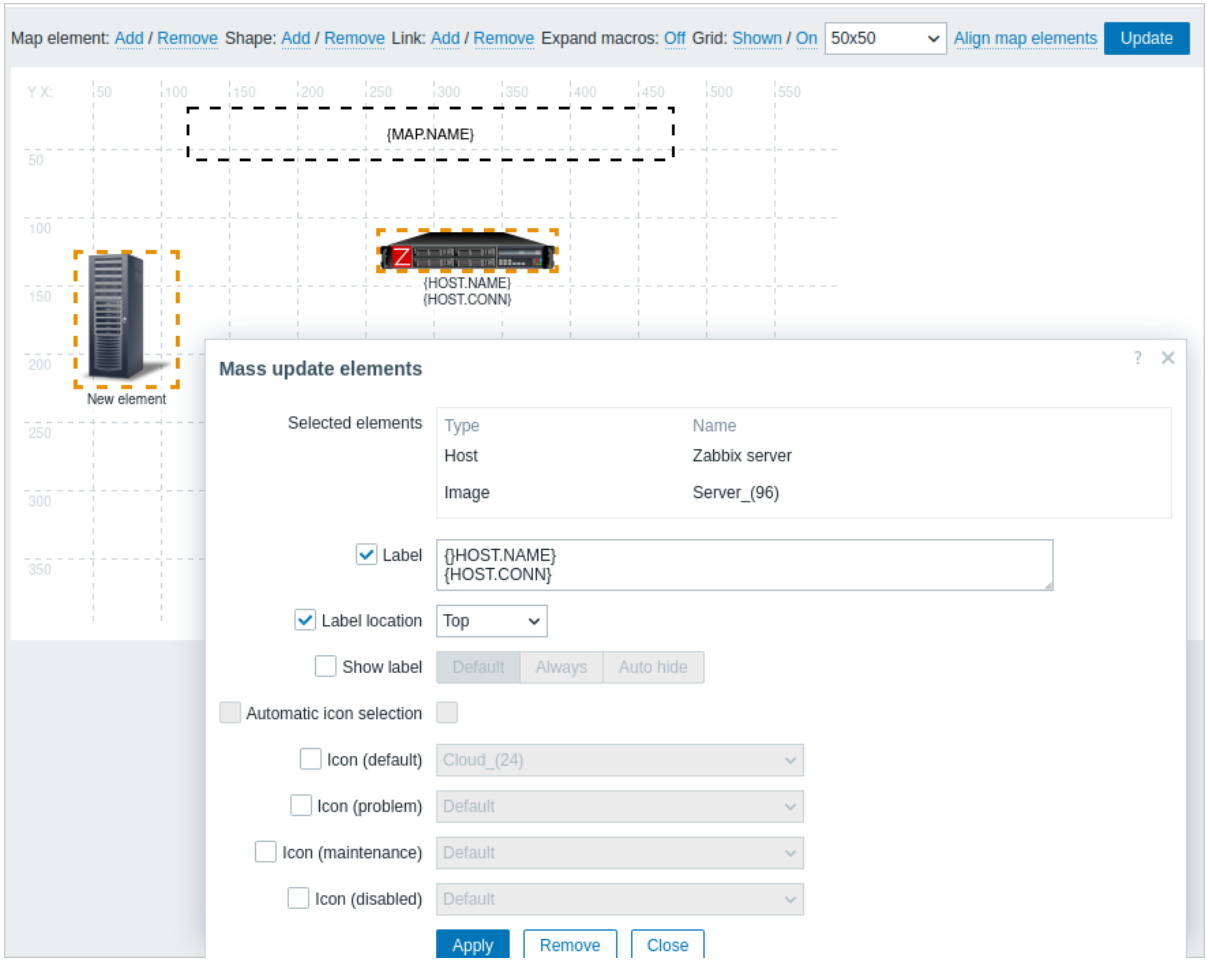
Attention:
追加した要素は自動的に保存されません。ページから移動すると、すべての変更が失われる可能性があります。
したがって、右上の更新ボタンをクリックすることをお勧めします。クリックすると、続くポップアップで何を選択しても変更は保存されます。
選択したグリッドオプションも各マップと一緒に保存されます。

要素の選択

要素を選択するには、1 つを選択してから *Ctrl* を押しながら他の要素を選択します。

また、編集可能な領域で矩形をドラッグし、その中のすべての要素を選択することで、複数の要素を選択することができます。

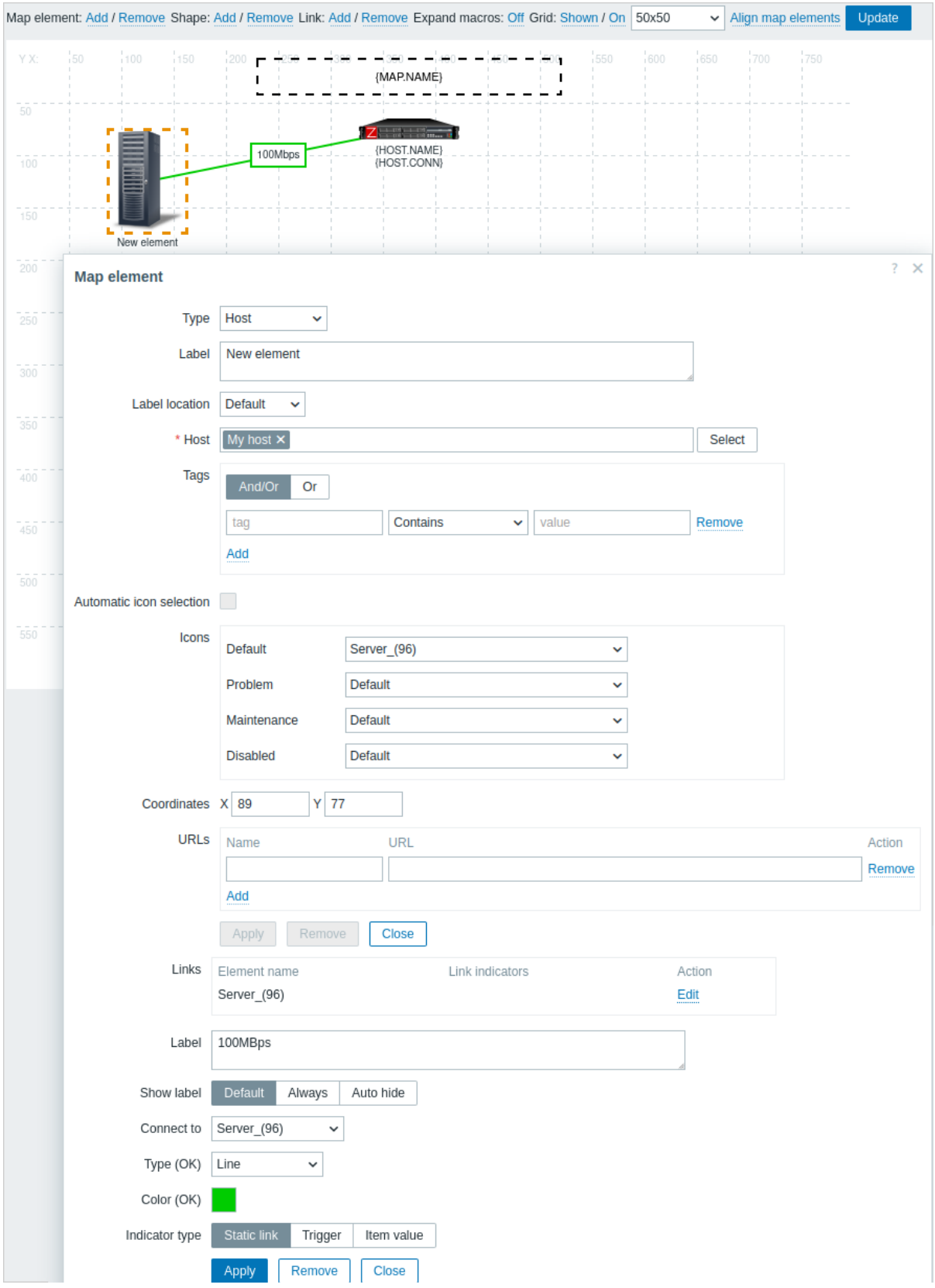
複数の要素を選択すると、要素のプロパティフォームが一括更新モードに移行し、選択した要素の属性を一括して変更する
 ことができます。そのためには、チェックボックスを使って属性をマークし、その属性に新しい値を入力します。
 ここでマクロを使用することもできます (例えば、要素のラベルに {HOST.NAME} を使用)。



要素のリンク

マップにいくつかの要素を配置したら、それらをリンクし始める時です。2 つの要素をリンクするには、まずそれらを選択する必要があります。要素を選択した状態で、リンクの横にある追加をクリックします。

リンクが作成されると、単一要素フォームに追加のリンクセクションが表示されます。編集をクリックしてリンク属性を編集します。



リンク属性:

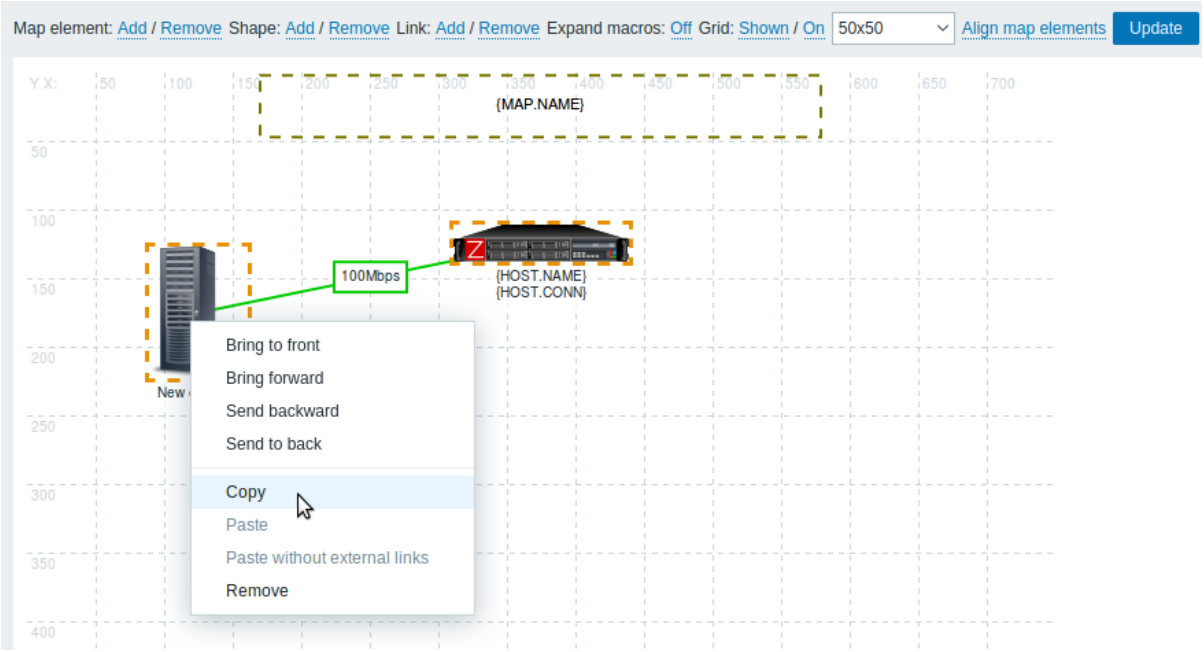
パラメータ	説明
ラベル	リンクの上に表示されるラベル。 このフィールドではマクロを含む式がサポートされますが、avg、last、min、max 関数のみがサポートされ、パラメータとして時間を指定します（例: {?avg(/host/key,1h)}）。
ラベルの表示	リンクラベルの表示方法を選択: デフォルト - マップ設定のマップ要素ラベルの表示設定を使用（デフォルト）； 常に - 常に表示（デフォルト）； 自動非表示 - ホバーまたは選択されていない場合はラベルを非表示にします。

パラメータ	説明
接続先	リンクが接続する要素。
タイプ *(OK)*	デフォルトのリンクスタイル: 線 - 一本線 太線 - 太い線 点線 - 点線 破線 - 破線
色 *(OK)*	デフォルトのリンク色。
インジケータータイプ	リンクインジケーターのタイプを選択: 静的リンク - インジケーターなし; トリガー - トリガーベースのリンクインジケーターを許可; アイテム値 - アイテム値ベースのリンクインジケーターを許可。
アイテム	アイテムを選択します。このアイテムの値がリンクスタイルに影響します。 このフィールドは、インジケータータイプフィールドで「アイテム値」が選択されている場合のみ利用可能です。
インジケーター	リンクインジケーターのリスト。 トリガー インジケータータイプフィールドで「トリガー」が選択されている場合、追加をクリックしてトリガーを追加します。トリガースタイル（色、線の種類）を選択します。トリガーが PROBLEM ステータスの場合、そのスタイルがリンクに適用されます。 アイテム値のしきい値/パターン インジケータータイプフィールドで「アイテム値」が選択されている場合、アイテム値のしきい値（数値アイテムの場合）またはパターン（テキストアイテムの場合）を追加します。しきい値/パターンのスタイル（色、線の種類）を選択します。アイテム値が指定したしきい値に達した場合（またはパターンに一致した場合）、そのスタイルがリンクに適用されます。

要素の移動とコピーペースト

選択した要素の 1 つをクリックし、マウスボタンを押したままカーソルを目的の場所に移動すると、選択した複数の要素を
 マップ内の別の場所に 移動することができます。

要素を選択し、マウスの右ボタンで選択した要素をクリックし、メニューから **Copy** を選択することで、1 つまたは
 複数の要素を コピーすることができます。

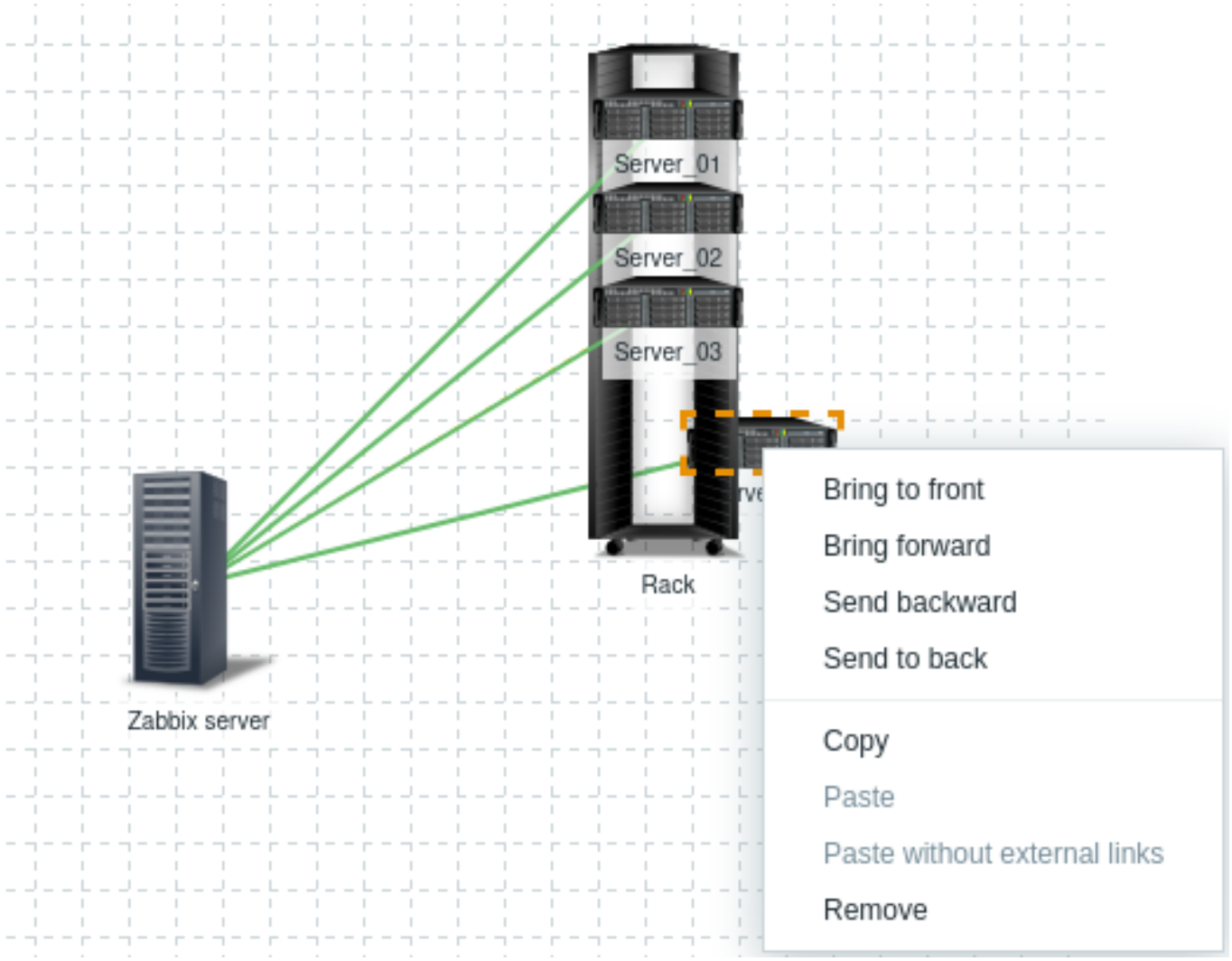


要素を貼り付けるには、マウスの右ボタンで地図上の領域をクリックし、メニューから *Paste* を選択します。*Paste without external links* を選択すると、選択した要素間のリンクのみを残して貼り付けられます。

コピー＆ペーストは、同じブラウザウィンドウ内で動作します。キーボードショートカットはサポートされていません。

要素の順序

ある要素を他の要素の前面または背面に移動するには、要素を右クリックし、前面に移動/最前面に移動または背面に移動/最背面に移動を選択します。

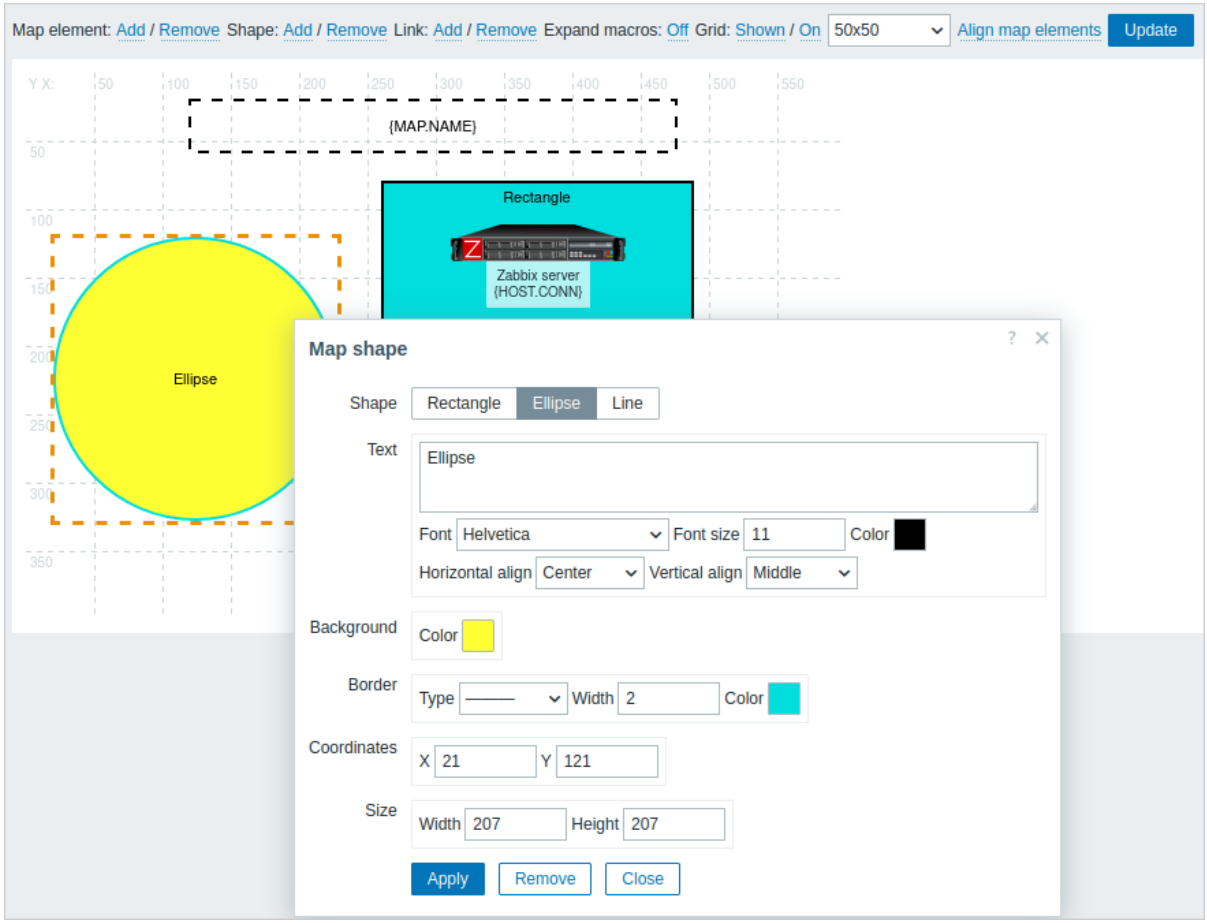


図形の追加

マップ要素に加えて、いくつかの図形を追加することも可能です。図形はマップ要素ではなく、単なる視覚的な表現です。例えば、長方形の図形は、いくつかのホストをグループ化するための背景として使用できます。長方形と楕円形の図形を追加できます。

図形を追加するには、図形の横にある追加をクリックします。新しい図形はマップの左上隅に表示されます。ドラッグ＆ドロップで好きな場所に移動できます。

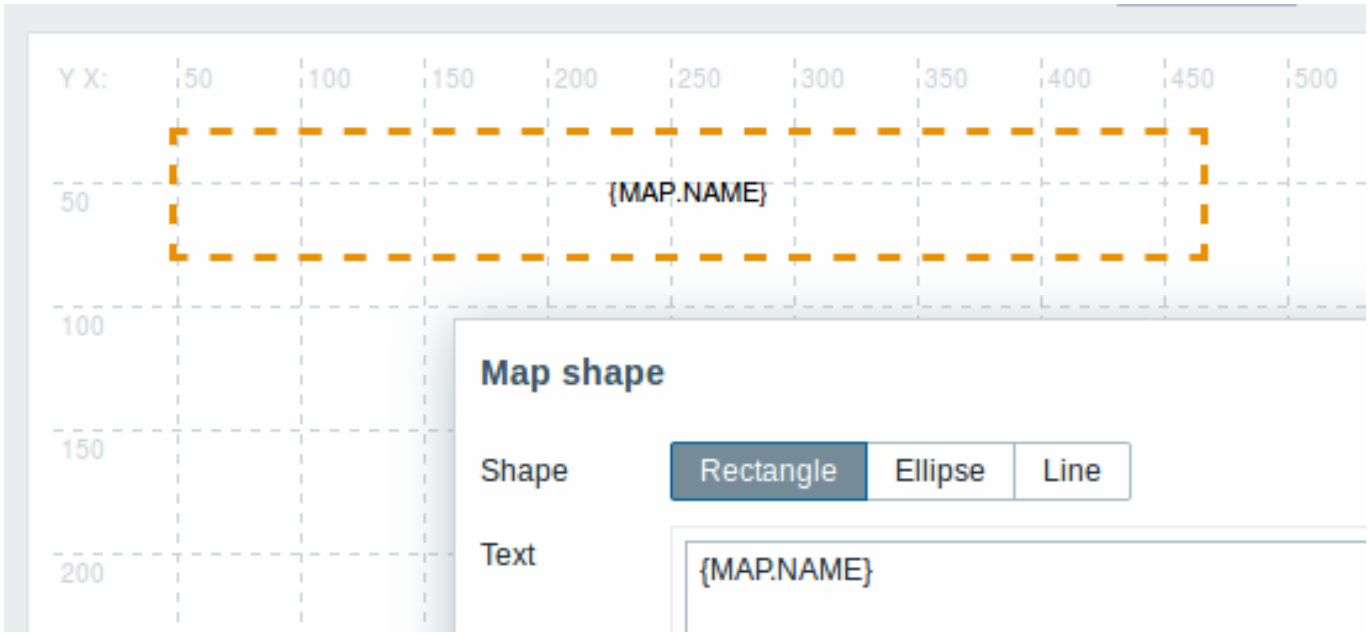
新しい図形はデフォルトの色で追加されます。図形をクリックするとフォームが表示され、図形の見た目をカスタマイズしたり、テキストを追加したりできます。



図形を選択するには、1 つ選択してから *Ctrl* を押しながら他の図形を選択します。複数の図形を選択した状態で、要素と同様に共通プロパティを一括更新できます。

図形内にテキストを追加できます。テキスト内ではマクロ式がサポートされていますが、`avg`、`last`、`min`、`max` 関数のみがサポートされ、パラメータとして時間を指定します（例：`{?avg(/host/key,1h)}`）。

テキストのみを表示する場合は、図形の枠線を削除して図形を不可視にできます（枠線フィールドで「なし」を選択）。例えば、上記のスクリーンショットで表示されている {MAP.NAME} マクロは、実際にはテキスト付きの長方形の図形であり、マクロをクリックすると確認できます。



{MAP.NAME} は、マップの表示時に設定されたマップ名に解決されます。

テキスト内でハイパーリンクを使用すると、マップの表示時にクリック可能になります。

図形内のテキストの改行は常に「オン」です。ただし、楕円形内では、楕円形が長方形であるかのように行が折り返されます。単語の折り返しは実装されていないため、長い単語（図形に収まらない単語）は折り返されず、（マップ編集ページでは）マスクされる

か、(他のマップページでは）切り捨てられます。

線の追加

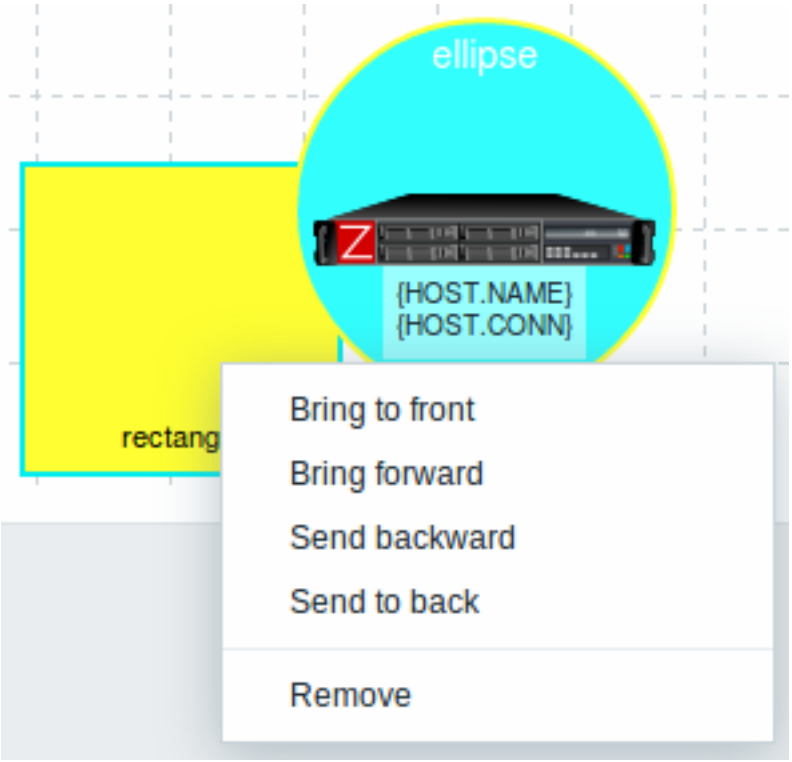
図形に加えて、いくつかの線を追加することもできます。線は、マップ内の要素や図形をリンクするために使用できます。

線を追加するには、図形の横にある追加をクリックします。新しい図形がマップの左上隅に表示されます。それを選択し、編集フォームで線をクリックして図形を線に変更します。その後、線の種類、幅、色などの線のプロパティを調整します。

map_line.png

図形と線の順序付け

ある図形を他の図形より前に（またはその逆に）持ってくるには、その図形をマウスの右ボタンでクリックし、地図の
 図形メニューを表示させます。

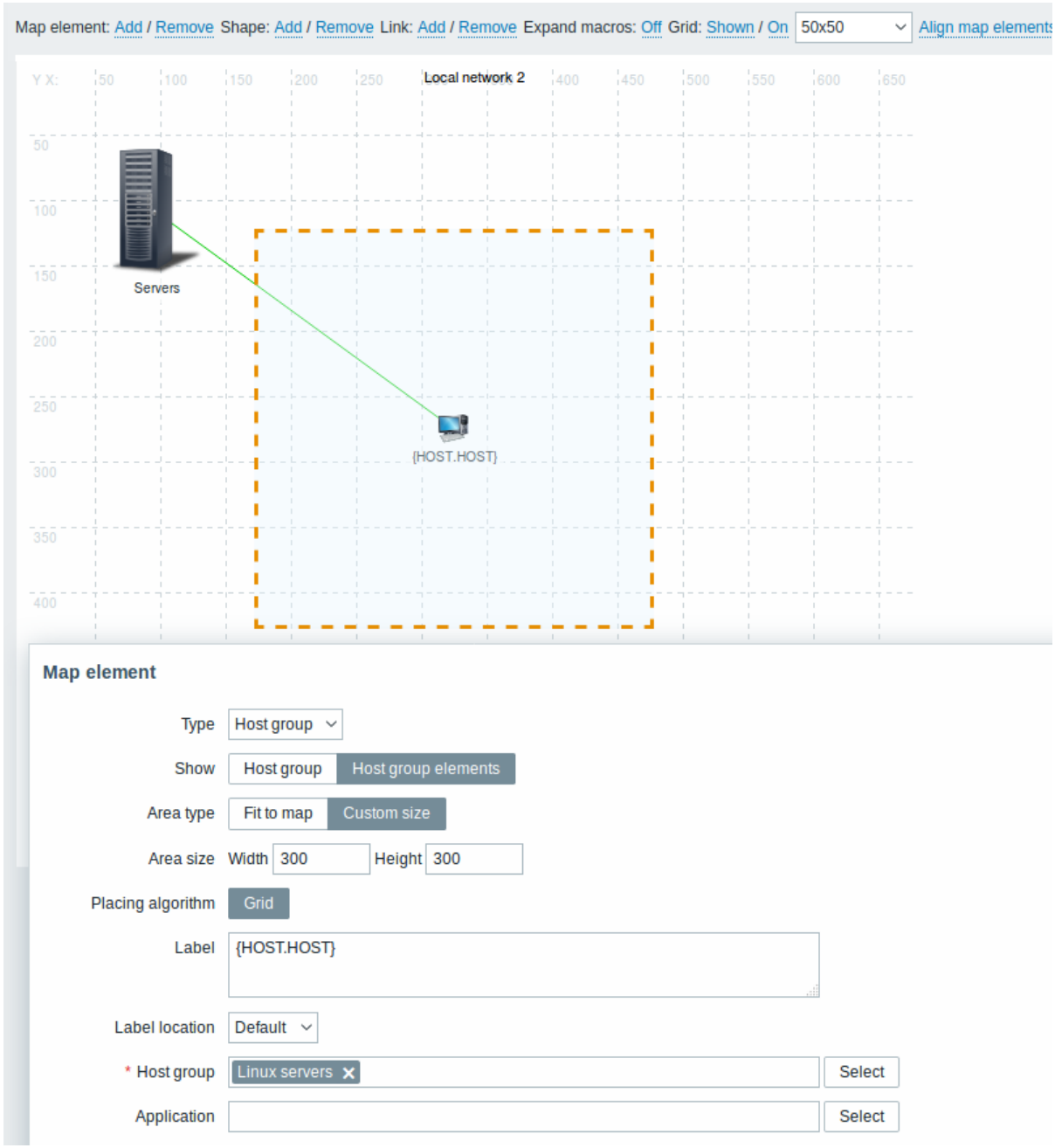


2 ホストグループの構成要素

概要

このセクションでは、ネットワークマップの設定時に” ホストグループ”タイプの要素を追加する方法について説明します。

設定



必須入力フィールドは赤いアスタリスクでマークされています。

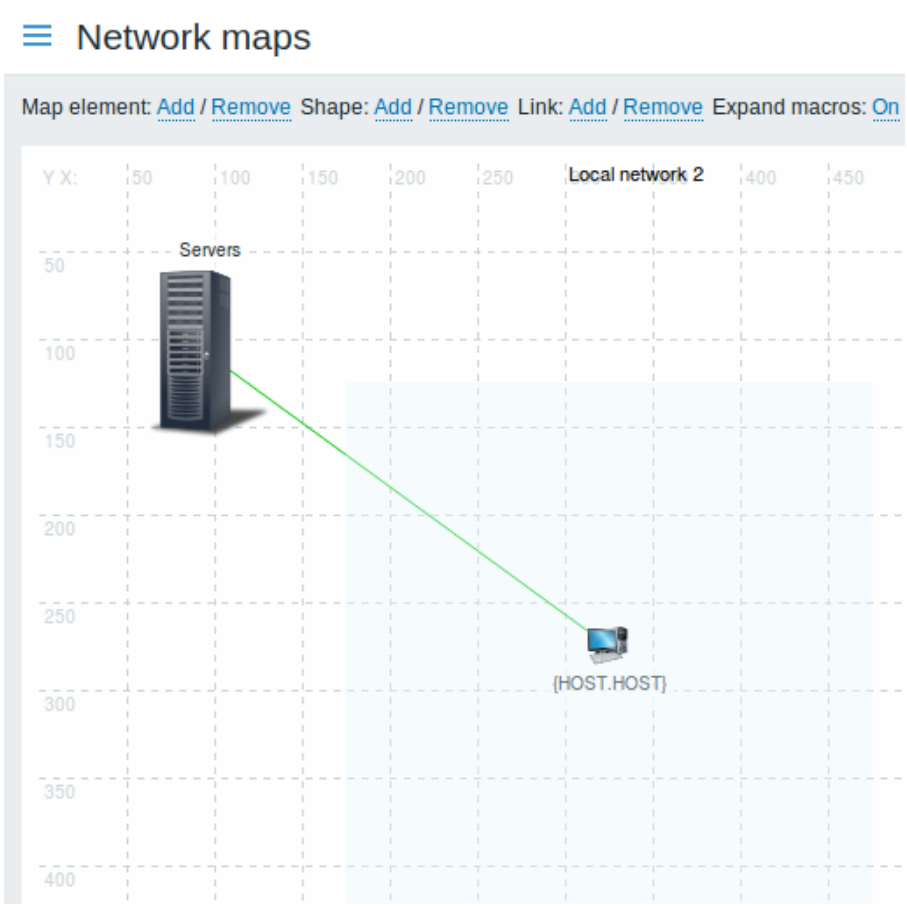
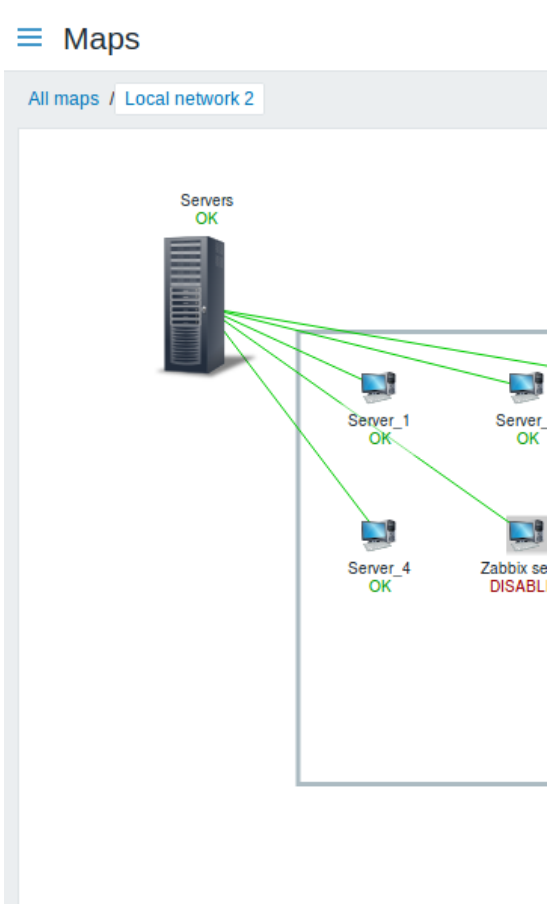
このテーブルは、ホストグループ要素タイプに特有のパラメータで構成されています:

パラメータ	説明
タイプ	要素のタイプを選択: ホストグループ - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表すアイコン
表示	表示オプション: ホストグループ - このオプションを選択すると、特定のホストグループに関する情報を表示する単一のアイコンが表示されます ホストグループ要素 - このオプションを選択すると、特定のホストグループの各要素 (ホスト) に関する情報を表示する複数のアイコンが表示されます
エリアタイプ	表示が「ホストグループ要素」に設定されている場合に利用可能: マップに合わせる - すべてのホストグループ要素がマップ内に均等に配置されます カスタムサイズ - すべてのホストグループ要素を表示するためのマップエリアを手動で設定します
エリアサイズ	エリアタイプが「カスタムサイズ」に設定されている場合に利用可能: 幅 - マップエリアの幅を指定する数値を入力 高さ - マップエリアの高さを指定する数値を入力
配置アルゴリズム	グリッド - すべてのホストグループ要素を表示する唯一のオプション

パラメータ	説明
ラベル	アイコンラベル、任意の文字列。 マクロや複数行の文字列をラベルに使用できます。 マップ要素のタイプが「ホストグループ」の場合、特定のマクロを指定すると、各ホストに関する情報を表示するマップビューに影響します。例えば、{HOST.IP}マクロを使用した場合、マップ編集ビューではマクロ {HOST.IP} 自体のみが表示されますが、マップビューでは各ホストの固有の IP アドレスが含まれ表示されます。

ホストグループ要素の表示

このオプションは、「ホストグループ要素」を表示オプションとして選択した場合に利用できます。*show* オプションとして「ホストグループ要素」を選択すると、最初はホストグループのアイコンが 1 つだけ表示されます。ただし、マップを保存してマップビューに移動すると、特定のホストグループのすべての要素（ホスト）がマップに含まれていることがわかります。

マップ編集ビュー	マップビュー
	

{HOST.HOST} マクロの使い方に注目してください。マップ編集ではマクロ名は解決されませんが、マップビューではすべてのホストのユニークな名前が表示されます。

3 リンクインジケーター

概要

ネットワークマップの要素間のリンクにインジケーターを割り当てることができます。

インジケーターはトリガーやアイテム値に基づくことができます。トリガーが障害状態になった場合や、アイテム値がしきい値に達した場合（数値アイテムの場合）、または正規表現に一致した場合（テキストアイテムの場合）に、リンクのスタイルや色を変更して表示することができます。

リンクを設定する際に、デフォルトのリンクタイプと色を設定します。リンクにインジケーターを割り当てることで、リンクのスタイルや色をトリガーの状態やアイテム値に応じて変更できるようになります。

例えば、割り当てたトリガーのいずれかが障害状態になった場合、リンクのスタイルや色がそれを反映して変化します。デフォルトのリンクが緑色の線だった場合でも、トリガーが障害状態になったことで、リンクが太い赤色になる（そのように定義していれば）ことがあります。

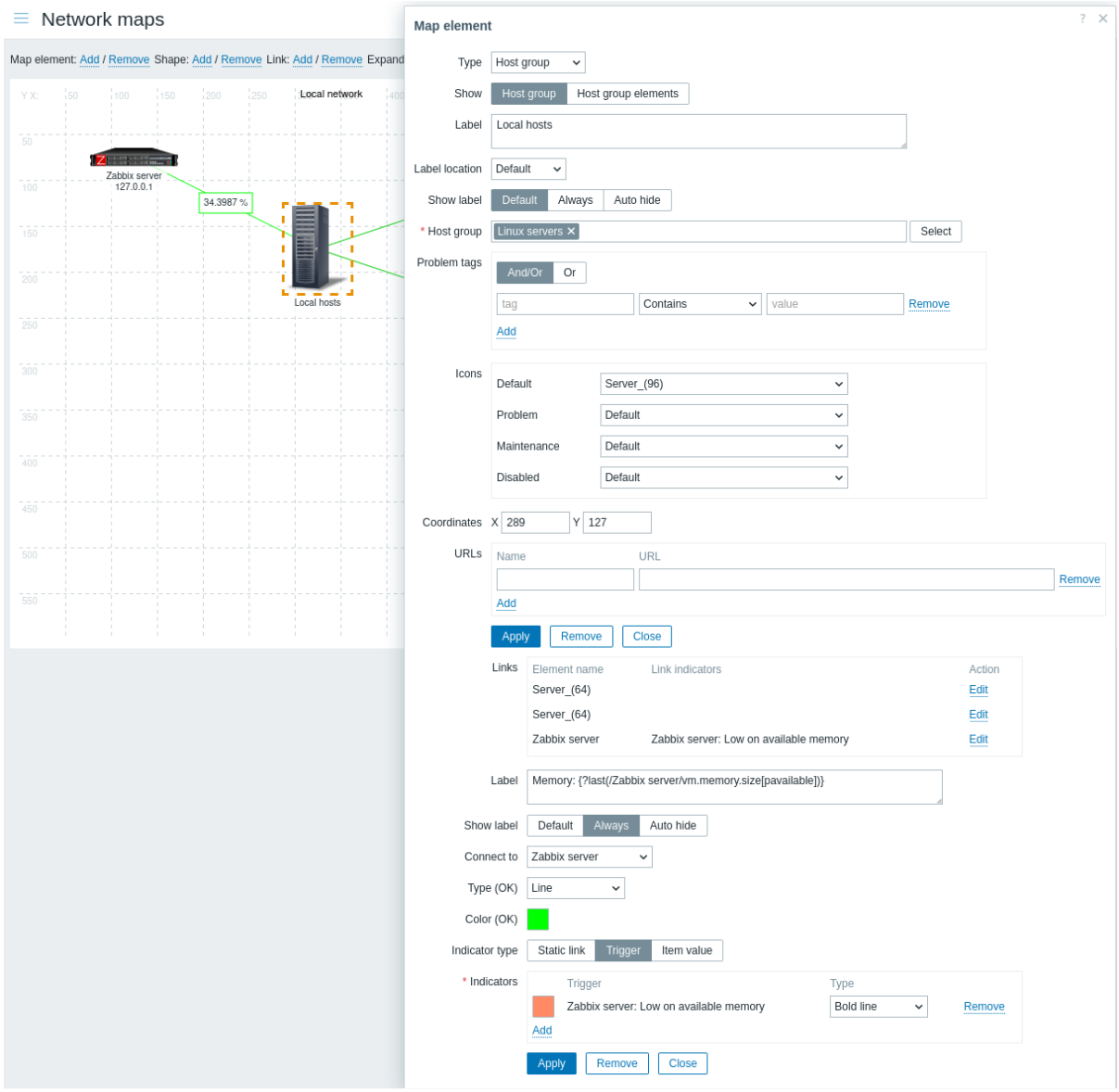
同様に、アイテム値が指定したしきい値に達した場合や、指定した正規表現に一致した場合にも、リンクのスタイルがそれを反映することがあります。

設定

トリガー

リンクインジケーターとしてトリガーを割り当てるには、以下の手順を実行します。

- マップ要素を選択します
- 該当するリンクのリンクセクションで編集をクリックします
- インジケータータイプとしてトリガーを選択します
- インジケーターブロックで追加をクリックし、1 つまたは複数のトリガーを選択します

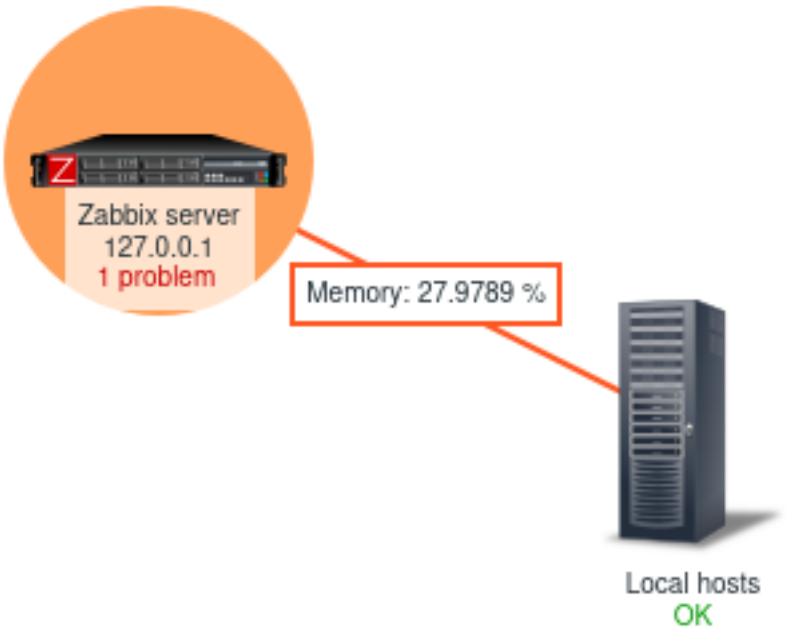


すべての必須入力フィールドには赤いアスタリスクが付いています。

追加されたトリガーはインジケーターリストで確認できます。

各インジケーターのリンクタイプと色はリストから直接設定できます。完了したら適用をクリックし、フォームを閉じて更新をクリックしてマップの変更を保存します。

監視 → マップでは、トリガーが障害状態になった場合に、該当するリンクタイプと色が表示されます。



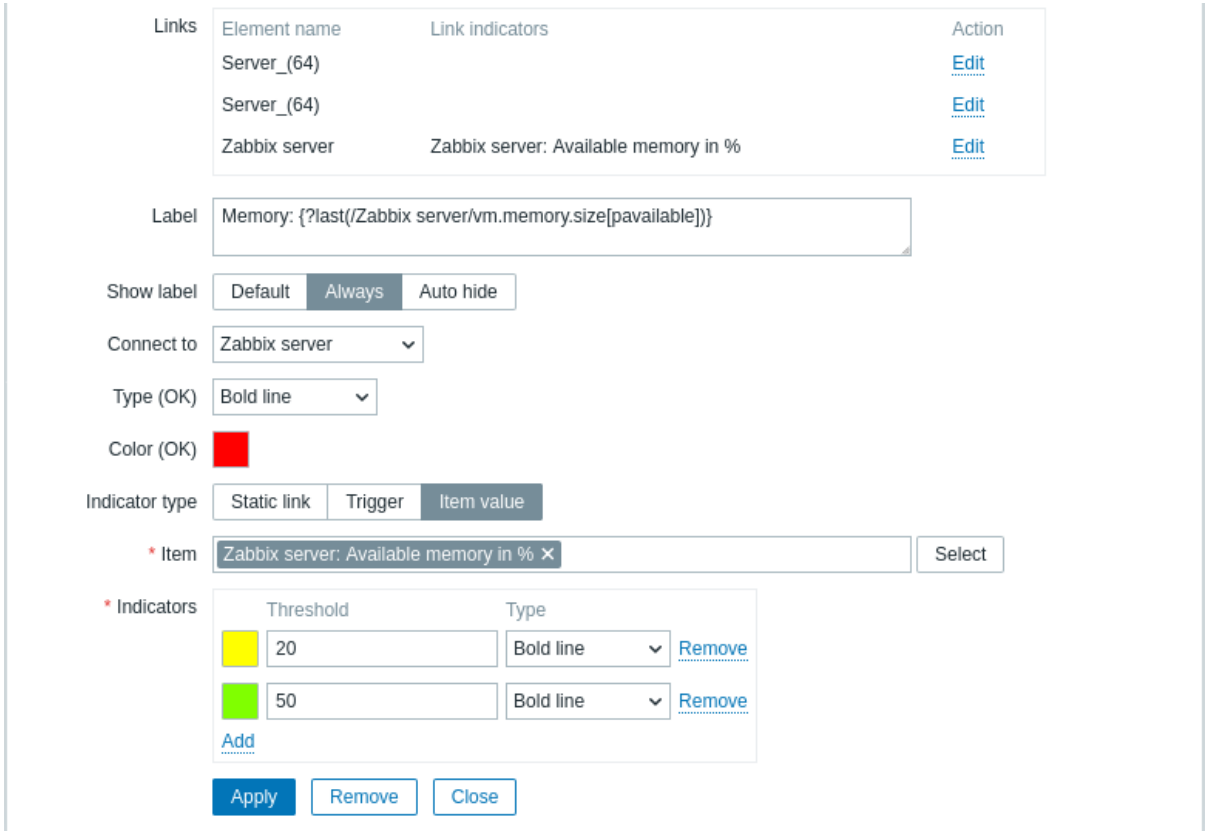
Note:
複数のトリガーが障害状態になった場合、最も深刻度の高い障害がリンクのスタイルと色を決定します。同じ深刻度のトリガーが同じマップリンクに割り当てられている場合は、ID が最も小さいものが優先されます。また、以下にも注意してください。

1. マップ設定の最小トリガー深刻度および抑制された障害を表示の設定は、考慮される障害に影響します。
2. 複数の障害を持つトリガー（複数障害生成）の場合、各障害はトリガーの深刻度と異なる深刻度（手動で変更されたもの）を持つ場合があり、異なるタグ（マクロによる）を持つ場合があり、抑制されている場合もあります。

アイテム値

リンクインジケーターとしてアイテム値を割り当てるには、以下の手順を実行します。

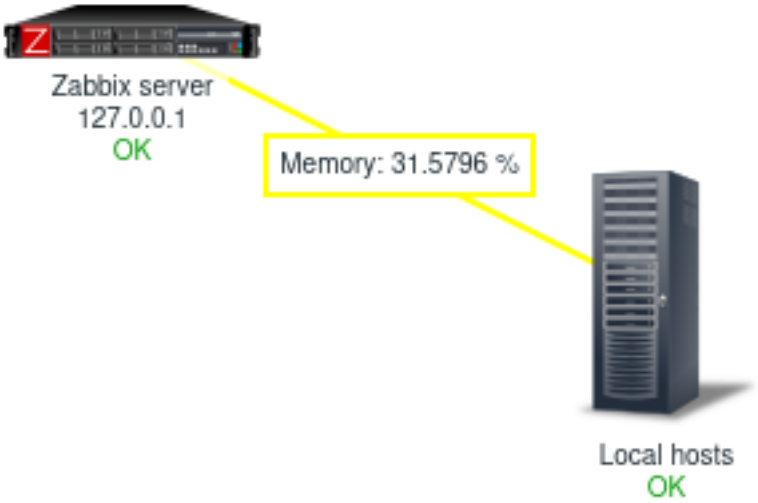
- マップ要素を選択します
- 該当するリンクのリンクセクションで編集をクリックします
- インジケータータイプとしてアイテム値を選択します
- アイテムを選択します
- インジケーターブロックで 1 つまたは複数のアイテム値のしきい値またはパターンを追加します



追加されたアイテムのしきい値/パターンはインジケーターリストで確認できます。

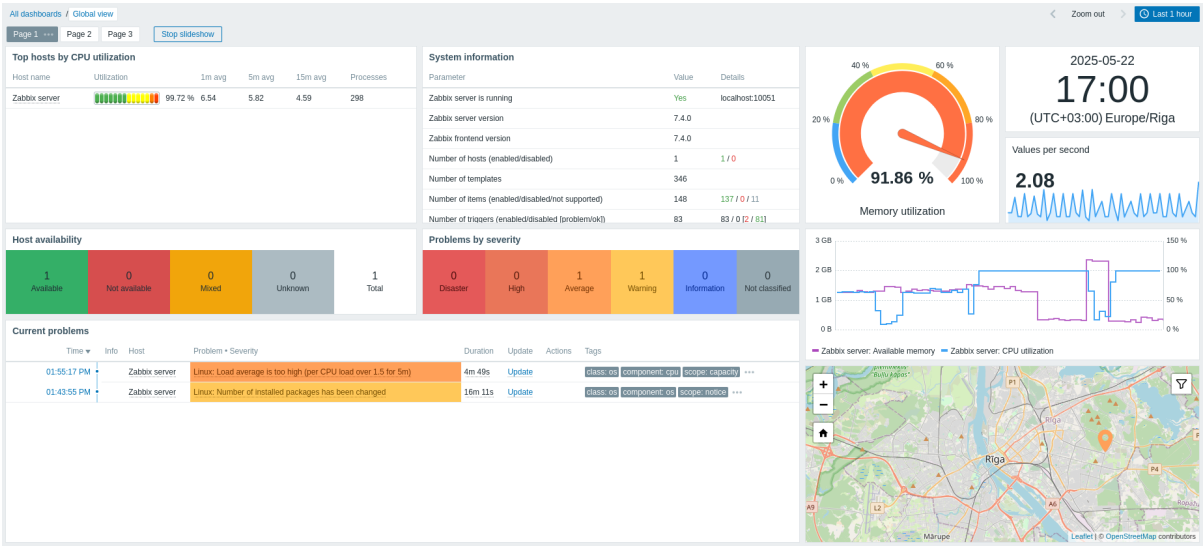
各インジケーターのリンクタイプと色はリストから直接設定できます。完了したら適用をクリックし、フォームを閉じて更新をクリックしてマップの変更を保存します。

監視 → マップでは、アイテム値が設定したしきい値（数値データ型の場合）に達した場合、または正規表現パターン（テキストデータ型の場合）に一致した場合に、該当するリンクタイプと色が表示されます。



3 ダッシュボード

ダッシュボード - グローバルダッシュボードとホストダッシュボードの両方とも - 最新のグラフ、地図、スライドショーなどのウィジェットやツールを備えた強力な視覚化プラットフォームを提供します。



8 テンプレートとテンプレートグループ

概要

テンプレートを使用することで、作業負荷を軽減し、Zabbix の設定を効率化することができます。テンプレートとは、複数のホストに便利に適用できるエンティティのセットです。

エンティティには以下が含まれます:

- アイテム
- トリガー
- グラフ
- ダッシュボード
- ローレベルディスカバリルール
- Web シナリオ

実際には多くのホストが同一または非常に似ているため、1 つのホストのために作成したエンティティ（アイテム、トリガー、グラフなど）のセットは、多くのホストにも有用であることが自然にわかります。もちろん、それらを各新規ホストにコピーすることもできますが、それでは多くの手作業が必要になります。代わりに、テンプレートを使えば、それらを 1 つのテンプレートにコピーし、必要なだけ多くのホストにテンプレートを適用できます。

テンプレートがホストにリンクされると、テンプレートのすべてのエンティティ（アイテム、トリガー、グラフなど）がホストに追加されます。テンプレートは各ホストに直接割り当てられます（ホストグループには割り当てられません）。

テンプレートは、特定のサービスやアプリケーション（Apache、MySQL、PostgreSQL、Postfix など）ごとにエンティティをグループ化し、それらのサービスを実行しているホストに適用するためによく使用されます。

テンプレートを使用するもう 1 つの利点は、すべてのホストに対して何かを変更する必要がある場合です。テンプレートレベルで 1 回変更すれば、その変更はリンクされているすべてのホストに反映されます。

テンプレートはテンプレートグループで整理されます。

テンプレートの作成と設定に進みます。

9 すぐに使えるテンプレート

概要

Zabbix は、監視対象のセットアップを簡素化し、迅速化するために、事前に構成されたテンプレートのセットを提供しています。

すべての標準テンプレートは、データ収集 > テンプレートで利用できます。

Zabbix をアップグレードする際、既存のテンプレートはカスタム修正の上書きを避けるため、自動的には更新されません。既存のテンプレートをアップグレードしたり、新しいテンプレートを追加したりするには、テンプレートのアップグレードを参照してください。

特定のテンプレートタイプや運用要件については、サイドバーからアクセスしてください。

関連情報:

- テンプレートのリンク
- テンプレートの既知の問題

テンプレートのアップグレード

Zabbix のアップグレード後に既存のテンプレートをアップグレードしたり、新しいテンプレートを追加するには、以下の手順を実行します。

1. Zabbix Git リポジトリにアクセスします。
2. ブランチセレクタ（*Source* ラベルの下）で、Zabbix のバージョンに一致するブランチを選択します（例：最新の開発バージョンの場合は *master*）。
3. 追加またはアップグレードしたいテンプレートを開きます（例：MySQL by Zabbix agent）。
4. テンプレートファイル（例：`template_db_mysql_agent.yaml`）の内容をコピーし、ローカルファイルに保存します。
5. Zabbix フロントエンドで、データ収集 > テンプレートに移動し、ローカルファイルをインポートします。

ホストウィザードの互換性

ホストウィザードと互換性のある複数のテンプレートをアップグレードするには、以下のテンプレートバンドルをインポートできます:

- cloud_aws_azure.yaml
- cloud_gcp_oracle.yaml
- database.yaml
- network_os.yaml
- software.yaml

互換性のあるテンプレートの全リストについては、templates_in_bundles.mdを参照してください。

カスタムテンプレートをホストウィザードと互換性を持たせるには:

1. Zabbix フロントエンドで、データ収集 > テンプレートに移動し、カスタムテンプレートをエクスポートします。
2. エクスポートしたファイルを編集します:
    - `wizard_ready` テンプレート要素を追加して、ホストウィザードとの互換性を示します。
    - 必要に応じて、`readme` または `config` テンプレート要素を追加します。
3. Zabbix フロントエンドで、データ収集 > テンプレートに移動し、更新したテンプレートファイルをインポートします。

1 Zabbix エージェントテンプレートの動作

Zabbix エージェントでメトリクスを収集するテンプレートが正しく動作するための手順:

1. 監視対象ホストに Zabbix エージェントがインストールされていることを確認します。アクティブチェックの場合は、エージェントの設定ファイルの「ServerActive」パラメータに Zabbix サーバー/プロキシのアドレスが追加されていることも確認してください。

2. テンプレートを対象ホストにリンクします（Zabbix インストール環境にテンプレートがない場合は、事前にインポートが必要な場合があります）。

3. 必要に応じて、テンプレートマクロの値を調整します。

4. 監視対象のインスタンスが Zabbix とのデータ共有を許可するように設定します。

テンプレートの詳細な説明（マクロ、アイテム、トリガーの全リストを含む）は、テンプレートの README ファイル（テンプレート名をクリックしてアクセス）で確認できます。

利用可能なテンプレートは以下の通りです:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active

- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

2 Zabbix agent 2 テンプレートの運用

Zabbix agent 2でメトリクスを収集するテンプレートが正しく動作するための手順:

1. ホストに agent 2 がインストールされており、インストールされているバージョンに必要なプラグインが含まれていることを確認します。場合によっては、最初に agent 2 をアップグレードする必要があるかもしれません。

2. テンプレートを対象ホストにリンクします (テンプレートが Zabbix インストール環境にない場合は、最初にテンプレートをインポートする必要があるかもしれません)。

3. 必要に応じて、テンプレートマクロの値を調整します。ユーザーマクロを使用して設定パラメータを上書きできることに注意してください。

4. 監視対象のインスタンスが Zabbix とのデータ共有を許可するように設定します。

Attention:
Zabbix agent 2 のテンプレートはプラグインと連携して動作します。基本的な設定はユーザーマクロの調整だけで行えますが、より詳細なカスタマイズはプラグイン自体の設定によって実現できます。
たとえば、プラグインが名前付きセッションをサポートしている場合、設定ファイルで各エンティティごとに独自の URI、ユーザー名、パスワードを指定した名前付きセッションを設定することで、同種の複数のエンティティ (例: MySQL1 と MySQL2) を監視できます。

テンプレートの詳細な説明 (マクロ、アイテム、トリガーの全リストを含む) は、テンプレートの README ファイル (テンプレート名をクリックしてアクセス) で確認できます。

利用可能なテンプレートは以下の通りです:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

3 HTTP テンプレートの操作

HTTP エージェントでメトリクスを収集するテンプレートが正しく動作するための手順:

1. 監視対象に対応するホストを Zabbix に作成します。監視対象の IP アドレス/DNS 名をメインのホストインターフェースとして指定します。これは、テンプレートアイテムで {HOST.CONN} マクロが正しく解決されるために必要です。

2. ステップ 1 で作成したホストにテンプレートをリンクします（テンプレートが Zabbix に存在しない場合は、事前にインポートが必要な場合があります）。

3. 必要に応じて、テンプレートレベルのマクロの値を調整します。

4. 監視対象のインスタンスが Zabbix とのデータ共有を許可するように設定します。

テンプレートの詳細な説明（マクロ、アイテム、トリガーの全リストを含む）は、テンプレート名をクリックしてアクセスできるテンプレートの README ファイルに記載されています。

利用可能なテンプレートは以下の通りです:

- Acronis Cyber Protect Cloud by HTTP
- Apache by HTTP
- Asterisk by HTTP
- AWS by HTTP
- AWS Cost Explorer by HTTP
- AWS EC2 by HTTP
- AWS ECS Cluster by HTTP
- AWS ECS Serverless Cluster by HTTP
- AWS ELB Application Load Balancer by HTTP
- AWS ELB Network Load Balancer by HTTP
- AWS Lambda by HTTP
- AWS RDS instance by HTTP
- AWS S3 bucket by HTTP
- Azure by HTTP
- Cisco Meraki organization by HTTP
- Cisco SD-WAN by HTTP
- Cisco Secure Firewall Threat Defense by HTTP
- ClickHouse by HTTP
- Cloudflare by HTTP
- CockroachDB by HTTP
- Control-M enterprise manager by HTTP
- Control-M server by HTTP
- DELL PowerEdge R720 by HTTP
- DELL PowerEdge R740 by HTTP
- DELL PowerEdge R820 by HTTP
- DELL PowerEdge R840 by HTTP
- Elasticsearch Cluster by HTTP
- Envoy Proxy by HTTP
- Etcd by HTTP
- FortiGate by HTTP
- GitHub repository by HTTP
- GitLab by HTTP
- Google Cloud Platform (GCP) by HTTP
- Hadoop by HTTP
- HAProxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP
- HashiCorp Nomad by HTTP
- HashiCorp Vault by HTTP
- Hikvision camera by HTTP
- HPE iLO by HTTP
- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP
- HPE Synergy by HTTP
- InfluxDB by HTTP
- Jenkins by HTTP
- Kubernetes API server by HTTP
- Kubernetes cluster state by HTTP
- Kubernetes Controller manager by HTTP
- Kubernetes kubelet by HTTP
- Kubernetes nodes by HTTP
- Kubernetes Scheduler by HTTP
- MantisBT by HTTP
- Microsoft 365 reports by HTTP
- Microsoft SharePoint by HTTP

- NetApp AFF A700 by HTTP
- Nextcloud by HTTP
- NGINX by HTTP
- NGINX Plus by HTTP
- Nutanix Prism Element by HTTP
- OpenAI Platform by HTTP
- OpenStack by HTTP
- OpenWeatherMap by HTTP
- Oracle Cloud by HTTP
- Palo Alto PA-440 by HTTP
- PHP-FPM by HTTP
- Proxmox VE by HTTP
- Pure Storage FlashArray v1 and v2 by HTTP
- RabbitMQ cluster by HTTP
- Ribbon SBC Edge by HTTP
- Ribbon SBC SWe CE by HTTP
- Ribbon SBC SWe Core by HTTP
- TiDB by HTTP
- TiDB PD by HTTP
- TiDB TiKV by HTTP
- Travis CI by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP
- VeloCloud SD-WAN by HTTP
- VeloCloud SD-WAN Edge by HTTP
- YugabyteDB by HTTP
- ZooKeeper by HTTP

4 IPMI テンプレートの操作

IPMI テンプレートには特別なセットアップは必要ありません。監視を開始するには、テンプレートを対象ホストにリンクします (テンプレートが Zabbix インストールに存在しない場合は、最初にインポートする必要があります)。

テンプレートの詳細な説明 (マクロ、アイテム、トリガーの完全なリストを含む) は、テンプレートの README ファイル (テンプレート名をクリックしてアクセス) で確認できます。

利用可能なテンプレート:

- Chassis by IPMI

5 JMX テンプレートの運用

JMXによってメトリクスを収集するテンプレートの正しい動作を確保する手順:

1. Zabbix Java ゲートウェイがインストールされ、正しく設定されていることを確認します。
2. テンプレートを対象ホストにリンクします。ホストには JMX インターフェースが設定されている必要があります。テンプレートが Zabbix インストール環境に存在しない場合は、最初にテンプレートをインポートする必要があります。
3. 必要に応じてテンプレートマクロの値を調整します。
4. 監視対象のインスタンスが Zabbix とのデータ共有を許可するように設定します。

テンプレートの詳細な説明 (マクロ、アイテム、トリガーの全リストを含む) は、テンプレート名をクリックしてアクセスできるテンプレートの README ファイルに記載されています。

利用可能なテンプレートは以下の通りです:

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX
- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX

- WildFly Domain by JMX
- WildFly Server by JMX

6 ODBC テンプレートの操作

ODBC 監視を介してメトリクスを収集するテンプレートが正しく動作するようにする手順:

1. 必要な ODBC ドライバーが Zabbix サーバーまたはプロキシにインストールされていることを確認します。
2. テンプレートを対象ホストにリンクします（テンプレートが Zabbix インストールに存在しない場合は、最初にインポートが必要な場合があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスが Zabbix とのデータ共有を許可するように設定します。

テンプレートの詳細な説明（マクロ、アイテム、トリガーの全リストを含む）は、テンプレート名をクリックしてアクセスできるテンプレートの README ファイルで確認できます。

以下のテンプレートが利用可能です:

- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- PostgreSQL by ODBC

7 ネットワークデバイス用の標準テンプレート

概要

スイッチやルーターなどのネットワーク機器の監視を提供するために、いわゆる 2 つのモデルを作成しました。ネットワーク機器自体 (基本的にはそのシャーシ) とネットワークインターフェース用です。

多くのネットワーク機器ファミリー用のテンプレートが提供されています。すべてのテンプレートは、(デバイスからこれらのアイテムを取得できる場合は) 以下をカバーしています。

- シャーシ障害監視 (電源、ファン、温度、全体のステータス)
- シャーシパフォーマンス監視 (CPU およびメモリアイテム)
- シャーシインベントリ収集 (シリアル番号、モデル名、ファームウェアバージョン)
- IF-MIB および EtherLike-MIB によるネットワークインターフェース監視 (インターフェースステータス、インターフェーストラフィック負荷、イーサネットのデュプレックスステータス)

テンプレートが Zabbix インストールに存在しない場合は、最初にテンプレートをインポートする必要があります。

新しい標準テンプレートをインポートする場合は、`@Network interfaces for discovery` グローバル正規表現も次のように更新することをお勧めします。

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

これにより、ほとんどのシステムでループバックやヌルインターフェースを除外できます。

デバイス

テンプレートが利用可能なデバイスファミリーの一覧:

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Aruba CX Switch Series	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certified
Brocade FC SNMP	Brocade	Brocade FC switches	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Fault
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certified
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Fault
Check Point Next Generation Firewall by SNMP	Check Point	Next Genera-tion Firewall	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certified
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certified
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	Certified
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T and 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX switches	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certified
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certified
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500V5/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infini-band devices	Infiniband 12300		ICS-CHASSIS-MIB	Fault Inventory
Juniper SNMP	Juniper	MX,SRX,EX models	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certified
Juniper MX NETCONF	Juniper	MX models	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Not applicable (uses NETCONF over SSH)*	Certified
Juniper MX SNMP	Juniper	MX models	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certified
Mellanox SNMP	Mellanox	Mellanox Infini-band devices	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certified
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Routers (CCR series)	MikroTik CCR1009-7G-1C-1S+、MikroTik CCR1009-7G-1C-1S+PC、MikroTik CCR1009-7G-1C-PC、MikroTik CCR1016-12G、MikroTik CCR1016-12S-1S+、MikroTik CCR1036-12G-4S-EM、MikroTik CCR1036-12G-4S、MikroTik CCR1036-8G-2S+、MikroTik CCR1036-8G-2S+EM、MikroTik CCR1072-1G-8S+、MikroTik CCR2004-16G-2S+、MikroTik CCR2004-1G-12S+2XS 用の個別専用テンプレート	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (CRS series)	MikroTik CRS106-1C-5S、MikroTik CRS109-8G-1S-2HnD-IN、MikroTik CRS112-8G-4S-IN、MikroTik CRS112-8P-4S-IN、MikroTik CRS125-24G-1S-2HnD-IN、MikroTik CRS212-1G-10S-1S+IN、MikroTik CRS305-1G-4S+IN、MikroTik CRS309-1G-8S+IN、MikroTik CRS312-4C+8XG-RM、MikroTik CRS317-1G-16S+RM、MikroTik CRS326-24G-2S+IN、MikroTik CRS326-24G-2S+RM、MikroTik CRS326-24S+2Q+RM、MikroTik CRS328-24P-4S+RM、MikroTik CRS328-4C-20S-4S+RM、MikroTik CRS354-48G-4S+2Q+RM、MikroTik CRS354-48P-4S+2Q+RM 用の個別専用テンプレート	RouterOS/SwitchOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (CSS series)	MikroTik CSS326-24G-2S+RM、MikroTik CSS610-8G-2S+IN 用の個別専用テンプレート	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	MikroTik hEX、MikroTik hEX lite、MikroTik hEX PoE、MikroTik hEX PoE lite、MikroTik hEX S 用の個別専用テンプレート	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	MikroTik netPower 15FR、MikroTik netPower 16P SNMP、MikroTik netPower Lite 7R 用の個別専用テンプレート	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	MikroTik PowerBox、MikroTik PowerBox Pro 用の個別専用テンプレート	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik RB<device model> SNMP	MikroTik	MikroTik RB series routers	MikroTik RB1100AHx4、MikroTik RB1100AHx4 Dude Edition、MikroTik RB2011iL-IN、MikroTik RB2011iL-RM、MikroTik RB2011iLS-IN、MikroTik RB2011UiAS-IN、MikroTik RB2011UiAS-RM、MikroTik RB260GS、MikroTik RB3011UiAS-RM、MikroTik RB4011iGS+RM、MikroTik RB5009UG+S+IN 用の個別専用テンプレート	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik SNMP	MikroTik	MikroTik RouterOS devices	MikroTik CCR1016-12G、MikroTik RB2011UAS-2HnD、MikroTik 912UAG-5HPnD、MikroTik 941-2nD、MikroTik 951G-2HnD、MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory
QTech QSW SNMP	QTech	Qtech devices	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventory

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用 MIB	タグ
Stormshield SNS by SNMP	Stormshield	Stormshield Network Security (SNS) firewalls	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS wireless devices	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Performance
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Performance Inventory

テンプレート設計

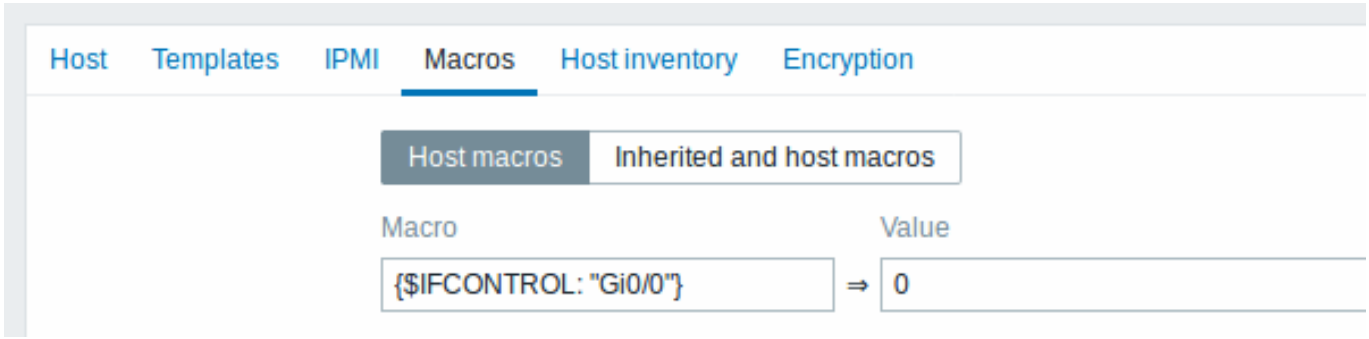
テンプレートは以下の点を考慮して設計されています。

- トリガーをユーザーが調整できるよう、できるだけユーザーマクロを使用しています。
- サポートされていないアイテムの数を最小限に抑えるため、できるだけローレベルディスカバリを使用しています。
- すべてのテンプレートは Template ICMP Ping に依存しているため、すべてのデバイスが ICMP でもチェックされます。
- アイテムでは MIB を使用せず、SNMP OID をアイテムやローレベルディスカバリで使用しています。そのため、テンプレートを動作させるために Zabbix に MIB をロードする必要はありません。
- ディスカバリ時にループバックインターフェースや ifAdminStatus = down(2) のインターフェースをフィルタリングします。
- 可能な場合は IF-MIB::ifXTable の 64bit カウンタを使用します。サポートされていない場合は、デフォルトの 32bit カウンタを使用します。

ディスカバリされたすべてのネットワークインターフェースには、その運用状態（リンク）を監視するトリガーがあります。例えば：

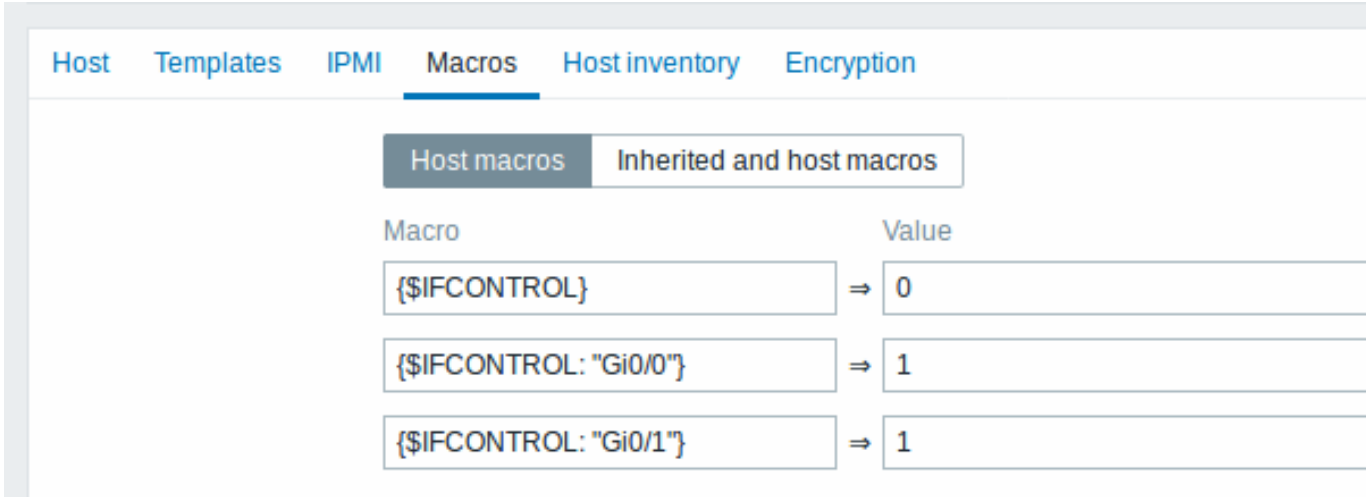
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 特定のインターフェースでこの条件を監視したくない場合は、値を 0 としたコンテキスト付きユーザーマクロを作成してください。例：



ここで、Gi0/0 は {#IFNAME} です。これにより、この特定のインターフェースではトリガーが使用されなくなります。

- すべてのトリガーが発火しないようにデフォルトの動作を変更し、アップリンクなど限られたインターフェースだけでこのトリガーを有効にすることもできます。



タグ

- Performance - デバイスファミリーの MIB は、CPU とメモリ item を監視する方法を提供します。
- Fault - デバイスファミリーの MIB は、少なくとも 1 つの温度センサを監視する方法を提供します。
- Inventory - デバイスファミリーの MIB は、少なくともデバイスのシリアル番号とモデル名を収集する方法を提供します。
- Certified - 上記 3 つの主要カテゴリを全てカバーします。

8 VMware テンプレートの操作

概要

Zabbix には、VMware vCenter および ESXi ハイパーバイザー用に設計された仮想マシン監視用のすぐに使えるテンプレートが用意されています。利用可能なテンプレートは、2 つの別々のテンプレートセットに含まれています。

- VMware - 対応するマクロに UUID データを使用
- VMware FQDN - 対応するマクロに FQDN データを使用

VMware vCenter または ESXi ハイパーバイザーの監視用に設計されたテンプレートには、仮想インストールの監視用にさまざまな組み込みチェックとともに、事前構成された低レベルディスカバリールールが含まれています。

Note:
VMware FQDN テンプレートが正しく機能するためには、監視対象の各仮想マシンが FQDN のルールに従った一意の OS 名を持っている必要があります。さらに、VMware Tools/Open Virtual Machine tools をすべてのマシンにインストールする必要があります。これらの前提条件が満たされている場合は、*VMware FQDN* テンプレートの使用を推奨します。*VMware FQDN* テンプレートは、カスタムインターフェースでホストを作成する機能が導入された Zabbix 5.2 以降で利用可能です。

 従来の *VMware* テンプレートも利用可能であり、FQDN の要件が満たされていない場合に使用できます。ただし、*VMware* テンプレートには既知の問題があります。検出された仮想マシン用のホストは、vCenter に保存されている名前で作成されます（例：”VM1”、”VM2” など）。これらのホストに Zabbix agent がインストールされていて、アクティブな Zabbix agent の自動登録が有効になっている場合、自動登録プロセスは起動時に登録されたホスト名（例：”vm1.example.com”、”vm2.example.com” など）を読み取ります。このため、既存の仮想マシンに対して新しいホストが作成される可能性があり（名前の一致が見つからないため）、異なる名前の重複したホストが作成されることになります。

ご注意ください:

- テンプレート *VMware Hypervisor* は、ホストに手動でリンクすることも、ディスカバリーで使用することもできます。
- テンプレート *VMware Guest* は、ホストに手動でリンクしてはいけません。ディスカバリーでのみ使用できます。

ホストマクロの設定

VMware のシンプルチェックを使用するには、ホストに以下のユーザーマクロを定義する必要があります。

- {$VMWARE.URL} - VMware サービス（vCenter または ESXi ハイパーバイザー）SDK の URL（https://servername/sdk）
- {$VMWARE.USERNAME} - VMware サービスのユーザー名
- {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード

10 イベント時の通知

概要

いくつかのアイテムとトリガーを構成し、トリガーの状態変更の結果としていくつかのイベントが発生していると仮定して、いくつかのアクションを検討します。

そもそも、トリガーやイベントのリストを常に眺めたくはありません。何か重大な、障害などが発生した場合に通知を受け取る方がはるかに良いでしょう。また、障害が発生した場合には、関係者全員に周知したいと考えております。

そのため、通知の送信は Zabbix が提供する主要なアクションの 1 つです。特定のイベントが発生したときに、誰に、いつ、通知するかを定義できます。

Zabbix から通知を送受信できるようにするには、次のことを行う必要があります。

- メディアの定義
- 定義されたメディアにメッセージを送信するアクションの設定

アクションは条件と操作で構成されます。基本的に条件が満たされた場合に操作が実行されます。2 つの主要な操作は、メッセージの送信 (通知) とリモートコマンドの実行です。

ディスカバリおよび自動登録によって作成されたイベントについては、いくつかの追加操作を使用できます。追加操作はホストの追加または削除、テンプレートのリンクなどです。

1 メディアタイプ

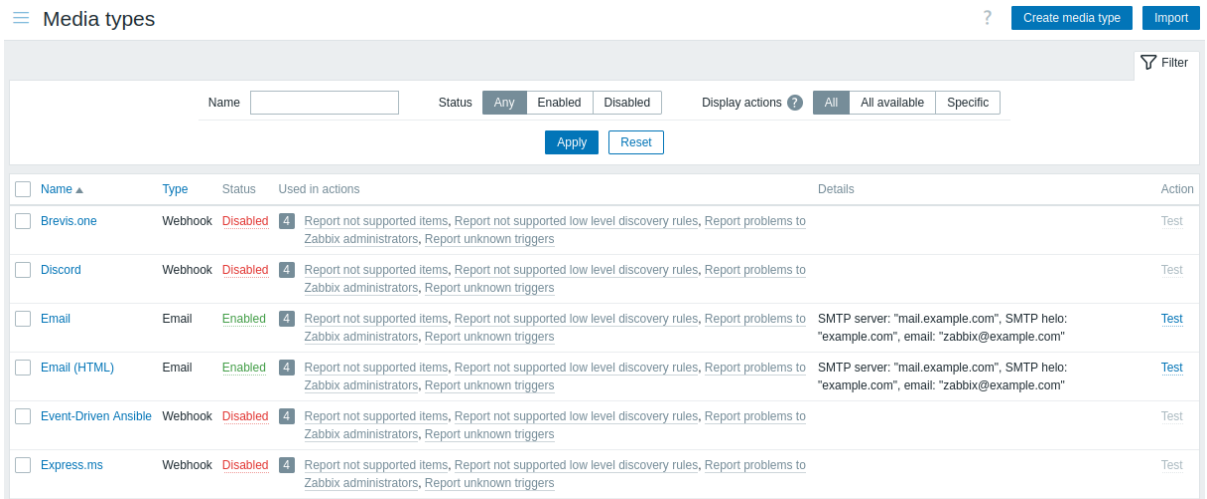
概要

メディアタイプは、Zabbix から通知やアラートを送信するための配信チャネルです。

メディアタイプは、以下の配信方法をサポートしています。

- メール
- SMS
- カスタムスクリプト
- Webhook

メディアタイプは アラート > メディアタイプで管理します。いくつかのメディアタイプは、デフォルトのデータセットにあらかじめ定義されています。それらを動作させるには、パラメータを微調整するだけです。



メディアタイプがアラートプロセスの中でどのような役割を果たすかを理解するために、Zabbix からエンドユーザーに通知を配信するための 3 つの要件を見てみましょう。

1. 通知を送信するアクションのオペレーションが定義されていること
2. 動作するメディアタイプが定義されていること（例：SMTP を使用してアラートを送信するメールなど）
3. ユーザーメディアで、ユーザーレベルの配信先情報（メールアドレス、電話番号など）が定義されていること

メディアタイプのテスト

設定したメディアタイプが動作するかどうかをテストするには、メディアタイプ一覧でテストをクリックします。

テストリクエストは Zabbix サーバーに送信されます。Zabbix サーバーは指定されたメディアタイプを使用してアラートの送信を試み、その結果をフロントエンドに返します。フロントエンドはサーバーが結果を返すのを待ちます。メディアタイプのテストはデフォルトで 65 秒のタイムアウトがあります（管理 > 一般 > タイムアウトで設定可能）。

以下のメディアタイプのテスト詳細も参照してください:

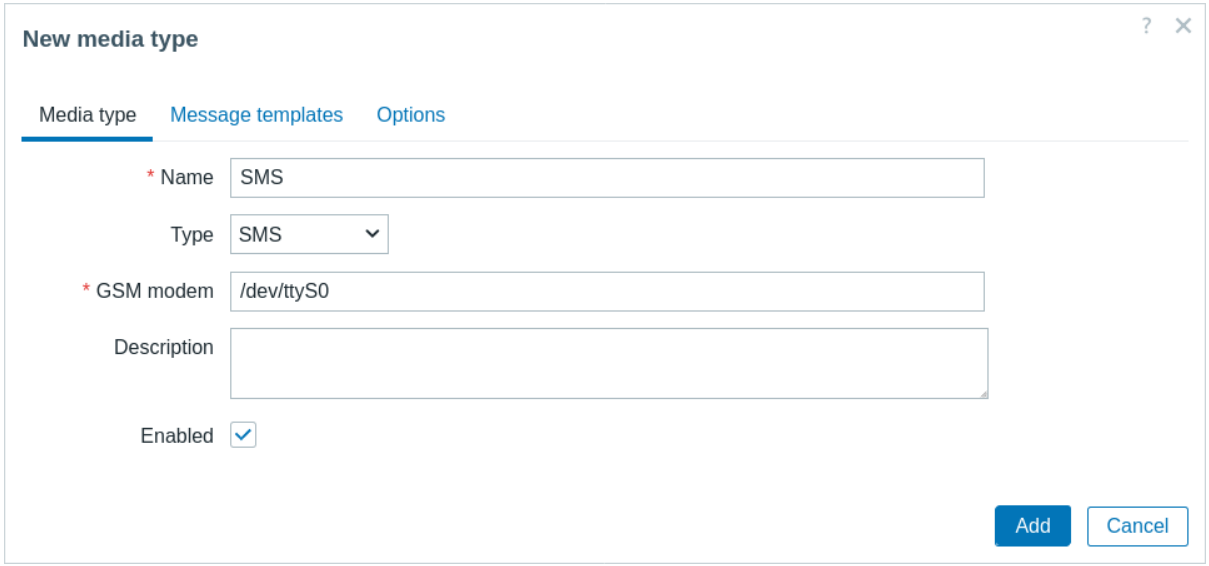
- メール
- Webhook
- スクリプト

設定

Zabbix フロントエンドでメディアタイプを作成するには:

- アラート > メディアタイプに移動します
- メディアタイプの作成をクリックします
- フォームにメディアタイプのパラメータを入力します

いくつかのパラメータは、すべての配信方法で共通です。

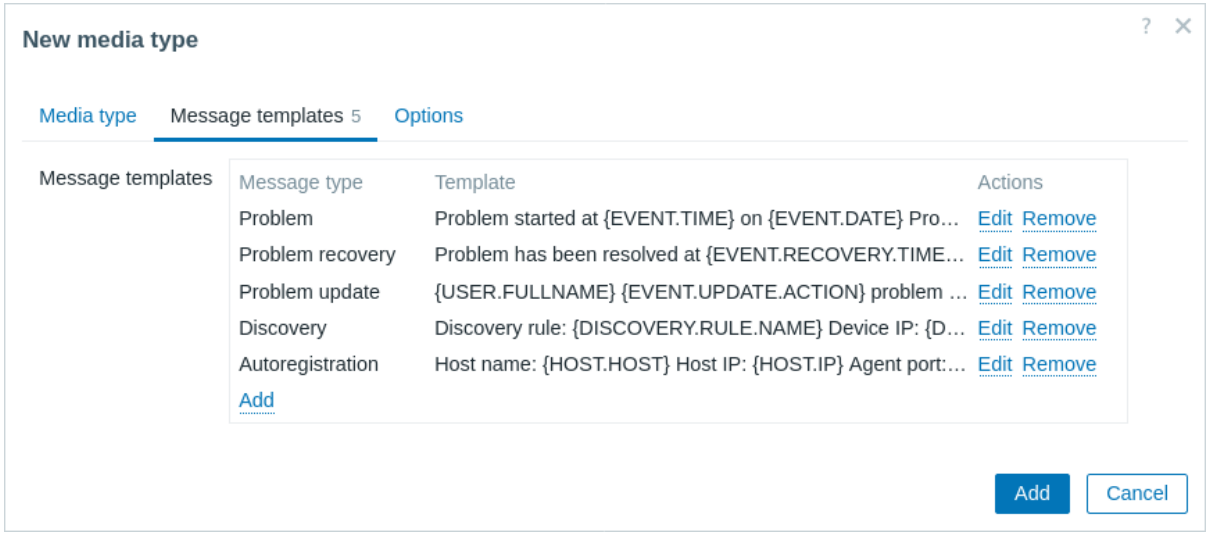


パラメータ	説明
名前	メディアタイプの名前です。
タイプ	メディアタイプの配信方法を選択します。
説明	メディアタイプの説明を入力します。
有効	メディアタイプを有効にするにはチェックボックスをオンにします。

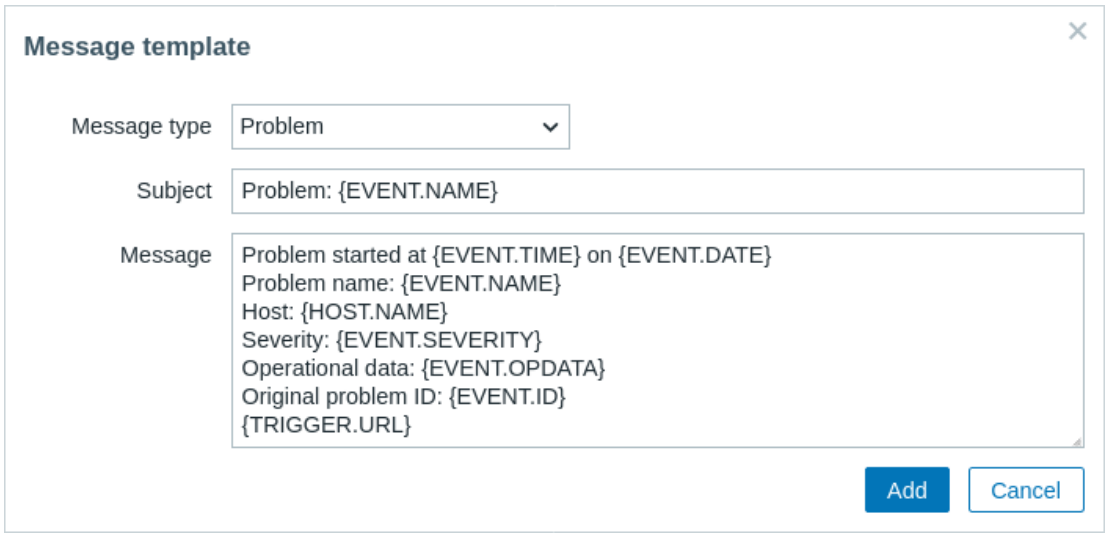
配信方法ごとのパラメータについては、メール、SMS、カスタムアラートスクリプト、またはWebhookのページを参照してください。

メッセージテンプレート

メッセージテンプレートタブには、イベントタイプごとのデフォルトメッセージ（障害、障害復旧、ディスカバリなど）が含まれています。



追加をクリックしてデフォルトメッセージを定義します（または編集をクリックして既存のメッセージを更新します）：



パラメータ	説明
メッセージタイプ	デフォルトメッセージを使用するイベントのタイプ。 各イベントタイプごとに 1 つのデフォルトメッセージのみ定義できます。
件名	デフォルトメッセージの件名。件名にはマクロを含めることができます。255 文字までに制限されています。 SMS メディアタイプでは件名は利用できません。
メッセージ	デフォルトメッセージ。データベースの種類によって一定の文字数に制限されています（詳細はメッセージの送信を参照してください）。 メッセージにはサポートされているマクロを含めることができます。 障害および障害更新メッセージでは、式マクロがサポートされています（例：`{?avg(/host/key,1h)}`）。

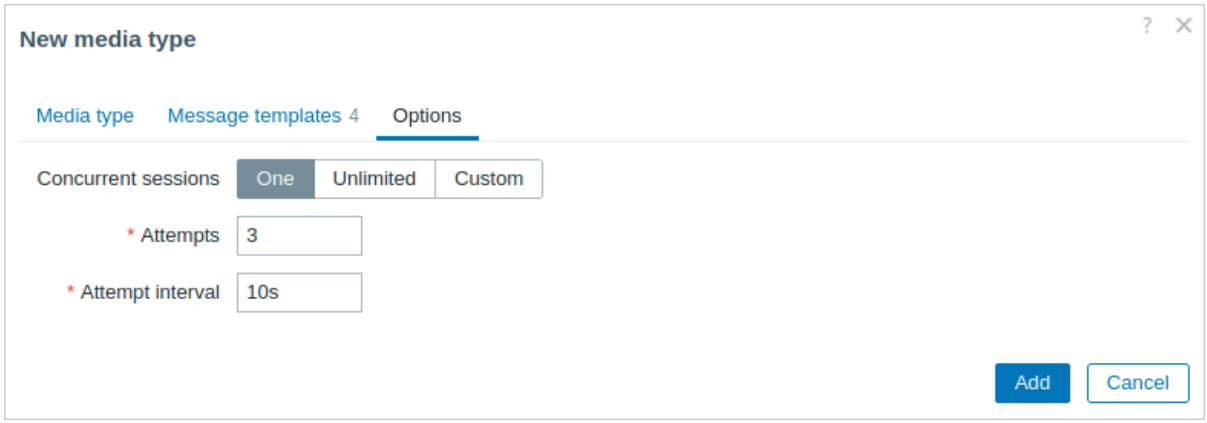
デフォルトメッセージは、アクションのオペレーションでカスタムメッセージが定義されている場合、それによって上書きされることに注意してください。

Warning:
メッセージテンプレートの定義は、デフォルトメッセージを通知に使用しない Webhook やカスタムアラートスクリプトを含む、すべての配信方法で必須です。たとえば、「Pushover webhook にメッセージを送信」アクションは、Pushover webhook の障害メッセージが定義されていない場合、障害通知の送信に失敗します。

オプション

オプションタブには、アラート処理の設定があります。同じオプションセットが各メディアタイプごとに設定可能です。

すべてのメディアタイプは並列で処理されます。メディアタイプごとに同時セッション数の最大値は設定できますが、サーバー上のアラート処理プロセスの総数は、StartAlertersパラメータでのみ制限できます。1 つのトリガーによって生成されたアラートは順次処理されます。したがって、複数の通知が同時に処理されるのは、複数のトリガーによって生成された場合のみです。



パラメータ	説明
同時セッション数	メディアタイプごとの並列アラートセッション数を選択します: **1** つ - 1 セッション 無制限 - 無制限のセッション数 カスタム - カスタムのセッション数を選択 無制限/高い値は、より多くの並列セッションと通知送信能力の増加を意味します。無制限/高い値は、多数の通知を同時に送信する必要がある大規模な環境で使用してください。 同時セッション数より多くの通知を送信する必要がある場合、残りの通知はキューに入れられます。失われることはありません。
試行回数	通知送信の試行回数。最大 100 回まで指定可能。デフォルト値は「3」です。「1」を指定した場合、Zabbix は 1 回だけ通知を送信し、失敗しても再試行しません。
試行間隔	通知送信に失敗した場合の再送信間隔（秒単位、0-3600）。「0」を指定した場合、Zabbix は即座に再試行します。 時間のサフィックス（例: 5s, 3m, 1h）がサポートされています。

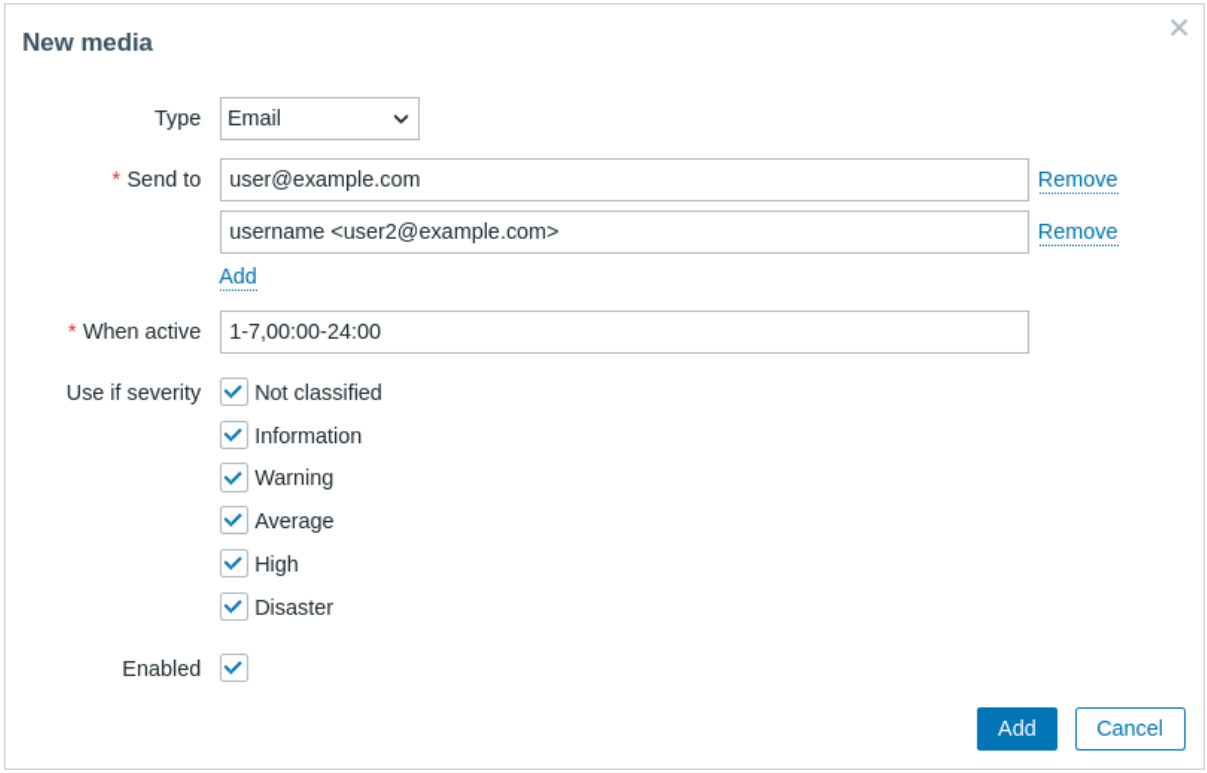
ユーザーメディア

メディアタイプが通知をどのように送信するかを定義するのに対し、ユーザーメディアは通知をどこに送信するかを定義します。

ユーザーメディア（例：メールアドレス、Webhook のユーザー ID など）は、配信方法に関係なくユーザープロファイルで定義する必要があります。Webhook *X* を使用して *Admin* ユーザーにメッセージを送信するアクションは、Webhook *X* の配信先が Admin ユーザープロファイルで定義されていない場合、配信に失敗します。

ユーザーメディアを定義するには：

- ユーザー > ユーザーでユーザーのプロパティフォームを開く（または自分のユーザープロファイルの ユーザー設定 > 通知へ）
- メディアタブで 追加をクリック



パラメータ	説明
タイプ	ドロップダウンリストには有効なメディアタイプ名が表示されます。 無効なメディアタイプのメディアを編集する場合、タイプは赤色で表示されます。
送信先	メッセージを送信する連絡先情報を入力します。 メールメディアタイプの場合、アドレス欄の下にある Add ボタンをクリックして複数のアドレスを追加できます。この場合、すべてのアドレスに通知が送信されます。アドレスの例については、メールメディアタイプのメールパラメータの説明を参照してください。
有効期間	メッセージが送信される時間を制限できます。たとえば、平日のみ（1-5,09:00-18:00）に設定できます。この制限はユーザーのタイムゾーンに基づいていることに注意してください。ユーザーのタイムゾーンが変更され、システムのタイムゾーンと異なる場合は、重要なメッセージを見逃さないようにこの制限を調整する必要があります。 書式の説明は期間の指定ページを参照してください。 ユーザーマクロがサポートされています。
深刻度で使用	通知を受信したいトリガーの深刻度のチェックボックスを選択します。 注意：デフォルトの深刻度（「未分類」）は、トリガー以外のイベントの通知を受信したい場合、必ずチェックする必要があります。 保存後、選択したトリガーの深刻度は対応する深刻度の色で表示され、未選択のものはグレー表示されます。
ステータス	ユーザーメディアのステータス。 有効 - 使用中。 無効 - 使用されていません。

1 E-mail

概要

メッセージの配信チャネルとしてメールを設定するには、メールをメディアタイプとして設定し、特定のアドレスをユーザーに割り当てる必要があります。

> **Note:**
> 単一イベントに対する複数の通知は、同じメールスレッドにまとめられます。

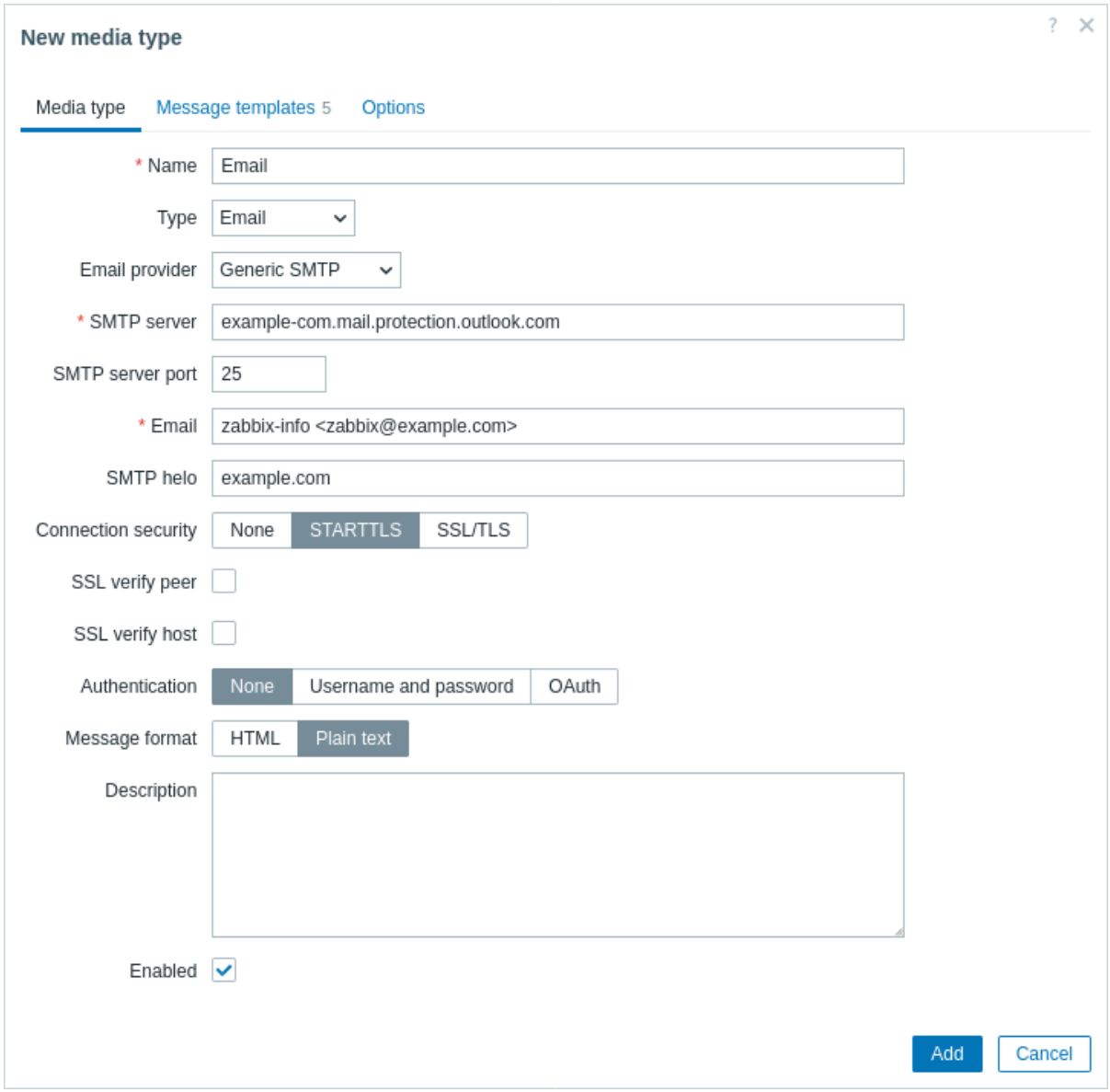
設定

メールをメディアタイプとして設定するには:

1. アラート > メディアタイプに移動します。

2. メディアタイプの作成をクリックするか、定義済みメディアタイプのリストから *Email* をクリックします。

メディアタイプタブには、一般的なメディアタイプ属性が含まれています:



すべての必須入力フィールドには赤いアスタリスクが付いています。

Warning:
メールメディアタイプの設定をエクスポートする際、パスワードは平文でエクスポートされます。

メールメディアタイプに固有のパラメータは以下の通りです:

パラメータ	説明
メールプロバイダー	メールプロバイダーを選択します: *Generic SMTP*、*Gmail*、*Gmail relay*、*Office365*、または *Office365 relay*。 Gmail/Office365 関連のオプションを選択した場合は、送信者のメールアドレスとパスワードのみを指定する必要があります。*SMTP* サーバー、*SMTP* サーバーポート、*SMTP helo*、接続セキュリティなどのオプションは Zabbix によって自動的に入力されます。詳細は自動化された Gmail/Office365 メディアタイプ を参照してください。
SMTP サーバー	送信メッセージを処理する SMTP サーバーを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* を選択した場合に利用できます。
SMTP サーバーポート	送信メッセージを処理する SMTP サーバーポートを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* を選択した場合に利用できます。

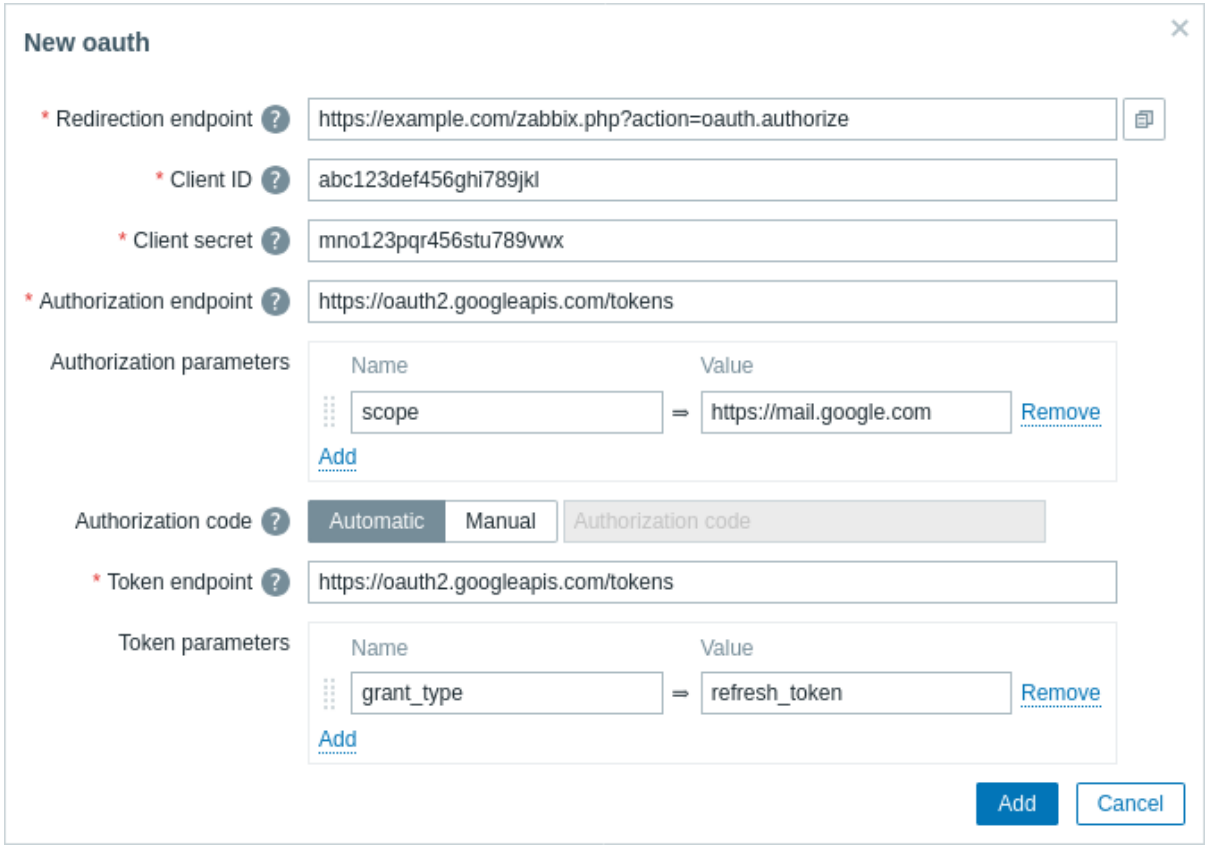
パラメータ	説明
メール	ここに入力したアドレスが、送信されるメッセージの **From** アドレスとして使用されます。 実際のメールアドレスとともに送信者表示名（上記スクリーンショットの *Zabbix_info <zabbix@company.com>* の”Zabbix_info” など）を追加することができます。 Zabbix メールの表示名には RFC 5322 で許可されているものと比較していくつか制限があります。 例で説明します: 有効な例: *zabbix@company.com*（メールアドレスのみ、山括弧は不要） *Zabbix_info <zabbix@company.com>*（表示名と山括弧内のメールアドレス） *∑Ω-monitoring <zabbix@company.com>*（表示名に UTF-8 文字） 無効な例: *Zabbix HQ zabbix@company.com*（表示名があるがメールアドレスが山括弧で囲まれていない） *”Zabbix\@\<H(comment)Q\>” <zabbix@company.com>*（RFC 5322 では有効だが、引用ペアやコメントは Zabbix メールではサポートされていない）
SMTP helo	正しい SMTP helo 値（通常はドメイン名）を設定します。 空の場合、メールのドメイン名（メールフィールドの @ 以降）が送信されます。ドメイン名の取得ができない場合は、デバッグレベルの警告が記録され、HELO コマンドのドメインとしてサーバーホスト名が送信されます。 このフィールドは、メールプロバイダーとして *Generic SMTP* を選択した場合に利用できます。
接続セキュリティ	接続セキュリティのレベルを選択します: なし - CURLOPT_USE_SSL オプションを使用しない **STARTTLS** - CURLOPT_USE_SSL オプションを CURLUSESSL_ALL 値で使用 **SSL/TLS** - CURLOPT_USE_SSL の使用は任意
SSL ピア検証	チェックボックスをマークして、SMTP サーバーの SSL 証明書を検証します。 証明書検証のために、サーバー設定ディレクティブ”SSLCALocation” の値が CURLOPT_CAPATH に設定されます。 これにより cURL オプション CURLOPT_SSL_VERIFYPEER が設定されます。
SSL ホスト検証	チェックボックスをマークして、SMTP サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致するかどうかを検証します。 これにより cURL オプション CURLOPT_SSL_VERIFYHOST が設定されます。
認証	認証レベルを選択します: なし - cURL オプションは設定されません ユーザー名とパスワード - ”AUTH=*” を意味し、認証メカニズムの選択は cURL に任されます **OAuth** - OAuth 認証 OAuth 認証は *Office365 relay* メールプロバイダーではサポートされていません。
ユーザー名	認証に使用するユーザー名。 CURLOPT_USERNAME の値として設定されます。 ユーザーマクロ がサポートされています。
パスワード	認証に使用するパスワード。 CURLOPT_PASSWORD の値として設定されます。 ユーザーマクロ がサポートされています。 ここに入力した値は、メディアタイプ設定をエクスポートする際、そのままエクスポートされます。 エクスポートファイルで機密情報が漏洩しないよう、リテラルなパスワードの代わりにユーザーマクロ（例: `{$EMAIL_SMTP_PASSWORD}`）を使用してください。マクロ参照がエクスポートされ、秘密値は移行先システムで再設定する必要があります。
OAuth トークン	設定をクリックして、新しいウィンドウでOAuth トークン を取得するためのパラメータを設定します。 このフィールドは、認証フィールドで”OAuth” を選択した場合のみ利用できます。
メッセージ形式	メッセージ形式を選択します: **HTML** - HTML として送信 プレーンテキスト - プレーンテキストとして送信

Attention:
SMTP 認証を有効にするには、Zabbix サーバーを `--with-libcurl` コンパイル オプション付きでビルドする必要があります。これは libcurl を必要とします（バージョンの詳細はサーバー要件を参照）。

デフォルトメッセージやアラート処理オプションの設定方法については、共通メディアタイプパラメータ も参照してください。

OAuth トークン

OAuth トークンを取得するには、以下のパラメータが必要です。



パラメータ	説明
リダイレクトエンドポイント	OAuth 認証後に OAuth サービスがリダイレクトする Zabbix フロントエンドの URL を入力します（形式：`https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`）。 フロントエンド URLが定義されている場合、新しいメールメディアタイプには自動的に設定されます。
クライアント *ID*	OAuth 認証サーバーに登録されたクライアントアプリの一意の識別子を入力します。
クライアントシークレット	OAuth 認証サーバーに登録されたクライアントアプリの秘密鍵を入力します。
認証エンドポイント	ユーザー認証を要求する OAuth 認証サーバーの URL を入力します。
認証パラメータ	認証エンドポイントのパラメータを入力します。
認証コード	認証コードを入力します： 自動 - リダイレクトページを介して自動的にコードを取得します 手動 - 自動取得に失敗した場合は手動でコードを入力します
トークンエンドポイント	認証コードをアクセストークンおよびリフレッシュトークンと交換するための OAuth 認証サーバーの URL を入力します。
トークンパラメータ	アクセストークンのパラメータを入力します。

Note:
OAuth トークンの取得は、**Gmail**、**Gmail** リレー、**Office365** のメールプロバイダーに対して自動化機能があります。リダイレクトエンドポイント、クライアント *ID*、クライアントシークレットの値を入力するだけで十分です。Zabbix は他の必要な値を自動的に入力します（プロバイダー別の OAuth URL デフォルト値を参照）。

OAuth アクセストークンとリフレッシュトークンの取得

OAuth パラメータを送信した後:

1. ブラウザのポップアップが開き、ユーザーを認可エンドポイントに誘導します。
2. ユーザーは OAuth サービスで Zabbix を認可します。
3. OAuth サービスは、認可コードとスコープ値を持って、ユーザーを Zabbix フロントエンドアクション `oauth.authorize` にリダイレクトします。
4. その応答として、Zabbix はトークンエンドポイントにリクエストを行い、認可コードをアクセストークンとリフレッシュトークンに交換します。

テスト

設定したメールメディアタイプが正しく動作するかどうかをテストするには:

1. メディアタイプのリストから該当するメールを探します。
2. リストの最終列にあるテストをクリックします (テスト用ウィンドウが開きます)。
3. 送信先の受信者アドレス、メッセージ本文、および必要に応じて件名を入力します。
4. テストをクリックしてテストメッセージを送信します。

テストの成功または失敗のメッセージが同じウィンドウに表示されます:

ユーザーメディア

メールメディアタイプの設定が完了したら、ユーザー > ユーザーセクションに移動し、ユーザープロファイルを編集してメールメディアをユーザーに割り当てます。ユーザーメディアの設定手順はすべてのメディアタイプで共通であり、メディアタイプのページで説明されています。

1 自動化された Gmail/Office365 メディアタイプ

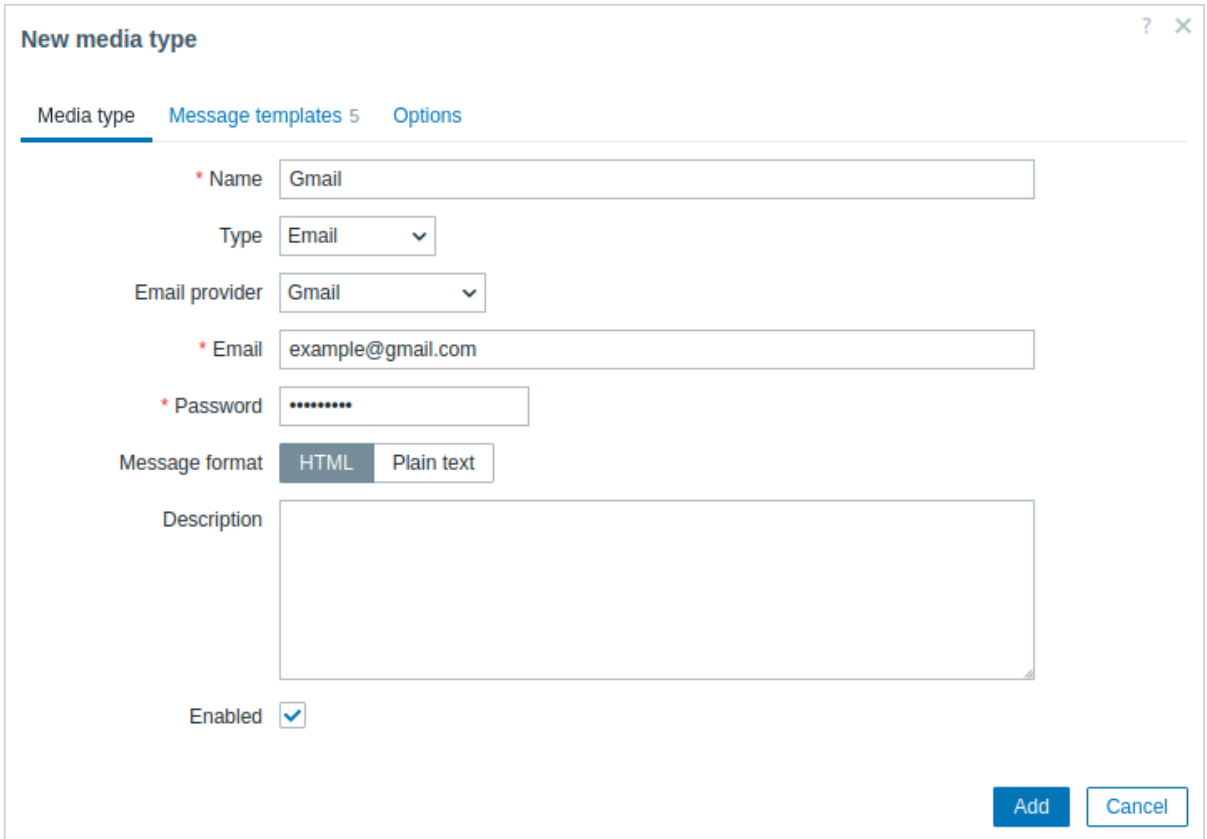
概要

Gmail や Office365 のユーザーは、メディアタイプの設定における自動化機能の恩恵を受けることができます。

設定

メールメディアタイプのメールプロバイダーフィールドでは、Gmail と Office 365 の事前設定されたオプションを選択できます。

Gmail/Office365 関連のオプションを選択すると、送信者のメールアドレス/パスワードを入力するだけで、動作するメディアタイプを作成できます。



メールアドレス/パスワードが入力されると、Zabbix は Gmail/Office365 メディアタイプに必要なすべての設定（*SMTP* サーバー、*SMTP* サーバーポート、*SMTP helo*、接続のセキュリティ）を自動的に実際の/推奨値で入力できるようになります。この自動化により、これらのフィールドは表示されませんが、メディアタイプリストの詳細列で SMTP サーバーやメールの詳細を確認できます。

また、以下にも注意してください。

- リレーオプションの場合、パスワードは不要です。
- Office365 リレーの場合、指定されたメールアドレスのドメイン名が SMTP サーバーを動的に入力するために使用されます（つまり、`example-com.mail.protection.outlook.com` の”example.com” を実際の値に置き換えます）。

OAuth トークン

OAuth トークンの取得は、*Gmail*、*Gmail* リレー、*Office365* のメールプロバイダーに対して自動化されています。



OAuth のアクセストークンおよびリフレッシュトークンの取得には、*Redirection endpoint*、*Client ID*、*Client secret* という限られたパラメータのみを指定する必要があります。Zabbix は他の必要な値を自動的に入力します（プロバイダー別の OAuth URL デフォルト値を参照）。

これらのプロバイダーに対して OAuth 認証を設定する場合、Generic SMTPフォームを使用することも可能です。

Note:
SmtpClientAuthentication は、Office365 でユーザー/共有メールボックスごと、またはテナント全体で有効にする必要があります（デフォルト設定では有効になっていません）。

プロバイダーごとの OAuth URL のデフォルト

以下の表は、Zabbix で使用されるプロバイダーごとのデフォルトの OAuth URL 値とパラメーターを示しています。

パラメーター		Gmail	Office365	Generic SMTP
認可エンドポイント		`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authori`	デフォルトなし
	scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	デフォルトなし
	access_type	`offline`	未使用	デフォルトなし
	prompt	`consent`	未使用	デフォルトなし
	redirect_uri	*Redirection endpoint* フィールドの値が自動的にコードで追加されます。*Authorization endpoint* パラメーターには記載されません。		
	state	OAuth トークンで更新されるメディアを識別する一意のハッシュが自動的にコードで追加されます。*Authorization endpoint* パラメーターには記載されません。		

パラメーター		Gmail	Office365	Generic SMTP
トークンエンドポイント		`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	デフォルトなし
	grant_type	`authorization_code`	`authorization_code`	デフォルトなし
	redirect_uri	*Redirection endpoint* フィールドの値が自動的にコードで追加されます。*Token endpoint* パラメーターには記載されません。		
	state	OAuth トークンで更新されるメディアを識別する一意のハッシュが自動的にコードで追加されます。*Token endpoint* パラメーターには記載されません。		

Office365 の個人アカウントの回避策

Office365 の個人アカウントは、リダイレクト URL のクエリ文字列をサポートしていません。

この問題を回避するために、Apache ウェブサーバーの管理者は、以下のリライトルールを .htaccess ファイルに追加できます。

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

これにより、以下のような簡略化されたリダイレクト URL が使用できるようになります。

```
http://server-name/zabbix/oauth.authorize
```

これにより、クエリパラメータ付きの URL の代わりに Office365 の個人アカウントとの互換性が確保されます。

2 SMS

概要

Zabbix は Zabbix server のシリアルポートに接続されたシリアル GSM モデムを使用した SMS メッセージの送信に対応しています。

以下の点を確認してください:

- シリアルデバイス (Linux では通常/dev/ttyS0) の速度は GSM モデムの速度に合わせます。Zabbix はシリアルリンクの速度を設定しません。デフォルトの設定を使用します。
- zabbix ユーザは、シリアルデバイスへのリード/ライトアクセス権を持ちます。ls -l /dev/ttyS0 コマンドを実行し、シリアルデバイスの現在のパーミッションを確認します。
- GSM モデムには PIN が入力されており、パワーリセット後も PIN が保持されます。また、SIM カードの PIN を無効化することもできます。PIN は Unix の minicom や Windows の HyperTerminal などのターミナルソフトで AT+CPIN="NNNN"（NNNN は PIN 番号、引用符は必ず入れてください）コマンドを発行することで入力することができます。

Zabbix は以下の GSM モデムでテストされています:

- Siemens MC35
- Teltonika ModemCOM/G10

メッセージの配信チャネルとして SMS を設定するには、メディアタイプとして SMS を設定し、ユーザーのそれぞれの電話番号を入力する
 必要があります。

設定

SMS をメディアタイプとして設定するには:

- アラート → メディアタイプに移動します
- メディアタイプの作成をクリックします（または、定義済みメディアタイプのリストから *SMS* をクリックします）。

SMS メディアタイプに固有のパラメータは以下の通りです:

パラメータ	説明
GSM モデム	GSM モデムのシリアルデバイス名を設定します。 ここで入力したパスは、SMSDevices サーバーパラメータ（指定されている場合）に対して検証されます。

デフォルトメッセージやアラート処理オプションの設定方法については、共通メディアタイプパラメータを参照してください。SMS 通知の送信の並列処理はできないことに注意してください。

ユーザーメディア

SMS メディアタイプを設定したら、ユーザー → ユーザーセクションに移動し、ユーザープロファイルを編集して SMS メディアをユーザーに割り当てます。ユーザーメディアの設定手順はすべてのメディアタイプで共通であり、メディアタイプのページで説明されています。

3 カスタムアラートスクリプト

概要

アラート送信用の既存のメディアタイプに満足できない場合、別の方法があります。独自の方法で通知を処理するスクリプトを作成できます。

カスタムアラートスクリプトは Zabbix サーバー上で実行されます。これらのスクリプトは、サーバーの設定ファイルのAlertScriptsPathパラメータで指定されたディレクトリに配置する必要があります。

以下はカスタムアラートスクリプトの例です。

```
#####!/bin/bash

to=$1
subject=$2
body=$3
host=$4
value=$5

cat <<EOF | mail -s "$subject" "$to"
$body

Host: $host
Value: $value
EOF
```

Attention:
Zabbix は実行されたコマンドやスクリプトの終了コードをチェックします。**0** 以外の終了コードは、コマンド実行エラーと見なされます。このような場合、Zabbix は失敗した実行を再試行します。

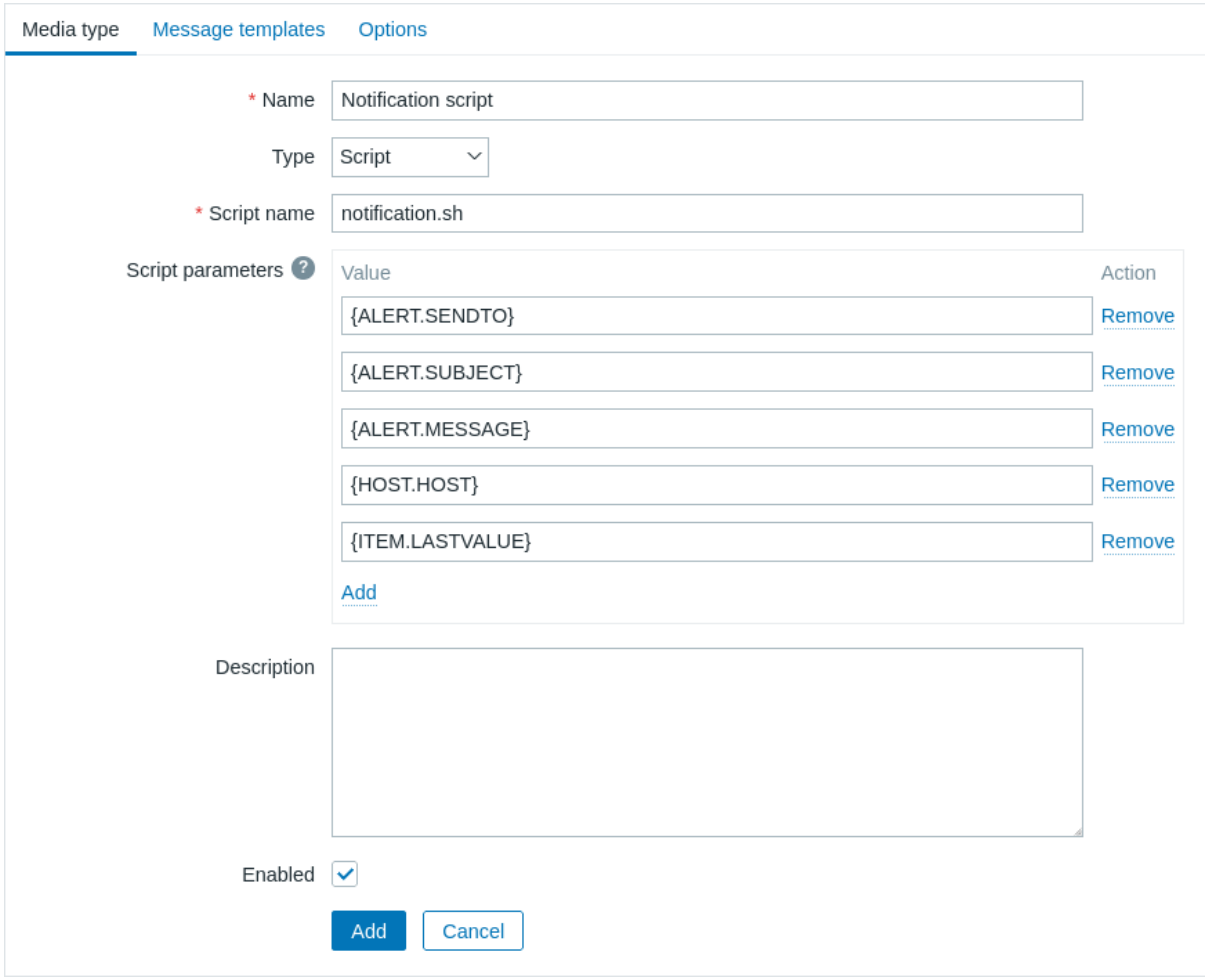
スクリプトには環境変数が保持または作成されないため、明示的に処理する必要があります。

設定

カスタムアラートスクリプトをメディアタイプとして設定するには:

1. 監視データ → メディアタイプに移動します。
2. メディアタイプの作成をクリックします。

メディアタイプタブには、一般的なメディアタイプの属性が含まれています:



すべての必須入力フィールドには赤いアスタリスクが付いています。

スクリプトメディアタイプに固有のパラメータは以下の通りです:

パラメータ	説明
スクリプト名	AlertScriptsPath サーバー設定パラメータで指定されたディレクトリにあるスクリプトファイル名（例: notification.sh）を入力します。
スクリプトパラメータ	コマンドライン引数としてスクリプトに渡されるオプションのスクリプトパラメータを、定義された順序で追加します。 スクリプトパラメータは、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロ、および通知でサポートされているすべてのマクロ、さらにユーザーマクロをサポートします。

デフォルトメッセージやアラート処理オプションの設定方法については、共通のメディアタイプパラメータを参照してください。

Warning:
アラートスクリプトがデフォルトメッセージを使用しない場合でも、このメディアタイプで使用される操作タイプのメッセージテンプレートを定義する必要があります。そうしないと、通知は送信されません。

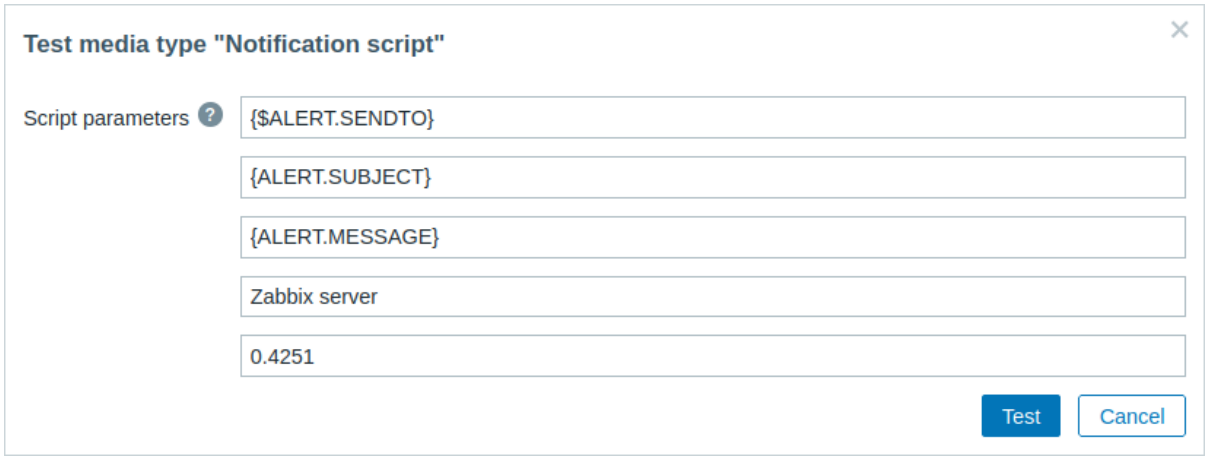
Attention:
複数のスクリプトメディアタイプが設定されている場合、これらのスクリプトは alerter プロセスによって並列に処理されることがあります。alerter プロセスの総数は、サーバー設定ファイルの`StartAlerters`パラメータで制限されます。

テスト

設定済みのスクリプトメディアタイプをテストするには:

1. メディアタイプのリストから該当するスクリプトを探します。
2. リストの最後の列にあるテストをクリックします。テスト用のフォームがポップアップウィンドウで開きます。テストフォームには、スクリプトメディアタイプに設定されているのと同じ数のパラメータが含まれます。

3. 必要に応じてスクリプトパラメータの値を編集します。編集はテスト手順にのみ影響し、実際の値は変更されません。
4. テストをクリックします。



Note:
設定済みのスクリプトメディアタイプをテストする際、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} およびユーザーマクロは値に展開されますが、イベントに関連するマクロ (例: {HOST.HOST}、{ITEM.LASTVALUE} など) は展開されません。テスト時には詳細を取得するための関連イベントが存在しないためです。また、{ALERT.SUBJECT} や {ALERT.MESSAGE} マクロ内のマクロも展開されないことに注意してください。たとえば、{ALERT.SUBJECT} の値が「Problem: {EVENT.NAME}」で構成されている場合、{EVENT.NAME} マクロは展開されません。

ユーザーメディア

メディアタイプを設定したら、ユーザー → ユーザーセクションに移動し、このメディアタイプをユーザーに割り当ててユーザープロファイルを編集します。ユーザーメディアの設定手順はすべてのメディアタイプで共通であり、メディアタイプのページで説明されています。

ユーザーメディアを定義する際、送信先フィールドは空にできないことに注意してください。このフィールドがアラートスクリプトで使用されない場合は、検証要件を回避するためにサポートされている任意の文字の組み合わせを入力してください。

4 Webhook

概要

Webhook メディアタイプは、カスタム JavaScript コードを使用して HTTP コールを行うのに便利で、ヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアとの簡単な統合が可能です。Zabbix が提供するインテグレーションをインポートするか、独自のインテグレーションをゼロから作成することができます。

インテグレーション

以下のインテグレーションが利用可能であり、Zabbix 通知をプッシュするための定義済み Webhook メディアタイプを使用できます:

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPi
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty

- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
ここに記載されているサービスに加えて、Zabbix は **Spiceworks** とも連携できます（Webhook は不要です）。Zabbix の通知を Spiceworks のチケットに変換するには、メールメディアタイプを作成し、指定した Zabbix ユーザーのプロファイル設定で Spiceworks ヘルプデスクのメールアドレス（例: help@zabbix.on.spiceworks.com）を入力してください。

設定

Webhook インテグレーションの使用を開始するには:

1. ダウンロードした Zabbix バージョンの `templates/media` ディレクトリで必要な.yaml ファイルを探すか、Zabbix のgit リポジトリからダウンロードします。
2. ファイルを Zabbix インストールにインポートします。Webhook はメディアタイプの一覧に表示されます。
3. *Readme.md* ファイルの手順に従って Webhook を設定します（上記の Webhook 名をクリックすると *Readme.md* にすぐアクセスできます）。

独自の Webhook をゼロから作成するには:

1. アラート > メディアタイプに移動します。
2. メディアタイプの作成をクリックします。

メディアタイプタブには、このメディアタイプ特有のさまざまな属性があります:

必須入力フィールドは赤いアスタリスクでマークされています。

Webhook メディアタイプ特有のパラメータは以下の通りです:

パラメータ	説明
パラメータ	Webhook の変数を属性と値のペアで指定します。 事前設定された Webhook の場合、パラメータのリストはサービスによって異なります。パラメータの説明は Webhook の *Readme.md* ファイルを確認してください。 新しい Webhook の場合、いくつかの共通変数がデフォルトで含まれています（URL:< 空 >, HTTPProxy:< 空 >, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}）。そのまま残すか削除してください。 Webhook パラメータはユーザーマクロ、問題通知でサポートされているすべてのマクロに加え、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロをサポートします。 HTTP プロキシを指定する場合、このフィールドはアイテム設定のHTTP プロキシフィールドと同じ機能をサポートします。プロキシ文字列は、どの種類のプロキシを使用するかを指定するために `[scheme]://`をプレフィックスとして付けることができます（例: https, socks4, socks5; ドキュメントを参照）。

パラメータ	説明
スクリプト	パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。このコードが Webhook 操作を実行します。 スクリプトはパラメータ-値のペアを受け取る関数コードです。値は JSON.parse() メソッドを使用して JSON オブジェクトに変換する必要があります。例: `var params = JSON.parse(value);`。 コードはすべてのパラメータにアクセスでき、HTTP GET、POST、PUT、DELETE リクエストを実行でき、HTTP ヘッダーやリクエストボディを制御できます。 スクリプトには return 演算子が含まれている必要があります。そうでない場合は無効になります。 OK ステータスとオプションでタグとタグ値のリスト（タグの処理オプションを参照）またはエラー文字列を返すことができます。 リカバリエベント（自動生成または手動クローズの結果として生成）はサーバーによって作成され、解決済みイベントタグ（テンプレート、ホスト、トリガーから継承されたタグを含む）を含みます。 Webhook スクリプトはアラート作成後に実行されるため、Webhook スクリプトで返されたタグは初回アラート作成後にのみ追加され、初回の問題メッセージや即時リカバリーメッセージの`{EVENT.TAGS}`および`{EVENT.RECOVERY.TAGS}`マクロには存在しません。 注意: 各スクリプトが独自のデータで動作し、同時呼び出し間の衝突を回避するために、グローバル変数（例: `global = 1`）ではなくローカル変数（例: `var local = 1`）を使用することを推奨します（既知の問題を参照）。 参考: Webhook 開発ガイドライン、Webhook スクリプト例、追加の JavaScript オブジェクト。
タイムアウト	JavaScript 実行のタイムアウト（1-60 秒、デフォルト 30 秒）。 30s、1m などの時間サフィックスがサポートされています。
タグの処理	返された JSON プロパティ値をタグとして処理するにはチェックボックスをマークします。これらのタグは既存の問題タグに追加されます。 Webhook タグを使用する場合、Webhook は少なくとも空の tags オブジェクトを含む JSON オブジェクトを返す必要があります: `var result = {tags: {}};` 返すことができるタグの例: *jira-id:prod-1234*、*responsible:John Smith*、*processed:*< 値なし >
イベントメニューエントリを含める	作成された外部チケットへのリンクを含むイベントメニューにエントリを含めるにはチェックボックスをマークします。 有効でこのチェックボックスがマークされている各 Webhook についてエントリが含まれます。メニューエントリ名およびメニューエントリ *URL* パラメータに`{EVENT.TAGS.<tag name>}`マクロが含まれている場合、これらのマクロが解決できる場合（つまり、イベントにこれらのタグが定義されている場合）にのみエントリが含まれます。 マークされている場合、Webhook は異なるユーザーへの通知送信には使用しないでください（代わりに専用ユーザーの作成を検討してください）し、単一の問題イベントに対する複数のアラートアクションで使用しないでください。
メニューエントリ名	メニューエントリ名を指定します。 `{EVENT.TAGS.<tag name>}`マクロがサポートされています。 このフィールドはイベントメニューエントリを含めるがマークされている場合のみ必須です。
メニューエントリ *URL*	メニューエントリの基礎となる URL を指定します。 `{EVENT.TAGS.<tag name>}`マクロがサポートされています。 このフィールドはイベントメニューエントリを含めるがマークされている場合のみ必須です。

デフォルトメッセージやアラート処理オプションの設定方法の詳細については、共通のメディアタイプパラメータを参照してください。

Warning:
Webhook がデフォルトメッセージを使用しない場合でも、この Webhook で使用される操作タイプのメッセージテンプレートは定義する必要があります。

テスト

設定済みの Webhook メディアタイプをテストするには:

1. メディアタイプの一覧から該当する Webhook を探します。
2. 一覧の最終列にあるテストをクリックします (テスト用ウィンドウが開きます)。
3. 必要に応じて Webhook パラメータの値を編集します。マクロを例示値に置き換えてください。そうしないとマクロが解決されず、テストが失敗します。
4. テストをクリックします。

テストウィンドウで値を置き換えたり削除したりしても、テスト手順のみに影響し、実際の Webhook 属性値は変更されません。

テストウィンドウを閉じずにメディアタイプのテストログエントリを表示するには、ログを開くをクリックします (新しいポップアップウィンドウが開きます)。

Webhook テストが成功した場合：

- *"Media type test successful."* メッセージが表示されます。
- サーバーのレスポンスがグレーの *Response* フィールドに表示されます。
- レスポンスタイプ（JSON または String）が *Response* フィールドの下に表示されます。

Webhook テストが失敗した場合**:**

- " メディアタイプのテストに失敗しました。" というメッセージが表示され、続いて追加の失敗の詳細が表示されます。

ユーザーメディア

メディアタイプを設定したら、ユーザー > ユーザーセクションに移動し、既存のユーザーに Webhook メディアを割り当てるか、Webhook を表す新しいユーザーを作成します。既存のユーザーのユーザーメディアの設定手順は、すべてのメディアタイプに共通であり、メディアタイプのページで説明されています。

Webhook がタグを使用してチケットやメッセージ ID を保存する場合、同じ Webhook を異なるユーザーにメディアとして割り当てると、Webhook エラーが発生する可能性があるため避けてください（イベントメニューエントリを含めるオプションを利用する大多数の Webhook に該当します）。この場合、Webhook を表す専用のユーザーを作成するのがベストプラクティスです:

1. Webhook メディアタイプを設定した後、ユーザー > ユーザーセクションに移動し、Webhook を表す専用の Zabbix ユーザーを作成します。たとえば、Slack Webhook の場合はユーザー名を *Slack* にします。このユーザーは Zabbix にログインしないため、メディア以外のすべての設定はデフォルトのままでかまいません。
2. ユーザープロファイルのメディアタブで、必要な連絡先情報を指定してWebhook を追加します。Webhook が送信先フィールドを使用しない場合は、検証要件を回避するためにサポートされている任意の文字の組み合わせを入力してください。
3. このユーザーに、アラートを送信する必要があるすべてのホストに対して、少なくとも読み取り権限を付与します。

アラートアクションを設定する際は、操作の詳細のユーザーへの送信フィールドにこのユーザーを追加します。これにより、Zabbix はこのアクションからの通知に Webhook を使用するようになります。

アラートアクションの設定

アクションは、どの通知をウェブフック経由で送信するかを決定します。ウェブフックを含むアクションの設定手順は、以下の例外を除き、他のすべてのメディアタイプと同じです。

- ウェブフックがチケットやメッセージ ID の保存や更新・解決操作の処理にウェブフックタグを使用する場合、1 つの問題イベントに対して同じウェブフックを複数のアラートアクションで使用しないでください。{EVENT.TAGS.<tag name>} が存在し、ウェブフック内で更新された場合、その結果の値は未定義になります。これを回避するには、更新された値を保存するためにウェブフックで新しいタグ名を使用してください。これは、Zabbix が提供する Jira、Jira Service Desk、Mattermost、Opsgenie、OTRS、Redmine、ServiceNow、Slack、Zammad、Zendesk のウェブフックおよびイベントメニューエントリを含めるオプションを利用するほとんどのウェブフックに該当します。ただし、同じアクションの複数の操作やエスカレーションステップ、または異なる条件により同じ問題イベントでトリガーされない異なるアクションで、1 つのウェブフックを使用することは可能です。
- 内部イベント用のアクションでウェブフックを使用する場合、カスタムメッセージのチェックボックスを必ずマークし、アクション操作の設定でカスタムメッセージを定義してください。そうしないと、通知は送信されません。

1 Webhook スクリプトの例

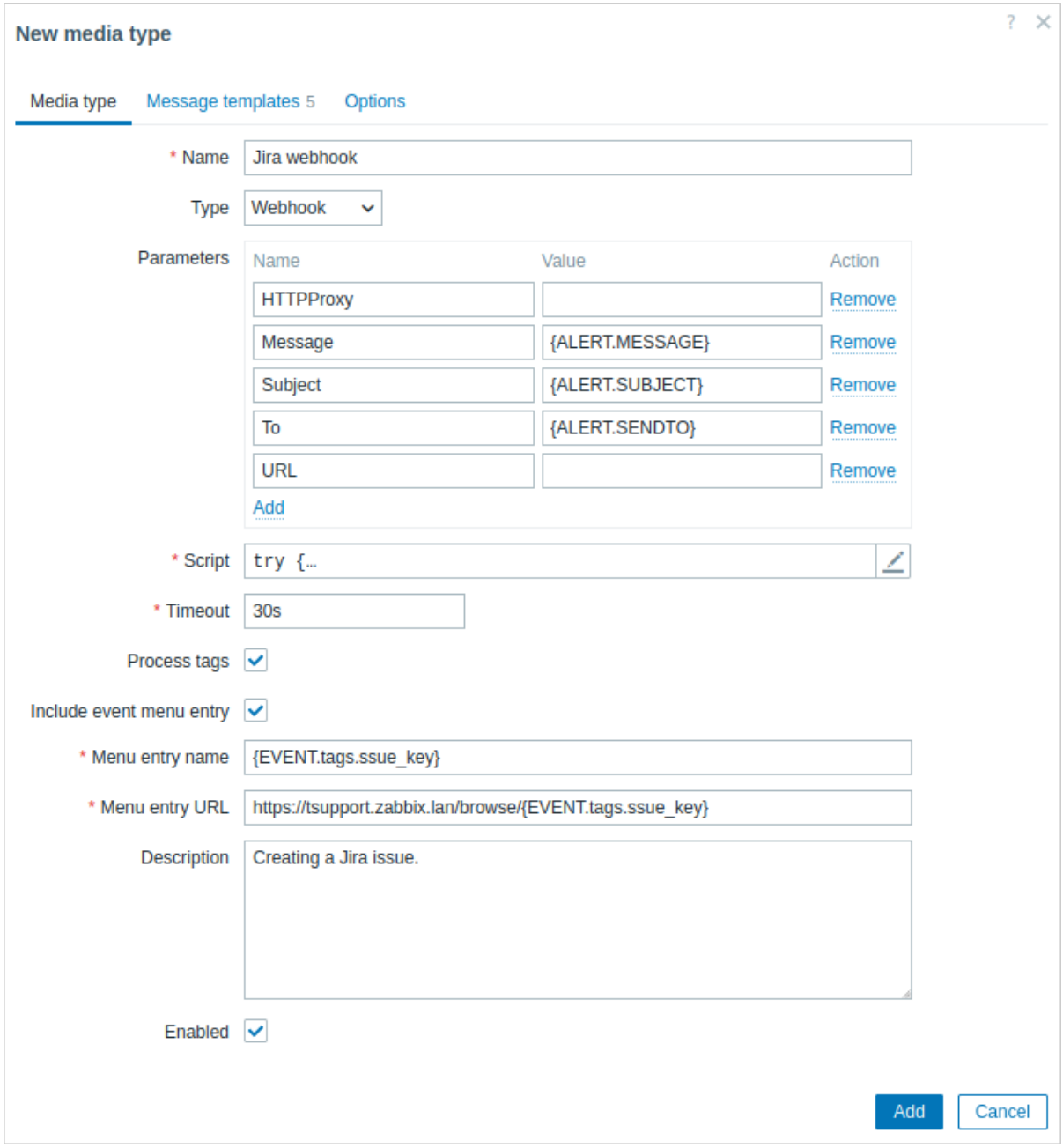
概要

Zabbix にはすぐに使える多くの Webhook 統合機能が用意されていますが、独自の Webhook を作成したい場合もあるでしょう。このセクションでは、カスタム Webhook スクリプト（*Script* パラメータで使用）の例を示します。他の Webhook パラメータの説明については、webhook セクションを参照してください。

> **Attention:**
> JavaScript の前処理で宣言されていない代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

リカバリーイベント（自動的に生成される場合や、手動クローズ に従う場合）は、解決済みイベントタグ（テンプレート、ホスト、トリガーから継承されたタグを含む）を含みます。Webhook スクリプトはアラートが作成された後に実行されるため、Webhook スクリプトによって返されたタグは初期アラート作成後に適用され、Webhook をトリガーしたアラートには存在しない場合があります。Webhook で生成されたタグが初期通知に存在する必要がある場合は、サーバーからイベントタグを取得する（たとえば、Event API を使用）か、Webhook で生成されたタグを外部の永続ストアに保存してそこで関連付けてください。

Jira webhook（カスタム）



このスクリプトは JIRA の課題を作成し、作成された課題に関する情報を返します。

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);
```

```
    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

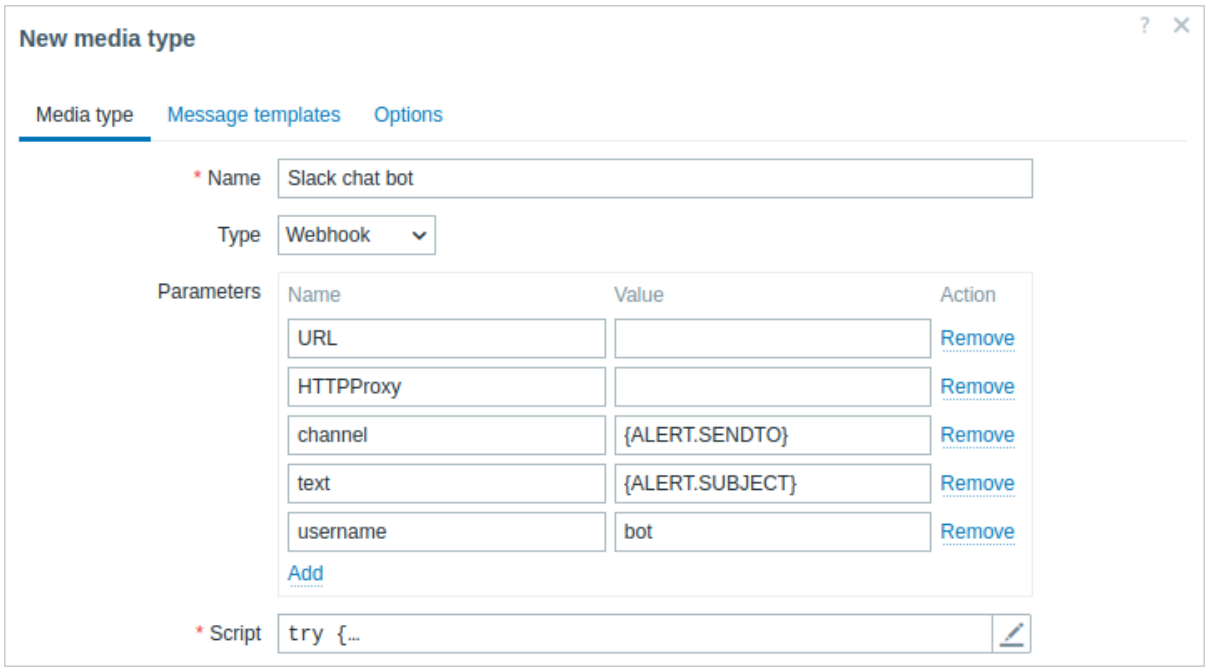
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Slack webhook (カスタム)

この Webhook は、Zabbix から Slack チャンネルへ通知を転送します。




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);
```

```
    response = req.post(params.hook_url, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

2 アクション

概要

イベントの結果として何らかの操作（たとえば通知の送信）が行われるようにしたい場合は、アクションを設定する必要があります。

アクションは、サポートされているすべてのタイプのイベントに対して定義できます。

- トリガーアクション - トリガーのステータスが *OK* から障害、またはその逆に変化したときのイベント
- サービスアクション - サービスのステータスが *OK* から障害、またはその逆に変化したときのイベント
- ディスカバリーアクション - ネットワークディスカバリーが行われたときのイベント
- 自動登録アクション - 新しいアクティブエージェントが自動登録されたとき（または登録済みホストのメタデータが変更されたとき）のイベント
- 内部アクション - アイテムがサポート対象外になったり、トリガーが不明な状態になったときのイベント

サービスアクションの主な違いは次のとおりです。

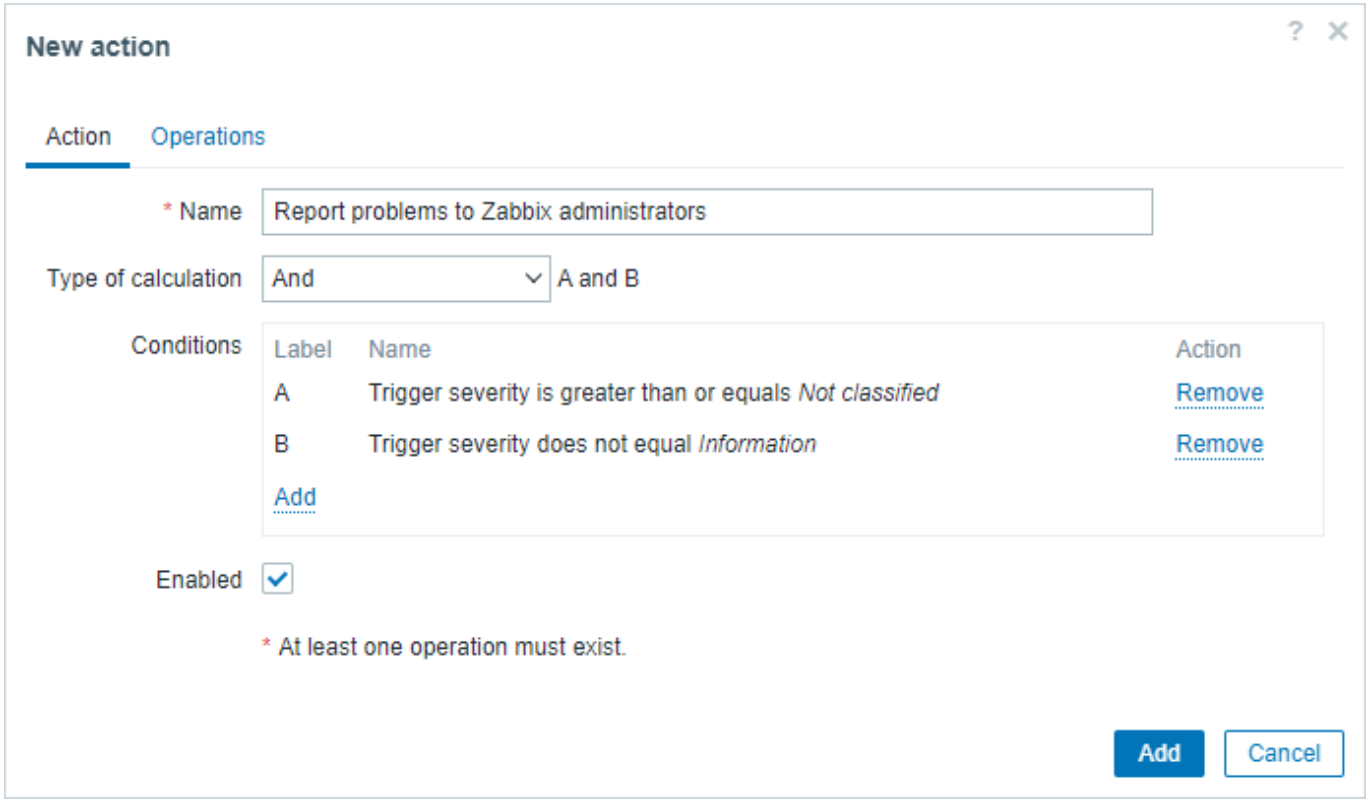
- サービスアクションへのユーザーアクセスは、ユーザーのロールによって付与されたサービスへのアクセス権に依存します
- サービスアクションは、条件の異なるセットをサポートしています

アクションの設定

アクションを設定するには、以下の手順を実行します。

- アラート > アクションに移動し、サブメニューから必要なアクションタイプを選択します（タイトルのドロップダウンで後から別のタイプに切り替えることもできます）。
- アクションの作成をクリックします。
- アクションに名前を付けます。
- 操作を実行する条件を選択します。
- 実行する操作を選択します。

アクションの一般的な属性:



必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	アクションの一意な名前。
計算タイプ	アクション条件（2 つ以上の条件がある場合）の評価オプションを選択します: **And** - すべての条件を満たす必要があります。 **Or** - いずれか 1 つの条件を満たせば十分です。 **And/Or** - 2 つの組み合わせ: 異なる条件タイプは AND、同じ条件タイプは OR。 カスタム式 - アクション条件を評価するためのユーザー定義の計算式。
条件	アクション条件のリスト。 追加をクリックして新しい条件を追加します。 条件が設定されていない場合、設定中のアクションタイプに対応するすべてのイベントでアクションが実行されます。
有効	チェックボックスをオンにするとアクションが有効になります。オフの場合は無効になります。

1 アクション

概要

イベントが定義済みの一連の条件に一致する場合にのみアクションが実行されるように定義できます。アクションの設定時に条件を設定します。

条件一致では大文字と小文字が区別されます。

トリガーアクション

トリガーベースのアクションでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
ホストグループ	equals does not equal	ホストグループまたは除外するホストグループを指定します。 **equals** - イベントがこのホストグループに属します。 **does not equal** - イベントがこのホストグループに属しません。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。親グループのみを指定するには、すべてのネストされたグループを **does not equal** 演算子で追加指定する必要があります。

条件タイプ	サポートされる演算子	説明
テンプレート	equals does not equal	テンプレートまたは除外するテンプレートを指定します。 **equals** - イベントがこのテンプレートから継承されたトリガーに属します。 **does not equal** - イベントがこのテンプレートから継承されたトリガーに属しません。
ホスト	equals does not equal	ホストまたは除外するホストを指定します。 **equals** - イベントがこのホストに属します。 **does not equal** - イベントがこのホストに属しません。
タグ名	equals does not equal contains does not contain	イベントタグまたは除外するイベントタグを指定します。 **equals** - イベントがこのタグを持ちます。 **does not equal** - イベントがこのタグを持ちません。 **contains** - イベントがこの文字列を含むタグを持ちます。 **does not contain** - イベントがこの文字列を含むタグを持ちません。
タグ値	equals does not equal contains does not contain	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。 **equals** - イベントがこのタグと値を持ちます。 **does not equal** - イベントがこのタグと値を持ちません。 **contains** - イベントがこの文字列を含むタグと値を持ちます。 **does not contain** - イベントがこの文字列を含むタグと値を持ちません。
トリガー	equals does not equal	トリガーまたは除外するトリガーを指定します。 **equals** - イベントがこのトリガーによって生成されます。 **does not equal** - イベントがこのトリガー以外のトリガーによって生成されます。
イベント名	contains does not contain	トリガーによって生成されたイベント名に含まれる文字列、または除外する文字列を指定します。デフォルトでは、イベント名はトリガー名と一致しますが、トリガー設定でカスタムイベント名が指定されている場合はその限りではありません。 **contains** - イベント名がこの文字列を含みます。 **does not contain** - この文字列がイベント名から除外されます。 注: 入力された値は、すべてのマクロが展開されたイベント名と比較されます。
トリガーの深刻度	equals does not equal is greater than or equals is less than or equals	トリガーの深刻度を指定します。 **equals** - トリガーの深刻度と等しい。 **does not equal** - トリガーの深刻度と等しくない。 **is greater than or equals** - トリガーの深刻度以上。 **is less than or equals** - トリガーの深刻度以下。
時間帯	in not in	時間帯または除外する時間帯を指定します。 **in** - イベントの発生時刻が時間帯内。 **not in** - イベントの発生時刻が時間帯外。 書式の説明については時間帯の指定ページを参照してください。 ユーザーマクロがサポートされています。
問題が抑制されている	no yes	ホストのメンテナンスにより問題が抑制（非表示）されているかどうかを指定します。 **no** - 問題は抑制されていません。 **yes** - 問題は抑制されています。

サービスアクション

サービスアクションでは、以下の条件を使用できます。

条件タイプ	サポートされている演算子	説明
サービス	等しい 等しくない	サービスまたは除外するサービスを指定します。 等しい - イベントがこのサービスに属している。 等しくない - イベントがこのサービスに属していない。 親サービスを指定すると、すべての子サービスが暗黙的に選択されます。親サービスのみを指定する場合は、すべてのネストされたサービスを等しくない演算子で追加指定する必要があります。

条件タイプ	サポートされている演算子	説明
サービス名	含む 含まない	サービス名に含まれる文字列または除外する文字列を指定します。 含む - サービス名にこの文字列が含まれるサービスによってイベントが生成される。 含まない - サービス名にこの文字列が含まれていない。
サービスタグ名	等しい 等しくない 含む 含まない	イベントタグまたは除外するイベントタグを指定します。サービスイベントタグは、サービスの設定セクションタグで定義できます。 等しい - イベントにこのタグがある。 等しくない - イベントにこのタグがない。 含む - この文字列を含むタグがイベントにある。 含まない - この文字列を含むタグがイベントにない。
サービスタグ値	等しい 等しくない 含む 含まない	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。サービスイベントタグは、サービスの設定セクションタグで定義できます。 等しい - イベントにこのタグと値がある。 等しくない - イベントにこのタグと値がない。 含む - この文字列を含むタグと値がイベントにある。 含まない - この文字列を含むタグと値がイベントにない。

Attention:
アラート → メディアタイプメニューで、サービスアクション用のメッセージテンプレートを必ず定義してください。そうしないと、通知は送信されません。

ディスカバリーアクション

ディスカバリーに基づくイベントでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
ホスト *IP*	等しい 等しくない	検出されたホストの IP アドレス範囲または除外する範囲を指定します。 等しい - ホスト IP が範囲内。 等しくない - ホスト IP が範囲外。 以下の形式が使用できます: 単一 IP: 192.168.1.33 IP アドレスの範囲: 192.168.1-10.1-254 IP マスク: 192.168.4.0/24 リスト: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 リスト形式ではスペースもサポートされます。
サービス種別	等しい 等しくない	検出されたサービスの種別または除外するサービス種別を指定します。 等しい - 検出されたサービスと一致。 等しくない - 検出されたサービスと一致しない。 利用可能なサービス種別: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping, telnet。
サービスポート	等しい 等しくない	検出されたサービスの TCP ポート範囲または除外する範囲を指定します。 等しい - サービスポートが範囲内。 等しくない - サービスポートが範囲外。
ディスカバリールール	等しい 等しくない	ディスカバリールールまたは除外するディスカバリールールを指定します。 等しい - このディスカバリールールを使用。 等しくない - このルール以外のディスカバリールールを使用。
ディスカバリーチェック	等しい 等しくない	ディスカバリーチェックまたは除外するディスカバリーチェックを指定します。 等しい - このディスカバリーチェックを使用。 等しくない - このチェック以外のディスカバリーチェックを使用。
ディスカバリーオブジェクト	等しい	検出されたオブジェクトを指定します。 等しい - 検出されたオブジェクト（デバイスまたはサービス）と等しい。

条件タイプ	サポートされる演算子	説明
ディスカバリーステータス	等しい	**Up** - 「ホスト Up」および「サービス Up」イベントと一致。 **Down** - 「ホスト Down」および「サービス Down」イベントと一致。 **Discovered** - 「ホスト検出」および「サービス検出」イベントと一致。 **Lost** - 「ホスト消失」および「サービス消失」イベントと一致。
稼働/停止時間	以上 以下	「ホスト Up」および「サービス Up」イベントの稼働時間。「ホスト Down」および「サービス Down」イベントの停止時間。 以上 - 以上。パラメータは秒単位。 以下 - 以下。パラメータは秒単位。
受信値	等しい 等しくない 以上 以下 含む 含まない	ディスカバリールールのエージェント（Zabbix, SNMP）チェックから受信した値を指定します。文字列比較。ルールに複数の Zabbix エージェントまたは SNMP チェックが設定されている場合、それぞれの受信値がチェックされます（各チェックが新しいイベントを生成し、すべての条件と照合されます）。 等しい - 値と等しい。 等しくない - 値と等しくない。 以上 - 値以上。 以下 - 値以下。 含む - 部分文字列を含む。パラメータは文字列。 含まない - 部分文字列を含まない。パラメータは文字列。
プロキシ	等しい 等しくない	プロキシまたは除外するプロキシを指定します。 等しい - このプロキシを使用。 等しくない - このプロキシ以外を使用。

Note:
ディスカバリールール内のサービスチェック（ディスカバリーイベントを生成する）は同時に実行されません。そのため、アクションのサービス種別、サービスポート、受信値条件に複数の値が設定されている場合、それらは一度に 1 つのディスカバリーイベントと比較され、複数のイベントと同時には比較されません。その結果、同じチェックタイプに複数の値を指定したアクションは正しく実行されない場合があります。

自動登録アクション

次の条件は、アクティブなエージェントの自動登録に基づくアクションで使用できます。

タイプ	オペレータ	説明
ホストメタデータ	含む 含まない 一致する 一致しない	ホストのメタデータまたは除外するホストのメタデータを指定します。 含む - ホストメタデータには指定文字列が含まれる。 含まない - ホストメタデータに指定文字列が含まれない。 ホストのメタデータはエージェント設定ファイルで指定できます。 一致する - ホストメタデータは正規表現と一致する。 一致しない - ホストメタデータは正規表現と一致しない。
ホスト名	含む 含まない 一致する 一致しない	ホスト名または除外するホスト名を指定します。 含む - ホスト名に文字列が含まれる。 含まない - ホスト名に文字列が含まれない。 ** 一致する ** - ホスト名が正規表現に一致する。 一致しない - ホスト名が正規表現に一致しない。
プロキシ	等しい 等しくない	プロキシまたは除外するプロキシを指定します。 等しい - このプロキシを使用します。 等しくない - このプロキシ以外の他のプロキシを使用します。

内部イベントアクション

内部イベントに基づくアクションには、次の条件を設定できます。

タイプ	オペレータ	説明
イベントタイプ	等しい	アイテムが”取得不可”の状態 - アイテムが’通常’から’サポートされていない’に移行するイベントに一致します ローレベルディスカバリが”取得不可”の状態 - ローレベルディスカバリルールが’通常’から’サポートされていない’に移行するイベントに一致します トリガーが”不明”の状態 - トリガーが’通常’から’不明’に移行するイベントに一致します
ホストグループ	等しい 等しくない	ホストグループまたは除外するホストグループを指定。 等しい - イベントはこのホストグループに属します。 等しくない - イベントはこのホストグループに属しません。
タグ名	等しい 等しくない 含む 含まない	イベントタグまたは除外するイベントタグを指定。 等しい - イベントにこのタグがある 等しくない - イベントにこのタグが無い 含む - イベントにこの文字列を含むタグがある * * 含まない ** - イベントにこの文字列を含むタグがない
タグの値	等しい 等しくない 含む 含まない	タグと値の組み合わせ、または除外するイベントタグと値の組み合わせを指定します。 等しい - イベントにこのタグと値がある 等しくない - イベントにこのタグと値がない 含む - イベントにこれらの文字列を含むタグと値がある 含まない - イベントにこれらの文字列を含むタグと値がない
テンプレート	等しい 等しくない	テンプレートまたは除外するテンプレートを指定します。 等しい - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属する 等しくない - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属さない
ホスト	等しい 等しくない	ホストまたは除外するホストを指定します。 等しい - イベントはこのホストに属する 等しくない - イベントはこのホストに属さない

計算タイプ

条件の計算には以下のオプションがあります:

- **And** - すべての条件が満たされる必要があります

”Trigger=” 条件として複数のトリガーを選択した場合、”And” 計算の使用は許可されていません。アクションは 1 つのトリガーのイベントに基づいてのみ実行できます。

- **Or** - いずれか 1 つの条件が満たされれば十分です
- **And/Or** - 2 つの組み合わせ: 異なる条件タイプには AND、同じ条件タイプには OR、例えば:

ホストグループが Oracle servers に等しい
ホストグループが MySQL servers に等しい
イベント名に’Database is down’ を含む
イベント名に’Database is unavailable’ を含む

これは次のように評価されます

(ホストグループが Oracle servers に等しい またはホストグループが MySQL servers に等しい**)** かつ **(**イベント名に’Database is down’ を含む またはイベント名に’Database is unavailable’ を含む**)**

- カスタム式 - アクション条件を評価するためのユーザー定義の計算式です。すべての条件（大文字の A、B、C、... で表される）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字を区別）、**or**（大文字小文字を区別）、**not**（大文字小文字を区別）を含めることができます。

And/Or の前述の例は (A or B) and (C or D) で表されますが、カスタム式では他にも複数の計算方法が可能です:

(A and B) and (C or D)
(A and B) or (C and D)
((A or B) and C) or D
(not (A or B) and C) or not D
など

アクションで使用されているオブジェクトの削除

アクションの条件/操作で使用されているオブジェクトを削除しようとすると、ユーザーには対応するエラーメッセージが表示されます。

アクションの条件または操作が以下の場合、アクションは無効になりません。

- 削除されたオブジェクトがある場合
- 低レベルディスカバリで検出されなくなり、(失われたリソースの削除期間後に) 削除されたオブジェクトがある場合

アクションの条件/操作に欠落しているオブジェクトがある場合、それらは削除されません。代わりに、削除されたオブジェクトは削除されたホストグループ、削除されたホスト、削除されたトリガーなどとして表示されます。アクションリストの「情報」列には、「このアクションには削除されたオブジェクトを参照する条件または操作があります。」というメッセージ付きの警告アイコンが表示されます。

2 実行内容

概要

すべてのイベントに対して、次の操作を定義できます。

- メッセージの送信
- リモートコマンドの実行

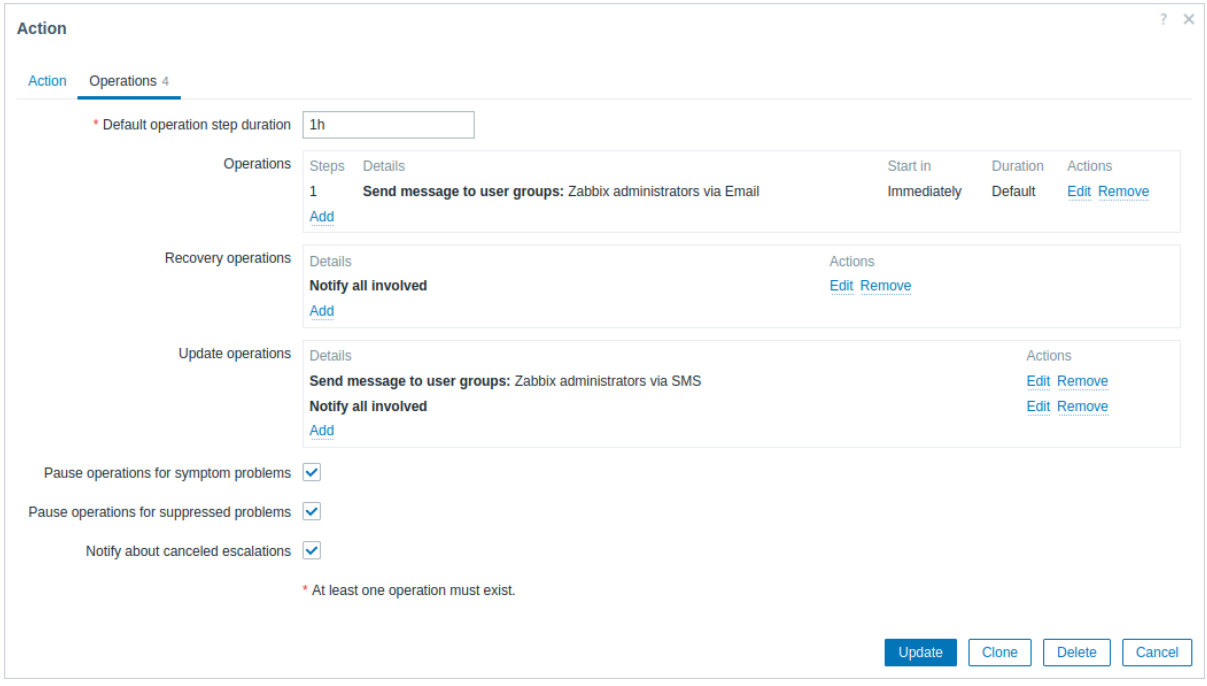
Attention:
アクション操作の受信者として定義されたユーザーに対して、ホストへのアクセスが明示的に「拒否」されている場合、またはユーザーにホストへの権限がまったく定義されていない場合、Zabbix サーバーはアラートを作成しません。

ディスカバリおよび自動登録イベントの場合、追加の操作が利用可能です。

- ホストの追加
- ホストの削除
- ホストの有効化
- ホストの無効化
- ホストグループへの追加
- ホストグループからの削除
- ホストタグの追加
- ホストタグの削除
- テンプレートのリンク
- テンプレートのリンク解除
- ホストインベントリモードの設定

オペレーションの設定

オペレーションを設定するには、アクション設定のオペレーションタブに移動します。



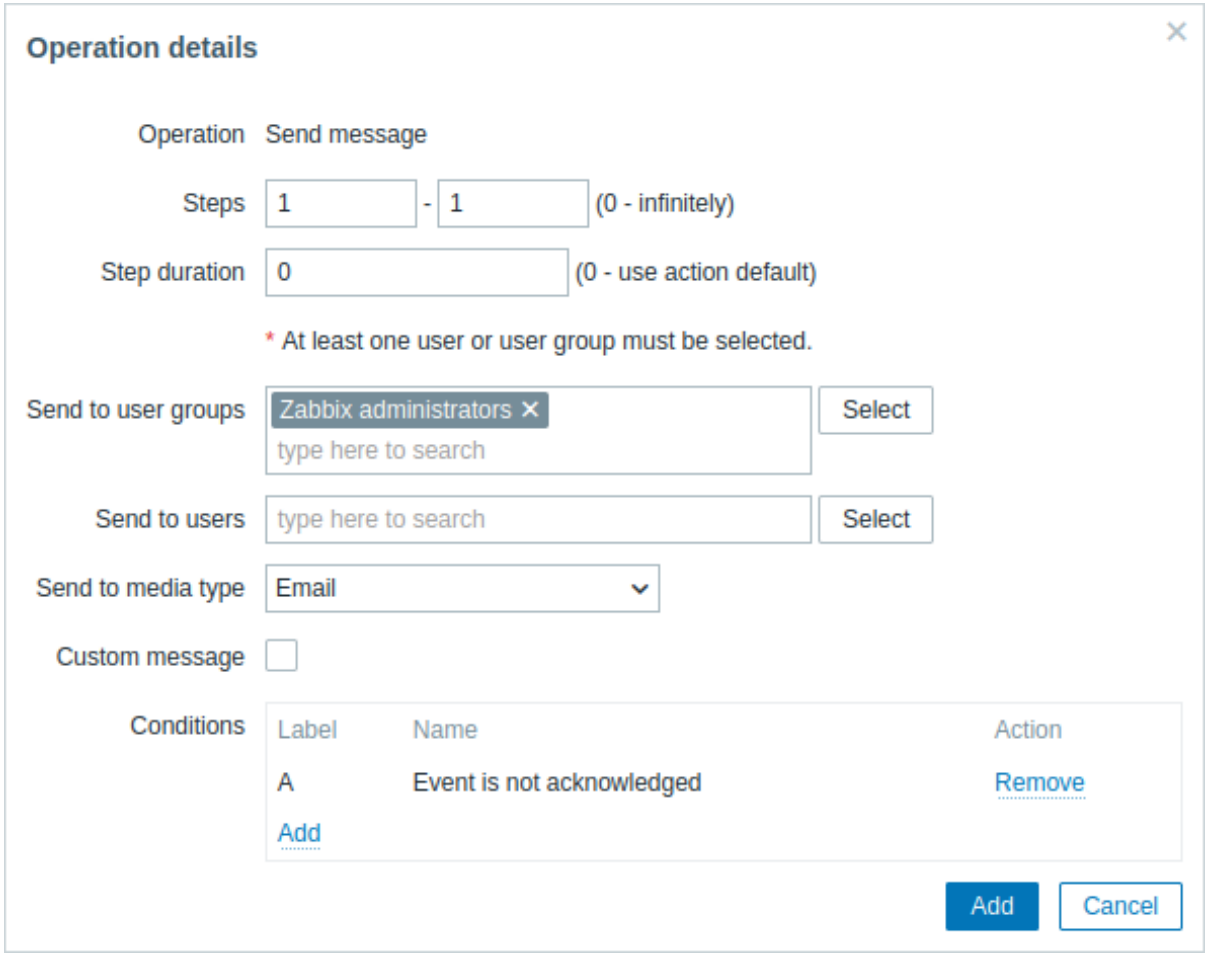
一般的なオペレーション属性:

パラメータ	説明
デフォルトのオペレーションステップの期間	デフォルトでの 1 つのオペレーションステップの期間（60 秒から 1 週間）。 例えば、1 時間のステップ期間は、オペレーションが実行された場合、次のステップまでに 1 時間が経過することを意味します。 時間のサフィックスがサポートされています。例: 60s、1m、2h、1d。 ユーザーマクロがサポートされています。
オペレーション	アクションのオペレーション（存在する場合）が、以下の詳細とともに表示されます: ステップ - オペレーションが割り当てられているエスカレーションステップ。 詳細 - オペレーションの種類とその受信者/ターゲット。 オペレーションリストには、使用されるメディアタイプ（メール、SMS、スクリプト）や、通知受信者の名前と姓（ユーザー名の後ろに括弧で表示）も表示されます。 開始まで - イベント発生後、オペレーションが実行されるまでの時間。 期間（秒） - ステップの期間が表示されます。デフォルトの期間を使用している場合は *Default*、カスタム期間を使用している場合は時間が表示されます。 アクション - オペレーションの編集および削除のリンクが表示されます。
リカバリーオペレーション	アクションのオペレーション（存在する場合）が、以下の詳細とともに表示されます: 詳細 - オペレーションの種類とその受信者/ターゲット。 オペレーションリストには、使用されるメディアタイプ（メール、SMS、スクリプト）や、通知受信者の名前と姓（ユーザー名の後ろに括弧で表示）も表示されます。 アクション - オペレーションの編集および削除のリンクが表示されます。
アップデートオペレーション	アクションのオペレーション（存在する場合）が、以下の詳細とともに表示されます: 詳細 - オペレーションの種類とその受信者/ターゲット。 オペレーションリストには、使用されるメディアタイプ（メール、SMS、スクリプト）や、通知受信者の名前と姓（ユーザー名の後ろに括弧で表示）も表示されます。 アクション - オペレーションの編集および削除のリンクが表示されます。
症状問題のオペレーションを一時停止	このチェックボックスをマークすると、症状問題に対するオペレーション（最初のオペレーションの後）が一時停止されます。 この設定は問題のエスカレーションのみに影響し、リカバリーおよびアップデートオペレーションには影響しません。 このオプションはトリガーアクションでのみ利用可能です。
抑制された問題のオペレーションを一時停止	このチェックボックスをマークすると、メンテナンス期間中はオペレーションの開始が遅延します。メンテナンス後にオペレーションが開始されると、メンテナンス中のイベントに対するものも含めてすべてのオペレーションが実行されます。 この設定は問題のエスカレーションのみに影響し、リカバリーおよびアップデートオペレーションには影響しません。 このチェックボックスのマークを外すと、メンテナンス期間中でも遅延なくオペレーションが実行されます。 このオプションはサービスアクションでは利用できません。
キャンセルされたエスカレーションの通知	このチェックボックスのマークを外すと、キャンセルされたエスカレーション（ホスト、アイテム、トリガー、アクションが無効化された場合）に関する通知が無効になります。

すべての必須入力フィールドには赤いアスタリスクが付いています。

新しいオペレーションの詳細を設定するには、オペレーションブロックの Add をクリックします。既存のオペレーションを編集するには、オペレーションの横にある Edit をクリックします。ポップアップウィンドウが開き、オペレーションステップの詳細を編集できます。

オペレーションの詳細



パラメータ	説明
オペレーションステップの期間	オペレーションを選択します: メッセージ送信 - ユーザーにメッセージを送信します。 < リモートコマンド名 > - リモートコマンドを実行します。コマンドは、事前にグローバルスクリプトでアクションオペレーションをスコープとして選択して定義されている場合に実行可能です。 ディスカバリおよび自動登録ベースのイベントには、さらに多くのオペレーションが利用可能です（上記参照）。 エスカレーションスケジュールでオペレーションを割り当てるステップを選択します: **From** - このステップから実行を開始します。 **To** - このステップまで実行します（0= 無制限、実行は制限されません）。 これらのステップのカスタム期間（0= デフォルトのステップ期間を使用）。 時間のサフィックスがサポートされています。例: 60s, 1m, 2h, 1d。 ユーザーマクロがサポートされています。 複数のオペレーションを同じステップに割り当てることができます。これらのオペレーションで異なるステップ期間が定義されている場合、最も短いものが考慮され、そのステップに適用されます。

パラメータ		説明
オペレーションタイプ: メッセージ送信		
	ユーザーグループに送信	メッセージを送信するユーザーグループを選択します。 ユーザーグループは、通知されるためにホストへの少なくとも「読み取り」権限を持っている必要があります。
	ユーザーに送信	メッセージを送信するユーザーを選択します。 ユーザーは、通知されるためにホストへの少なくとも「読み取り」権限を持っている必要があります。
	メディアタイプに送信	利用可能なすべての（設定され有効な）メディアタイプ、または特定のメディアタイプのみにメッセージを送信します。
	カスタムメッセージ	選択した場合、カスタムメッセージを設定できます。 Webhookによる内部イベントの通知の場合、カスタムメッセージは必須です。
	件名	カスタムメッセージの件名。件名にはマクロを含めることができます。255 文字までに制限されています。

パラメータ		説明
	メッセージ	カスタムメッセージ。メッセージにはマクロを含めることができます。データベースの種類によって一定の文字数に制限されています（詳細はメッセージ送信を参照してください）。
オペレーションタイプ: リモートコマンド		
	ターゲットリスト	コマンドを実行するターゲットを選択します: 現在のホスト - 問題イベントを引き起こしたトリガーのホストでコマンドを実行します。このオプションは、トリガーに複数のホストがある場合は機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。したがって、リモートコマンドはネストされたグループのホストにも実行されます。 ホストでのコマンドは一度だけ実行されます。たとえば、複数のホストグループから、または個別におよびホストグループからホストが複数回一致した場合でも、一度だけ実行されます。 カスタムスクリプトを Zabbix サーバーで実行する場合、ターゲットリストは意味がありません。この場合、より多くのターゲットを選択しても、サーバーでスクリプトが複数回実行されるだけです。 グローバルスクリプトの場合、ターゲットの選択はグローバルスクリプト設定のホストグループ設定にも依存することに注意してください。 ターゲットリストオプションはサービスアクションでは利用できません。この場合、リモートコマンドは常に Zabbix サーバーで実行されるためです。
条件		オペレーションを実行する条件: イベントが未確認 - イベントが未確認の場合のみ。 イベントが確認済み - イベントが確認済みの場合のみ。 条件オプションはトリガーアクションでのみ利用可能です。

完了したら、追加をクリックしてオペレーションをオペレーションのリストに追加します。

1 メッセージの送信

概要

メッセージの送信は、障害について人々に通知する最良の方法の 1 つです。これが、Zabbix が提供する主要なアクションの 1 つである理由です。

設定

Zabbix から通知を送受信できるようにするには、以下の作業が必要です。

- メッセージを送信するメディアを定義する

操作がユーザー設定で選択したメディアの有効期間外で行われた場合、メッセージは送信されません。

デフォルトのトリガー深刻度（「未分類」）は、検出、アクティブエージェントの自動登録、内部イベントなどのトリガー以外のイベントの通知を受信したい場合、ユーザーメディアの設定で必ずチェックする必要があります。

- 定義したメディアのいずれかにメッセージを送信するアクションの操作を設定する

Attention:
Zabbix は、イベントを生成したホストに対して少なくとも「読込」権限を持つユーザーにのみ通知を送信します。トリガー式の少なくとも 1 つのホストがアクセス可能である必要があります。

エスカレーションを使用して、メッセージ送信のカスタムシナリオを設定できます。

Zabbix からのメールを正常に受信・閲覧するには、メールサーバー/クライアントが標準の「SMTP/MIME メール」形式をサポートしている必要があります。Zabbix は UTF-8 データを送信します（件名が ASCII 文字のみの場合は UTF-8 エンコードされません）。メッセージの件名と本文は、「SMTP/MIME メール」形式の標準に従い base64 エンコードされます。

すべてのマクロ展開後のメッセージ制限は、リモートコマンドのメッセージ制限と同じです。

メッセージの追跡

監視 → 障害で送信されたメッセージのステータスを確認できます。

アクション列では、実行されたアクションの概要情報が表示されます。ここで、緑色の数字は送信されたメッセージ、赤色の数字は失敗したメッセージを表します。進行中はアクションが開始されたことを示します。失敗は、アクションが正常に実行されなかったことを示します。

イベントの時刻をクリックしてイベントの詳細を表示すると、アクションブロックでイベントにより送信された（または送信されなかった）メッセージの詳細を確認できます。

レポート → アクションログでは、アクションが設定されているイベントに対して実行されたすべてのアクションの詳細を確認できます。

2 リモートコマンド

概要

リモートコマンドを使用すると、特定の定義済みコマンドが監視対象ホストで何らかの条件で自動的に実行されるように定義できます。

リモートコマンドはスマートプロアクティブモニタリングの強力なメカニズムです。

この機能の最も明白な使用法で、次のことを試みることができます。

- 一部のアプリケーション (Web サーバー、ミドルウェア、CRM) が応答しない場合、自動的に再起動する
- 要求に応答しない場合、IPMI の'reboot' コマンドを使用してリモートサーバーを再起動する
- ディスク容量が不足した場合に自動でディスク容量を解放する (古いファイルの削除、/tmp のクリーニング)
- CPU 負荷に応じて、ある物理ボックスから別の物理ボックスに VM を移行する
- 不十分な CPU (またはディスク、メモリなど) リソースに新しいノードをクラウド環境に追加する

リモートコマンドのアクションの設定は、メッセージを送信する場合と似ていますが、Zabbix がメッセージを送信する代わりにコマンドを実行する点だけが異なります。

リモートコマンドは、Zabbix サーバー、プロキシ、またはエージェントによって実行できます。Zabbix エージェントのリモートコマンドは、Zabbix サーバーまたは Zabbix プロキシ経由で直接実行できます。Zabbix エージェントと Zabbix プロキシの両方で、リモートコマンドはデフォルトで無効になっています。それらは次の方法で有効にできます。

- エージェント設定に `AllowKey=system.run[*]` パラメーターを追加します。
- プロキシ設定で EnableRemoteCommands パラメータを'1' に設定します。

Zabbix サーバーによって実行されるリモートコマンドは、コマンドの実行 で説明されているように、終了コードのチェックを含めて実行されます。

対象ホストがメンテナンス中であってもリモートコマンドを実行します。

リモートコマンドの制限

すべてのマクロを解決した後のリモートコマンドの制限は、データベースの種類と文字セット（非 ASCII 文字は格納するのに 1 バイト以上必要）によって異なります。

データベース	文字数の制限	バイト数の制限
MySQL	65535	65535
PostgreSQL	65535	制限なし
SQLite（Zabbix プロキシのみ）	65535	制限なし

リモートコマンド実行の出力（戻り値）は 16MB に制限されます（末尾の空白も含み、切り捨てられます）。IPMI リモートコマンドの制限は、インストールされている IPMI ライブラリに依存します。すべてのリモートコマンドにはデータベースの制限が適用されることに注意してください。

設定

Zabbix エージェント上で実行されるリモートコマンド（カスタムスクリプト）は、エージェントの設定で最初に有効にする必要があります。

特定のコマンドを nowait モードで許可するには、エージェントの設定で AllowKey=system.run[<command>,*] パラメータを各許可コマンドごとに追加してください。このパラメータを変更した場合は、エージェントデーモンを再起動してください。

次に、アラート → アクション → トリガーアクションで新しいアクションを設定する場合:

1. 適切な条件を定義します。例えば、Apache アプリケーションのいずれかでディザスターの問題が発生した場合にアクションが有効になるように設定します。
2. オペレーションタブで、オペレーション、リカバリーオペレーション、またはアップデートオペレーションブロックの追加をクリックします。
3. オペレーションのドロップダウンリストから事前定義されたスクリプトのいずれかを選択し、スクリプトのターゲットリストを設定します。

事前定義されたスクリプト

アクション操作（Webhook、スクリプト、SSH、Telnet、IPMI）で利用可能なスクリプトは、グローバルスクリプトで定義されています。

例:

```
sudo /etc/init.d/apache restart
```

この場合、Zabbix は Apache プロセスの再起動を試みます。このコマンドを使用する場合は、Zabbix エージェントでコマンドが実行されることを確認してください（実行先の *Zabbix agent* ボタンをクリック）。

Attention:
sudo の使用に注意してください。Zabbix ユーザーにはデフォルトでシステムサービスを再起動する権限がありません。**sudo** の設定方法については、下記のヒントを参照してください。

Note:
Zabbix agent 7.0 以降、リモートコマンドはアクティブモードで動作するエージェントでも実行できます。Zabbix エージェント（アクティブまたはパッシブ）はリモートホストで実行され、コマンドはバックグラウンドで実行されます。

Zabbix エージェント上のリモートコマンドは、system.run[,nowait] キーによってタイムアウトなしで実行され、実行結果はチェックされません。Zabbix サーバーおよび Zabbix プロキシでは、リモートコマンドは zabbix_server.conf または zabbix_proxy.conf ファイルのTrapperTimeoutパラメータで設定されたタイムアウトで実行され、実行結果がチェックされます。詳細については、スクリプトのタイムアウトを参照してください。

アクセス権限

'zabbix' ユーザーが設定されたコマンドを実行できる権限を持っていることを確認してください。特権コマンドへのアクセスを許可するために **sudo** を使用することもできます。アクセスを設定するには、root で以下を実行します。

```
visudo
```

sudoers ファイルで使用できる例:

```
# 'zabbix'ユーザーがすべてのコマンドをパスワードなしで実行できるようにします。
zabbix ALL=NOPASSWD: ALL

# 'zabbix'ユーザーがapacheをパスワードなしで再起動できるようにします。
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
一部のシステムでは、*sudoers* ファイルがローカル以外のユーザーによるコマンドの実行を禁止しています。これを変更するには、*/etc/sudoers* の **requiretty** オプションをコメントアウトしてください。

複数のインターフェースを持つリモートコマンド

ターゲットシステムが選択されたタイプ (Zabbix エージェントまたは IPMI) の複数のインターフェースを持っている場合、リモートコマンドはデフォルトのインターフェースで実行されます。

Zabbix エージェント以外のインターフェースを使用して、SSH および Telnet 経由でリモートコマンドを実行することも可能です。使用可能なインターフェースは、次の順序で選択されます:

- Zabbix エージェントのデフォルトインターフェース
- SNMP のデフォルトインターフェース
- JMX のデフォルトインターフェース
- IPMI のデフォルトインターフェース

IPMI リモートコマンド

IPMI リモートコマンドの場合、次の構文を使用する必要があります。

```
<command> [<value>]
```

内容

- <command> - スペースなしの 1 つの IPMI コマンド
- <value> - 'on', 'off' または任意の符号なし整数。<value> はオプションパラメーターです。

例

アクション操作でリモートコマンドとして使用できるグローバルスクリプトの例。

例 **1**

特定の条件で Windows を再起動します。

Zabbix で検出された問題に応じて Windows を自動的に再起動するには、次のスクリプトを定義します。

スクリプトパラメータ	値
スコープ	' アクションオペレーション'
タイプ	' スクリプト'
コマンド	c:\windows\system32\shutdown.exe -r -f

例 **2**

IPMI 制御を使用してホストを再起動します。

スクリプトパラメータ	値
スコープ	' アクションオペレーション'
タイプ	'IPMI'
コマンド	reset

例 **3**

IPMI 制御を使用してホストの電源をオフにします。

スクリプトパラメータ	値
スコープ	' アクションオペレーション'
タイプ	'IPMI'
コマンド	power off

3 追加操作

概要

このセクションでは、ディスカバリ/自動登録イベントの追加操作の詳細を確認できます。

ホストの追加

ホストはディスカバリプロセス中に追加されます。ホストが検出されるとすぐに追加され、ディスカバリプロセスの最後ではありません。

> **Note:**
> ネットワークディスカバリは、多くの利用できないホスト/サービスがあるため、時間がかかる場合があります。忍耐強く、適切な IP レンジを使用することをお勧めします。

ホストを追加する際、その名前は標準の **gethostbyname** 関数によって決定されます。ホストが解決できる場合は、解決された名前が使用されます。できない場合は、IP アドレスが使用されます。また、ホスト名に IPv6 アドレスを使用する必要がある場合、ホスト名にコロン（:）は使用できないため、すべての「:」（コロン）は「_」（アンダースコア）に置き換えられます。

> **Attention:**
> プロキシによるディスカバリを実行している場合でも、現在はホスト名のルックアップは Zabbix サーバー上で行われます。

> **Attention:**
> Zabbix の設定に、検出された新しいホストと同じ名前のホストがすでに存在する場合、Zabbix はホスト名に **_N** を追加します。ここで **N** は 2 から始まる増加する番号です。

4 メッセージでのマクロ使用

概要

メッセージの件名やメッセージ本文では、より効率的な問題報告のためにマクロを使用できます。

多くの組み込みマクロに加えて、ユーザーマクロや式マクロもサポートされています。Zabbix でサポートされているマクロの全リストも利用できます。

例

以下の例は、メッセージでマクロを使用する方法を示しています。

例 1

メッセージの件名:

```
Problem: {TRIGGER.NAME}
```

メッセージを受信すると、メッセージの件名は次のように置き換えられます:

```
Problem: Processor load is too high on Zabbix server
```

例 2

メッセージ:

```
Processor load is: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

このメッセージを受信すると、メッセージは次のように置き換えられます:

```
Processor load is: 1.45
```

例 3

メッセージ:

```
最新値: {?last(/{HOST.HOST}/{ITEM.KEY})}
15分間の最大値: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
15分間の最小値: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

このメッセージを受信すると、次のように置き換えられます:

```
最新値: 1.45
15分間の最大値: 2.33
15分間の最小値: 1.01
```

例 4

メッセージ:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

このメッセージを受信すると、イベントの詳細ページへのリンクが含まれます。このページには、イベント、そのトリガー、および同じトリガーによって生成された最新のイベントのリストに関する情報が表示されます。

例 5

トリガー式で複数のホストからの値を通知する。

メッセージ:

```
Problem name: {TRIGGER.NAME}
Trigger expression: {TRIGGER.EXPRESSION}

1. Item value on {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Item value on {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

このメッセージを受信すると、次のように置き換えられます:

```
Problem name: Processor load is too high on a local host
Trigger expression: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[perc

1. Item value on Myhost: 0.83 (Processor load (1 min average per core))
2. Item value on Myotherhost: 5.125 (Processor load (1 min average per core))
```

例 6

リカバリメッセージで、問題イベントとリカバリエベントの両方の詳細を受信する:

メッセージ:

```
問題:

イベントID: {EVENT.ID}
イベント値: {EVENT.VALUE}
イベントステータス: {EVENT.STATUS}
イベント時刻: {EVENT.TIME}
イベント日付: {EVENT.DATE}
イベント経過時間: {EVENT.AGE}
イベントの確認: {EVENT.ACK.STATUS}
イベントの更新履歴: {EVENT.UPDATE.HISTORY}

リカバリ:

イベントID: {EVENT.RECOVERY.ID}
イベント値: {EVENT.RECOVERY.VALUE}
イベントステータス: {EVENT.RECOVERY.STATUS}
イベント時刻: {EVENT.RECOVERY.TIME}
イベント日付: {EVENT.RECOVERY.DATE}
運用データ: {EVENT.OPDATA}
```

メッセージを受信すると、マクロは次のように置き換えられます:

```
問題:

イベントID: 21874
イベント値: 1
イベントステータス: PROBLEM
イベント時刻: 13:04:30
イベント日付: 2018.01.02
イベント経過時間: 5m 0s
イベントの確認: Yes
イベントの更新履歴: 2018.01.02 13:05:51 "John Smith (Admin)"
Actions: acknowledged.

リカバリ:
```

```
イベントID: 21896
イベント値: 0
イベントステータス: OK
イベント時刻: 13:10:07
イベント日付: 2018.01.02
運用データ: Current value is 0.83
```

3 復旧時実行

概要

復旧操作により、障害が解決されたときに通知を受けることができます。

復旧操作では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加できますが、エスカレーションはサポートされていません。すべての操作が 1 つのステップに割り当てられるため、同時に実行されます。

使用例

リカバリー操作の使用例は以下の通りです。

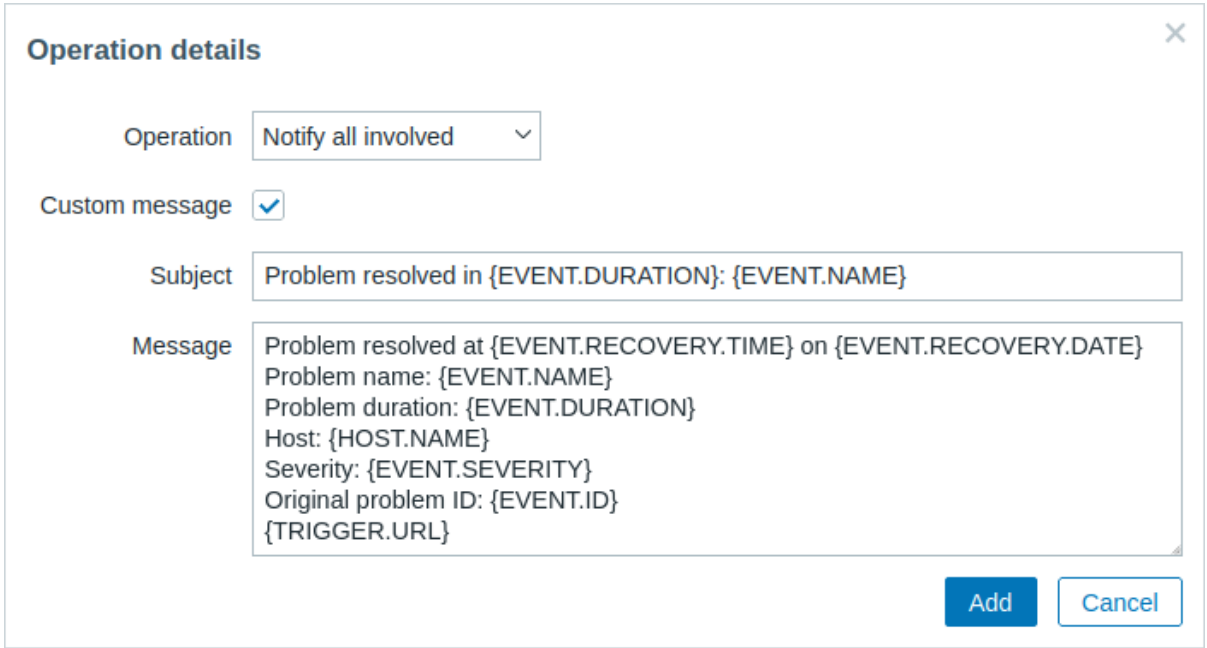
1. 問題時に通知されたすべてのユーザーにリカバリー時も通知する:
    - 操作タイプとしてすべての関係者に通知を選択します。
2. リカバリー時に複数の操作を行う: 通知を送信し、リモートコマンドを実行する:
    - メッセージ送信とコマンド実行の操作タイプを追加します。
3. 外部のヘルプデスク/チケッティングシステムでチケットをオープンし、問題が解決したらクローズする:
    - ヘルプデスクシステムと通信する外部スクリプトを作成します。
    - このスクリプトを実行してチケットをオープンする操作を持つアクションを作成します。
    - このスクリプトを他のパラメータで実行してチケットをクローズするリカバリー操作を作成します。
    - {EVENT.ID} マクロを使用して元の問題を参照します。

リカバリー操作の設定

リカバリー操作を設定するには、アクションの設定で操作タブに移動します。

新しいリカバリー操作の詳細を設定するには、リカバリー操作ブロックでをクリックします。既存の操作を編集するには、操作の横にあるをクリックします。操作ステップの詳細を編集できるポップアップウィンドウが開きます。

復旧操作の詳細



復旧イベントには 3 つの操作タイプがあります:

- メッセージ送信 - 指定したユーザーに復旧メッセージを送信します。
- 関係者全員に通知 - 問題イベントで通知されたすべてのユーザーに復旧メッセージを送信します。
- < リモートコマンド名 > - リモートコマンドを実行します。コマンドは、アクション操作をスコープとして選択してグローバルスクリプトで事前に定義されている場合に実行できます。

各操作タイプのパラメータについては以下で説明します。すべての必須入力フィールドには赤いアスタリスクが付いています。完了したら、追加をクリックして操作を復旧操作のリストに追加します。

> **Note:**
> 同じ受信者が複数の操作タイプでカスタムメッセージを指定せずに定義されている場合、重複した通知は送信されません。

操作タイプ：メッセージの送信

パラメータ		説明
ユーザーグループに送信		リカバリーメッセージを送信するユーザーグループを選択します。 通知を受けるには、ユーザーグループがホストに対して少なくとも「読み取り」権限を持っている必要があります。
ユーザーに送信		リカバリーメッセージを送信するユーザーを選択します。 通知を受けるには、ユーザーがホストに対して少なくとも「読み取り」権限を持っている必要があります。
メディアタイプに送信		すべての利用可能な（設定済みかつ有効な）メディアタイプ、または特定のメディアタイプのみにデフォルトのリカバリーメッセージを送信します。
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージ。メッセージにはマクロを含めることができます。

操作タイプ: リモートコマンド

パラメータ	説明
ターゲットリスト	コマンドを実行するターゲットを選択します: 現在のホスト - 問題イベントを引き起こしたトリガーのホストでコマンドが実行されます。このオプションは、トリガーに複数のホストがある場合は機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。したがって、リモートコマンドはネストされたグループのホストでも実行されます。 ホストでのコマンドは一度だけ実行されます。たとえば、複数のホストグループや個別、ホストグループの両方で一致した場合でも一度だけです。 コマンドが Zabbix サーバーで実行される場合、ターゲットリストは意味を持ちません。この場合、より多くのターゲットを選択しても、サーバーでコマンドが複数回実行されるだけです。 グローバルスクリプトの場合、ターゲットの選択はグローバルスクリプトの設定のホストグループ設定にも依存することに注意してください。

操作タイプ : 関係者全員に通知

パラメータ		説明
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージ。メッセージにはマクロを含めることができます。

4 更新時実行

概要

更新操作は、以下のイベントソースを持つアクションで利用できます。

- トリガー - 他のユーザーによって問題が更新された場合、つまりコメントが追加された、確認された、深刻度が変更された、手動でクローズされた場合
- サービス - サービスの深刻度が変更されたが、サービスがまだ復旧していない場合

ユーザーは自分自身の更新については通知を受け取らないことに注意してください。

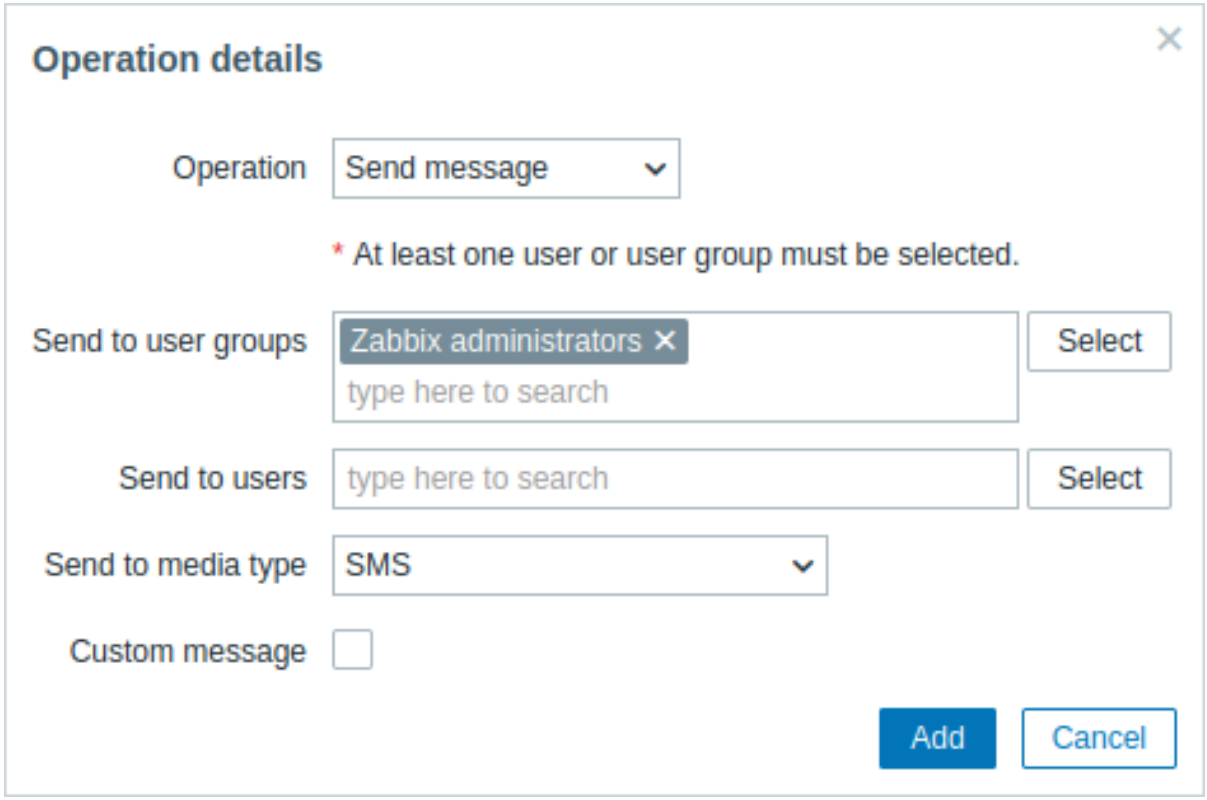
更新操作では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加することはできますが、エスカレーションはサポートされていません。すべての操作は単一のステップに割り当てられるため、同時に実行されます。

更新操作の設定

更新操作を設定するには、アクションの設定の操作タブに移動します。

新しい更新操作の詳細を設定するには、更新操作ブロックのをクリックします。既存の操作を編集するには、操作の横にあるをクリックします。操作ステップの詳細を編集できるポップアップウィンドウが開きます。

更新操作の詳細



更新操作は、リカバリ操作と同じパラメータセットを提供します。

5 エスカレーション

概要

エスカレーションを使用すると、通知を送信したり、リモートコマンドを実行したりするためのカスタムシナリオを作成できます。

具体的には、次のようなことです。

- ユーザーは新しい問題をすぐに通知できます
- 問題が解決するまで通知を繰り返すことができます
- 通知の送信を遅らせることができます
- 通知は、別の" より権限が上の" ユーザーグループにエスカレートできます
- リモートコマンドは、障害発生時すぐに、または障害が長期間解決されない場合に実行できます

アクションは、エスカレーションステップに基づいてエスカレートされます。各ステップには期間があります。

個々のステップのデフォルト期間とカスタム期間の両方を定義できます。1 つのエスカレーションステップの最小所要時間は 60 秒です。

通知の送信やコマンドの実行などのアクションは、どのステップからでも開始できます。ステップ 1 は即時のアクションです。アクションを遅らせたい場合は、後のステップに割り当てることができます。ステップごとに、いくつかのアクションを定義できます。

エスカレーションのステップ数に制限はありません。

エスカレーションは、オペレーションの設定時に定義されます。エスカレーションは障害のある操作に対してのみサポートされており、復旧はサポートされていません。

エスカレーション動作その他の例

アクションに複数のエスカレーション手順が含まれている場合、さまざまな状況で何が起こるかを考えてみましょう。

状況	動作
障害が発生したホストは、最初の障害通知が送信された後にメンテナンスに入る場合	アクション設定 の メンテナンス中の場合に実行を保留の設定に応じて、残りのエスカレーション手順はすべて、メンテナンス期間中遅延するか、遅延なしで即時実行されます。メンテナンス期間によって操作がキャンセルされることはありません。

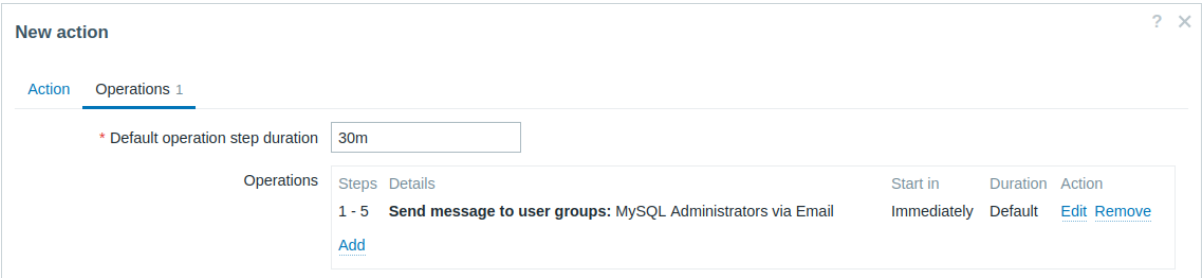
状況	動作
* 期間アクション条件で定義された期間は、最初の通知が送信された後、即時終了する場合 *	残りのすべてのエスカレーション手順が実行されます。期間条件では操作を停止できません。操作ではなく、アクションが開始されるかされないかに関して効果があります。
メンテナンス中に不具合が発生し、メンテナンス終了後も継続する（解消されない）場合	アクション設定 の メンテナンス中の場合に実行を保留の設定に応じて、すべてのエスカレーション手順は、メンテナンスが終了した瞬間、または即時実行されます。
データなしのメンテナンス中に障害が発生し、メンテナンス終了後も継続する (解決されない) 場合	すべてのエスカレーション手順が実行される前に、トリガーが起動するのを待つ必要があります。
さまざまなエスカレーションが連続して続き、重なる場合	新しい各エスカレーションの実行は前のエスカレーションに取って代わりますが、前のエスカレーションで常に実行されるエスカレーション手順が少なくとも 1 つ必要です。この動作は、トリガーのすべての障害評価で作成されたイベントに対するアクションに関連しています。
進行中のエスカレーション中 (メッセージの送信など)、あらゆるタイプのイベントに基づいて: - アクションは無効 トリガーイベントに基づいて: - トリガーは無効 - ホストまたはアイテムは無効 トリガーに関する内部イベントに基づいて: - トリガーは無効 アイテム/ローレベルディスカバリルールに関する内部イベントに基づいて: - アイテムは無効 - ホストは無効	進行中のメッセージが送信され、エスカレーションに関するメッセージがもう 1 つ送信されます。フォローアップメッセージには、メッセージ本文の先頭にキャンセルテキスト (注: エスカレーションがキャンセルされました) があり、理由が示されます (例: 注: エスカレーションがキャンセルされました: アクション'< アクション名 >' が無効化されました)。このように、エスカレーションがキャンセルされ、それ以上の手順が実行されないことが受信者に通知されます。このメッセージは、以前に通知を受け取ったすべての人に送信されます。キャンセルの理由は、サーバーのログファイルにも記録されます (デバッグレベル 3=Warning 以上)。 エスカレーションがキャンセルされましたメッセージもオペレーションが終了したが、リカバリオペレーションが設定されていて、まだ実行されていない場合に送信されます。
エスカレーションの進行中 (メッセージの送信など) にアクションが削除される	これ以上メッセージは送信されません。情報はサーバーログファイルに記録されます (デバッグレベル 3= 警告以上)

エスカレーション例

例 1

30 分ごとに繰り返し通知を送信する（合計 5 回）"MySQL Administrators" グループに送信します。設定方法は以下の通りです。

- オペレーションタブで、デフォルトのオペレーションステップの期間を"30m"（30 分）に設定します。
- エスカレーションのステップを"1" から"5" までに設定します。
- メッセージの受信者として"MySQL Administrators" グループを選択します。



問題が発生してから 0:00、0:30、1:00、1:30、2:00 の時点で通知が送信されます（もちろん、問題がそれより早く解決された場合は除きます）。

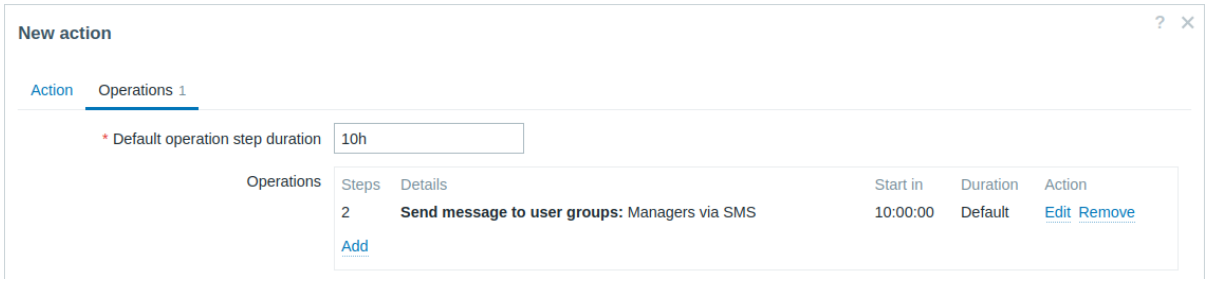
問題が解決され、リカバリーメッセージが設定されている場合、このエスカレーションシナリオで少なくとも 1 つの問題メッセージを受信した人に送信されます。

Note:
アクティブなエスカレーションを生成したトリガーが無効化された場合、Zabbix はすでに通知を受信したすべてのユーザーにその旨の情報メッセージを送信します。

例 2

長時間継続している問題について遅延通知を送信します。設定方法:

- オペレーションタブで、デフォルトのオペレーションステップの期間を「10h」(10 時間) に設定します。
- エスカレーションのステップを「2」から「2」に設定します。



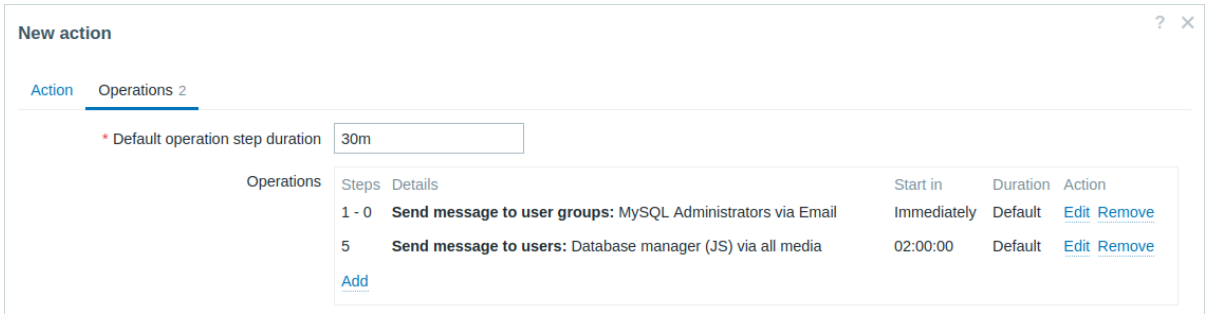
通知は、エスカレーションシナリオのステップ 2、つまり問題発生から 10 時間後にのみ送信されます。

メッセージテキストを「問題が 10 時間以上継続しています」などにカスタマイズすることもできます。

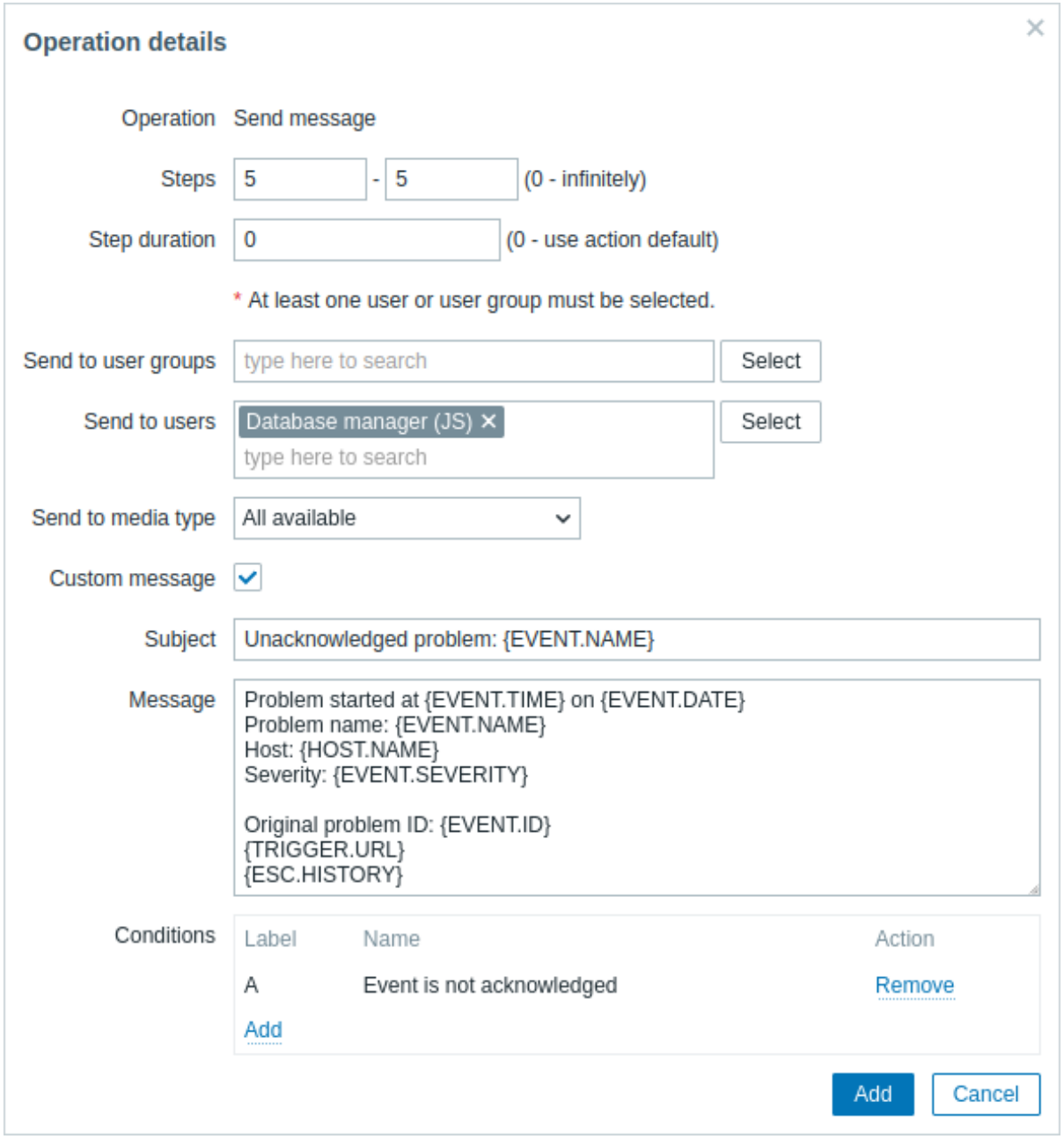
例 3

問題を上司にエスカレーションする。

上記の最初の例では、MySQL 管理者への定期的なメッセージ送信を設定しました。この場合、管理者は問題がデータベースマネージャーにエスカレーションされる前に 4 回メッセージを受け取ります。マネージャーは、問題がまだ確認されていない場合、つまり誰も対応していない場合にのみメッセージを受け取ることに注意してください。



操作 2 の詳細:



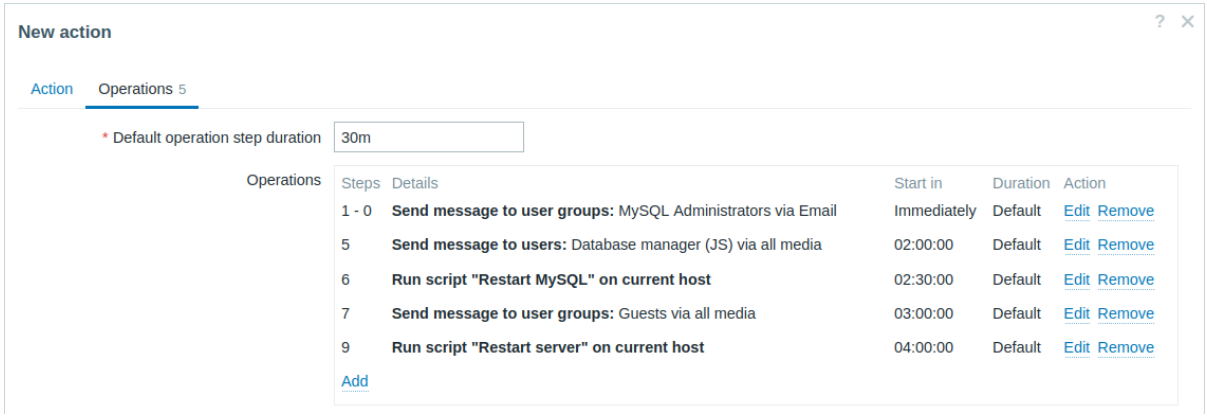
カスタマイズされたメッセージで {ESC.HISTORY} マクロを使用していることに注意してください。このマクロには、このエスカレーションで以前に実行されたすべてのステップ（送信された通知や実行されたコマンドなど）に関する情報が含まれます。

例 4

より複雑なシナリオ。MySQL 管理者への複数のメッセージとマネージャーへのエスカレーションの後、Zabbix は MySQL データベースの再起動を試みます。障害が 2:30 時間存在し、確認されていない場合に発生します。

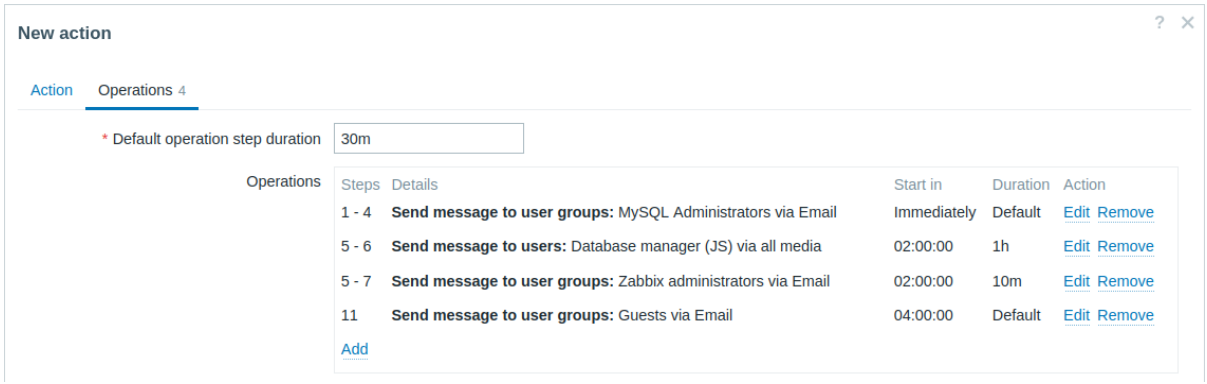
それでも障害が解決しない場合は、さらに 30 分後に Zabbix がすべてのゲストユーザーにメッセージを送信します。

これで障害が解決しない場合は、さらに 1 時間後に、Zabbix は IPMI コマンドを使用して MySQL データベース (2 番目のリモートコマンド) でサーバーを再起動します。



例 5

1 つのステップに複数のオペレーションが割り当てられ、カスタム間隔が使用されているエスカレーション。デフォルトのオペレーションステップの期間は 30 分です。



通知は以下のように送信されます:

- 問題発生後、0:00、0:30、1:00、1:30 に MySQL 管理者へ。
- 2:00 と 2:10 にデータベースマネージャーへ。(3:00 には送信されません。ステップ 5 と 6 が次のオペレーションと重複しているため、次のオペレーションで 10 分の短いカスタムステップ期間が、ここで設定しようとした 1 時間の長いステップ期間を上書きします)。
- 問題発生後、2:00、2:10、2:20 に Zabbix 管理者へ (10 分のカスタムステップ期間が有効)。
- 問題発生後、4:00 にゲストユーザーへ (ステップ 8 と 11 の間でデフォルトの 30 分のステップ期間が再び適用される)。

####3 サポート対象外通知の受信 {#manual-config-notifications-unsupported_item}

概要

Zabbix では、サポートされていないアイテムに関する通知を受け取ることができます。

これは Zabbix の内部イベントのコンセプトの一部であり、ユーザーがこれらの状況で通知を受け取ることができます。内部イベントは状態の変化を反映します:

- アイテムが「通常」から「サポート外」になったとき（およびその逆）
- トリガーが「通常」から「不明」になったとき（およびその逆）
- LLD ルールが「通常」から「サポート外」になったとき（およびその逆）

このセクションでは、アイテムがサポート外になったときに通知を受け取る方法を紹介します。

設定

おおまかには通知を設定するプロセスは以前に Zabbix でアラートを設定したことがある人には馴染みのあるものです。

ステップ 1

通知に使用するメディア（メール、SMS、スクリプトなど）を設定します。この作業を行うには、マニュアルの該当セクションを参照してください。

Attention:
内部イベントの通知にはデフォルトの重大度（「未分類」）が使用されるため、内部イベントの通知を受け取りたい場合は、ユーザーメディアの設定時にチェックを外さないでください。

ステップ 2

アラート → アクション → 内部アクションに移動します。

ページ右上のアクションの作成をクリックして、アクションの設定フォームを開きます。

ステップ 3

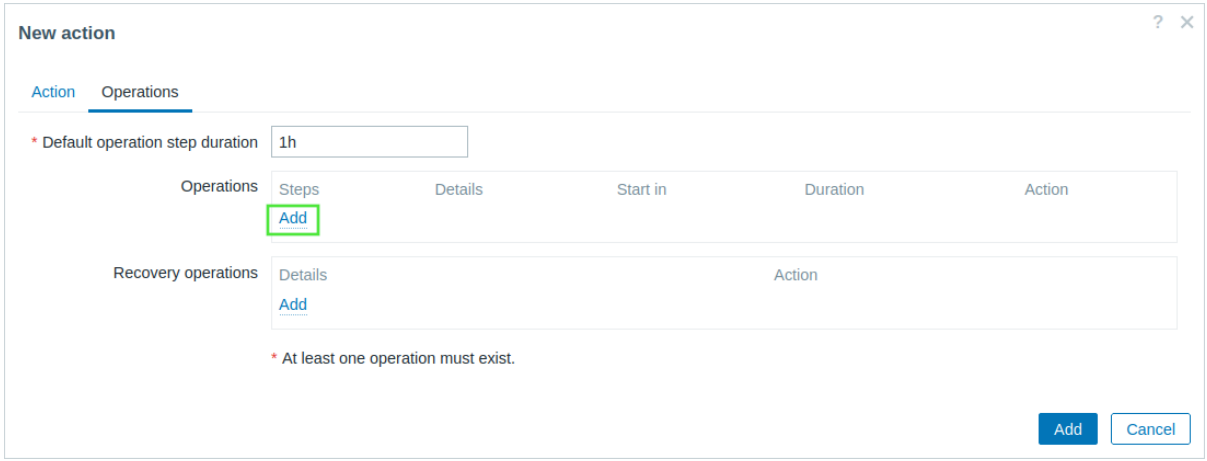
アクションタブで、アクションの名前を入力します。次に、条件ブロックの 追加をクリックして新しい条件を追加します。

新しい条件のポップアップウィンドウで、条件タイプとして「イベントタイプ」を選択し、イベントタイプとして「' サポートされていない' 状態のアイテム」を選択します。

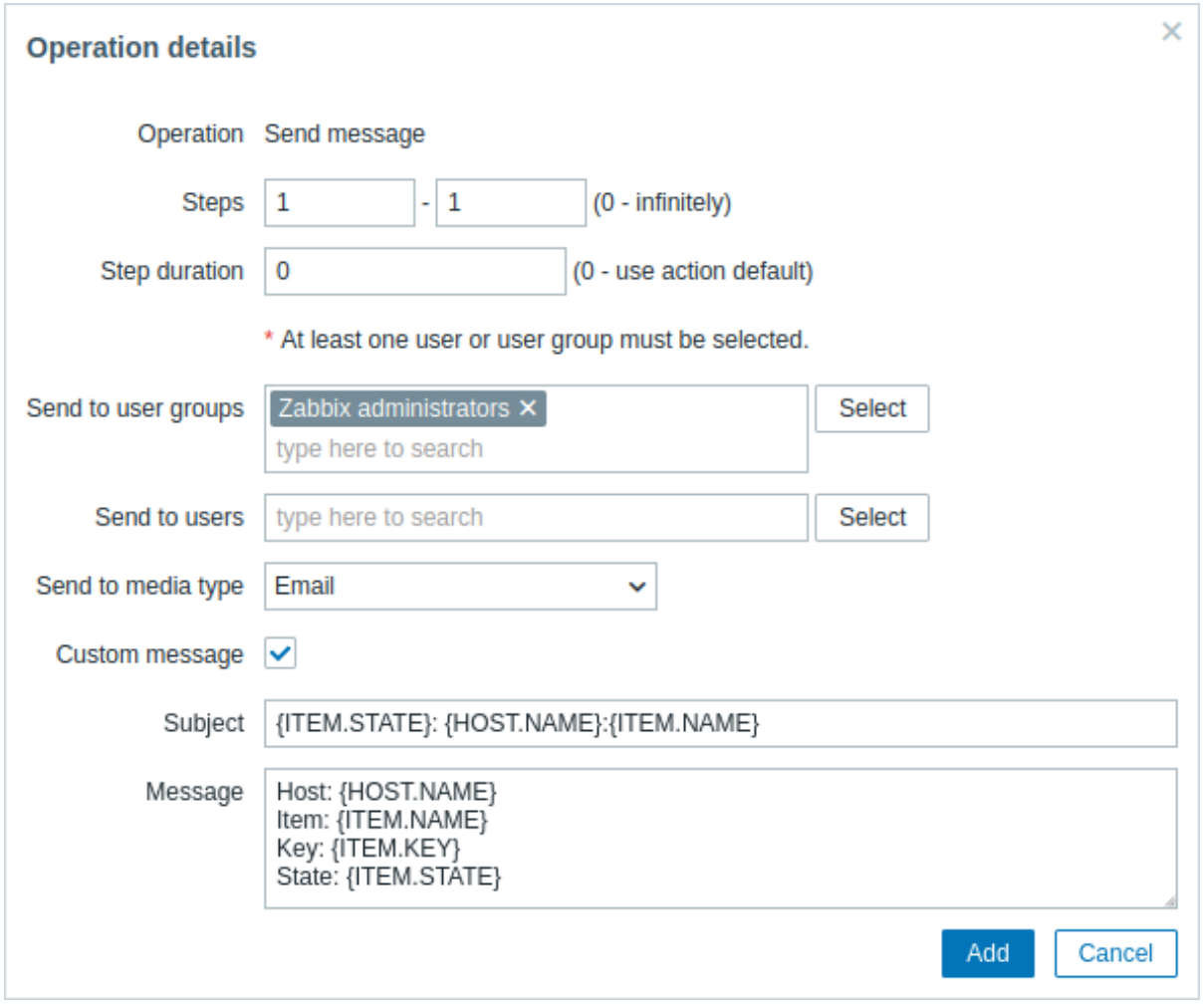
追加をクリックして、条件ブロックに条件を実際にリストすることを忘れないでください。

ステップ 4

オペレーションタブで、オペレーションブロックの 追加をクリックして新しいオペレーションを追加します。



メッセージの受信者 (ユーザーグループ/ユーザー) と、配信に使用するメディアタイプ (または「すべて」) を選択します。問題メッセージのカスタム件名/内容を入力したい場合は、カスタムメッセージチェックボックスをオンにします。



追加をクリックすると、オペレーションブロックにオペレーションが実際にリストされます。

複数の通知を受け取りたい場合は、オペレーションステップの期間 (送信されるメッセージ間の間隔) を設定し、別のステップを追加します。

ステップ 5

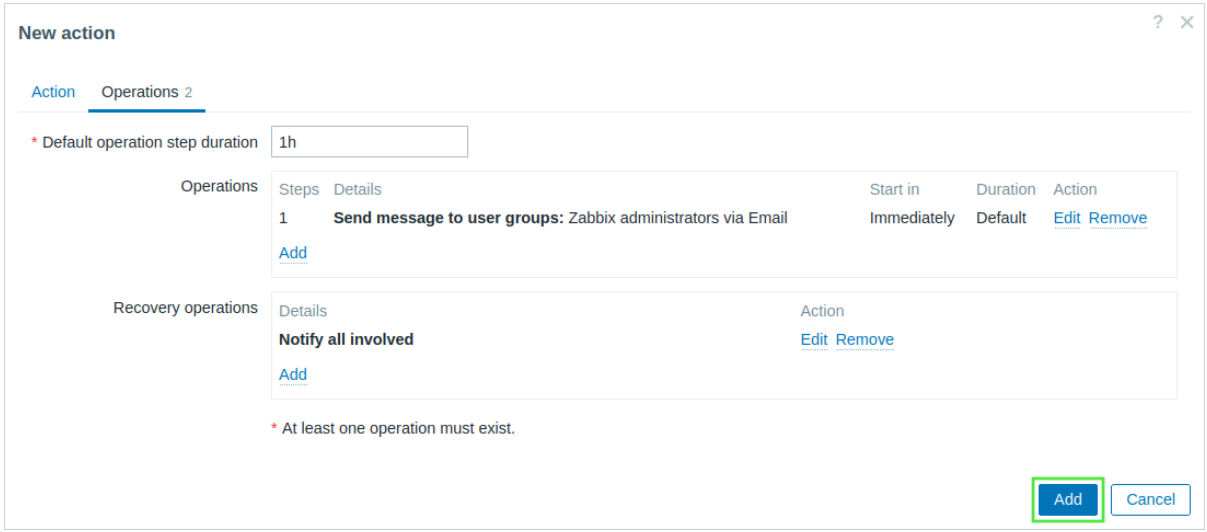
リカバリー操作ブロックでは、アイテムが正常な状態に戻ったときにリカバリ通知を設定できます。リカバリー操作ブロックで追加をクリックして、新しいリカバリー操作を追加します。

操作タイプ「関係者全員に通知」を選択します。問題メッセージのカスタム件名/内容を入力したい場合は、カスタムメッセージチェックボックスを選択します。

操作の詳細ポップアップウィンドウで追加をクリックすると、リカバリー操作ブロックに実際に操作がリストされます。

ステップ 6

完了したら、フォームの下部にある追加ボタンをクリックします。



これで完了です！これで、アイテムが未サポートになった場合に Zabbix から最初の通知を受け取ることができます。

11 マクロ

概要

Zabbix は、さまざまな状況で使用できる多数の組み込みマクロをサポートしています。これらのマクロは変数であり、特定の構文で識別されます:

`{MACRO}`

マクロは、コンテキストに応じて特定の値に解決されます。

マクロを効果的に使用することで、時間を節約し、Zabbix の設定をより透過的にすることができます。

典型的な用途の 1 つとして、テンプレートでマクロを使用することがあげられます。例えばテンプレートのトリガーの名前を”{HOST.NAME} のプロセッサ負荷が高すぎます” とします。テンプレートが Zabbix サーバーなどのホストに適用され、トリガーが監視セクションに表示されると、この名前は”Zabbix サーバーのプロセッサ負荷が高すぎます” に解決されます。

マクロは、アイテムキーパラメーターで使用できます。`item.key[server_{HOST.HOST}_local]` のようにパラメーターの一部としても可能です。解決されたマクロにあいまいな特殊記号存在する場合、Zabbix が処理するのでパラメーターをダブルクオートで囲む必要はありません。

Zabbix には他の種類のマクロもあります。

Zabbix は次のマクロをサポートしています。

- `{MACRO}` - 組み込みマクロ (全リストを参照)
- `{<macro>.<func>(<params>)}` - マクロ関数
- `{$MACRO}` - ユーザー定義のマクロ、オプションでコンテキスト付き
- `{#MACRO}` - ローレベルディスカバリのマクロ
- `{?EXPRESSION}` - 式マクロ

1 マクロ関数

概要

マクロ関数は、マクロの値をカスタマイズする機能を提供します（例えば、特定の部分文字列を短縮または抽出するなど）。これにより、マクロの扱いが容易になります。

マクロ関数の構文は以下の通りです。

```
{macro.func(params)}
```

ここで

- **macro** - カスタマイズするマクロ
- **func** - 適用する関数（サポートされている関数を参照）
- **params** - 関数パラメータのカンマ区切りリスト。以下の場合はダブルクォートで囲む必要があります。
  - 先頭がスペースまたはダブルクォートで始まる場合

- 閉じ括弧またはカンマを含む場合

例：

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

マクロ関数は以下でサポートされています。

- 組み込みマクロ
- ユーザーマクロ
- ローレベルディスカバリーマクロ
- 式マクロ

マクロ関数は、上記のマクロがサポートされているすべての場所で使用できます。これは、明示的にマクロのみが期待されると記載されている場合（例えば、ホストマクロやローレベルディスカバリールールのフィルターを設定する場合）を除きます。

1 つのマクロにつき 1 つの関数のみサポートされており、複数のマクロ関数を連鎖して使用することはできません。

Note:
マクロ関数が他のコンテキスト（関数、アイテムキー、別のマクロなど）内で使用される場合のエスケープ例については、エスケープ例を参照してください。

サポートされている関数

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
btoa	マクロ値を Base64 エンコーディングに変換します。
fmtnum	小数点以下の桁数を制御するための数値フォーマット。
fmttime	時刻のフォーマット。
htmldecode	マクロ値を HTML エンコーディングからデコードします。
htmlencode	マクロ値を HTML エンコーディングに変換します。
iregsub	正規表現（大文字小文字を区別しない）による部分文字列の抽出。
lowercase	マクロ値の文字を小文字に変換します。
regrepl	マクロ値内の文字/部分文字列の置換。
regsub	正規表現（大文字小文字を区別する）による部分文字列の抽出。
tr	マクロ値の文字の変換。
uppercase	マクロ値の文字を大文字に変換します。
urldecode	マクロ値を URL エンコーディングからデコードします。
urlencode	マクロ値を URL エンコーディングに変換します。

関数の詳細

オプションの関数パラメータは < > で示されます。

btoa

マクロ値を Base64 エンコーディングにエンコードします。Base64 エンコーディングは、バイナリデータをテキストとして表現する方法であり、テキストベースのプロトコルでバイナリコンテンツを保存および安全に転送するのに便利です。

例:

```
{{ITEM.VALUE}.btoa()} - "zabbix" のような値を Base64 で"emFiYml4" にエンコードします
```

fmtnum(digits)

小数点以下の桁数を制御するための数値フォーマット。

パラメータ:

- **digits** - 小数点以下の桁数。有効範囲: 0-20。末尾のゼロは出力されません。

例:

```
{{ITEM.VALUE}.fmtnum(2)} - 受信値 "24.3483523" から "24.35" を返します
{{ITEM.VALUE}.fmtnum(0)} - 受信値 "24.3483523" から "24" を返します
```

fmttime(format,<time_shift>)

時刻のフォーマット変換。
この関数は、以下のいずれかの時刻フォーマットの値に解決されるマクロで使用できることに注意してください。

- hh:mm:ss
- yyyy-mm-ddThh:mm:ss[tz] (ISO8601 標準)
- UNIX タイムスタンプ

パラメータ:

- **format** - 必須のフォーマット文字列、strftime 関数のフォーマットに対応;
- **time_shift** (オプション) - フォーマット前に時刻に適用するシフト; -<N><time_unit> または +<N><time_unit> で始まる必要があり、ここで:
  - N - 加算または減算する時間単位の数;
  - time_unit - h (時), d (日), w (週), M (月), y (年)。

コメント:

- time_shift パラメータは複数段階の時刻操作をサポートし、時間単位の先頭にシフトするために/<time_unit> を含めることができます (/d - 深夜, /w - 週の最初の日 (月曜日), /M - 月の最初の日など)。例: -1w - ちょうど 7 日前; -1w/w - 前週の月曜日; -1w/w+1d - 前週の火曜日。
- 時刻操作は優先順位なしで左から右に計算されます。例えば、-1M/d+1h/w は ((-1M/d)+1h)/w として解釈されます。

例:

```
{{TIME}.fmttime(%B)} - "1633098961" という値から"October" を返します
{{TIME}.fmttime(%d %B,-1M/M)} - "1633098961" という値から"1 September" を返します
```

htmldecode

マクロ値を HTML エンコードからデコードします。

以下の文字がサポートされています:

値	デコード後の値
&	&
<	<
>	>
"	"
'	'
'	'

例:

```
{{ITEM.VALUE}.htmldecode()} - "&lt;" のような値を HTML デコードして"<" にします
```

htmlencode

マクロ値を HTML エンコーディングに変換します。

以下の文字がサポートされています:

値	エンコード値
&	& <
>	>
"	"
'	'

例:

```
{{ITEM.VALUE}.htmlencode()} - "<" のような文字を "&lt;" に HTML エンコードします
```

iregsub(pattern,output)

正規表現 (大文字・小文字を区別しない) による部分文字列の抽出。

パラメータ:

- **pattern** - マッチさせる正規表現
- **output** - 出力オプション。**\1 - \9** プレースホルダーはキャプチャグループに対応。**\0** はマッチしたテキストを返します。

コメント:

- 正規表現にマッチしない場合、関数は空文字列を返します。
- 関数パターンが不正な正規表現の場合、マクロは「UNKNOWN」と評価されます (ただし、ローレベルディスカバリーマクロの場合は関数が無視され、マクロは未解決のままになります)。
- 置換文字列内で存在しないキャプチャグループへの参照は空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、"problem" のいずれかの
```

lowercase

すべてのマクロ値の文字を小文字に変換します。シングルバイト文字セット（ASCII など）で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.lowercase()} - "Zabbix SERVER" のような値を"zabbix server"（小文字）に変換します
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

マクロ値内の文字または部分文字列の置換。

パラメータ:

- **pattern** - マッチさせる正規表現
- **replacement** - 置換する文字列 **\1 - \9** のプレースホルダーは、置換文字列内でキャプチャグループとしてサポートされています。

コメント:

- パターンと置換は順番に処理され、各ペアは前の置換の結果に従って適用されます。
- 置換文字列内で存在しないキャプチャグループへの参照は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - "oldParam" を "newParam" に置換します
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - 英字以外のすべての文字をバックスラッシュでエスケープします
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - 末尾のバックスラッシュを削除します（例: "C:\" を "C:" に置換）
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - アイテム値内の複数の部分を置換します
```

regsub(pattern,output)

正規表現マッチによる部分文字列の抽出（大文字と小文字を区別）。

パラメータ:

- **pattern** - マッチさせる正規表現
- **output** - 出力オプション。**\1 - \9** プレースホルダーはキャプチャグループに対応。**\0** はマッチしたテキストを返します。

コメント:

- 正規表現にマッチしない場合、関数は空文字列を返します。
- 関数パターンが不正な正規表現の場合、マクロは'UNKNOWN' に評価されます（ただし、ローレベルディスカバリーマクロの場合は関数が無視され、マクロは未解決のままになります）。
- 置換文字列で存在しないキャプチャグループへの参照は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - "123 Log line" のような値を受信した場合、"Problem ID: 123"
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、"problem" のいずれかの部
```

追加の例も参照してください。

tr(characters,replacement)

マクロ値の文字の音訳。

- **characters** - 置換する文字のセット
- **replacement** - 位置的に対応する置換文字のセット

例:

```
{{ITEM.VALUE}.tr(abc, xyz)} - "a" を"x"、"b" を"y"、"c" を"z" にすべて置換します
{{ITEM.VALUE}.tr(abc, xyzq)} - "a" を"x"、"b" を"y"、"c" を"z" にすべて置換します（"q" は無視されます）
{{ITEM.VALUE}.tr(abcde, xyz)} - "a" を"x"、"b" を"y"、"c" を"z"、"d" を"z"、"e" を"z" にすべて置換します（つまり x
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - バックスラッシュをスラッシュに、シングルクォートをダブルクォートにすべて置換します
{{ITEM.VALUE}.tr(A-Z,a-z)} - すべての文字を小文字に変換します
{{ITEM.VALUE}.tr(0-9a-z,*)} - すべての数字と小文字を"*" に置換します
{{ITEM.VALUE}.tr(0-9,ab)} - 0 を"a" に、1,2,3,4,5,6,7,8,9 を"b" にすべて置換します
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - すべての数字、"abc" 文字、A-L 範囲を"*" に置換します
{{ITEM.VALUE}.tr("\n","*")} - 行末を * に置換します
{{ITEM.VALUE}.tr("e", "\n")} - すべての"e" を行末に置換します
```

リテラル文字を含めるには:

```
バックスラッシュ - \\でエスケープする必要があります
シングルクォート - \'でエスケープする必要があります
ダブルクォート - \" でエスケープする必要があります
```

バックスラッシュによるサポートされるエスケープシーケンス:

```
\\\\ => \\ - ダブルバックスラッシュをシングルバックスラッシュに
\\a  => \a - アラート
\\b  => \b - バックスペース
\\f  => \f - フォームフィード
\\n  => \n - 改行
\\r  => \r - リターン
\\t  => \t - 水平タブ
\\v  => \v - 垂直タブ
```

uppercase

すべてのマクロ値の文字を大文字に変換します。シングルバイト文字セット (ASCII など) で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.uppercase()} - "Zabbix Server" のような値を"ZABBIX SERVER"（大文字）に変換します
```

urldecode

マクロ値を URL エンコードからデコードします。

例:

```
{{ITEM.VALUE}.urldecode()} - "%2F" のような値を"/" に URL デコードします
```

urlencode

マクロ値を URL エンコーディングに変換します。

例:

```
{{ITEM.VALUE}.urlencode()} - "/" のような文字を"%2F" に URL エンコードします
```

追加例

以下の表は、マクロ関数の使用例をさらに示しています。

{#IFALIAS} はLLD マクロであり、低レベルディスカバリのコンテキスト（ディスカバリルール、プロトタイプ、およびそれらから作成されたアイテム/トリガー）でのみ定義されます。LLD 以外で使用すると、トークンは展開されません。

マクロ関数	受信値	出力
{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}	123Log line	Problem
{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}	123 Log line	Problem
{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	Log line	Problem ID:
{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}	MySQL crashed errno 123	Problem ID: MySQL_123
{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}	123 Log line	*UNKNOWN*（無効な正規表現）

マクロ関数	受信値	出力
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}	customername_1	customername
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}	customername_11	
{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}	customername_1	{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}（無効な正規表現）
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"customername"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}	customername_1	{$MACRO:"1"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}（無効な正規表現）
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"	customername_1	"{$MACRO:"\customername\"}"
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \2)}\"}"	customername_1	"{$MACRO:"\1\"}"
"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"	customername_1	"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"（無効な正規表現）

アイテム値全体の表示

テキスト/ログアイテムの解決された {ITEM.VALUE} および {ITEM.LASTVALUE} マクロの長い値は、一部のフロントエンドの場所で20 文字に切り捨てられます。これらのマクロの完全な値を表示するには、以下のようなマクロ関数を使用できます。

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

詳細は、{ITEM.VALUE} および {ITEM.LASTVALUE} のマクロ詳細も参照してください。

2 ユーザーマクロ

概要

Zabbix では、標準でサポートされているマクロに加えて、より柔軟性を持たせるためにユーザーマクロがサポートされています。

ユーザーマクロは、グローバル、テンプレート、ホストレベルで定義できます。これらのマクロには特別な構文があります。

{$MACRO}

Zabbix は、以下の優先順位でマクロを解決します。

1. ホストレベルのマクロ（最初にチェックされる）
2. ホストの第 1 レベルテンプレート（ホストに直接リンクされているテンプレート）で定義されたマクロ（テンプレート ID でソート）
3. ホストの第 2 レベルテンプレートで定義されたマクロ（テンプレート ID でソート）
4. ホストの第 3 レベルテンプレートで定義されたマクロ（テンプレート ID でソート）、以下同様
5. グローバルマクロ（最後にチェックされる）

つまり、ホストにマクロが存在しない場合、Zabbix は深さを増しながらホストテンプレート内でマクロを探します。それでも見つからない場合、存在すればグローバルマクロが使用されます。

Warning:
同じ名前のマクロが同じレベルの複数のリンクされたテンプレートに存在する場合、ID が最も小さいテンプレートのマクロが使用されます。したがって、複数のテンプレートで同じ名前のマクロを持つことは、設定上のリスクとなります。

Zabbix がマクロを見つけられない場合、そのマクロは解決されません。

Attention:
マクロ（ユーザーマクロを含む）は、複雑な設定をより明確にするために、設計上、設定セクション（たとえばトリガーリスト）では未解決のまま残されます。

ユーザーマクロは以下で使用できます。

- アイテム名

- アイテムキーのパラメータ
- アイテムの更新間隔および柔軟な間隔
- トリガー名および説明
- トリガー式のパラメータおよび定数（例を参照）
- その他多くの場所 - 完全なリストを参照

グローバルマクロとホストマクロの一般的な使用例

- グローバルマクロを複数の場所で使用し、マクロの値を変更すると、ワンクリックですべての場所に設定変更が適用されます。
- ホスト固有の属性（パスワード、ポート番号、ファイル名、正規表現など）を持つテンプレートを活用します。

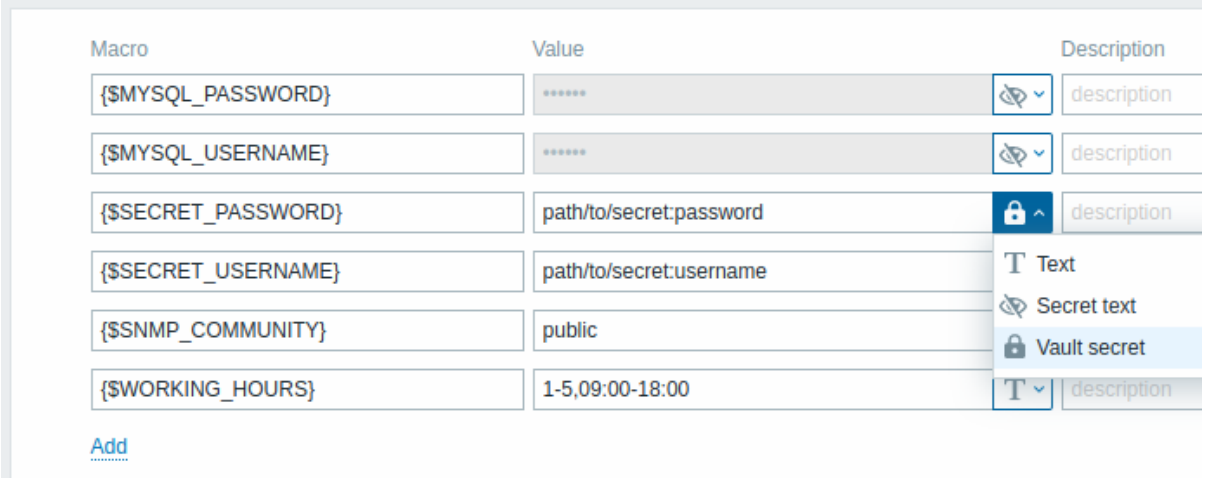
Note:
グローバルマクロの追加、更新、削除はすべてのホストのインクリメンタル設定の更新を強制するため、グローバルマクロよりもホストマクロの使用を推奨します。詳細は、Passive and active agent checksを参照してください。

設定

ユーザーマクロを定義するには、フロントエンドの該当する場所に移動します。

- グローバルマクロの場合は、管理 → マクロにアクセスします
- ホストおよびテンプレートレベルのマクロの場合は、ホストまたはテンプレートのプロパティを開き、マクロタブを探します

ユーザーマクロには以下の属性があります。



パラメータ	説明
マクロ	マクロ名。名前は中括弧で囲み、ドル記号で始める必要があります。 例: {$FRONTEND_URL}。マクロ名で使用できる文字は、**A-Z** (大文字のみ)、**0-9**、_、**.** です。
値	マクロの値。3 つの値タイプがサポートされています: テキスト (デフォルト) - プレーンテキスト値 シークレットテキスト - 値はアスタリスクでマスクされます **Vault** シークレット - 値はVault シークレットへのパス/クエリを含みます。 値タイプを変更するには、値入力フィールドの最後にあるボタンをクリックします。 ユーザーマクロ値の最大長は 2048 文字です。
説明	このマクロに関する追加情報を提供するためのテキストフィールドです。

ユーザーマクロを設定する際は、以下のコンテキスト固有の動作に注意してください。

- テンプレートのアイテムやトリガーでユーザーマクロを使用する場合は、それらのマクロをテンプレートにも追加することを検討してください（グローバルに定義されている場合でも）。そうすることで、テンプレートを XML にエクスポートして別のシステムにインポートした後も、テキストタイプのマクロが期待通りに動作します（シークレットマクロ値はエクスポートされません）
- トリガー式でユーザーマクロを使用する場合、それらのマクロはパラメータまたは定数を参照している場合のみ解決されます。ホスト、アイテムキー、関数、演算子、または他のトリガー式を参照している場合は解決されません（シークレットマクロはトリガー式で使用できません）
- ホストでユーザーマクロを使用し、そのホストにホストプロトタイプを持つ低レベルディスカバリールールがある場合、検出されたホストは、そのホストで定義されたすべてのユーザーマクロを継承します

例

例 1

”Status of SSH daemon” item キーにホストレベルマクロを使用する。

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

この項目は、複数のホストで **{$SSH_PORT}** の値が定義されている場合に、複数のホストに割り当てることができます。

例 2

”CPU load is too high” トリガーでのホストレベルマクロの使用:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

このようなトリガーは、個々のホストで編集するのではなく、テンプレート上で作成されることになります。

Note:
値の量を関数のパラメーターとして使用する場合（例えば, **max(/host/key,#3)**),
 SOME_PERIOD => #3
 というようにマクロ定義にハッシュ記号を入れてください。

例 3

”CPU load is too high” トリガーで 2 つのマクロを使用する:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

マクロはトリガー関数（この例では関数 **min()**）のパラメータとして使用することができることに注意。

例 4

agent の利用不可条件と item の更新間隔を同期させる。:

- {INTERVAL} マクロを定義し、item の更新間隔に使用する。
- {INTERVAL} を agent 使用不可トリガーのパラメータとして使用する。

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

例 5

勤務時間の設定を集中管理する:

- グローバルマクロ {$WORKING_HOURS} を 1-5,09:00-18:00 に設定する
- 管理 → 一般 → *GUI* の勤務時間フィールドで使用する
- ユーザー → ユーザーのメディアタブの有効期間フィールドで使用する
- 勤務時間中により頻繁にアイテムのポーリングを行うために使用する

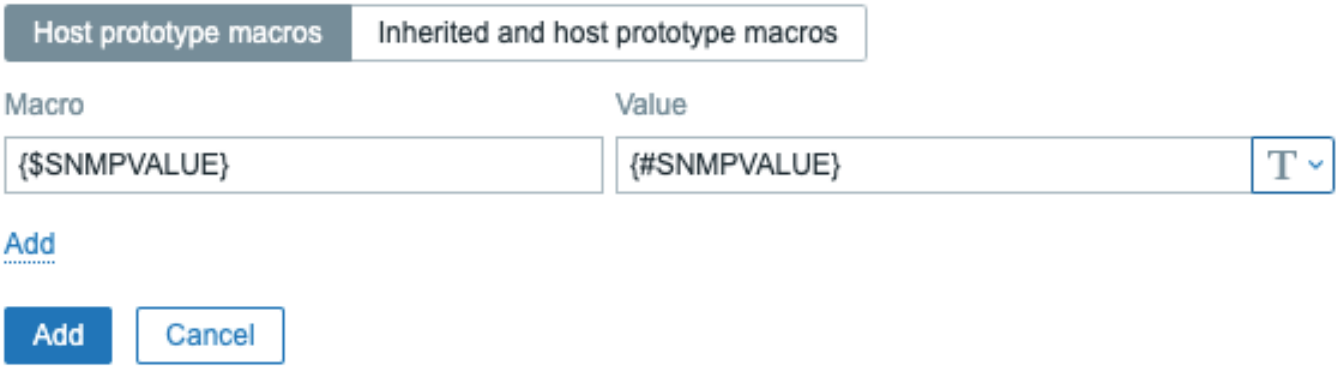


- アクションの期間条件で使用する
- 必要に応じて管理 → マクロで勤務時間を調整する

例 6

ホストプロトタイプマクロを使用して、検出されたホストの item を設定します:

ホストプロトタイプで、{$SNMPVALUE} を {#SNMPVALUE}low-level discovery マクロ値として定義します:



- ホストプロトタイプに *Generic SNMPv2* テンプレートを割り当てます。
- {SNMPVALUE} を使用します。汎用 SNMPv2 テンプレート項目の SNMP OID* フィールドに {$SNMPVALUE} を使用します。

ユーザーマクロのコンテキスト

user macros with context を参照してください。

3 コンテキストユーザーマクロ

概要

ユーザーマクロ でオプションのコンテキストを使用して、デフォルト値をコンテキスト固有の値で上書きすることができます。

マクロ名にコンテキストが追加されます。構文は、コンテキストが静的テキスト値であるかどうかによって異なります。

```
{$MACRO:"static text"}
```

または正規表現:

```
{$MACRO:regex:"regular expression"}
```

正規表現コンテキストを持つマクロは、ユーザーマクロ構成でのみ定義できることに注意してください。`regex:` プレフィックスがトリガー式のようにユーザーマクロコンテキストとして他の場所で使用されている場合、静的コンテキストとして扱われます。

コンテキストの引用はオプションです (重要な注意事項 も参照してください)

マクロコンテキストの例:

例	説明
{$LOW_SPACE_LIMIT}	コンテキストのないユーザーマクロ
{$LOW_SPACE_LIMIT:/tmp}	コンテキストを持つユーザーマクロ (静的文字列)
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	コンテキストを持つユーザーマクロ (正規表現)。{$LOW_SPACE_LIMIT:/tmp} と同じ。
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	コンテキストを持つユーザーマクロ (正規表現)。/var/log/ で始まるすべての文字列に一致。

ユースケース

コンテキストを持つユーザーマクロを定義して、トリガー式で (ローレベルディスカバリによって取得された値に基づく) より柔軟なしきい値を実現できます。たとえば、次のマクロを定義できます。

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:"^/[a-z]+$"} = 30

次に、ローレベルディスカバリマクロを、マウントされたファイルシステムディスカバリのトリガープロトタイプでマクロコンテキストとして使用できます:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

検出後、検出されたマウントポイントまたはファイルシステムの種類に応じて、さまざまな低容量しきい値がトリガーに適用されます。障害イベントは、次の場合に生成されます。

- /home フォルダーの空きディスク容量が 20% 未満
- 正規表現パターン (/etc、/tmp、/var など) に一致するフォルダーの空きディスク容量が 30% 未満
- 正規表現パターンに一致せず、/home ではないフォルダーの空きディスク容量が 10% 未満

重要事項

- コンテキストを持つ複数のユーザーマクロが存在する場合、Zabbix は最初に単純なコンテキストマクロを照合し、次にコンテキストマクロを未定義の順序で正規表現と照合しようとします。

Warning:
未定義の動作を避けるために、同じ文字列に一致する別のコンテキストマクロを作成しないでください。

- コンテキストを持つマクロがホスト、リンクされたテンプレート、またはグローバルで見つからない場合、コンテキストのないマクロが検索されます。
- コンテキストでは、ローレベルディスカバリマクロのみがサポートされています。その他のマクロはすべて無視され、プレーンテキストとして扱われます。

技術的には、マクロコンテキストはitem key パラメータと同様のルールを使用して指定されますが、, 文字がある場合、マクロコンテキストは複数のパラメータとして解析されません。

- コンテキストに } 文字が含まれている場合、または " 文字で始まる場合、マクロコンテキストは " で引用する必要があります。引用されたコンテキスト内の引用符は、\ 文字でエスケープする必要があります。
- \ 文字自体はエスケープされません。つまり、\ 文字で終わる引用符付きコンテキストを持つことは不可能です - マクロ {$MACRO:"a:\b\c\"} は無効です。
- コンテキスト内の先頭のスペースは無視されますが、末尾のスペースは無視されます:
  - たとえば、{$MACRO:A} は {$MACRO: A} と同じですが、{$MACRO:A } とは異なります。
- 先頭の引用符の前と末尾の引用符の後のすべてのスペースは無視されますが、引用符内のすべてのスペースは無視されません。
  - マクロ {$MACRO:"A"}、{$MACRO: "A"}、{$MACRO:"A" }、および {$MACRO: "A" } は同じですが、マクロ {$MACRO:"A"} および {$MACRO:" A "} ではありません。

{$MACRO:A}、{$MACRO: A}、および {$MACRO:"A"} のマクロは、同じコンテキストを持っているため、すべて同等です。これは、'key[a]'、'key[a]'、および'key["a"]' が意味的に同じである項目キーとは対照的ですが、一意性のために異なります。

4 Secret ユーザーマクロ

概要

Zabbix は、ユーザーマクロ値の機密情報を保護するために 2 つのオプションを提供しています。

- シークレットテキスト
- Vault シークレット

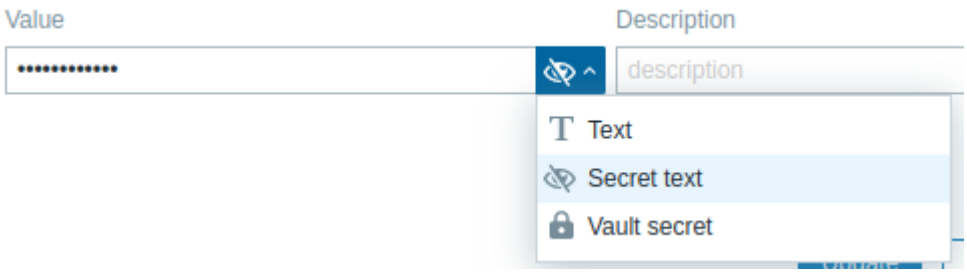
Note:
シークレットマクロの値は非表示ですが、アイテムで使用することで明らかになる場合があります。たとえば、外部スクリプトでシークレットマクロを参照する echo 文を使用すると、Zabbix サーバーが実際のマクロ値にアクセスできるため、マクロ値がフロントエンドに表示される場合があります。シークレットマクロ値がマスク解除される場所を参照してください。

シークレットマクロはトリガー式では使用できません。

シークレットテキスト

シークレットテキストマクロでは、マクロ値がアスタリスクでマスクされます。

マクロ値をシークレットにするには、値フィールドの末尾にあるボタンをクリックし、シークレットテキストオプションを選択します。



設定を保存すると、値を表示することはできなくなります。

マクロ値を変更するには、値フィールドにカーソルを合わせ、新しい値を設定ボタン（ホバー時に表示）をクリックします。



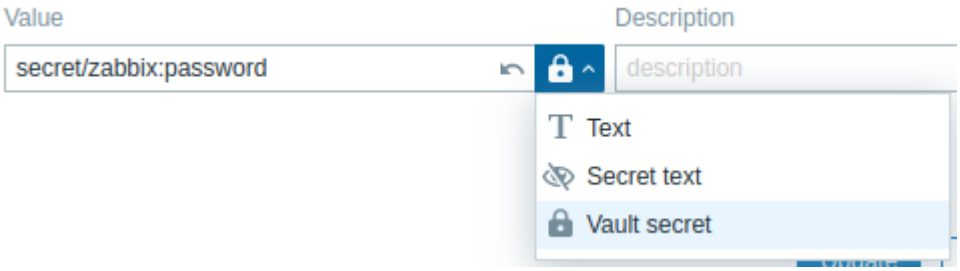
新しい値を設定ボタンをクリックする（またはマクロ値のタイプを変更する）と、現在の値は消去されます。値フィールドの末尾にある 矢印をクリックすると、元の値を復元できます（新しい設定を保存する前のみ利用可能）。元の値を復元しても、その値が表示されることはありません。

Note:
シークレットマクロを含む URL は機能しません。なぜなら、URL 内のマクロは「******」として解決されるためです。

Vault シークレット

Vault シークレットマクロでは、マクロの値が外部のシークレット管理ソフトウェア（vault）に保存されます。

Vault シークレットマクロを設定するには、値フィールドの末尾にあるボタンをクリックし、*Vault* シークレットオプションを選択します。



マクロの値は vault シークレットを指す必要があります。入力フォーマットは vault プロバイダーによって異なります。プロバイダー固有の設定例については、以下を参照してください。

- HashiCorp
- CyberArk

Vault シークレットマクロの値は、Zabbix サーバー（およびシークレット *Vault* マクロの解決方法がZabbix サーバーとプロキシに設定されている場合は Zabbix プロキシ）によって、構成データのリフレッシュごとに Vault から取得され、その後構成キャッシュに保存されます。Zabbix サーバーと Zabbix プロキシは異なる Vault を使用することができます。

シークレット *Vault* マクロの解決方法がZabbix サーバーに設定されている場合、Vault シークレットはサーバーのみが取得し、Zabbix プロキシは各構成同期時に Zabbix サーバーから Vault シークレットマクロの値を受け取り、自身の構成キャッシュに保存します。つまり、Zabbix プロキシは、Zabbix サーバーから構成の更新を受け取るまで、再起動後にデータ収集を開始できません。

Vault からシークレット値を手動でリフレッシュするには、`secrets_reload` ランタイムコントロールオプション（サーバーのみ）を使用します。

Zabbix サーバーとプロキシ間で暗号化を有効にする必要があります。有効でない場合、サーバーの警告メッセージが記録されます。

Warning:
マクロ値を正常に取得できない場合、その値を使用している対応するアイテムはサポート外になります。

マスク解除される場所

このリストは、シークレットマクロ値がマスク解除されるパラメータの場所を示します。

Note:
シークレットマクロ値が間接的に参照された場合、下記の場所でもマスクされたままになります。たとえば、メディアタイプ（スクリプトまたは Webhook パラメータ）で使用される {ITEM.KEY}、{ITEM.KEY<1-9>}、{LLDRULE.KEY} 組み込みマクロは、`net.tcp.port[192.0.2.0,80]` の代わりに `net.tcp.port[******,******]` のようなマスクされたシークレットマクロを含むアイテムキーに解決されます。

コンテキスト		パラメータ
アイテム、アイテムプロトタイプ、**LLD** ルール		
	アイテム	アイテムキーのパラメータ
	アイテムプロトタイプ	アイテムプロトタイプキーのパラメータ
	ローレベルディスカバリルール	ディスカバリアイテムキーのパラメータ
	SNMP エージェント	*SNMP* コミュニティ
		コンテキスト名 (SNMPv3)
		セキュリティ名 (SNMPv3)
		認証パスフレーズ (SNMPv3)
		プライバシーパスフレーズ (SNMPv3)

コンテキスト		パラメータ
	HTTP エージェント	*URL* クエリフィールド *Post* ヘッダー ユーザー名 パスワード *SSL* キーのパスワード
	スクリプト	パラメータ スクリプト
	ブラウザ	パラメータ スクリプト
	データベースモニタ	*SQL* クエリ
	TELNET エージェント	スクリプト ユーザー名 パスワード
	SSH エージェント	スクリプト ユーザー名 パスワード
	シンプルチェック	ユーザー名 パスワード
	JMX エージェント	ユーザー名 パスワード
アイテム値の事前処理		
	JavaScript 事前処理ステップ	スクリプト
Web シナリオ		
	Web シナリオ	変数値 ヘッダー値 *URL* クエリフィールド値 *Post* フィールド値 *Raw post*
	Web シナリオ認証	ユーザー パスワード *SSL* キーのパスワード
コネクタ		
	コネクタ	*URL* ユーザー名 パスワード トークン *HTTP* プロキシ *SSL* 証明書ファイル *SSL* キーのファイル *SSL* キーのパスワード
ネットワークディスカバリ		
	SNMP	*SNMP* コミュニティ コンテキスト名 (SNMPv3) セキュリティ名 (SNMPv3) 認証パスフレーズ (SNMPv3) プライバシーパスフレーズ (SNMPv3)
グローバルスクリプト		
	Webhook	*JavaScript* スクリプト *JavaScript* スクリプトパラメータ値
	Telnet	ユーザー名 パスワード
	SSH	ユーザー名 パスワード
	スクリプト	スクリプト
メディアタイプ		
	スクリプト	スクリプトパラメータ
	Webhook	パラメータ
IPMI 管理		

コンテキスト	パラメータ
ホスト	ユーザー名 パスワード

5 ローレベルディスカバリマクロ

概要

ローレベルディスカバリ（LLD）機能内で使用されるマクロの種類があります。

`{#MACRO}`

これは LLD ルールで使用され、ファイルシステム名、ネットワークインターフェース、SNMP OID などの実際の値を返すマクロです。

これらのマクロは、アイテム、トリガー、グラフ、および LLD ルールのプロトタイプを作成するために使用できます。その後、実際のファイルシステムやネットワークインターフェースなどをディスカバリする際に、これらのマクロは実際の値に置き換えられ、実体を作成する基礎となります。

これらのマクロは、仮想マシンディスカバリにおけるホストおよびホストグループのプロトタイプ作成にも使用されます。

一部のローレベルディスカバリマクロは、Zabbix の LLD 機能に「プリパッケージ」されています - {#FSNAME}、{#FSTYPE}、{#IFNAME}、{#SNMPINDEX}、{#SNMPVALUE}。ただし、カスタムローレベルディスカバリルールを作成する際に、これらの名前に従う必要はありません。その場合は、他の任意の LLD マクロ名を使用し、その名前を参照することができます。

サポートされているデータ型

カスタムディスカバリールールを定義する際、LLD マクロ用の JSON オブジェクトで返されるプロパティ値は、以下のいずれかのプリミティブ型でなければなりません。

- string
- number
- boolean

配列、オブジェクト、null 値はサポートされていません。そのような値を参照する LLD マクロは展開されず、アイテムの事前処理や作成時にリテラル（例：`'{#MY_MACRO}'`）として表示されます。

サポートされている場所

LLD マクロは以下で使用できます:

- 低レベルディスカバリールールのフィルター
- ネストされた低レベルディスカバリールール内の
  - JSONPath の前処理パラメータ
  - カスタム LLD マクロ用の JSONPath フィールド
- アイテムプロトタイプおよびディスカバリープロトタイプの
  - 名前
  - キーパラメータ
  - 単位
  - 更新間隔[1]
  - タイムアウト[1]
  - ヒストリ保存期間[1]
  - トレンド保存期間[1]
  - アイテム値の前処理ステップ
  - SNMP OID
  - IPMI センサーフィールド
  - 計算/集約アイテム式の
    - 式の定数および関数パラメータ
    - アイテムキーパラメータ
  - 集約アイテムのフィルター条件（ホストグループ名およびタグ名）
  - SSH スクリプトおよび Telnet スクリプト
  - データベース監視 SQL クエリ
  - JMX アイテムのエンドポイントフィールド
  - 説明
  - HTTP エージェントの URL フィールド
  - HTTP エージェントの HTTP クエリフィールド
  - HTTP エージェントのリクエストボディフィールド
  - HTTP エージェントの必要なステータスコードフィールド
  - HTTP エージェントのヘッダーフィールドのキーと値

    - HTTP エージェントの HTTP 認証ユーザ名フィールド
    - HTTP エージェントの HTTP 認証パスワードフィールド
    - HTTP エージェントの HTTP プロキシフィールド
    - HTTP エージェントの HTTP SSL 証明書ファイルフィールド
    - HTTP エージェントの HTTP SSL キーのファイルフィールド
    - HTTP エージェントの HTTP SSL キーのパスワードフィールド
    - タグ
- トリガープロトタイプの
    - 名前
    - 運用データ
    - 式（定数および関数パラメータのみ）
    - URL
    - 説明
    - タグ
- グラフプロトタイプの
    - 名前
- ホストプロトタイプの
    - 名前
    - 表示名
    - カスタムインターフェースフィールド: IP、DNS、ポート、SNMP v1/v2 コミュニティ、SNMP v3 コンテキスト名、SNMP v3 セキュリティ名、SNMP v3 認証パスフレーズ、SNMP v3 プライバシーパスフレーズ
    - ホストグループプロトタイプ名
    - ホストタグ値
    - ホストマクロ値
    - (完全なリストを参照)

これらの場所すべてで、低レベルディスカバリールールのフィルターを除き、LLD マクロは静的ユーザーマクロコンテキスト内で使用できます。

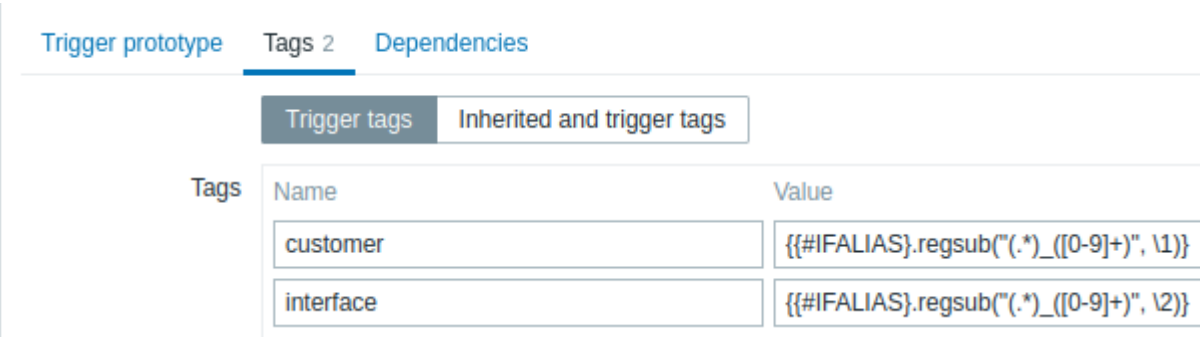
マクロ関数の使用

マクロ関数は、ローレベルディスカバリマクロでサポートされています（ただし、ローレベルディスカバリルールのフィルタではサポートされていません）。これにより、正規表現を使用してマクロ値の特定の部分を抽出できます。

たとえば、イベントタグ付けの目的で、次の LLD マクロから顧客名とインターフェース番号を抽出したい場合があります。

```
{#IFALIAS}=customername_1
```

この場合、regsub マクロ関数をトリガープロトタイプのイベントタグ値フィールドでマクロとともに使用できます。



カンマは引用符で囲まれていないアイテムのキーのパラメータでは使用できないため、マクロ関数を含むパラメータは引用符で囲む必要があります。パラメータ内で二重引用符をエスケープするには、バックスラッシュ（\）文字を使用してください。例：

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

マクロ関数の構文の詳細については、マクロ関数を参照してください。

マクロ関数は Zabbix 4.0 以降、ローレベルディスカバリマクロでサポートされています。

脚注

[1] [1] でマークされたフィールドでは、1 つのマクロでフィールド全体を埋める必要があります。
 フィールド内の複数のマクロや、テキストと混在するマクロはサポートされていません。

6 式マクロ

概要

式マクロを使用すると、フィールド内で計算を実行できます。

その値は、最初に内部マクロを解決し、その後、得られた式を評価することで計算されます。

構文:

```
{?EXPRESSION}
```

EXPRESSION は、トリガー式と同じ構文を使用し、同じ関数をサポートします。

例:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

サポートされている場所については、サポートされているマクロを参照してください。

使用上の注意:

- トリガー式/リカバリー式の関数値を参照するには、{FUNCTION.*}マクロを使用します。
- ホストやアイテムを参照するには、{HOST.HOST<1-9>}や{ITEM.KEY<1-9>}マクロを使用します。
- テンプレートでは、テンプレート名はテンプレートのリンク時にホスト名に置き換えられないため、最初のホストには{HOST.HOST<1-9>}マクロを使用するか、ホストを省略します（例: {?avg(//item1,1h)}）。

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

イベント名で式マクロを使用する例については、トリガー式の例も参照してください。

12 ユーザーとユーザーグループ

概要

Zabbix のすべてのユーザは、Web ベースのフロントエンドから Zabbix アプリケーションにアクセスします。各ユーザーには
 固有のログイン名とパスワードが割り当てられます。

全てのユーザパスワードは暗号化され、Zabbix のデータベースに保存されます。ユーザーは、UNIX に応じた設定をしていない限り、
 ユーザー ID とパスワードを使用して UNIX サーバに直接ログインすることはできません。Web サーバとユーザブラウザ間の通信は SSL を使用して
 保護することができます。

柔軟なuser permission schemaにより、以下の権限を制限し差別化することができます:

- Zabbix フロントエンドの管理用機能へのアクセス
- フロントエンドでの特定のアクションの実行
- ホストグループ内の監視対象ホストへのアクセス
- 特定の API メソッドの使用

1 ユーザの設定

概要

初期の Zabbix インストールには、2 つの事前定義されたユーザーが含まれています。

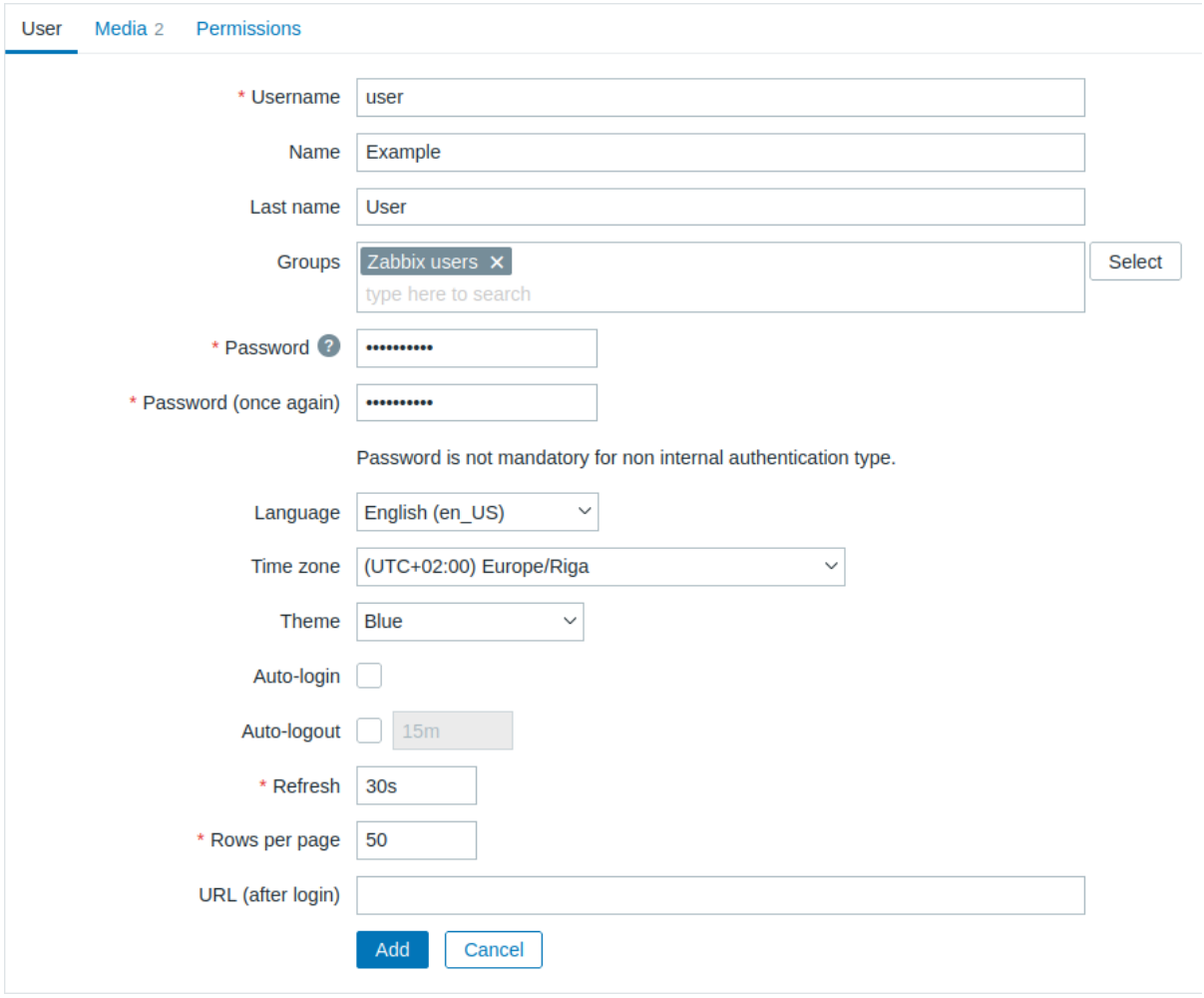
- *Admin* - Zabbix のスーパーユーザー で、すべての権限を持っています。
- *guest* - 特別な Zabbix ユーザー です。'guest' ユーザーはデフォルトで無効になっています。これを Guests ユーザーグループに追加すると、このユーザーでログインし、Zabbix の監視ページにアクセスできます。ただし、デフォルトでは'guest' には Zabbix オブジェクトへの権限はありません。

ユーザーを設定するには:

- ユーザー → ユーザーに移動します。
- ユーザーの作成（または既存ユーザーを編集する場合はユーザー名）をクリックします。
- フォームでユーザー属性を編集します。

一般属性

ユーザータブには、一般的なユーザー属性が含まれています。



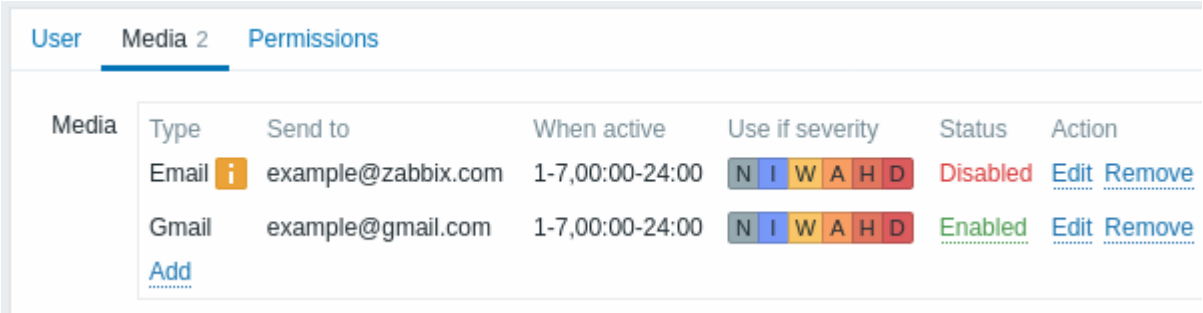
すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
ユーザー名	ログイン名として使用される一意のユーザー名。
名	ユーザーの名（省略可能）。 空でない場合、承認情報や通知受信者情報に表示されます。
姓	ユーザーの姓（省略可能）。 空でない場合、承認情報や通知受信者情報に表示されます。
グループ	ユーザーが所属するユーザーグループを選択します。このフィールドはオートコンプリート対応なので、ユーザーグループ名の入力を始めると一致するグループのドロップダウンが表示されます。スクロールして選択します。あるいは、選択をクリックしてグループを追加します。選択したグループを削除するには「x」をクリックします。 ユーザーグループへの所属により、ユーザーがアクセスできるホストグループやホストが決まります。
パスワード	ユーザーパスワードを入力する 2 つのフィールド、または既存ユーザーの場合はパスワード変更ボタン。 パスワード変更ボタンをクリックすると、新しいパスワードを入力する 2 つのフィールドが表示されます。 スーパ管理者ロールを持つユーザーが自分のパスワードを変更する場合、パスワード変更ボタンをクリックすると、現在の（古い）パスワードを入力する追加フィールドが表示されます。 パスワードの変更が成功すると、パスワードが変更されたユーザーはすべてのアクティブセッションからログアウトされます。 パスワードは、Zabbix の内部認証を使用しているユーザーのみ変更できます。
言語	Zabbix フロントエンドの言語。 翻訳を有効にするには php の gettext 拡張が必要です。
タイムゾーン	ユーザーレベルでグローバルなタイムゾーンを上書きするタイムゾーンを選択するか、システムデフォルトを選択してグローバルなタイムゾーン設定を使用します。

パラメータ	説明
テーマ	フロントエンドの見た目を定義します： システムデフォルト - デフォルトのシステム設定を使用 ブルー - 標準のブルーテーマ ダーク - 代替のダークテーマ ハイコントラスト（ライト） - ハイコントラストのライトテーマ ハイコントラスト（ダーク） - ハイコントラストのダークテーマ
自動ログイン	このチェックボックスをオンにすると、Zabbix がユーザーを記憶し、30 日間自動的にログインします。 *30* 日間記憶するでログインした場合： - 自動ログアウトがリセットされます（セッションは 30 日間持続）。 - シームレスな再認証のために自動ログインが有効になります。 *30* 日間記憶するを使わずにログインした場合： - シームレスな再認証のために自動ログインが有効になります。 - 自動ログアウトは標準のタイムアウト設定に従います。 この機能にはブラウザのクッキーが使用されます。
自動ログアウト	このチェックボックスをオンにすると、設定した秒数（最小 90 秒、最大 1 日）後に自動的にログアウトされます。 この設定は、*30* 日間記憶するが有効な場合は上書きされ、セッションは全期間延長されます。 時間のサフィックスがサポートされています（例：90s、5m、2h、1d）。 このオプションは以下の場合は動作しません： * グローバル設定「Zabbix サーバーがダウンしている場合に警告を表示する」が有効で、Zabbix フロントエンドを開いたままにしている場合。 * 監視メニューのページでバックグラウンド情報の更新が行われている場合。 * *30* 日間記憶するオプションをチェックしてログインした場合。
リフレッシュ	グラフやプレーンテキストデータなどで使用するリフレッシュレートを設定します。0 に設定すると無効になります。 時間のサフィックスがサポートされています（例：90s、5m、1h）。
1 ページあたりの行数	リストで 1 ページあたりに表示する行数を指定できます。
URL（ログイン後）	ログイン後にユーザーを特定の URL に転送することができます。例えば障害ページなど。

ユーザーメディア

メディアタブには、ユーザーに定義されているすべてのメディアの一覧が表示されます。メディアは通知の送信に使用されます。



追加をクリックして、ユーザーにメディアを割り当てます。

メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に無効と表示されます。

Note:
ユーザー自身のメディア詳細を変更する権限は、ユーザーロール（自身のメディアの作成と編集オプションを参照）に基づいて付与または取り消すことができます。他のユーザーのメディア詳細を変更するスーパ管理者の権限も、ユーザーロール（ユーザーメディアの作成と編集オプションを参照）に基づいて付与または取り消すことができます。

ユーザーメディアの設定の詳細については、メディアタイプのセクションを参照してください。

権限

権限タブには、以下の要素に関する情報が含まれています。

- ユーザーロール（新規作成ユーザーには必須）。スーパ管理者タイプのユーザーのみが変更できます。

Attention:
ユーザーロールがないとユーザーは作成できません（Zabbix User APIを除く）。以前に作成されたロールのないユーザーは、ロールを割り当てずに編集できます。ただし、一度ロールを割り当てると、変更はできますが削除はできません。

ロールのないユーザーは、LDAPまたはSAML認証を使用してのみ Zabbix にログインできます（LDAP/SAML 情報が Zabbix で設定されたユーザーグループのマッピングと一致する場合）。

- ユーザータイプ（ユーザー、管理者、スーパ管理者）。ユーザーロールの設定で定義されます。
- ユーザーがアクセスできるホストグループとテンプレートグループ。
  - ユーザーおよび 管理者タイプのユーザーは、デフォルトではグループ、テンプレート、ホストへのアクセス権がありません。アクセス権を付与するには、該当するエンティティへの権限を持つユーザーグループにユーザーを含める必要があります。
- Zabbix フロントエンド、モジュール、API メソッドのセクションや要素へのアクセス権。
  - アクセスが許可されている要素は緑色、拒否されている要素は薄いグレーで表示されます。
- 特定のアクションを実行する権限。
  - ユーザーが実行できるアクションは緑色、拒否されているアクションは薄いグレーで表示されます。

詳細は権限ページを参照してください。

2 ユーザー権限

概要

Zabbix の権限は、ユーザータイプ、カスタマイズされたユーザーロール、およびユーザーグループに基づいて指定されるホストへのアクセスに依存します。

ユーザータイプ

Zabbix の権限は、主にユーザータイプによって決まります。

- ユーザー - メニューセクションへのアクセス権が制限されており（下記参照）、デフォルトではいかなるリソースにもアクセスできません。ホストまたはテンプレートグループへの権限は明示的に割り当てる必要があります。
- 管理者 - メニューセクションへのアクセス権が一部制限されています（下記参照）。デフォルトではいかなるホストグループにもアクセスできません。ホストまたはテンプレートグループへの権限は明示的に付与する必要があります。
- スーパ管理者 - すべてのメニューセクションにアクセスできます。すべてのホストおよびテンプレートグループに対して読み書き権限を持ちます。特定のグループへのアクセスを拒否しても権限を取り消すことはできません。

メニューアクセス

Note:
一部の UI 要素へのアクセス制限は、そのページの表示を防ぐだけであり、インターフェースの他の部分で基礎データへのアクセスができなくなるわけではありません。

以下の表は、ユーザータイプごとの Zabbix メニューセクションへのアクセスを示しています。

メニューセクション		ユーザー	管理者	スーパ管理者
ダッシュボード		+	+	+
監視		+	+	+
	障害	+	+	+
	ホスト	+	+	+
	最新データ	+	+	+
	マップ	+	+	+
	ディスカバリ		+	+
サービス		+	+	+
	サービス	+	+	+
	SLA		+	+
	SLA レポート	+	+	+
インベントリ		+	+	+
	概要	+	+	+
	ホスト	+	+	+
レポート		+	+	+
	システム情報			+
	スケジュールレポート		+	+
	可用性レポート	+	+	+
	トップ *100* トリガー	+	+	+

メニューセクション		ユーザー	管理者	スーパ管理者
	監査ログ			+
	アクションログ			+
	通知		+	+
データ収集			+	+
	テンプレートグループ		+	+
	ホストグループ		+	+
	テンプレート		+	+
	ホスト		+	+
	メンテナンス		+	+
	イベント相関			+
	ディスカバリ		+	+
アラート			+	+
	トリガーアクション		+	+
	サービスアクション		+	+
	ディスカバリアクション		+	+
	自動登録アクション		+	+
	内部アクション		+	+
	メディアタイプ			+
	スクリプト			+
ユーザー				+
	ユーザーグループ			+
	ユーザーロール			+
	ユーザー			+
	API トークン			+
	認証			+
管理				+
	一般			+
	監査ログ			+
	ハウスキーピング			+
	プロキシグループ			+
	プロキシ			+
	マクロ			+
	キュー			+

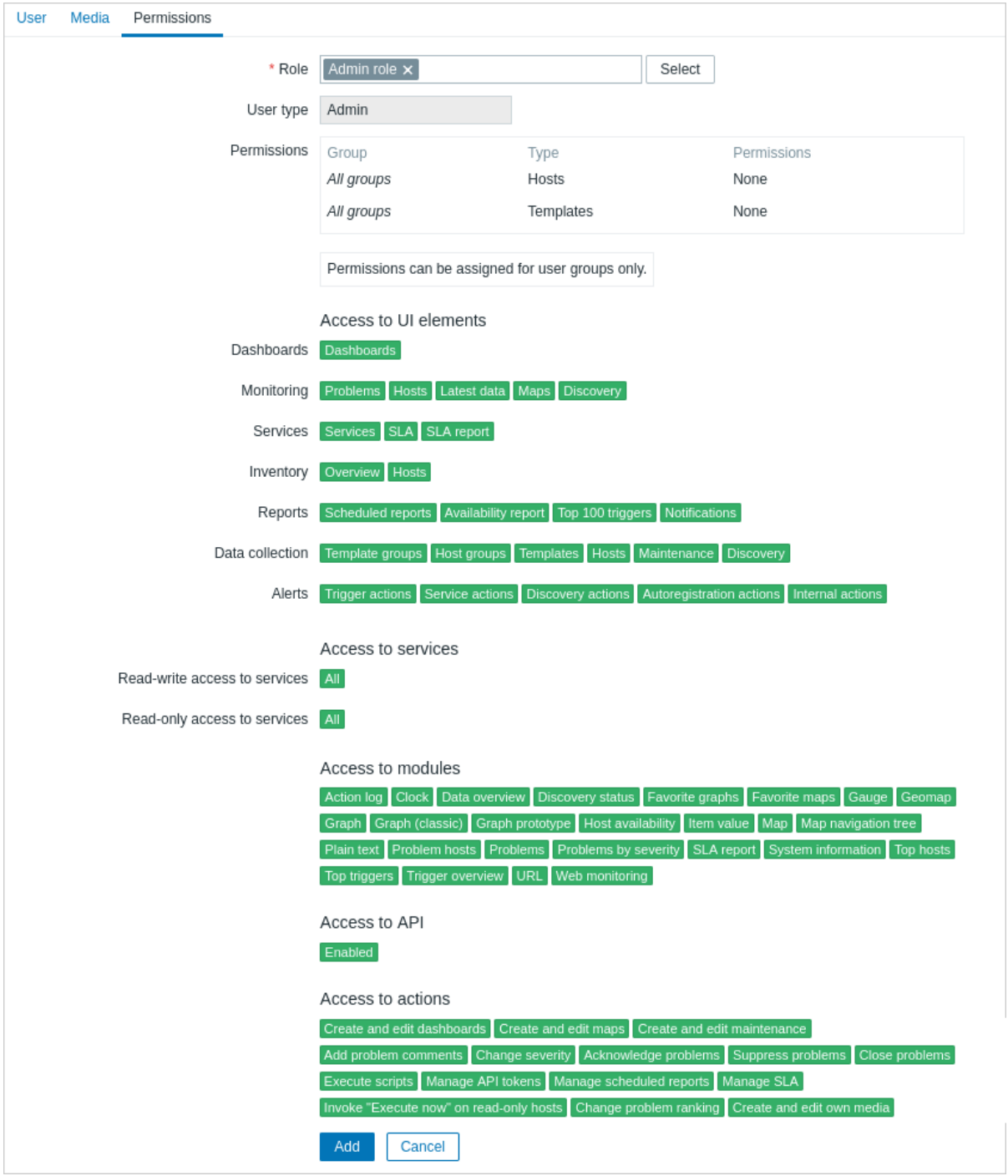
ユーザーロール

ユーザーロールを使用すると、ユーザータイプで定義された権限をカスタマイズできます。(ユーザータイプの権限を超える）権限を追加することはできませんが、一部の権限を取り消すことができます。

さらに、ユーザーロールはメニューセクションへのアクセスだけでなく、サービス、モジュール、API メソッド、およびフロントエンドでのさまざまなアクションへのアクセスも決定します。

ユーザーロール は、ユーザー → ユーザーロールセクションでスーパ管理者ユーザーによって設定されます。

ユーザーロールは、スーパ管理者ユーザーがユーザー設定フォームの 権限タブでユーザーに割り当てます。



ホストへのアクセス

Zabbix でのホストやテンプレートデータへのアクセスは、ホスト/テンプレートグループレベルでユーザーグループにのみ付与されます。

つまり、個々のユーザーに直接ホスト（またはホストグループ）へのアクセス権を付与することはできません。ホストを含むホストグループへのアクセス権が付与されたユーザーグループのメンバーであることによってのみ、ホストへのアクセス権が付与されます。

同様に、テンプレートを含むテンプレートグループへのアクセス権が付与されたユーザーグループのメンバーであることによってのみ、テンプレートへのアクセス権が付与されます。

3 ユーザグループ

概要

ユーザーグループは、組織的な目的とデータへの権限割り当ての両方のためにユーザーをグループ化することができます。ホストグループおよびテンプレートグループのデータの表示および設定の権限は、個々のユーザーではなくユーザーグループに割り当てられます。

どのユーザーグループにどの情報を利用可能にするかを分けることは、しばしば理にかなっています。これは、ユーザーをグループ

化し、ホストグループやテンプレートグループに異なる権限を割り当てることで実現できます。

ユーザーは任意の数のグループに所属できます。

設定

ユーザーグループを設定するには:

- ユーザー → ユーザーグループに移動します
- ユーザーグループの作成をクリックします（または既存のグループを編集する場合はグループ名をクリックします）
- フォームでグループ属性を編集します

ユーザーグループタブには、一般的なグループ属性が含まれています:

すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
グループ名	一意のグループ名。
ユーザー	グループにユーザーを追加するには、既存のユーザー名の入力を開始します。一致するユーザー名のドロップダウンが表示されたら、スクロールして選択します。 または、選択ボタンをクリックしてポップアップでユーザーを選択することもできます。
フロントエンドアクセス	グループのユーザーがどのように認証されるか。 システムデフォルト - デフォルトの認証方法を使用（グローバルで設定） 内部 - Zabbix 内部認証を使用（グローバルで LDAP 認証が使用されていても）。 グローバルデフォルトが HTTP 認証の場合は無視されます。 **LDAP** - LDAP 認証を使用（グローバルで内部認証が使用されていても）。 グローバルデフォルトが HTTP 認証の場合は無視されます。 無効 - このグループの Zabbix フロントエンドへのアクセスは禁止されます
LDAP サーバー	ユーザーの認証に使用するLDAP サーバーを選択します。 このフィールドは、フロントエンドアクセスが LDAP またはシステムデフォルトに設定されている場合のみ有効です。
多要素認証	ユーザーの認証に使用する多要素認証方法を選択します: デフォルト - MFA 設定でデフォルトとして設定された方法を使用します。このオプションは、MFA が有効な場合、新しいユーザーグループのデフォルトで選択されます。 **<** 方法名 **>** - 選択した方法を使用します（例: ”Zabbix TOTP”）。 無効 - このグループでは MFA が無効です。このオプションは、MFA が無効な場合、新しいユーザーグループのデフォルトで選択されます。 ユーザーが MFA が有効な複数のユーザーグループに属している場合（または少なくとも 1 つのグループで MFA が有効な場合）、次の認証ルールが適用されます: いずれかのグループが「デフォルト」MFA 方法を使用している場合、それがユーザーの認証に使用されます。それ以外の場合は、最初の方法（アルファベット順）が認証に使用されます。
有効	ユーザーグループおよびグループメンバーのステータス。 チェックあり - ユーザーグループとユーザーが有効 チェックなし - ユーザーグループとユーザーが無効
デバッグモード	このチェックボックスをマークすると、ユーザーのデバッグモードが有効になります。

テンプレート権限タブでは、ユーザーグループのテンプレートグループ（およびテンプレート）データへのアクセスを指定できます:

ホスト権限タブでは、ユーザーグループのホストグループ（およびホスト）データへのアクセスを指定できます:

をクリックして、テンプレート/ホストグループ（親またはネストされたグループ）を選択し、それらに権限を割り当てます。グループ名の入力を開始すると（一致するグループのドロップダウンが表示されます）、または選択をクリックしてすべてのグループのリストが表示されるポップアップウィンドウを開くことができます。

次に、オプションボタンを使用して選択したグループに権限を割り当てます。可能な権限は以下の通りです:

- 読書**-**書き込み - グループへの読み書きアクセス;
- 読書 - グループへの読み取り専用アクセス;
- 拒否 - グループへのアクセスを拒否。

同じテンプレート/ホストグループが異なる権限で複数行に追加されている場合、最も厳しい権限が適用されます。

スーパ管理者ユーザーは、ネストされたグループに親グループと同じレベルの権限を強制することができます。これはホスト/テンプレートグループの設定フォームで行うことができます。

テンプレート権限および ホスト権限タブは同じパラメータセットをサポートしています。

現在のグループへの権限は 権限ブロックに表示され、変更または削除できます。

Note:
ユーザーグループがホストに対して読書-書き込み権限を持ち、このホストにリンクされたテンプレートに対して拒否または権限がない場合、そのグループのユーザーはホスト上のテンプレート化されたアイテムを編集できず、テンプレート名はアクセス不可テンプレートとして表示されます。

問題タグフィルタータブでは、ユーザーグループがタグ名と値でフィルタリングされた問題を表示できるように、タグベースの権限を設定できます:

をクリックしてホストグループを選択します。タグフィルターを適用するホストグループを選択するには、選択をクリックして既存のホストグループの完全なリストを取得するか、ホストグループ名の入力を開始して一致するグループのドロップダウンを取得します。テンプレートグループには問題タグフィルターを適用できないため、ホストグループのみが表示されます。

次に、すべてのタグから タグリストに切り替えて、フィルタリングする特定のタグとその値を設定できます。値のないタグ名は追加できますが、名前のない値は追加できません。権限ブロックには最初の 3 つのタグ（値がある場合は値も）が表示されます。3 つ以上ある場合は、アイコンをクリックまたはホバーすることで確認できます。

タグフィルターを使用すると、ホストグループへのアクセスと問題の表示権限を分離できます。

たとえば、データベース管理者が「MySQL」データベースの問題のみを表示する必要がある場合、まずデータベース管理者用のユーザーグループを作成し、「target」タグ名と「mysql」値を指定する必要があります。

「target」タグ名が指定され、値フィールドが空白の場合、ユーザーグループは選択したホストグループの「target」タグ名を持つすべての問題を表示できます。すべてのタグが選択されている場合、ユーザーグループは指定したホストグループのすべての問題を表示できます。

タグ名とタグ値が正しく指定されていることを確認してください。そうでない場合、ユーザーグループは問題を表示できません。

ユーザーが複数のユーザーグループのメンバーである場合の例を見てみましょう。この場合、タグのフィルタリングには OR 条件が使用されます。

ユーザーグループ **A**			ユーザーグループ **B**			両方のグループのメンバーであるユーザーに表示される結果
タグフィルター						
ホストグループ	タグ名	タグ値	ホストグループ	タグ名	タグ値	
Databases	target	mysql	Databases	target	oracle	target:mysql または target:oracle の問題が表示される
Databases	すべてのタグに設定		Databases	target	oracle	すべての問題が表示される
問題タグフィルターで未設定			Databases	target	oracle	target:oracle の問題が表示される

Attention:
フィルター（例: 特定のホストグループ「Databases」のすべてのタグ）を追加すると、他のホストグループの問題が表示できなくなります。

複数のユーザーグループからのアクセス

ユーザーは任意の数のユーザーグループに所属することができます。これらのグループは、ホストやテンプレートに対して異なるアクセス権限を持つことができます。

したがって、特権のないユーザーが最終的にどのエンティティにアクセスできるかを知ることが重要です。以下の例では、ユーザーグループ A と B に所属するユーザーが、ホストグループ 1 のホスト **X** へのアクセスがさまざまな状況でどのように影響を受けるかを考えます。

- グループ A がホストグループ 1 に対して読込権限のみを持ち、グループ B がホストグループ 1 に対して読書/書込権限を持つ場合、ユーザーは'X' に対して読書/書込権限を取得します。

Attention:
「読書/書込」権限は「読込」権限よりも優先されます。

- 上記と同じシナリオで、'X' が同時にグループ A または B に拒否されているホストグループ 2 にも所属している場合、ホストグループ 1 に読書/書込権限があっても、'X' へのアクセスは利用できません。
- グループ A に権限が定義されておらず、グループ B がホストグループ 1 に読書/書込権限を持っている場合、ユーザーは'X' に対して読書/書込権限を取得します。
- グループ A がホストグループ 1 に拒否権限を持ち、グループ B がホストグループ 1 に読書/書込権限を持っている場合、ユーザーは'X' へのアクセスが拒否されます。

その他の詳細

- ホストへの読み書きアクセス権を持つ管理者レベルのユーザーは、そのテンプレートが属するテンプレートグループへのアクセス権がない場合、テンプレートのリンク/リンク解除を行うことができません。テンプレートグループへの読み取りアクセス権があれば、ホストへのテンプレートのリンク/リンク解除は可能ですが、テンプレートリストにはテンプレートが表示されず、他の場所でテンプレートを操作することもできません。
- ホストへの読み取りアクセス権を持つ管理者レベルのユーザーは、設定セクションのホストリストにホストが表示されませんが、IT サービスの設定ではホストのトリガーにアクセスできます。
- スーパ管理者以外のユーザー（「guest」を含む）は、マップが空であるか画像のみの場合、ネットワークマップを表示できます。マップにホスト、ホストグループ、トリガーが追加されると、権限が適用されます。
- Zabbix サーバーは、該当するホストへのアクセスが明示的に「拒否」されている場合、アクション操作の受信者として定義されたユーザーに通知を送信しません。

13 secrets の保存

概要

Zabbix は、セキュアなボールトから機密情報を取得するように設定できます。サポートされているシークレット管理サービスは、HashiCorp Vault KV Secrets Engine - Version 2、CyberArk Vault CV12 です。

シークレットは、以下の取得に使用できます:

- ユーザーマクロ値
- データベースアクセス認証情報

Zabbix は、ボールト内のシークレットへの読み取り専用アクセスを提供します。シークレットは他の誰かによって管理されていることを前提としています。

特定の Vault プロバイダーの設定については、以下を参照してください。

- HashiCorp の設定
- CyberArk の設定

シークレット値のキャッシュ

デフォルトでは、Vault シークレットマクロの値は、Zabbix サーバーが設定データをリフレッシュするたびに取得され、その後設定キャッシュに保存されます。Zabbix プロキシは、各設定同期時に Zabbix サーバーから Vault シークレットマクロの値を受け取り、自身の設定キャッシュに保存します。

> **Attention:**
> Zabbix サーバーとプロキシ間で暗号化を有効にする必要があります。有効でない場合、サーバーに警告メッセージが記録されます。

また、マクロ値を Zabbix サーバーと Zabbix プロキシが独立して取得するように設定することも可能です。

Vault からキャッシュされたシークレット値のリフレッシュを手動でトリガーするには、コマンドラインの'secrets_reload' オプションを使用します。

Zabbix フロントエンドのデータベース認証情報のキャッシュはデフォルトで無効になっていますが、zabbix.conf.php でオプション `$DB['VAULT_CACHE'] = true` を設定することで有効にできます。認証情報は、ファイルシステムの一時ファイルディレクトリを使用してローカルキャッシュに保存されます。Web サーバーは、プライベートな一時フォルダへの書き込みを許可する必要があります（たとえば、Apache の場合は構成オプション `PrivateTmp=True` を設定する必要があります）。データキャッシュのリフレッシュ/無効化の頻度を制御するには、ZBX_DATA_CACHE_TTL 定数を使用します。

TLS の設定

Zabbix コンポーネントと Vault 間の通信で TLS を設定するには、認証局（CA）によって署名された証明書をシステム全体のデフォルト CA ストアに追加します。別の場所を使用する場合は、SSLCALocation Zabbix server/proxy設定パラメータでディレクトリを指定し、証明書ファイルをそのディレクトリ内に配置し、次の CLIコマンドを実行します。

```
c_rehash .
```

1 CyberArk の設定

このセクションでは、Zabbix が CyberArk Vault CV12 からシークレットを取得する方法を説明します。

Vault は、公式のCyberArk ドキュメントに記載されているようにインストールおよび設定する必要があります。

Zabbix での TLS の設定については、シークレットの保存を参照してください。

データベース認証情報

データベース認証情報を含むシークレットへのアクセスは、各 Zabbix コンポーネントごとに個別に設定されます。

サーバーとプロキシ

Zabbixサーバーまたはプロキシがボールトからデータベース認証情報を取得するには、設定ファイルに以下のパラメータを指定します。

- `Vault` - どのボールトプロバイダーを使用するか;
- `VaultURL` - ボールトサーバーの HTTP[S] URL;
- `VaultDBPath` - データベース認証情報を含むボールトシークレットへのクエリ。キー”Content” と”UserName” で取得されます（このオプションは DBUser および DBPassword が指定されていない場合のみ使用できます）;
- `VaultTLSCertFile`, `VaultTLSKeyFile` - SSL 証明書およびキーファイル名。これらのオプションの設定は必須ではありませんが、強く推奨されます;
- `VaultPrefix` - ボールトのパスまたはクエリのカスタムプレフィックス。指定しない場合は、最も適切なデフォルトが使用されます。

Attention:
`Vault`、`VaultURL`、`VaultTLSCertFile`、`VaultTLSKeyFile`、および `VaultPrefix` の設定パラメータは、Zabbix サーバー（および設定済みの場合は Zabbix プロキシ）がシークレットボールトマクロを処理する際のボールト認証にも使用されます。Zabbix サーバーおよびプロキシは、VaultDBPath から DB 認証情報を含むボールトシークレットマクロを開きません。

Zabbix サーバーおよび Zabbix プロキシは、起動時に *zabbix_server.conf* および *zabbix_proxy.conf* ファイルからボールト関連の設定パラメータを読み込みます。

例

1. *zabbix_server.conf* で、以下のパラメータを指定します。

```
Vault=CyberArk
VaultURL=https://127.0.0.1:1858
VaultDBPath=AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
VaultTLSCertFile=cert.pem
VaultTLSKeyFile=key.pem
VaultPrefix=/AIMWebService/api/Accounts?
```

2. Zabbix は、ボールトに以下の API リクエストを送信します。

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. ボールトのレスポンスには”Content” と”UserName” のキーが含まれます。

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

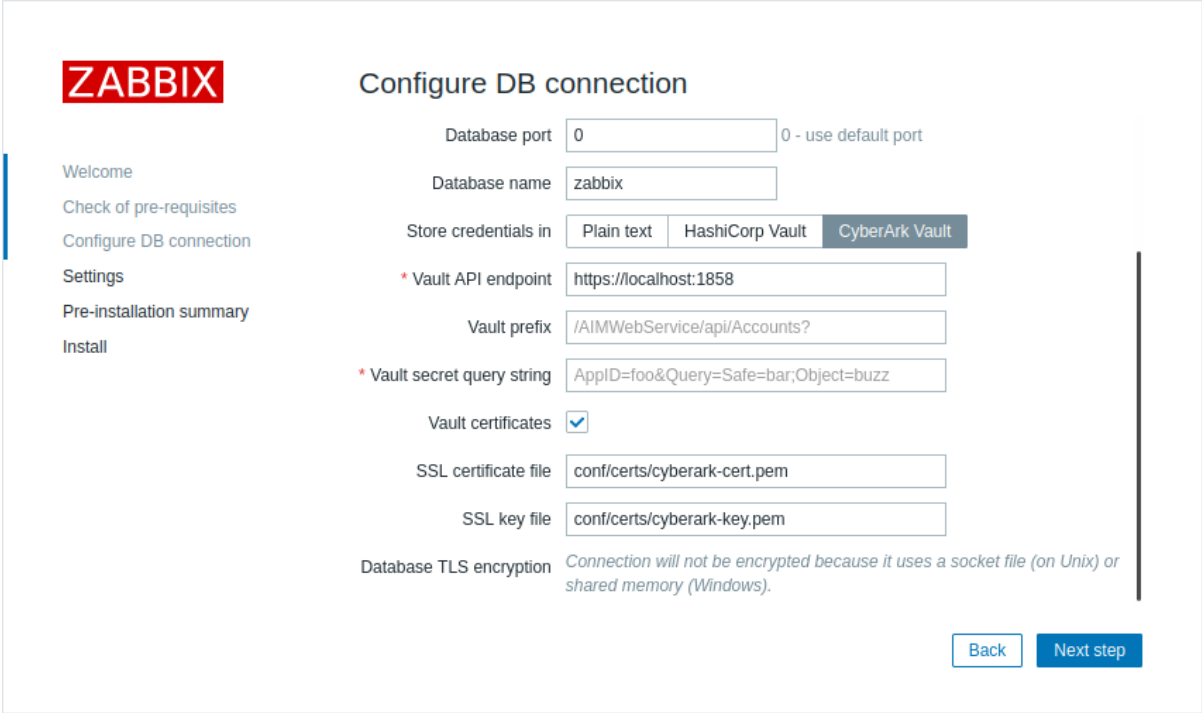
4. その結果、Zabbix はデータベース認証に以下の認証情報を使用します。

- ユーザー名: <username>
- パスワード: <password>

フロントエンド

Zabbix フロントエンドがボールトからデータベースの認証情報を取得するには、フロントエンドのインストール時に以下のパラメータを指定します。

1. *Configure DB Connection* ステップで、*Store credentials in* パラメータを「CyberArk Vault」に設定します。



2. 次に、追加パラメータを入力します:

パラメータ	必須	デフォルト値	説明
Vault API endpoint	はい	https://localhost:1858	`scheme://host:port` 形式でボールトへの接続 URL を指定します
Vault prefix	いいえ	/AIMWebService/api/Accounts?	ボールトパスまたはクエリのカスタムプレフィックスを指定します。指定しない場合はデフォルトが使用されます。
Vault secret query string	はい		データベースの認証情報を取得する場所を指定するクエリ。 例: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	いいえ		チェックボックスをオンにすると、クライアント認証を設定するための追加パラメータが表示されます。このパラメータは任意ですが、CyberArk Vault との通信には有効にすることを強く推奨します。
SSL certificate file	いいえ	conf/certs/cyberark-cert.pem	SSL 証明書ファイルへのパス。ファイルは PEM 形式である必要があります。 証明書ファイルに秘密鍵も含まれている場合は、SSL key file パラメータを空にしてください。
SSL key file	いいえ	conf/certs/cyberark-key.pem	クライアント認証に使用する SSL 秘密鍵ファイル名。ファイルは PEM 形式である必要があります。

ユーザーマクロの値

Vault secret ユーザーマクロの値を保存するために CyberArk Vault を使用するには、以下を確認してください。

- Zabbix サーバーが CyberArk Vault で動作するように設定されていること
- 管理 → 一般 → その他の *Vault* プロバイダーパラメータが「CyberArk Vault」に設定されていること



Note:
Zabbix サーバー（および設定されている場合は Zabbix プロキシ）は、Vault から *Vault secret* マクロ値へのアクセスが必要です。Zabbix フロントエンドにはそのようなアクセスは必要ありません。

マクロ値にはクエリ（query:key）を含める必要があります。

Zabbix によるマクロ値の処理の詳細については、*Vault secret* マクロを参照してください。

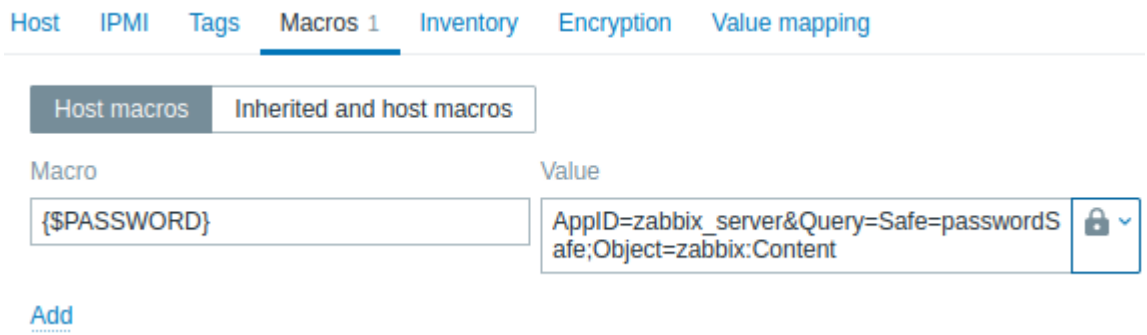
クエリ構文

コロン記号（":"）は、クエリとキーを区切るために予約されています。

クエリ自体にスラッシュやコロンが含まれている場合、これらの記号は URL エンコードする必要があります（"/" は"%2F"、":" は"%3A" としてエンコードされます）。

例

1. Zabbix で、タイプ *Vault secret* のユーザーマクロ {$PASSWORD} を追加し、値を AppID=zabbix_server&Query=Safe=passwordSaf とします。



2. Zabbix は以下の API リクエストをボールトに送信します:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. ボールトのレスポンスには"Content" キーが含まれます:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. その結果、Zabbix はマクロ {$PASSWORD} を値-<password> に解決します。

既存の設定の更新

CyberArk Vault からシークレットを取得するための既存の設定を更新するには、以下の手順を実行します。

1. データベース認証情報のセクションで説明されているように、Zabbix サーバーまたはプロキシの設定ファイルパラメータを更新します。
2. フロントエンドのセクションで説明されているように、Zabbix フロントエンドを再設定し、必要なパラメータを指定して DB 接続設定を更新します。Zabbix フロントエンドを再設定するには、ブラウザでフロントエンドセットアップ URL を開きます。

- Apache の場合: http://<server_ip_or_name>/zabbix/setup.php
- Nginx の場合: http://<server_ip_or_name>/setup.php

または、これらのパラメータはフロントエンド設定ファイル（*zabbix.conf.php*）で設定できます。

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
```

```
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. 必要に応じて、ユーザーマクロ値のセクションで説明されているように、ユーザーマクロを設定します。

HashiCorp Vault からシークレットを取得するための既存の設定を更新するには、*HashiCorp の設定*を参照してください。

2 HashiCorp の設定

概要

このセクションでは、HashiCorp Vault KV Secrets Engine - Version 2 からシークレットを取得するための Zabbix の設定方法について説明します。

Vault は公式の HashiCorp ドキュメント に記載されているようにデプロイおよび設定する必要があります。

Zabbix での TLS の設定方法については、シークレットの保存 を参照してください。

データベース認証情報の取得

データベース認証情報を持つシークレットを正常に取得するには、以下の両方を設定する必要があります。

- Zabbix サーバー/プロキシ
- Zabbix フロントエンド

サーバー/プロキシ

Zabbix サーバーまたはプロキシを設定するには、設定ファイルに以下のパラメータを指定します。

- `Vault` - 使用するボールトプロバイダー
- `VaultToken` - ボールト認証トークン（詳細は Zabbix サーバー/プロキシの設定ファイルを参照）
- `VaultURL` - ボールトサーバーの HTTP[S] URL
- `VaultDBPath` - データベース認証情報を含むボールトシークレットへのパス（このオプションは DBUser および DBPassword が指定されていない場合のみ使用可能）；Zabbix サーバーまたはプロキシは「password」と「username」キーで認証情報を取得します
- `VaultPrefix` - ボールトごとにパスまたはクエリのカスタムプレフィックス；指定しない場合は最適なデフォルト値が使用されます

Attention:
`Vault`、`VaultToken`、`VaultURL`、`VaultPrefix` の設定パラメータは、Zabbix サーバー（および設定済みの場合は Zabbix プロキシ）によるシークレットボールトマクロの処理時にもボールト認証に使用されます。Zabbix サーバーおよびプロキシは、VaultDBPath から DB 認証情報を含むボールトシークレットマクロを開きません。

 異なるプロキシごとに異なるトークンを使用することを強く推奨します。

Zabbix サーバーおよび Zabbix プロキシは、起動時に *zabbix_server.conf* および *zabbix_proxy.conf* からボールト関連の設定パラメータを読み込みます。さらに、Zabbix サーバーおよび Zabbix プロキシは、起動時に一度だけ `VAULT_TOKEN` 環境変数を読み込み、フォークされたスクリプトから利用できないようにアンセットします；`VaultToken` と `VAULT_TOKEN` の両方に値が含まれている場合はエラーとなります。

例

1. *zabbix_server.conf* で、以下のパラメータを指定します。

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. 必要なシークレットをボールトに作成するために、以下の CLI コマンドを実行します。

```
#### まだ有効化されていない場合は "secret/" マウントポイントを有効化します。"kv-v2" を使用する必要があります。
vault secrets enable -path=secret/ kv-v2

#### "secret/" マウントポイントと "zabbix/database" パスの下に、username と password のキーで新しいシークレットを追加
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### シークレットが正常に追加されたことをテストします。
```

```
vault kv get secret/zabbix/database

#### 最後に Curl でテストします。"data" をマウントポイントの後に手動で追加し、"/v1" をマウントポイントの前に追加する必要があ
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. その結果、Zabbix サーバーはデータベース認証のために以下の認証情報を取得します。

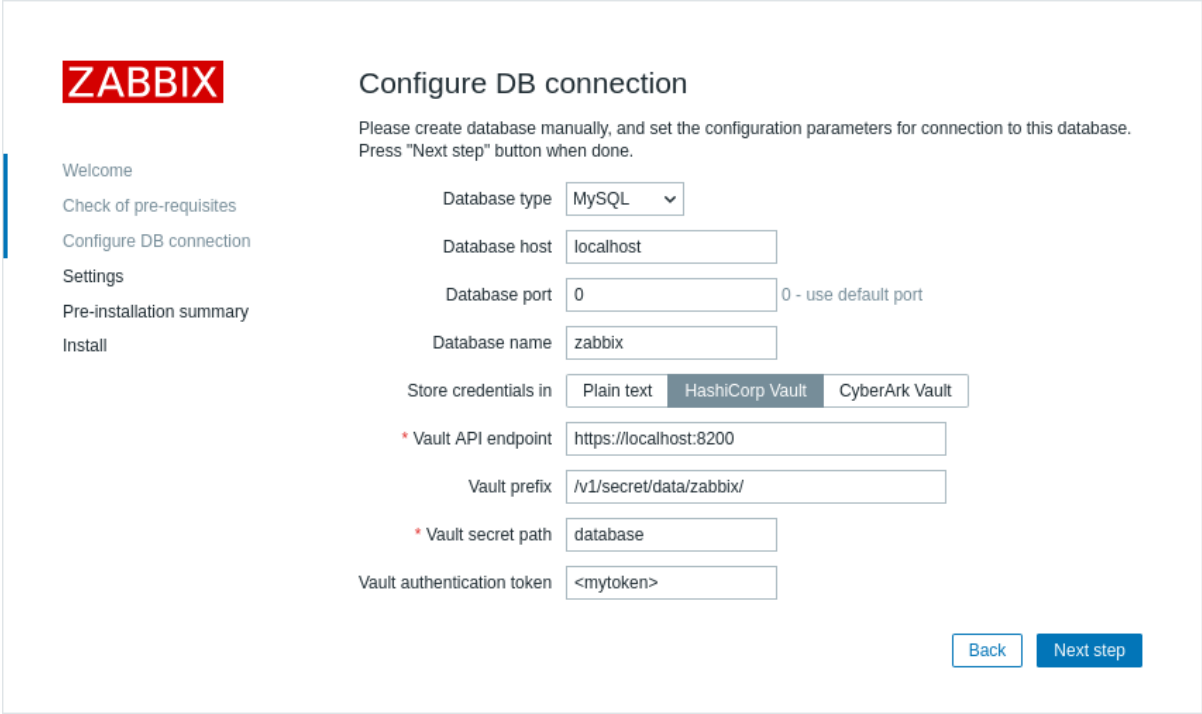
- ユーザー名: zabbix
- パスワード: <password>

フロントエンド

Zabbix フロントエンドは、フロントエンドのインストール時、またはフロントエンドの設定ファイル（zabbix.conf.php）を更新することで、Vault からデータベースの認証情報を取得するように設定できます。

Attention:
前回のフロントエンドインストール以降に Vault の認証情報が変更された場合は、フロントエンドのインストールを再実行するか、zabbix.conf.php を更新してください。詳細は既存設定の更新も参照してください。

フロントエンドのインストール時には、*Configure DB Connection* ステップで設定パラメータを指定する必要があります。



- *Store credentials in* パラメータを「HashiCorp Vault」に設定します。
- 接続パラメータを指定します。

パラメータ	必須	デフォルト値	説明
Vault API endpoint	はい	https://localhost:8200	scheme://host:port 形式で Vault への接続 URL を指定します
Vault prefix	いいえ	/v1/secret/data/	Vault パスまたはクエリのカスタムプレフィックスを指定します。指定しない場合はデフォルト値が使用されます。 例: /v1/secret/data/zabbix/
Vault secret path	いいえ		データベースの認証情報を「password」と「username」キーで取得するためのシークレットへのパス。 例: database
Vault authentication token	いいえ		シークレットパスへの読み取り専用アクセス用の認証トークンを指定します。 トークンや Vault ポリシーの作成方法については HashiCorp ドキュメントを参照してください。

ユーザーマクロ値の取得

Vault secret ユーザーマクロ値の保存に HashiCorp Vault を使用するには、以下を確認してください。

- Zabbix サーバー/プロキシが HashiCorp Vault で動作するように設定されていること
- 管理 → 一般 → その他の *Vault* プロバイダーパラメーターが「HashiCorp Vault」（デフォルト）に設定されていること



Note:
Zabbix サーバー（および設定されている場合は Zabbix プロキシ）は、Vault から *Vault secret* マクロ値へのアクセスが必要です。Zabbix フロントエンドにはそのようなアクセスは必要ありません。

マクロ値には参照パス（例：`path:key`、`macros:password` など）を指定する必要があります。Zabbix サーバー/プロキシの設定時に指定した認証トークン（`VaultToken` パラメーター）は、このパスへの読み取り専用アクセス権を持っている必要があります。

Zabbix によるマクロ値の処理の詳細については、*Vault secret* マクロを参照してください。

パス構文

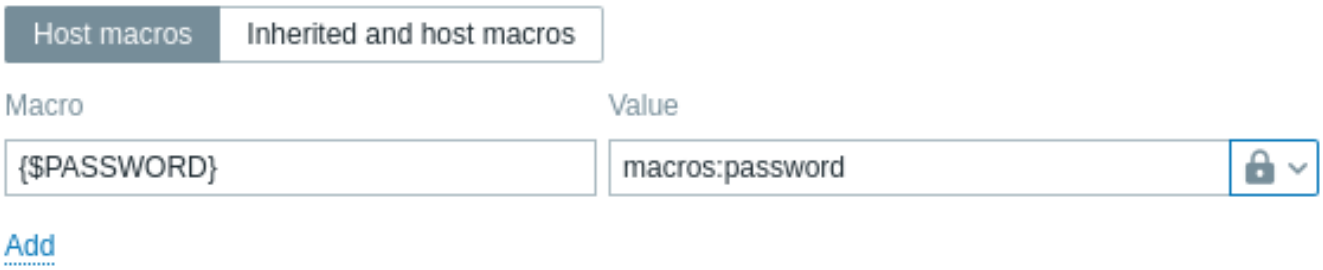
スラッシュ (”/”) とコロン (”:”) の記号は予約されています。

スラッシュは、マウントポイントとパスを区切る場合にのみ使用できます (例: *secret/zabbix* では、マウントポイントは”secret”、パスは”zabbix”)。Vault マクロの場合、コロンはパス/クエリとキーを区切る場合にのみ使用できます。

スラッシュで区切られた名前のマウントポイントを作成する必要がある場合 (例: *foo/bar/zabbix* では、マウントポイントは”foo/bar”、パスは”zabbix” であり、”foo%2Fbar/zabbix” のようにエンコードできます)、またはマウントポイント名やパスにコロンを含める必要がある場合は、スラッシュやコロンの記号を URL エンコードすることができます。

例

1. Zabbix で、タイプ「Vault シークレット」のユーザーマクロ {$PASSWORD} を追加し、値を `macros:password` に設定します。



2. 必要なシークレットを Vault に作成するために、以下の CLI コマンドを実行します。

```
#### まだ有効化されていない場合は、"secret/" マウントポイントを有効化します。"kv-v2" を使用する必要があることに注意してくだ
vault secrets enable -path=secret/ kv-v2

#### マウントポイント"secret/" およびパス"zabbix/macros" の下に、キー"password" で新しいシークレットを追加します。
vault kv put -mount=secret zabbix/macros password=<password>

#### シークレットが正常に追加されたかテストします。
vault kv get secret/zabbix/macros

#### 最後に Curl でテストします。"data" をマウントポイントの後に手動で追加し、"/v1" をマウントポイントの前に追加する必要があ
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. その結果、Zabbix はマクロ {$PASSWORD} を値 <password> に解決します。

既存の設定の更新

HashiCorp Vault からシークレットを取得するための既存の設定を更新するには:

1. データベース認証情報 セクションで説明されているように、Zabbix サーバーまたはプロキシの設定ファイルパラメーターを更新します。
2. フロントエンド セクションで説明されているように、Zabbix フロントエンドを再設定し、必要なパラメーターを指定して DB 接続設定を更新します。Zabbix フロントエンドを再設定するには、ブラウザでフロントエンドセットアップ URL を開きます:

- Apache の場合: http://<server_ip_or_name>/zabbix/setup.php
- Nginx の場合: http://<server_ip_or_name>/setup.php

または、これらのパラメーターはフロントエンド設定ファイル (*zabbix.conf.php*) で設定できます:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

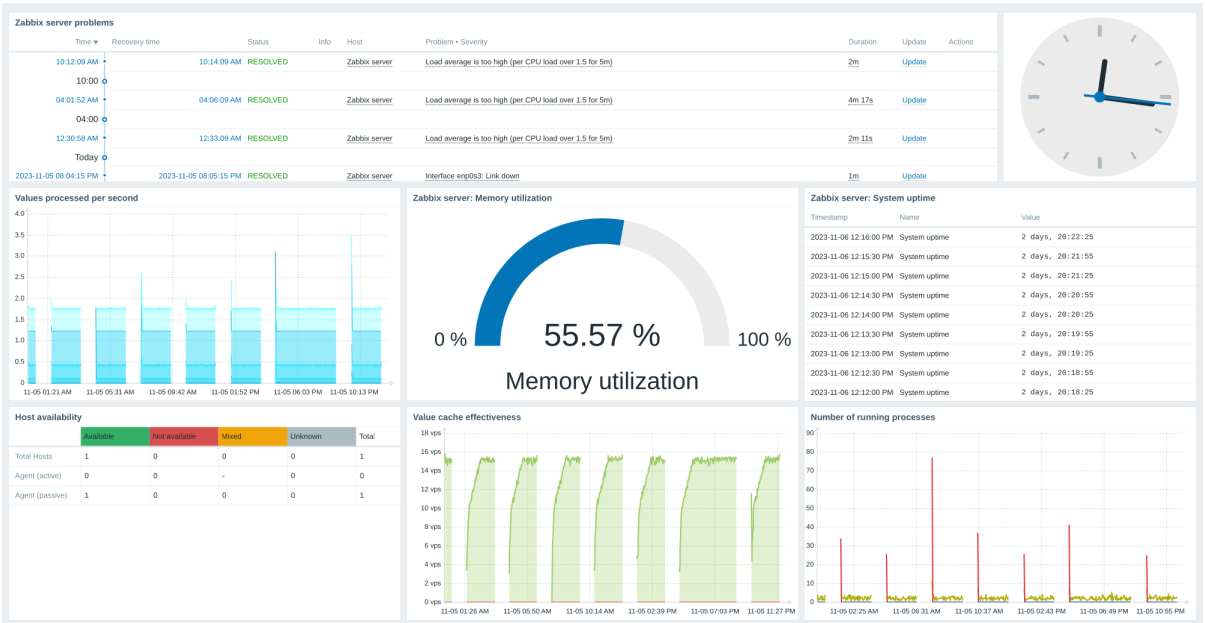
3. 必要に応じて、ユーザーマクロ値 セクションで説明されているようにユーザーマクロを設定します。

CyberArk Vault からシークレットを取得するための既存の設定を更新するには、*CyberArk の設定* を参照してください。

14 スケジュールレポート

概要

スケジュールレポート機能を使用すると、指定したダッシュボードの PDF バージョンを、指定した受信者に定期的に送信するように設定できます。



前提条件:

- スケジュールレポートの生成を有効にするために、Zabbix Web サービスが正しくインストールおよび設定されている必要があります。手順については、スケジュールレポートの設定を参照してください。
- ユーザーは、次の権限を持つ *Admin* または *Super admin* タイプのユーザーロールを持っている必要があります。
  - *UI* 要素へのアクセスブロックのスケジュールレポート（レポート設定の表示用）
  - アクションへのアクセスブロックのスケジュールレポートの管理（レポートの作成/編集用）

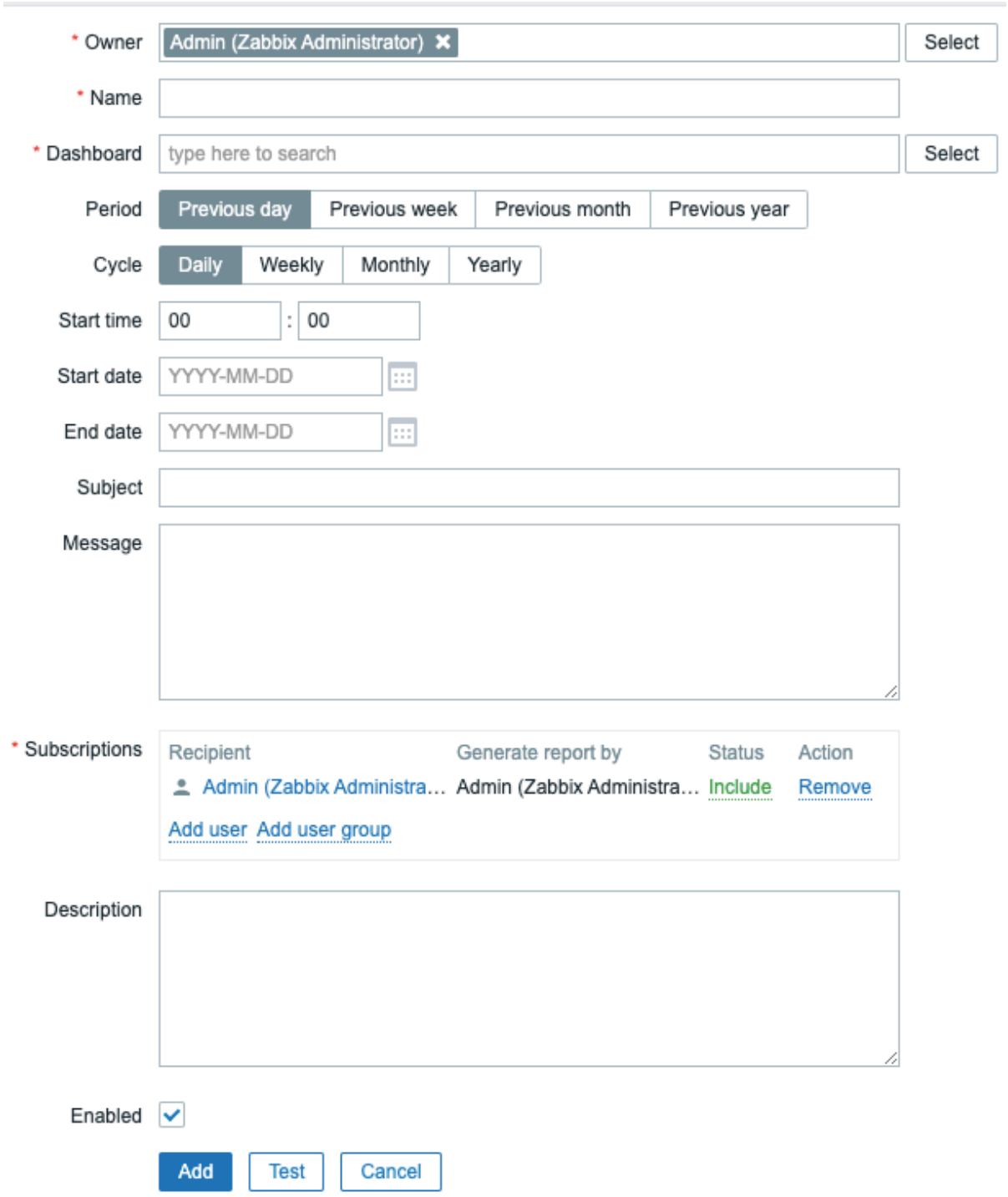
Zabbix フロントエンドでスケジュールレポートを作成するには、次の手順を実行します。

- レポート > スケジュールレポートに移動します。
- 画面右上のレポートの作成をクリックします。
- フォームにレポートのパラメータを入力します。

既存のレポートを開き、クローンボタンをクリックして、別の名前で保存することでレポートを作成することもできます。

設定

定期レポートタブには、レポートの一般的な属性が含まれています。



すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
所有者	レポートを作成するユーザー。スーパ管理者レベルのユーザーは所有者を変更できます。管理者レベルのユーザーの場合、このフィールドは読み取り専用です。
名前	レポートの名前。一意である必要があります。
ダッシュボード	レポートの元となるダッシュボード。1 つのダッシュボードのみ選択できます。ダッシュボードを選択するには、名前を入力し始めると一致するダッシュボードのリストが表示されます。スクロールして選択してください。または、フィールドの横にある選択をクリックして、表示されたリストからダッシュボードを選択することもできます。
期間	レポートを作成する期間。前日、前週、前月、前年から選択します。

パラメータ	説明
サイクル	レポートの生成頻度。レポートは毎日、毎週、毎月、毎年送信できます。「毎週」モードでは、レポートを送信する曜日を選択できます。
開始時刻	レポートを作成する時刻（hh:mm 形式）。Zabbix サーバーのタイムゾーンが使用されることに注意してください。
繰り返し	レポートを送信する曜日。このフィールドはサイクルが「毎週」に設定されている場合のみ利用できます。
開始日	定期レポートの生成を開始する日付。
終了日	定期レポートの生成を停止する日付。
件名	レポートメールの件名。サポートされているマクロ：{TIME}、{TIMESTAMP}。
メッセージ	レポートメールの本文。サポートされているマクロ：{TIME}、{TIMESTAMP}。
購読者	レポートの受信者リスト。デフォルトではレポートの所有者のみが含まれます。メールメディアが設定されている任意の Zabbix ユーザーをレポートの受信者として指定できます。 ユーザーの追加またはユーザーグループの追加をクリックして、受信者を追加します。 ユーザー名をクリックして設定を編集します： レポート生成者 - レポートデータを現在のユーザーまたは受信者のダッシュボード権限に基づいて生成するかどうか。 ステータス - 「含める」を選択するとユーザーにレポートを送信し、「除外」を選択するとこのユーザーへのレポート送信を防止します。少なくとも 1 人のユーザーが「含める」ステータスである必要があります。「除外」ステータスは、含まれているユーザーグループから特定のユーザーを除外するために使用できます。 権限が不十分なユーザー（つまり、管理者ユーザータイプに基づくロールを持ち、受信者またはレポート所有者と同じユーザーグループのメンバーでないユーザー）は、受信者およびレポート生成者フィールドで実際の名前の代わりに「アクセスできないユーザー」または「アクセスできないユーザーグループ」と表示されます。ステータスおよびアクションフィールドは読み取り専用で表示されます。
有効	レポートのステータス。このチェックボックスをクリアすると、レポートが無効になります。
説明	レポートの任意の説明。この説明は内部用であり、レポート受信者には送信されません。

フォームボタン

フォームの下部にあるボタンで、いくつかの操作を実行できます。

Add	レポートを追加します。このボタンは新しいレポートの場合のみ利用できます。
Update	レポートのプロパティを更新します。
Clone	現在のレポートのプロパティを元に別のレポートを作成します。
Test	レポートの設定が正しいかどうかを、現在のユーザーにレポートを送信してテストします。
Delete	レポートを削除します。
Cancel	レポートプロパティの編集をキャンセルします。

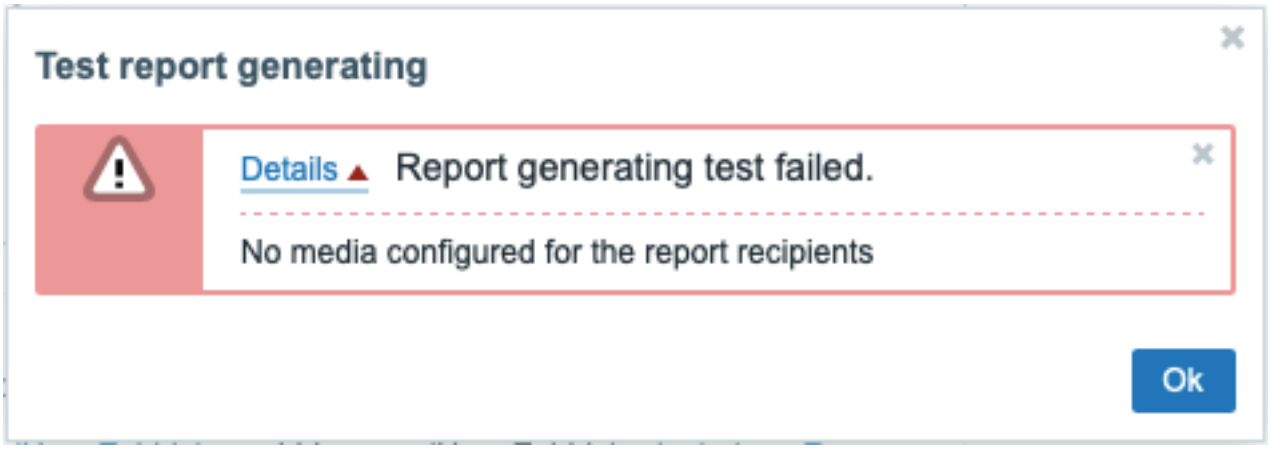
テスト

レポートをテストするには、レポート設定フォームの下部にあるテストボタンをクリックします。

Note:
テストボタンは、ダッシュボードのアクションメニューからレポート設定フォームを開いた場合は利用できません。

設定が正しければ、テストレポートは現在のユーザーにすぐに送信されます。テストレポートでは、購読者とレポート生成者のユーザー設定は無視されます。

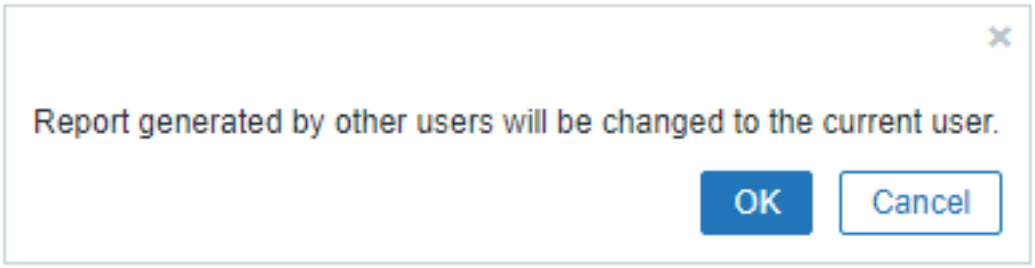
設定が正しくない場合は、考えられる原因を説明するエラーメッセージが表示されます。



レポートの更新

既存のレポートを更新するには、レポート名をクリックし、必要な設定変更を行い、更新ボタンをクリックします。

既存のレポートが他のユーザーによって更新され、このユーザーがダッシュボードを変更した場合、更新ボタンをクリックすると「他のユーザーによって生成されたレポートは現在のユーザーに変更されます」という警告メッセージが表示されます。



この段階で *OK* をクリックすると、以下の変更が行われます。

- レポートの生成者の設定が、レポートを最後に編集したユーザーを表示するように更新されます（レポートの生成者が受信者に設定されていない場合）。
- 「アクセスできないユーザー」または「アクセスできないユーザーグループ」と表示されていたユーザーは、レポート購読者のリストから削除されます。

キャンセルをクリックすると、設定フォームが閉じられ、レポートの更新がキャンセルされます。

レポートのクローン作成

既存のレポートをすばやくクローンするには、既存のレポート設定フォームの下部にあるクローンボタンをクリックします。他のユーザーが作成したレポートをクローンする場合、新しいレポートの所有者は現在のユーザーになります。

ユーザーの権限に応じて、レポート設定が新しいレポート設定フォームにコピーされます。

- レポートをクローンするユーザーがダッシュボードへの権限を持っていない場合、ダッシュボードフィールドはクリアされます。
- レポートをクローンするユーザーがサブスクリプションリスト内の一部のユーザーまたはユーザーグループへの権限を持っていない場合、アクセスできない受信者はクローンされません。
- レポートの生成者の設定は、（レポートの生成者が受信者に設定されていない限り）現在のユーザーを表示するように更新されます。

必要な設定とレポート名を変更し、追加をクリックします。

15 データのエクスポート

概要

Zabbix は、次の 2 つの方法でリアルタイムでのデータエクスポートをサポートします。

- ファイルにエクスポート
- 外部システムへのストリーミング

次のエンティティをエクスポートできます。

- トリガーイベント

- アイテムの値
- トレンド (ファイルにエクスポートのみ)

1 ファイルにエクスポート

概要

トリガーイベント、アイテム値、トレンドのリアルタイムエクスポートを、改行区切りの JSON 形式で設定することができます。

エクスポートはファイルに対して行われ、エクスポートファイルの各行が JSON オブジェクトとなります。値のマッピングは適用されません。

エラーが発生した場合 (エクスポートファイルにデータを書き込めない、エクスポートファイルのリネームができない、リネーム後に新しいファイルを作成できないなど)、そのデータアイテムは破棄され、エクスポートファイルには書き込まれません。Zabbix データベースにのみ書き込まれます。エクスポートファイルへの書き込みは、書き込みの問題が解決されると再開されます。

エクスポートファイルは、ファイル所有者のみが読み書きできる権限で作成されます。さらに、所有グループによって読み取り可能です。他のすべての権限は拒否されます。

エクスポートされる情報の詳細については、エクスポートプロトコルのページを参照してください。

ホスト/アイテムが、データ受信後に削除され、サーバーがデータをエクスポートする前に削除された場合、ホストグループ、ホスト名、アイテム名などのメタデータが存在しない場合があることに注意してください。

設定

トリガーイベント、アイテム値、トレンドのリアルタイムエクスポートは、エクスポートファイルのディレクトリを指定することで設定します。サーバー設定のExportDirパラメータを参照してください。

他に 2 つのパラメータがあります。

- ExportFileSize は、個々のエクスポートファイルの最大許容サイズを設定するために使用できます。プロセスがファイルに書き込む必要がある場合、最初にファイルのサイズを確認します。設定されたサイズ制限を超えている場合、ファイル名の末尾に.old を付けてファイル名を変更し、元の名前で新しいファイルを作成します。

> **Attention:**
> データを書き込む各プロセスごとにファイルが作成されます（つまり、約 4～30 ファイル）。エクスポートファイルごとのデフォルトサイズは 1G のため、大きなエクスポートファイルを保持するとディスク容量が急速に消費される可能性があります。

- ExportType は、エクスポートするエンティティタイプ（イベント、履歴、トレンド）を指定できます。

2 外部システムへのストリーミング

概要

Zabbix から外部システムへ HTTP 経由でアイテム値やイベントをストリーミングすることができます（プロトコルの詳細を参照）。

タグフィルターを使用して、アイテム値やイベントのサブセットをストリーミングすることができます。

データストリーミングには、2 つの Zabbix サーバープロセス `connector manager` と `connector worker` が関与します。Zabbix の内部アイテム `zabbix[connector_queue]` を使用して、コネクタキューにエンキューされた値の数を監視できます。

設定

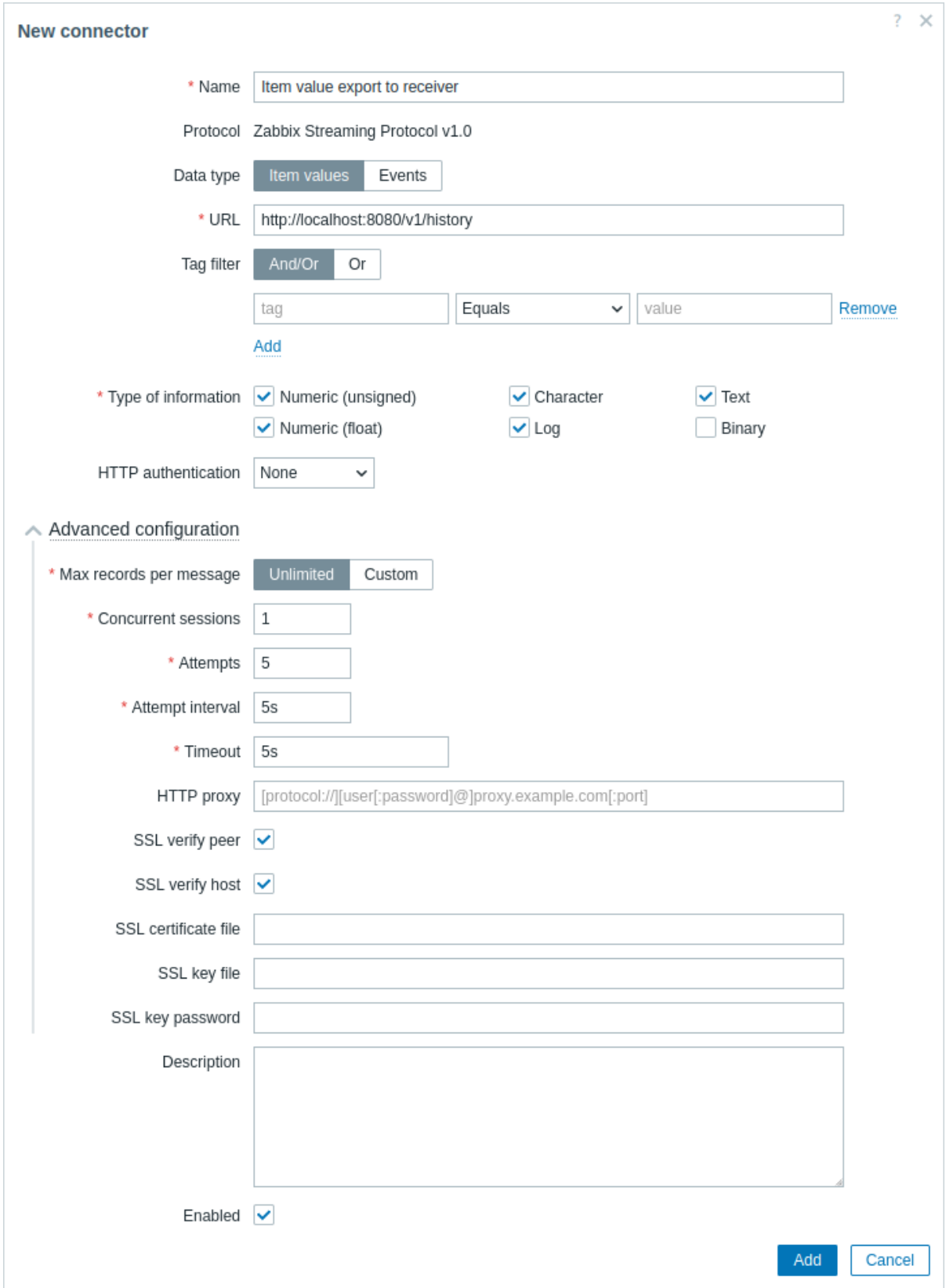
外部システムへのデータストリーミングを設定するには、以下の手順が必要です。

1. Zabbix からデータを受信するためのリモートシステムをセットアップします。この目的のために、以下のツールが利用可能です。

- 受信した情報を `events.ndjson` および `history.ndjson` ファイルに記録するシンプルな受信プログラムの例。
- Zabbix サーバー用 Kafka コネクタ - Zabbix サーバーから Kafka ブローカーにアイテム値やイベントを転送するために設計された Go 言語製の軽量サーバー。

2. `zabbix_server.conf` のStartConnectorsパラメータを調整して、Zabbix で必要な数のコネクタワーカーを設定します。コネクタワーカーの数は、Zabbix フロントエンドで設定されたコネクタ数と一致する必要があります（または同時セッション数が 1 より多い場合はそれ以上に設定します）。その後、Zabbix サーバーを再起動します。

3. Zabbix フロントエンドで新しいコネクタを設定します (管理 > 一般 > コネクタ)、`zabbix_server -R config_cache_reload` コマンドでサーバーキャッシュをリロードします。



必須フィールドはアスタリスクでマークされています。

パラメータ	説明
名前	コネクタ名を入力します。
データ型	ストリーミングするデータ型を選択します: アイテム値 - Zabbix から外部システムへアイテム値をストリーミング; イベント - Zabbix から外部システムへイベントをストリーミング。
URL	受信プログラムの URL を入力します。ユーザーマクロがサポートされています。

パラメータ	説明
タグフィルタ	タグフィルタに一致するアイテム値またはイベントのみをエクスポートします。設定しない場合はすべてをエクスポートします。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件の計算タイプは 2 つあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
情報の型	コネクタがストリーミングするアイテム値をフィルタするための情報の型（数値（符号なし）、数値(浮動小数点)、文字列など）を選択します。 このフィールドはデータ型が「アイテム値」に設定されている場合に利用できます。
HTTP 認証	認証オプションを選択します: なし - 認証を使用しない; **Basic** - Basic 認証を使用; **NTLM** - NTLM（Windows NT LAN Manager）認証を使用; **Kerberos** - Kerberos 認証を使用（参考: Zabbix での Kerberos 設定）; **Digest** - Digest 認証を使用; **Bearer** - Bearer 認証を使用。
ユーザー名	ユーザー名（最大 255 文字）を入力します。ユーザーマクロがサポートされています。 このフィールドは *HTTP* 認証が「Basic」「NTLM」「Kerberos」「Digest」に設定されている場合に利用できます。
パスワード	ユーザーパスワード（最大 255 文字）を入力します。ユーザーマクロがサポートされています。 このフィールドは *HTTP* 認証が「Basic」「NTLM」「Kerberos」「Digest」に設定されている場合に利用できます。
Bearer トークン	Bearer トークンを入力します。ユーザーマクロがサポートされています。 このフィールドは *HTTP* 認証が「Bearer」に設定されている場合に利用でき、必須です。
詳細設定	詳細設定ラベルをクリックすると詳細設定オプションが表示されます（下記参照）。
メッセージあたりの最大レコード数	1 つのメッセージ内でストリーミングできる値またはイベントの最大数を指定します。
同時セッション数	このコネクタで実行する送信プロセスの数を選択します。最大 100 セッションまで指定でき、デフォルト値は「1」です。
試行回数	データストリーミングの試行回数。最大 5 回まで指定でき、デフォルト値は「1」です。
試行間隔	データストリーミングが失敗した場合にコネクタが待機する時間を指定します。最大 10 秒まで指定でき、デフォルト値は「5 秒」です。 このフィールドは試行回数が「2」以上に設定されている場合に利用できます。 失敗した試行とは、接続の確立に失敗した場合、または HTTP レスポンスコードが 200、201、202、203、204 でない場合です。通信エラーや HTTP レスポンスコードが 200、201、202、203、204、400、401、403、404、405、415、422 でない場合に再試行が行われます。リダイレクトは追従されるため、302→200 は正常応答ですが、302→503 は再試行が行われます。
タイムアウト	メッセージのタイムアウトを指定します（1～60 秒、デフォルトは 5 秒）。 30s、1m などの時間サフィックスがサポートされています。ユーザーマクロがサポートされています。

パラメータ	説明
HTTP プロキシ	以下の形式で使用する HTTP プロキシを指定できます: `[protocol://][username[:password]@]proxy.example.com[:port]` ユーザーマクロがサポートされています。 オプションの `protocol://`プレフィックスを使用して、代替プロキシプロトコルを指定できます(プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました)。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。デフォルトでは 1080 ポートが使用されます。 *HTTP* プロキシが指定されている場合、プロキシは `http_proxy`、`HTTPS_PROXY` などのプロキシ関連の環境変数を上書きします。指定しない場合、プロキシはプロキシ関連の環境変数を上書きしません。入力された値はそのまま渡され、整合性チェックは行われません。 SOCKS プロキシアドレスを入力することもできます。間違ったプロトコルを指定すると、接続に失敗し、アイテムが未サポートになります。 HTTP プロキシでサポートされるのはシンプル認証のみであることに注意してください。
SSL ピア検証	チェックボックスをマークすると、Web サーバーの SSL 証明書を検証します。 サーバー証明書はシステム全体の証明書認証局(CA)ロケーションから自動的に取得されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCALocation`で上書きできます。
SSL ホスト検証	チェックボックスをマークすると、Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致するかどうかを検証します。 これは`CURLOPT_SSL_VERIFYHOST` cURL オプションを設定します。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイル名。証明書ファイルは PEM[1] 形式である必要があります。ユーザーマクロがサポートされています。 証明書ファイルに秘密鍵も含まれている場合は、*SSL* 鍵ファイルフィールドを空にします。鍵が暗号化されている場合は、*SSL* 鍵パスワードフィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCertLocation`で指定します。
SSL 鍵ファイル	クライアント認証に使用する SSL 秘密鍵ファイル名。秘密鍵ファイルは PEM[1] 形式である必要があります。ユーザーマクロがサポートされています。 このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLKeyLocation`で指定します。
SSL 鍵パスワード	SSL 秘密鍵ファイルのパスワード。ユーザーマクロがサポートされています。
説明	コネクタの説明を入力します。
有効	チェックボックスをマークするとコネクタが有効になります。

Kafka コネクタがカンマ区切りのブートストラップブローカーアドレスリスト(例:`Kafka.URL=kafka1.example.com:9093,kafka2.exar`で設定されている場合、Kafka クライアントは最初に応答したブローカーに接続し、そのクラスタのメタデータを使用します。リストに異なる Kafka クラスタのアドレスが含まれている場合、最も応答が早いクラスタのみが使用され、他のアドレスは利用不可として記録されます。その結果、コネクタが接続されていても、以下のような起動時警告が表示される場合があります。

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

一部の環境(プライベートネットワーク、コンテナネットワーク、非標準の DNS/hosts 設定など)では、ホスト名や IP がループバックアドレス(例:`127.0.0.1/localhost`)に解決されたり、クライアントによって正規化されたりすることがあり、このような警告が誤解を招く場合があります。混乱を避けるため、すべての `Kafka.URL` アドレスが同じ Kafka クラスタに属していることを確認し、コネクタホストからの DNS 解決やブローカーの `advertised.listeners` を検証し、ブローカーの advertised アドレスに解決されるアドレスを優先してください。

プロトコル

サーバーとレシーバー間の通信は、REST API、NDJSON、"Content-Type: application/x-ndjson" を使用して HTTP 経由で行われます。

詳細については、改行区切り JSON エクスポートプロトコルを参照してください。

サーバーリクエスト

ストリーミングアイテム値の例:

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
```

```
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

ストリーミングイベントの例:

```
POST /v1/events HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 333
Content-Type: application/x-ndjson

{"clock":1673454303,"ns":800155804,"value":1,"eventid":5,"name":"trigger for foo being 0","severity":0,"ho
{"clock":1673454303,"ns":832290669,"value":0,"eventid":6,"p_eventid":5}
```

レシーバーのレスポンス

レスポンスは、HTTP レスポンスステータスコードと JSON ペイロードで構成されます。HTTP レスポンスステータスコードは、リクエストが正常に処理された場合は”200”、”201”、”202”、”203”、”204” のいずれかでなければならず、失敗したリクエストの場合はそれ以外となります。

成功時の例:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

エラー時の例:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 SNMP ゲートウェイ

概要

Zabbix SNMP ゲートウェイは、SNMP ポーリングとトラッピングの両方をサポートする snmpd 用の AgentX 拡張機能です。

Zabbix SNMP ゲートウェイを使用すると、SNMP プロトコルを使用して以下の情報を取得できます:

- トリガーデータ
- 問題トリガーデータ
- ホストグループのステータス（グループごとのトリガーステータス別のトリガー数）

データは OID によって取得されます。OID は共通のベースと特定のサフィックスの組み合わせです。共通のベースは SNMP ゲートウェイの設定ファイルで設定されます。例:

- BaseOID=1.3.6.1.4.1.3043.7.55 - 任意のトリガーデータ用
- ProblemBaseOID=1.3.6.1.4.1.3047.7.55 - 問題トリガーデータ用
- BaseOID=1.3.6.1.4.1.3046.7.55 - ホストグループのステータス用

OID のサフィックスは、フロントエンドでホストトリガーの設定時にタグ（例: OIDSuffix:3）として設定します。

この場合、トリガーに関するすべての情報は OID=1.3.6.1.4.1.3043.7.55.X.3 で利用できます。ここでの”X” はトリガーデータフィールドの番号です（例: 1 - サフィックス、2 - ID、3 - 式、4 - 説明など）。

より詳細な説明や設定ファイルの例については、SNMP ゲートウェイの readme ファイルを参照してください。

インストールとセットアップ

インストール手順については、SNMP ゲートウェイのreadmeファイルを参照してください。

- snmpd のインストールと設定
- AgentX サポートの有効化
- Zabbix SNMP ゲートウェイの設定
- トリガー状態変更のための SNMP トラップの設定

データの取得

すべてが正しく設定されていれば、snmpwalk および snmpget コマンドを使用してデータを取得できます。

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

フィルタリングオプション

SNMP ゲートウェイの設定で、問題トリガー情報を以下のように制限できます。

- 深刻度で制限（デフォルトは ProblemMinSeverity=-1）
- 確認済みの問題を非表示にする（デフォルトは ProblemHideAck=false）

SNMP ゲートウェイの設定で、ホストグループごとの問題数を以下のように制限できます。

- 不明な状態のトリガーで制限（デフォルトは CountUnknown=false）
- 確認済み/未確認/すべての問題のトリガーで制限（デフォルトは CountAcknowledgeStatus=all）

6 サービス監視

概要　サービス監視は、IT インフラ全体のサービスツリーの概要を把握し、インフラの弱点を特定し、さまざまな IT サービスの SLA を計算し、より高いレベルで他の情報を確認するために使用できるビジネスレベルの監視です。サービス監視は、ディスク容量不足やプロセッサ負荷の増大などの低レベルの詳細ではなく、サービス全体の可用性に焦点を当てています。サービス監視は、サービスが期待通りに動作していない場合に問題の根本原因を特定する機能も提供します。

サービス監視では、監視データの階層構造を作成できます。

非常にシンプルなサービス構造は次のようになります。

```
サービス
|
|-ワークステーション
| |
| |-ワークステーション1
| |
| |-ワークステーション2
|
|-サーバー
```

構造の各ノードにはステータス属性があります。ステータスは選択したアルゴリズムに従って計算され、上位レベルに伝播されます。個々のノードのステータスは、マッピングされた問題のステータスによって影響を受けます。問題のマッピングはタグ付けで行います。

Zabbix は、サービスステータスの変更が検出された場合に通知を送信したり、Zabbix サーバー上でスクリプトを自動的に実行したりできます。親サービスが子サービスのステータスに基づいて「問題状態」になるかどうかを柔軟なルールで定義することができます。サービスの問題データは、SLA の計算や柔軟な条件セットに基づく SLA レポートの送信に使用できます。

サービス監視は、以下のセクションで構成されるサービスメニューで設定します。

- サービス

サービスセクションでは、親サービスを追加し、親サービスに子サービスを追加することで、監視対象インフラの階層を構築できます。

サービスツリーの設定に加えて、このセクションではインフラ全体の概要を提供し、サービスステータスの変更につながった問題を迅速に特定できます。

- SLA

このセクションでは、サービスレベル契約を定義し、特定のサービスのサービスレベル目標を設定できます。

- SLA レポート

このセクションでは、SLA レポートを表示できます。

サービスアクション

サービスアクションも設定できます。

サービスアクションは任意であり、以下を可能にします。

- サービスがダウンしたことを通知する
- サービスステータスの変更時に Zabbix サーバーでリモートコマンドを実行する
- サービスが再び稼働したときに復旧通知を送信する

関連情報**:**

- SLA 監視設定の例
- Zabbix 6.0 未満のバージョンからのサービスアップグレードに関する注意事項

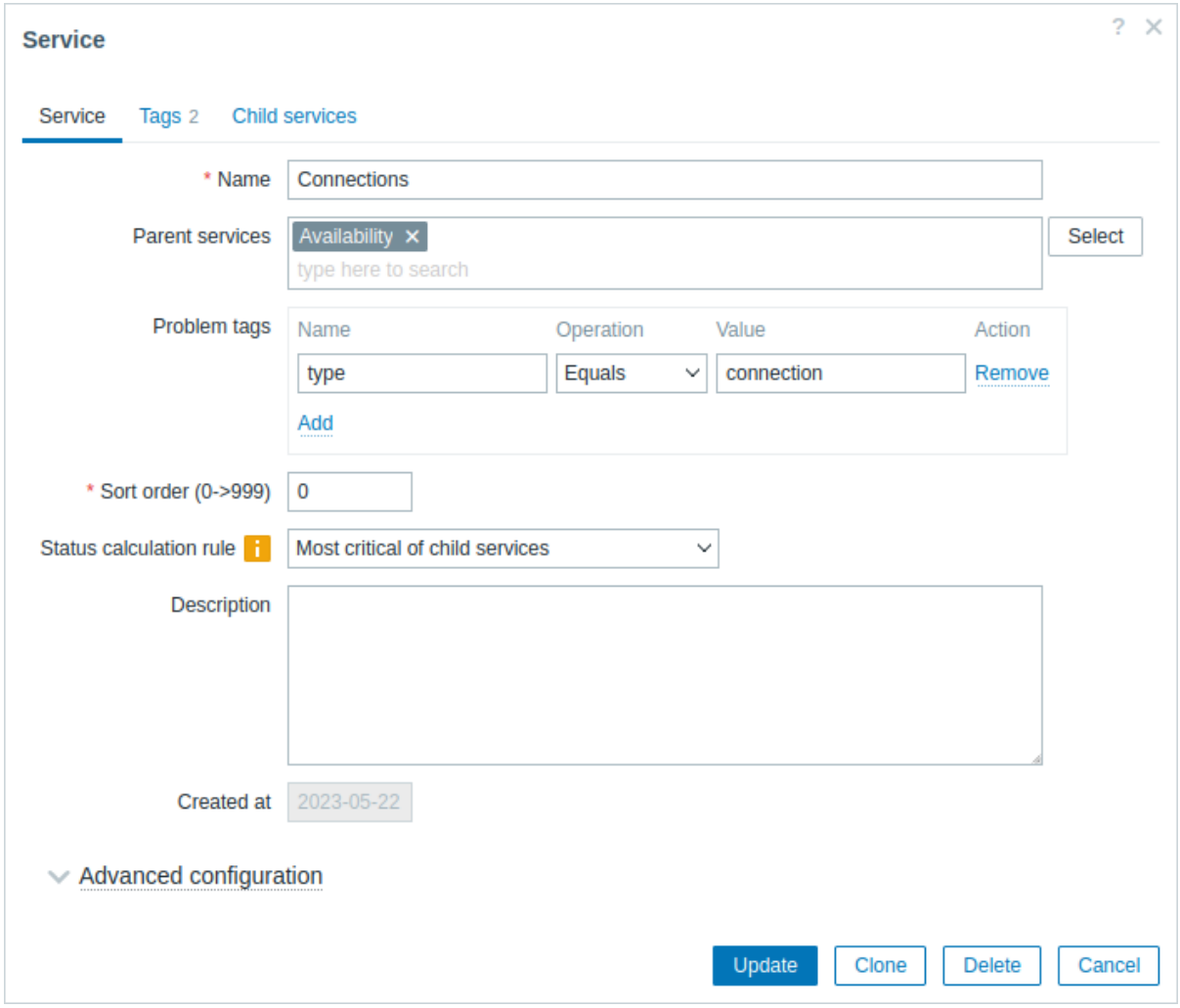
1 サービストリー

サービストリーは、サービス -> サービスメニューセクションで設定します。右上隅で表示から編集モードに切り替えます。

新しいサービスを設定するには、右上隅の サービスの作成ボタンをクリックします。

子サービスをすばやく追加するには、親サービスの横にあるプラスアイコンを押すこともできます。これにより、同じサービス設定フォームが開きますが、親サービスパラメータが事前に入力されます。

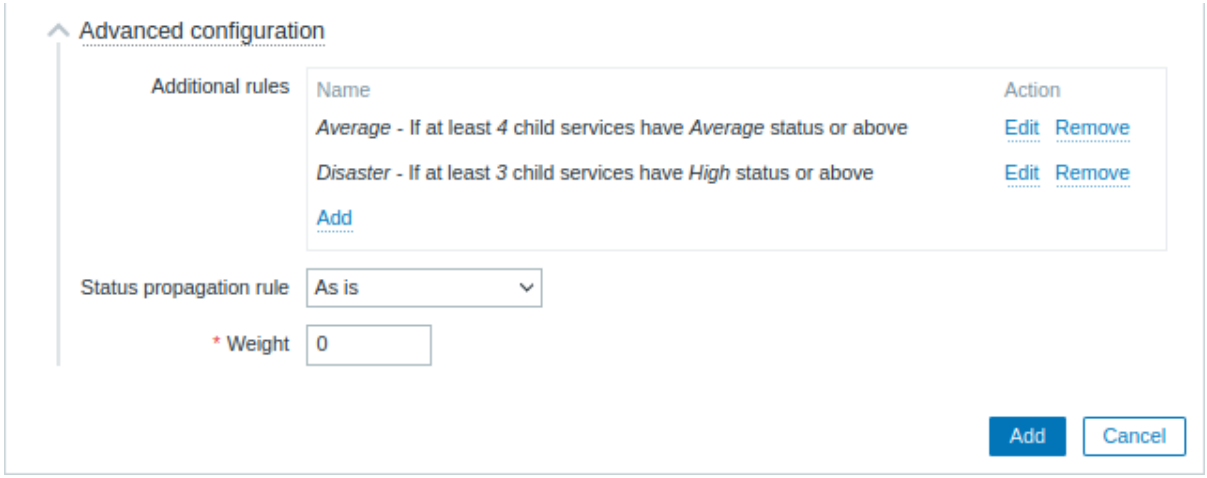
サービスの設定　サービスタブで、必要なサービスパラメータを指定します。



すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	サービス名。
親サービス	サービスが属する親サービス。 最上位のサービスを追加する場合は、このフィールドを空のままにします。 1 つのサービスに複数の親サービスを設定できます。この場合、サービスツリーの各親サービスの下に表示されます。
問題タグ	サービスに問題データをマッピングするためのタグを指定します。 複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件には 2 つの演算子があります: 等しい - 指定したタグ名と値を含める（大文字と小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字と小文字を区別）
表示順序	表示のための並び順。値が小さいほど先に表示されます。
ステータス計算ルール	サービスステータスを計算するルール: すべての子サービスに問題がある場合は最も重大なもの - すべての子サービスに問題がある場合、子サービスの中で最も重大な問題でサービスステータスを色付けします 子サービスの中で最も重大なもの - 子サービスの中で最も重大な問題でサービスステータスを色付けします ステータスを **OK** に設定 - サービスステータスを計算しません 追加のステータス計算ルールは、詳細設定オプションで設定できます。
説明	サービスの説明。
作成日時	サービスが作成された日時。既存のサービスを編集する場合に表示されます。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。

高度な設定



パラメータ	説明
追加ルール	追加をクリックして、追加のステータス計算ルールを設定します。
ステータスを設定	条件が一致した場合、サービスのステータスを *OK*（デフォルト）、未分類、情報、警告、軽度障害、重度障害、致命的障害のいずれかに設定します。
条件	直接の子サービスの条件を選択します： 少なくとも **(N)** 個の子サービスが **(Status)** ステータス以上の場合 少なくとも **(N%)** の子サービスが **(Status)** ステータス以上の場合 **(N)** 個未満の子サービスが **(Status)** ステータス以下の場合 **(N%)** 未満の子サービスが **(Status)** ステータス以下の場合 **(Status)** ステータス以上の子サービスの重みが少なくとも **(W)** の場合 **(Status)** ステータス以上の子サービスの重みが少なくとも **(N%)** の場合 **(Status)** ステータス以下の子サービスの重みが **(W)** 未満の場合 **(Status)** ステータス以下の子サービスの重みが **(N%)** 未満の場合 複数の条件が指定され、状況が複数の条件に一致する場合は、最も高い深刻度が設定されます。
N (W)	条件の N または W（1-100000）、または N%（1-100）の値を設定します。
ステータス	条件のステータスの値を選択します：*OK*（デフォルト）、未分類、情報、警告、軽度障害、重度障害、致命的障害。
ステータス伝播ルール	サービスのステータスを親サービスに伝播するルール： そのまま - ステータスは変更せずに伝播されます 増加 - 伝播されるステータスを 1 から 5 段階増加できます 減少 - 伝播されるステータスを 1 から 5 段階減少できます このサービスを無視 - ステータスは親サービスに全く伝播されません 固定ステータス - ステータスは静的に伝播され、常に同じになります
重み	サービスの重み（0（デフォルト）から 1000000 までの整数）。

Note:
追加のステータス計算ルールは、メインのステータス計算ルールパラメータで計算されたレベルよりも深刻度レベルを上げるためにのみ使用できます。追加ルールによるとステータスが警告であるべき場合でも、ステータス計算ルールによるとステータスが致命的障害である場合、サービスは致命的障害のステータスになります。

Tags タブにはservice-level tagsが含まれています。サービスレベルのタグは、サービスを識別するために使用されます。このタイプのタグは問題をサービスにマップするためには使用されません。（そのためには、最初のタブから *Problem tags(#problem-tags)* を使用します）

Child services タブでは、依存サービスを指定できます。*Add* をクリックして、既存のサービスのリストからサービスを追加します。新しい子サービスを追加する場合は、最初にこのサービスを保存してから、作成したサービスの横にあるプラスアイコンをクリックします。

タグ　サービスには 2 種類のタグがあります:

- サービスタグ
- 問題タグ

サービスタグ

サービスタグは、サービスアクションやSLAとサービスをマッピングするために使用されます。これらのタグは、タグサービス設定タブで指定します。SLA のマッピングには *OR* ロジックが使用されます: サービスが少なくとも 1 つの一致するタグを持っていれば、

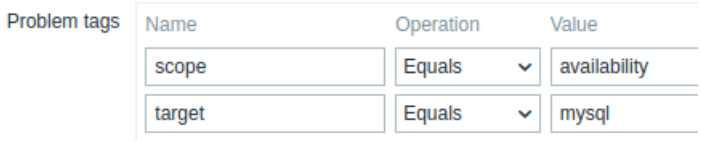
SLA にマッピングされます。サービスアクションでは、マッピングルールは設定可能で、*AND*、*OR*、または *AND/OR* ロジックのいずれかを使用できます。

問題タグ

問題タグは、問題とサービスを一致させるために使用されます。これらのタグは、主要なサービス設定タブで指定されます。

最下層の子サービスのみが問題タグを定義でき、問題に直接関連付けることができます。問題タグが一致すると、サービスのステータスは問題のステータスと同じに変更されます。複数の問題がある場合、サービスは最も深刻な問題のステータスを持ちます。親サービスのステータスは、ステータス計算ルールに従って子サービスのステータスに基づいて計算されます。

複数のタグが指定されている場合、*AND* ロジックが使用されます。サービスにマッピングされるには、問題がサービス設定で指定されたすべてのタグを持っている必要があります。



Note:
Zabbix の問題は、テンプレート、ホスト、アイテム、Web シナリオ、トリガーの全チェーンからタグを継承します。これらのタグのいずれかを使用して、問題とサービスを一致させることができます。

例:

問題 *Web camera 3 is down* には、タグ `type:video-surveillance`、`floor:1`、`name:webcam-3` があり、ステータスは *Warning* です。

サービス **Web camera 3** には、指定された問題タグが 1 つだけあります: `name:webcam-3`



この問題が検出されると、サービスのステータスは *OK* から *Warning* に変わります。

もしサービス **Web camera 3** に問題タグ `name:webcam-3` と `floor:2` があった場合、条件が部分的にしか満たされないため、問題が検出されてもステータスは変更されません。

既存サービスの修正

Note:
以下に説明するボタンは、サービスセクションが編集モードのときのみ表示されます。

既存のサービスを編集するには、サービスの横にある鉛筆アイコンを押します。

既存のサービスをクローンするには、鉛筆アイコンを押して設定を開き、クローンボタンを押します。サービスをクローンすると、親リンクは保持されますが、子リンクは保持されません。

サービスを削除するには、その横にある x アイコンを押します。親サービスを削除すると、その子サービスは削除されず、サービスツリーで 1 つ上のレベルに移動します（第 1 レベルの子サービスは、削除された親サービスと同じレベルになります）。

サービス一覧の下にある 2 つのボタンで、一括編集オプションが利用できます。

- 一括更新 - サービスプロパティの一括更新
- 削除 - サービスの削除

これらのオプションを使用するには、該当するサービスの前のチェックボックスをマークし、必要なボタンをクリックします。

2 SLA

概要　サービスが作成されたら、サービスレベルアグリーメント（SLA）に沿ってパフォーマンスが順調かどうかを監視できます。

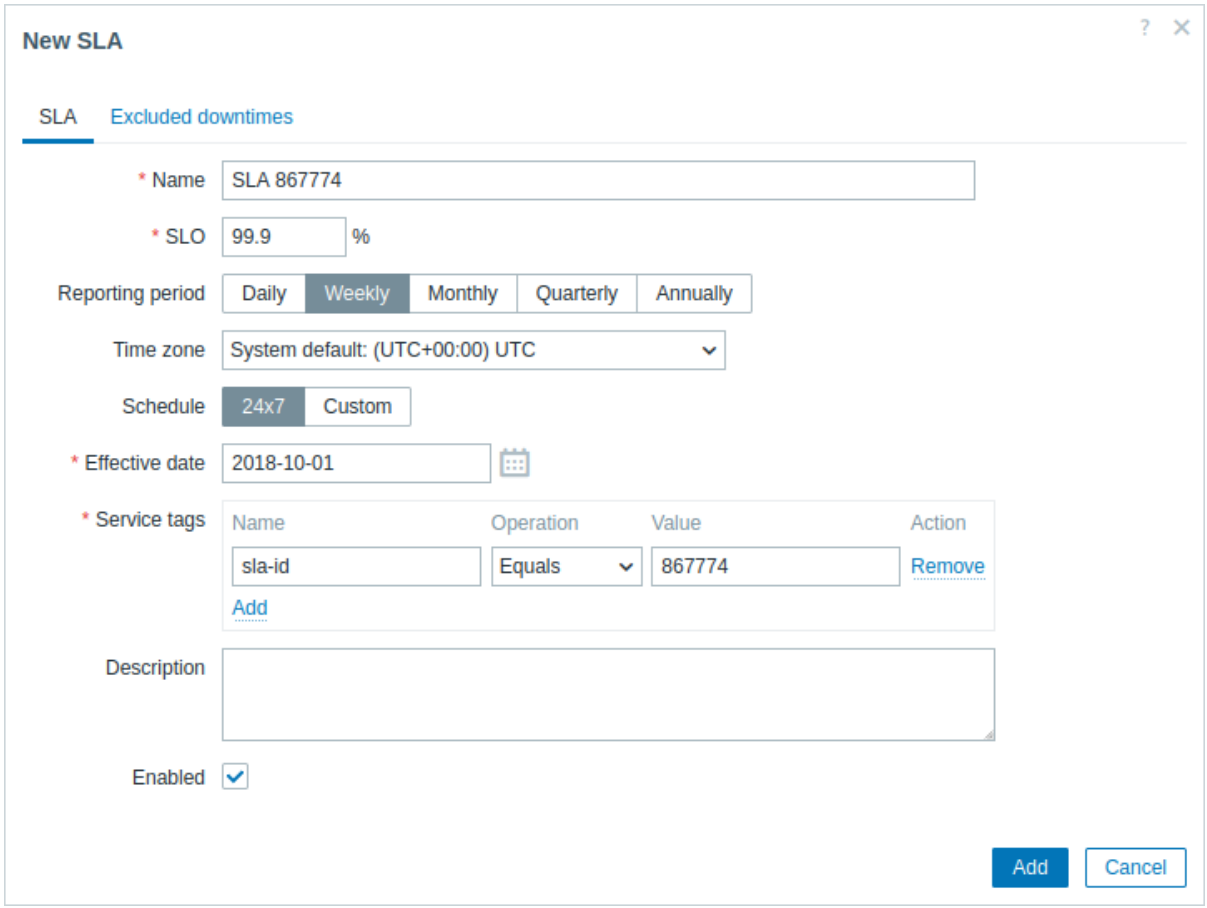
サービス->*SLA* メニューセクションでは、さまざまなサービスの SLA を設定できます。Zabbix の SLA は、サービスレベル目標（SLO）、期待される稼働スケジュール、および計画されたダウンタイムを定義します。

SLA とサービスはサービスタグで一致します。同じ SLA を複数のサービスに適用することができ、パフォーマンスは一致する各サービスごとに個別に測定されます。1 つのサービスに複数の SLA を割り当てることもでき、各 SLA のデータは個別に表示されます。

SLA レポートでは、Zabbix はサービスレベルインジケーター（SLI）データを提供し、実際のサービス可用性を測定します。サービスが SLA 目標を満たしているかどうかは、SLO（期待される可用性%）と SLI（実際の可用性%）を比較して判断されます。

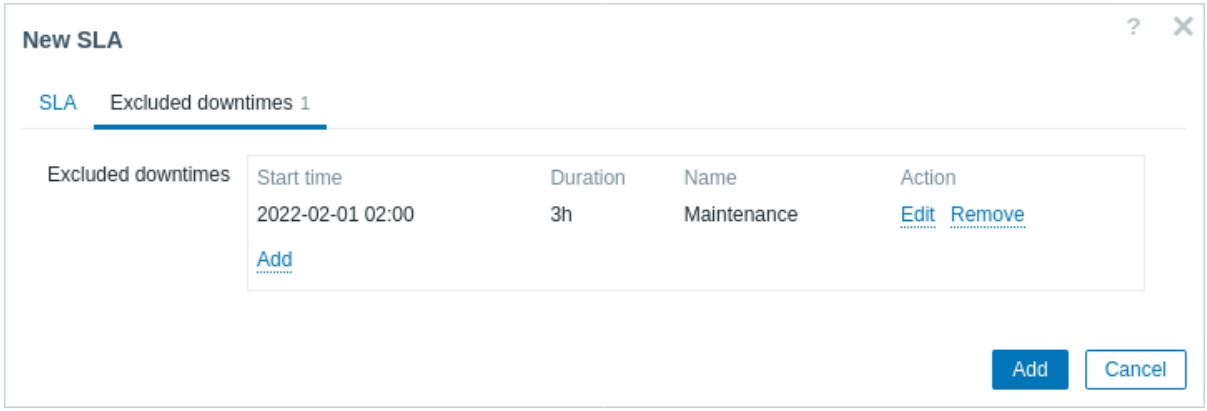
設定　新しい SLA を作成するには、*SLA* の作成ボタンをクリックします。

SLA タブでは、一般的な SLA パラメータを指定できます。



パラメータ	説明
名前	SLA 名を入力します。
SLO	サービスレベル目標（SLO）をパーセンテージで入力します。
レポート期間	SLA レポートで SLI データを計算するために使用される SLI レポート期間を選択します。 日次 - 毎日、00:00:00 から 23:59:59 まで。 週次 - 毎週、日曜日 00:00:00 から土曜日 23:59:59 まで。 月次 - 毎月、1 日 00:00:00 から最終日 23:59:59 まで。 四半期 - 各暦四半期：Q1（1 月～3 月）、Q2（4 月～6 月）、Q3（7 月～9 月）、Q4（10 月～12 月）。 年次 - 各暦年、1 月 1 日 00:00:00 から 12 月 31 日 23:59:59 まで。
タイムゾーン	SLA のタイムゾーンを選択します。
スケジュール	SLA スケジュール（24x7 またはカスタム）を選択します。
有効日	SLA 計算の開始日を選択します。
サービスタグ	この SLA を適用するサービスを識別するためのサービスタグを追加します。 名前 - サービスタグ名。完全一致、大文字小文字を区別します。 操作 - タグ値が完全一致（大文字小文字を区別）する場合は等しい、タグ値の一部が一致（大文字小文字を区別しない）する場合は含むを選択します。 値 - 選択した操作に従って検索するサービスタグ値。 少なくとも 1 つのサービスタグが一致する場合、SLA はサービスに適用されます。
説明	SLA の説明を追加します。
有効	チェックボックスをオンにすると、SLA 計算が有効になります。

除外ダウンタイムタブでは、SLA 計算から除外するダウンタイムを指定できます。



追加をクリックして除外ダウンタイムを設定し、期間名、開始日、期間を入力します。

SLA レポート　サービスが SLA と比較してどのようにパフォーマンスを発揮しているかは、SLA レポートで確認できます。SLA レポートは以下から表示できます。

- *SLA* セクションで SLA レポートのハイパーリンクをクリック
- サービスセクションで情報タブの SLA 名をクリック
- ダッシュボードのウィジェット *SLA* レポートで表示

SLA が設定されると、サービスセクションの情報タブにもサービスパフォーマンスに関する情報が表示されます。

3 セットアップ例

概要　このセクションでは、Zabbix 高可用性クラスターをサービスとして監視するためのセットアップについて説明します。

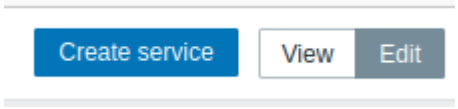
前提条件　サービス監視を設定する前に、以下のホストが設定されている必要があります。

- *HA* ノード *1*：少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-1`
- *HA* ノード *2*：少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-2`

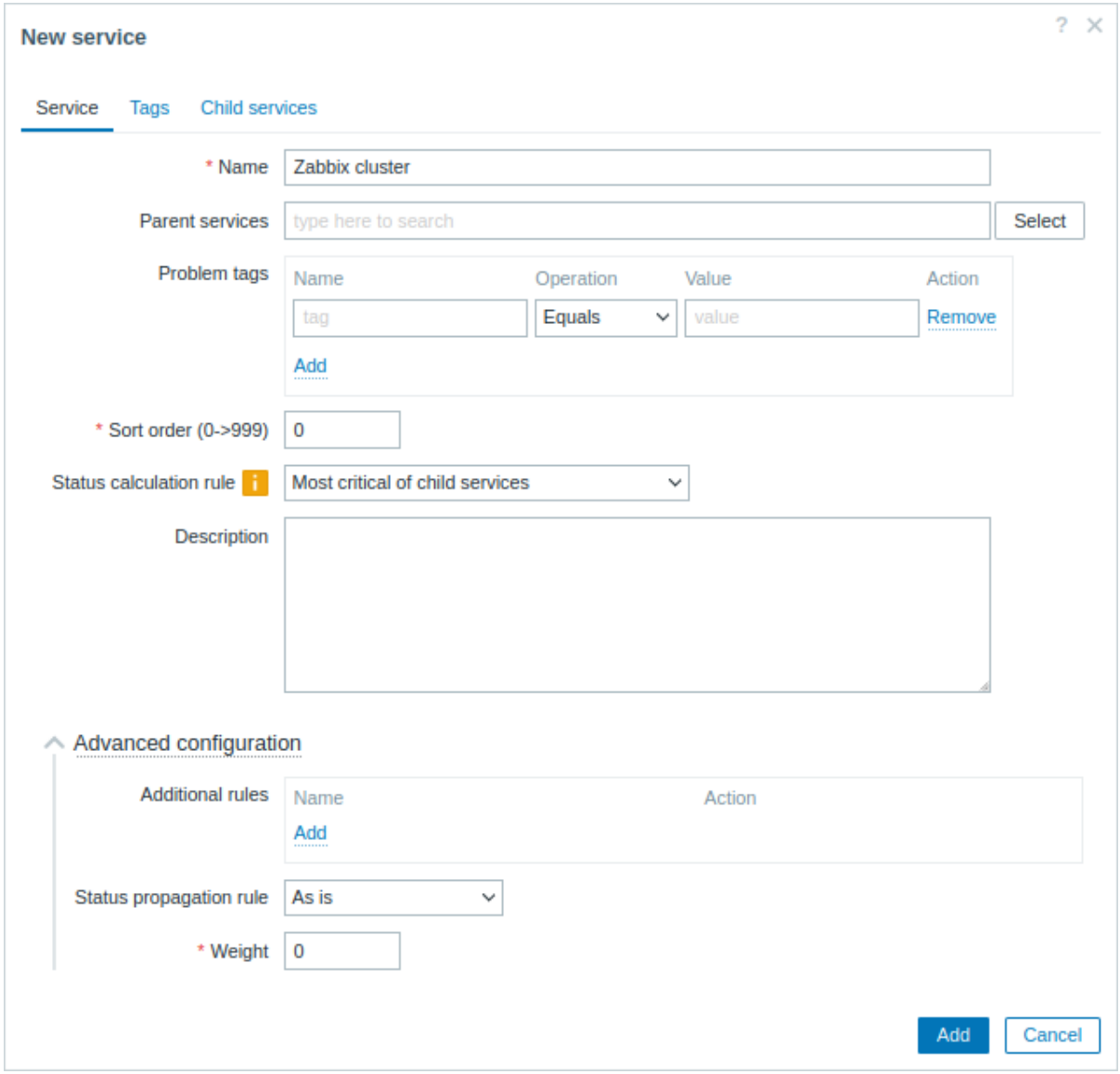
サービストリー　次のステップは、サービストリーを構築することです。この例では、インフラストラクチャは非常に基本的なもので、3 つのサービスで構成されています。*Zabbix* クラスタ（親）と、2 つの子サービス *Zabbix* サーバーノード *1*、*Zabbix* サーバーノード *2* です。

```
Zabbixクラスタ
|
|- Zabbixサーバーノード1
|- Zabbixサーバーノード2
```

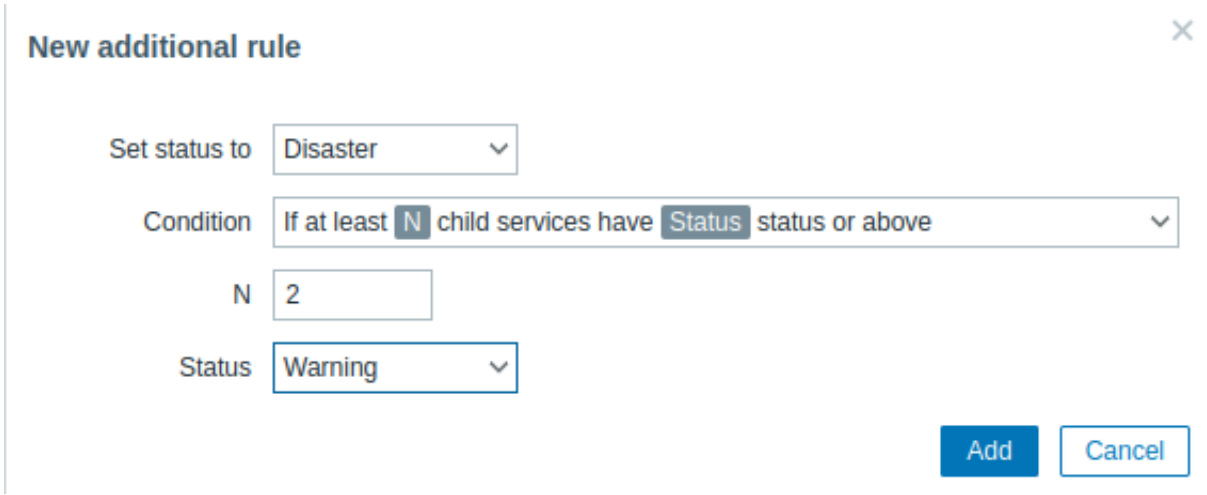
サービスページで編集モードをオンにし、サービスの作成をクリックします。



サービスの設定ウィンドウで、名前に *Zabbix* クラスタと入力し、詳細設定ラベルをクリックして詳細設定オプションを表示します。



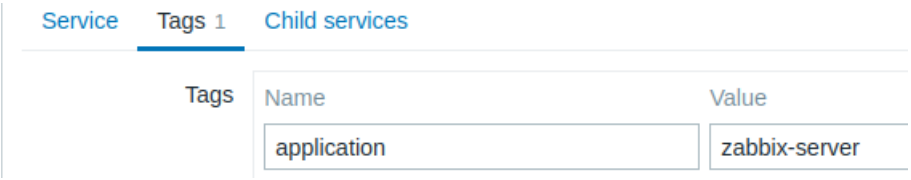
追加ルールを設定します。



Zabbix クラスタには、HA ノードごとに 1 つずつ、2 つの子サービスがあります。両方の HA ノードに警告以上の問題がある場合、親サービスのステータスは障害に設定されるべきです。これを実現するために、追加ルールを次のように設定します。

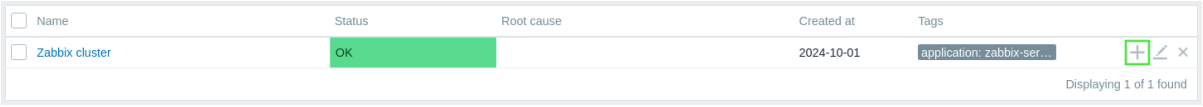
- ステータスを設定: 障害
- 条件: N 個以上の子サービスがステータス以上の場合
- N: 2
- ステータス: 警告

タグタブに切り替え、タグ `application:zabbix-server` を追加します。このタグは、後でサービスアクションや SLA レポートで使用します。



新しいサービスを保存します。

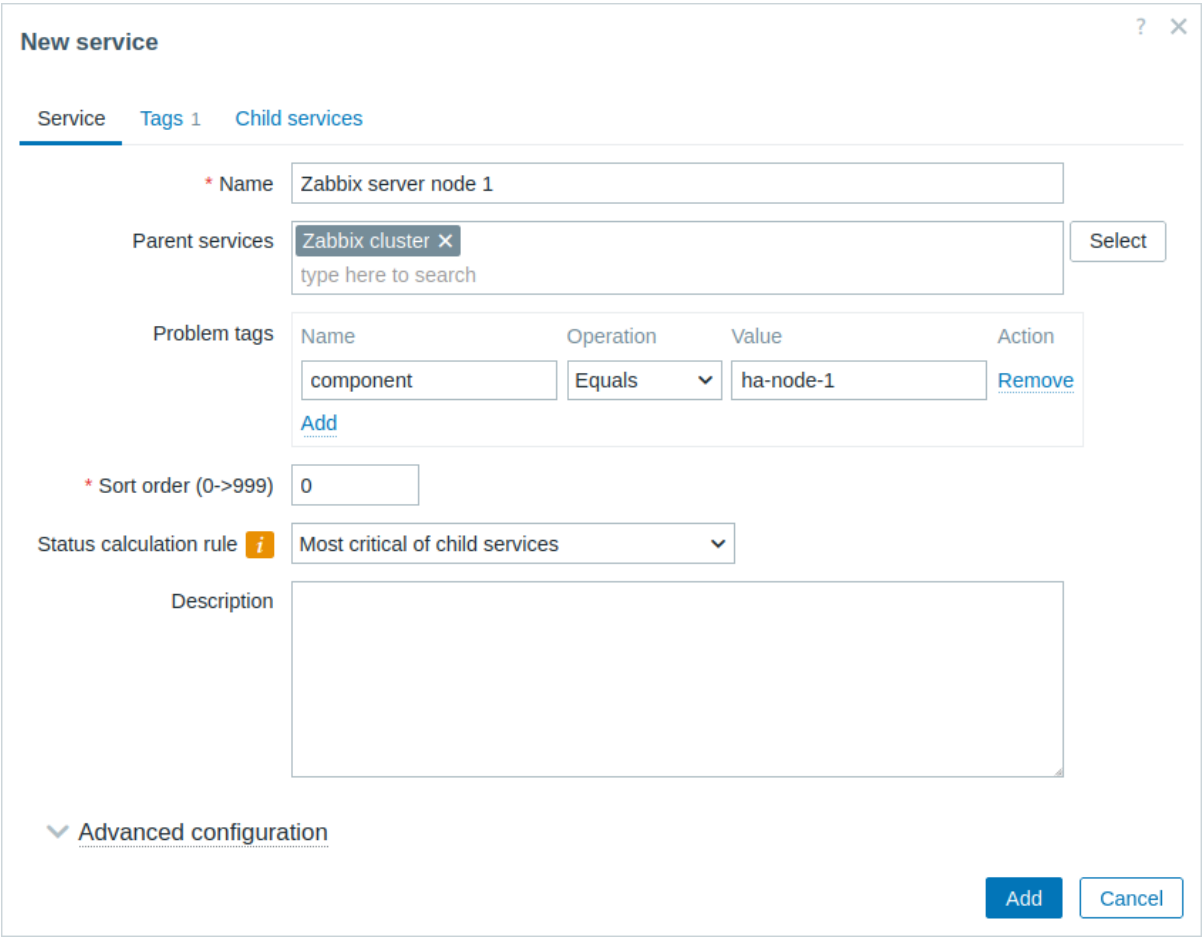
子サービスを追加するには、Zabbix クラスタサービスの横にあるプラスアイコンを押します（このアイコンは編集モードでのみ表示されます）。



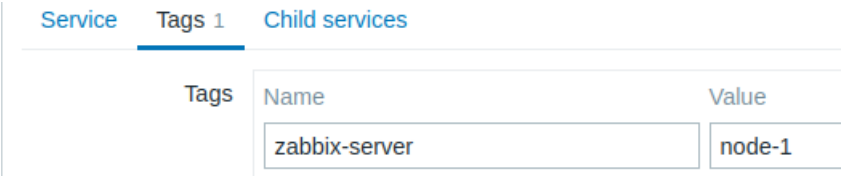
サービス設定ウィンドウで、名前に *Zabbix server node 1* と入力します。親サービスパラメータにはすでに *Zabbix cluster* が入力されていることに注意してください。

このサービスの可用性は、ホスト *HA node 1* の問題によって影響を受けます。問題タグ `component:ha-node-1` が付与されます。問題タグパラメータには以下を入力します。

- 名前: component
- 演算子: 等しい
- 値: ha-node-1



タグタブに切り替え、サービスタグ `zabbix-server:node-1` を追加します。このタグは後でサービスアクションや SLA レポートで使用します。



新しいサービスを保存します。

「Zabbix server node 2」という名前で Zabbix クラスタの別の子サービスを作成します。

問題タグを次のように設定します。

- 名前: component
- 演算子: Equals
- 値: ha-node-2

タグタブに切り替え、サービスタグ `zabbix-server:node-2` を追加します。

新しいサービスを保存します。

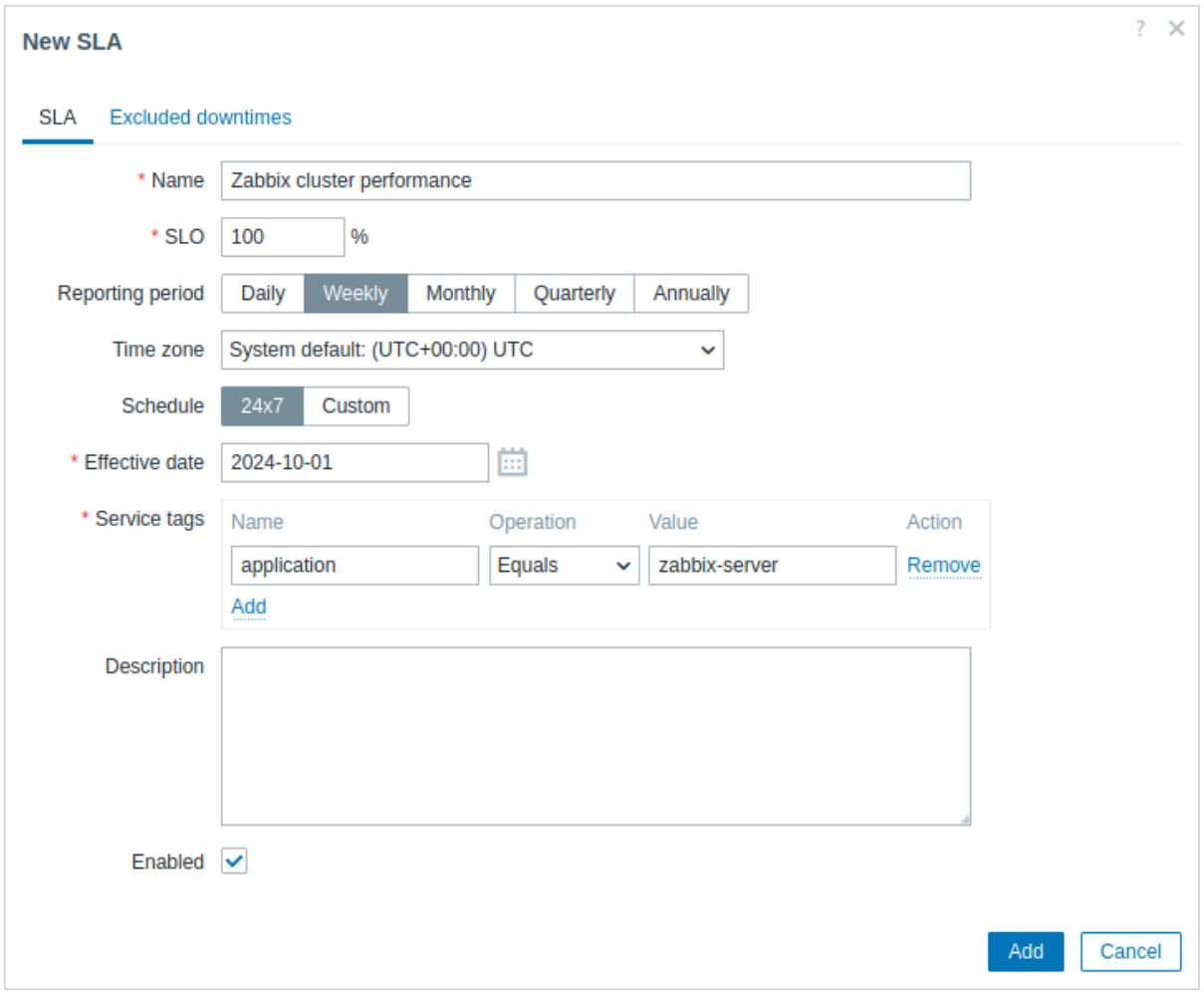
SLA　この例では、半年ごとの 1 時間のメンテナンス期間を除き、Zabbix クラスタの期待されるパフォーマンスは 100% です。

まず、新しいサービスレベルアグリーメントを追加する必要があります。

サービス -> *SLA* メニューセクションに移動し、「SLA の作成」を押します。名前に *Zabbix* クラスタのパフォーマンスを入力し、SLO を 100% に設定します。

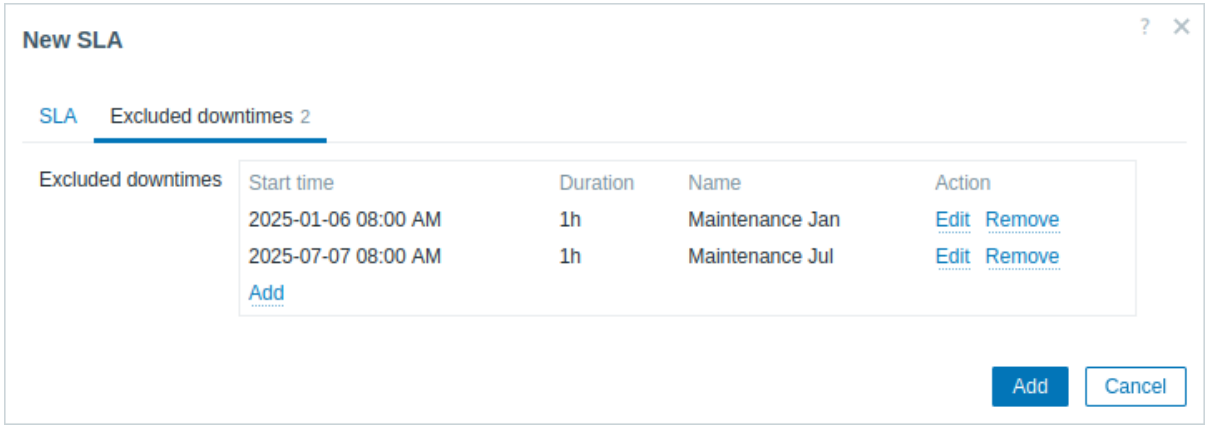
サービス Zabbix クラスタにはサービスタグ `application:zabbix-server` があります。この SLA を Zabbix クラスタのパフォーマンス測定に使用するには、サービスタグパラメータで以下を指定します。

- 名前: application
- 演算子: 等しい
- 値: zabbix-server



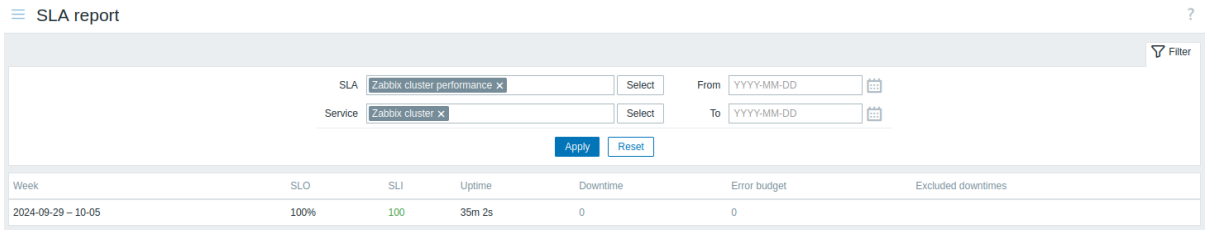
実際の運用では、必要に応じてレポート期間、タイムゾーン、開始日を更新したり、スケジュールを 24/7 からカスタムに変更したりできます。この例では、デフォルト設定で十分です。

除外するダウンタイムタブに切り替え、スケジュールされたメンテナンス期間のダウンタイムを追加して、これらの期間を SLA 計算から除外します。[除外するダウンタイム] セクションで「追加」リンクを押し、ダウンタイム名、予定開始時刻、期間を入力します。

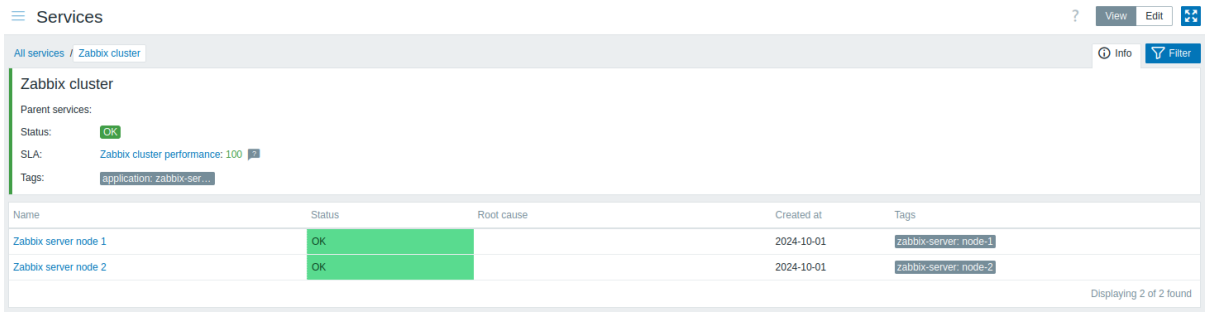


「追加」を押して新しい SLA を保存します。

SLA レポートセクションに切り替えて、Zabbix クラスタの SLA レポートを表示します。



SLA 情報は サービスセクションでも確認できます。



7 Web 監視

概要　Zabbix を使用すると、Web サイトのさまざまな可用性の側面をチェックできます。

Attention:
Web 監視を実行するには、Zabbix サーバーが最初に cURL（libcurl）サポート付きで構成されている必要があります。

Web 監視を有効にするには、Web シナリオを定義する必要があります。Web シナリオは、1 つまたは複数の HTTP リクエストまたは「ステップ」で構成されます。ステップは、Zabbix サーバーによって事前定義された順序で定期的に実行されます。ホストがプロキシによって監視されている場合、ステップはプロキシによって実行されます。

Web シナリオは、アイテムやトリガーなどと同じ方法でホスト/テンプレートに関連付けられます。つまり、Web シナリオはテンプレートレベルでも作成でき、複数のホストに一度に適用できます。

任意の Web シナリオで収集される情報は次のとおりです。

- シナリオ全体のすべてのステップの 1 秒あたりの平均ダウンロード速度
- 失敗したステップの番号
- 最後のエラーメッセージ

任意の Web シナリオステップで収集される情報は次のとおりです。

- 1 秒あたりのダウンロード速度
- 応答時間
- 応答コード

詳細については、Web 監視アイテムを参照してください。

Web シナリオの実行から収集されたデータはデータベースに保持されます。このデータは自動的にグラフ、トリガー、通知に使用されます。

Zabbix は、取得した HTML ページに事前定義された文字列が含まれているかどうかもチェックできます。シミュレートされたログインを実行し、ページ上のシミュレートされたマウスクリックのパスをたどることができます。

Zabbix の Web 監視は、HTTP と HTTPS の両方をサポートしています。Web シナリオを実行するとき、Zabbix はオプションでリダイレクトを追跡します（下記のリダイレクトを追跡オプションを参照）。リダイレクトの最大数は 10 にハードコードされています (cURL オプションCURLOPT_MAXREDIRSを使用)。すべての Cookie は、単一のシナリオの実行中に保持されます。

Web シナリオの設定　Web シナリオを設定するには:

- データ収集 → ホスト（またはテンプレート）に移動します
- ホスト/テンプレートの行で *Web* をクリックします
- 右側の *Web* シナリオの作成をクリックします（またはシナリオ名をクリックして既存のシナリオを編集します）
- フォームにシナリオのパラメータを入力します

シナリオタブでは、Web シナリオの一般的なパラメータを設定できます。

必須入力フィールドは赤いアスタリスクでマークされています。

シナリオパラメータ:

パラメータ	説明
名前	一意のシナリオ名。 ユーザーマクロがサポートされています。注意：ユーザーマクロを使用した場合、これらのマクロはWeb 監視アイテム名で未解決のままになります。
更新間隔	シナリオが実行される頻度。 時間のサフィックスがサポートされています（例：30s、1m、2h、1d)。 ユーザーマクロがサポートされています。注意：ユーザーマクロを使用し、その値を変更した場合 (例：5m → 30s)、次のチェックは前の値に従って実行されます（例の値では将来のより遅いタイミングで実行されます)。 新しい Web シナリオは作成から 60 秒以内にチェックされます。
試行回数	Web シナリオステップの実行試行回数。ネットワークの問題（タイムアウト、接続不可など）の場合、Zabbix はステップの実行を複数回繰り返すことができます。設定した数値はシナリオの各ステップに等しく適用されます。最大 10 回まで指定でき、デフォルト値は 1 です。 注意：Zabbix は、応答コードが間違っている場合や必要な文字列が一致しない場合にはステップを繰り返しません。
エージェント	クライアントエージェントを選択します。 Zabbix は選択したブラウザになりすまします。これは、Web サイトが異なるブラウザに対して異なるコンテンツを返す場合に便利です。 このフィールドではユーザーマクロが使用できます。
HTTP プロキシ	使用する HTTP プロキシを `[protocol://][username[:password]@]proxy.example.com[:port]` の形式で指定できます。 これはCURLOPT_PROXY cURL オプションを設定します。 オプションの `protocol://`プレフィックスを使用して代替プロキシプロトコルを指定できます (プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました)。プロトコルが指定されていない場合、プロキシは HTTP プロキシとして扱われます。 デフォルトでは、1080 ポートが使用されます。 指定した場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連の環境変数を上書きします。指定しない場合、プロキシはプロキシ関連の環境変数を上書きしません。入力された値は「そのまま」渡され、整合性チェックは行われません。 SOCKS プロキシアドレスも入力できます。間違ったプロトコルを指定すると、接続は失敗し、アイテムはサポート対象外になります。 注意：HTTP プロキシではシンプル認証のみサポートされています。 このフィールドではユーザーマクロが使用できます。

パラメータ	説明
変数	シナリオステップ（URL、POST 変数）で使用できる変数。 次の形式で指定します： **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:< 正規表現 > **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()}（マクロ関数を参照） 例： {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} これらのマクロは、ステップ内で {username}、{password}、{hostid} などとして参照できます。Zabbix は自動的に実際の値に置き換えます。`regex:` 付きの変数は、正規表現の値を取得するために 1 ステップ必要なので、抽出した値は次のステップでのみ適用できます。 値の部分が `regex:` で始まる場合、その後の部分は Web ページを検索する正規表現として扱われ、見つかった場合は一致した値が変数に格納されます。少なくとも 1 つのサブグループが必要です。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。 変数は、クエリフィールドや POST 変数のフォームデータで使用する場合は自動的に URL エンコードされますが、raw post や URL で直接使用する場合は手動で URL エンコードする必要があります。
ヘッダー	リクエスト実行時に使用される HTTP ヘッダー。デフォルトおよびカスタムヘッダーが使用できます。 ヘッダーは、シナリオレベルでドロップダウンリストから選択したエージェントタイプに応じてデフォルト設定で割り当てられ、すべてのステップに適用されます。ただし、ステップレベルでカスタム定義されている場合は除きます。 ステップレベルでヘッダーを定義すると、シナリオレベルで以前に定義されたすべてのヘッダーが自動的に破棄されます。ただし、シナリオレベルでドロップダウンリストから「**User-Agent**」を選択して割り当てたデフォルトヘッダーは除きます。 ただし、「User-Agent」デフォルトヘッダーもステップレベルで指定することで上書きできます。 シナリオレベルでヘッダーを解除するには、ステップレベルでヘッダー名を指定し、値を空にします。 ヘッダーは、HTTP プロトコルで表示されるのと同じ構文でリストし、CURLOPT_HTTPHEADER cURL オプションでサポートされている追加機能をオプションで使用できます。 例： Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 ユーザーマクロおよび {HOST.*} マクロがサポートされています。
有効	このボックスにチェックを入れるとシナリオが有効になります。チェックを外すと無効になります。

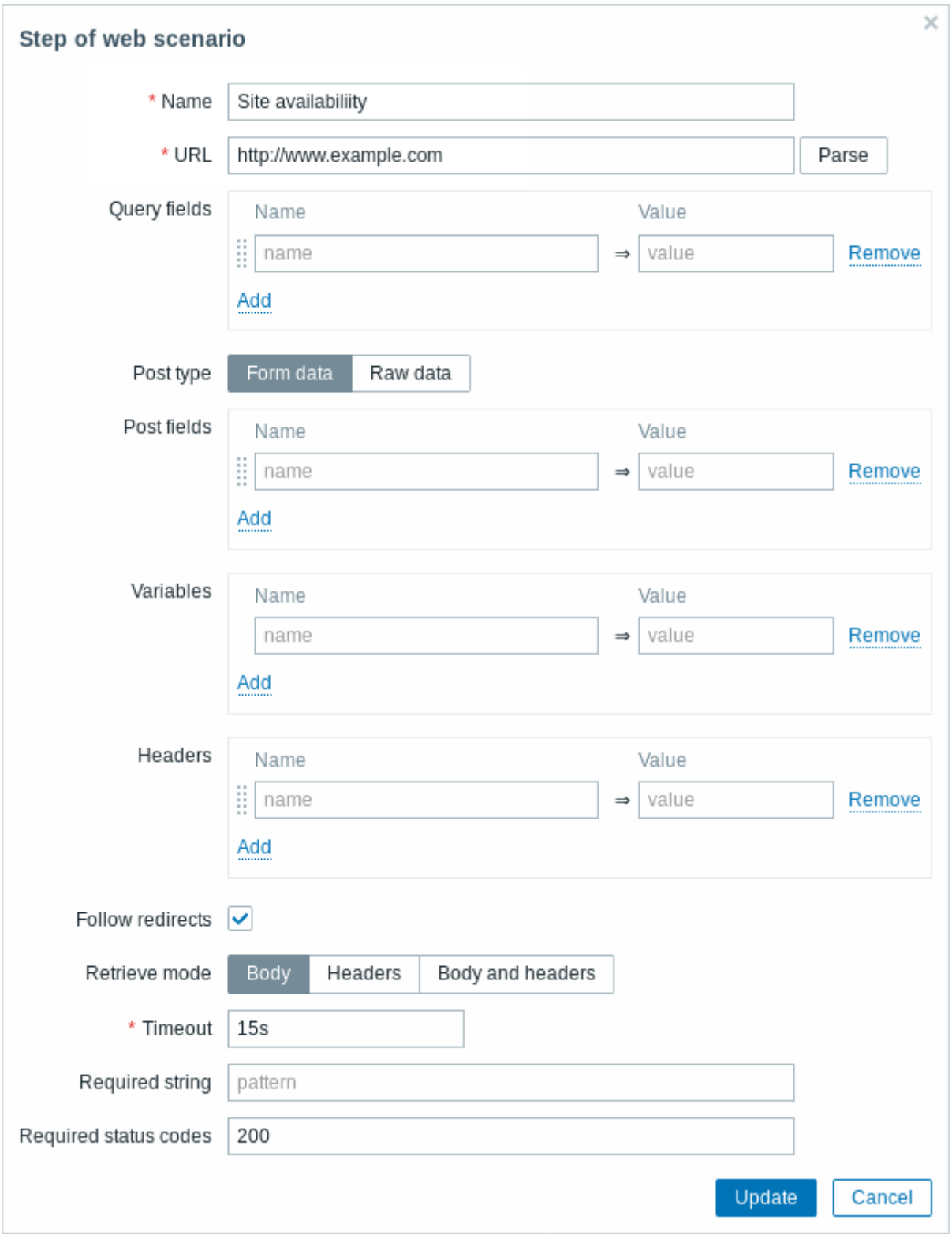
既存のシナリオを編集する場合、フォームに 2 つの追加ボタンが表示されます。

Clone	既存のシナリオのプロパティを元に別のシナリオを作成します。
Clear history and trends	シナリオの履歴およびトレンドデータを削除します。これにより、データ削除後すぐにサーバーがシ

Note:
HTTP プロキシフィールドを空のままにした場合、HTTP プロキシを使用する別の方法としてプロキシ関連の環境変数を設定できます。
HTTP チェックの場合は、Zabbix サーバーユーザーの **http_proxy** 環境変数を設定します。例：`http_proxy=http://proxy_ip:proxy_port`。
HTTPS チェックの場合は、**HTTPS_PROXY** 環境変数を設定します。例：`HTTPS_PROXY=http://proxy_ip:proxy_port`。
詳細はシェルコマンド：*# man curl* で確認できます。

ステップタブでは、Web シナリオのステップを設定できます。Web シナリオのステップを追加するには、ステップブロックで追加をクリックします。

Note:
秘密のユーザーマクロは、URL で使用しないでください。”******” に解決されます。



ステップの設定

ステップパラメータ:

パラメータ	説明
名前	一意のステップ名。 ユーザーマクロがサポートされています。注意：ユーザーマクロを使用した場合、これらのマクロはWeb 監視アイテム名では解決されません。

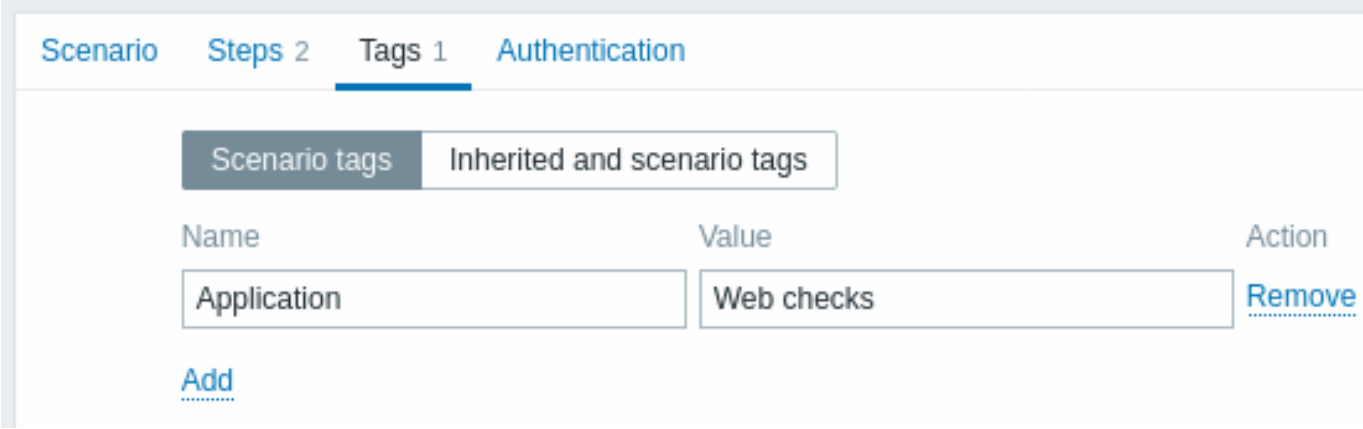
パラメータ	説明
URL	接続してデータを取得する URL。例： https://www.example.com http://www.example.com/download ドメイン名は Unicode 文字で指定できます。Web シナリオステップの実行時に自動的に punycode で ASCII に変換されます。 解析ボタンを使用すると、オプションのクエリフィールド（例：?name=Admin&password=mypassword）を URL から分離し、属性と値をクエリフィールドに移動して自動的に URL エンコードできます。 URL には {macro} 構文で変数を使用できます。変数を手動で URL エンコードするには、{{macro}.urlencode()} 構文を使用します。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。 2048 文字まで。
クエリフィールド	URL の HTTP GET 変数。 属性と値のペアで指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、{HOST.*} マクロの値は解決され、その後自動的に URL エンコードされます。{{macro}.urlencode()} 構文を使用すると、二重に URL エンコードされます。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。
Post	HTTP POST 変数。 フォームデータモードでは、属性と値のペアで指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、{HOST.*} マクロの値は解決され、その後自動的に URL エンコードされます。 生データモードでは、属性/値が 1 行で表示され、**&** 記号で連結されます。 生の値は、{{macro}.urlencode()} または {{macro}.urldecode()} 構文を使用して手動で URL エンコード/デコードできます。 例：id=2345&userid={user} {user} が Web シナリオの変数として定義されている場合、ステップ実行時にその値に置き換えられます。変数を URL エンコードしたい場合は、{user} を {{user}.urlencode()} に置き換えてください。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。
変数	GET および POST 関数で使用できるステップレベルの変数。 属性と値のペアで指定します。 ステップレベルの変数は、シナリオレベルの変数や前のステップの変数を上書きします。ただし、ステップレベルの変数の値は次のステップにのみ影響し（現在のステップには影響しません）。 以下の形式です： **{macro}**= 値 **{macro}**=regex:< 正規表現 > 詳細はシナリオレベルの変数説明を参照してください。 変数は、クエリフィールドや POST 変数のフォームデータで使用する場合は自動的に URL エンコードされますが、生の POST や URL で直接使用する場合は手動で URL エンコードする必要があります。
ヘッダー	リクエスト実行時に送信されるカスタム HTTP ヘッダー。 属性と値のペアで指定します。 ステップレベルで定義されたヘッダーは、その特定のステップで使用されます。 ステップレベルでヘッダーを定義すると、シナリオレベルでドロップダウンリストから**'User-Agent'** を選択して割り当てたデフォルトヘッダーを除き、以前に定義されたすべてのヘッダーが自動的に破棄されることに注意してください。 ただし、'User-Agent' のデフォルトヘッダーもステップレベルで指定することで上書きできます。 例えば、ヘッダーに名前を割り当てて値を設定しない場合、シナリオレベルのデフォルトヘッダーが解除されます。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。 これはCURLOPT_HTTPHEADER cURL オプションを設定します。
リダイレクトを追跡	HTTP リダイレクトを追跡する場合はチェックボックスをオンにします。 これはCURLOPT_FOLLOWLOCATION cURL オプションを設定します。
取得モード	取得モードを選択します： 本文 - HTTP レスポンスから本文のみ取得 ヘッダー - HTTP レスポンスからヘッダーのみ取得 本文とヘッダー - HTTP レスポンスから本文とヘッダーを取得

パラメータ	説明
タイムアウト	Zabbix は、URL の処理に設定された時間（1 秒から最大 1 時間）以上は費やしません。実際には、このパラメータは URL への接続の最大時間と HTTP リクエストの実行の最大時間を定義します。したがって、Zabbix はステップに **2 x** タイムアウト秒以上は費やしません。 時間のサフィックス（例：30s、1m、1h）がサポートされています。ユーザーマクロがサポートされています。
必須文字列	必須の正規表現パターン。 取得したコンテンツ（HTML）が必須パターンに一致しない場合、ステップは失敗します。空の場合、必須文字列のチェックは行われません。 例： Homepage of Zabbix Welcome.*admin 注意: このフィールドでは、Zabbix フロントエンドで作成した正規表現の参照はサポートされていません。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。
必須ステータスコード	期待される HTTP ステータスコードのリスト。Zabbix がリストにないコードを取得した場合、ステップは失敗します。 空の場合、ステータスコードのチェックは行われません。 例：200,201,210-299 ユーザーマクロがサポートされています。

Note:
Web シナリオステップの変更は、シナリオ全体を保存した場合にのみ保存されます。

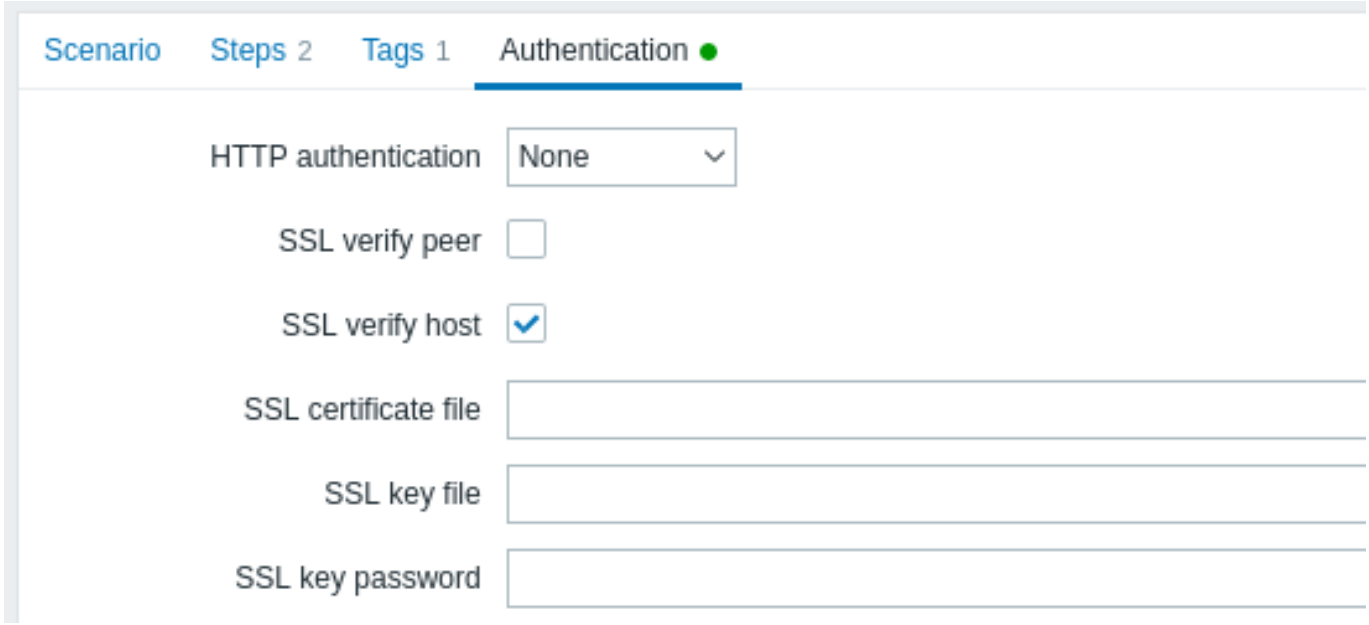
Web 監視ステップの設定方法については、実際の例も参照してください。

タグの構成　**Tags** タブでは、シナリオレベルのタグを定義できます。



タグ付けにより、Web シナリオと Web 監視アイテムをフィルタリングできます。

認証の設定　認証タブでは、シナリオの認証オプションを設定できます。タブ名の横に緑色のドットが表示されている場合は、何らかの HTTP 認証が有効になっていることを示します。



認証パラメータ:

パラメータ	説明
HTTP 認証	認証オプションを選択します: なし - 認証を使用しません; **Basic** - Basic 認証を使用します; **NTLM** - NTLM (Windows NT LAN Manager) 認証を使用します; **Kerberos** - Kerberos 認証を使用します（参考: Zabbix での Kerberos の設定）; **Digest** - Digest 認証を使用します。
ユーザー	ユーザー名を入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、Digest のいずれかに設定されている場合に利用できます。 ユーザーマクロがサポートされています。
パスワード	ユーザーパスワードを入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、Digest のいずれかに設定されている場合に利用できます。 ユーザーマクロがサポートされています。
SSL ピアの検証	Web サーバーの SSL 証明書を検証する場合はチェックボックスをオンにします。 サーバー証明書はシステム全体の証明書認証局（CA）の場所から自動的に取得されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータSSLCALocationで上書きできます。 この設定はCURLOPT_SSL_VERIFYPEERの cURL オプションを設定します。
SSL ホストの検証	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致するかどうかを検証する場合はチェックボックスをオンにします。 この設定はCURLOPT_SSL_VERIFYHOSTの cURL オプションを設定します。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイル名。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、*SSL* キーファイルフィールドを空にしてください。秘密鍵が暗号化されている場合は、*SSL* キーパスワードフィールドにパスワードを指定してください。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLCertLocationで指定します。 このフィールドでは `HOST.*` マクロおよびユーザーマクロが使用できます。 この設定はCURLOPT_SSLCERTの cURL オプションを設定します。
SSL キーファイル	クライアント認証に使用する SSL 秘密鍵ファイル名。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLKeyLocationで指定します。 このフィールドでは `HOST.*` マクロおよびユーザーマクロが使用できます。 この設定はCURLOPT_SSLKEYの cURL オプションを設定します。
SSL キーパスワード	SSL 秘密鍵ファイルのパスワード。 このフィールドではユーザーマクロが使用できます。 この設定はCURLOPT_KEYPASSWDの cURL オプションを設定します。

Attention:
[1] Zabbix は PEM 形式のみの証明書および秘密鍵ファイルをサポートしています。証明書および秘密鍵データが PKCS #12 形式ファイル（通常は *.p12 または *.pfx 拡張子）にある場合は、以下のコマンドで PEM ファイルを生成できます:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Note:
Zabbix サーバーは証明書の変更を再起動なしで反映します。

Note:
クライアント証明書と秘密鍵が 1 つのファイルにある場合は、「SSL 証明書ファイル」フィールドにそのファイルを指定し、「SSL キーファイル」フィールドは空にしてください。証明書と鍵は PEM 形式である必要があります。証明書と鍵の結合は簡単です:

```
cat client.crt client.key > client.pem
```

表示　ホストに設定された Web シナリオを表示するには、監視 → ホストに移動し、リストからホストを見つけて最後の列の *Web* ハイパーリンクをクリックします。シナリオ名をクリックすると詳細情報が表示されます。



Web シナリオの概要は、Web 監視ウィジェットによってダッシュボードにも表示できます。

Web シナリオ実行の最近の結果は、監視 → 最新データセクションで確認できます。

拡張監視　受信した HTML ページの内容を記録する必要がある場合があります。これは、Web シナリオのステップが失敗した場合に特に便利です。デバッグレベル 5（トレース）はその目的に使用されます。このレベルは、サーバーおよびプロキシの設定ファイル、またはランタイム制御オプション（-R log_level_increase="http poller,N"、N はプロセス番号）を使用して設定できます。次の例は、デバッグレベル 4 がすでに設定されている場合に拡張監視を開始する方法を示しています。

```
# すべてのhttp pollerのログレベルを上げる:
zabbix_server -R log_level_increase="http poller"

# 2番目のhttp pollerのログレベルを上げる:
zabbix_server -R log_level_increase="http poller,2"
```

拡張 Web 監視が不要な場合は、-R log_level_decrease オプションを使用して停止できます。

1 Web 監視アイテム

概要

Web シナリオが作成されると、モニタリング用にいくつかの新しいアイテムが自動的に追加されます。

すべてのアイテムは、Web シナリオからタグを継承します

シナリオアイテム

シナリオが作成されると、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for scenario <Scenario>	このアイテムはシナリオ全体のダウンロード速度 (1 秒あたりのバイト数)、つまりすべてのステップの平均速度を収集します。 アイテムキー: web.test.in[Scenario,,bps] タイプ: *Numeric(float)*
Failed step of scenario <Scenario>	このアイテムはシナリオで失敗したステップの番号を表示します。すべてのステップが正常に実行されると、0 が返されます。 アイテムキー: web.test.fail[Scenario] タイプ: *Numeric(unsigned)*
Last error message of scenario <Scenario>	このアイテムはシナリオの最後のエラーメッセージテキストを返します。新しい値は、シナリオに失敗したステップがある場合にのみ保存されます。すべての手順に問題がない場合、新しい値は収集されません。 アイテムキー: web.test.error[Scenario] タイプ: *Character*

”Scenario” の代わりに実際のシナリオ名が使用されます。

Note:
シナリオ名にユーザーマクロが含まれている場合、これらのマクロは Web 監視アイテム名で未解決のままになります。

 シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

Note:
Web 監視アイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

これらのアイテムを使用して、トリガーを作成し、通知条件を定義できます。

例 1

「Web シナリオに失敗しました」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.fail[Scenario])<>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 2

トリガー名に問題の説明を含む"Web scenario failed" トリガーを作成するには、次の名前でトリガーを定義します。

```
Web scenario "Scenario" failed: {ITEM.VALUE}
```

さらに次のようにトリガー式を定義します

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 3

「Web アプリケーションが遅い」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.in[Scenario,,bps])<10000
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

シナリオのステップアイテム

ステップが作成されるとすぐに、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for step <Step> of scenario <Scenario>	このアイテムはステップのダウンロード速度（バイト/秒）を取得します。 Item key: web.test.in[Scenario,Step,bps] Type: *Numeric(float)*
Response time for step <Step> of scenario <Scenario>	このアイテムはステップの応答時間を秒単位で取得します。リクエストの開始からすべての情報が転送されるまでがカウントされます。 Item key: web.test.time[Scenario,Step,resp] Type:*Numeric(float)*
Response code for step <Step> of scenario <Scenario>	このアイテムはステップの応答コードを取得します。 Item key:web.test.rspcode[Scenario,Step]Type: *Numeric(unsigned)*

"Scenario" と"Step" の代わりに、実際のシナリオとステップ名がそれぞれ使用されます。

Note:
Web モニタリングアイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

Note:
シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

これらのアイテムを使用してトリガーを作成し、通知条件を定義できます。たとえば「ZabbixGUI ログインが遅すぎる」トリガーを作成するには次のようにトリガー式を定義します。

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 実際のシナリオ

概要

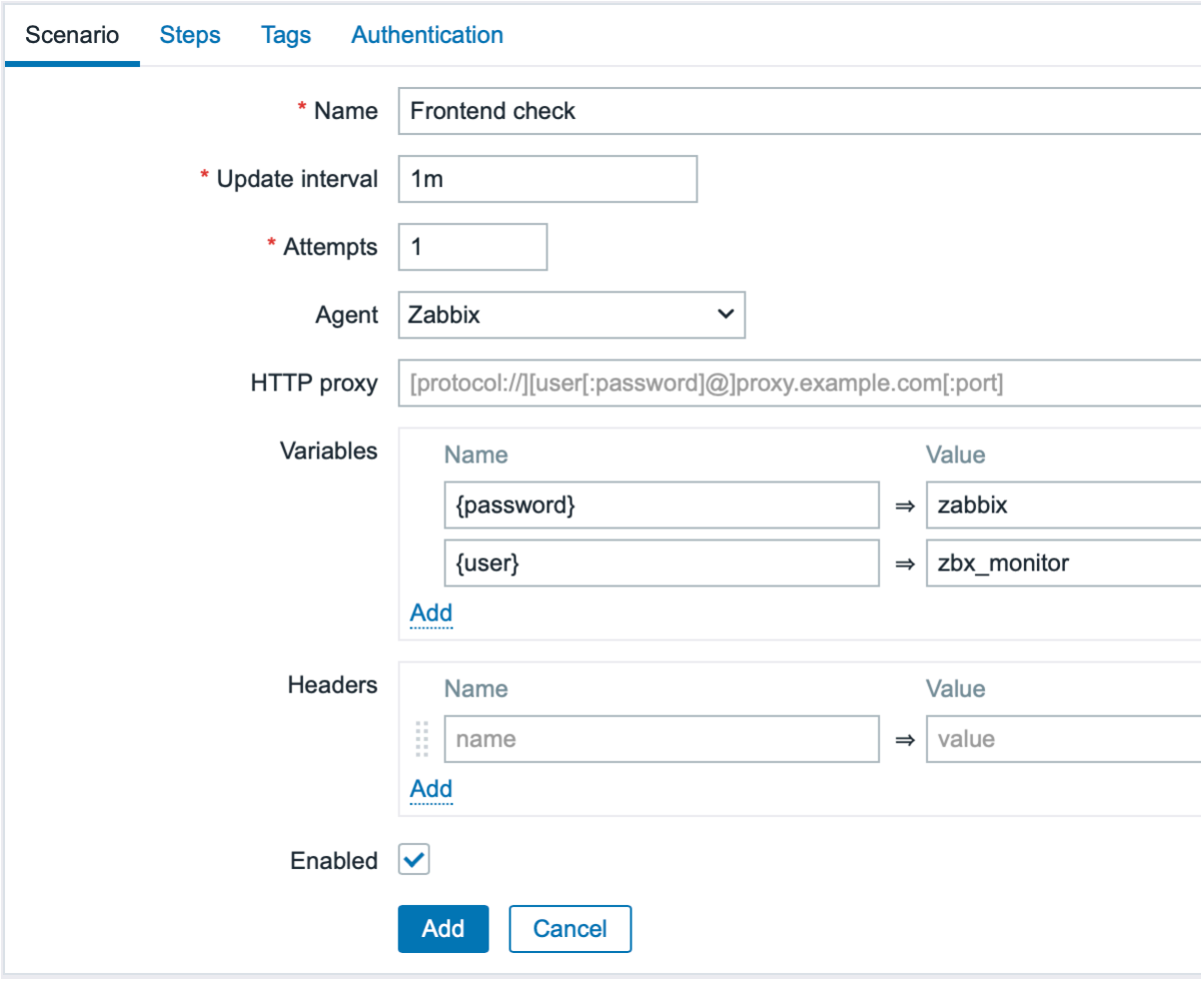
このセクションでは、Web 監視をどのように使用できるかについて、実際のステップバイステップの例を示します。

Zabbix の Web 監視を使用して、Zabbix フロントエンドを監視します。目的は、Zabbix フロントエンドが利用可能であるか、正しいコンテンツを提供しているか、どれだけ迅速に動作するかを判断することです。これを達成するために、最初のページの可用性の確認、ユーザー名とパスワードでのログイン、ログイン成功の検証、ログアウト、ログアウトの確認など、いくつかのステップが必要です。

シナリオ

新しい Web シナリオの追加

データ収集 → ホストに移動し、ホストを選択してそのホストの行の *Web* をクリックします。
次に *Web* シナリオの作成をクリックします。



新しいシナリオフォームで、次のフィールドに入力します:

- 名前 - フロントエンドチェック
- 更新間隔 - 1m
- 試行回数 - 1
- エージェント - Zabbix

変数セクションで、*{password}* と *{user}* の 2 つの変数を追加します。
既存の Zabbix ユーザーの認証情報を値として入力します。

安全のため、監視目的で使用するために最小限の権限を持つ別のユーザーを作成することをお勧めします。

必要に応じて、タグタブに切り替えて Web シナリオタグを追加します。

Note:
完全に設定されると、この Web シナリオは自動的に Zabbix トラッパーアイテムをホストに追加します。
Web シナリオタグを使用して、関連するアイテムやトリガーをすばやく特定したり、収集したデータを検索したりできます。
たとえば、このチュートリアルに適したタグは `component: web-scenario` や `target: frontend` です。

Web シナリオのステップの設定

ステップタブに切り替えて、シナリオのステップを定義します。追加ボタンをクリックして、個々のステップを追加します。

共通フィールド

以下に説明する各ステップについて、ステップ固有のフィールドに加えて、以下のフィールドも入力してください。

- **URL** - Zabbix フロントエンドの URL
- **Timeout** - 15 秒
- **Required status codes** - 200

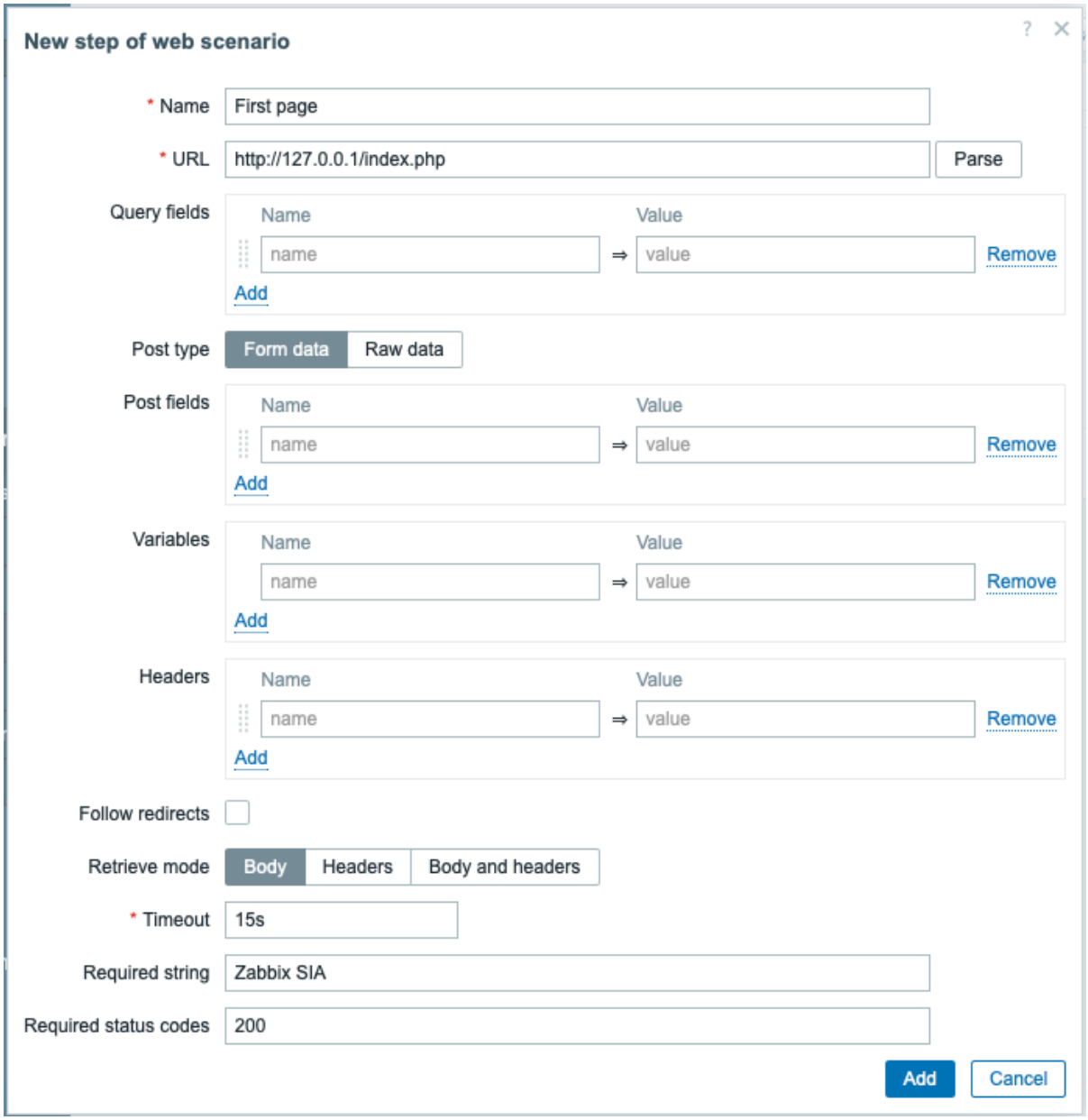
Web シナリオステップ 1

最初のページが正しく応答し、HTTP レスポンスコード 200 を返し、”Zabbix SIA” というテキストが含まれていることを確認します。

- 名前フィールドに *First page* と入力します。
- 必須文字列フィールドに *Zabbix SIA* と入力します。

- 共通フィールドを入力します。

ステップの設定が完了したら、追加ボタンを押します。



Web シナリオステップ 2

シナリオレベルで定義されたマクロ（変数）*{user}* と *{password}* を使用して Zabbix フロントエンドにログインします。

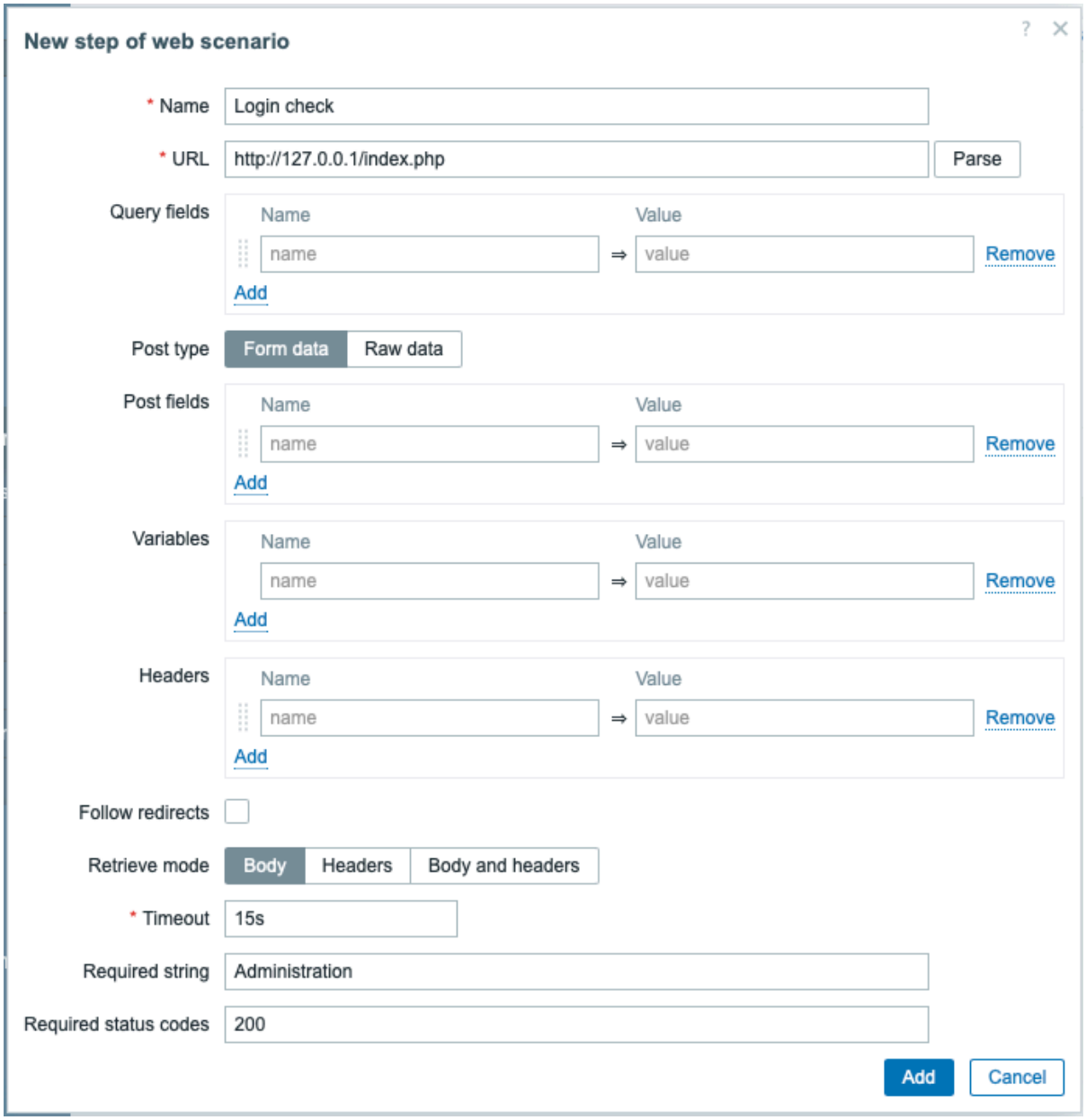
- 名前フィールドに *Login* と入力します。
- **Post** フィールドセクションで、3 つの Post フィールドを追加します:
  - *name* に値 *{user}*
  - *password* に値 *{password}*
  - *enter* に値 *Sign in*
- 変数セクションで、新しい変数 *{csrf_token}* を値 *regex:([0-9a-z]{64})* で追加します。この変数は割り当てられた CSRF トークンの値をキャッチし、ステップ 4で再利用します。
- 共通フィールドを入力します。

Attention:
Zabbix フロントエンドはログイン時に JavaScript リダイレクトを使用するため、最初にログインを行い、ログイン後の機能は以降のステップでのみ確認できることに注意してください。また、ログインステップでは **index.php** ファイルへのフル URL を使用する必要があります。

Web シナリオステップ 3

ログイン後、ログイン時にのみ表示される文字列（例：管理）をチェックして成功を確認します。

- 名前フィールドにログインチェックと入力します。
- 必須文字列フィールドに管理と入力します。
- 共通フィールドを入力します。

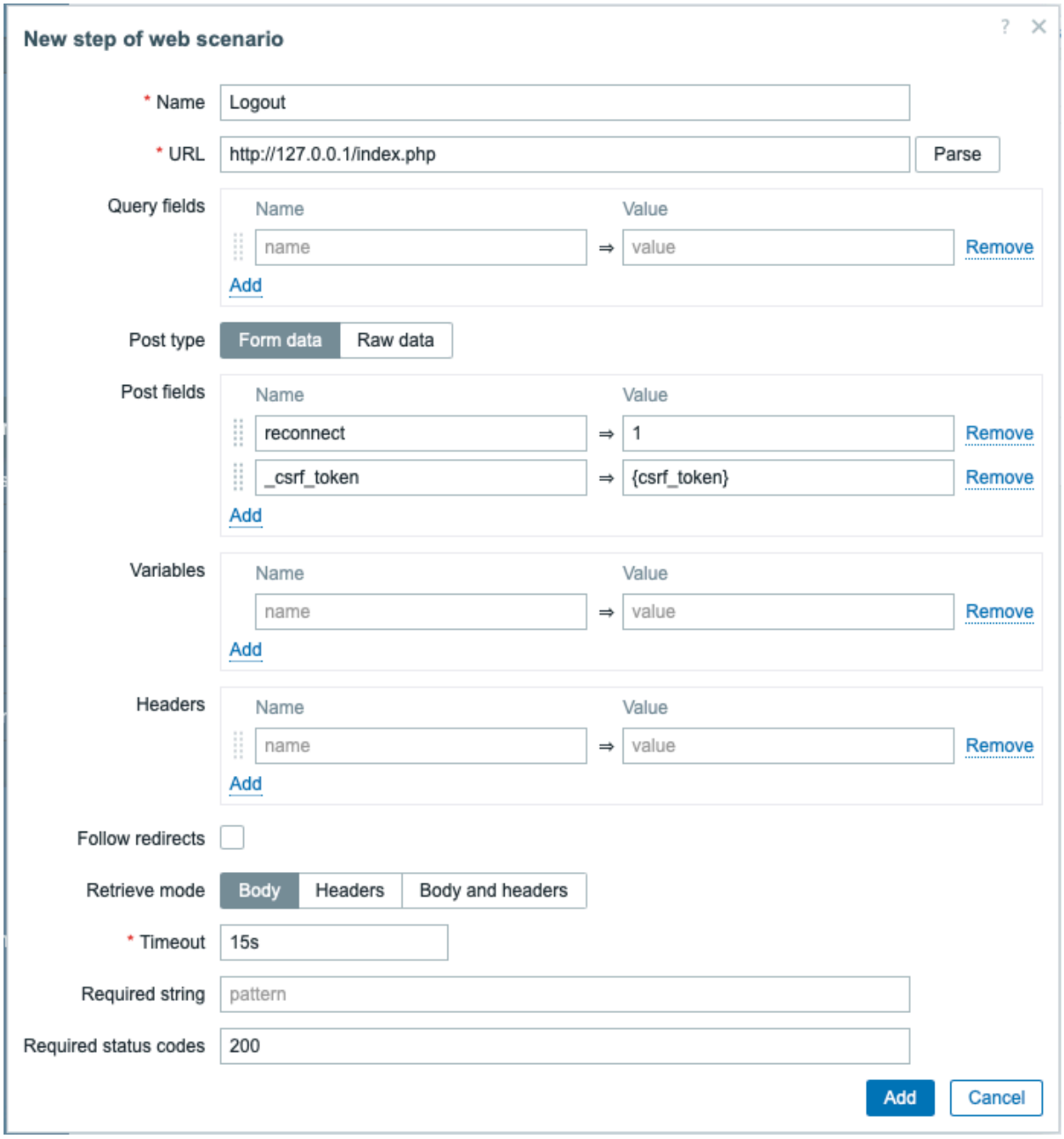


Web シナリオステップ 4

フロントエンドのアクセシビリティとログインが確認できたら、ログアウトステップを追加します。そうしないと、Zabbix データベースに多くのオープンセッションレコードが蓄積されてしまいます。

- 名前フィールドに *Logout* と入力します。
- **Post** フィールドセクションで、2 つの post フィールドを追加します:
  - *reconnect* に値 *1* を設定
  - *_csrf_token* に値 *{csrf_token}* を設定
- 共通フィールドを入力します。

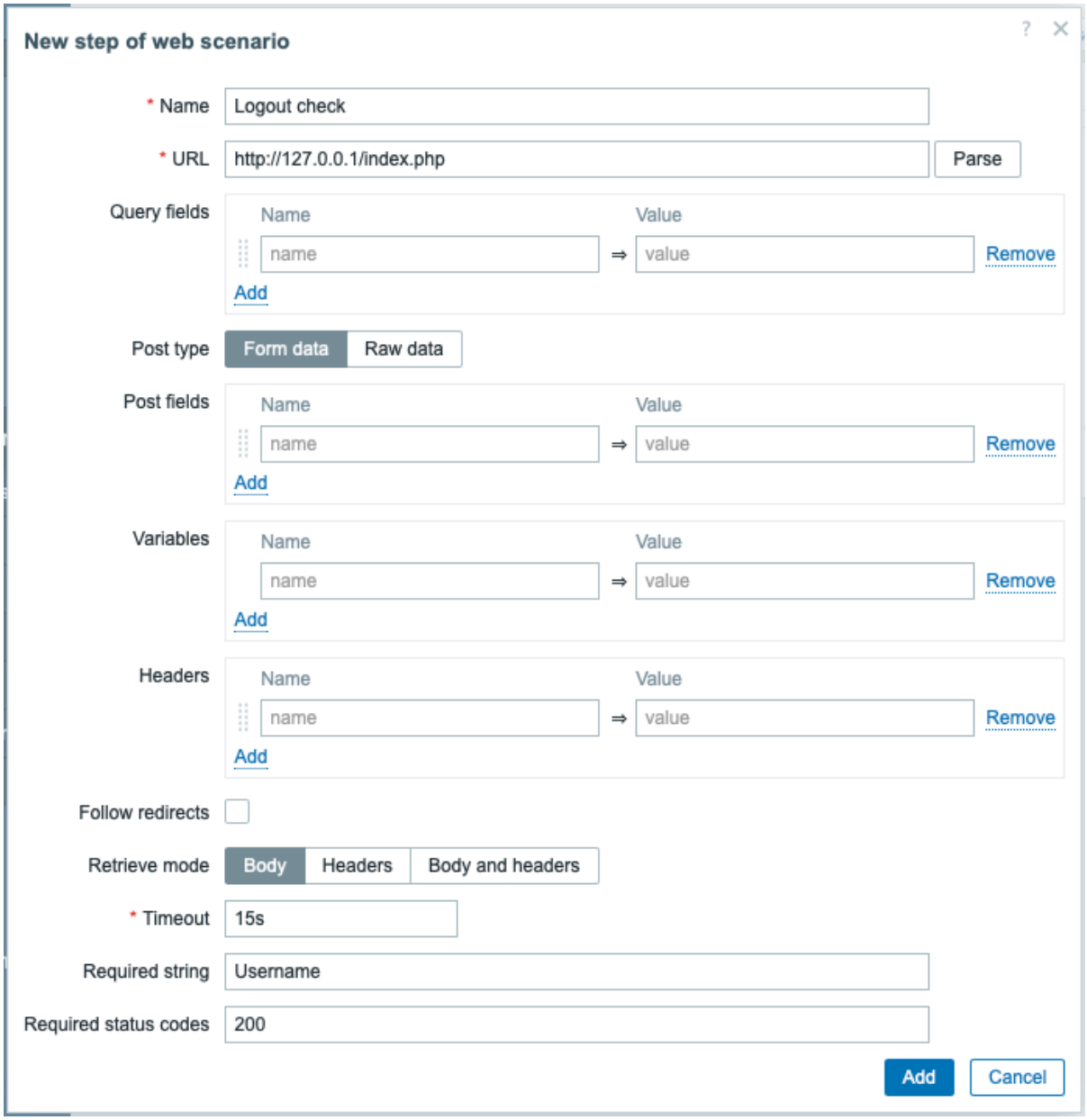
このステップでは、ステップ 2で取得した変数 {csrf_token} を使用します。



Web シナリオステップ 5

ログアウトを確認するために、**Username** 文字列をチェックします。

- 名前フィールドに *Logout check* を入力します。
- 必須文字列フィールドに *Username* を入力します。
- 共通フィールドを入力します。



ステップの完全な設定

Web シナリオステップの完全な設定は次のようになります。

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati...	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

結果を確認する

作成した Web 監視シナリオを保存します。

シナリオはホストに追加されます。Web シナリオ情報を表示するには、監視 → ホストに移動し、リストからホストを見つけて、最後の列の Web ハイパーリンクをクリックします。



シナリオ名をクリックすると、より詳細な統計情報が表示されます。



8 仮想マシンの監視

概要　Zabbix は、ローレベルディスカバリルールを使用して、VMware ハイパーバイザーおよび仮想マシンを自動的に検出し、事前定義されたホストプロトタイプに基づいて監視するホストを作成できます。

Zabbix には、VMware vCenter または ESXi ハイパーバイザーの監視用にすぐに使えるテンプレートも含まれています。

必要な VMware vCenter または vSphere の最小バージョンは 5.1 です。

データ収集　仮想マシンの監視は 2 つのステップで構成されます。

1. Zabbix の *vmware collector* プロセスが仮想マシンのデータを収集します。これらのプロセスは SOAP プロトコルを介して VMware の Web サービスから必要な情報を取得し、前処理を行い、Zabbix サーバーの共有メモリに保存します。
2. Zabbix の *poller* プロセスが、Zabbix のシンプルチェックであるVMware 監視アイテムキーを使用してデータを取得します。

Zabbix は収集したデータを VMware の構成データと VMware のパフォーマンスカウンターデータに分けます。両方のタイプのデータは *vmware collector* プロセスによって独立して収集されます。

VMware のパフォーマンスカウンター情報に基づいて、以下の統計情報が利用可能です。

- データストア
- ディスクデバイス
- CPU
- 電源
- ネットワークインターフェース
- カスタムパフォーマンスカウンターアイテム

VMware のパフォーマンスカウンターからデータを取得するアイテムの完全なリストについては、*VMware* 監視アイテムキーを参照してください。

VMware イベントの取得頻度はvmware.eventlogのポーリング間隔に依存しますが、5 秒未満にはできませんのでご注意ください。

設定 Zabbix サーバーをソースからコンパイルする場合、仮想マシンの監視を有効にするには、`--with-libcurl --with-libxml2` の設定オプションでコンパイルする必要があります。Zabbix パッケージは、これらのオプションがすでに有効になった状態でコンパイルされています。

仮想マシンの監視のために変更できる Zabbix サーバーの設定ファイルパラメーターは以下の通りです。

- StartVMwareCollectors

Note:
監視対象の VMware サービス数よりも多くのコレクターを有効にすることを推奨します。そうしないと、VMware のパフォーマンスカウンタ統計の取得が、VMware 構成データの取得によって遅延する可能性があります（大規模なインストールでは構成データの取得に時間がかかります）。

 一般的に、StartVMwareCollectors の値は 2 未満にせず、監視対象の VMware サービス数の 2 倍を超えないようにしてください。サービス数 < StartVMwareCollectors < (サービス数 * 2) となるようにします。例えば、1 つの VMware サービスを監視する場合は StartVMwareCollectors を 2 に、3 つのサービスを監視する場合は 5 に設定します。

 必要なコレクター数は、VMware 環境の規模や、VMwareFrequency および VMwarePerfFrequency の設定パラメータにも依存することに注意してください。

- VMwareCacheSize
- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
データストア容量メトリクスをサポートするには、VMware の vpxd.stats.maxQueryMetrics キーの値が少なくとも 64 に設定されていることを確認してください。詳細は*VMware Knowledge Base* の記事を参照してください。

ディスカバリ

Zabbix は、低レベルディスカバリルール（例：*vmware.hv.discovery[{$VMWARE.URL}]*）を使用して、VMware ハイパーバイザーや仮想マシンを自動的に検出できます。さらに、Zabbix はホストプロトタイプを使用して、検出されたエンティティの実際のホストを自動的に生成できます。詳細については、ホストプロトタイプを参照してください。

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

拡張ロギング *vmware collector* プロセスによって収集されたデータは、デバッグレベル 5 を使用して詳細なデバッグのためにログに記録できます。デバッグレベルは、server およびproxy の設定ファイルで設定するか、ランタイム制御オプション `-R log_level_increase="vmware collector,N"` を使用して設定できます。ここで”N” はプロセス番号です。

たとえば、すべての *vmware collector* プロセスのデバッグレベルを 4 から 5 に上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector"
```

2 番目の *vmware collector* プロセスのデバッグレベルを 4 から 5 に上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector,2"
```

VMware collector データの拡張ロギングが不要になった場合は、`-R log_level_decrease` コマンドを実行してデバッグレベルをデフォルト値（3）に下げることを推奨します。

トラブルシューティング

- 利用できないメトリックがある場合は、それらが最近の VMware vSphere バージョンで利用不可またはデフォルトでオフになっていないか、またはパフォーマンスメトリックのデータベースクエリに何らかの制限が設けられていないかを確認してください。詳細については、ZBX-12094 を参照してください。
- `config.vpxd.stats.maxQueryMetrics` が無効であるか、許可されている最大文字数を超えている場合は、vCenter Server の設定に `config.vpxd.stats.maxQueryMetrics` パラメータを追加してください。このパラメータの値は、VMware の *web.xml* ファイルの `maxQuerysize` の値と同じにする必要があります。詳細については、*VMware Knowledge Base の記事* を参照してください。
- Zabbix インストールが過剰なメモリを使用していると思われる場合は、tcmalloc による過剰なメモリ使用量のプロファイリング を参照してください。

1 VMware 監視アイテムキー

概要　このページでは、VMware 環境の監視に使用できるシンプルチェックの詳細を説明します。メトリクスは監視対象ごとにグループ化されています。

サポートされているアイテムキー　アイテムキーはパラメータや追加情報なしで一覧表示されています。アイテムキーをクリックすると、詳細が表示されます。

アイテムキー	説明	アイテムグループ
vmware.eventlog	VMware イベントログ。	一般サービス
vmware.fullname	VMware サービスのフルネーム。	
vmware.version	VMware サービスのバージョン。	
vmware.cl.perfcounter	VMware クラスタのパフォーマンスカウンタメトリクス。	クラスタ
vmware.cluster.alarms.get	VMware クラスタのアラームデータ。	
vmware.cluster.discovery	VMware クラスタのディスカバリ。	
vmware.cluster.property	VMware クラスタのプロパティ。	
vmware.cluster.status	VMware クラスタのステータス。	
vmware.cluster.tags.get	VMware クラスタのタグ配列。	
vmware.datastore.alarms.get	VMware データストアのアラームデータ。	データストア
vmware.datastore.discovery	VMware データストアのディスカバリ。	
vmware.datastore.hv.list	データストアのハイパーバイザーリスト。	
vmware.datastore.perfcounter	VMware データストアのパフォーマンスカウンタ値。	
vmware.datastore.property	VMware データストアのプロパティ。	
vmware.datastore.read	データストアからの読み取り操作にかかる時間。	
vmware.datastore.size	VMware データストアの容量（バイトまたは全体のパーセンテージ）。	
vmware.datastore.tags.get	VMware データストアのタグ配列。	
vmware.datastore.write	データストアへの書き込み操作にかかる時間。	
vmware.dc.alarms.get	VMware データセンターのアラームデータ。	データセンター
vmware.dc.discovery	VMware データセンターのディスカバリ。	
vmware.dc.tags.get	VMware データセンターのタグ配列。	
vmware.dvswitch.discovery	VMware vSphere Distributed Switch のディスカバリ。	vSphere Distributed Switch
vmware.dvswitch.fetchports	VMware vSphere Distributed Switch のポートデータ。	
vmware.hv.alarms.get	VMware ハイパーバイザーのアラームデータ。	ハイパーバイザー
vmware.hv.cluster.name	VMware ハイパーバイザーのクラスタ名。	
vmware.hv.connectionstate	VMware ハイパーバイザーの接続状態。	

アイテムキー	説明	アイテムグループ
vmware.hv.cpu.usage	VMware ハイパーバイザーのプロセッサ使用率（Hz）。	
vmware.hv.cpu.usage.perf	VMware ハイパーバイザーのプロセッサ使用率（インターバル中のパーセンテージ）。	
vmware.hv.cpu.utilization	VMware ハイパーバイザーのプロセッサ使用率（インターバル中のパーセンテージ、電源管理または HT に依存）。	
vmware.hv.datacenter.name	VMware ハイパーバイザーのデータセンター名。	
vmware.hv.datastore.discovery	VMware ハイパーバイザーのデータストアのディスカバリ。	
vmware.hv.datastore.list	VMware ハイパーバイザーのデータストアリスト。	
vmware.hv.datastore.multipath	利用可能なデータストアパスの数。	
vmware.hv.datastore.read	データストアからの読み取り操作の平均時間。	
vmware.hv.datastore.size	VMware データストアの容量（バイトまたは全体のパーセンテージ）。	
vmware.hv.datastore.write	データストアへの書き込み操作の平均時間。	
vmware.hv.discovery	VMware ハイパーバイザーのディスカバリ。	
vmware.hv.diskinfo.get	VMware ハイパーバイザーのディスクデータ。	
vmware.hv.fullname	VMware ハイパーバイザーの名前。	
vmware.hv.hw.cpu.freq	VMware ハイパーバイザーのプロセッサ周波数。	
vmware.hv.hw.cpu.model	VMware ハイパーバイザーのプロセッサモデル。	
vmware.hv.hw.cpu.num	VMware ハイパーバイザーのプロセッサコア数。	
vmware.hv.hw.cpu.threads	VMware ハイパーバイザーのプロセッサスレッド数。	
vmware.hv.hw.memory	VMware ハイパーバイザーのメモリ総容量。	
vmware.hv.hw.model	VMware ハイパーバイザーのモデル。	
vmware.hv.hw.sensors.get	VMware ハイパーバイザーのハードウェアセンサー値。	
vmware.hv.hw.serialnumber	VMware ハイパーバイザーのシリアル番号。	
vmware.hv.hw.uuid	VMware ハイパーバイザーの BIOS UUID。	
vmware.hv.hw.vendor	VMware ハイパーバイザーのベンダー名。	
vmware.hv.maintenance	VMware ハイパーバイザーのメンテナンス状態。	
vmware.hv.memory.size.ballooned	VMware ハイパーバイザーのバルーンメモリサイズ。	
vmware.hv.memory.used	VMware ハイパーバイザーの使用済みメモリサイズ。	
vmware.hv.net.if.discovery	VMware ハイパーバイザーのネットワークインターフェースのディスカバリ。	
vmware.hv.network.in	VMware ハイパーバイザーのネットワーク入力統計。	
vmware.hv.network.linkspeed	VMware ハイパーバイザーのネットワークインターフェース速度。	
vmware.hv.network.out	VMware ハイパーバイザーのネットワーク出力統計。	
vmware.hv.perfcounter	VMware ハイパーバイザーのパフォーマンスカウンタ値。	
vmware.hv.property	VMware ハイパーバイザーのプロパティ。	
vmware.hv.power	VMware ハイパーバイザーの電力使用量。	
vmware.hv.sensor.health.state	VMware ハイパーバイザーのヘルス状態ロールアップセンサー。	
vmware.hv.sensors.get	VMware ハイパーバイザーの HW ベンダーステートセンサー。	
vmware.hv.status	VMware ハイパーバイザーのステータス。	
vmware.hv.tags.get	VMware ハイパーバイザーのタグ配列。	
vmware.hv.uptime	VMware ハイパーバイザーの稼働時間。	
vmware.hv.version	VMware ハイパーバイザーのバージョン。	
vmware.hv.vm.num	VMware ハイパーバイザー上の仮想マシン数。	
vmware.rp.cpu.usage	VMware リソースプールのインターバル中の CPU 使用率（Hz）。	リソースプール
vmware.rp.memory	VMware リソースプールのメモリメトリクス。	
vmware.alarms.get	VMware バーチャルセンターのアラームデータ。	バーチャルセンター
vmware.vm.alarms.get	VMware 仮想マシンのアラームデータ。	仮想マシン
vmware.vm.attribute	VMware 仮想マシンのカスタム属性値。	
vmware.vm.cluster.name	VMware 仮想マシンの名前。	
vmware.vm.consolidationneeded	VMware 仮想マシンディスクの統合が必要。	
vmware.vm.cpu.latency	仮想マシンが物理 CPU へのアクセスを競合して実行できなかった時間の割合。	
vmware.vm.cpu.num	VMware 仮想マシンのプロセッサ数。	
vmware.vm.cpu.readiness	仮想マシンが実行準備ができていたが、物理 CPU でスケジュールできなかった時間の割合。	
vmware.vm.cpu.ready	仮想マシンが実行準備ができていたが、物理 CPU でスケジュールできなかった時間。	
vmware.vm.cpu.swapwait	スワップインを待っていた CPU 時間の割合。	
vmware.vm.cpu.usage	VMware 仮想マシンのプロセッサ使用率（Hz）。	
vmware.vm.cpu.usage.perf	VMware 仮想マシンのプロセッサ使用率（インターバル中のパーセンテージ）。	
vmware.vm.datacenter.name	VMware 仮想マシンのデータセンター名。	
vmware.vm.discovery	VMware 仮想マシンのディスカバリ。	
vmware.vm.guest.memory.size.swapped	スワップ領域にスワップアウトされたゲスト物理メモリの量。	

アイテムキー	説明	アイテムグループ
vmware.vm.guest.osuptime	最後の OS 起動からの経過時間。	
vmware.vm.hv.maintenance	VMware 仮想マシンのハイパーバイザーメンテナンス状態。	
vmware.vm.hv.name	VMware 仮想マシンのハイパーバイザー名。	
vmware.vm.memory.size	VMware 仮想マシンのメモリ総容量。	
vmware.vm.memory.size.ballooned	VMware 仮想マシンのバルーンメモリサイズ。	
vmware.vm.memory.size.compressed	VMware 仮想マシンの圧縮メモリサイズ。	
vmware.vm.memory.size.consumed	ゲスト物理メモリページのバックアップに消費されたホスト物理メモリの量。	
vmware.vm.memory.size.private	VMware 仮想マシンのプライベートメモリサイズ。	
vmware.vm.memory.size.shared	VMware 仮想マシンの共有メモリサイズ。	
vmware.vm.memory.size.swapped	VMware 仮想マシンのスワップメモリサイズ。	
vmware.vm.memory.size.usage.guest	VMware 仮想マシンのゲストメモリ使用量。	
vmware.vm.memory.size.usage.host	VMware 仮想マシンのホストメモリ使用量。	
vmware.vm.memory.usage	消費されたホスト物理メモリの割合。	
vmware.vm.net.if.discovery	VMware 仮想マシンのネットワークインターフェースのディスカバリ。	
vmware.vm.net.if.in	VMware 仮想マシンのネットワークインターフェース入力統計。	
vmware.vm.net.if.out	VMware 仮想マシンのネットワークインターフェース出力統計。	
vmware.vm.net.if.usage	インターバル中の VMware 仮想マシンのネットワーク使用率。	
vmware.vm.perfcounter	VMware 仮想マシンのパフォーマンスカウンタ値。	
vmware.vm.powerstate	VMware 仮想マシンの電源状態。	
vmware.vm.property	VMware 仮想マシンのプロパティ。	
vmware.vm.snapshot.get	VMware 仮想マシンのスナップショット状態。	
vmware.vm.state	VMware 仮想マシンの状態。	
vmware.vm.storage.committed	VMware 仮想マシンのコミット済みストレージ容量。	
vmware.vm.storage.readoio	収集インターバル中の仮想ディスクへの未処理の読み取り要求の平均数。	
vmware.vm.storage.totalreadlatency	仮想ディスクからの読み取りにかかる平均時間。	
vmware.vm.storage.totalwritelatency	仮想ディスクへの書き込みにかかる平均時間。	
vmware.vm.storage.uncommitted	VMware 仮想マシンの未コミットストレージ容量。	
vmware.vm.storage.unshared	VMware 仮想マシンの未共有ストレージ容量。	
vmware.vm.storage.writeoio	収集インターバル中の仮想ディスクへの未処理の書き込み要求の平均数。	
vmware.vm.tags.get	VMware 仮想マシンのタグ配列。	
vmware.vm.tools	VMware 仮想マシンのゲストツールの状態またはバージョン。	
vmware.vm.uptime	VMware 仮想マシンの稼働時間。	
vmware.vm.vfs.dev.discovery	VMware 仮想マシンのディスクデバイスのディスカバリ。	
vmware.vm.vfs.dev.read	VMware 仮想マシンのディスクデバイスの読み取り統計。	
vmware.vm.vfs.dev.write	VMware 仮想マシンのディスクデバイスの書き込み統計。	
vmware.vm.vfs.fs.discovery	VMware 仮想マシンのファイルシステムのディスカバリ。	
vmware.vm.vfs.fs.size	VMware 仮想マシンのファイルシステム統計。	

アイテムキーの詳細　山括弧のないパラメータは必須です。山括弧 **<** **>** で囲まれたパラメータはオプションです。

vmware.eventlog[url,<mode>,<severity>]

 VMware イベントログ。
 戻り値: *Log*。

パラメータ:

- **url** - VMware サービスの URL
- **mode** - *all* (デフォルト) または *skip* - 古いデータの処理をスキップ
- **severity** - 深刻度でフィルタリング: *error*、*warning*、*info* または *user*。複数の深刻度をカンマ区切りリストで指定する場合は、このパラメータを引用符で囲む必要があります (例: `"error,warning,info,user"`)。デフォルトでは無効。

コメント:

- URL ごとに `vmware.eventlog` アイテムキーは 1 つだけでなければなりません。
- VMware イベントログレコードのフィルタリング例も参照してください。

vmware.fullnameurl

 VMware サービスのフルネーム。
 戻り値：文字列。

パラメータ:

- **url** - VMware サービスの URL。

vmware.versionurl

 VMware サービスのバージョン。
 戻り値：*String*。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cl.perfcounter[url,id,path,<instance>]

 VMware クラスタのパフォーマンスカウンタのメトリクス。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID。id は `vmware.cluster.discovery[]` から {#CLUSTER.ID} として取得できます。
- **path** - パフォーマンスカウンタのパス[1]
- **instance** - パフォーマンスカウンタのインスタンス

vmware.cluster.alarms.get[url,id]

 VMware クラスタのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID

vmware.cluster.discoveryurl

 VMware クラスタのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cluster.property[url,id,prop]

 VMware クラスタのプロパティ。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID
- **prop** - VMware SDKで定義されている VM オブジェクトのプロパティ名であるプロパティパス

例:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 VMware クラスタのステータス。
 戻り値: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **name** - VMware クラスタ名

vmware.cluster.tags.get[url,id]

 VMware クラスタタグの配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID

このアイテムは vSphere 6.5 以降で動作します。

vmware.datastore.alarms.get[url,uuid]

 VMware データストアのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子

vmware.datastore.discovery[url,<filter_uuid>]

 VMware データストアのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **filter_uuid** - 単一の HV または VM の一意の識別子 (デフォルト: 空)。指定した場合、リンクされたデータストアのみがディスカバリされます。

vmware.datastore.hv.list[url,datastore]

 データストアのハイパーバイザーのリスト。
 戻り値: *String*。

パラメーター:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前

出力例:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 VMware データストアのパフォーマンスカウンタ値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子
- **path** - パフォーマンスカウンタのパス[1]
- **instance** - パフォーマンスカウンタのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。`instance` は `vmware.datastore.discovery[]` から {#DATASTORE.EXTENT} 配列の一部として取得できます。

vmware.datastore.property[url,uuid,prop]

 VMware データストアのプロパティ。
 戻り値：文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子
- **prop** - プロパティパス

vmware.datastore.read[url,datastore,<mode>]

 データストアからの読み取り操作にかかった時間（ミリ秒）。
 戻り値：整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - *latency*（平均値、デフォルト）または *maxlatency*（最大値）

vmware.datastore.size[url,datastore,<mode>]

 VMware データストアの容量（バイト単位または全体に対するパーセンテージ）。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - 指定可能な値: *total*（デフォルト）、*free*、*pfree*（空き容量のパーセンテージ）、*uncommitted*

vmware.datastore.tags.get[url,uuid]

 VMware データストアタグの配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子

このアイテムは vSphere 6.5 以降で動作します。

vmware.datastore.write[url,datastore,<mode>]

 データストアへの書き込み操作にかかる時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (平均値、デフォルト) または *maxlatency* (最大値)

vmware.dc.alarms.get[url,id]

 VMware データセンターのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware データセンターの ID

vmware.dc.discoveryurl

 VMware データセンターのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dc.tags.get[url,id]

 VMware データセンタータグの配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware データセンターの ID

このアイテムは vSphere 6.5 以降で動作します。

vmware.dvswitch.discoveryurl

 VMware vSphere Distributed Switch のディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 VMware vSphere Distributed Switch ポートデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware DVSwitch のグローバル一意識別子
- **filter** - ポートを選択するためのカンマ区切りの条件を含む単一の文字列
- **mode** - *state* (すべての XML から”config” XML ノードを除外、デフォルト) または *full*

filter パラメータは、VMware データオブジェクト DistributedVirtualSwitchPortCriteria で利用可能な criteria をサポートします。

例:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 VMware ハイパーバイザーのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cluster.name[url,uuid]

 VMware ハイパーバイザークラスタ名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL

- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.connectionstate[url,uuid]

 VMware ハイパーバイザーの接続状態。
 戻り値: *String*: *connected*、*disconnected*、または *notResponding*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.usage[url,uuid]

 VMware ハイパーバイザーのプロセッサ使用率 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.usage.perf[url,uuid]

 インターバル中の VMware ハイパーバイザーのプロセッサ使用率 (%)。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.utilization[url,uuid]

 インターバル中の VMware ハイパーバイザーのプロセッサ使用率 (%)。電源管理または HT に依存します。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.datacenter.name[url,uuid]

 VMware ハイパーバイザのデータセンター名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザのグローバル一意識別子

vmware.hv.datastore.discovery[url,uuid]

 VMware ハイパーバイザーデータストアのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.datastore.list[url,uuid]

 VMware ハイパーバイザーのデータストアのリスト。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

出力例:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 利用可能なデータストアパスの数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前

- **partitionid** - `vmware.hv.datastore.discovery` からの物理デバイスの内部 ID

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 データストアからの読み取り操作の平均時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (デフォルト)

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 VMware データストアの容量 (バイト単位または全体に対するパーセンテージ)。
 返り値: *Integer* - バイト単位; *Float* - パーセンテージ。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - 指定可能な値: *total* (デフォルト), *free*, *pfree* (空き容量のパーセンテージ), *uncommitted*

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 データストアへの書き込み操作にかかる平均時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (デフォルト)

vmware.hv.discoveryurl

 VMware ハイパーバイザーのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.hv.diskinfo.get[url,uuid]

 VMware ハイパーバイザーディスクデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.fullname[url,uuid]

 VMware ハイパーバイザー名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.freq[url,uuid]

 VMware ハイパーバイザーのプロセッサ周波数 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.model[url,uuid]

 VMware ハイパーバイザーのプロセッサモデル。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.num[url,uuid]

 VMware ハイパーバイザーのプロセッサコア数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.threads[url,uuid]

 VMware ハイパーバイザー上のプロセッサスレッド数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.memory[url,uuid]

 VMware ハイパーバイザーのメモリ総容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.model[url,uuid]

 VMware ハイパーバイザーモデル。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.sensors.get[url,uuid]

 VMware ハイパーバイザーのハードウェアセンサーの値。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.serialnumber[url,uuid]

 VMware ハイパーバイザーのシリアル番号。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.hw.uuid[url,uuid]

 VMware ハイパーバイザーの BIOS UUID。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.hw.vendor[url,uuid]

 VMware ハイパーバイザーのベンダー名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.maintenance[url,uuid]

 VMware ハイパーバイザーのメンテナンスステータスです。
 戻り値: *0* - メンテナンス中ではない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.memory.size.ballooned[url,uuid]

 VMware ハイパーバイザーのバルーンメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.memory.used[url,uuid]

 VMware ハイパーバイザーの使用メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.net.if.discovery[url,uuid]

 VMware ハイパーバイザーのネットワークインターフェースのディスカバリ。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.network.in[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク入力統計情報 (1 秒あたりのバイト数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **mode** - *bps* (デフォルト), *packets*, *dropped*, *errors*, *broadcast*

vmware.hv.network.linkspeed[url,uuid,ifname]

 VMware ハイパーバイザーのネットワークインターフェースの速度。
 戻り値：整数。ネットワークインターフェースがダウンしている場合は *0*、それ以外の場合はインターフェースの速度値を返します。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **ifname** - インターフェース名

vmware.hv.network.out[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク出力統計情報 (1 秒あたりのバイト数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **mode** - *bps* (デフォルト), *packets*, *dropped*, *errors*, *broadcast*

vmware.hv.perfcounter[url,uuid,path,<instance>]

 VMware ハイパーバイザのパフォーマンスカウンタ値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザのグローバル一意識別子
- **path** - パフォーマンスカウンタのパス [1]
- **instance** - パフォーマンスカウンタのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。

vmware.hv.property[url,uuid,prop]

 VMware ハイパーバイザーのプロパティ。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **prop** - プロパティパス

vmware.hv.power[url,uuid,<max>]

 VMware ハイパーバイザーの消費電力 (W)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **max** - 許容される最大消費電力

vmware.hv.sensor.health.state[url,uuid]

 VMware ハイパーバイザーのヘルス状態ロールアップセンサー。
 戻り値: 整数: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

VMware は *VMware Rollup Health State* センサーを非推奨にしたため、このアイテムは VMware vSphere 6.5 以降では動作しない場合があります。

vmware.hv.sensors.get[url,uuid]

 VMware ハイパーバイザーの HW ベンダーステートセンサー。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバルユニーク識別子

vmware.hv.status[url,uuid]

 VMware ハイパーバイザーのステータス。
 戻り値: 整数: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムはホストシステムの全体的なステータスプロパティを使用します。

vmware.hv.tags.get[url,uuid]

 VMware ハイパーバイザーのタグ配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere 6.5 以降で動作します。

vmware.hv.uptime[url,uuid]

 VMware ハイパーバイザーの稼働時間 (秒)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムはホストシステムの全体的なステータスプロパティを使用します。

vmware.hv.version[url,uuid]

 VMware ハイパーバイザーのバージョン。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.vm.num[url,uuid]

 VMware ハイパーバイザー上の仮想マシンの数。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.rp.cpu.usage[url,rpid]

 VMware リソースプールの間隔中の CPU 使用率 (ヘルツ単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **rpid** - VMware リソースプール ID

vmware.rp.memory[url,rpid,<mode>]

 VMware リソースプールのメモリメトリック。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **rpid** - VMware リソースプール ID
- **mode** - 可能な値:
consumed (デフォルト) - ゲスト物理メモリページのバックアップに消費されたホスト物理メモリの量
 ballooned - ゲスト内のバルーンドライバによって仮想マシンから回収されたゲスト物理メモリの量
overhead - 仮想マシンを実行するために ESXi データ構造によって消費されたホスト物理メモリの量

vmware.alarms.geturl

 VMware 仮想センターのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.vm.alarms.get[url,uuid]

 VMware 仮想マシンのアラームデータ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.attribute[url,uuid,name]

 VMware 仮想マシンのカスタム属性値。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **name** - カスタム属性名

vmware.vm.cluster.name[url,uuid]

 VMware 仮想マシン名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **name** - カスタム属性名

vmware.vm.consolidationneeded[url,uuid]

 VMware 仮想マシンディスクの統合が必要です。
 戻り値: *String*: *true* - 統合が必要; *false* - 統合が不要。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.latency[url,uuid]

 仮想マシンが物理 CPU へのアクセスを競合しているために実行できない時間の割合（パーセンテージ）。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.num[url,uuid]

 VMware 仮想マシンのプロセッサ数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.readiness[url,uuid,<instance>]

 仮想マシンが実行する準備ができていたが、物理 CPU でスケジュールできなかった時間の割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - CPU インスタンス

vmware.vm.cpu.ready[url,uuid]

 仮想マシンが実行の準備ができていたが、物理 CPU でスケジュールできなかった時間（ミリ秒単位）。CPU レディタイムは、ホスト上の仮想マシンの数とその CPU 負荷（%）に依存します。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 スワップインを待っている CPU 時間の割合。
 戻り値：*Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - CPU インスタンス

vmware.vm.cpu.usage[url,uuid]

 VMware 仮想マシンのプロセッサ使用率 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.usage.perf[url,uuid]

 インターバル中の VMware 仮想マシンのプロセッサ使用率（パーセント）。
 戻り値：*Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.datacenter.name[url,uuid]

 VMware 仮想マシンのデータセンター名。
 戻り値: *String*。

パラメーター:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.discoveryurl

 VMware 仮想マシンのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.vm.guest.memory.size.swapped[url,uuid]

 スワップ領域にスワップアウトされたゲスト物理メモリの量 (KB)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.guest.osuptime[url,uuid]

 最後のオペレーティングシステムの起動から経過した合計時間（秒単位）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.hv.maintenance[url,uuid]

 VMware 仮想マシンのハイパーバイザーのメンテナンス状態。
 戻り値: *0* - メンテナンス中でない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.hv.name[url,uuid]

 VMware 仮想マシンのハイパーバイザー名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size[url,uuid]

 VMware 仮想マシンの合計メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.ballooned[url,uuid]

 VMware 仮想マシンのバルーンメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.compressed[url,uuid]

 VMware 仮想マシンの圧縮メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.consumed[url,uuid]

 ゲスト物理メモリページのバックアップに消費されたホスト物理メモリの量 (KB)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.private[url,uuid]

 VMware 仮想マシンのプライベートメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL

- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.shared[url,uuid]

 VMware 仮想マシンの共有メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.swapped[url,uuid]

 VMware 仮想マシンのスワップメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.usage.guest[url,uuid]

 VMware 仮想マシンのゲストメモリ使用量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.usage.host[url,uuid]

 VMware 仮想マシンのホストメモリ使用量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.usage[url,uuid]

 消費されたホスト物理メモリの割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.net.if.discovery[url,uuid]

 VMware 仮想マシンのネットワークインターフェースのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの入力統計情報 (1 秒あたりのバイト数/パケット数)。
 戻り値: 整数[2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス
- **mode** - *bps* (デフォルト) または *pps* - 1 秒あたりのバイト数またはパケット数

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの出力統計情報 (1 秒あたりのバイト数/パケット数)。
 戻り値: 整数[2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス
- **mode** - *bps* (デフォルト) または *pps* - 1 秒あたりのバイト数またはパケット数

vmware.vm.net.if.usage[url,uuid,<instance>]

 インターバル中の VMware 仮想マシンのネットワーク使用率（送信レートと受信レートの合計）(KBps)。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス

vmware.vm.perfcounter[url,uuid,path,<instance>]

 VMware 仮想マシンのパフォーマンスカウンター値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **path** - パフォーマンスカウンターのパス [1]
- **instance** - パフォーマンスカウンターのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。

vmware.vm.powerstate[url,uuid]

 VMware 仮想マシンの電源状態。
 戻り値: *0* - poweredOff; *1* - poweredOn; *2* - suspended。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.property[url,uuid,prop]

 VMware 仮想マシンのプロパティ。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **prop** - プロパティパス

vmware.vm.snapshot.get[url,uuid]

 VMware 仮想マシンのスナップショットの状態。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.state[url,uuid]

 VMware 仮想マシンの状態。
 戻り値：*String*：*notRunning*、*resetting*、*running*、*shuttingDown*、*standby*、または *unknown*。

パラメータ：

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.committed[url,uuid]

 VMware 仮想マシンのコミット済みストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.readoio[url,uuid,instance]

 収集間隔中に仮想ディスクへの未処理の読み取り要求の平均数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 仮想ディスクからの読み取りにかかる平均時間（ミリ秒）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 仮想ディスクへの書き込みにかかる平均時間（ミリ秒）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスインスタンス

vmware.vm.storage.uncommitted[url,uuid]

 VMware 仮想マシンの未コミットストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.unshared[url,uuid]

 VMware 仮想マシンの未共有ストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.writeoio[url,uuid,instance]

 収集間隔中に仮想ディスクへの未処理の書き込み要求の平均数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスインスタンス

vmware.vm.tags.get[url,uuid]

 VMware 仮想マシンのタグの配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

このアイテムは vSphere 6.5 以降で動作します。

vmware.vm.tools[url,uuid,<mode>]

 VMware 仮想マシンのゲストツールの状態。
 戻り値: *String*。
 status モードの場合: *guestToolsExecutingScripts* - VMware Tools が起動中; *guestToolsNotRunning* - VMware Tools が実行されていない; *guestToolsRunning* - VMware Tools が実行されている。
 version モードの場合: バージョン。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **mode** - *version*、*status* (デフォルト)

vmware.vm.uptime[url,uuid]

 VMware 仮想マシンの稼働時間 (秒)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.vfs.dev.discovery[url,uuid]

 VMware 仮想マシンのディスクデバイスのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイスの読み取り統計情報 (1 秒あたりのバイト数/操作数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス
- **mode** - *bps* (デフォルト) または *ops* - 1 秒あたりのバイト数または操作数

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイスの書き込み統計情報 (1 秒あたりのバイト数/操作数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス
- **mode** - *bps* (デフォルト) または *ops* - 1 秒あたりのバイト数または操作数

vmware.vm.vfs.fs.discovery[url,uuid]

 VMware 仮想マシンのファイルシステムのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

このアイテムを動作させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 VMware 仮想マシンのファイルシステム統計情報 (バイト/パーセンテージ)。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **fsname** - ファイルシステム名
- **mode** - *total*、*free*、*used*、*pfree*、または *pused*

このアイテムを動作させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

脚注

[1] VMware 用のカスタムパフォーマンスカウンタ名の作成を参照してください。

[2] これらのアイテムの値は VMware のパフォーマンスカウンタから取得され、VMwarePerfFrequency パラメータが Zabbix VMware キャッシュ内のデータを更新するために使用されます:

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in
- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

詳細

Zabbix で VMware 環境を監視するための詳細な設定方法は、仮想マシン監視 を参照してください。

2 仮想マシンディスカバリキーフィールド

次の表は、仮想マシン関連のディスカバリキーが返すフィールドの一覧です。

アイテムキー		
説明	フィールド	取得したコンテンツ
vmware.cluster.discovery		
Performs cluster discovery.	{#CLUSTER.ID}	Cluster identifier.
	{#CLUSTER.NAME}	Cluster name.

	”resource_pool”	An array containing resource pool data, including resource group ID, tags array, resource pool path, number of virtual machines.
		
		Array structure:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`

	”tags”	For `"tags"` array structure, see the ”tags” field.
		An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Performs datastore discovery.	{#DATASTORE}	Datastore name.

	{#DATASTORE.EXTENT}	An array containing datastore extent partition ID and instance name.
		
		Array structure:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Datastore type. Value examples: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Datastore identifier.

	”tags”	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dc.discovery		
Performs datacenter discovery.	{#DATACENTER}	Datacenter name.

アイテムキー		
	{#DATACENTERID}	Datacenter identifier.
	”tags”	An array containing tags with tag name, description and category. Array structure: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dvswitch.discovery		
Performs vSphere distributed switches discovery.	{#DVS.NAME}	Switch name.
	{#DVS.UUID}	Switch identifier.
vmware.hv.discovery		
Performs hypervisor discovery.	{#HV.UUID}	Unique hypervisor identifier.
	{#HV.ID}	Hypervisor identifier (HostSystem managed object name).
	{#HV.NAME}	Hypervisor name.
	{#HV.NETNAME}	Hypervisor network host name.
	{#HV.IP}	Hypervisor IP address, might be empty. In case of an HA configuration with multiple net interfaces, the following selection priority for interface is observed: - prefer the IP which shares the IP-subnet with the vCenter IP; - prefer the IP from IP-subnet with default gateway; - prefer the IP from interface with the lowest ID.
	{#CLUSTER.NAME}	Cluster name, might be empty.
	{#DATACENTER.NAME}	Datacenter name.
	{#PARENT.NAME}	Name of container that stores the hypervisor.
	{#PARENT.TYPE}	Type of container in which the hypervisor is stored. The values could be `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, where ”VMware” stands for unknown container type.
	”resource_pool”	An array containing resource pool data, including resource group ID, tags array, resource pool path, number of virtual machines. Array structure: [{ "rpid":"resource group id", "tags":[{}], "rpath":"resource group path", "vm_count":0 }] For `"tags"` array structure, see the ”tags” field.
	”tags”	An array containing tags with tag name, description and category. Array structure: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.hv.datastore.discovery		
Performs hypervisor datastore discovery. Note that multiple hypervisors can use the same datastore.	{#DATASTORE}	Datastore name.

アイテムキー		
	{#DATASTORE.TYPE}	Datastore type. Value examples: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Datastore identifier.
	{#MULTIPATH.COUNT}	Registered number of datastore paths.
	{#MULTIPATH.PARTITION.COUNT}	Number of available disk partitions.

	”datastore_extent”	An array containing datastore extent instance name and partition ID.
		
		Array structure:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	”tags”	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Performs hypervisor network interfaces discovery.	{#IFNAME}	Interface name.
	{#IFDRIVER}	Interface driver.
	{#IFDUPLEX}	Interface duplex settings.
	{#IFSPEED}	Interface speed.
	{#IFMAC}	Interface mac address.
vmware.vm.discovery		
Performs virtual machine discovery.	{#VM.UUID}	Unique virtual machine identifier.
	{#VM.ID}	Virtual machine identifier (VirtualMachine managed object name).
	{#VM.NAME}	Virtual machine name.
	{#HV.NAME}	Hypervisor name.
	{#HV.UUID}	Unique hypervisor identifier.
	{#HV.ID}	Hypervisor identifier (HostSystem managed object name).
	{#CLUSTER.NAME}	Cluster name, might be empty.
	{#DATACENTER.NAME}	Datacenter name.
	{#DATASTORE.NAME}	Datastore name.
	{#DATASTORE.UUID}	Datastore identifier.
	{#VM.IP}	Virtual machine IP address, might be empty.
	{#VM.DNS}	Virtual machine DNS name, might be empty.
	{#VM.GUESTFAMILY}	Guest virtual machine OS family, might be empty.
	{#VM.GUESTFULLNAME}	Full guest virtual machine OS name, might be empty.
	{#VM.FOLDER}	The chain of virtual machine parent folders, can be used as value for nested groups; folder names are combined with ”/”. Might be empty.
	{#VM.TOOLS.STATUS}	VMware virtual machine tools state.
	{#VM.POWERSTATE}	VMware virtual machine power state (`poweredOff`, `poweredOn`, or `suspended`).
	{#VM.RPOOL.ID}	Resource pool identifier.
	{#VM.RPOOL.PATH}	Full resource pool path excluding the ”root” name ”Resources”. Folder names are combined with ”/”.
	{#VM.SNAPSHOT.COUNT}	Number of VM snapshots.

アイテムキー		

	”tags”	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	”vm_customattribute”	An array of virtual machine custom attributes (if defined).
		
		Array structure:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	”net_if”	An array of virtual machine network interfaces.
		
		Array structure:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		For the description of returned data, see the ”vmware.vm.net.if.discovery” item key.

vmware.vm.net.if.discovery		
Performs virtual machine network interface discovery.	{#IFNAME}	Network interface name.
	{#IFDESC}	Interface description.
	{#IFMAC}	Interface mac address.
	{#IFCONNECTED}	Interface connection status (`false` - disconnected; `true` - connected).
	{#IFTYPE}	Interface type.
	{#IFBACKINGDEVICE}	Name of the backing device.
	{#IFDVSWITCH.UUID}	Unique vSphere Distributed Switch identifier.
	{#IFDVSWITCH.PORTGROUP}	Distributed port group.
	{#IFDVSWITCH.PORT}	vSphere Distributed Switch port.
	”ifip”	An array of interface addresses.
vmware.vm.vfs.dev.discovery		
Performs virtual machine disk device discovery.	{#DISKNAME}	Disk device name.
vmware.vm.vfs.fs.discovery		
Performs virtual machine file system discovery.	{#FSNAME}	File system name.

3 VMware アイテムの **JSON** の例

概要　このセクションでは、さまざまな VMwareアイテムによって返される JSON オブジェクトに関する追加情報を提供します。

vmware.*.alarms.get　**vmware.alarms.get[]**、**vmware.cluster.alarms.get[]**、**vmware.datastore.alarms.get[]**、**vmware.dc.alarms.**
vmware.hv.alarms.get[]、**vmware.vm.alarms.get[]** のアイテムは、以下の構造を持つ JSON オブジェクトを返します（値は例です）：

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get　**vmware.cluster.tags.get[]**、**vmware.datastore.tags.get[]**、**vmware.dc.tags.get[]**、**vmware.hv.tags.get[]**、
vmware.vm.tags.get[] のアイテムは、以下の構造を持つ JSON オブジェクトを返します（値は例として示しています）：

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "tag for cat OS type",
        "category": "OS type"
    },
    {
            "name": "SQL Server",
        "description": "tag for cat application name",
        "category": "application name"
        },
    {
            // other tags
        }
    ]
}
```

vmware.hv.diskinfo.get　アイテム **vmware.hv.diskinfo.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例です）：

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
    "datastore_uuid": "",
    "operational_state": [
```

```
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    // 他のインスタンス
  }
]
```

vmware.dvswitch.fetchports.get　アイテム **vmware.dvswitch.fetchports.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例として提供されています）：

```
{
    "FetchDVPortsResponse":
    {
        "returnval": [
            {
                "key": "0",
                "dvsUuid": "50 36 6a 24 25 c0 10 9e-05 4a f6 ea 4e 3d 09 88",
                "portgroupKey": "dvportgroup-2023",
                "proxyHost":
                {
                    "@type": "HostSystem",
                    "#text": "host-2021"
                },
                "connectee":
                {
                    "connectedEntity":
                    {
                        "@type": "HostSystem",
                        "#text": "host-2021"
                    },
                    "nicKey": "vmnic0",
                    "type": "pnic"
                },
                "conflict": "false",
                "state":
                {
                    "runtimeInfo":
                    {
                        "linkUp": "true",
                        "blocked": "false",
                        "vlanIds":
                        {
                            "start": "0",
                            "end": "4094"
                        },
                        "trunkingMode": "true",
                        "linkPeer": "vmnic0",
                        "macAddress": "00:00:00:00:00:00",
                        "statusDetail": null,
                        "vmDirectPathGen2Active": "false",
                        "vmDirectPathGen2InactiveReasonOther": "portNptIncompatibleConnectee"
                    },
                    "stats":
                    {
                        "packetsInMulticast": "2385470",
```

```
                        "packetsOutMulticast": "45",
                        "bytesInMulticast": "309250248",
                        "bytesOutMulticast": "5890",
                        "packetsInUnicast": "155601537",
                        "packetsOutUnicast": "113008658",
                        "bytesInUnicast": "121609489384",
                        "bytesOutUnicast": "47240279759",
                        "packetsInBroadcast": "1040420",
                        "packetsOutBroadcast": "7051",
                        "bytesInBroadcast": "77339771",
                        "bytesOutBroadcast": "430392",
                        "packetsInDropped": "0",
                        "packetsOutDropped": "0",
                        "packetsInException": "0",
                        "packetsOutException": "0"
                    }
                },
                "connectionCookie": "1702765133",
                "lastStatusChange": "2022-03-25T14:01:11Z",
                "hostLocalPort": "false"
            },
            {
                //other keys
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get アイテム **vmware.hv.hw.sensors.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例として提供されています）：

```
{
    "val":
        {
            "@type": "HostHardwareStatusInfo",
            "storageStatusInfo": [
                {
                    "name": "Intel Corporation HD Graphics 630 #2",
                "status":
                        {
                            "label": "Unknown",
                "summary": "Cannot report on the current status of the physical element",
                "key": "Unknown"
                }
            },
            {
                    "name": "Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller #20"
                "status":
                        {
                     "label": "Unknown",
                     "summary": "Cannot report on the current status of the physical element",
                     "key": "Unknown"
                 }
            },
            {
                    // other hv hw sensors
                }
        ]
        }
}
```

vmware.hv.sensors.get アイテム **vmware.hv.sensors.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例です）：

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // other hv sensors
                }
        ]
        }
}
```


vmware.vm.snapshot.get　スナップショットが存在する場合、アイテム **vmware.snapshot.get[]** は以下の構造の JSON オブジェクトを返します（値は例です）：


```
{
  "snapshot": [
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
      "size": 5755795171,
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
```

```
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```


スナップショットが存在しない場合、アイテム **vmware.snapshot.get[]** は空の値を持つ JSON オブジェクトを返します:


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


4 VMware 監視設定例

概要

次の例では、VMware 仮想マシンの監視のために Zabbix をセットアップする方法について説明します。これには以下が含まれます。

- VMware 環境を表すホストの作成
- VMware 環境内の仮想マシンを検出するローレベルディスカバリルールの作成
- ローレベルディスカバリルールによって検出された仮想マシンのために、Zabbix が実際のホストを生成するためのホストプロトタイプの作成

前提条件

> **Note:**
> この例では VMware の設定については説明していません。VMware がすでに設定されていることを前提としています。

続行する前に、Zabbix サーバーの設定ファイルで`StartVMwareCollectors`パラメーターを 2以上に設定してください（デフォルト値は 0 です）。

ホストの作成

1. データ収集 → ホストに移動します。

2. ホストを作成します。

- ホスト名フィールドにホスト名（例：”VMware VMs”）を入力します。
- ホストグループフィールドにホストグループ（例：”Virtual machines”）を入力または選択します。
- マクロタブで、以下のホストマクロを設定します。
  - {$VMWARE.URL} - VMware サービス（ESXi ハイパーバイザー）SDK の URL（https://servername/sdk）
  - {$VMWARE.USERNAME} - VMware サービスのユーザー名
  - {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード

3. 追加ボタンをクリックしてホストを作成します。このホストは VMware 環境を表します。

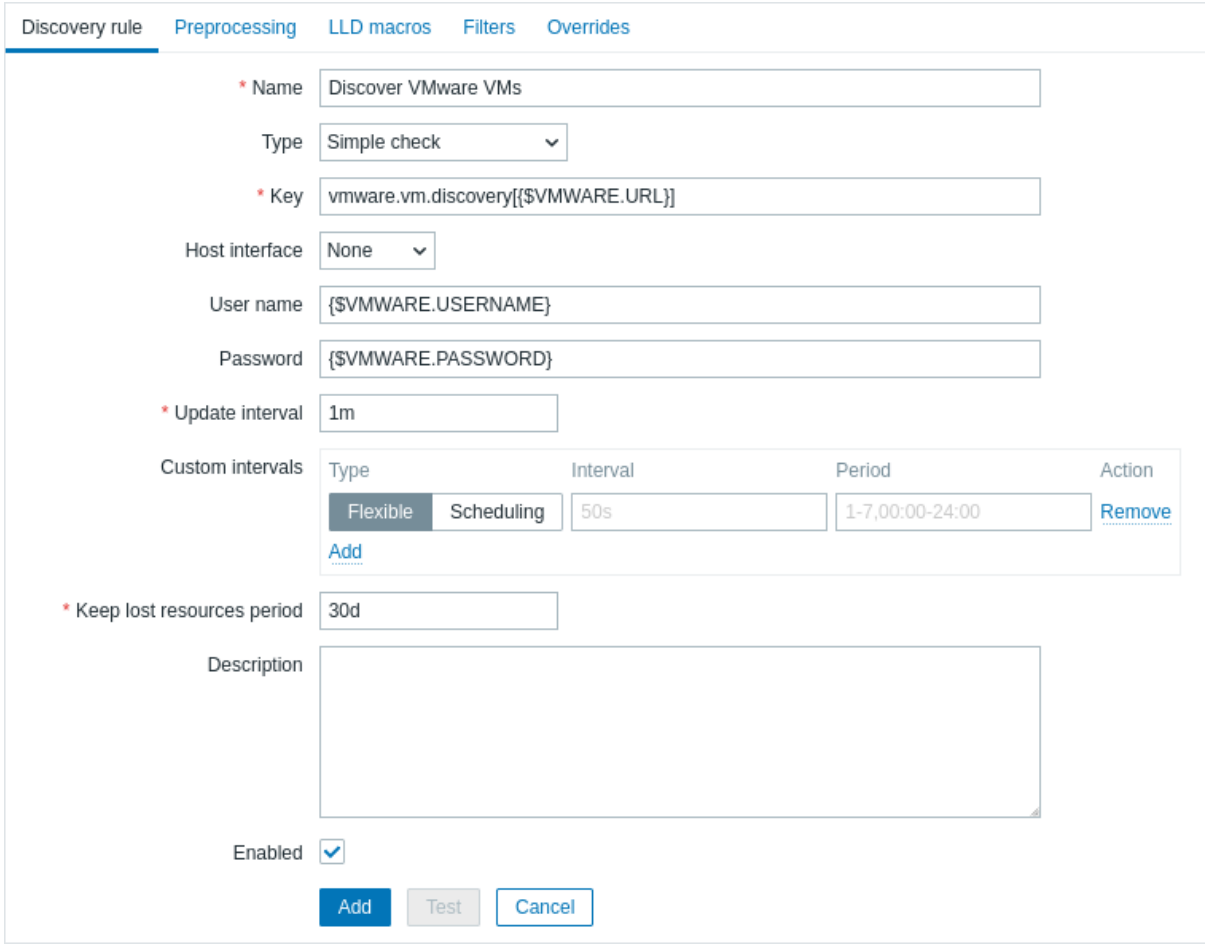
ローレベルディスカバリルールの作成

1. 作成したホストのディスカバリをクリックして、そのホストのローレベルディスカバリルールの一覧に移動します。

2. ローレベルディスカバリルールを作成します:

- 名前フィールドに、ローレベルディスカバリルール名（例：”Discover VMware VMs”）を入力します。

- タイプフィールドで"シンプルチェック" を選択します。
- キーフィールドに、VMware 仮想マシンをディスカバリするための組み込みアイテムキー`vmware.vm.discovery[{$VMWARE.URL}]`を入力します。
- ユーザー名およびパスワードフィールドに、ホストで事前に設定した対応するマクロを入力します。



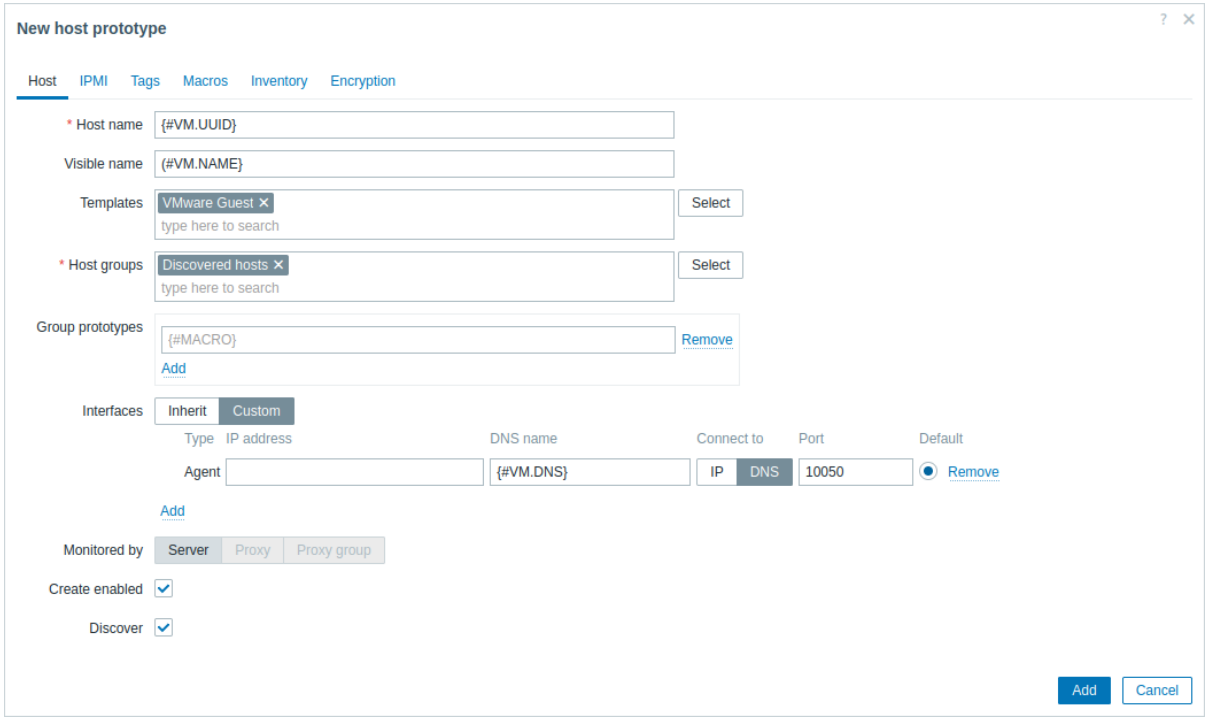
3. 追加ボタンをクリックして、ローレベルディスカバリルールを作成します。このディスカバリルールは、VMware 環境内の仮想マシンをディスカバリします。

ホストプロトタイプの作成

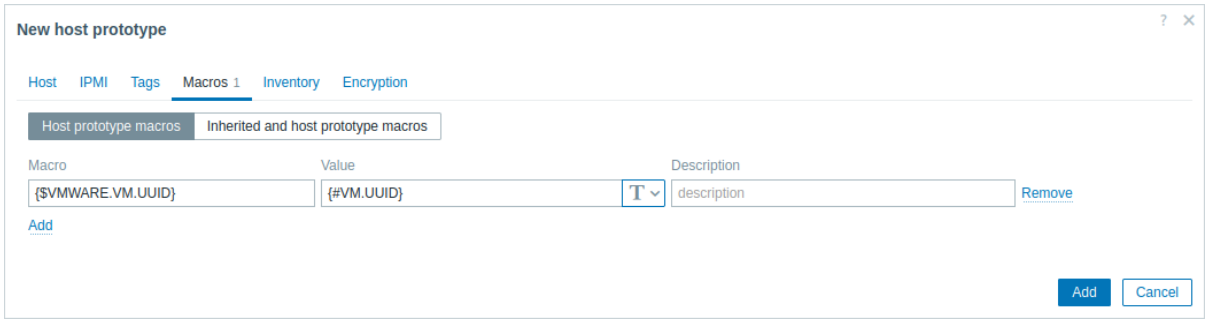
1. 低レベルディスカバリルールの一覧で、先ほど作成した低レベルディスカバリルールのホストプロトタイプをクリックします。

2. ホストプロトタイプを作成します。ホストプロトタイプは低レベルディスカバリルールによってホストを作成するための設計図であるため、ほとんどのフィールドには低レベルディスカバリマクロが含まれます。これにより、先ほど作成した低レベルディスカバリルールで取得した内容に基づいてホストが作成されます。

- ホスト名フィールドには、マクロ {#VM.UUID} を入力します。
- 表示名フィールドには、マクロ {#VM.NAME} を入力します。
- テンプレートフィールドには、「VMware Guest」テンプレートを入力または選択します。このテンプレートには、仮想マシンの電源状態、CPU 使用率、メモリ使用率、ネットワークデバイスなどを監視するためのVMware アイテムおよびディスカバリルールが含まれています。
- ホストグループフィールドには、ホストグループ (例:「Discovered hosts」) を入力または選択します。
- インターフェースフィールドには、カスタムホストインターフェースを追加します。その後、*DNS* 名フィールドにマクロ {#VM.DNS} を入力するか、*IP* アドレスフィールドにマクロ {#VM.IP} を入力します。または、VMware 環境の仮想マシンに複数のインターフェースがある場合は、高度なホストインターフェースの設定セクションに進んでください。カスタムホストインターフェースの設定は、*VMware Guest* テンプレートを正しく機能させるために必要です。



- マクロタブで、{$VMWARE.VM.UUID} マクロに値 {#VM.UUID} を設定します。これは、このマクロをアイテムパラメータのホストレベルのユーザーマクロとして使用する *VMware Guest* テンプレートを正しく機能させるために必要です (例: *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*)。



3. 追加ボタンをクリックしてホストプロトタイプを作成します。このホストプロトタイプは、先ほど作成した低レベルディスカバリルールで検出された仮想マシンのホストを作成するために使用されます。

ホストとメトリクスの表示

ホストプロトタイプが作成されると、ローレベルディスカバリルールによって検出された VMware 仮想マシンのホストが作成され、Zabbix はそれらの監視を開始します。必要に応じて、ホストのディスカバリと作成は手動で実行することもできます。

作成されたホストを表示するには、データ収集 → ホストメニューセクションに移動します。

収集されたメトリクスを表示するには、監視 → ホストメニューセクションに移動し、いずれかのホストの最新データをクリックします。

高度なホストインターフェースの設定

「低レベルディスカバリルールの作成」セクションで設定した `vmware.vm.discovery[{$VMWARE.URL}]` アイテムキーは、"net_if" フィールドでネットワークインターフェースのデータを返します。

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
```

```
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

このデータを使用してカスタムホストインターフェースを設定できます。

1. 低レベルディスカバリルールを作成する際に、追加で低レベルディスカバリマクロを設定します。*LLD* マクロタブで、JSONPath値を持つカスタム LLD マクロを作成します。例:

- {#MYLAB.NET.IF} - $.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()

2. ホストプロトタイプを作成する際に、カスタムホストインターフェースを追加し、*DNS* 名または *IP* アドレスフィールドに LLD マクロを入力します。

9 メンテナンス

概要　Zabbix では、ホストやホストグループに対してメンテナンス期間を定義できます。

さらに、トリガータグを指定することで、単一のトリガー（またはトリガーのサブセット）に対してのみメンテナンスを定義することも可能です。この場合、メンテナンスはそれらのトリガーにのみ有効となり、ホストやホストグループの他のすべてのトリガーはメンテナンス対象外となります。

メンテナンスタイプには、「データ収集あり」と「データ収集なし」の 2 種類があります。

「データ収集あり」メンテナンス中は、トリガーは通常通り処理され、必要に応じてイベントが作成されます。ただし、アクション設定で 抑制された障害の処理を一時停止オプションがチェックされている場合、メンテナンス中のホスト/トリガーに対する障害のエスカレーションは一時停止されます。この場合、通知の送信やリモートコマンドの実行を含む可能性のあるエスカレーションステップは、メンテナンス期間が続く限り無視されます。なお、メンテナンス中に抑制されるのはエスカレーションのみであり、障害の復旧や更新の処理は抑制されません。

例えば、障害発生後 0 分、30 分、60 分にエスカレーションステップがスケジュールされていて、実際の障害発生から 10 分後から 40 分後までの 30 分間メンテナンスが行われる場合、2 番目と 3 番目のステップは 30 分遅れて、60 分後と 90 分後に実行されます（障害がまだ存在する場合）。同様に、メンテナンス中に障害が発生した場合、エスカレーションはメンテナンス終了後に開始されます。

メンテナンス中も通常通り（遅延なく）障害通知を受け取りたい場合は、アクション設定で 抑制された障害の処理を一時停止オプションのチェックを外してください。

> **Note:**
> トリガー式で使用されているホストのうち、少なくとも 1 つがメンテナンスモードでない場合、Zabbix は障害通知を送信します。

メンテナンス中は Zabbix サーバーが稼働している必要があります。メンテナンスは毎分、またはメンテナンス期間に変更があった場合は設定キャッシュがリロードされるとすぐに再計算されます。

タイマープロセスは、毎分 0 秒にホストのステータスをメンテナンスに変更する必要があるかどうかをチェックします。さらに、設定更新後にメンテナンス期間に変更があったかどうかに基づいて、毎秒タイマープロセスがメンテナンスの開始/終了をチェックします。したがって、メンテナンス期間の開始/終了の速度は、設定の更新間隔（デフォルトは 10 秒）に依存します。なお、メンテナンス期間の変更には 有効期間/有効終了の設定は含まれません。また、既存のアクティブなメンテナンス期間にホスト/ホストグループが追加された場合、変更は次の分の開始時にタイマープロセスによって有効化されます。

ホストがメンテナンスに入ると、Zabbix サーバーのタイマープロセスはすべての未解決の障害を読み取り、それらを抑制する必要があるかどうかをチェックします。未解決の障害が多数ある場合、パフォーマンスに影響を与える可能性があります。また、Zabbix サーバーは起動時にもすべての未解決の障害を読み取ります（その時点でメンテナンスが設定されていなくても）。

Zabbix サーバー（またはプロキシ）は、メンテナンスタイプに関係なく（「データ収集なし」メンテナンスを含む）常にデータを収集します。「データ収集なし」が設定されている場合、そのデータはサーバーによって後で無視されます。

「データ収集なし」メンテナンスが終了すると、nodata() 関数を使用するトリガーは、監視期間中の次のチェックまで発火しません。

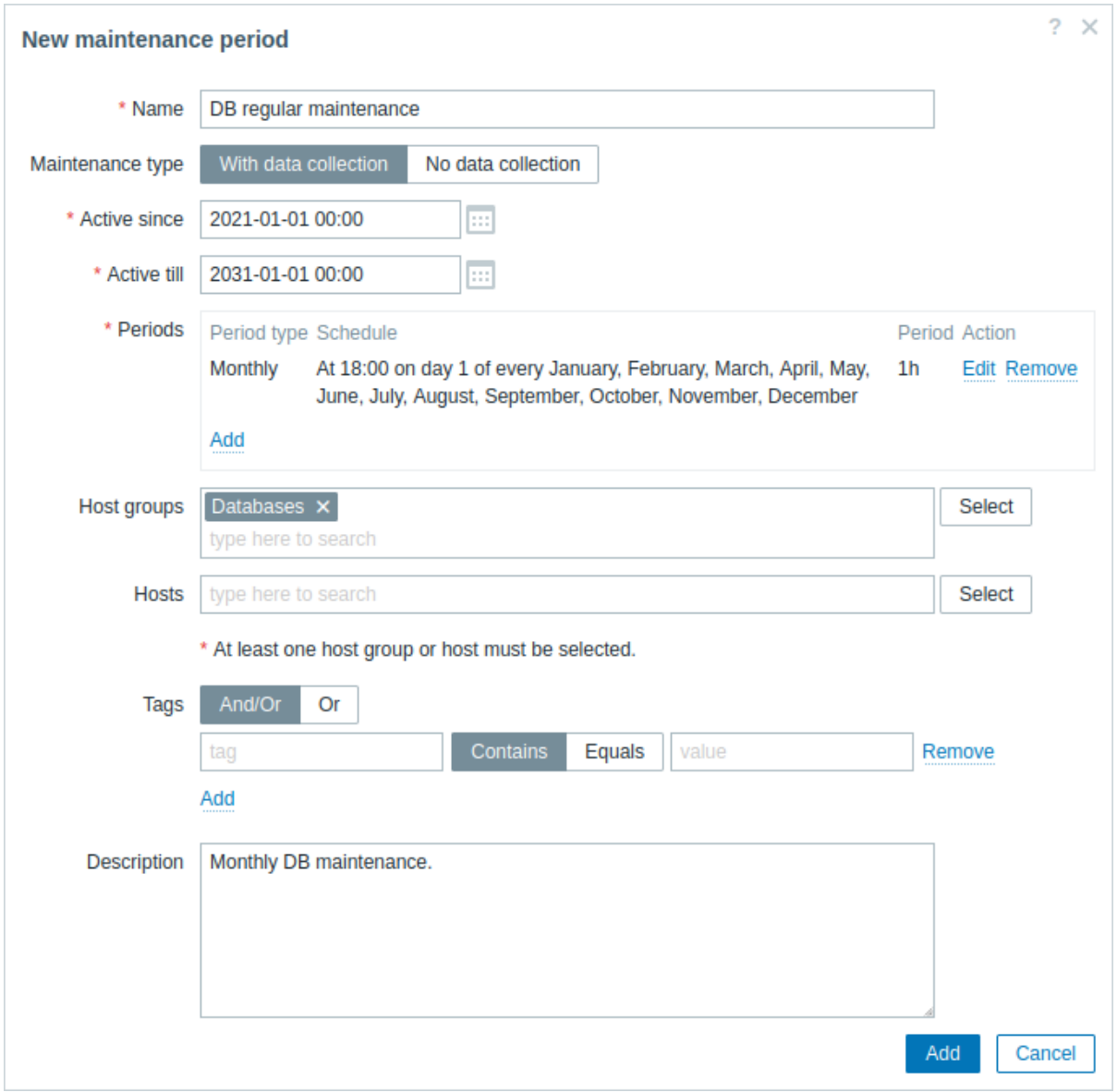
ホストがメンテナンス中にログアイテムが追加され、メンテナンスが終了した場合、メンテナンス終了以降の新しいログファイルエントリのみが収集されます。

「データ収集なし」メンテナンスタイプのホストにタイムスタンプ付きの値が送信された場合（例：Zabbix senderを使用）、この値は破棄されますが、期限切れのメンテナンス期間にタイムスタンプ付きの値を送信することは可能であり、その場合は受け入れられます。

ユーザーがメンテナンス期間、ホスト、グループ、タグを変更した場合、変更は設定キャッシュの同期後にのみ有効になります。

設定　メンテナンス期間を設定するには:

1. データ収集 > メンテナンスに移動します。
2. メンテナンス期間の作成 (または既存のメンテナンス期間の名前) をクリックします。
3. フォームにメンテナンスのパラメータを入力します。



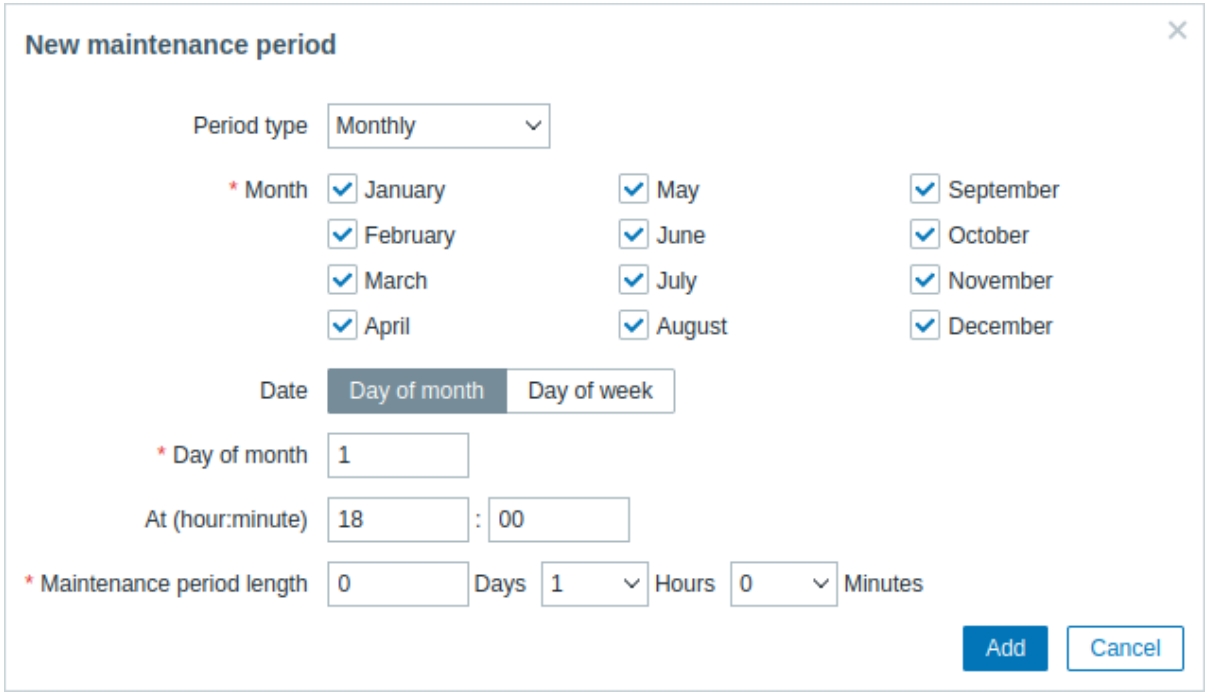
すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	メンテナンス期間の名前。

パラメータ	説明
メンテナンスタイプ	2 種類のメンテナンスを設定できます: データ収集あり - メンテナンス中もサーバーがデータを収集し、トリガーを処理します; データ収集なし - メンテナンス中はサーバーがデータを収集しません。 各タイプが可用性レポートにどのように影響するかは、メンテナンス期間の効果を参照してください。
有効期間 (開始)	メンテナンス期間の実行が有効になる日時。 注: この時間を設定するだけではメンテナンス期間は有効になりません。期間 (下記参照) でメンテナンス期間を設定する必要があります。
有効期間 (終了)	メンテナンス期間の実行が無効になる日時。
期間	このブロックでは、メンテナンスが行われる正確な日付と時間を定義できます。Addをクリックすると、柔軟なメンテナンス期間フォームを持つポップアップウィンドウが開き、メンテナンススケジュールを定義できます。詳細はメンテナンス期間を参照してください。
ホストグループ	メンテナンスを有効にするホストグループを選択します。指定したホストグループのすべてのホストに対してメンテナンスが有効になります。このフィールドはオートコンプリート対応なので、入力を始めると利用可能なホストグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループも暗黙的に選択されます。したがって、ネストされたグループのホストにもメンテナンスが有効になります。
ホスト	メンテナンスを有効にするホストを選択します。このフィールドはオートコンプリート対応なので、入力を始めると利用可能なホストのドロップダウンが表示されます。
タグ	メンテナンス中のホストで一致するタグを持つ問題を抑制するためのタグを指定します。 複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件には 2 つの演算子があります: 含む - 指定したタグ名のうち、タグ値が入力した文字列を含むものを含めます (部分一致、大文字小文字区別); 等しい - 指定したタグ名と値を含めます (大文字小文字区別)。 条件の計算タイプは 2 つあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 タグはデータ収集ありモードが選択されている場合のみ指定できます。
説明	メンテナンス期間の説明。

メンテナンス期間

メンテナンス期間ウィンドウは、定期的または一度限りのメンテナンスのスケジュールを設定するためのものです。フォームは動的で、選択した期間タイプに応じて利用可能なフィールドが変化します。



期間タイプ	説明
一度限り	一度限りのメンテナンス期間を設定します: 日付 - メンテナンス期間の日付と時刻; メンテナンス期間の長さ - メンテナンスが有効である期間。
毎日	毎日のメンテナンス期間を設定します: 毎日 - メンテナンスの頻度（1 - *(デフォルト)* 毎日、2 - 2 日ごと、など）; 時刻（時*:* 分） - メンテナンスが開始される時刻; メンテナンス期間の長さ - メンテナンスが有効である期間。 毎日パラメータが「1」より大きい場合、開始日は有効期間開始の時刻が属する日となります。例: - 有効期間開始が「2021-01-01 12:00」、毎日が「2」、時刻（時*:* 分）が「23:00」の場合、最初のメンテナンス期間は 1 月 1 日 23:00 に開始され、2 回目のメンテナンス期間は 1 月 3 日 23:00 に開始されます; - 有効期間開始が「2021-01-01 12:00」、毎日が「2」、時刻（時*:* 分）が「01:00」の場合、最初のメンテナンス期間は 1 月 3 日 01:00 に開始され、2 回目のメンテナンス期間は 1 月 5 日 01:00 に開始されます。
毎週	毎週のメンテナンス期間を設定します: 毎週 - メンテナンスの頻度（1 - *(デフォルト)* 毎週、2 - 2 週間ごと、など）; 曜日 - メンテナンスを実施する曜日; 時刻（時*:* 分） - メンテナンスが開始される時刻; メンテナンス期間の長さ - メンテナンスが有効である期間。 毎週パラメータが「1」より大きい場合、開始週は有効期間開始の時刻が属する週となります。例については、上記の毎日パラメータの説明を参照してください。
毎月	毎月のメンテナンス期間を設定します: 月 - 定期メンテナンスを実施するすべての月を選択; 日付*:* 月の日 - 毎月同じ日にメンテナンスを実施する場合はこのオプションを選択し、表示される月の日フィールドで必要な日を選択; 日付*:* 曜日 - 特定の曜日にのみメンテナンスを実施する場合はこのオプションを選択し、（ドロップダウンで）月の必要な週（第 1 週、第 2 週、第 3 週、第 4 週、または最終週）を選択し、メンテナンス日をチェックボックスで指定; 時刻（時*:* 分） - メンテナンスが開始される時刻; メンテナンス期間の長さ - メンテナンスが有効である期間。

Attention:
メンテナンス期間を作成する際には、作成者のタイムゾーンが使用されます。ただし、定期的なメンテナンス期間（日次、週次、月次）がスケジュールされている場合は、Zabbix サーバーのタイムゾーンが使用されます。定期的なメンテナンス期間の予測可能な動作を保証するために、Zabbix のすべての部分で共通のタイムゾーンを使用する必要があります。

完了したら、追加を押してメンテナンス期間を期間ブロックに追加します。

サマータイム（DST）の変更は、メンテナンスの期間には影響しません。例えば、通常 01:00 に開始し 03:00 に終了する 2 時間のメンテナンスが設定されているとします。

- 1 時間のメンテナンス後（02:00）に DST の変更が発生し、現在時刻が 02:00 から 03:00 に変わった場合、メンテナンスはさらに 1 時間（04:00 まで）続きます。
- 2 時間のメンテナンス後（03:00）に DST の変更が発生し、現在時刻が 03:00 から 02:00 に変わった場合、2 時間が経過しているためメンテナンスは終了します。
- メンテナンス期間が DST の変更によってスキップされる 1 時間の間に開始される場合、メンテナンスは開始されません。

メンテナンス期間が「1 日」(Zabbix は日数を時間で計算するため、実際の期間は 24 時間）に設定され、00:00 に開始し翌日の 00:00 に終了する場合：

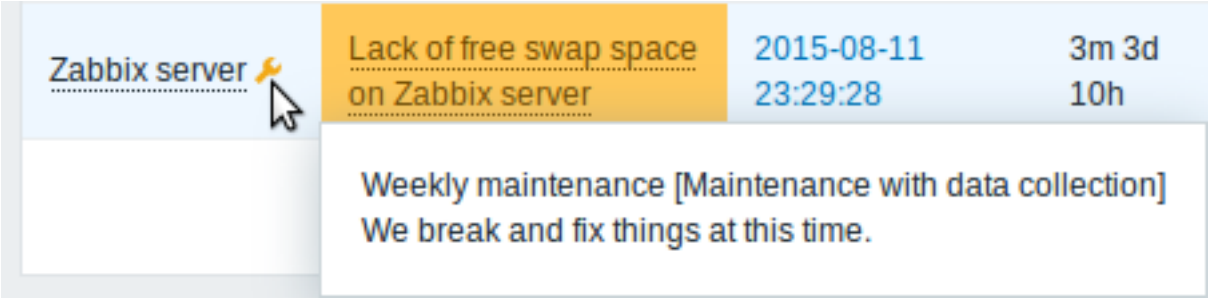
- 現在時刻が 1 時間進むと、メンテナンスは翌日の 01:00 に終了します。
- 現在時刻が 1 時間戻ると、メンテナンスはその日の 23:00 に終了します。

表示　メンテナンス中のホストの表示

ホスト名の横にオレンジ色のレンチアイコン が表示されている場合、そのホストは以下の画面でメンテナンス中であることを示します:

- ダッシュボード
- 監視 > 障害

- インベントリ > ホスト > ホストインベントリ詳細
- データ収集 > ホスト（「ステータス」列を参照）



アイコンの上にマウスポインタを置くと、メンテナンスの詳細が表示されます。

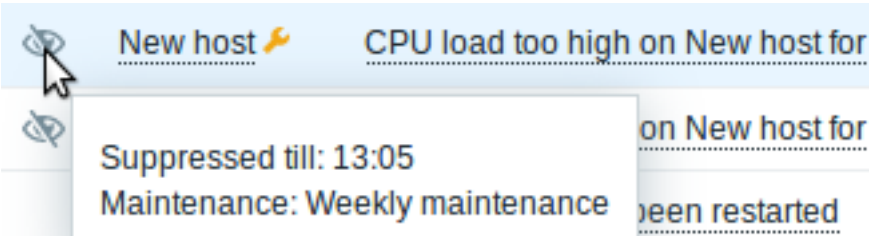
さらに、監視 > マップでは、メンテナンス中のホストはオレンジ色の背景で表示されます。

抑制された障害の表示

通常、メンテナンス中のホストの障害は抑制され、フロントエンドには表示されません。ただし、以下の場所で抑制された障害を表示オプションを選択することで、抑制された障害を表示するように設定することもできます。

- ダッシュボード（障害ホスト、障害、深刻度別障害、トリガー概要ウィジェットの設定）
- 監視 > 障害（フィルタ内）
- 監視 > マップ（マップ設定内）
- グローバル通知（ユーザープロファイル設定内）

抑制された障害が表示される場合、次のアイコンが表示されます: 。アイコンにマウスを重ねると、詳細が表示されます。



メンテナンス中のキューの計算　Zabbix フロントエンド（管理 > キュー）に表示されるキューは、Zabbix サーバーによって計算されます。これらは、データ収集のないメンテナンス中のアイテムを含みません。これらのアイテムの値が遅延している場合でも、キューの長さは常にゼロです。データ収集を伴うメンテナンス中の遅延アイテムは、引き続きキューにカウントされます。

Zabbix プロキシは、Zabbix サーバーとプロキシ間でメンテナンス設定の同期がないため、メンテナンス期間を認識しません。Zabbix プロキシで計算される内部チェック（たとえば、`zabbix[queue,,]`や`zabbix[stats,,,queue,,]`）は、Zabbix サーバー上のメンテナンスステータスに関係なく、遅延アイテムを報告します。

その結果、Zabbix フロントエンドと Zabbix プロキシの内部チェックによって、データ収集のないメンテナンス中の同じアイテムに対して異なるキューの長さが報告される場合があります。

10 正規表現

概要　Perl 互換の正規表現 (PCRE、PCRE2) が Zabbix でサポートされています。

Zabbix で正規表現を使用するには 2 つの方法があります。

- 正規表現を手動で入力する
- Zabbix で作成したグローバルな正規表現を使用する。

正規表現　サポートされている場所に正規表現を手動で入力することができます。ただし、Zabbix では @ 記号がグローバル正規表現を参照するために使用されているため、式の先頭に @ を付けることはできません。

> **Warning:**
> 正規表現を使用するとスタックが不足する可能性があります。詳細については pcrestack man ページ を参照してください。

複数行マッチングでは、^ および $ のアンカーはそれぞれ各行の先頭/末尾にマッチし、文字列全体の先頭/末尾にはマッチしないことに注意してください。

さまざまなコンテキストでの正しいエスケープの例も参照してください。

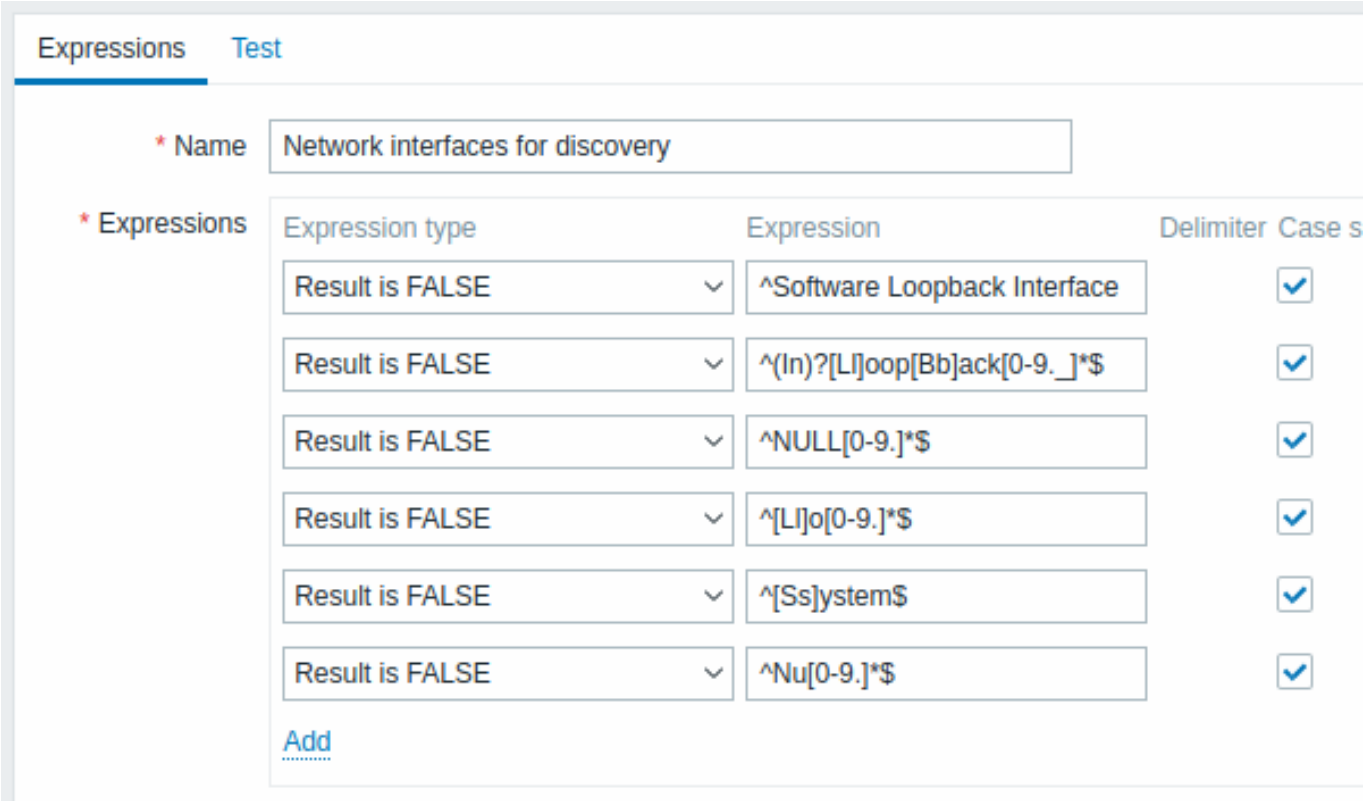
グローバルな正規表現　Zabbix フロントエンドで複雑な正規表現を作成およびテストするための高度なエディターがあります。

このようにして作成した正規表現は、*@mycustomregexp* のように、先頭に @ を付けたその名前を参照することで、フロントエンドのさまざまな場所で使用できます。

グローバルな正規表現を作成するには:

- 管理 → 一般設定に移動。
- ドロップダウンから正規表現を選択。
- 正規表現の作成をクリック。

条件式タブでは、正規表現名を設定し、部分式を追加できます。



必須入力項目には、赤いアスタリスクが表示されています。

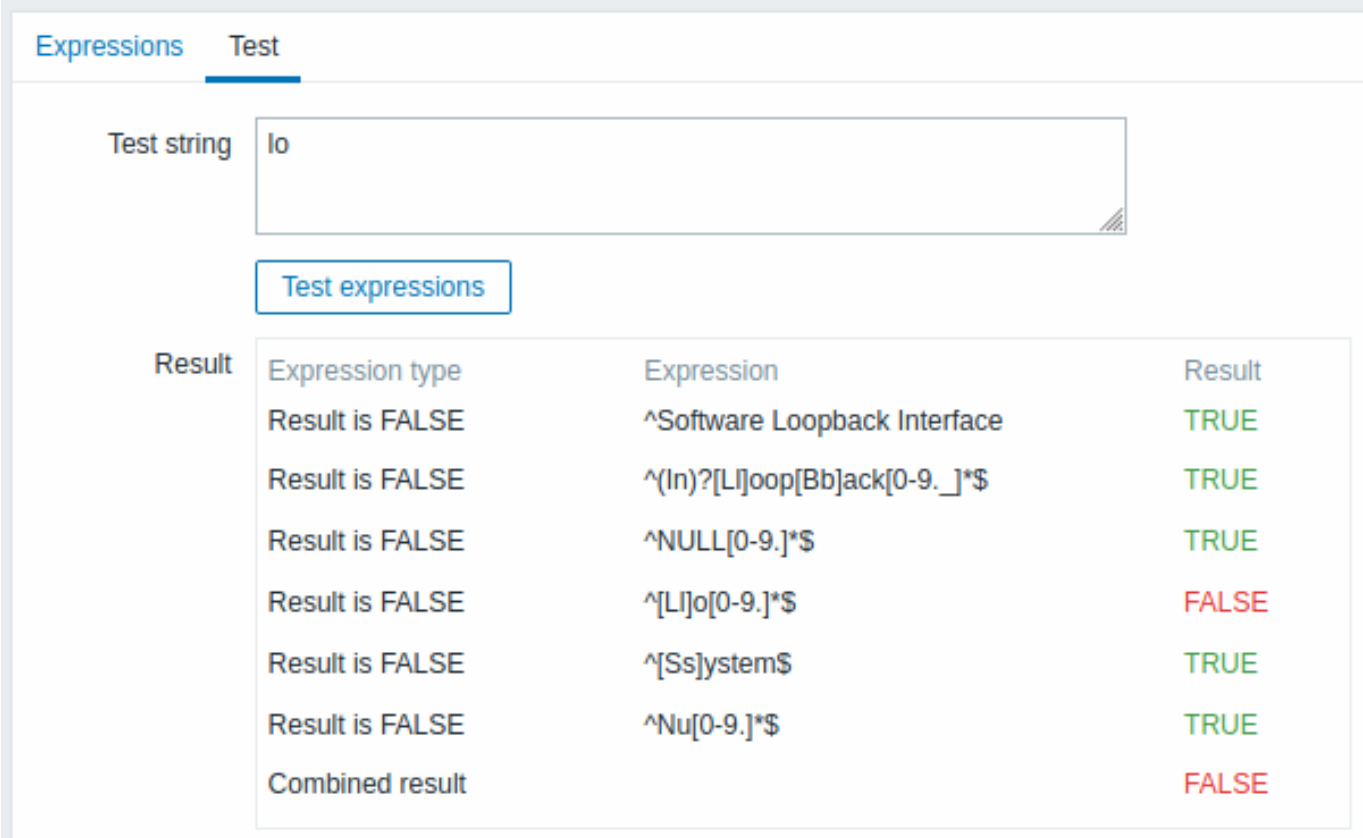
パラメータ		説明
名前		正規表現名を設定します。すべての Unicode 文字を使用できます。
条件式		条件式ブロックの 追加をクリックして、新しい部分式を追加します。
	条件式の形式	条件式の形式を選択します: 文字列が含まれる - 部分文字列と一致します いずれかの文字列が含まれる - 区切りリストの任意の部分文字列と一致します。区切りリストには、コンマ (,)、ドット (.)、またはスラッシュ (/) が含まれます。 文字列が含まれない - 部分文字列以外の任意の文字列に一致します 結果が真 - 正規表現に一致する 結果が偽 - 正規表現に一致しない
	条件式	部分文字列/正規表現を入力します。
区切り文字		正規表現でテキスト文字列を区切るには、コンマ (,)、ドット (.)、またはスラッシュ (/) を使用します。このパラメーターは条件式の形式が” いずれかの文字列が含まれる” を選択している場合にのみ有効です。

パラメータ	説明
大文字と小文字を区別	正規表現が文字の大文字と小文字を区別するかどうかを指定するチェックボックス

式内のスラッシュ (/) は、区切り文字ではなく文字どおりに扱われるので、スラッシュを含む式はエラーなしで保存できます。

Attention:
Zabbix のカスタム正規表現名には、カンマやスペースなどを含むことができます。そのため、参照時に誤解を招く可能性がある場合 (例えば、アイテムキーのパラメータにカンマがある場合)、参照全体を次のように引用符で囲むことができます。
"@My custom regexp for purpose1, purpose2".
正規表現名は、他の場所（たとえば LLD ルールのプロパティ）では引用符で囲まないでください。

テストタブでは、テスト文字列を指定することで、正規表現とその部分式をテストできます。



結果には、各部分式のステータスとカスタム式の合計ステータスが表示されます。

カスタム式の合計ステータスは、結合結果として定義されます。複数のサブ式が定義されている場合、Zabbix は AND 論理演算子を使用して 結合結果を計算します。これは、少なくとも 1 つの結果が False の場合、結合された結果も False ステータスになることを意味します。

デフォルトのグローバル正規表現　Zabbix には、デフォルトのデータセットにいくつかのグローバル正規表現が含まれています。

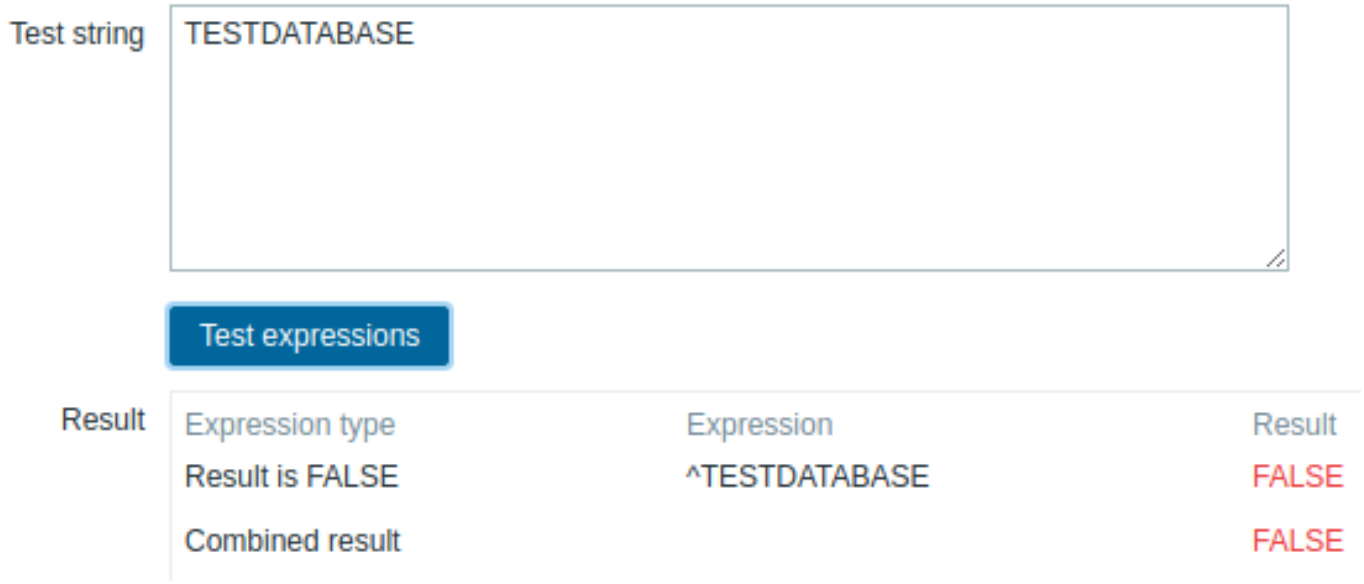
名前	式	一致する内容
ディスカバリ用ファイルシステム	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)$`	"btrfs" または"ext2" または"ext3" または"ext4" または"jfs" または"reiser" または"xfs" または"ffs" または"ufs" または"jfs" または"jfs2" または"vxfs" または"hfs" または"refs" または"apfs" または"ntfs" または"fat32" または"zfs"
ディスカバリ用ネットワークインターフェース	`^Software Loopback Interface`	"Software Loopback Interface" で始まる文字列
	`^lo$`	"lo"

名前	式	一致する内容
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	任意で”In” で始まり、次に”L” または”l”、”oop”、”B” または”b”、”ack” が続き、その後に任意の桁数の数字、ドット、アンダースコアが続く文字列
	`^NULL[0-9.]*$`	”NULL” で始まり、その後に任意の桁数の数字またはドットが続く文字列
	`^[Ll]o[0-9.]*$`	”Lo” または”lo” で始まり、その後に任意の桁数の数字またはドットが続く文字列
	`^[Ss]ystem$`	”System” または”system”
	`^Nu[0-9.]*$`	”Nu” で始まり、その後に任意の桁数の数字またはドットが続く文字列
SNMP ディスカバリ用ストレージデバイス	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” または”Virtual memory” または”Memory buffers” または”Cached memory” または”Swap space”
ディスカバリ用 *Windows* サービス名	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v4.0.30319_32)$`	”MMCSS” または”gupdate” または”SysmonLog” または”clr_optimization_v2.0.50727_32” や”clr_optimization_v4.0.30319_32” のような文字列（ドットの代わりに改行以外の任意の文字を入れることができる）
ディスカバリ用 *Windows* サービスの起動状態	`^(automatic|automatic delayed)$`	”automatic” または”automatic delayed”

例　例 1

低レベルの検出で次の式を使用すると、特定の名前のデータベース以外を検出できます。

`^TESTDATABASE$`

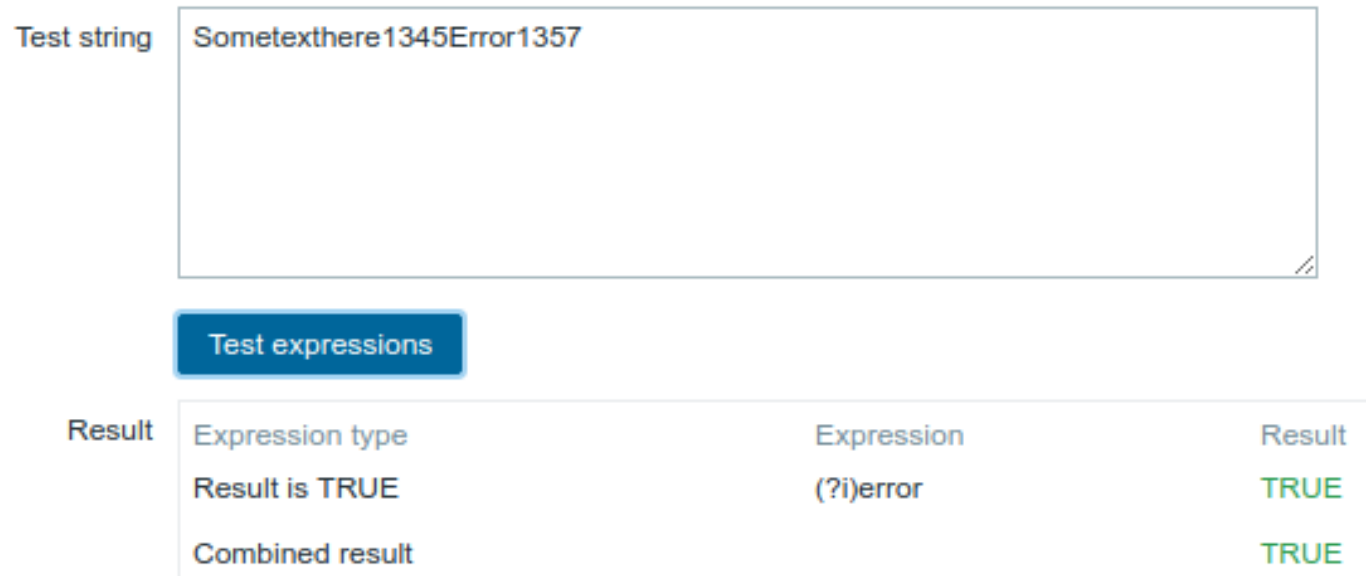


選択された *Expression type* : ” 結果は FALSE ” です。”*TESTDATABASE*” を含んだ名前と一致しません。

インライン型正規表現修飾子を含む例

インラインモディファイアを含む次の正規表現を使用します。
 (?i) を使用して、”error” という文字にマッチさせます。

`(?i)error`



選択された *Expression type* : ” 結果は TRUE ” です。文字”error ” にマッチする。

インライン正規表現修飾子を含む別の例

複数のインライン修飾子を含む次の正規表現を使用すると、特定の行以降の文字にマッチします。

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow .match newline charact
```



選択された Expression type : ” 結果は TRUE ” です。
 特定の行以降の文字にマッチします。

Attention:
g 修飾子は、行内で指定することはできません。利用可能な修飾子の一覧は
 利用可能な修飾子のリストは、pcresyntax manpageにあります。
 PCRE の構文に関するより詳しい情報は PCRE HTML documentation を参照してください。

ロケーションごとの正規表現サポート

ロケーション		正規表現	グローバル正規表現	複数行マッチング	コメント
エージェントアイテム					
	eventlog[]	Yes	Yes	Yes	regexp, severity, so
	eventlog.count[]				regexp, severity, so
	log[]				regexp パラメータ
	log.count[]				
	logrt[]		Yes/No		regexp パラメータは両
	logrt.count[]				
	proc.cpu.util[]		No	No	cmdline パラメータ
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				Linux 2.4 の device お
	system.hw.macaddr[]				interface パラメータ
	system.sw.packages[]				regexp パラメータ

ロケーション		正規表現	グローバル正規表現	複数行マッチング	コメント
	system.sw.packages.get[]				regexp パラメータ
	vfs.dir.count[]				regex_incl, regex_e
	vfs.dir.get[]				regex_incl, regex_e
	vfs.dir.size[]				regex_incl, regex_e
	vfs.file.regexp[]			Yes	regexp パラメータ
	vfs.file.regmatch[]				
	web.page.regexp[]				
SNMP トラップ					
	snmptrap[]	Yes	Yes	No	regexp パラメータ
アイテム値の事前処理		Yes	No	No	pattern パラメータ
トリガー/計算アイテムの関数					
	count()	Yes	Yes	Yes	operator パラメータか
	countunique()	Yes	Yes		
	find()	Yes	Yes		
	logeventid()	Yes	Yes	No	pattern パラメータ
	logsource()				
ローレベルディスカバリ					
	フィルター	Yes	Yes	No	正規表現フィールド
	オーバーライド	Yes	No		操作条件の 一致する、一
アクション条件		Yes	No	No	一致する、一致しないオ
スクリプト		Yes	Yes	No	入力検証ルールフィール
Web 監視		Yes	No	Yes	**regex:** プレフィックス 必須文字列フィールド
ユーザーマクロのコンテキスト		Yes	No	No	**regex:** プレフィックス
マクロ関数					
	regsub() iregsub()	Yes	No	No	pattern パラメータ
マップのリンクインジケータ		Yes	No	No	(テキストアイテム用の
アイコンマッピング		Yes	Yes	No	式フィールド
値のマッピング		Yes	No	No	マッピングタイプが re

11 問題の認識

概要　Zabbix の障害イベントはユーザーが確認することができます。

ユーザーは障害イベントの通知を受けると、Zabbix フロントエンドに移動し、以下のいずれかの方法で障害の更新ポップアップウィンドウを開き、障害を確認することができます。確認する際に、障害に対するコメント (対応中など) を入力することができます。

これにより、他のシステムユーザーが同じ問題を発見した場合、すぐに、それが確認済かどうかや、これまでのコメントを見ることができます。

このように、複数のシステムユーザーとの問題解決のワークフローを連携して行うことができます。

確認状況は、アクション操作を定義する際にも使用されます。例えば、あるイベントがしばらく確認されなかった場合にのみ、上位のマネージャに通知を送信するように定義できます。

イベントを確認し、それについてコメントするには、ユーザーは対応するトリガーに対して最低限の読み取り権限を持っている必要があります。障害の深刻度を変更したり、障害をクローズしたりするには、ユーザーは対応するトリガーに対して読み取り/書き込みの権限を持っている必要があります。

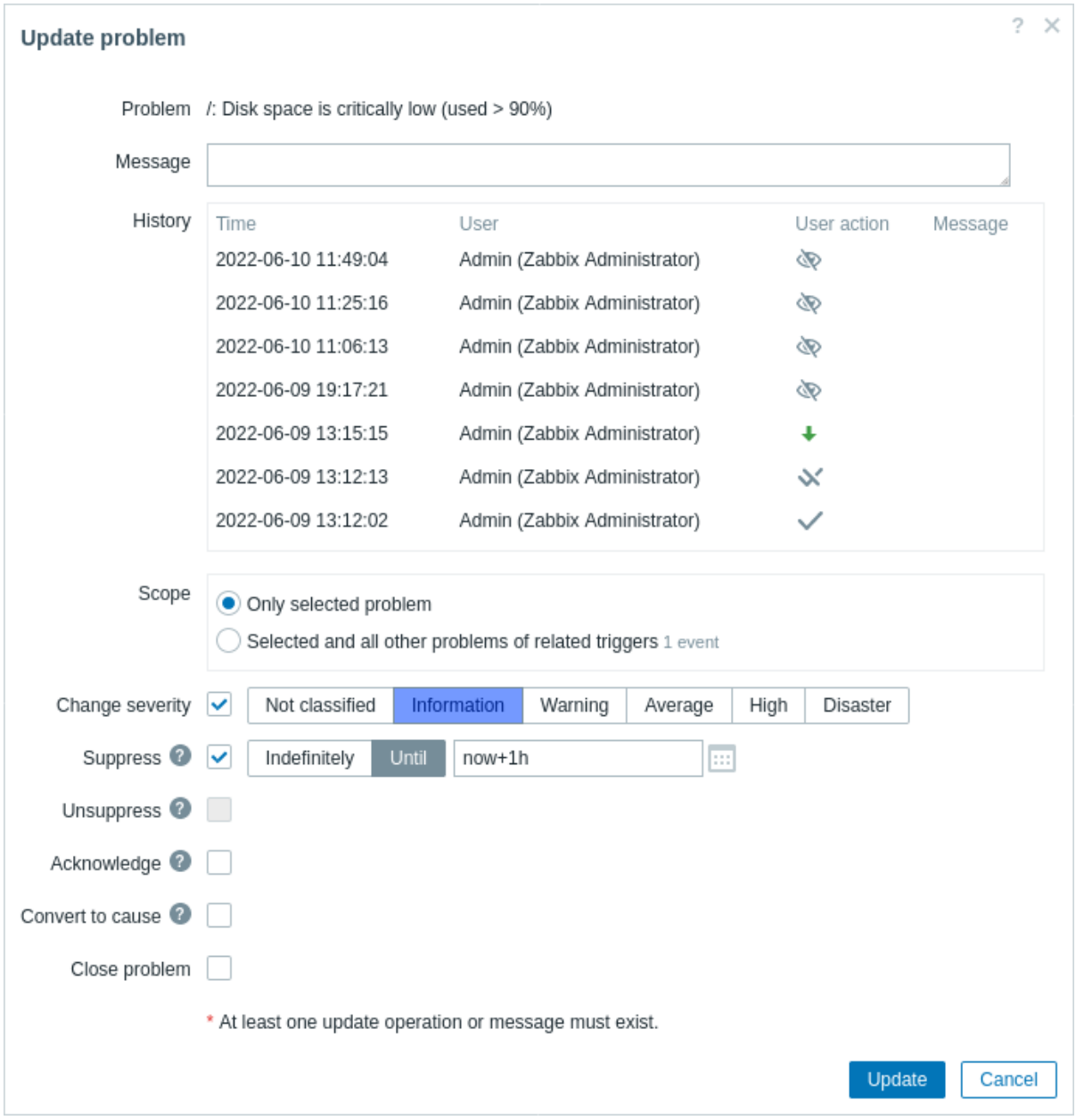
障害を確認するための、障害更新ポップアップウィンドウにアクセスするには、いくつかの方法があります。

- 監視 → 障害で障害を選択した状態で、リストの下の一括更新をクリックします。
- 障害の 更新列にある 更新をクリックします。
  - ダッシュボード (障害と 障害・深刻度ウィジェット)
  - 監視 → 障害
  - 監視 → 障害 → イベント詳細
- 未解決の障害のセルをクリックします。
  - ダッシュボード (トリガー概要ウィジェット)

ポップアップメニューの 更新オプションで、障害更新ウィンドウに遷移します。

障害の更新　障害更新ポップアップでは次のことが可能です。

- 障害に対するコメント
- これまでのコメントとアクションを表示
- 障害の深刻度を変更
- 障害の抑制状態の切り替え
- 障害の確認状態の切り替え
- 症状の障害を障害の原因に変える
- 障害を手動でクローズする



すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
障害	障害が 1 つだけ選択されている場合は障害名が表示されます。 複数の障害が選択されている場合は、*N* 個の障害が選択されていますと表示されます。
メッセージ	障害に対するコメントをテキストで入力してください（最大 2048 文字）。
履歴	障害に対するこれまでの活動やコメントが、時間やユーザーの詳細とともに一覧表示されます。 ユーザーの操作を表すアイコンの意味については、イベント詳細のページをご覧ください。 履歴は更新対象の障害が 1 つだけ選択されている場合に表示されます。

パラメータ	説明
スコープ	障害の深刻度の変更、確認、手動によるクローズなどのアクションの範囲を定義する。 選択した障害のみ - このイベントのみに影響します。 選択した障害と関連するトリガーのすべての障害 - 確認/クローズされた障害がある場合、このイベントと、これまでに確認/クローズされていない他のすべての障害に影響します。スコープにすでに確認またはクローズされた障害が含まれている場合、これらの障害が繰り返し確認/クローズされることはありません。一方、メッセージと深刻度の変更操作の数は制限されていません。
深刻度の変更	チェックボックスをマークし、深刻度ボタンをクリックすると、障害の深刻度が更新されます。 深刻度を変更するチェックボックスは、選択した障害のうち少なくとも 1 つに読み取り/書き込み権限がある場合に使用できます。更新をクリックすると、読み取り/書き込み可能な障害のみが更新されます。 選択したトリガーのいずれにも読み取り/書き込み権限が存在しない場合、チェックボックスは無効になります。
抑制	障害を抑制するにはチェックボックスをオンにします。 **Indefinitely** - 無期限に抑制する **Until** - 一定時間抑制する。時間形式は絶対と相対の両方がサポートされています。例: `now+1d` - 今から 1 日間 (デフォルト) `now/w` - 今週末まで `2022-05-28 12:00:00` - 絶対日時まで 単純な期間 (例: '1d'、'1w') はサポートされていないことに注意してください。 このオプションが使用できるかどうかは、" 問題の抑制" のユーザーの役割設定によって異なります。 参照: 障害の抑制
抑制の解除	障害の抑制を解除するには、チェックボックスをオンにします。このチェックボックスは、選択した障害の少なくとも 1 つが抑制されている場合にのみアクティブになります。 このオプションが使用できるかどうかは、" 問題の抑制" のユーザーの役割設定によって異なります。
確認	チェックボックスをマークして、障害を確認します。 このチェックボックスは、選択した障害の中に未確認のものが 1 つでもある場合に有効です。 すでに確認されている障害に対して、さらに確認を追加することはできません（ただし、別のコメントを追加することは可能です）。
未確認	チェックボックスをマークして、障害を未確認状態にします。 このチェックボックスは、選択された中に少なくとも 1 つの確認済の障害がある場合に利用できます。
原因に変更する	症状の障害を障害の原因に変換するには、チェックボックスをオンにします。
障害のクローズ	選択した障害を手動でクローズする場合は、チェックボックスをオンにします。 障害をクローズするためのチェックボックスは、選択した障害の少なくとも 1 つについて、トリガー設定で手動クローズを許可するオプションがチェックされている場合に利用可能です。更新をクリックすると、クローズが許可されている障害のみがクローズされます。 手動でクローズできる障害がない場合、チェックボックスは無効になります。 すでにクローズした障害を、繰り返しクローズすることはできません。

表示　障害の確認情報に基づいて、ダッシュボードまたはマップに障害数を表示できます。設定するには、マップ設定と深刻度別の障害ダッシュボードウィジェットの両方で使用できる障害表示オプションを選択する必要があります。すべての障害数や未確認の障害数を分けて表示することが可能です。

障害の更新情報（確認状況など）をもとに、メッセージの送信やリモートコマンドの実行といった更新操作を設定することができます。

1 障害の抑制

概要

問題の抑制は、後程対処できる問題を一時的に非表示にする方法を提供します。これは最も緊急の問題へ最優先順位を与えるために問題リストを整理するのに役立ちます。

たとえば、週末に問題が発生することがありますが、すぐに対処するほど緊急ではないため、月曜日の朝まで「スヌーズ」することができます。

問題の抑制では、メンテナンスホストのすべての問題が非表示になるホストメンテナンスによる問題の抑制とは対照的に、単一の問題のみを非表示にできます。

抑制された問題に対するトリガーアクションの操作は、ホストメンテナンス の場合と同じように一時停止されます。

構成

問題は、**problem update** ウィンドウから抑制できます。抑制は、コメント、重大度の変更、確認などとともに、問題の更新オプションの 1 つです。

同じ問題の更新ウィンドウから、問題の抑制を解除することもできます。

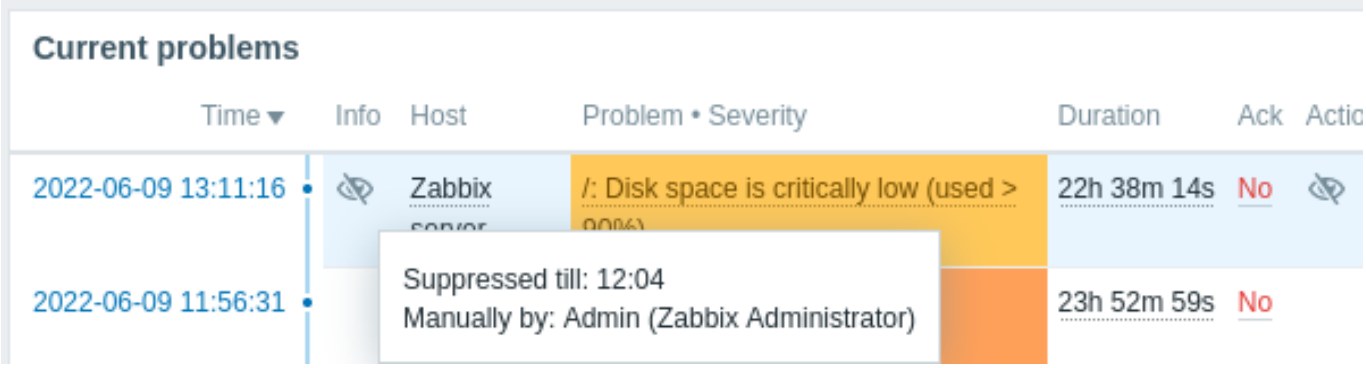
表示

抑制された問題は、情報列に点滅する抑制アイコンでマークされ、その後非表示になります。

抑制タスクが待機リストにある間、抑制アイコンは点滅します。タスクマネージャーが問題を抑制したら、アイコンの点滅は停止します。抑制アイコンが長時間点滅し続ける場合は、サーバーに問題がある可能性があります。例えば、サーバーがダウンしていてタスクマネージャーがタスクを完了できない場合などです。抑制解除についても同様です。タスクが送信されてからサーバーがタスクを完了していない間は、抑制解除アイコンが点滅します。

抑制された問題は、問題フィルター/ウィジェットの設定に応じて、非表示または表示されることがあります。

問題リストに表示される抑制された問題には抑制アイコンが表示され、マウスオーバーすると抑制の詳細が表示されます。



アクション列の抑制アイコンにマウスを合わせると、抑制の詳細がポップアップ表示されます。

12 設定のエクスポート/インポート

概要　Zabbix のエクスポート/インポート機能により、Zabbix システム間で様々な設定情報を共有することができます。

この機能のよくある使用例:

- テンプレートやネットワークマップの共有 - Zabbix ユーザ同士での設定パラメータの共有
- Zabbix コミュニティテンプレートにテンプレートをアップロードする - 他のユーザはテンプレートをダウンロードし、そのファイルを Zabbix にインポートできます。
- サードパーティツールとの統合 - 汎用的な YAML、XML、JSON 形式を使用することで、データのインポート/エクスポートを行ってサードパーティのツールやアプリケーションと統合することができます。

エクスポート/インポートできるもの

エクスポート/インポートできるオブジェクトは次の通りです。

- ホストグループ (*Zabbix API* 経由のみ)
- テンプレートグループ (*Zabbix API* 経由のみ)
- テンプレート
- ホスト
- ネットワークマップ
- メディアタイプ
- イメージ

エクスポート形式

Zabbix の Web インターフェースまたはZabbix APIを使用してデータをエクスポートすることができます。サポートされているエクスポート形式は YAML、XML、JSON です。

エクスポートの詳細

- サポートされているすべての要素が 1 つのファイルにエクスポートされます。
- リンクされたテンプレートから継承されたホストおよびテンプレートのエンティティ（アイテム、トリガー、グラフ、ディスカバリールール）はエクスポートされません。ホストレベルでそれらのエンティティに加えられた変更（アイテムの間隔の変更、正規表現の修正、低レベルディスカバリールールへのプロトタイプの追加など）は、エクスポート時に失われます。インポート時には、リンクされたテンプレートからすべてのエンティティが元のリンクされたテンプレートと同様に再作成されます。
- 低レベルディスカバリによって作成されたエンティティおよびそれらに依存するエンティティはエクスポートされません。たとえば、LLD ルールで生成されたアイテムに対して作成されたトリガーはエクスポートされません。
- エクスポートされたホスト/テンプレートにタイムアウトをサポートするエンティティが含まれている場合、これらのエンティティに独自のタイムアウトが設定されていれば、タイムアウト値もエクスポートされます。

インポートの詳細

- インポートは最初のエラーで停止します。
- イメージのインポート時に既存のイメージを更新する場合、”imagetype” フィールドは無視されます。つまり、インポートによってイメージタイプを変更することはできません。
- ”Delete missing” オプションを使用してホスト/テンプレートをインポートする場合、インポートファイルに存在しないホスト/テンプレートマクロは、インポート後にホスト/テンプレートから削除されます。
- アイテム、トリガー、グラフ、discoveryRules、itemPrototypes、triggerPrototypes、graphPrototypes の空のタグは意味がありません。つまり、タグがない場合と同じです。
- インポートされたホスト/テンプレートのエンティティに独自のタイムアウトが設定されている場合はそれが適用され、設定されていない場合はプロキシ/グローバルのタイムアウトが適用されます。
- インポートは YAML、XML、JSON をサポートしており、インポートファイルには正しい拡張子が必要です。YAML は.yaml または.yml、XML は.xml、JSON は.json です。サポートされている XML バージョンについては互換性情報を参照してください。
- インポートは UTF-8 エンコーディング（BOMあり/なし）でのみ構成ファイルをサポートします。他のエンコーディング（UTF16LE、UTF16BE、UTF32LE、UTF32BE など）は、インポート変換エラーとなります。

YAML ベースフォーマット　YAML エクスポートフォーマットには以下のノードが含まれます:

- Zabbix YAML エクスポートのルートノード
- エクスポートバージョン

```
zabbix_export:
  version: '8.0'
```

その他のノードはエクスポートするオブジェクトに依存します。

XML 形式　XML エクスポート形式には以下のタグが含まれます。

- XML ドキュメントのデフォルトヘッダー
- Zabbix XML エクスポートのルートタグ
- エクスポートバージョン

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

その他のタグはエクスポートされるオブジェクトによって異なります。

JSON 形式　JSON エクスポート形式には以下のオブジェクトが含まれます。

- Zabbix JSON エクスポートのルートオブジェクト
- エクスポートバージョン

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

他のオブジェクトはエクスポートされたオブジェクトに依存します。

1 テンプレートグループ

概要

フロントエンドでは、テンプレートグループはテンプレートエクスポートでのみエクスポートできます。テンプレートをエクスポートすると、そのテンプレートが属するすべてのグループも自動的にエクスポートされます。

API を使用すると、テンプレートグループをテンプレートから独立してエクスポートできます。

エクスポートフォーマット

```
template_groups:
  - uuid: 36bff6c29af64692839d077febfc7079
    name: 'Network devices'
```

エクスポートされた要素

要素	タイプ	説明
uuid	string	テンプレートグループのユニーク ID。
name	string	テンプレートグループの名前。

2 ホストグループ

概要

Web インターフェースでは、ホストグループはホストのエクスポートでのみエクスポートできます。ホストがエクスポートされると、それが属するすべてのグループも自動的にエクスポートされます。

API を使用すると、ホストから独立してホストグループをエクスポートできます。

エクスポート形式

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix servers'
```

エクスポートされる要素

要素	タイプ	説明
uuid	文字列	ホストグループのユニーク ID。
name	文字列	グループ名。

3 テンプレート

概要

テンプレートは、多くの関連オブジェクトやオブジェクトの関係とともにエクスポートされます。

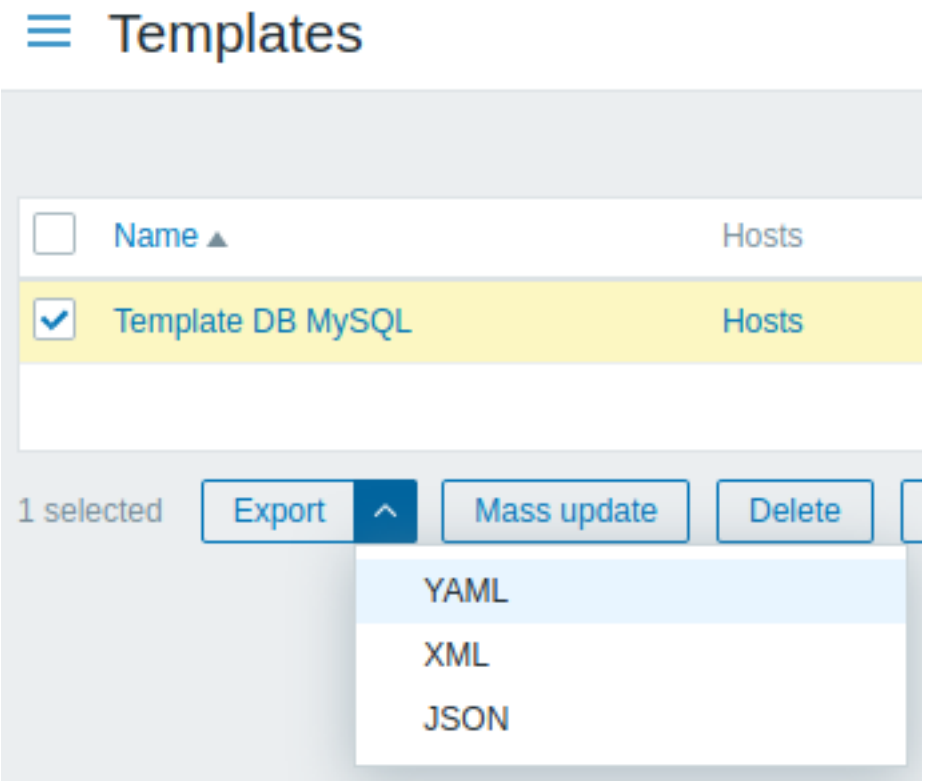
テンプレートのエクスポートには以下が含まれます:

- リンクされたテンプレートグループ
- (ホストプロトタイプの設定で使用されている場合) リンクされたホストグループ
- テンプレートデータ
- 他のテンプレートへのリンク
- テンプレートグループへのリンク
- 直接リンクされたアイテム
- 直接リンクされたトリガー
- 直接リンクされたグラフ
- 直接リンクされたダッシュボード
- すべてのプロトタイプを含む直接リンクされたディスカバリルール
- 直接リンクされた Web シナリオ
- 値のマッピング

エクスポート

テンプレートをエクスポートするには、以下の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. エクスポートするテンプレートのチェックボックスをオンにします。
3. リストの下にある エクスポートをクリックします。



選択したフォーマットに応じて、テンプレートはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_templates.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）;
- `zabbix_export_templates.xml` - XML エクスポートの場合;
- `zabbix_export_templates.json` - JSON エクスポートの場合。

インポート

テンプレートをインポートするには、以下の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. 右上の インポートをクリックします。
3. インポートファイルを選択します。
4. 設定フォームの右下にある インポートをクリックします。

詳細オプションチェックボックスをオンにすると、インポート可能なすべての要素の詳細リストが表示されます。必要に応じて各インポートルールをオンまたはオフにします。

すべて行のチェックボックスをクリックすると、その下のすべての要素がオンまたはオフになります。

インポートルール:

ルール	説明
既存のものを更新	既存の要素はインポートファイルのデータで更新されます。そうでない場合は更新されません。
新規作成	新しい要素はインポートファイルのデータで作成されます。そうでない場合は作成されません。
存在しないものを削除	インポートファイルに存在しない既存の要素は削除されます。そうでない場合は削除されません。 テンプレートリンクで 存在しないものを削除をオンにすると、インポートファイルに存在しない現在のテンプレートリンクが解除されます。リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は（各エンティティについて 存在しないものを削除オプションが選択されていない限り）削除されません。

次の画面では、インポートされるテンプレートの内容を確認できます。新しいテンプレートの場合、すべての要素が緑色で表示されます。既存のテンプレートを更新する場合、新しいテンプレート要素は緑色で強調表示され、削除されたテンプレート要素は赤色で強調表示され、変更されていない要素はグレーの背景で表示されます。

左側のメニューを使用して、変更リストをナビゲートできます。更新済みセクションは、既存のテンプレート要素に加えられたすべての変更を強調表示します。追加済みセクションには、新しいテンプレート要素が一覧表示されます。各セクションの要素は要素タイプごとにグループ化されており、グレーの矢印をクリックすると要素グループを展開または折りたたむことができます。

テンプレートの変更内容を確認し、インポートをクリックしてテンプレートのインポートを実行します。インポートの成功または失敗のメッセージがフロントエンドに表示されます。

エクスポート形式　テンプレートをエクスポートすると、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。エクスポートには、テンプレートのメタデータ、アイテム、マクロ、トリガー、ダッシュボードなどのテンプレート要素が含まれます。

各要素は特定の目的を持ち、ネストされた要素を含む場合があります。

以下のセクションでは、エクスポート形式の各要素について説明します。例では、Linux by Zabbix agent テンプレートを使用しています。

省略記号（...）は、簡潔にするために省略された要素を示します。（see table below）の注記は、後のセクションで要素がより詳細に説明されている場合に使用されます。

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

テンプレートメタデータ

要素		型	説明
version		string	(必須) エクスポートバージョン。
template_groups			(必須) テンプレートグループのルート要素。
	uuid	string	(必須) このテンプレートグループの一意の識別子。
	name	string	(必須) テンプレートグループ名。
host_groups			ホストプロトタイプで使用されるホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
templates			テンプレートのルート要素。
triggers			テンプレートトリガー要素のルート要素。テンプレートアイテムトリガーと同じです。
graphs			テンプレートグラフ要素のルート要素。ホストグラフと同じです。

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        これは公式の Linux テンプレートです。Zabbix agent 8.0 以降が必要です。(...)
      wizard_ready: 'YES'
      readme: |
        ## 概要

        これは公式の Linux テンプレートです。Zabbix agent 8.0 以降が必要です。(...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (下記の表を参照)
      discovery_rules: (下記の表を参照)
      tags:
```

```
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'エージェントが利用できないと見なされるまでのタイムアウト。Zabbix サーバー/プロキシから到達可能なエ
          config: (下記の表を参照)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'CPU 使用率のクリティカルしきい値 (% で指定)。'
          config: (下記の表を参照)
        (...)
      dashboards: (下記の表を参照)
      valuemaps: (下記の表を参照)
    (...)
```

テンプレート

要素	型	説明
uuid	string	(必須) このテンプレートの一意の識別子。
template	string	(必須) 一意のテンプレート名。
name	string	表示用テンプレート名。
description	text	テンプレートの説明。
wizard_ready	text	ホストウィザードでテンプレートを選択可能にするかどうか。 指定可能な値:[1] NO (0, デフォルト), YES (1)。 参考: テンプレートオブジェクト (wizard_ready)。
readme	text	ホストウィザードで表示するテンプレート固有の設定手順。Markdown 形式をサポート。
vendor		テンプレートベンダーのルート要素 (エクスポートされたテンプレートにベンダーデータが含まれている場合に存在)。
name	string	(必須) テンプレートベンダー名。
version	string	(必須) テンプレートバージョン。 標準テンプレートの場合、バージョンは Zabbix のメジャーバージョン、区切り文字 (”-”)、リビジョン番号 (テンプレートの新バージョンごとに増加し、Zabbix のメジャーバージョンごとにリセット) で表示されます。例: 7.0-0, 7.0-3, 8.0-0, 8.0-3。
templates		リンクされたテンプレートのルート要素。
name	string	(必須) テンプレート名。
groups		テンプレートグループのルート要素。
name	string	(必須) テンプレートグループ名。
items		テンプレートアイテムのルート要素。
discovery_rules		テンプレートのローレベルディスカバリールールのルート要素。
httptests		テンプレート Web シナリオのルート要素。
tags		テンプレートタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
macros		テンプレートユーザーマクロのルート要素。
macro	string	(必須) ユーザーマクロ名。
type	string	ユーザーマクロの型。 指定可能な値:[1] TEXT (0, デフォルト), SECRET_TEXT (1), VAULT (2)。 参考: ユーザーマクロオブジェクト (type)。
value	string	ユーザーマクロ値。
description	string	ユーザーマクロの説明。
config		テンプレートマクロ設定のルート要素。ホストウィザードでのマクロの表示方法を制御。
dashboards		テンプレートダッシュボードのルート要素。
valuemaps		テンプレート値のマッピングのルート要素。

テンプレートアイテム

```
zabbix_export:
  (...)
```

```
templates:
  (...)
    items:
      - uuid: f94f9f4699e94c369e6c98b2a2f485ce
        name: 'Zabbix agent ping'
        key: agent.ping
        description: 'このアイテムに対してエージェントは常に"1" を返します。`nodata()`と組み合わせて可用性チェックに使
        valuemap:
          name: 'Zabbix agent ping status'
        tags:
          - tag: component
            value: system
      (...)
      - uuid: 58818005e76d46dda14d6592f601ab00
        name: 'Number of installed packages'
        key: system.sw.packages.get
        delay: 1h
        preprocessing: （下記の表を参照）
        tags:
          - tag: component
            value: os
        triggers: （下記の表を参照）
      - uuid: 403cebed115441369e94d35d070ca7b8
        name: 'Memory utilization'
        type: DEPENDENT
        key: vm.memory.utilization
        value_type: FLOAT
        units: '%'
        description: '使用メモリの割合は`100-pavailable`として計算されます。'
        preprocessing: （下記の表を参照）
        master_item:
          key: 'vm.memory.size[pavailable]'
        tags:
          - tag: component
            value: memory
        triggers: （下記の表を参照）
      (...)
```

要素	型	説明
uuid	string	(必須) このアイテムの一意の識別子。
name	string	(必須) アイテム名。
type	string	アイテムタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22)。
snmp_oid	string	(SNMP_AGENT アイテムの場合は必須) SNMP オブジェクト ID。
key	string	(必須) アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムの場合は常に 0 になります。
history	string	履歴データを保存する期間 (時間のサフィックス、ユーザーマクロ、またはLLD マクロを使用可能)。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間 (時間のサフィックス、ユーザーマクロ、またはLLD マクロを使用可能)。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
value_type	string	受信値の型。 指定可能な値:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, デフォルト), TEXT (4), BINARY (5)。

要素	型	説明
allowed_hosts	string	アイテムにデータを送信できるホストの IP アドレス (カンマ区切り) のリスト。 TRAP および HTTP_AGENT アイテムでサポート。
units	string	受信値の単位 (bps, B など)。
params	text	アイテムのタイプに応じた追加パラメータ (SSH および TELNET アイテムの場合は実行スクリプト、ODBC アイテムの場合は SQL クエリ、CALCULATED アイテムの場合は式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムの場合はスクリプト)。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポート。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポート。 SSH アイテムの場合の指定可能な値:[1] PASSWORD (0, デフォルト), PUBLIC_KEY (1)。 HTTP_AGENT アイテムの場合の指定可能な値:[1] NONE (0, デフォルト), BASIC (1), NTLM (2)。
username	string	(SSH および TELNET アイテムの場合は必須) 認証用ユーザー名。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポート。 JMX アイテムで使用する場合は、`password`(下記参照) も指定するか、両方の要素を空白にしてください。
password	string	(SSH および TELNET アイテムの場合は必須) 認証用パスワード。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポート。 JMX アイテムで使用する場合は、`username`(上記参照) も指定するか、両方の要素を空白にしてください。
publickey	string	(SSH アイテムの場合は必須) 公開鍵ファイル名。
privatekey	string	(SSH アイテムの場合は必須) 秘密鍵ファイル名。
description	text	アイテムの説明。
inventory_link	string	アイテムによって入力されるホストインベントリフィールド。 指定可能な値:[1] NONE (0), ALIAS (4) など (ホストインベントリでサポートされているフィールドを参照)。
valuemap		アイテム値マップのルート要素。
name	string	(必須) アイテムで使用する値マップ名。
logtimefmt	string	ログエントリ内の時刻のフォーマット。 LOG 値タイプのアイテムでサポート。
preprocessing		アイテム値の事前処理のルート要素。
step		テンプレートアイテム値の事前処理ステップのルート要素。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポート。
master_item		(DEPENDENT アイテムの場合は必須) 従属アイテムのマスターアイテムのルート要素。
key	string	(必須) 従属アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 タイムアウトリストのアイテムタイプでサポート。
url	string	(HTTP_AGENT アイテムの場合は必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポート。
name	string	(HTTP_AGENT アイテムの場合は必須) クエリパラメータ名。
value	string	クエリパラメータ値。 HTTP_AGENT アイテムでサポート。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポート。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムの場合は必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータ値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポート。
posts	string	HTTP(S) リクエストボディデータ。 HTTP_AGENT アイテムでサポート。
status_codes	string	必要な HTTP ステータスコードの範囲 (カンマ区切り)。 HTTP_AGENT アイテムでサポート。
follow_redirects	string	データポーリング時にレスポンスリダイレクトを追従するかどうか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
post_type	string	ポストデータボディのタイプ。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] RAW (0, デフォルト), JSON (2), XML (3)。

要素	型	説明
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポート。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポート。
name	string	(HTTP_AGENT アイテムの場合は必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムの場合は必須) ヘッダー値。
retrieve_mode	string	レスポンスのどの部分を保存するか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
request_method	string	リクエストメソッドタイプ。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] GET (0, デフォルト), POST (1), PUT (2), HEAD (3)。
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] RAW (0, デフォルト), JSON (1)。
allow_traps	string	trapper アイテムと同様に値を入力できるようにするかどうか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
ssl_cert_file	string	公開 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポート。
ssl_key_file	string	秘密 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポート。
ssl_key_password	string	SSL キーのパスワード。 HTTP_AGENT アイテムでサポート。
verify_peer	string	ホストの証明書が正当かどうかを検証するかどうか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
verify_host	string	接続先のホスト名が証明書のものと一致するかどうかを検証するかどうか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
tags		アイテムタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
triggers		テンプレートアイテムトリガーのルート要素。

Note:
関連項目: アイテムオブジェクト (同名のプロパティを参照)。

テンプレートアイテムの値の事前処理ステップ

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

要素	型	説明
type	string	(必須) アイテム値の事前処理ステップのタイプ。 指定可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, 計算式: 受信値 - 前回値), CHANGE_PER_SECOND (10, 計算式: (受信値 - 前回値)/(現在時刻 - 前回チェック時刻)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30)。
parameters		(必須) アイテム値の事前処理ステップのパラメータのルート要素。
parameter	string	アイテム値の事前処理ステップの個別パラメータ。
error_handler	string	事前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値:[1] ORIGINAL_ERROR (0, デフォルト), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3)。
error_handler_params	string	エラーハンドラのパラメータ。

Note:
参照: アイテムの事前処理オブジェクト (同名のプロパティを参照)。

テンプレートアイテムのトリガー

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Number of installed packages has been changed'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
                  value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: High memory utilization'
              event_name: 'Linux: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)'
              priority: AVERAGE
              description: 'The system is running out of free memory.'
              dependencies:
                - name: 'Linux: Lack of available memory'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

要素	型	説明
uuid	string	(必須) このトリガーの一意な識別子。
expression	string	(必須) トリガーの式。
recovery_mode	string	OK イベント生成の基準。 可能な値:[1] EXPRESSION (0, デフォルト), RECOVERY_EXPRESSION (1), NONE (2)。
recovery_expression	string	トリガーのリカバリー式。
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関）。 可能な値:[1] DISABLED (0, デフォルト), TAG_VALUE (1)。
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(必須) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。
status	string	トリガーのステータス。 可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
priority	string	トリガーの深刻度。 可能な値:[1] NOT_CLASSIFIED (0, デフォルト), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5)。
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の問題イベントまたは複数の問題イベント）。 可能な値:[1] SINGLE (0, デフォルト), MULTIPLE (1)。
manual_close	string	問題イベントの手動クローズ。 可能な値:[1] NO (0, デフォルト), YES (1)。
dependencies		依存関係のルート要素。
name	string	(必須) 依存トリガー名。
expression	string	(必須) 依存トリガーの式。
recovery_expression	string	依存トリガーのリカバリー式。
tags		トリガータグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
参照: Trigger object（同名のプロパティを参照）。

テンプレートのローレベルディスカバリルール

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
テンプレートのローレベルディスカバリルール要素のほとんどは、テンプレートアイテムと同じです。下記の表では、テンプレートアイテムと異なる要素について説明します。

要素	型	説明
type	string	アイテムタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22)。
key	string	(必須) ローレベルディスカバリルールのキー。
filter		テンプレートのローレベルディスカバリルールフィルタのルート要素。
lifetime	string	発見されなくなったリソースが削除されるまでの期間（秒、時間サフィックス、またはユーザーマクロを使用)。 デフォルト: 7d。
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 指定可能な値: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER。
enabled_lifetime	string	発見されなくなったリソースが無効化されるまでの期間（秒、時間サフィックス、またはユーザーマクロを使用)。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 指定可能な値: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER。
item_prototypes		テンプレートアイテムプロトタイプ要素のルート要素。テンプレートアイテムと同じです。
trigger_prototypes		テンプレートトリガープロトタイプ要素のルート要素。テンプレートアイテムトリガーと同じです。
graph_prototypes		テンプレートグラフプロトタイプ要素のルート要素。ホストグラフと同じです。
host_prototypes		テンプレートホストプロトタイプ要素のルート要素。ホストと同じです。
parent_discovery_rule		ローレベルディスカバリルールプロトタイプの親ローレベルディスカバリルール（またはルールプロトタイプ）のルート要素。このプロパティは、参照されたルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	(必須) 親ローレベルディスカバリルール（またはルールプロトタイプ）のキー。
master_item	string	(DEPENDENT ルールの場合は必須) 従属ルールのマスターアイテムのルート要素。
lld_macro_paths		ローレベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	(必須) ローレベルディスカバリルールのマクロ名。
path	string	(必須) 対応するマクロに割り当てる値のセレクタ。
preprocessing		ローレベルディスカバリルール値のプリプロセスのルート要素。
step		テンプレートアイテム値のプリプロセスステップと同じローレベルディスカバリルール値のプリプロセスステップ要素のルート要素。ただし、指定可能な値は少なくなります。参照: LLD ルールプリプロセスオブジェクト。
overrides		ローレベルディスカバリルールのオーバーライドルールのルート要素。
name	string	(必須) 一意のオーバーライド名。
step	string	(必須) オーバーライドの一意の順序番号。
stop	string	一致した場合、次のオーバーライドの処理を停止します。
filter		テンプレートのローレベルディスカバリルールフィルタと同じテンプレートのローレベルディスカバリルールオーバーライドルールフィルタ要素のルート要素。
operations		テンプレートのローレベルディスカバリルールオーバーライド操作のルート要素。

Note:
参照: LLD ルールオブジェクト（同名の該当プロパティを参照)。

テンプレートのローレベルディスカバリールールフィルター

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
```

```
            value: '{$NET.IF.IFNAME.NOT_MATCHES}'
            operator: NOT_MATCHES_REGEX
      (...)
    - uuid: 2bbdc79f082d4c618e01bec625e9c90a
      (...)
      filter:
        evaltype: AND
        conditions:
          - macro: '{#DEVNAME}'
            value: '{$VFS.DEV.DEVNAME.MATCHES}'
          - macro: '{#DEVNAME}'
            value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
            operator: NOT_MATCHES_REGEX
          - macro: '{#DEVTYPE}'
            value: disk
      (...)
```

要素	型	説明
evaltype	string	フィルター条件の評価方法を上書きします。 使用可能な値:[1] AND_OR (0, デフォルト), AND (1), OR (2), FORMULA (3)。
formula	string	フィルター条件のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	string	(必須) チェックを実行するローレベルディスカバリーマクロの名前。
value	string	比較する値。
operator	string	条件演算子。 使用可能な値:[1] MATCHES_REGEX (8, デフォルト), NOT_MATCHES_REGEX (9)。
formulaid	string	(必須) カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみ使用可能。フィルター条件を変更する際にユーザーが定義する必要がありますが、その後リクエストする際には新たに生成されます。

Note:
参照: LLD ルールフィルターオブジェクト（該当するプロパティ名で参照）。

テンプレートのローレベルディスカバリールールのオーバーライド操作

要素	型	説明
operationobject	string	操作が適用されるオブジェクト。 可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3)。
operator	string	オーバーライド条件演算子。 可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6)。
value	string	オーバーライド条件演算子のための正規表現または文字列。
status	string	オーバーライド操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトが追加されるかどうか。
delay	string	オーバーライド操作時にアイテムプロトタイプに設定される更新間隔。
history	string	オーバーライド操作時にアイテムプロトタイプに設定される履歴保存期間。
trends	string	オーバーライド操作時にアイテムプロトタイプに設定されるトレンド保存期間。
severity	string	オーバーライド操作時に設定されるトリガープロトタイプの深刻度。
tags		オーバーライド操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
templates		オーバーライド操作時にホストプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	オーバーライド操作時に設定されるホストプロトタイプのインベントリモード。

Note:
関連項目: LLD ルールのオーバーライド操作オブジェクト（同名のプロパティを参照）。

テンプレート Web シナリオ

要素	型	説明
uuid	string	(必須) この Web シナリオの一意の識別子。
name	string	(必須) Web シナリオ名。
delay	string	Web シナリオを実行する頻度 (秒、時間サフィックス、またはユーザーマクロを使用)。 デフォルト: 1m。
attempts	integer	Web シナリオステップの実行試行回数。 可能な値:[1] 1-10 (デフォルト: 1)。
agent	string	クライアントエージェント。Zabbix は選択されたブラウザを装います。 これは、Web サイトが異なるブラウザに対して異なるコンテンツを返す場合に便利です。 デフォルト: Zabbix。
http_proxy	string	Web シナリオで使用されるプロキシ。 `http://[username[:password]@]proxy.example.com[:port]` の形式で指定。
variables		シナリオステップで使用できる Web シナリオ変数のルート要素。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。 ヘッダーは HTTP プロトコルで表示されるのと同じ構文でリストする必要があります。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
status	string	Web シナリオのステータス。 可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
authentication	string	認証方式。 可能な値:[1] NONE (0, デフォルト), BASIC (1), NTLM (2)。
http_user	string	BASIC(HTTP) または NTLM 認証で使用されるユーザー名。
http_password	string	BASIC(HTTP) または NTLM 認証で使用されるパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証するかどうか。 可能な値:[1] NO (0, デフォルト), YES (1)。
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致するかどうかを検証します。 可能な値:[1] NO (0, デフォルト), YES (1)。
ssl_cert_file	string	クライアント認証で使用される SSL 証明書ファイル名 (PEM 形式である必要があります)。
ssl_key_file	string	クライアント認証で使用される SSL 秘密鍵ファイル名 (PEM 形式である必要があります)。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		(必須) テンプレート Web シナリオステップのルート要素。
tags		Web シナリオタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
参照: Web シナリオオブジェクト (同名の関連プロパティを参照)。

テンプレート Web シナリオステップ

要素	型	説明
name	string	(必須) Web シナリオステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL に追加される HTTP フィールドの配列)。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（生の POST データ）または HTTP フィールドの配列（フォームフィールドデータ))。
name	string	(必須) POST フィールド名。

要素	型	説明
value	string	(必須) POST フィールド値。
variables		このステップの後に適用されるべきステップレベルの変数（マクロ）のルート要素。 変数値に'regex:' プレフィックスがある場合、その値はこのステップで返されたデータから'regex:' プレフィックスに続く正規表現パターンに従って抽出されます
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトの追従。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、時間のサフィックス、またはユーザーマクロを使用）。 デフォルト: 15s。
required	string	レスポンスに含まれている必要があるテキスト（空の場合は無視）。
status_codes	string	許可される HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。空の場合は無視）。

Note:
参照: Web シナリオステップオブジェクト（同名の関連プロパティを参照）。

テンプレートマクロの設定

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Seconds since the last Zabbix agent seen'
            description: 'Timeout after which agent is considered unavailable.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'Threshold of CPU utilization expressed'
            description: 'Critical threshold of CPU utilization expressed in %. In the range from 0 to 100
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Interfaces control'
            description: 'Fire a trigger when the interface operational status changes to "Link down".'
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

要素	型	説明
type	string	(必須) マクロ入力フィールドのタイプ。 指定可能な値:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3)。
priority	string	マクロリスト内でのマクロの位置。
section_name	string	マクロがグループ化される折りたたみセクションのラベル。
label	string	(TEXT、LIST、CHECKBOX マクロで必須) マクロのラベル。
description	text	マクロのヘルプテキスト。Markdown 書式をサポート。
required	string	マクロを必須としてマークします。 指定可能な値:[1] NO (0), YES (1)。 TEXT および LIST マクロでサポート。
regex	string	テキストボックスフィールドでのユーザー入力を検証するための正規表現。 TEXT マクロでサポート。
options		LIST アイテムまたは CHECKBOX 値のルート要素。
value	string	(LIST マクロで必須) LIST アイテムの値。
text	string	(LIST マクロで必須) LIST アイテムのテキスト。
checked	string	(CHECKBOX マクロで必須) チェック状態を表す値。 指定可能な値:[1] 0 (false), 1 (true)。 CHECKBOX マクロでサポート。
unchecked	string	(CHECKBOX マクロで必須) 未チェック状態を表す値。 指定可能な値:[1] 0 (false), 1 (true)。 CHECKBOX マクロでサポート。

Note:
参考: マクロ設定オブジェクト（同名のプロパティを参照）。

テンプレートダッシュボード

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: （下表参照）
        (...)
```

要素		型	説明
uuid		string	(必須) このダッシュボードの一意の識別子。
name		string	(必須) テンプレートダッシュボード名。
display period		integer	ダッシュボードページの表示期間。
auto_start		string	スライドショーの自動開始。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
pages			テンプレートダッシュボードページのルート要素。
	name	string	ページ名。
	display period	integer	ページの表示期間。
	sortorder	integer	ページの並び順。
widgets			テンプレートダッシュボードウィジェットのルート要素。

Note:
参照: テンプレートダッシュボードオブジェクト（同名のプロパティを参照）。

テンプレートダッシュボードウィジェット

```
zabbix_export:
  (...)
  templates:
```

```
  (...)
    dashboards:
      - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
        name: Filesystems
        pages:
          - name: Overview
            widgets:
              - type: graphprototype
                width: '72'
                height: '5'
                fields:
                  - type: INTEGER
                    name: columns
                    value: '3'
                  - type: GRAPH_PROTOTYPE
                    name: graphid.0
                    value:
                      host: 'Linux by Zabbix agent'
                      name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                  - type: STRING
                    name: reference
                    value: AAAEB
              - type: graphprototype
                'y': '5'
                width: '72'
                height: '5'
                fields:
                  - type: INTEGER
                    name: columns
                    value: '1'
                  - type: GRAPH_PROTOTYPE
                    name: graphid.0
                    value:
                      host: 'Linux by Zabbix agent'
                      name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                  - type: STRING
                    name: reference
                    value: AAAEC
      (...)
```

要素	型	説明
type	string	(必須) ウィジェットタイプ。
name	string	ウィジェット名。
x	integer	テンプレートダッシュボードの左端からの水平位置。 可能な値:[1] 0-71。
y	integer	テンプレートダッシュボードの上端からの垂直位置。 可能な値:[1] 0-63。
width	integer	ウィジェットの幅。 可能な値:[1] 1-72。
height	integer	ウィジェットの高さ。 可能な値:[1] 1-64。
hide_header	string	ウィジェットヘッダーの非表示。 可能な値:[1] NO (0, デフォルト), YES (1)。
fields		テンプレートダッシュボードウィジェットフィールドのルート要素。
type	string	(必須) ウィジェットフィールドタイプ。 可能な値:[1] INTEGER (0), STRING (1), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13)。
name	string	(必須) ウィジェットフィールド名。
value	mixed	(必須) フィールドタイプに応じたウィジェットフィールド値。

Note:
関連項目: テンプレートダッシュボードウィジェットオブジェクト（同名のプロパティを参照）。

テンプレートの値のマッピング

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
              newvalue: up
        (...)
```

要素	型	説明
uuid	文字列	(必須) この値のマッピングの一意の識別子。
name	文字列	(必須) 値のマッピング名。
mapping		マッピングのルート要素。
type	文字列	マッピングの一致タイプ。 可能な値:[1] EQUAL (0, デフォルト), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6)。
value	文字列	元の値。
newvalue	文字列	(必須) 元の値がマッピングされる値。

Note:
参照: 値のマッピングオブジェクト (同名の関連プロパティを参照)。

脚注 [1] 括弧内の API 整数値（例：ENABLED（0））は参照用にのみ記載されています。詳細については、表のエントリまたは各セクションの最後にあるリンクされた API オブジェクトページを参照してください。

4 ホスト

概要

ホストは、多くの関連オブジェクトおよびオブジェクトの関係とともにエクスポートされます。

ホストのエクスポートには以下が含まれます:

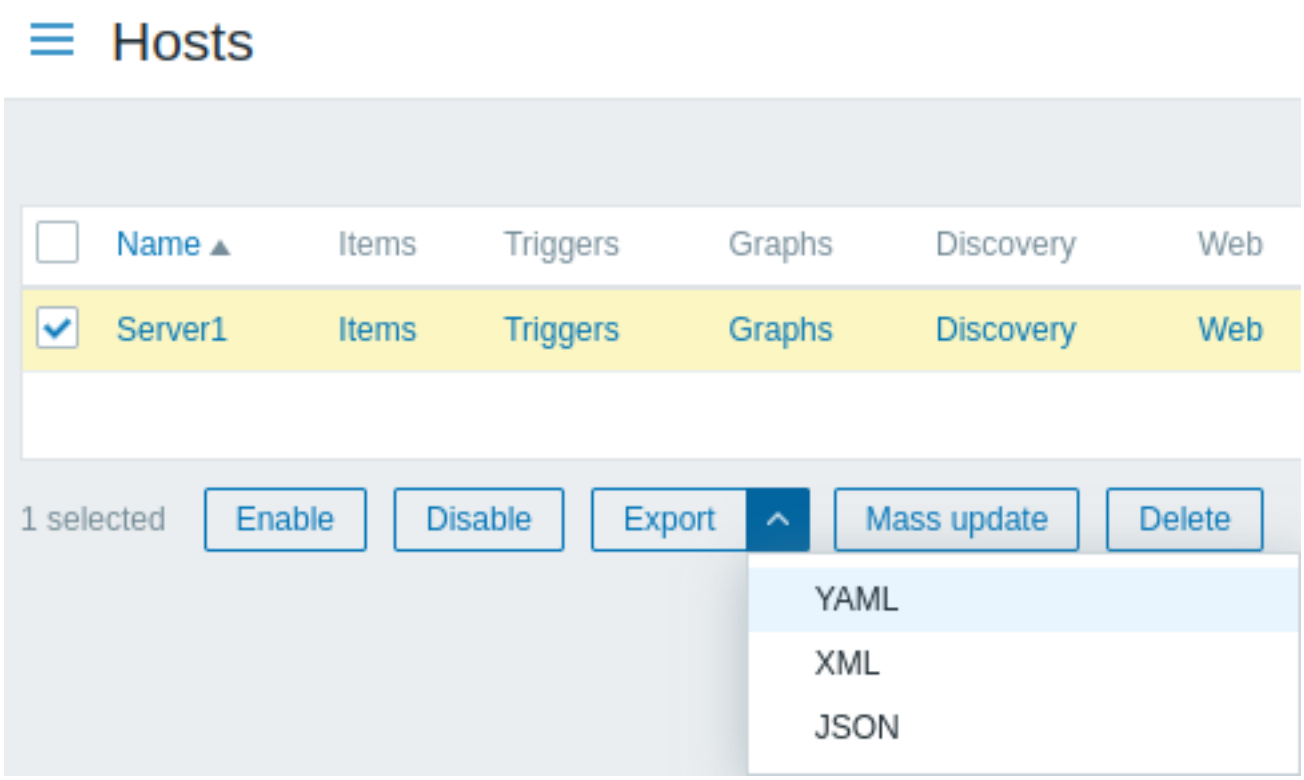
- リンクされたホストグループ
- ホストデータ
- テンプレートのリンク
- ホストグループのリンク
- ホストインターフェース
- 直接リンクされたアイテム

- 直接リンクされたトリガー
- すべてのプロトタイプを含む直接リンクされたディスカバリールール
- 直接リンクされた Web シナリオ
- ホストマクロ
- ホストインベントリデータ
- 値のマッピング
- リンクされたグラフ

エクスポート

ホストをエクスポートするには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. エクスポートするホストのチェックボックスをオンにします。
3. リストの下にある エクスポートをクリックします。



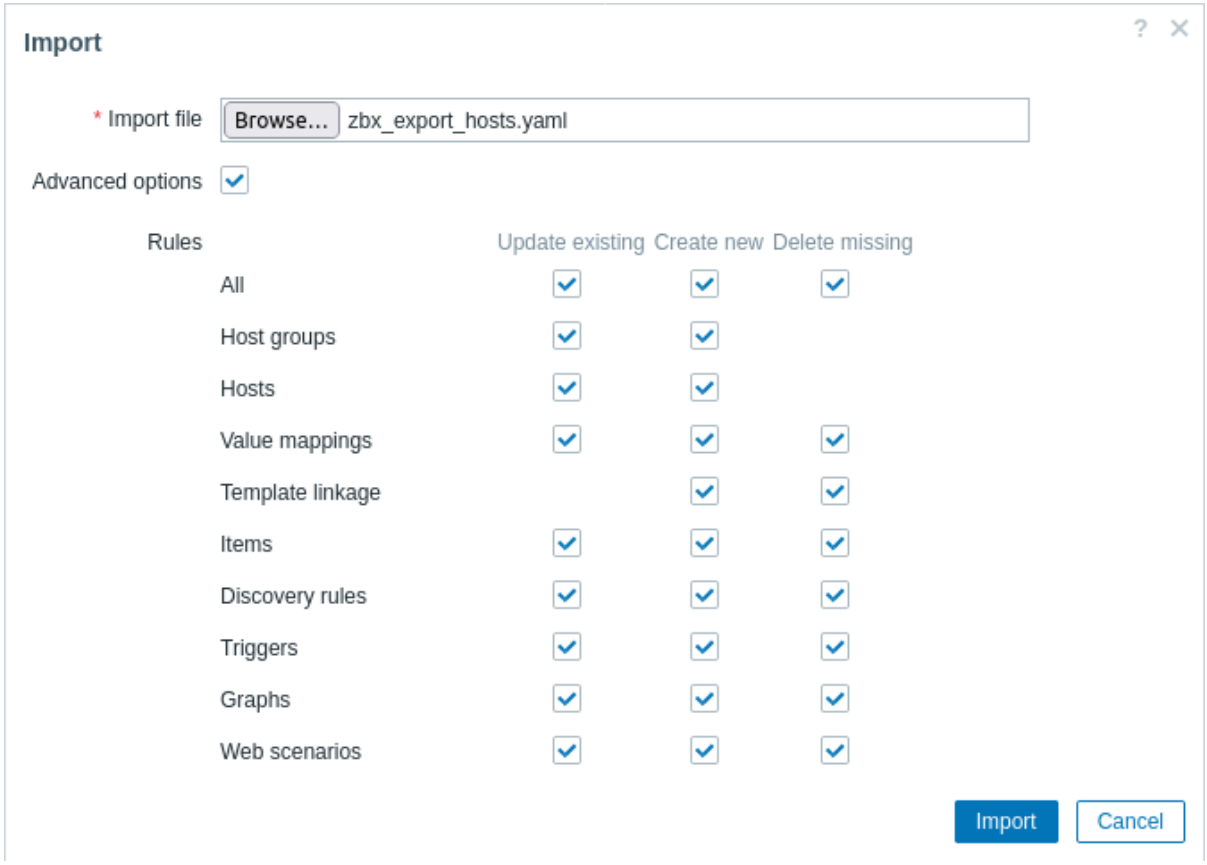
選択したフォーマットに応じて、ホストはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_hosts.yaml` - YAML エクスポート（エクスポートのデフォルトオプション）;
- `zabbix_export_hosts.xml` - XML エクスポート;
- `zabbix_export_hosts.json` - JSON エクスポート。

インポート

ホストをインポートするには、以下の手順を実行します。

1. データ収集 → ホストに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 設定フォームの右下隅にある インポートをクリックします。



詳細オプションチェックボックスをマークすると、インポート可能なすべての要素の詳細リストが表示されます。必要に応じて各インポートルールをマークまたはマーク解除してください。

すべて行のチェックボックスをクリックすると、その下のすべての要素がマークまたはマーク解除されます。

インポートルール:

ルール	説明
既存のものを更新	既存の要素は、インポートファイルのデータを使用して更新されます。そうでない場合は、更新されません。
新規作成	新しい要素は、インポートファイルのデータを使用して作成されます。そうでない場合は、作成されません。
存在しないものを削除	インポートファイルに存在しない既存の要素は削除されます。そうでない場合は、削除されません。 テンプレートリンクに対して 存在しないものを削除がマークされている場合、インポートファイルに存在しない現在のテンプレートリンクはリンク解除されます。リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は削除されません（各エンティティに対して 存在しないものを削除オプションが選択されている場合を除く）。

インポートの成功または失敗のメッセージがフロントエンドに表示されます。

エクスポート形式

ホストをエクスポートすると、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。エクスポートには、ホストのメタデータ、アイテム、マクロ、トリガーなどのホスト要素が含まれます。

各要素は特定の目的を持ち、ネストされた要素を含む場合があります。

以下のセクションでは、エクスポート形式の各要素について説明します。例では、MySQL by Zabbix agent 2 テンプレートを持つホストを使用しています。要素の説明のために、ホスト作成後にテンプレートのリンクを解除しています。リンク解除は、すべての要素を設定に保持します（リンク解除とクリアは要素を削除します）。

省略記号（...）は、簡潔にするために省略された要素を示します。要素が後のセクションでより詳細に説明されている場合は、（下表参照）という注記が使用されています。

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (下表参照)
  graphs: (下表参照)
```

ホストメタデータ

要素		型	説明
version		string	(必須) Zabbix バージョン。
host_groups			(必須) ホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
hosts			ホストのルート要素。
graphs			ホストグラフのルート要素。

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (下表参照)
      items: (下表参照)
      discovery_rules: (下表参照)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: (下表参照)
  (...)
```

ホスト

要素	型	説明
host	string	(必須) 一意のホスト名。
name	string	表示用ホスト名。
description	text	ホストの説明。
monitored_by	string	ホストの監視方法。 指定可能な値:[1] SERVER (0, デフォルト), PROXY (1), PROXY_GROUP (2)。
proxy		プロキシのルート要素。
name	string	(必須) ホストを監視するプロキシ名（存在する場合）。
proxy_group		プロキシグループのルート要素。
name	string	(必須) ホストの監視に使用されるプロキシグループ名（存在する場合）。
status	string	ホストのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。

要素	型	説明
ipmi_authtype	string	IPMI セッション認証タイプ。 指定可能な値:[1] DEFAULT (-1, デフォルト), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6)。
ipmi_privilege	string	IPMI セッション権限レベル。 指定可能な値:[1] CALLBACK (1), USER (2, デフォルト), OPERATOR (3), ADMIN (4), OEM (5)。
ipmi_username	string	IPMI チェック用ユーザー名。
ipmi_password	string	IPMI チェック用パスワード。
templates		リンクされたテンプレートのルート要素。
name	string	(必須) テンプレート名。
groups		ホストが所属するホストグループのルート要素。
name	string	(必須) ホストグループ名。
interfaces		ホストインターフェースのルート要素。
items		ホストアイテムのルート要素。
discovery_rules		ホストのローレベルディスカバリルールのルート要素。
http_tests		ホストの Web シナリオのルート要素。
tags		ホストタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
macros		ホストマクロのルート要素。
macro	string	(必須) ユーザーマクロ名。
type	string	ユーザーマクロタイプ。 指定可能な値:[1] TEXT (0, デフォルト), SECRET_TEXT (1), VAULT (2)。
value	string	ユーザーマクロ値。
description	string	ユーザーマクロの説明。
inventory		ホストインベントリのルート要素。
<inventory_property>	string	インベントリプロパティ。すべてのプロパティはそれぞれの要素（type、name、os など）を持つ（例: エクスポートフォーマット参照）。
inventory_mode	string	インベントリモード。 指定可能な値:[1] DISABLED (-1), MANUAL (0, デフォルト), AUTOMATIC (1)。
valuemaps		ホストの値のマッピングのルート要素。

Note:
関連項目: ホストオブジェクト（同名のプロパティを参照）。

ホストインターフェース

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

要素	型	説明
default	string	このインターフェースがプライマリホストインターフェースかどうか。1 つのホストに 1 つのタイプのプライマリインターフェースしか設定できません。 可能な値:[1] NO (0), YES (1, デフォルト)。
type	string	インターフェースタイプ。 可能な値:[1] ZABBIX (1, デフォルト), SNMP (2), IPMI (3), JMX (4)。
useip	string	ホストへの接続に IP を使用するかどうか（使用しない場合は DNS が使用されます）。 可能な値:[1] NO (0), YES (1, デフォルト)。
ip	string	(IP 接続の場合は必須）IP アドレス（IPv4 または IPv6）。
dns	string	(DNS 接続の場合は必須）DNS 名。
port	string	ポート番号。

要素	型	説明
details		インターフェース詳細のルート要素。
version	string	この SNMP バージョンを使用します。 可能な値:[1] SNMPV1 (1), SNMP_V2C (2, デフォルト), SNMP_V3 (3)。
community	string	(SNMPv1 および SNMPv2 アイテムの場合は必須) SNMP コミュニティ。
max_repetitions	string	ネイティブ SNMP バルクリクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 SNMPv2 および SNMPv3 アイテム (`discovery[]` および `walk[]` アイテム) でサポート。 デフォルト: 10。
contextname	string	SNMPv3 コンテキスト名。 SNMPv3 アイテムでサポート。
securityname	string	SNMPv3 セキュリティ名。 SNMPv3 アイテムでサポート。
securitylevel	string	SNMPv3 セキュリティレベル。 SNMPv3 アイテムでサポート。 可能な値:[1] NOAUTHNOPRIV (0, デフォルト), AUTHNOPRIV (1), AUTHPRIV (2)。
authprotocol	string	SNMPv3 認証プロトコル。 SNMPv3 アイテムでサポート。 可能な値:[1] MD5 (0, デフォルト), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5)。
authpassphrase	string	SNMPv3 認証パスフレーズ。 SNMPv3 アイテムでサポート。
privprotocol	string	SNMPv3 プライバシープロトコル。 SNMPv3 アイテムでサポート。 可能な値:[1] DES (0, デフォルト), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5)。
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。 SNMPv3 アイテムでサポート。
bulk	string	SNMP でバルクリクエストを使用するかどうか。 可能な値:[1] NO (0), YES (1, デフォルト)。
interface_ref	string	アイテムで使用するインターフェース参照名 (形式: `if<N>`)。

Note:
参考: ホストインターフェースオブジェクト (同名のプロパティを参照)。

ホストアイテム

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: '通常のバイナリログキャッシュ (`binlog_cache_size`より大きい) が収まらなかったため、一時ディスク
          preprocessing: (下表参照)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
```

```
        params: |
          ( last(//mysql.innodb_buffer_pool_pages_total) -
          last(//mysql.innodb_buffer_pool_pages_free) ) /
          ( last(//mysql.innodb_buffer_pool_pages_total) +
          ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
          ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
        description: 'バッファプール内の使用済みページ数と総ページ数の比率。'
        tags:
          - tag: component
            value: memory
        triggers: (下表参照)
      (...)
      - name: Uptime
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: 'サーバーが稼働している秒数。'
        preprocessing: (下表参照)
        master_item:
          key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
        tags:
          - tag: component
            value: application
        triggers: (下表参照)
      (...)
```

要素	型	説明
name	string	(必須) アイテム名。
type	string	アイテムタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23)。
snmp_oid	string	(SNMP_AGENT アイテムの場合は必須) SNMP オブジェクト ID。
key	string	(必須) アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムの場合は常に 0。
history	string	履歴データを保存する期間 (時間サフィックス、ユーザーマクロ、LLD マクロが使用可能)。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間 (時間サフィックス、ユーザーマクロ、LLD マクロが使用可能)。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
value_type	string	受信値の型。 指定可能な値:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, デフォルト), TEXT (4), BINARY (5)。
allowed_hosts	string	アイテムにデータ送信を許可するホストの IP アドレス (カンマ区切り)。 TRAP および HTTP_AGENT アイテムでサポート。
units	string	受信値の単位 (bps、B など)。
params	text	アイテムタイプに応じた追加パラメータ (SSH および TELNET アイテムの場合は実行スクリプト、ODBC アイテムの場合は SQL クエリ、CALCULATED アイテムの場合は計算式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムの場合はスクリプト)。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポート。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポート。 SSH アイテムの場合の指定可能な値:[1] PASSWORD (0, デフォルト), PUBLIC_KEY (1)。 HTTP_AGENT アイテムの場合の指定可能な値:[1] NONE (0, デフォルト), BASIC (1), NTLM (2)。

要素	型	説明
username	string	(SSH および TELNET アイテムの場合は必須) 認証用ユーザー名。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポート。 JMX アイテムで使用する場合は、password（下記）も指定するか、両方空欄にする必要があります。
password	string	(SSH および TELNET アイテムの場合は必須) 認証用パスワード。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポート。 JMX アイテムで使用する場合は、username（上記）も指定するか、両方空欄にする必要があります。
publickey	string	(SSH アイテムの場合は必須) 公開鍵ファイル名。
privatekey	string	(SSH アイテムの場合は必須) 秘密鍵ファイル名。
description	text	アイテムの説明。
inventory_link	string	アイテムによって入力されるホストインベントリフィールド。 指定可能な値:[1] NONE (0), ALIAS (4) など（サポートされるフィールドはホストインベントリを参照）。
valuemap		アイテムの値のマッピングのルート要素。
name	string	(必須) アイテムで使用する値のマッピング名。
logtimefmt	string	ログエントリ内の時刻のフォーマット。 LOG 値タイプのアイテムでサポート。
preprocessing		アイテム値の事前処理のルート要素。
step		ホストアイテム値の事前処理ステップのルート要素。
interface_ref	string	ホストインターフェースへの参照（フォーマット: if<N>）。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポート。
master_item		(DEPENDENT アイテムの場合は必須) 従属アイテムのマスターアイテムのルート要素。
key	string	(必須) 従属アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 タイムアウトリストのアイテムタイプでサポート。
url	string	(HTTP_AGENT アイテムの場合は必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポート。
name	string	(HTTP_AGENT アイテムの場合は必須) クエリパラメータ名。
value	string	クエリパラメータ値。 HTTP_AGENT アイテムでサポート。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポート。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムの場合は必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータ値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポート。
posts	string	HTTP(S) リクエストボディデータ。 HTTP_AGENT アイテムでサポート。
status_codes	string	必要な HTTP ステータスコードの範囲（カンマ区切り）。 HTTP_AGENT アイテムでサポート。
follow_redirects	string	データ取得時にレスポンスのリダイレクトを追従するか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
post_type	string	ポストデータボディのタイプ。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] RAW (0, デフォルト), JSON (2), XML (3)。
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポート。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポート。
name	string	(HTTP_AGENT アイテムの場合は必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムの場合は必須) ヘッダー値。
retrieve_mode	string	レスポンスのどの部分を保存するか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
request_method	string	リクエストメソッドタイプ。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] GET (0, デフォルト), POST (1), PUT (2), HEAD (3)。

要素	型	説明
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] RAW (0, デフォルト), JSON (1)。
allow_traps	string	trapper アイテムと同様に値を入力できるか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
ssl_cert_file	string	公開 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポート。
ssl_key_file	string	秘密 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポート。
ssl_key_password	string	SSL キーのパスワード。 HTTP_AGENT アイテムでサポート。
verify_peer	string	ホストの証明書が正当かどうか検証するか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
verify_host	string	接続先のホスト名が証明書のホスト名と一致するか検証するか。 HTTP_AGENT アイテムでサポート。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
tags		アイテムタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
triggers		ホストアイテムトリガーのルート要素。

Note:
関連項目: アイテムオブジェクト（同名のプロパティを参照）。

ホストアイテムの値の事前処理ステップ

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

要素	型	説明
type	string	(必須) アイテム値の事前処理ステップのタイプ。 指定可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, 計算式: 受信値 - 前回値), CHANGE_PER_SECOND (10, 計算式: (受信値 - 前回値)/(現在時刻 - 前回チェック時刻)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30)。

要素	型	説明
parameters		(必須) アイテム値の事前処理ステップのパラメータのルート要素。
parameter	string	アイテム値の事前処理ステップの個別パラメータ。
error_handler	string	事前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値:[1] ORIGINAL_ERROR (0, デフォルト), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3)。
error_handler_params	string	エラーハンドラのパラメータ。

Note:
参照: アイテム事前処理オブジェクト (同名のプロパティを参照)。

ホストアイテムのトリガー

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: Failed to fetch info data'
              event_name: 'MySQL: Failed to fetch info data (or no data for 30m)'
              priority: INFO
              description: 'Zabbix has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

要素	型	説明
uuid	string	(必須) このトリガーの一意の識別子。
expression	string	(必須) トリガー式。
recovery_mode	string	OK イベント生成の基準。 指定可能な値:[1] EXPRESSION (0, デフォルト), RECOVERY_EXPRESSION (1), NONE (2)。
recovery_expression	string	トリガーの復旧式。
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関)。 指定可能な値:[1] DISABLED (0, デフォルト), TAG_VALUE (1)。
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(必須) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。

要素	型	説明
status	string	トリガーのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
priority	string	トリガーの深刻度。 指定可能な値:[1] NOT_CLASSIFIED (0, デフォルト), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5)。
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の障害イベントまたは複数の障害イベント）。 指定可能な値:[1] SINGLE (0, デフォルト), MULTIPLE (1)。
manual_close	string	障害イベントの手動クローズ。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
dependencies		依存関係のルート要素。
name	string	(必須) 依存トリガー名。
expression	string	(必須) 依存トリガー式。
recovery_expression	string	依存トリガーの復旧式。
tags		トリガータグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
関連項目: トリガーオブジェクト（同名のプロパティを参照）。

ホストのローレベルディスカバリルール

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
                  value: storage
                - tag: database
                  value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
              path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
ほとんどのホストのローレベルディスカバリルール要素は、ホストアイテムと同じです。下表では、ホストアイテムと異なる要素について説明します。

要素	型	説明
type	string	アイテムタイプ。 可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22)。
key	string	(必須) ローレベルディスカバリルールのキー。
filter		ホストのローレベルディスカバリルールフィルタのルート要素。
lifetime	string	検出されなくなったリソースが削除されるまでの期間（秒、時間サフィックス、またはユーザーマクロを使用)。 デフォルト: 7d。
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 可能な値: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER。
enabled_lifetime	string	検出されなくなったリソースが無効化されるまでの期間（秒、時間サフィックス、またはユーザーマクロを使用)。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 可能な値: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER。
item_prototypes		ホストアイテムプロトタイプ要素のルート要素。ホストアイテムと同じです。
trigger_prototypes		ホストトリガープロトタイプ要素のルート要素。ホストアイテムトリガーと同じです。
graph_prototypes		ホストグラフプロトタイプのルート要素。ホストグラフと同じです。
host_prototypes		ホストプロトタイプのルート要素。ホストと同じです。
parent_discovery_rule		ローレベルディスカバリルールプロトタイプの親ローレベルディスカバリルール（またはルールプロトタイプ）のルート要素。このプロパティは、参照されたルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	(必須) 親ローレベルディスカバリルール（またはルールプロトタイプ）のキー。
master_item	string	(DEPENDENT ルールの場合は必須) 従属ルールのマスターアイテムのルート要素。
lld_macro_paths		ローレベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	(必須) ローレベルディスカバリルールのマクロ名。
path	string	(必須) 対応するマクロに割り当てられる値のセレクタ。
preprocessing		ローレベルディスカバリルール値のプリプロセスのルート要素。
step		ホストアイテム値のプリプロセスステップと同じですが、可能な値が少なくなっています。 詳細はLLD ルールプリプロセスオブジェクト も参照してください。
overrides		ローレベルディスカバリルールのオーバーライドルールのルート要素。
name	string	(必須) 一意のオーバーライド名。
step	string	(必須) オーバーライドの一意の順序番号。
stop	string	一致した場合、次のオーバーライドの処理を停止します。
filter		ホストのローレベルディスカバリルールフィルタと同じローレベルディスカバリルールオーバーライドルールフィルタ要素のルート要素。
operations		ホストのローレベルディスカバリルールオーバーライド操作のルート要素。

Note:
関連するプロパティ名で参照: LLD ルールオブジェクト も参照してください。

ホストのローレベルディスカバリールールのフィルター

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
            evaltype: AND
            conditions:
```

```
            - macro: '{#DATABASE}'
              value: '{$MYSQL.DBNAME.MATCHES}'
            - macro: '{#DATABASE}'
              value: '{$MYSQL.DBNAME.NOT_MATCHES}'
              operator: NOT_MATCHES_REGEX
        (...)
```

要素	型	説明
evaltype	文字列	フィルター条件の評価方法を上書きします。 指定可能な値:[1] AND_OR (0, デフォルト), AND (1), OR (2), FORMULA (3)。
formula	文字列	フィルター条件のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	文字列	(必須) チェックを実行するローレベルディスカバリーマクロの名前。
value	文字列	比較する値。
operator	文字列	条件演算子。 指定可能な値:[1] MATCHES_REGEX (8, デフォルト), NOT_MATCHES_REGEX (9)。
formulaid	文字列	(必須) カスタム式から条件を参照するために使用される任意の一意な ID。大文字のみ使用可能。フィルター条件を変更する際にユーザーが定義する必要がありますが、その後リクエストする際には新たに生成されます。

Note:
参照: LLD ルールフィルターオブジェクト（該当するプロパティ名で参照）。

ホストのローレベルディスカバリールールのオーバーライド操作

要素	型	説明
operationobject	string	操作が適用されるオブジェクト。 可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4)。
operator	string	オーバーライド条件の演算子。 可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6)。
value	string	オーバーライド条件演算子のための正規表現または文字列。
status	string	オーバーライド操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトが追加されるかどうか。
delay	string	オーバーライド操作時にアイテムプロトタイプに設定される更新間隔。
history	string	オーバーライド操作時にアイテムプロトタイプに設定される履歴保存期間。
trends	string	オーバーライド操作時にアイテムプロトタイプに設定されるトレンド保存期間。
severity	string	オーバーライド操作時に設定されるトリガープロトタイプの深刻度。
tags		オーバーライド操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
templates		オーバーライド操作時にホストプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	オーバーライド操作時に設定されるホストプロトタイプのインベントリモード。

Note:
関連項目: LLD ルールのオーバーライド操作オブジェクト（同名のプロパティを参照）。

ホスト Web シナリオ

要素	型	説明
uuid	string	(必須) この Web シナリオの一意の識別子。
name	string	(必須) Web シナリオ名。

要素	型	説明
delay	string	Web シナリオを実行する頻度（秒、時間サフィックス、またはユーザーマクロを使用）。 デフォルト: 1m。
attempts	integer	Web シナリオステップの実行試行回数。 可能な値: 1-10（デフォルト: 1）。
agent	string	クライアントエージェント。Zabbix は選択したブラウザになりすまします。 これは、Web サイトが異なるブラウザに対して異なるコンテンツを返す場合に便利です。 デフォルト: Zabbix。
http_proxy	string	Web シナリオで使用するプロキシ。 `http://[username[:password]@]proxy.example.com[:port]` の形式で指定。
variables		シナリオステップで使用できる Web シナリオ変数のルート要素。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信する HTTP ヘッダーのルート要素。 ヘッダーは HTTP プロトコルで表示されるのと同じ構文でリストする必要があります。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
status	string	Web シナリオのステータス。 可能な値:[1] 有効 (0, デフォルト)、無効 (1)。
authentication	string	認証方式。 可能な値:[1] なし (0, デフォルト)、BASIC (1)、NTLM (2)。
http_user	string	BASIC(HTTP) または NTLM 認証で使用するユーザー名。
http_password	string	BASIC(HTTP) または NTLM 認証で使用するパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証するかどうか。 可能な値:[1] いいえ (0, デフォルト)、はい (1)。
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致するかどうかを検証する。 可能な値:[1] いいえ (0, デフォルト)、はい (1)。
ssl_cert_file	string	クライアント認証で使用する SSL 証明書ファイル名（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証で使用する SSL 秘密鍵ファイル名（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		(必須) ホスト Web シナリオステップのルート要素。
tags		Web シナリオタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
関連項目: Web シナリオオブジェクト（同名のプロパティを参照）。

ホストの Web シナリオステップ

要素	型	説明
name	string	(必須) Web シナリオステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL に追加される HTTP フィールドの配列）。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（生の POST データ）または HTTP フィールドの配列（フォームフィールドデータ））。
name	string	(必須) POST フィールド名。
value	string	(必須) POST フィールド値。
variables		このステップの後に適用されるべきステップレベルの変数（マクロ）のルート要素。 変数値に'regex:' プレフィックスがある場合、その値はこのステップで返されたデータから'regex:' プレフィックスに続く正規表現パターンに従って抽出されます
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。
name	string	(必須) ヘッダー名。

要素	型	説明
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトの追従。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、時間のサフィックス、またはユーザーマクロを使用)。 デフォルト: 15s。
required	string	レスポンスに含まれている必要があるテキスト（空の場合は無視)。
status_codes	string	許可される HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。空の場合は無視)。

Note:
参照: Web シナリオステップオブジェクト（一致する名前の関連プロパティを参照)。

ホストグラフ

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (see table below)
    (...)
```

要素	型	説明
uuid	string	グラフの一意の識別子。
name	string	(必須) グラフ名。
width	integer	グラフの幅（ピクセル単位)。 プレビューおよび円グラフ/爆発グラフで使用。 可能な値:[1] 20-65535 (デフォルト: 900)。
height	integer	グラフの高さ（ピクセル単位)。 プレビューおよび円グラフ/爆発グラフで使用。 可能な値:[1] 20-65535 (デフォルト: 900)。
yaxismin	double	Y 軸の最小値。 Y 軸の最小値が FIXED の場合に対応。 デフォルト: 0。
yaxismax	double	Y 軸の最大値。 X 軸の最大値が FIXED の場合に対応。 デフォルト: 0。
show_work_period	string	非稼働時間のハイライト。 NORMAL および STACKED グラフで対応。 可能な値:[1] NO (0), YES (1, デフォルト)。
show_triggers	string	シンプルトリガー値を線で表示。 NORMAL および STACKED グラフで対応。 可能な値:[1] NO (0), YES (1, デフォルト)。
type	string	グラフタイプ。 可能な値:[1] NORMAL (0, デフォルト), STACKED (1), PIE (2), EXPLODED (3)。
show_legend	string	グラフの凡例を表示。 可能な値:[1] NO (0), YES (1, デフォルト)。
show_3d	string	3D スタイルを有効化。 NORMAL および STACKED グラフで対応。 可能な値:[1] NO (0, デフォルト), YES (1)。
percent_left	double	左軸のパーセンタイル線を表示。 NORMAL グラフで対応。 デフォルト: 0。
percent_right	double	右軸のパーセンタイル線を表示。 NORMAL グラフで対応。 デフォルト: 0。

要素	型	説明
ymin_type_1	string	Y 軸の最小値。 NORMAL および STACKED グラフで対応。 可能な値:[1] CALCULATED (0, デフォルト), FIXED (1), ITEM (2)。
ymin_item_1		(ymin_type_1 が ITEM の場合は必須) 個別アイテム詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムのキー。
ymax_type_1	string	Y 軸の最大値。 NORMAL および STACKED グラフで対応。 可能な値:[1] CALCULATED (0, デフォルト), FIXED (1), ITEM (2)。
ymax_item_1		(ymax_type_1 が ITEM の場合は必須) 個別アイテム詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムのキー。
graph_items		(必須) ホストグラフアイテムのルート要素。

Note:
参考: Graph object（同名のプロパティを参照）。

ホストグラフアイテム

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

要素	型	説明
sortorder	integer	描画順序。値が小さいほど先に描画されます。線や領域を他の線や領域の後ろ（または前）に描画するために使用できます。
drawtype	string	グラフアイテムの描画スタイル。 NORMAL グラフでサポート。 指定可能な値:[1] SINGLE_LINE (0, デフォルト), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5)。
color	string	要素の色（6 桁の 16 進数）。
yaxisside	string	グラフアイテムの Y スケールが描画されるグラフの側。 NORMAL および STACKED グラフでサポート。
calc_fnc	string	アイテムに複数の値が存在する場合に描画するデータ。 指定可能な値:[1] MIN (1), AVG (2, デフォルト), MAX (4), ALL (7; 最小値、平均値、最大値; シンプルグラフでサポート), LAST (9; 円グラフ/爆発グラフでサポート)。
type	string	グラフアイテムのタイプ。 指定可能な値:[1] SIMPLE (0, デフォルト), GRAPH_SUM (2; アイテムの値が円グラフ全体を表す; 円グラフ/爆発グラフでサポート)。
item		(必須) 個々のアイテム。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムのキー。

Note:
関連項目: グラフアイテムオブジェクト（同名のプロパティを参照）。

ホストの値のマッピング

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

要素	タイプ	説明
uuid	string	(必須) この値のマッピングの一意の識別子。
name	string	(必須) 値のマッピング名。
mapping		マッピングのルート要素。
type	string	マッピングの一致タイプ。 可能な値:[1] EQUAL (0, デフォルト), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6)。
value	string	元の値。
newvalue	string	(必須) 元の値がマッピングされる値。

Note:
参照: 値のマッピングオブジェクト（同名のプロパティを参照）。

脚注 [1] 括弧内の API 整数値（例：ENABLED（0））は参考のためにのみ記載されています。
 詳細は、表のエントリまたは各セクションの最後にあるリンク先の API オブジェクトページを参照してください。

5 ネットワークマップ

概要

ネットワークマップのエクスポートには以下が含まれます。

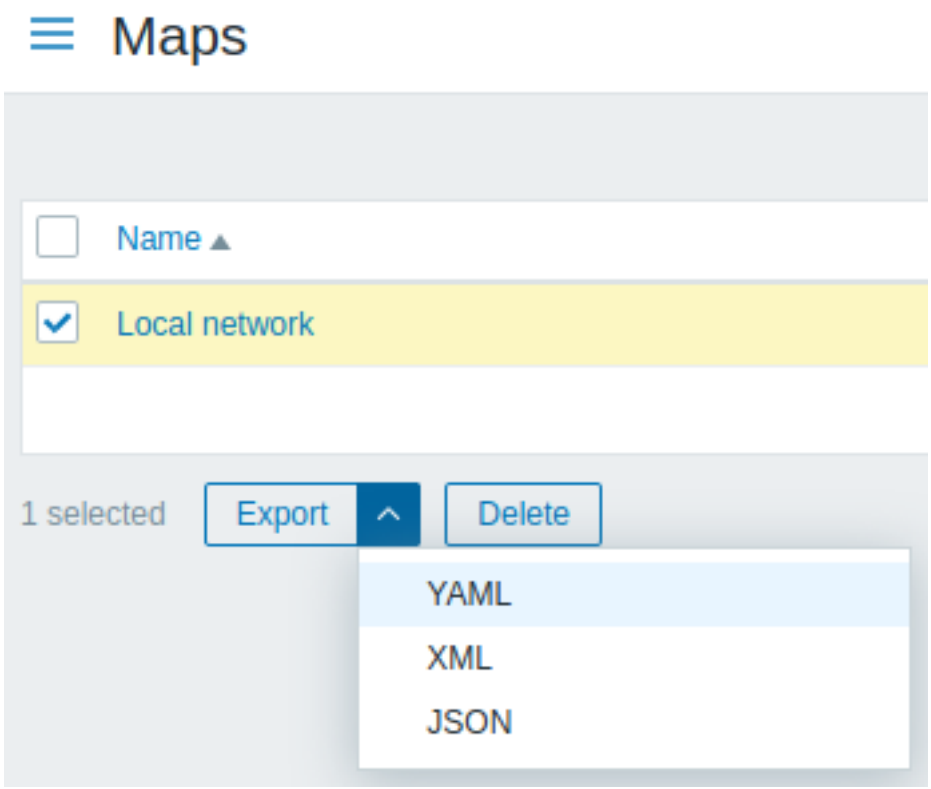
- 関連するすべてのイメージ
- マップ構造（すべてのマップ設定、すべての要素とその設定、マップリンクおよびマップリンクステータスインジケータ）

Warning:
エクスポートされたマップに関連するホストグループ、ホスト、トリガー、他のマップや他の要素はエクスポートされません。したがって、マップが参照する要素のうち少なくとも 1 つが存在しない場合、インポートは失敗します。

エクスポート

ネットワークマップをエクスポートするには、以下の手順を実行します。

1. 監視 → マップに移動します。
2. エクスポートするネットワークマップのチェックボックスをオンにします。
3. リストの下にある エクスポートをクリックします。



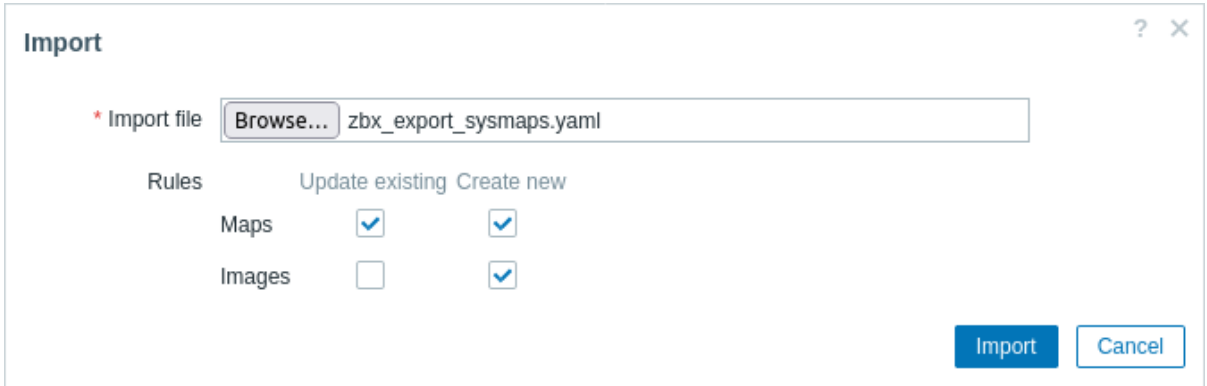
選択したフォーマットに応じて、マップはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_maps.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）
- `zabbix_export_maps.xml` - XML エクスポートの場合
- `zabbix_export_maps.json` - JSON エクスポートの場合

インポート

ネットワークマップをインポートするには、次の手順を実行します。

1. 監視 → マップに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. インポートルールで必要なオプションをマークします。
5. 設定フォームの右下隅にある インポートをクリックします。



インポートルール:

ルール	説明
既存のものを更新	既存のマップはインポートファイルのデータで更新されます。チェックしない場合は更新されません。
新規作成	インポートファイルのデータで新しいマップが作成されます。チェックしない場合は作成されません。

両方のマップオプションのチェックを外し、画像の該当オプションにチェックを入れると、画像のみがインポートされます。画像のインポートは スーパ管理者ユーザーのみが利用できます。

インポートの成功または失敗のメッセージがフロントエンドに表示されます。

Warning:
既存の画像を置き換える場合、その画像を使用しているすべてのマップに影響します。

エクスポートフォーマット

YAML へのエクスポート:

```
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
            name: Zabbix_server_3D_(128)
          icon_on: {  }
```

```
        icon_disabled: {  }
        icon_maintenance: {  }
        urls: {  }
        evaltype: '0'
    shapes:
      - type: '0'
        x: '0'
        'y': '0'
        width: '680'
        height: '15'
        text: '{MAP.NAME}'
        font: '9'
        font_size: '11'
        font_color: '000000'
        text_halign: '0'
        text_valign: '0'
        border_type: '0'
        border_width: '0'
        border_color: '000000'
        background_color: ''
        zindex: '0'
    lines: {  }
    links: {  }
```

エクスポートされた要素　エクスポートされた要素については、以下の表で説明します。

要素		型	説明
images			イメージのルート要素。
	name	string	一意のイメージ名。
	imagetype	integer	イメージタイプ。 指定可能な値： 1 - イメージ; 2 - 背景。
	encodedImage	string	Base64 エンコードされたイメージ。
maps			マップのルート要素。

マップ

要素	タイプ	説明
name	string	一意のマップ名。
width	integer	マップの幅（ピクセル単位）。
height	integer	マップの高さ（ピクセル単位）。
label_type	integer	マップ要素のラベルタイプ。 設定可能な値： 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし。
label_location	integer	マップ要素のデフォルトのラベル位置。 設定可能な値： 0 - 下; 1 - 左; 2 - 右; 3 - 上。
highlight	integer	アクティブなトリガーやホストステータスのアイコンハイライトを有効にする。 設定可能な値： 0 - いいえ; 1 - はい。

要素	タイプ	説明
expandproblem	integer	単一の問題を持つ要素の問題トリガーを表示する。 設定可能な値： 0 - いいえ; 1 - はい。
markelements	integer	最近ステータスが変化したマップ要素をハイライトする。 設定可能な値： 0 - いいえ; 1 - はい。
show_unack	integer	問題の表示。 設定可能な値： 0 - すべての問題の数; 1 - 未確認の問題の数; 2 - 確認済みと未確認の問題の数を個別に表示。
severity_min	integer	デフォルトでマップに表示するトリガーの最小深刻度。 設定可能な値： 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度; 4 - 高度; 5 - 障害。
show_suppressed	integer	ホストメンテナンスのために抑制（非表示）されるはずの問題を表示する。 設定可能な値： 0 - いいえ; 1 - はい。
grid_size	integer	マップグリッドのセルサイズ（ピクセル単位）。 `grid_show` が 0 の場合にサポート。 設定可能な値：20, 40, 50, 75, 100。
grid_show	integer	マップ設定でグリッドを表示する。 設定可能な値： 0 - はい; 1 - いいえ。
grid_align	integer	マップ設定でアイコンを自動的に整列する。 設定可能な値： 0 - はい; 1 - いいえ。
label_format	integer	高度なラベル設定を使用する。 設定可能な値： 0 - いいえ; 1 - はい。
label_type_host	integer	ラベルをホストラベルとして表示する。 `label_format` が 1 の場合にサポート。 設定可能な値： 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_hostgroup	integer	ラベルをホストグループラベルとして表示する。 `label_format` が 1 の場合にサポート。 設定可能な値： 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。

要素	タイプ	説明
label_type_trigger	integer	ラベルをトリガーラベルとして表示する。 label_format が 1 の場合にサポート。 設定可能な値： 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_map	integer	ラベルをマップラベルとして表示する。 label_format が 1 の場合にサポート。 設定可能な値： 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_image	integer	ラベルをイメージラベルとして表示する。 label_format が 1 の場合にサポート。 設定可能な値： 0 - ラベル; 2 - 要素名; 4 - なし; 5 - カスタムラベル。
label_string_host	string	ホスト要素のカスタムラベル。 label_type_host が 5 の場合にサポート。
label_string_hostgroup	string	ホストグループ要素のカスタムラベル。 label_type_hostgroup が 5 の場合にサポート。
label_string_trigger	string	トリガー要素のカスタムラベル。 label_type_trigger が 5 の場合にサポート。
label_string_map	string	マップ要素のカスタムラベル。 label_type_map が 5 の場合にサポート。
label_string_image	string	イメージ要素のカスタムラベル。 label_type_image が 5 の場合にサポート。
expand_macros	integer	マップ設定でラベル内のマクロを展開する。 設定可能な値： 0 - いいえ; 1 - はい。
background		背景画像のルート要素（ある場合）。 imagetype が 2 の場合にサポート。
name	string	背景画像名。
iconmap		アイコンマッピングのルート要素（ある場合）。
name	string	アイコンマッピング名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
elementtype	integer	リンクが属するマップアイテムタイプ。 設定可能な値： 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。
selements		マップ要素のルート要素。
shapes		マップ図形のルート要素。
type	integer	図形タイプ。 設定可能な値： 0 - 長方形; 1 - 楕円。
x	integer	図形の X 座標（ピクセル単位）。
y	integer	図形の Y 座標（ピクセル単位）。
width	integer	図形の幅。

要素		タイプ	説明
	height	integer	図形の高さ。
	text	string	図形内のテキスト。
	font	integer	テキストのフォントスタイル。 設定可能な値： 0 - Georgia, serif; 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif; 2 - ”Times New Roman”, Times, serif; 3 - Arial, Helvetica, sans-serif; 4 - ”Arial Black”, Gadget, sans-serif; 5 - ”Comic Sans MS”, cursive, sans-serif; 6 - Impact, Charcoal, sans-serif; 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif; 8 - Tahoma, Geneva, sans-serif; 9 - ”Trebuchet MS”, Helvetica, sans-serif; 10 - Verdana, Geneva, sans-serif; 11 - ”Courier New”, Courier, monospace; 12 - ”Lucida Console”, Monaco, monospace。
	font_size	integer	フォントサイズ（ピクセル単位）。
	font_color	string	16 進数コードで表現されたフォントカラー。
	text_halign	integer	テキストの水平位置揃え。 設定可能な値： 0 - 中央; 1 - 左; 2 - 右。
	text_valign	integer	テキストの垂直位置揃え。 設定可能な値： 0 - 中央; 1 - 上; 2 - 下。
	border_type	integer	図形の枠線タイプ。 設定可能な値： 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	border_width	integer	枠線の幅（ピクセル単位）。
	border_color	string	16 進数コードで表現された枠線カラー。
	background_color	string	16 進数コードで表現された背景（塗りつぶし）カラー。
	zindex	integer	すべての図形と線の順序付け値（z-index）。
lines			マップ線のルート要素。
	x1	integer	線の点 1 の X 座標（ピクセル単位）。
	y1	integer	線の点 1 の Y 座標（ピクセル単位）。
	x2	integer	線の点 2 の X 座標（ピクセル単位）。
	y2	integer	線の点 2 の Y 座標（ピクセル単位）。
	line_type	integer	線のタイプ。 設定可能な値： 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	line_width	integer	線の幅（ピクセル単位）。
	line_color	string	16 進数コードで表現された線のカラー。
	zindex	integer	すべての図形と線の順序付け値（z-index）。
links			マップ要素間のリンクのルート要素。
	drawtype	integer	リンクスタイル。 設定可能な値： 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
	color	string	リンクカラー（6 桁の 16 進数）。

要素	タイプ	説明
label	string	リンクラベル。
selementid1	id	接続する 1 つ目の要素の ID。
selementid2	id	接続する 2 つ目の要素の ID。
linktriggers		リンクステータスインジケーターのルート要素。

Note:
関連項目：マップオブジェクト（同名のプロパティを参照）。

マップ要素

要素	型	説明
elementtype	integer	マップ要素のタイプ。 可能な値： 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。
elements		マップ上に表現される Zabbix エンティティ（ホスト、ホストグループ、マップなど）のルート要素。 すべてのエンティティは、それぞれの要素（host など；エクスポート形式を参照）を持つ。
label	string	アイコンラベル。
label_location	integer	ラベルの位置。 可能な値： -1 - マップのデフォルトを使用; 0 - 下; 1 - 左; 2 - 右; 3 - 上。
x	integer	X 軸上の位置。
y	integer	Y 軸上の位置。
elementsubtype	integer	要素のサブタイプ。 elementtype が 3 に設定されている場合にサポート。 可能な値： 0 - 単一のホストグループ; 1 - すべてのホストグループ。
areatype	integer	エリアサイズ。 elementtype が 1 に設定されている場合にサポート。 可能な値： 0 - マップ全体と同じ; 1 - カスタムサイズ。
width	integer	エリアの幅。 areatype が 1 に設定されている場合にサポート。
height	integer	エリアの高さ。 areatype が 1 に設定されている場合にサポート。
viewtype	integer	エリア配置アルゴリズム。 elementsubtype が 1 に設定されている場合にサポート。 可能な値： 0 - エリア内に均等に配置。
use_iconmap	integer	この要素にアイコンマッピングを使用。マップレベルでアイコンマッピングが有効な場合のみ有効。 可能な値： 0 - いいえ; 1 - はい。
selementid	id	一意の要素レコード ID。
icon_off		要素が「OK」ステータスのときに使用するイメージのルート要素。
name	string	一意のイメージ名。

要素	型	説明
icon_on		要素が「障害」ステータスのときに使用するイメージのルート要素。
name	string	一意のイメージ名。
icon_disabled		要素が無効のときに使用するイメージのルート要素。
name	string	一意のイメージ名。
icon_maintenance		要素がメンテナンス中のときに使用するイメージのルート要素。
name	string	一意のイメージ名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
evaltype	integer	タグの評価タイプ。
tags		問題タグのルート要素（ホストおよびホストグループ要素用）。 タグが指定されている場合、これらのタグを持つ問題のみがマップ上に表示される。
tag	string	タグ名。
value	string	タグ値。
operator	integer	演算子。

Note:
関連項目：マップ要素オブジェクト（該当するプロパティは同名のものを参照）。

マップリンクステータスインジケーター

要素	タイプ	説明
drawtype	integer	トリガーが「障害」状態のときのリンクスタイル。 指定可能な値： 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	トリガーが「障害」状態のときのリンク色（6 桁の 16 進数）。
trigger		リンクステータスを示すために使用されるトリガーのルート要素。
description	string	トリガー名。
expression	string	トリガー式。
recovery_expression	string	トリガーのリカバリー式。

Note:
参照: マップリンクトリガーオブジェクト（該当するプロパティ名で参照）。

6 メディアタイプ

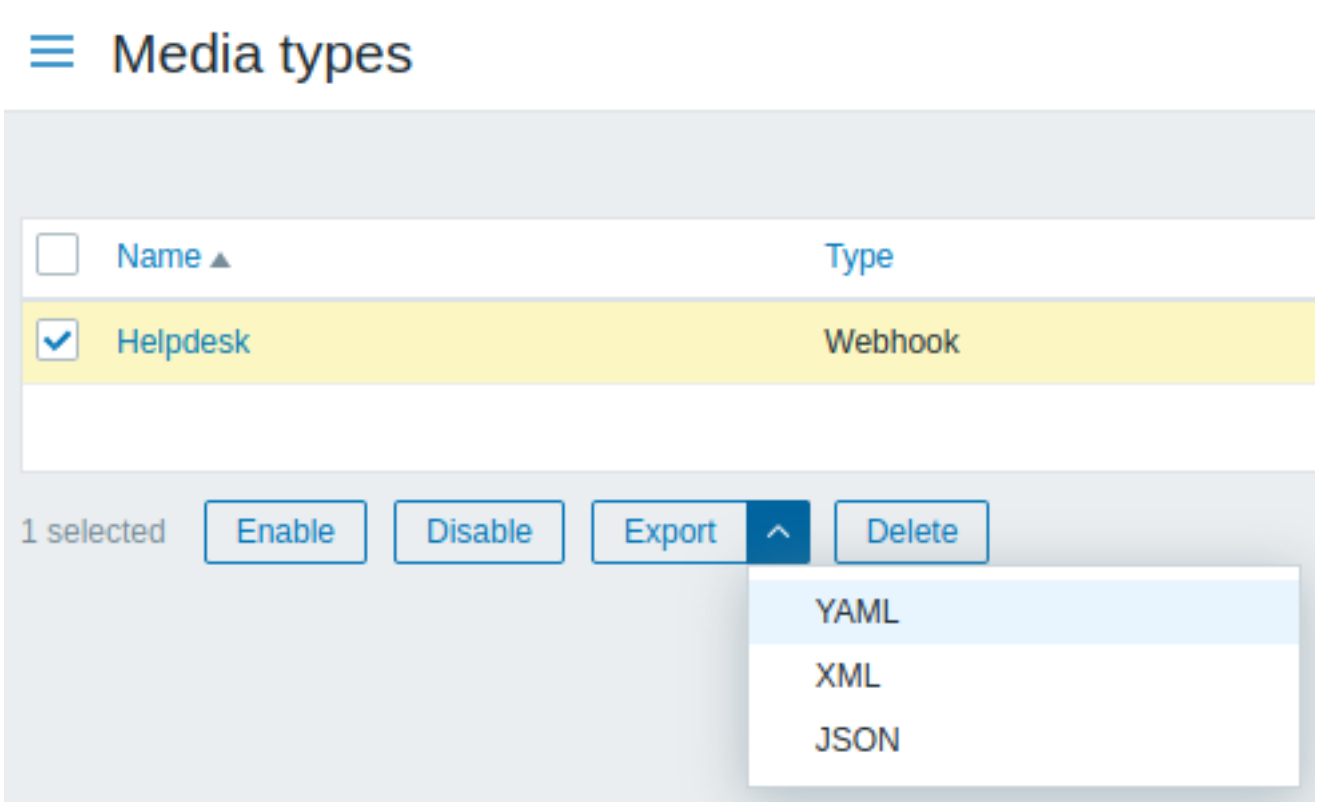
概要

メディアタイプは、すべての関連オブジェクトおよびオブジェクトの関連付けとともにエクスポートされます。

エクスポート

メディアタイプをエクスポートするには、以下の手順を実行します。

1. アラート → メディアタイプに移動します。
2. エクスポートするメディアタイプのチェックボックスをオンにします。
3. リストの下にある エクスポートをクリックします。



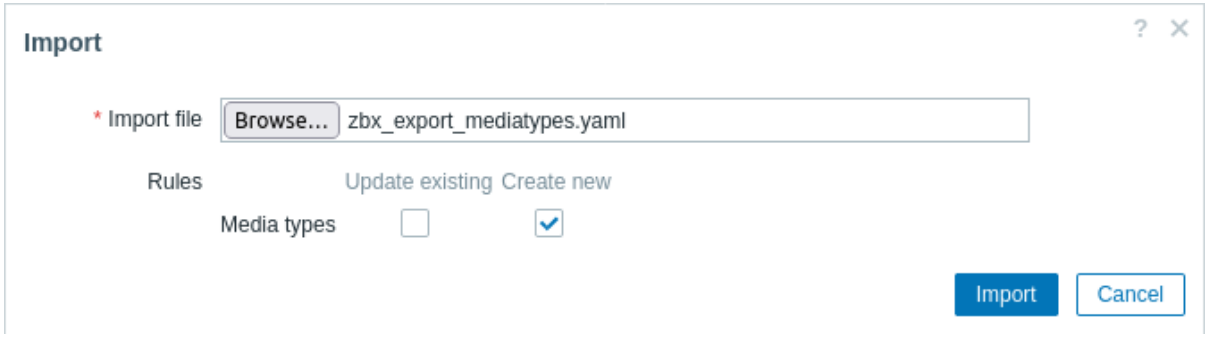
選択したフォーマットに応じて、メディアタイプはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_mediatypes.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）;
- `zabbix_export_mediatypes.xml` - XML エクスポートの場合;
- `zabbix_export_mediatypes.json` - JSON エクスポートの場合。

インポート

メディアタイプをインポートするには、次の手順を実行します。

1. アラート → メディアタイプに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. インポートルールで必要なオプションにチェックを入れます。
5. 設定フォームの右下隅にある インポートをクリックします。



インポートルール:

ルール	説明
既存のものを更新	既存の要素はインポートファイルのデータで更新されます。チェックしない場合は更新されません。
新規作成	新しい要素はインポートファイルのデータで作成されます。チェックしない場合は作成されません。

インポートの成功または失敗メッセージがフロントエンドに表示されます。

エクスポート形式

YAML へのエクスポート:

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
                    {name: 'average', color: '#FFA059'},
                    {name: 'high', color: '#E97659'},
```

```
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw 'field "retry" should be a number with value of at least 30 if "priority" is set
        }
```

```
            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    セットアップガイドはこちらを参照してください: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/temp

    token パラメータに Pushover アプリケーションキーを設定してください。
    Zabbix ユーザーに Pushover メディアを割り当てる際は、送信先フィールドにユーザーキーを追加してください。
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problem: {EVENT.NAME}'
      message: |
        {EVENT.DATE} {EVENT.TIME} に問題が発生しました
        問題名: {EVENT.NAME}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        オペレーションデータ: {EVENT.OPDATA}
        元の問題 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: RECOVERY
      subject: 'Resolved in {EVENT.DURATION}: {EVENT.NAME}'
```

```
      message: |
        {EVENT.RECOVERY.DATE} {EVENT.RECOVERY.TIME} に問題が解決されました
        問題名: {EVENT.NAME}
        問題の継続時間: {EVENT.DURATION}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        元の問題 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: 'Updated problem in {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} が {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME} に {EVENT.UPDATE.ACTION} しました。
        {EVENT.UPDATE.MESSAGE}

        現在の問題ステータスは {EVENT.STATUS}、経過時間は {EVENT.AGE}、確認済み: {EVENT.ACK.STATUS}。
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        検出ルール: {DISCOVERY.RULE.NAME}

        デバイス IP: {DISCOVERY.DEVICE.IPADDRESS}
        デバイス DNS: {DISCOVERY.DEVICE.DNS}
        デバイスステータス: {DISCOVERY.DEVICE.STATUS}
        デバイス稼働時間: {DISCOVERY.DEVICE.UPTIME}

        デバイスサービス名: {DISCOVERY.SERVICE.NAME}
        デバイスサービスポート: {DISCOVERY.SERVICE.PORT}
        デバイスサービスステータス: {DISCOVERY.SERVICE.STATUS}
        デバイスサービス稼働時間: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: 'Autoregistration: {HOST.HOST}'
      message: |
        ホスト名: {HOST.HOST}
        ホスト IP: {HOST.IP}
        エージェントポート: {HOST.PORT}
```

エクスポートされた要素

エクスポートされた要素については、以下の表で説明します。

要素	型	説明
name	string	(必須) メディアタイプ名。
type	string	(必須) メディアタイプで使用されるトランスポート。 可能な値:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4)。
status	string	メディアタイプが有効かどうか。 可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)
max_sessions	integer	並列で処理できるアラートの最大数。 SMS の可能な値:[1] 1 (デフォルト)。 他のメディアタイプの可能な値:[1] 0-100 (0 は無制限)。
attempts	integer	アラート送信の最大試行回数。 可能な値:[1] 1-10 (デフォルト: 3)。
attempt_interval	string	再試行間隔 (秒または時間のサフィックスを使用)。 可能な値:[1] 0-60s (デフォルト: 10s)。
description	string	メディアタイプの説明。
message_templates		メディアタイプのメッセージテンプレートのルート要素。
event_source	string	(必須) イベントソース。 可能な値:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4)。

要素	型	説明
operation_mode	string	動作モード。 可能な値:[1] PROBLEM (0), RECOVERY (1), UPDATE (2)。
subject	string	メッセージの件名。
message	string	メッセージ本文。

Note:
参照: メディアタイプオブジェクト（同名のプロパティを参照）。

Email

Email メディアタイプに対してのみ、以下の追加要素がエクスポートされます。

要素	型	説明
provider	string	メールプロバイダー。
smtp_server	string	SMTP サーバー。
smtp_port	integer	接続する SMTP サーバーポート。 デフォルト: 25。
smtp_helo	string	SMTP helo。
smtp_email	string	通知を送信するメールアドレス。
smtp_security	string	使用する SMTP 接続のセキュリティレベル。 指定可能な値:[1] NONE (0, デフォルト), STARTTLS (1), SSL_OR_TLS (2)。
smtp_verify_host	string	SMTP の SSL verify host。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
smtp_verify_peer	string	SMTP の SSL verify peer。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
smtp_authentication	string	使用する SMTP 認証方式。 指定可能な値:[1] NONE (0, デフォルト), PASSWORD (1), OAUTH (2)。
username	string	ユーザー名。
password	string	認証パスワード。
redirection_url	string	OAuth 認証後にリダイレクトする Zabbix フロントエンドの URL。
client_id	string	OAuth 認可サーバーに登録されたクライアント識別子。
authorization_url	string	アクセストークンとリフレッシュトークンを取得するためのパラメータ付き OAuth URL。
token_url	string	認可トークンをアクセストークンとリフレッシュトークンに交換するための OAuth URL。
message_format	string	メッセージフォーマット。 指定可能な値:[1] TEXT (0), HTML (1, デフォルト)。

Note:
参照: メディアタイプオブジェクト（同名の関連プロパティを参照）。

SMS

SMS メディアタイプに対してのみ、以下の追加要素がエクスポートされます。

要素	タイプ	説明
gsm_modem	string	(必須) GSM モデムのシリアルデバイス名

Note:
参照: メディアタイプオブジェクト（同名のプロパティを参照）。

スクリプト

スクリプトメディアタイプの場合、次の追加要素のみがエクスポートされます。

要素		タイプ	説明
スクリプト名		string	(必須) スクリプト名。
パラメータ			スクリプトパラメータのルート要素。
	sortorder	string	(必須) コマンドライン引数としてスクリプトに渡されるスクリプトパラメータの順序。
	value	string	スクリプトパラメータの値。

Note:
参照: メディアタイプオブジェクト（同名のプロパティを参照）。

Webhook

Webhook メディアタイプに対してのみ、以下の追加要素がエクスポートされます。

要素	型	説明
script	文字列	スクリプト。
timeout	文字列	Javascript スクリプトの HTTP リクエストタイムアウト間隔。 可能な値:[1] 1-60s (デフォルト: 30s)。
process_tags	文字列	返されたタグを処理するかどうか。 可能な値:[1] NO (0, デフォルト), YES (1)。
show_event_menu	文字列	`event_menu_url` および `event_menu_name` フィールドで {EVENT.TAGS.*} マクロが正常に解決された場合、イベントメニューにエントリが存在することを示します。 可能な値:[1] NO (0, デフォルト), YES (1)。
event_menu_url	文字列	イベントメニューエントリの URL。{EVENT.TAGS.*} マクロをサポート。
event_menu_name	文字列	イベントメニューエントリの名前。{EVENT.TAGS.*} マクロをサポート。
parameters		Webhook メディアタイプパラメータのルート要素。
name	文字列	(必須) Webhook パラメータ名。
value	文字列	Webhook パラメータ値。

Note:
参照: メディアタイプオブジェクト（同名の関連プロパティを参照）。

脚注

[1] 括弧内の API 整数値（例：ENABLED（0））は、参照用にのみ記載されています。詳細については、表のエントリまたは各セクションの最後にあるリンクされた API オブジェクトページを参照してください。

13 ディスカバリ

サイドバーを使用して、ディスカバリセクションのコンテンツにアクセスしてください。

1 ネットワークディスカバリ

概要

Zabbix は、自動ネットワークディスカバリ機能を提供しており、効果的かつ非常に柔軟です。

ネットワークディスカバリを適切に設定することで、以下のことが可能になります。

- Zabbix の導入を迅速化
- 管理の簡素化
- 過度な管理作業をせずに、急速に変化する環境で Zabbix を利用

Zabbix のネットワークディスカバリは、以下の情報に基づいています。

- IP レンジ
- 外部サービスの可用性 (FTP、SSH、WEB、POP3、IMAP、TCP など)
- Zabbix エージェントから受信した情報 (暗号化されていないモードのみサポート)

- SNMP エージェントから受信した情報

以下は提供しません。

- ネットワークトポロジーのディスカバリ

ネットワークディスカバリは基本的に、ディスカバリとアクションの 2 つのフェーズで構成されます。

ディスカバリ

Zabbix は、ネットワークディスカバリルールで定義された IP レンジを定期的にスキャンします。チェックの頻度は、各ルールごとに個別に設定できます。

各ルールには、IP レンジに対して実行するサービスチェックのセットが定義されています。

ディスカバリルールはディスカバリマネージャによって処理されます。ディスカバリマネージャは、各ルールごとにタスク（ネットワークチェック）のリストを持つジョブを作成します。ネットワークチェックは、利用可能なディスカバリワーカー（数はフロントエンドで各ルールごとに設定可能）によって並列に実行されます。同じ IP とポートのチェックのみが順次スケジューリングされます。なぜなら、一部のデバイスは同じポートへの並列接続を受け付けないためです。

ネットワークチェックのキューサイズは、約 2000000 または 4GB のメモリに制限されています。キューがいっぱいになると、そのディスカバリルールはスキップされ、警告メッセージがログに出力されます。キュー内のディスカバリチェック数を監視するには、`zabbix[discovery_queue]` 内部アイテムを使用できます。

ディスカバリチェックは他のチェックとは独立して処理されます。いずれかのチェックでサービスが見つからない（または失敗する）場合でも、他のチェックは引き続き処理されます。

> **Note:**
> ディスカバリルールが実行中に変更された場合、現在のディスカバリ実行は中止されます。

ネットワークディスカバリモジュールによって実行されるサービスおよびホスト（IP）の各チェックは、ディスカバリエベントを生成します。

イベント	サービスチェックの結果
サービス検出	サービスが「ダウン」から「アップ」になった場合、または初めて検出された場合。
サービスアップ	サービスがすでに「アップ」だった後に「アップ」である場合。
サービスロスト	サービスが「アップ」から「ダウン」になった場合。
サービスダウン	サービスがすでに「ダウン」だった後に「ダウン」である場合。
ホスト検出	ホストのすべてのサービスが「ダウン」だった後に少なくとも 1 つのサービスが「アップ」になった場合、または未登録ホストに属するサービスが検出された場合。
ホストアップ	ホストの少なくとも 1 つのサービスがすでに「アップ」だった後に少なくとも 1 つのサービスが「アップ」である場合。
ホストロスト	ホストのすべてのサービスが少なくとも 1 つが「アップ」だった後に「ダウン」になった場合。
ホストダウン	ホストのすべてのサービスがすでに「ダウン」だった後に「ダウン」である場合。

アクション

ディスカバリーイベントは、以下のような関連するアクションの基礎となります。

- 通知の送信
- ホストの追加/削除
- ホストの有効化/無効化
- ホストをグループに追加
- ホストをグループから削除
- ホストにタグを追加
- ホストからタグを削除
- テンプレートをホストにリンク/テンプレートをホストからリンク解除
- リモートスクリプトの実行

これらのアクションは、デバイスタイプ、IP、ステータス、稼働/停止時間などに応じて設定できます。ネットワークディスカバリーベースのイベントに対するアクションの設定の詳細については、アクションのオペレーションおよび条件のページを参照してください。

ネットワークディスカバリーアクションはイベントベースであるため、検出されたホストがオンラインの場合もオフラインの場合もトリガーされます。*Add host* のようなアクションが *Service Lost/Service Down* イベントでトリガーされるのを避けるために、アクション条件として *Discovery status: up* を追加することを強く推奨します。そうしないと、検出されたホストが手動で削除された場合でも *Service Lost/Service Down* イベントが生成され、次のディスカバリーサイクルで再作成されてしまいます。

Note:
検出されたホストにテンプレートをリンクする場合、リンク可能なテンプレートのいずれかに、ホストまたは他のリンク可能なテンプレートに既に存在する一意のエンティティ（例：アイテムキー）と同じ一意のエンティティ（例：アイテムキー）がある場合、テンプレートのリンクはまとめて失敗します。

ホストの作成

ホストの追加操作が選択されている場合、ホストが追加されます。ホストの追加操作がなくても、ホストに対するアクションを伴う操作を選択した場合もホストが追加されます。そのような操作は以下の通りです:

- ホストの有効化
- ホストの無効化
- ホストをホストグループに追加
- テンプレートをホストにリンク

作成されたホストは、検出されたホストグループに追加されます（デフォルトでは管理 > 一般 > その他で設定可能）。ホストを別のグループに追加したい場合は、ホストグループから削除操作（" 検出されたホスト " を指定）とホストグループに追加操作（別のホストグループを指定）を追加してください。ホストは必ずホストグループに所属する必要があるためです。

検出されたデバイスの IP アドレスは、検出元（Zabbix サーバー、Zabbix プロキシ、またはプロキシグループ）およびインターフェイスタイプとともに、システム内でホストを検索する基準として使用されます。同じ IP アドレス、インターフェイスタイプ、検出元を持つホストがすでに存在する場合、そのホストが操作の対象となります。検出元が異なる場合、検出されたエンティティは別のホストとして扱われ、新しいホストが作成される場合があります。

検出されたホストの IP アドレスが変更されたり、インターフェイスが削除された場合、次回の検出時に新しいホストが作成されます。

ホスト名の付け方

ホストを追加する際、ホスト名はリバース DNS ルックアップの結果、またはリバースルックアップが失敗した場合は IP アドレスとなります。ルックアップは、ディスカバリを実行している Zabbix サーバーまたは Zabbix プロキシから実行されます。プロキシでルックアップに失敗した場合、サーバーで再試行されることはありません。そのような名前のホストがすでに存在する場合、次のホストには名前の末尾に **_2** が付加され、さらに **_3** と続きます。

DNS/IP ルックアップを上書きし、ホスト名にアイテム値を使用することも可能です。例えば:

- ディスカバリ用の Zabbix エージェントアイテムを使用して、Zabbix エージェントが稼働している複数のサーバーをディスカバリし、このアイテムが返す文字列値に基づいて自動的に適切な名前を割り当てることができます
- ディスカバリ用の SNMP エージェントアイテムを使用して、複数の SNMP ネットワークデバイスをディスカバリし、このアイテムが返す文字列値に基づいて自動的に適切な名前を割り当てることができます

ホスト名がアイテム値で設定されている場合、以降のディスカバリチェックでは更新されません。アイテム値でホスト名を設定できない場合は、デフォルト値 (DNS 名) が使用されます。

ディスカバリされた IP アドレスで既にホストが存在し、ディスカバリ元 (Zabbix サーバー、プロキシまたはプロキシグループ) が変更されていない場合、新しいホストは作成されません。ディスカバリ元が異なる場合、ディスカバリされたエンティティは別物として扱われ、新しいホストが作成される場合があります。ただし、ディスカバリアクションに操作 (テンプレートのリンク、ホストグループへの追加など) が含まれている場合は、IP アドレス、インターフェイスタイプ、ディスカバリ元で一致する既存のホストに対して実行されます。

ホストの削除

ネットワークディスカバリールールによって検出されたホストは、検出されたエンティティがルールの IP 範囲に存在しなくなった場合、監視データ > ディスカバリから自動的に削除されます。ホストは即座に削除されます。

ホスト追加時のインターフェース作成

ネットワークディスカバリの結果としてホストが追加される場合、以下のルールに従ってインターフェースが作成されます。

- 検出されたサービス - 例えば、SNMP チェックが成功した場合、SNMP インターフェースが作成されます。
- ホストが Zabbix エージェントと SNMP の両方のリクエストに応答した場合、両方のタイプのインターフェースが作成されます。
- 一意性の基準が Zabbix エージェントまたは SNMP から返されたデータである場合、ホストに対して最初に見つかったインターフェースがデフォルトのものとして作成されます。他の IP アドレスは追加のインターフェースとして追加されます。アクションの条件（ホスト IP など）は、インターフェースの追加には影響しません。注これは、すべてのインターフェースが同じディスカバリルールによって検出された場合に機能します。異なるディスカバリルールが同じホストの異なるインターフェースを検出した場合、追加のホストが追加されます。
- ホストがエージェントチェックのみに応答した場合、エージェントインターフェースのみで作成されます。後で SNMP に応答し始めた場合は、追加の SNMP インターフェースが追加されます。
- 最初に「IP」一意性基準で検出された 3 つの個別のホストが作成され、その後ディスカバリルールが変更されてホスト A、B、C が同一の一意性基準の結果を持つようになった場合、B と C は A（最初のホスト）の追加インターフェースとして作成されま

す。個別のホスト B と C は残ります。監視 > ディスカバリでは、追加されたインターフェースは「検出されたデバイス」列に黒字でインデントされて表示されますが、「監視対象ホスト」列には最初に作成されたホスト A のみが表示されます。「稼働/停止時間」は、追加のインターフェースと見なされる IP には測定されません。

プロキシ設定の変更

異なるプロキシによって検出されたホストは、常に異なるホストとして扱われるわけではありません。ディスカバリと一意性のチェックは、プロキシグループの構造に依存します。プロキシがディスカバリルールを実行してホストを作成すると、そのホストはプロキシ自体ではなく、プロキシの親プロキシグループに追加されます。Zabbix がディスカバリ中に IP の一意性を評価する際には、親プロキシグループによって監視されているホストをチェックします。そのグループ内の個々のプロキシによって監視されているホスト（ディスカバリを実行したプロキシを含む）は、一意性チェックの対象外となるため、複数のプロキシが重複するサブネットを監視している場合、重複したホストが作成される可能性があります。

この動作により、異なるサブネットで使用される重複する IP 範囲にまたがってディスカバリを実行できますが、すでに監視されているサブネットに割り当てられているプロキシを変更する場合は、重複を避けるために、ディスカバリされたホストと親プロキシグループのメンバーシップの両方に一貫してプロキシの変更を適用する必要があるため、より複雑になります。

たとえば、ディスカバリルールでプロキシを置き換える手順は次のとおりです。

1. ディスカバリルールを無効にする
2. プロキシの設定を同期する
3. ディスカバリルールのプロキシを置き換える
4. このルールで検出されたすべてのホストのプロキシを置き換える（親プロキシグループ内のホストと、そのグループ内の個々のプロキシによって監視されているホストが重複しないように更新されていることを確認する）
5. ディスカバリルールを有効にする

1 ネットワークディスカバリルールの設定

概要

Zabbix がホストやサービスを検出するために使用するネットワークディスカバリールールを設定するには、以下の手順を実行します。

- データ収集 → ディスカバリーに移動します
- ディスカバリールールの作成をクリックします（または既存のルールを編集する場合はルール名をクリックします）
- ディスカバリールールの属性を編集します

ルール属性

必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	ルールの一意な名前。例: ”Local network”
検出方法	検出を実行するのは: サーバー - Zabbix サーバー プロキシ - Zabbix プロキシ (プロキシ名フィールドで選択)
IP 範囲	検出対象の IP アドレス範囲。以下の形式が使用可能: 単一 IP: 192.168.1.33 IP アドレスの範囲: 192.168.1-10.1-255。範囲はカバーされるアドレスの総数 (64K 未満) で制限されます。 IP マスク: 192.168.4.0/24 サポートされる IP マスク: /16 - /30 (IPv4 アドレス用) /112 - /128 (IPv6 アドレス用) リスト: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 このフィールドはスペース、タブ、複数行をサポートします。
更新間隔	このパラメータは、Zabbix がルールを実行する頻度を定義します。 間隔は、前回の検出インスタンスの実行終了後に測定されるため、重複はありません。 時間のサフィックスがサポートされています (例: 30s, 1m, 2h, 1d)。 ユーザーマクロがサポートされています。 注意: ユーザーマクロを使用していて、その値が変更された場合 (例: 1w → 1h)、次のチェックは以前の値 (例の場合はかなり先の未来) に従って実行されます。

パラメータ	説明
タイプごとの最大同時チェック数	検出チェックの並列処理のため、サービスチェックごとの検出スレッド (ワーカー) の最大数を設定します: **1** つ - 1 スレッド 無制限 - 無制限のスレッド数 (ただしStartDiscoverersパラメータで指定された数を超えません) カスタム - カスタムのスレッド数を設定 (0-999) SNMPv3 非同期サービスチェックを持つすべての検出ルールは libsnmp 実装の特性により 1 つのワーカーで処理されるため、ワーカー数を増やしても検出速度は向上しません。
チェック	Zabbix はこのチェックリストを使って検出を行います。をクリックすると、ポップアップウィンドウで新しいチェックを設定できます。 サポートされるチェック: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping。 プロトコルベースの検出は、SNMP 以外は **net.tcp.service[]** 機能を使って各ホストをテストします。SNMP は SNMP OID をクエリします。Zabbix agent は暗号化なしでアイテムをクエリしてテストします。詳細はエージェントアイテムを参照してください。 ' ポート ' パラメータは以下のいずれか: 単一ポート: 22 ポート範囲: 22-45 リスト: 22-45,55,60-70 Zabbix 7.0 以降、すべてのサービスチェックは LDAP チェックを除き非同期で実行されます。 Zabbix 7.0 以降、HTTP/HTTPS チェックは libcurl 経由で行われます。Zabbix サーバー/プロキシが libcurl なしでコンパイルされている場合、HTTP チェックは従来通り (TCP チェックとして) 動作しますが、HTTPS チェックは動作しません。
デバイスの一意性基準	一意性基準は以下のいずれか: **IP** アドレス - 単一 IP デバイスを複数処理しません。同じ IP のデバイスが既に存在する場合、それは既に検出済みとみなされ、新しいホストは追加されません。 **<** 検出チェック **>** - Zabbix agent または SNMP agent チェック。 検出時に使用される一意性基準は、アクション実行時のシステム内のホスト識別とは異なることに注意してください。検出時の一意性基準は、検出された 2 つ以上のデバイスが同一かどうかを定義しますが、Zabbix でのホスト識別の基準は IP アドレスのみです (ホスト作成参照)。
ホスト名	作成されるホストの技術的なホスト名を以下で設定: **DNS** 名 - DNS 名 (デフォルト) **IP** アドレス - IP アドレス **<** 検出チェック **>** - 検出チェックで受信した文字列値 (例: Zabbix agent, SNMP agent チェック) 詳細はホスト名付与も参照。
表示名	作成されるホストの表示名を以下で設定: ホスト名 - 技術的なホスト名 (デフォルト) **DNS** 名 - DNS 名 **IP** アドレス - IP アドレス **<** 検出チェック **>** - 検出チェックで受信した文字列値 (例: Zabbix agent, SNMP agent チェック) 詳細はホスト名付与も参照。
有効	チェックボックスがオンの場合、ルールは有効で Zabbix サーバーによって実行されます。 オフの場合、ルールは無効で実行されません。

ファイルディスクリプタの制限を超える

多数の同時チェックが発生する場合、ディスカバリマネージャのファイルディスクリプタ制限を使い果たす可能性があります。

ディスカバリに必要なファイルディスクリプタの数は、ディスカバリワーカー数 * 1000 です。デフォルトでは、ディスカバリワーカーは 5 つですが、システムのソフトリミットは約 1024 です。

この制限に近づくと、Zabbix は各ワーカーごとにタイプごとの同時チェック数のデフォルト値を減らし、ログファイルに警告を書き込みます。ただし、ユーザーがタイプごとの最大同時チェック数に Zabbix が計算した値よりも大きい値を設定している場合、Zabbix は 1 つのワーカーに対してユーザー定義値を使用します。

実際に使用するシナリオ

この例では、ローカルネットワークの 192.168.1.1～192.168.1.254 の範囲の IP アドレスをチェックするネットワークディスカバリを設定します。

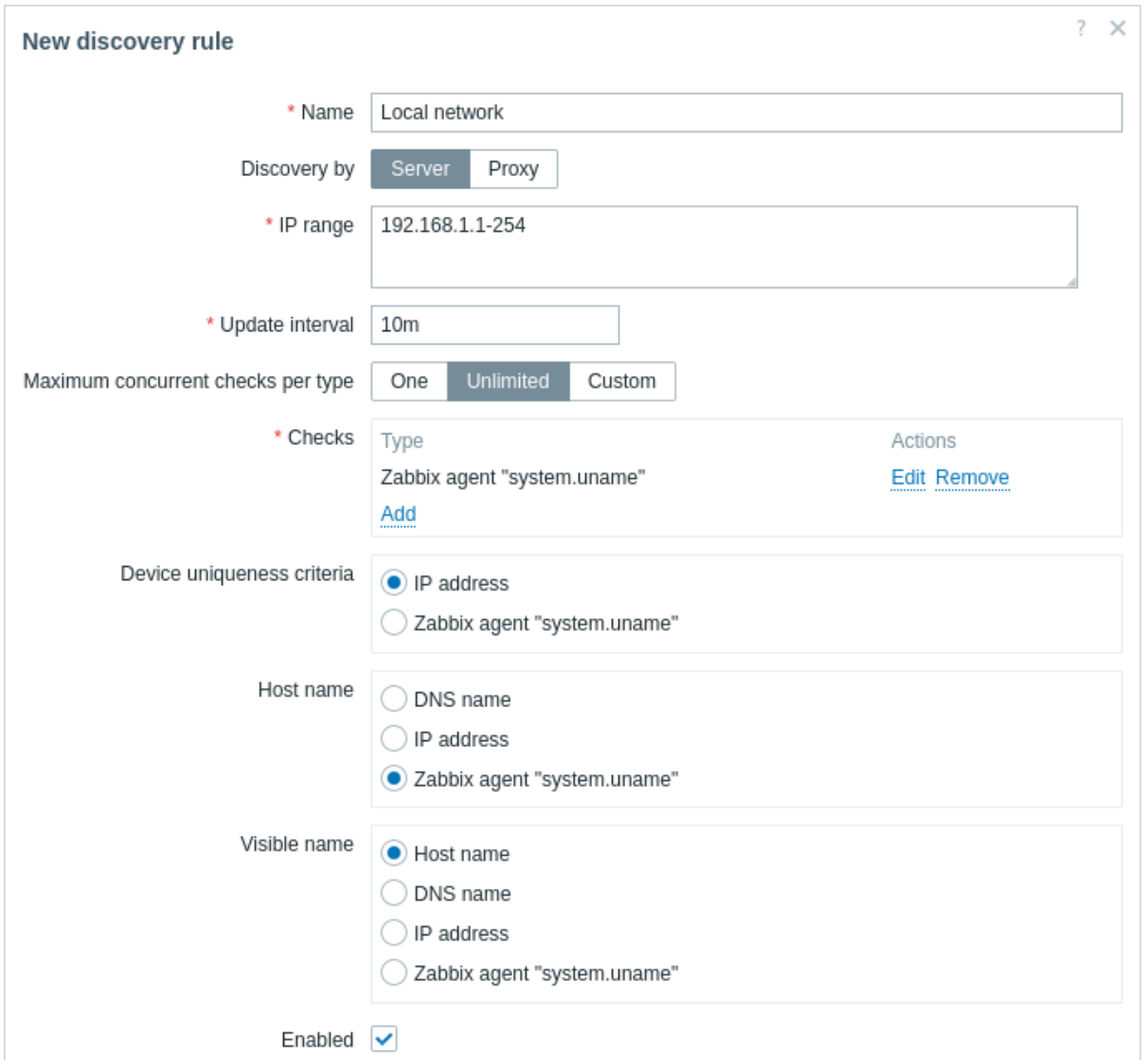
このシナリオでは、以下のことをおこなう設定をします：

- Zabbix エージェントが動作しているホストを検出
- 検出は 10 分間隔で実行
- アップタイムが 1 時間以上のホストは監視対象に追加

- ダウンタイムが 24 時間以上のホストは監視対象から削除
- Linux ホストはグループ「Linux servers」に追加
- Windows ホストはグループ「Windows servers」に追加
- Linux ホストではテンプレート *Linux* を使用
- Windows ホストではテンプレート *Windows* を使用

ステップ 1

IP 範囲のネットワークディスカバリールールを定義します。



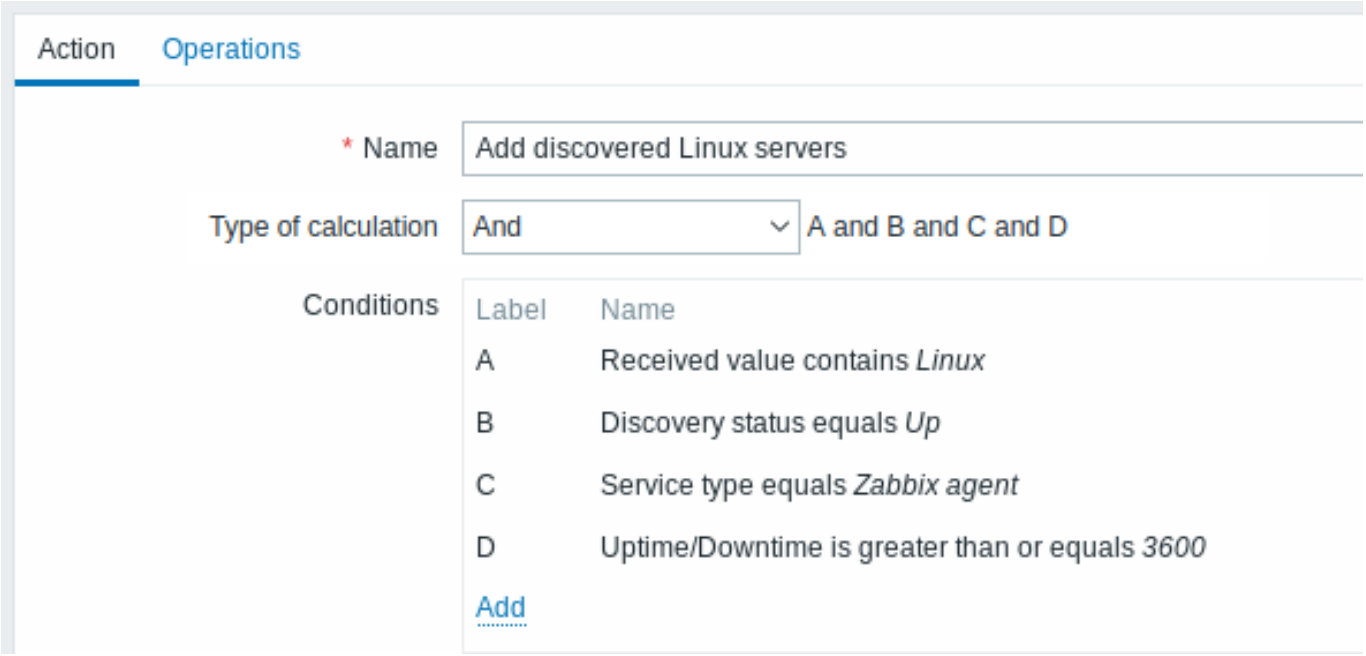
Zabbix は、192.168.1.1-192.168.1.254 の IP 範囲で Zabbix エージェントに接続し、**system.uname** キーの値を取得することでホストを検出しようとします。エージェントから受信した値は、ホストの名前付けや、異なるオペレーティングシステムごとに異なるアクションを適用するために使用できます。たとえば、Windows サーバーにはテンプレート *Windows* を、Linux サーバーにはテンプレート *Linux* をリンクします。

このルールは 10 分ごとに実行されます。

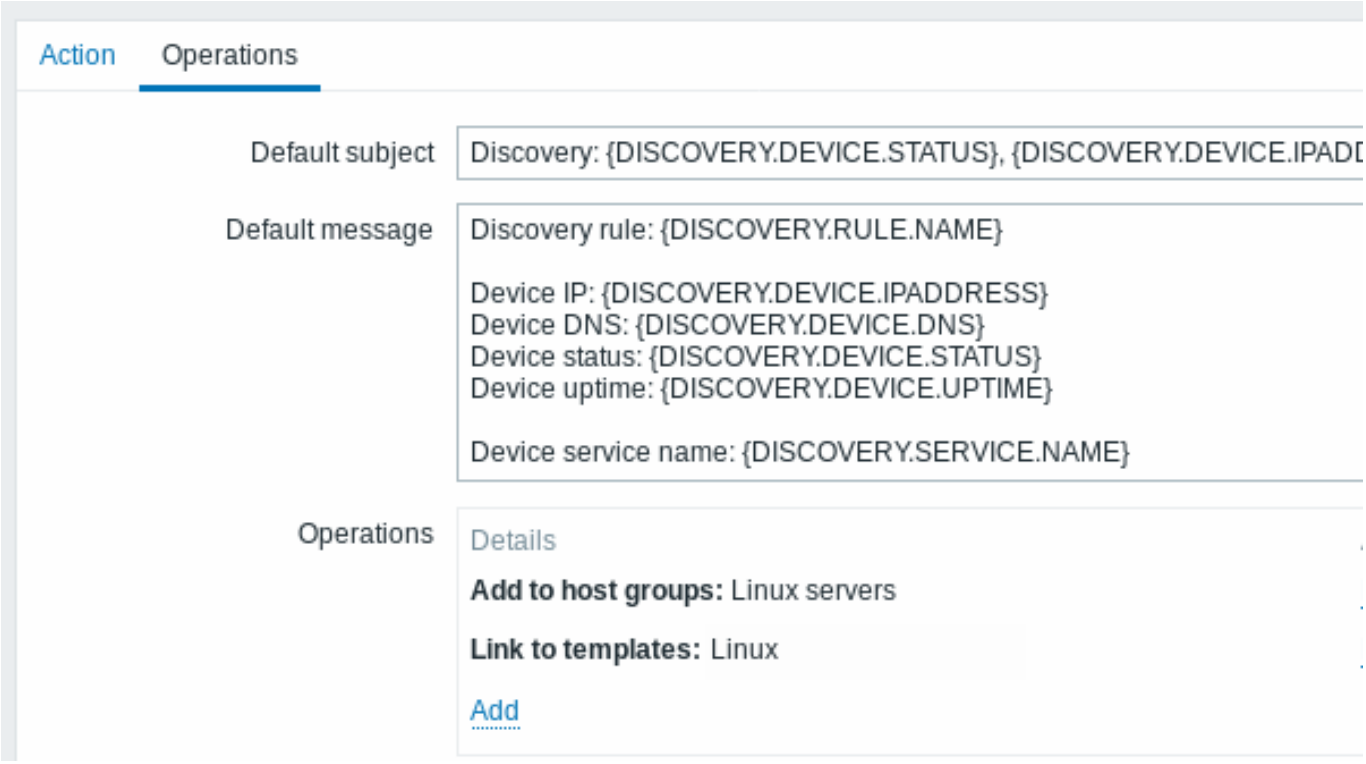
このルールが追加されると、Zabbix は自動的にディスカバリーを開始し、さらなる処理のためにディスカバリーベースのイベントを生成します。

ステップ 2

新たに検出された Linux サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。



このアクションは以下のコンディションが成立する場合有効になります。- サービス”Zabbix agent” が” 動作” している。- system.uname （ルールを定義する際に使用した Zabbix エージェントのキー）の値に”Linux” という文字列が含まれる。- アップタイムが 1 時間（3600 秒）を超えている。

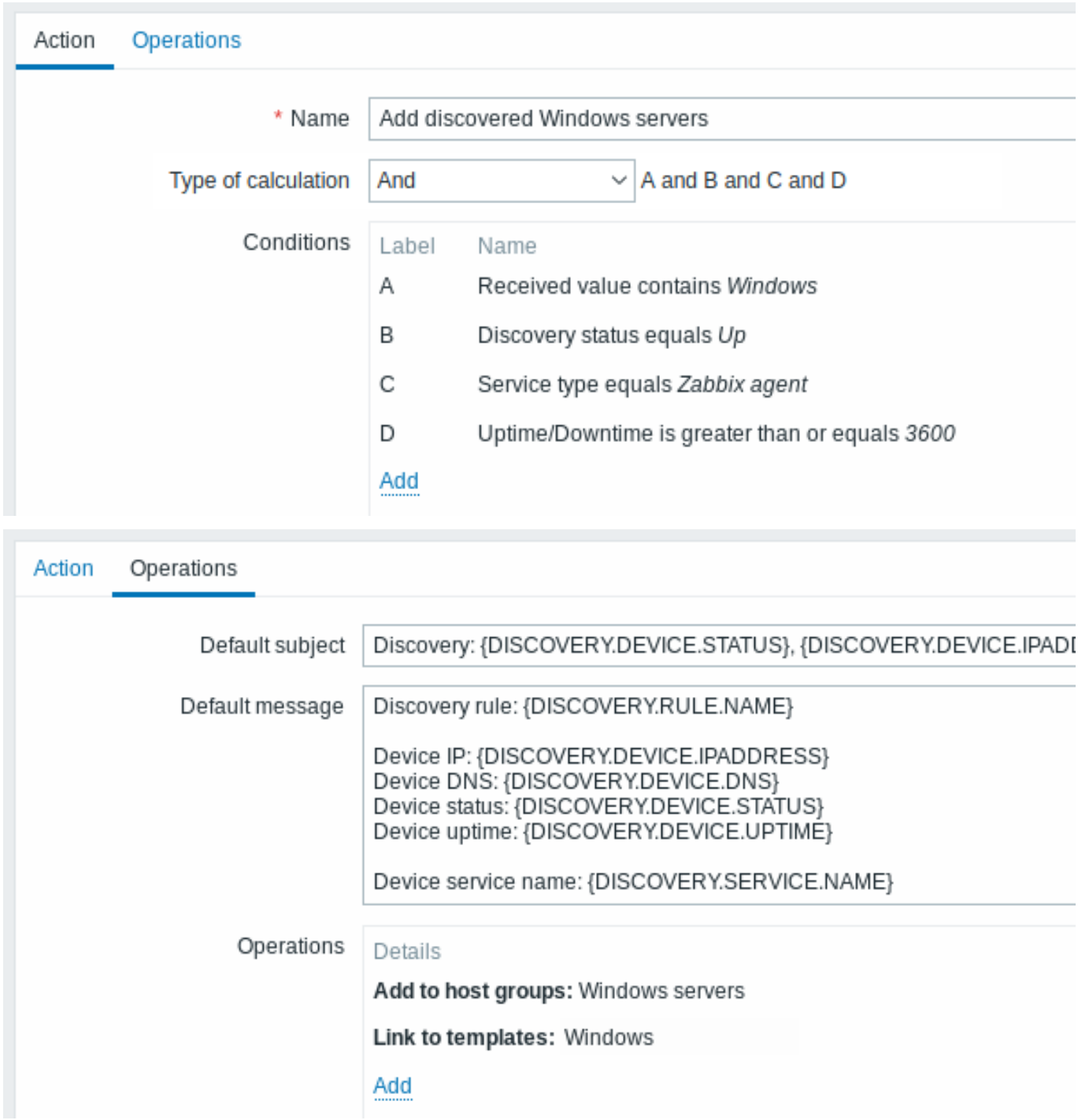


アクションによって以下のオペレーションが実行されます。

- 新たに検出したホストをグループ”Linux servers” に追加する。（まだ追加されていないホストの場合はホストの追加も実行する）
- ホストをテンプレート”Linux” にリンクする。”Linux” テンプレートに含まれるアイテムとトリガーを使用してホストの監視が自動的に開始される。

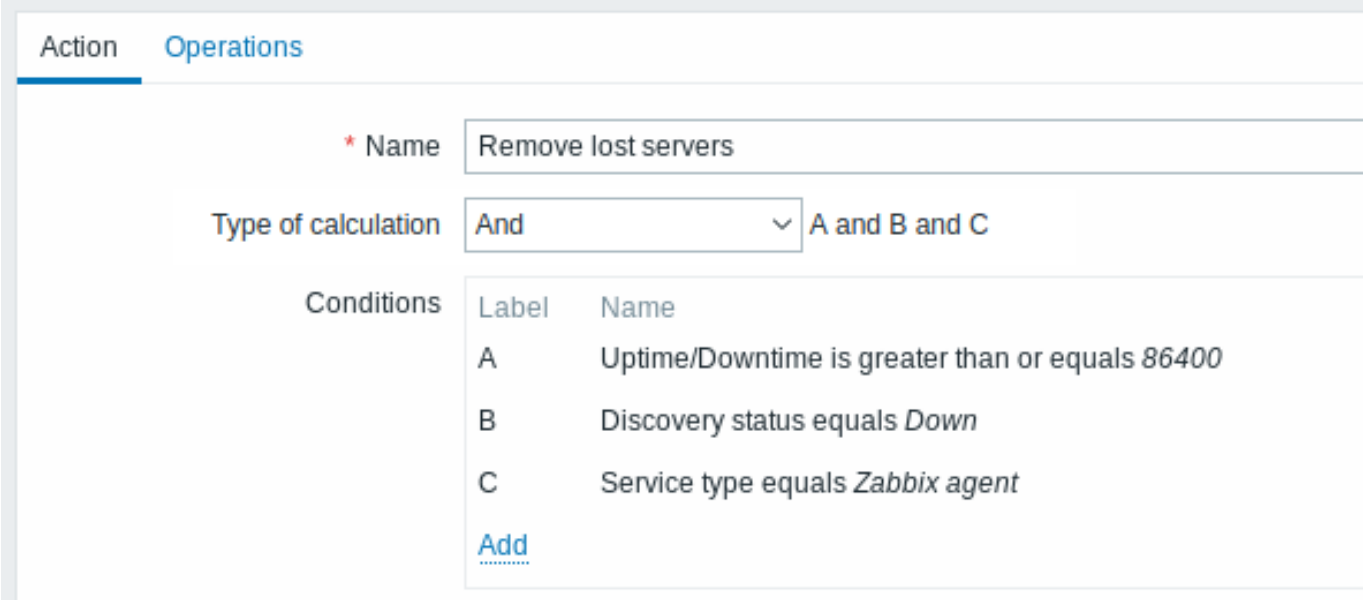
ステップ 3

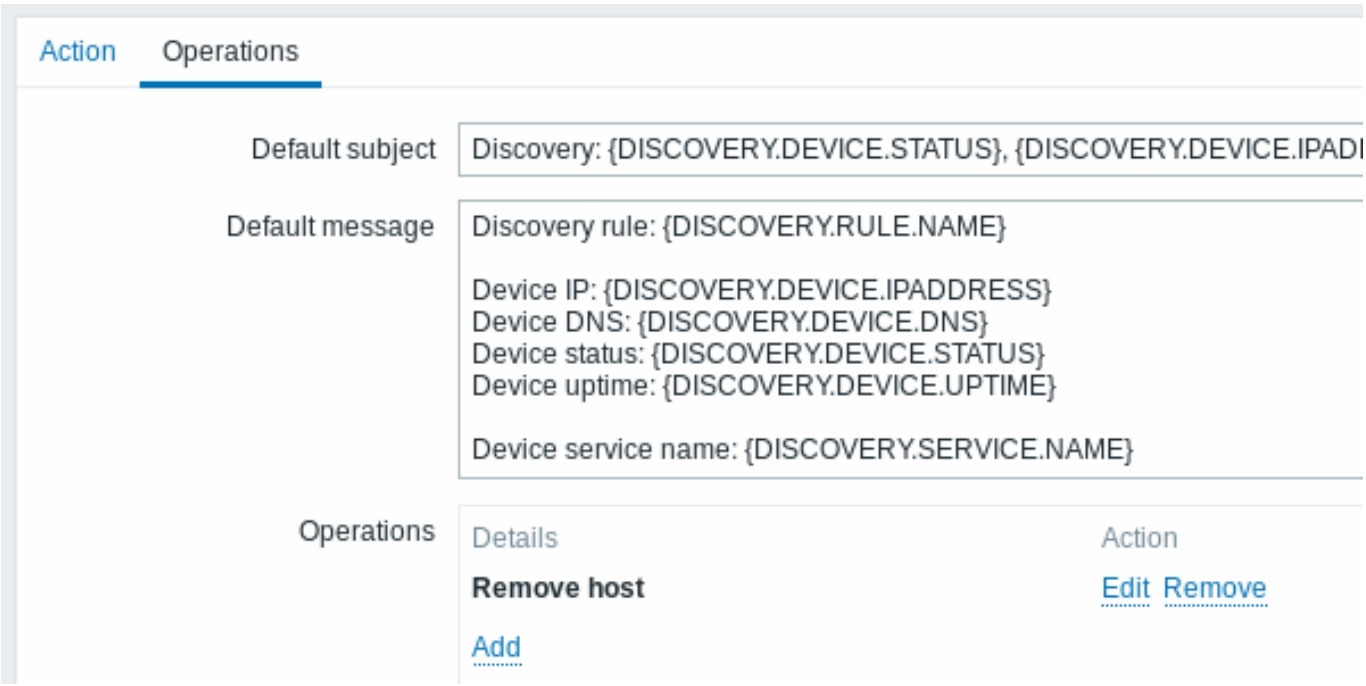
新たに検出された Windows サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。



ステップ 4

動作していないサーバーを削除するための検出アクションを定義します。





”Zabbix agent” サービスが 24 時間（86400 秒）以上 down している場合、サーバーは削除されます。A server will be removed if ”Zabbix agent” service is ’down’ for more than 24 hours (86400 seconds).

2 アクティブエージェントの自動登録

概要

アクティブ Zabbix エージェントの自動登録を許可することができ、その後サーバーはそれらの監視を開始できます。この方法では、新しいホストをサーバー上で手動で設定することなく監視に追加できます。

自動登録は、以前に知られていなかったアクティブエージェントがチェックを要求したときに発生する可能性があります。

この機能は、新しいクラウドノードの自動監視に非常に便利です。クラウドに新しいノードが追加されると、Zabbix は自動的にそのホストのパフォーマンスおよび可用性データの収集を開始します。

アクティブエージェントの自動登録は、パッシブチェックによる追加ホストの監視もサポートしています。アクティブエージェントがチェックを要求する際、設定ファイルで「ListenIP」または「ListenPort」構成パラメータが定義されていれば、それらもサーバーに送信されます。(複数の IP アドレスが指定されている場合は、最初のものがサーバーに送信されます。)

サーバーは、新しく自動登録されたホストを追加する際、受信した IP アドレスとポートを使用してエージェントを設定します。IP アドレス値が受信されない場合は、着信接続に使用されたものが使用されます。ポート値が受信されない場合は、10050 が使用されます。

ホストがDNS 名でデフォルトのエージェントインターフェースとして自動登録されるように指定することも可能です。

自動登録は以下の場合に再実行されます:

- ホストのメタデータ情報が変更された場合:
  - HostMetadata が変更され、エージェントが再起動された場合
  - HostMetadataItem によって返される値が変更された場合
- メタデータが欠落している手動作成ホストの場合
- ホストが手動で別の Zabbix プロキシによって監視されるように変更された場合
- 同じホストの自動登録が新しい Zabbix プロキシから来た場合

Zabbix サーバーおよび Zabbix プロキシのアクティブエージェント自動登録のハートビートは 120 秒です。そのため、検出されたホストが削除された場合、自動登録は 120 秒後に再実行されます。

設定

サーバーを指定する

エージェント設定ファイル - zabbix_agentd.conf で Zabbix サーバーが識別されていることを確認してください

```
ServerActive=10.0.0.1
```

zabbix_agentd.conf でホスト名を明確に定義しない限り、エージェントの場所のシステムホスト名がサーバーによってホストの命名に使用されます。Linux のシステムホスト名は `hostname` コマンドを実行することで取得できます。

ホスト名が Zabbix エージェント設定でホストのコンマ区切りリストとして定義されている場合、リストされているすべてのホスト名に対してホストが作成されます。

設定ファイル変更後、エージェントを再起動します。

アクティブエージェントの自動登録のためのアクション

サーバーがエージェントから自動登録リクエストを受信すると、アクションを呼び出します。エージェントの自動登録には、イベントソース「自動登録」のアクションを設定する必要があります。

> **Note:**
> アクティブエージェントを自動登録するためにネットワークディスカバリを設定する必要はありません。

Zabbix フロントエンドで、アラート → アクションに移動し、自動登録アクションを選択してアクションの作成をクリックします:

- アクションタブで、アクションに名前を付けます
- 必要に応じて、条件を指定します。ホスト名/ホストメタデータの条件で部分一致や正規表現一致が可能です。「ホストメタデータ」条件を使用する場合は、次のセクションを参照してください。
- オペレーションタブで、' ホストの追加'、' ホストグループへの追加'（例：検出されたホスト）、' テンプレートのリンク' など、関連するオペレーションを追加します。

> **Note:**
> 自動登録されるホストがアクティブ監視専用でサポートされる可能性が高い場合（たとえば、Zabbix サーバーからファイアウォールで保護されているホストなど）、*Template_Linux-active* のような特定のテンプレートを作成してリンクすることを検討してください。

作成されたホストは検出されたホストグループに追加されます（デフォルト、管理 → 一般 → その他で設定可能）。ホストを別のグループに追加したい場合は、ホストグループから削除オペレーション（「検出されたホスト」を指定）とホストグループへの追加オペレーション（別のホストグループを指定）を追加してください。ホストは必ずホストグループに所属する必要があります。

セキュアな自動登録

暗号化された接続で PSK ベースの認証を設定することで、自動登録をセキュアに行うことができます。

暗号化のレベルは、管理 → 一般 → 自動登録 でグローバルに設定します。暗号化なし、PSK 認証による TLS 暗号化、または両方を選択できます（暗号化なしで登録するホストと、暗号化で登録するホストの両方を許可できます）。

PSK による認証は、ホストを追加する前に Zabbix サーバーによって検証されます。認証に成功すると、ホストが追加され、ホストから/への接続 はグローバルな自動登録設定と同じ ID/事前共有キーで「PSK」のみに設定されます。

> **Attention:**
> プロキシを使用するインストールで自動登録のセキュリティを確保するには、Zabbix サーバーとプロキシ間の暗号化を有効にする必要があります。

デフォルトインターフェースとして DNS を使用する

HostInterface および HostInterfaceItem設定パラメータを使用すると、自動登録時にホストインターフェースのカスタム値を指定できます。

より具体的には、ホストを IP アドレスではなく DNS 名をデフォルトのエージェントインターフェースとして自動登録したい場合に便利です。その場合、DNS 名を HostInterface または HostInterfaceItem パラメータの値として指定または返す必要があります。これらのパラメータのいずれかの値が変更された場合（たとえば、IP アドレスから DNS 名、またはその逆）、自動登録されたホストのデフォルトインターフェースもそれに応じて更新されます。この更新は既存のホストに適用され、新しいホストが作成されるわけではありません。新しい値を送信するには、エージェントを再起動して自動登録プロセスを再実行する必要があります。

Note:
HostInterface または HostInterfaceItem パラメータが設定されていない場合は、代わりに listen_dns パラメータが使用されます。この値は、エージェントの IP アドレスの逆引き DNS ルックアップを実行することで決定されます。逆引き DNS が正しく設定されていない場合や無効な名前を返す場合、無効なインターフェース値が原因で自動登録が正しく行われない、または失敗する可能性があります。

ホストメタデータの使用

エージェントがサーバーに自動登録リクエストを送信する際、ホスト名を送信します。場合によっては（例えば、Amazon クラウドノードなど）、Zabbix サーバーが検出したホストを区別するためにはホスト名だけでは不十分です。ホストメタデータを使用することで、エージェントからサーバーに他の情報を送信することができます。

ホストメタデータは、エージェントの設定ファイル（zabbix_agentd.conf）で設定します。設定ファイルでホストメタデータを指定する方法は 2 つあります。

```
HostMetadata
HostMetadataItem
```

各オプションの説明については、上記リンクを参照してください。

HostMetadataItem パラメータは最大 65535 の UTF-8 コードポイントを返すことができます。これより長い値は切り捨てられます。

MySQL の場合、返される値にマルチバイト文字が含まれていると、実際の最大文字数は少なくなります。例えば、3 バイト文字のみを含む値の場合、最大 21844 文字、4 バイト文字のみを含む値の場合は最大 16383 文字に制限されます。

Attention:
自動登録の試行は、アクティブエージェントがアクティブチェックの更新リクエストをサーバーに送信するたびに行われます。リクエスト間の遅延は、エージェントのRefreshActiveChecksパラメータで指定します。最初のリクエストは、エージェントの再起動直後に送信されます。

例 1

ホストメタデータを使用して Linux ホストと Windows ホストを区別します。

ホストを Zabbix サーバーによって自動登録したいとします。ネットワーク上にアクティブな Zabbix エージェント（上記の” 設定” セクションを参照）があります。ネットワーク上に Windows ホストと Linux ホストがあり、Zabbix フロントエンドで”Linux by Zabbix agent” と”Windows by Zabbix agent” のテンプレートを使用したいとします。その場合、ホストの登録時に適切な Linux/Windows テンプレートを適用する必要があります。デフォルトでは、自動登録時にホスト名のみがサーバーに送信されますが、これでは不十分な場合があります。適切なテンプレートがホストに適用されていることを確認するには、ホストメタデータを使用する必要があります。

フロントエンドの設定

最初に行うべきことは、フロントエンドの設定です。2 つのアクションを作成します。最初のアクション:

- 名前: Linux ホストの自動登録
- 条件: ホストメタデータに *Linux* を含む
- 操作: テンプレートのリンク: Linux

Note:
この場合、「ホストの追加」操作は省略できます。テンプレートをホストにリンクするには、最初にホストを追加する必要があるため、サーバーが自動的にそれを行います。

2 つ目のアクション:

- 名前: Windows ホストの自動登録
- 条件: ホストメタデータに *Windows* を含む
- 操作: テンプレートのリンク: Windows

エージェントの設定

次に、エージェントを設定する必要があります。エージェントの設定ファイルに次の行を追加します。

```
HostMetadataItem=system.uname
```

このようにすることで、エージェントが実行されているホストに応じて、ホストメタデータに「Linux」または「Windows」が含まれることを確認できます。この場合のホストメタデータの例：

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-0PXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

設定ファイルを変更した後は、エージェントを再起動するのを忘れないでください。

例 2

ステップ ***1***

ホストメタデータを使用して、不要なホストの登録に対する基本的な保護を行います。

フロントエンドの設定

フロントエンドでアクションを作成し、推測しにくいシークレットコードを使用して不要なホストを拒否します。

- 名前: Autoregistration action Linux
- 条件:
  - 計算タイプ: AND
  - 条件 (A): ホストメタデータに//Linux//を含む
  - 条件 (B): ホストメタデータに//21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//を含む
- オペレーション:
  - ユーザーにメッセージを送信: Admin すべてのメディア経由
  - ホストグループに追加: Linux servers
  - テンプレートをリンク: Linux

この方法だけではデータが平文で送信されるため、強力な保護は提供されないことに注意してください。変更を即座に反映させるには、設定キャッシュのリロードが必要です。

エージェントの設定

エージェントの設定ファイルに次の行を追加します。

```
HostMetadata=Linux      21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

ここで”Linux” はプラットフォーム名であり、残りの文字列は推測しにくい秘密のテキストです。

設定ファイルを変更した後は、エージェントの再起動を忘れないでください。

ステップ ***2***

すでに登録されているホストに追加の監視を追加することも可能です。

フロントエンドの設定

フロントエンドでアクションを更新します:

- 名前: Autoregistration action Linux
- 条件:
  - 計算タイプ: AND
  - 条件 (A): ホストメタデータに Linux を含む
  - 条件 (B): ホストメタデータに 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae を含む
- オペレーション:
  - ユーザーにメッセージを送信: Admin (すべてのメディア)
  - ホストグループに追加: Linux servers
  - テンプレートをリンク: Linux
  - テンプレートをリンク: MySQL by Zabbix Agent

エージェントの設定

エージェントの設定ファイルの次の行を更新します。

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

設定ファイルを変更した後は、エージェントの再起動を忘れないでください。

3 ローレベルディスカバリ

概要　低レベルディスカバリ（LLD）は、ホスト上のさまざまなエンティティに対してアイテム、トリガー、グラフを自動的に作成する方法を提供します。たとえば、Zabbix は、各ファイルシステムやネットワークインターフェースのアイテムを手動で作成することなく、マシン上のファイルシステムやネットワークインターフェースの監視を自動的に開始できます。LLD は、ハイパーバイザー上で検出された仮想マシンを自動的にホストとして作成したり、ネストされたディスカバリルールを作成して多段階のディスカバリを実現したりすることもできます。さらに、Zabbix を設定して、定期的に実行されるディスカバリの実際の結果に基づいて不要なエンティティを自動的に削除することも可能です。

ユーザーは、特定の JSON プロトコルに従うことで、独自のディスカバリタイプを定義できます。

ディスカバリプロセスの一般的なアーキテクチャは以下の通りです。

まず、ユーザーはデータ収集 > テンプレートのディスカバリ列でディスカバリルールを作成します。ディスカバリルールは、(1) 必要なエンティティ（たとえば、ファイルシステムやネットワークインターフェース）を検出するアイテムと、(2) そのアイテムの値に基づいて作成されるアイテム、トリガー、グラフのプロトタイプで構成されます。

必要なエンティティを検出するアイテムは、他の場所で見られる通常のアイテムと同様です。サーバーは Zabbix エージェント（またはアイテムのタイプに設定されているもの）にそのアイテムの値を要求し、エージェントはテキスト値で応答します。違いは、エージェントが応答する値が JSON 形式の検出されたエンティティのリストを含む必要があることです。この形式の詳細はカスタムディスカバリチェックの実装者にとってのみ重要ですが、返される値にはマクロ → 値のペアのリストが含まれていることを知っておく必要があります。たとえば、アイテム「net.if.discovery」は、2 つのペア「{#IFNAME}」→「lo」と「{#IFNAME}」→「eth0」を返す場合があります。

これらのマクロは、名前、キー、およびその他のプロトタイプフィールドで使用され、実際のアイテム、トリガー、グラフ、さらには検出された各エンティティのホストを作成するために受信した値に置き換えられます。LLD マクロの使用に関するオプションの完全なリストを参照してください。

サーバーがディスカバリアイテムの値を受信すると、マクロ → 値のペアを調べ、各ペアに基づいてプロトタイプから実際のアイテム、トリガー、グラフを生成します。上記の「net.if.discovery」の例では、サーバーはループバックインターフェース「lo」用のアイテム、トリガー、グラフのセットと、インターフェース「eth0」用の別のセットを生成します。

Zabbix 4.2 以降、低レベルディスカバリルールによって返される JSON の形式が変更されていることに注意してください。JSON に「data」オブジェクトが含まれていることは期待されなくなりました。低レベルディスカバリは、アイテム値の前処理や JSON ドキュメント内の低レベルディスカバリマクロ値へのカスタムパスなどの新機能をサポートするために、配列を含む通常の JSON を受け入れるようになりました。

組み込みのディスカバリキーは、JSON ドキュメントのルートで LLD 行の配列を返すように更新されています。配列フィールドが {#MACRO} 構文をキーとして使用している場合、Zabbix はマクロと値を自動的に抽出します。新しいネイティブディスカバリチェックは、「data」要素なしで新しい構文を使用します。低レベルディスカバリ値を処理する際、最初にルート（`$.` または `$.data` の配列）が特定されます。

「data」要素はディスカバリに関連するすべてのネイティブアイテムから削除されていますが、後方互換性のために Zabbix は「data」要素を含む JSON 表記も引き続き受け入れますが、その使用は推奨されません。JSON が「data」配列要素のみを持つオブジェクトを含む場合、自動的に JSONPath `$.data` を使用して要素の内容を抽出します。低レベルディスカバリは、JSONPath 構文で指定されたカスタムパスを持つユーザー定義の LLD マクロもオプションで受け入れるようになりました。

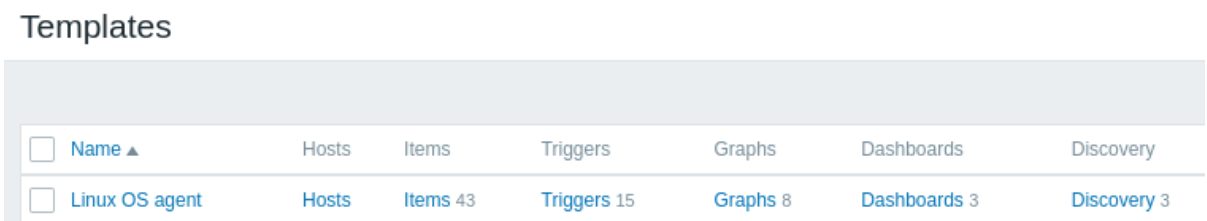
> **Warning:**
> 上記の変更の結果、新しいエージェントは古い Zabbix サーバーと連携できなくなります。

参照: 検出されたエンティティ

ローレベルディスカバリの設定　ファイルシステムディスカバリの例に基づいて、ローレベルディスカバリを説明します。

ディスカバリを設定するには、次の手順を実行します。

- データ収集 > テンプレートまたは ホストに移動します。
- 適切なテンプレート/ホストの行で ディスカバリをクリックします。



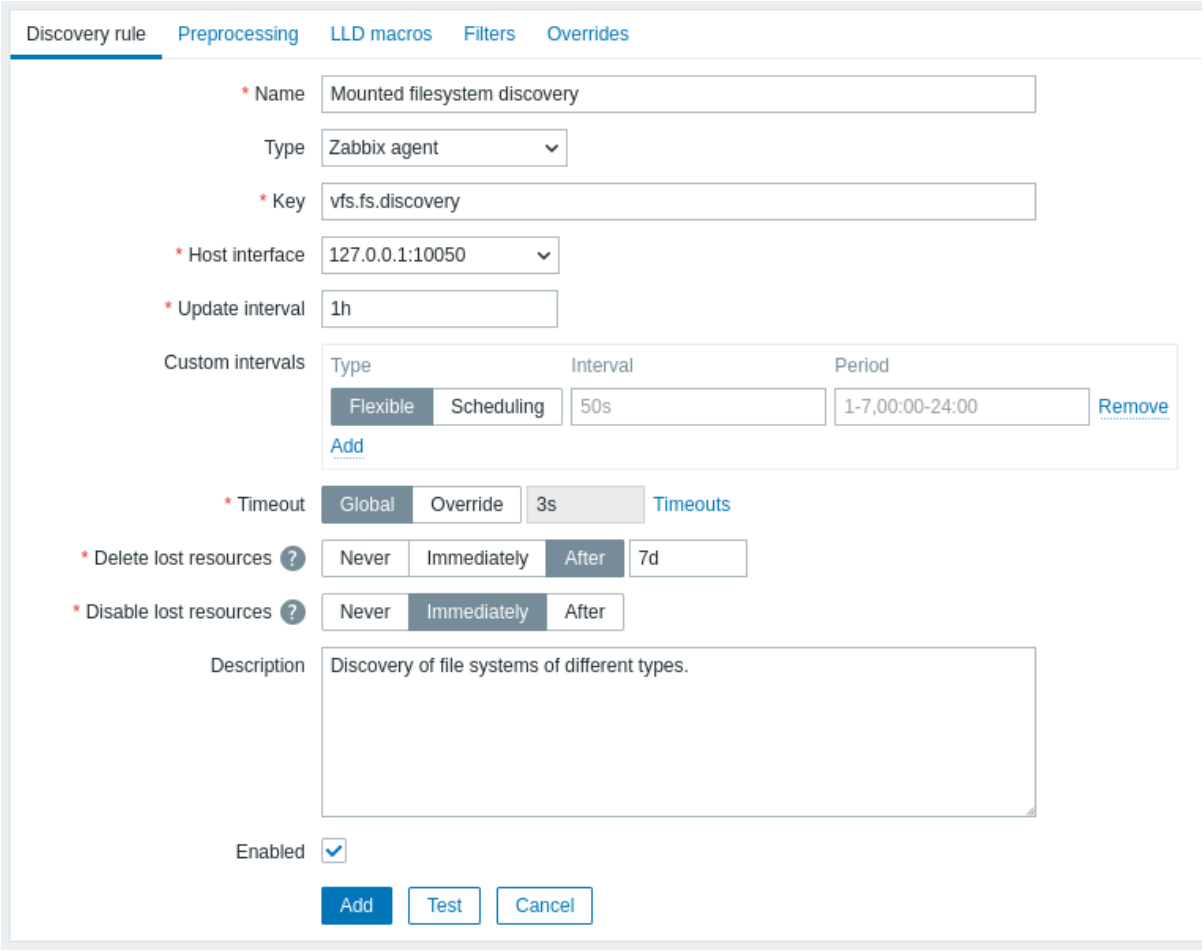
- 画面右上の ディスカバリルールの作成をクリックします。
- ディスカバリルールのフォームに必要な詳細を入力します。

ディスカバリールール

ディスカバリールールのフォームには 5 つのタブがあり、左から右へディスカバリー中のデータフローを表しています:

- ディスカバリールール - ディスカバリーデータを取得するための組み込みアイテムまたはカスタムスクリプトを指定します。
- 前処理 - 検出されたデータに前処理を適用します。
- *LLD* マクロ - 検出されたアイテムやトリガーなどで使用するマクロ値を抽出できます。
- フィルター - 検出された値をフィルタリングできます。
- オーバーライド - 特定の検出オブジェクトに適用する際に、アイテム、トリガー、グラフ、ホストプロトタイプを変更できます。

ディスカバリールールタブには、ディスカバリーに使用するアイテムキー（およびいくつかの一般的なディスカバリールール属性）が含まれています:



すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	ディスカバリールールの名前。
タイプ	ディスカバリーを実行するチェックのタイプ。 この例では、*Zabbix* エージェントアイテムタイプを使用しています。 ディスカバリールールは、通常のアイテムに依存する依存アイテムにもできます。別のディスカバリールールに依存することはできません。依存アイテムの場合は、該当するタイプ（依存アイテム）を選択し、「マスターアイテム」フィールドにマスターアイテムを指定します。マスターアイテムは存在している必要があります。
キー	ディスカバリーアイテムキーを入力します（最大 2048 文字）。 たとえば、組み込みの「vfs.fs.discovery」アイテムキーを使用して、コンピューター上に存在するファイルシステムのリスト、そのタイプ、マウントオプションを含む JSON 文字列を返すことができます。 ファイルシステムディスカバリーのもう 1 つのオプションとして、「vfs.fs.get」エージェントキーによるディスカバリー結果の使用も可能です（例を参照）。

パラメータ	説明
更新間隔	Zabbix がディスカバリーを実行する頻度を指定します。最初にファイルシステムディスカバリーを設定する際は短い間隔に設定し、動作を確認したら 30 分以上に設定することをお勧めします。ファイルシステムは通常あまり変更されないためです。 時間のサフィックスがサポートされています（例: 30s, 1m, 2h, 1d）。 ユーザーマクロがサポートされています。 注: カスタム間隔が非ゼロ値で存在する場合のみ、更新間隔を「0」に設定できます。0 に設定され、カスタム間隔（柔軟またはスケジュール）が非ゼロ値で存在する場合、アイテムはカスタム間隔の期間中にポーリングされます。 新しいディスカバリールールは、作成から 60 秒以内にチェックされます。ただし、スケジューリングまたは柔軟な更新間隔があり、更新間隔が 0 に設定されている場合は除きます。 既存のディスカバリールールについては、今すぐ実行ボタンを押すことで、ディスカバリーを即時に実行できます。
カスタム間隔	アイテムのチェック用にカスタムルールを作成できます: 柔軟 - 更新間隔の例外を作成（異なる頻度の間隔） スケジューリング - カスタムポーリングスケジュールを作成。 詳細はカスタム間隔を参照してください。
タイムアウト	ディスカバリーチェックのタイムアウトを設定します。タイムアウトオプションを選択: グローバル - プロキシ/グローバルタイムアウトが使用されます（グレー表示のタイムアウトフィールドに表示）; オーバーライド - カスタムタイムアウトが使用されます（タイムアウトフィールドで設定、許容範囲: 1 - 600 秒）。時間のサフィックス（例: 30s, 1m）やユーザーマクロがサポートされています。 タイムアウトリンクをクリックすると、プロキシタイムアウトやグローバルタイムアウト（プロキシ未使用時）を設定できます。タイムアウトリンクは、管理者タイプのユーザーで、管理 > 一般または管理 > プロキシフロントエンドセクションへの権限がある場合のみ表示されます。
失われたリソースの削除	ディスカバリーステータスが「未検出」になった後、検出されたエンティティが削除されるまでの期間を指定します: しない - 削除しません; 即時 - すぐに削除します; 指定時間後 - 指定した期間後に削除します。値は失われたリソースの無効化値より大きくする必要があります。 時間のサフィックスがサポートされています（例: 2h, 1d）。 ユーザーマクロがサポートされています。 注: 「即時」は推奨されません。フィルターを誤って編集しただけで、すべての履歴データとともにエンティティが削除される可能性があるためです。 手動で無効化されたリソースは、低レベルディスカバリーによって削除されません。
失われたリソースの無効化	ディスカバリーステータスが「未検出」になった後、検出されたエンティティが無効化されるまでの期間を指定します: しない - 無効化しません; 即時 - すぐに無効化します; 指定時間後 - 指定した期間後に無効化します。値はディスカバリールールの更新間隔より大きくする必要があります。 自動的に無効化されたリソースは、低レベルディスカバリーで再検出された場合、再度有効になります。手動で無効化されたリソースは、再検出されても有効になりません。 失われたリソースの削除が「即時」に設定されている場合、このフィールドは表示されません。 時間のサフィックスがサポートされています（例: 2h, 1d）。 ユーザーマクロがサポートされています。
説明	説明を入力します。
有効	チェックされている場合、ルールが処理されます。

Note:
ディスカバリールールの履歴は保持されません。

前処理

前処理タブでは、ディスカバリの結果に適用する変換ルールを定義できます。このステップでは、1 つまたは複数の変換が可能です。変換は定義された順序で実行されます。すべての前処理は Zabbix サーバーによって実行されます。

関連情報:

- 前処理の詳細
- 前処理のテスト

タイプ	変換	説明
テキスト		
	正規表現	受信した値を <pattern> 正規表現に一致させ、抽出した <output> で値を置き換えます。正規表現は、\N シーケンスで最大 10 個のキャプチャグループの抽出をサポートします。 パラメータ: **pattern** - 正規表現 **output** - 出力フォーマットテンプレート。\N（N=1...9）エスケープシーケンスは N 番目の一致グループに置き換えられます。\0 エスケープシーケンスは一致したテキストに置き換えられます。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	置換	検索文字列を見つけて別の文字列（または何もなし）に置き換えます。検索文字列のすべての出現箇所が置き換えられます。 パラメータ: 検索文字列 - 検索して置換する文字列、大文字小文字を区別（必須） 置換文字列 - 検索文字列を置換する文字列。置換文字列を空にすることで、検索文字列を見つけたときに削除することもできます。 改行、復帰、タブ、スペースを検索または置換するためにエスケープシーケンス”\n \r \t \s” を使用できます。バックスラッシュは”\\” でエスケープでき、エスケープシーケンスは”\\n” でエスケープできます。改行、復帰、タブのエスケープは低レベルディスカバリ中に自動的に行われます。
構造化データ		
	JSONPath	JSONPath 機能を使用して、JSON データから値またはフラグメントを抽出します。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	XML XPath	XPath 機能を使用して、XML データから値またはフラグメントを抽出します。 このオプションを使用するには、Zabbix サーバーが libxml サポートでコンパイルされている必要があります。 例: `number(/document/item/value)` は `<document><item><value>10</value></item></document>` から 10 を抽出します。 `number(/document/item/@attribute)` は `<document><item attribute="10"></item></document>` から 10 を抽出します。 `/document/item` は `<document><item><value>10</value></item></document>` から `<item><value>10</value></item>` を抽出します。 名前空間はサポートされていないことに注意してください。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	CSV から JSON	CSV ファイルデータを JSON 形式に変換します。 詳細については、CSV から JSON への前処理を参照してください。
	XML から JSON	XML 形式のデータを JSON に変換します。 詳細については、シリアライズルールを参照してください。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
SNMP		

タイプ		
	SNMP walk 値	指定した OID/MIB 名で値を抽出し、フォーマットオプションを適用します: 変更なし - Hex-STRING をエスケープされていない 16 進文字列として返します（表示ヒントは引き続き適用されることに注意）; **Hex-STRING** から **UTF-8** - Hex-STRING を UTF-8 文字列に変換します ; **Hex-STRING** から **MAC** - Hex-STRING を MAC アドレス文字列に変換します（' 'は':'に置換されます）; **BITS** から整数 - 16 進文字列として表現されたビット文字列の最初の 8 バイト（例 : "1A 2B 3C 4D"）を 64 ビットの符号なし整数に変換します。8 バイトを超えるビット文字列の場合、以降のバイトは無視されます。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	SNMP walk から *JSON*	SNMP 値を JSON に変換します。JSON 内のフィールド名と対応する SNMP OID パスを指定します。フィールド値は指定した SNMP OID パスの値で埋められます。 この前処理ステップは、SNMP OID ディスカバリで使用できます。 *SNMP walk* 値ステップと同様の値のフォーマットオプションが利用可能です。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	SNMP get 値	SNMP get 値にフォーマットオプションを適用します: **Hex-STRING** から **UTF-8** - Hex-STRING を UTF-8 文字列に変換します ; **Hex-STRING** から **MAC** - Hex-STRING を MAC アドレス文字列に変換します（' 'は':'に置換されます）; **BITS** から整数 - 16 進文字列として表現されたビット文字列の最初の 8 バイト（例 : "1A 2B 3C 4D"）を 64 ビットの符号なし整数に変換します。8 バイトを超えるビット文字列の場合、以降のバイトは無視されます。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
カスタムスクリプト		
	JavaScript	パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。 利用可能な JavaScript の長さは使用するデータベースによって異なります。 詳細については、Javascript 前処理を参照してください。
検証		
	正規表現に一致しない	値が一致してはいけない正規表現を指定します。 例: `Error:(.*?)\.` 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	JSON のエラーをチェック	JSONPath にあるアプリケーションレベルのエラーメッセージをチェックします。成功し、メッセージが空でない場合は処理を停止し、それ以外の場合はこの前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップ情報を追加せずに、そのままユーザーに報告されることに注意してください。 例: `$.errors`。`{"errors":"e1"}` のような JSON が受信された場合、次の前処理ステップは実行されません。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。

タイプ		
	XML のエラーをチェック	Xpath にあるアプリケーションレベルのエラーメッセージをチェックします。成功し、メッセージが空でない場合は処理を停止し、それ以外の場合はこの前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップ情報を追加せずに、そのままユーザーに報告されることに注意してください。 無効な XML の解析に失敗した場合はエラーは報告されません。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
	正規表現に一致する	値が一致しなければならない正規表現を指定します。 失敗時にカスタムチェックボックスをマークすると、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定するなど、カスタムのエラー処理オプションを指定できます。
スロットリング		
	ハートビート付きで変更なしを破棄	定義された期間（秒）内に値が変更されていない場合、その値を破棄します。 秒数を指定するために正の整数値がサポートされています（最小値は 1 秒）。このフィールドでは時間のサフィックス（例: 30s, 1m, 2h, 1d）を使用できます。ユーザーマクロや低レベルディスカバリマクロもこのフィールドで使用できます。 ディスカバリアイテムにはスロットリングオプションを 1 つだけ指定できます。 例: `1m`。同一のテキストが 60 秒以内にこのルールに 2 回渡された場合、破棄されます。 注意: アイテムプロトタイプを変更してもスロットリングはリセットされません。スロットリングは前処理ステップが変更されたときのみリセットされます。
Prometheus		
	Prometheus から *JSON*	必要な Prometheus メトリクスを JSON に変換します。 詳細はPrometheus チェックを参照してください。

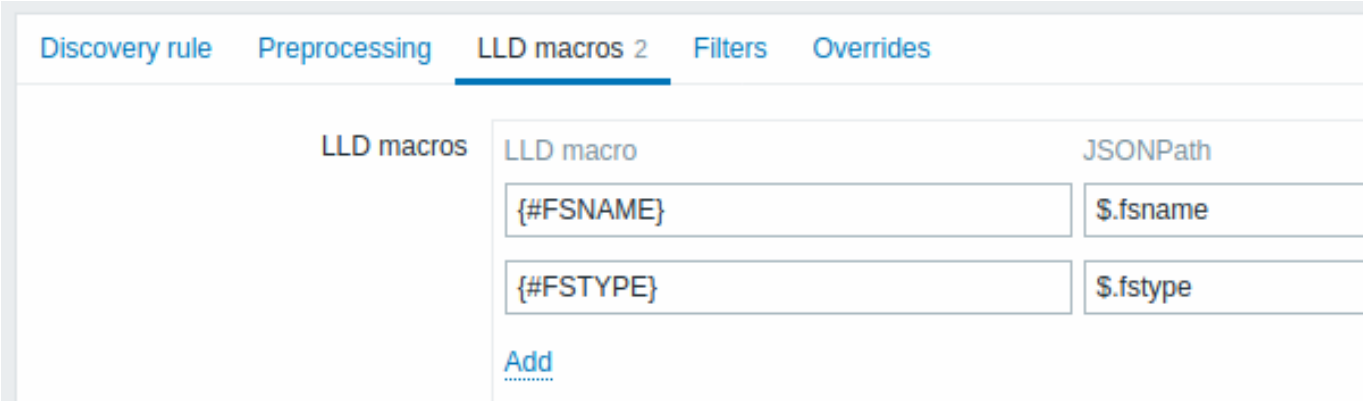
ディスカバリルールがテンプレート経由でホストに適用されている場合、このタブの内容は読み取り専用であることに注意してください。

カスタムマクロ

LLD マクロタブでは、カスタムのローレベルディスカバリマクロを指定できます。

カスタムマクロは、返される JSON に必要なマクロがすでに定義されていない場合に便利です。例えば、以下のような場合です。

- ファイルシステムディスカバリ用のネイティブな `vfs.fs.discovery` キーは、`{#FSNAME}`、`{#FSTYPE}` などのいくつかの事前定義された LLD マクロを持つ JSON を返します。これらのマクロは、アイテムやトリガープロトタイプ（ページの後続セクションを参照）で直接使用できます。カスタムマクロを定義する必要はありません。
- `vfs.fs.get` エージェントアイテムもファイルシステムデータを含む JSON を返しますが、事前定義された LLD マクロはありません。この場合、マクロを自分で定義し、JSONPath を使用して JSON 内の値にマッピングできます。

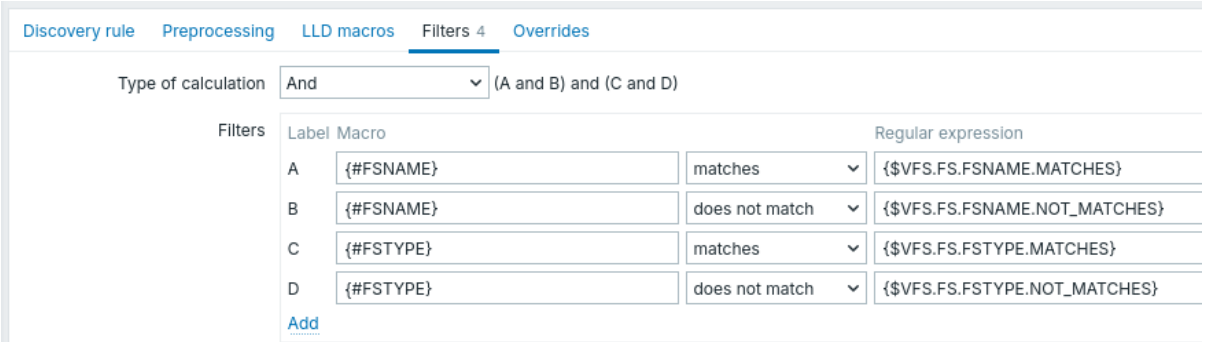


抽出された値は、ディスカバリされたアイテムやトリガーなどで使用できます。値はディスカバリの結果およびこれまでの前処理ステップから抽出されることに注意してください。

パラメータ	説明
LLD マクロ *JSONPath*	ローレベルディスカバリマクロの名前。次の構文を使用します: {#MACRO}。 JSONPath 構文を使用して、LLD 行から LLD マクロ値を抽出するために使用されるパス。 返された JSON から抽出された値は、アイテム、トリガーなどのプロトタイプフィールドの LLD マクロを置き換えるために使用されます。 JSONPath はドット表記またはブラケット表記で指定できます。特殊文字や Unicode の場合はブラケット表記を使用する必要があります。例: `$['unicode + special chars #1']['unicode + special chars #2']`。 例えば、`$.foo` はこの JSON から"bar" と"baz" を抽出します: `[{"foo":"bar"},{"foo":"baz"}]` また、`$.foo` はこの JSON からも"bar" と"baz" を抽出します: `{"data":[{"foo":"bar"},{"foo":"baz"}]}`。これは、単一の"data" オブジェクトが自動的に処理されるためです（Zabbix 4.2 以前のローレベルディスカバリ実装との後方互換性のため）。

フィルター

フィルターを使用して、条件に一致するエンティティの実際のアイテム、トリガー、グラフのみを生成できます。フィルタータブには、ディスカバリ値をフィルタリングするためのディスカバリルールフィルター定義が含まれています。



パラメータ	説明
計算タイプ	フィルターの計算には以下のオプションがあります: **And** - すべてのフィルターを通過する必要があります; **Or** - 1 つのフィルターを通過すれば十分です; **And/Or** - 異なるマクロ名には *And*、同じマクロ名には *Or* を使用します; カスタム式 - フィルターのカスタム計算を定義することができます。式にはリスト内のすべてのフィルターを含める必要があります。255 文字までに制限されています。
フィルター	以下のフィルター条件演算子が利用可能です: 一致する、一致しない、存在する、存在しない。 一致するおよび一致しない演算子はPerl 互換正規表現（PCRE）を想定しています。たとえば、C:、D:、E: ファイルシステムのみに関心がある場合は、「マクロ」に {#FSNAME} を、「正規表現」に"^C|^D|^E" を入力します。{#FSTYPE} マクロ（例: "^ext|^reiserfs"）を使用してファイルシステムタイプで、{#FSDRIVETYPE} マクロ（例: "fixed"、Windows エージェントのみ対応）を使用してドライブタイプでフィルタリングすることも可能です。 「正規表現」フィールドには、正規表現を入力するか、グローバル正規表現を参照できます。 正規表現をテストするには、「grep -E」を使用できます。例: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` 存在するおよび存在しない演算子は、レスポンス内の指定された LLD マクロの有無に基づいてエンティティをフィルタリングできます。 フィルターのマクロがレスポンスに存在しない場合、そのマクロに「存在しない」条件が指定されていない限り、検出されたエンティティは無視されることに注意してください。 マクロの不在が式の結果に影響する場合、警告が表示されます。たとえば、{#B} が存在しない場合: {#A} が 1 に一致し、かつ {#B} が 2 に一致する - 警告が表示されます {#A} が 1 に一致する、または {#B} が 2 に一致する - 警告なし

Warning:
LLD ルールで使用される正規表現にミスやタイプミスがある場合（例: 不正な「ディスカバリ用ファイルシステム」正規表現）、多数のホストの構成要素、履歴値、イベントが削除される可能性があります。

Attention:
ファイルシステム名が大文字小文字のみ異なる場合に正しくディスカバリするには、MySQL で Zabbix データベースを大文字小文字を区別するように作成する必要があります。

オーバーライド

オーバーライドタブでは、指定した条件を満たす検出オブジェクトに対して、アイテム、トリガー、グラフ、ホスト、ディスカバリプロトタイプのリストや属性を変更するルールを設定できます。

オーバーライド（存在する場合）はドラッグ＆ドロップで並べ替え可能なリストで表示され、定義された順に実行されます。新しいオーバーライドの詳細を設定するには、オーバーライドブロックのをクリックします。既存のオーバーライドを編集するには、オーバーライド名をクリックします。オーバーライドルールの詳細を編集できるポップアップウィンドウが開きます。

すべての必須パラメータは赤いアスタリスクでマークされています。

パラメータ	説明
名前	LLD ルールごとに一意のオーバーライド名。
フィルターが一致した場合	フィルター条件が一致した場合に次のオーバーライドを処理するかどうかを定義します： オーバーライドを継続 - 後続のオーバーライドも処理されます。 処理を停止 - 先行する（存在する場合）およびこのオーバーライドの操作が実行され、一致した LLD 行に対して後続のオーバーライドは無視されます。
フィルター	オーバーライドを適用する検出エンティティを決定します。オーバーライドフィルターはディスカバリルールのフィルターの後に処理され、同じ機能を持ちます。
操作	オーバーライド操作は以下の詳細で表示されます： 条件 - オブジェクトタイプ（アイテムプロトタイプ/トリガープロトタイプ/グラフプロトタイプ/ホストプロトタイプ）と満たすべき条件（等しい/等しくない/含む/含まない/一致する/一致しない） アクション - 操作の編集および削除用のリンクが表示されます。

操作の設定

新しい操作の詳細を設定するには、操作ブロックのをクリックします。既存の操作を編集するには、操作の横にあるをクリックします。操作の詳細を編集できるポップアップウィンドウが開きます。

パラメータ		説明
オブジェクト		4 種類のオブジェクトが利用可能です： アイテムプロトタイプ トリガープロトタイプ グラフプロトタイプ ホストプロトタイプ ディスカバリプロトタイプ
条件		操作を適用するエンティティをフィルタリングできます。
	演算子	サポートされる演算子： 等しい - このプロトタイプに適用 等しくない - このプロトタイプ以外すべてに適用 含む - プロトタイプ名がこの文字列を含む場合に適用 含まない - プロトタイプ名がこの文字列を含まない場合に適用 一致する - プロトタイプ名が正規表現に一致する場合に適用 一致しない - プロトタイプ名が正規表現に一致しない場合に適用
	パターン	正規表現または検索する文字列。

パラメータ		説明
オブジェクト: アイテムプロトタイプ		
	有効で作成	チェックボックスをオンにすると、元のアイテムプロトタイプ設定を上書きするボタンが表示されます： はい - アイテムは有効な状態で追加されます。 いいえ - アイテムは検出されたエンティティに追加されますが、無効な状態になります。
	ディスカバリ	チェックボックスをオンにすると、元のアイテムプロトタイプ設定を上書きするボタンが表示されます： はい - アイテムが追加されます。 いいえ - アイテムは追加されません。
	更新間隔	チェックボックスをオンにすると、アイテムの異なる間隔を設定できる 2 つのオプションが表示されます： 間隔 - アイテムの更新間隔。ユーザーマクロおよび時間サフィックス（例：30s、1m、2h、1d）がサポートされています。カスタム間隔を使用する場合は 0 に設定する必要があります。 カスタム間隔 - 柔軟/スケジューリング間隔を指定するにはをクリックします。詳細はカスタム間隔を参照してください。
	ヒストリ	チェックボックスをオンにすると、アイテムの異なるヒストリ保存期間を設定できるボタンが表示されます： 保存しない - 選択した場合、ヒストリは保存されません。 保存期間 - 選択した場合、右側に保存期間を指定する入力フィールドが表示されます。ユーザーマクロおよびLLD マクロがサポートされています。
	トレンド	チェックボックスをオンにすると、アイテムの異なるトレンド保存期間を設定できるボタンが表示されます： 保存しない - 選択した場合、トレンドは保存されません。 保存期間 - 選択した場合、右側に保存期間を指定する入力フィールドが表示されます。ユーザーマクロおよびLLD マクロがサポートされています。
	タグ	チェックボックスをオンにすると、タグと値のペアを指定できる新しいブロックが表示されます。 これらのタグは、タグ名が一致していても、アイテムプロトタイプで指定されたタグに追加されます。

パラメータ		説明
オブジェクト: トリガープロトタイプ		
	有効で作成	チェックボックスをオンにすると、元のトリガープロトタイプ設定を上書きするボタンが表示されます： はい - トリガーは有効な状態で追加されます。 いいえ - トリガーは検出されたエンティティに追加されますが、無効な状態になります。
	ディスカバリ	チェックボックスをオンにすると、元のトリガープロトタイプ設定を上書きするボタンが表示されます： はい - トリガーが追加されます。 いいえ - トリガーは追加されません。
	深刻度	チェックボックスをオンにすると、トリガーの深刻度を変更できるボタンが表示されます。
	タグ	チェックボックスをオンにすると、タグと値のペアを指定できる新しいブロックが表示されます。 これらのタグは、タグ名が一致していても、トリガープロトタイプで指定されたタグに追加されます。
オブジェクト: グラフプロトタイプ		
	ディスカバリ	チェックボックスをオンにすると、元のグラフプロトタイプ設定を上書きするボタンが表示されます： はい - グラフが追加されます。 いいえ - グラフは追加されません。

パラメータ		説明
オブジェクト: ホストプロトタイプ		
	有効で作成	チェックボックスをオンにすると、元のホストプロトタイプ設定を上書きするボタンが表示されます： はい - ホストは有効な状態で作成されます。 いいえ - ホストは無効な状態で作成されます。
	ディスカバリ	チェックボックスをオンにすると、元のホストプロトタイプ設定を上書きするボタンが表示されます： はい - ホストが検出されます。 いいえ - ホストは検出されません。
	テンプレートのリンク	チェックボックスをオンにすると、テンプレートを指定する入力フィールドが表示されます。テンプレート名の入力を開始するか、フィールド横の選択をクリックしてポップアップウィンドウのリストからテンプレートを選択します。 このオーバーライドのテンプレートは、すでにホストプロトタイプにリンクされているすべてのテンプレートに追加されます。
	タグ	チェックボックスをオンにすると、タグと値のペアを指定できる新しいブロックが表示されます。 これらのタグは、タグ名が一致していても、ホストプロトタイプで指定されたタグに追加されます。
	ホストインベントリ	チェックボックスをオンにすると、ホストプロトタイプの異なるインベントリモードを選択できるボタンが表示されます： 無効 - ホストインベントリを入力しない 手動 - 手動で詳細を入力 自動 - 収集したメトリクスに基づいてホストインベントリデータを自動入力します。

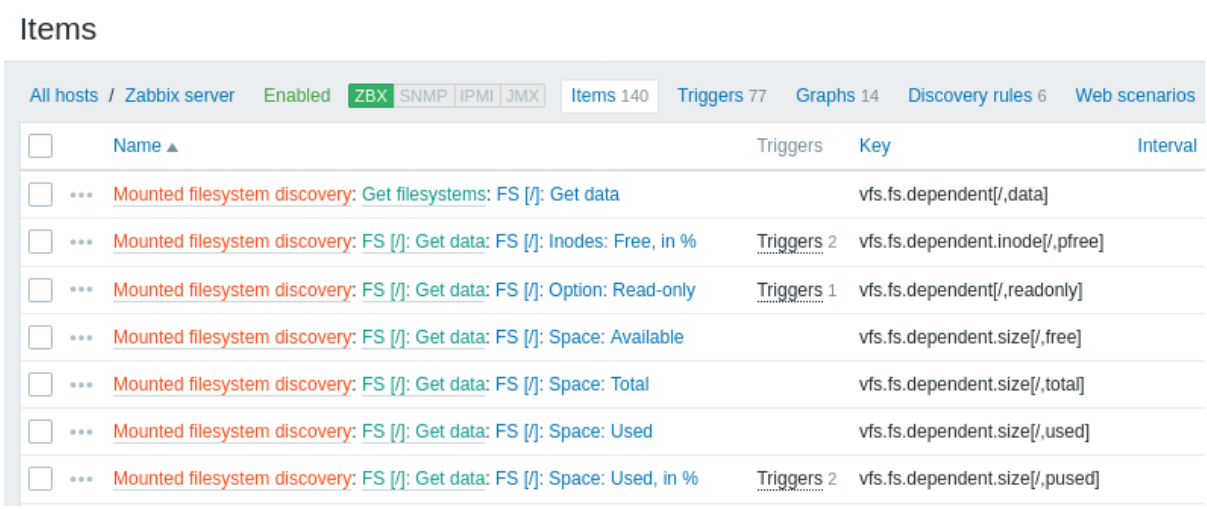
フォームボタン

フォームの下部にあるボタンで、いくつかの操作を実行できます。

Add	ディスカバリルールを追加します。このボタンは新しいディスカバリルールの場合のみ利用できます。
Update	ディスカバリルールのプロパティを更新します。このボタンは既存のディスカバリルールの場合のみ利用できます。
Clone	現在のディスカバリルールのプロパティを元に、別のディスカバリルールを作成します。
Execute now	ディスカバリルールに基づいて即座にディスカバリを実行します。ディスカバリルールはすでに存在している必要があります。詳細はこちらを参照してください。 ※ ディスカバリを即座に実行した場合、設定キャッシュは更新されないため、ディスカバリルール設定のごく最近の変更は反映されません。

Test	ディスカバリルールの設定をテストします。このボタンを使用して、設定（接続性やパラメータの正しさなど）を恒久的に変更を適用せずに検証できます。
Delete	ディスカバリルールを削除します。
Cancel	ディスカバリルールのプロパティの編集をキャンセルします。

検出されたエンティティ　以下のスクリーンショットは、ホストの設定で検出されたアイテム、トリガー、グラフがどのように表示されるかを示しています。検出されたエンティティには、それが由来するディスカバリールールへのオレンジ色のリンクが接頭辞として付けられます。



すでに同じ一意性基準を持つエンティティ（例えば、同じキーのアイテムや同じ名前のグラフ）が存在する場合、検出されたエンティティは作成されません。この場合、フロントエンドには、低レベルディスカバリールールが特定のエンティティを作成できなかったというエラーメッセージが表示されます。ただし、いくつかのエンティティが作成できずスキップされた場合でも、ディスカバリールール自体は未サポートにはなりません。ディスカバリールールは他のエンティティの作成/更新を続行します。

検出されたエンティティ（ホスト、ファイルシステム、インターフェースなど）が検出されなくなった場合（またはフィルターを通過しなくなった場合）、それに基づいて作成されたエンティティは自動的に無効化され、最終的に削除されることがあります。

失われたリソースは、失われたリソースを無効化パラメータの値に基づいて自動的に無効化される場合があります。これは失われたホスト、アイテム、トリガーに影響します。

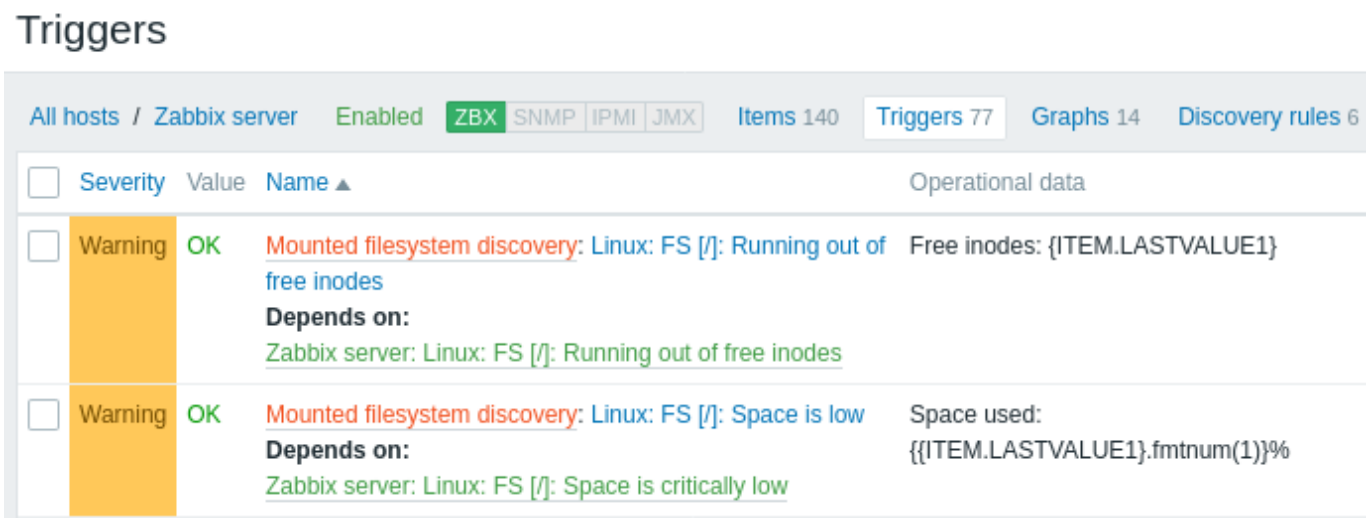
失われたリソースは、失われたリソースを削除パラメータの値に基づいて自動的に削除される場合があります。これは失われたホスト、ホストグループ、アイテム、トリガー、グラフに影響します。

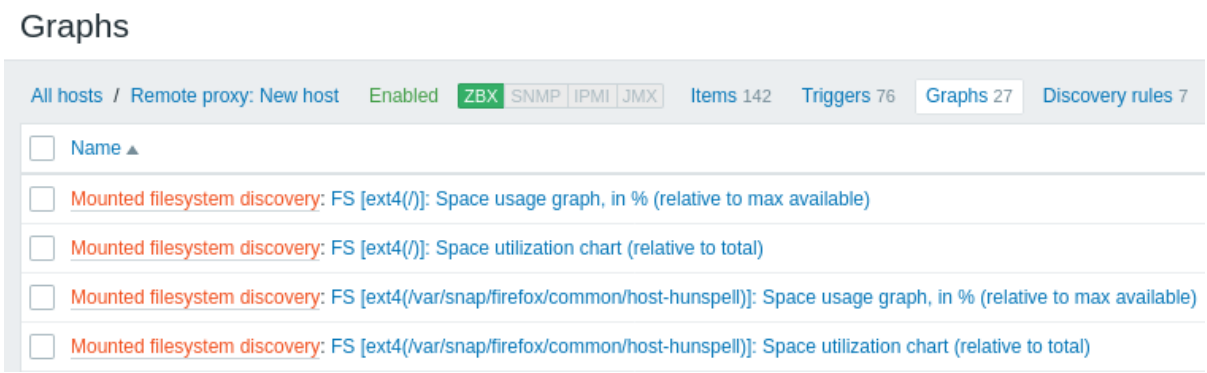
検出されたエンティティが「これ以上検出されない」状態になると、エンティティリストにライフタイムインジケーターが表示されます。マウスポインターをその上に移動すると、ステータスの詳細を示すメッセージが表示されます。



エンティティが削除対象としてマークされたが、期待された時刻に削除されなかった場合（ディスカバリールールやアイテムホストが無効化された場合）、次回ディスカバリールールが処理される際に削除されます。

他のエンティティを含むエンティティが削除対象としてマークされている場合、ディスカバリールールレベルで変更されても更新されません。例えば、LLD ベースのトリガーが削除対象のアイテムを含んでいる場合、それらは更新されません。





その他のディスカバリタイプ　他の種類の標準ディスカバリについての詳細やハウツーは、以下のセクションで確認できます。

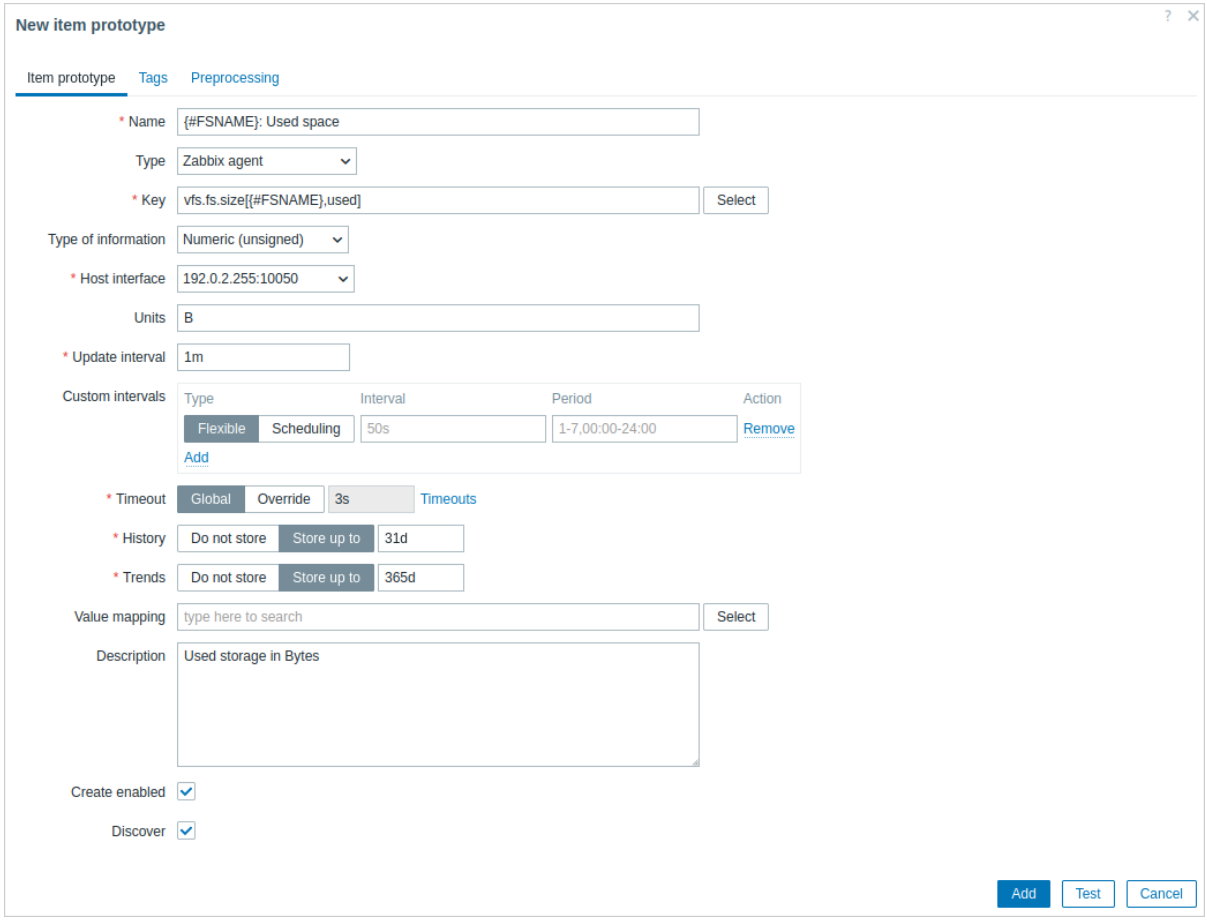
- ネットワークインターフェースのディスカバリ
- CPU および CPU コアのディスカバリ
- SNMP OIDのディスカバリ
- JMX オブジェクトのディスカバリ
- ODBC SQL クエリを使用したディスカバリ
- Windows サービスのディスカバリ
- Zabbix でのホストインターフェースのディスカバリ

ディスカバリアイテムの JSON 形式の詳細や、独自のファイルシステムディスカバリを Perl スクリプトとして実装する例については、カスタム LLD ルールの作成を参照してください。

1 アイテムプロトタイプ

ルールを作成したら、そのルールのアイテムに移動し、「アイテムプロトタイプの作成」を押してアイテムプロトタイプを作成します。

ファイルシステム名が必要な場所で {#FSNAME} マクロがどのように使用されているかに注目してください。ディスカバリが正しく処理されるように、アイテムキーには低レベルディスカバリマクロの使用が必須です。ディスカバリルールが処理されると、このマクロは検出されたファイルシステムに置き換えられます。



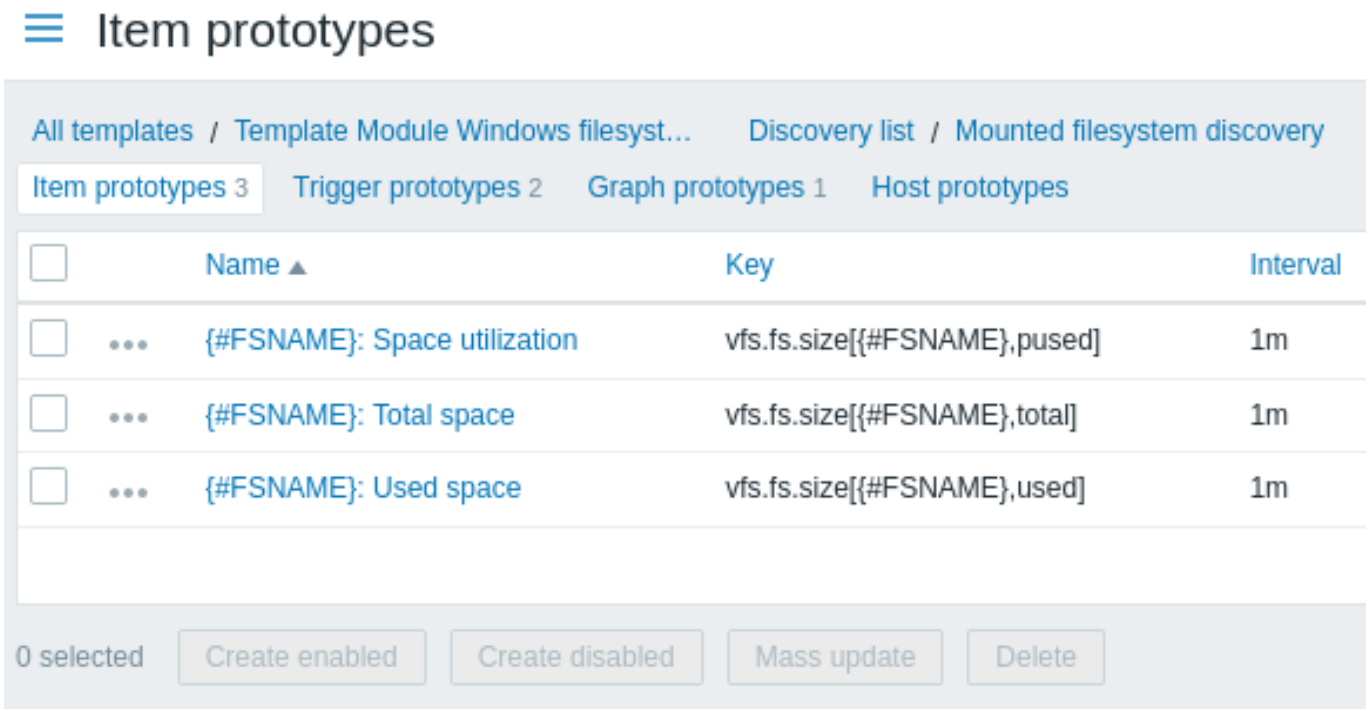
アイテムプロトタイプの設定やアイテム値の前処理パラメータでは、低レベルディスカバリマクロやユーザーマクロがサポートされています。更新間隔で使用する場合、1 つのマクロがフィールド全体を埋める必要があることに注意してください。1 つのフィールドに複数のマクロを使用したり、テキストとマクロを混在させたりすることはできません。

Note:
正規表現や XPath の前処理パラメータで安全に使用できるよう、低レベルディスカバリマクロのコンテキスト固有のエスケープが行われます。

アイテムプロトタイプ固有の属性:

パラメータ	説明
有効で作成	チェックを入れると、アイテムは有効な状態で追加されます。 チェックを外すと、アイテムは検出されたエンティティに追加されますが、無効な状態になります。
ディスカバリ	チェックを入れると（デフォルト）、アイテムは検出されたエンティティに追加されます。 チェックを外すと、ディスカバリルールでこの設定が上書きされない限り、アイテムは検出されたエンティティに追加されません。

関心のある各ファイルシステムメトリックについて、複数のアイテムプロトタイプを作成できます。



三点リーダーアイコンをクリックすると、特定のアイテムプロトタイプのメニューが開き、以下のオプションが表示されます。

- トリガープロトタイプの作成 - このアイテムプロトタイプに基づいてトリガープロトタイプを作成します
- トリガープロトタイプ - クリックすると、このアイテムプロトタイプの既に設定されているトリガープロトタイプのリンク付きリストが表示されます
- 従属アイテムの作成 - このアイテムプロトタイプの従属アイテムを作成します

複数のアイテムプロトタイプのプロパティを一度に更新したい場合は、一括更新オプションが利用できます。

2 トリガープロトタイプ

アイテムプロトタイプと同様の方法でトリガープロトタイプを作成します。

トリガープロトタイプに固有の属性:

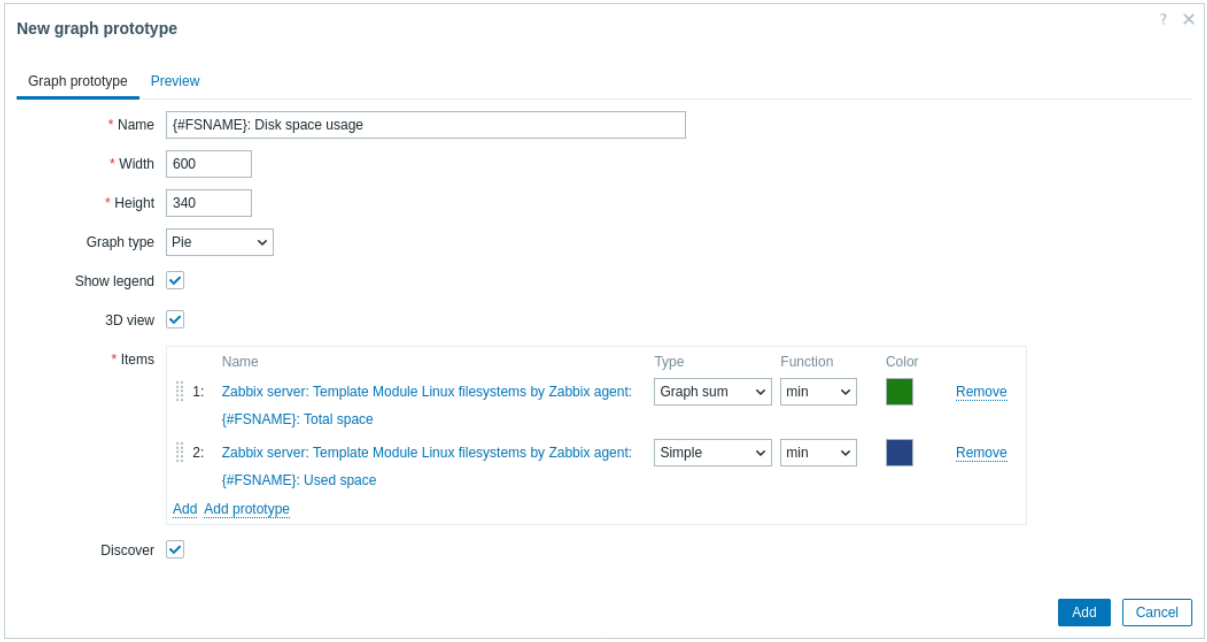
パラメータ	説明
有効で作成	チェックされている場合、トリガーは有効な状態で追加されます。 チェックされていない場合、トリガーは検出されたエンティティに追加されますが、無効な状態になります。
検出	チェックされている場合 (デフォルト)、トリガーは検出されたエンティティに追加されます。 チェックされていない場合、ディスカバリールールでこの設定が上書きされない限り、トリガーは検出されたエンティティに追加されません。

プロトタイプから実際のトリガーが作成されるとき、式で比較に使用する定数 (この例では'20') を柔軟に変更したい場合があります。そのような柔軟性を実現するために、コンテキスト付きユーザーマクロがどのように役立つかをご覧ください。

トリガープロトタイプ間の依存関係を定義できます。そのためには、依存関係タブに移動します。トリガープロトタイプは、同じローレベルディスカバリ (LLD) ルールの別のトリガープロトタイプ、または通常のトリガーに依存することができます。トリガープロトタイプは、異なる LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーに依存することはできません。ホストのトリガープロトタイプは、テンプレートのトリガーに依存することはできません。

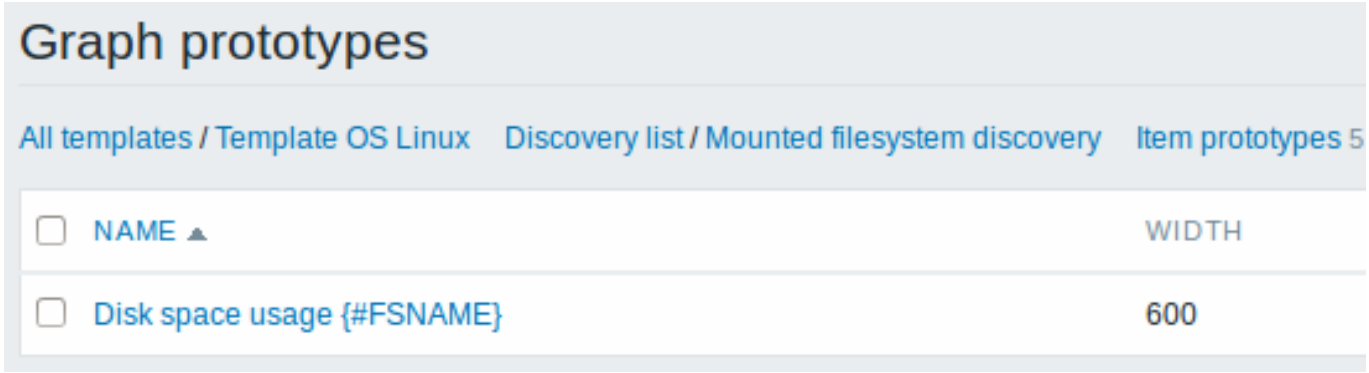
3 グラフのプロトタイプ

グラフのプロトタイプも作成できます。

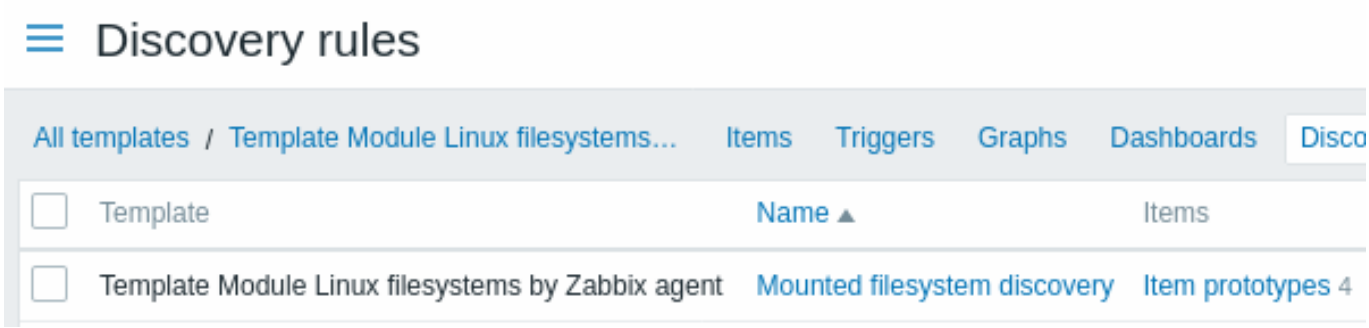


グラフのプロトタイプに固有の属性:

パラメータ	説明
ディスカバー	チェックされている場合 (デフォルト)、グラフは検出されたエンティティに追加されます。 チェックされていない場合、ディスカバリールールでこの設定が上書きされない限り、グラフは検出されたエンティティに追加されません。



最後に、以下のようなディスカバリールールを作成しました。5 つのアイテムプロトタイプ、2 つのトリガープロトタイプ、1 つのグラフプロトタイプがあります。



4 ホストプロトタイプ

ホストプロトタイプは、ローレベルディスカバリルールを通じてホストを作成するための設計図です。これらのプロトタイプは、ホストとして検出される前は、テンプレートからリンクされたものを除き、アイテムやトリガーを持つことはできません。

設定

ホストプロトタイプは、ローレベルディスカバリールールで設定します。

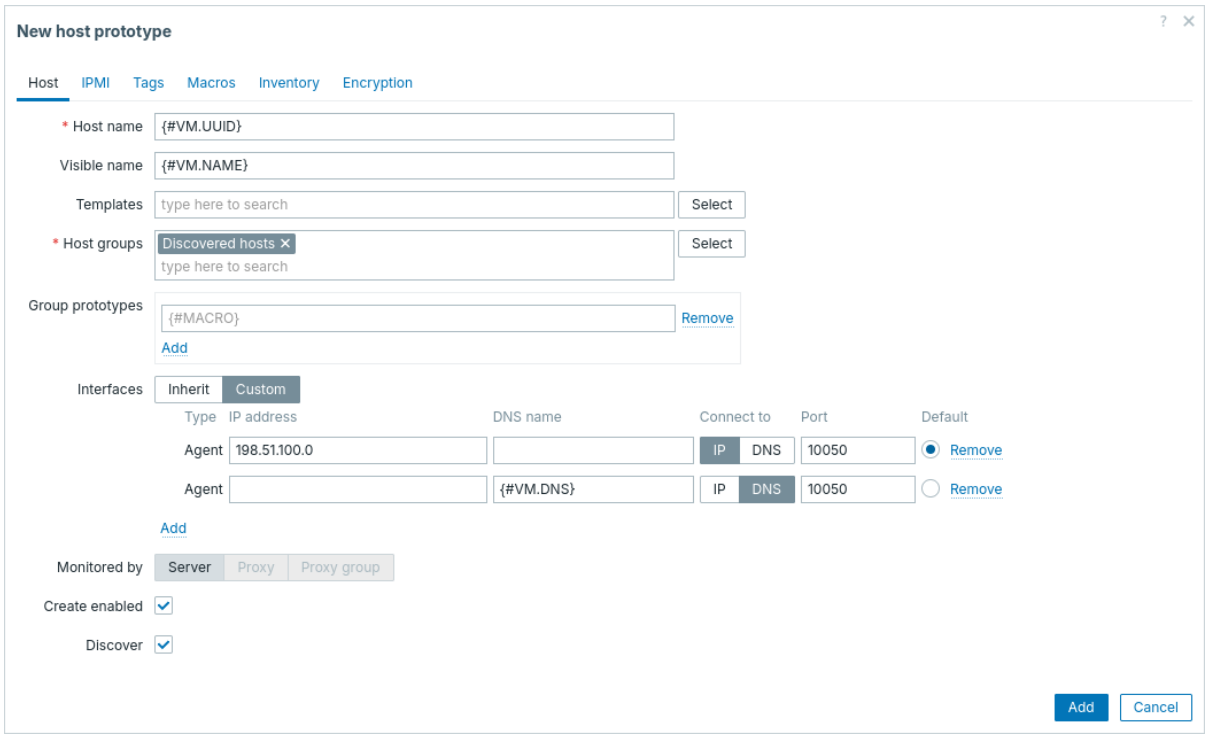
ホストプロトタイプを作成するには:

1. データ収集 → ホストに移動します。

2. 必要なホストのディスカバリをクリックして、そのホストに設定されているローレベルディスカバリールールの一覧に移動します。

3. 必要なディスカバリールールのホストプロトタイプをクリックします。

4. 右上のホストプロトタイプの作成ボタンをクリックし、設定フォームに入力します。



必須入力フィールドは赤いアスタリスクでマークされています。

パラメータ	説明
ホスト名	このパラメータには、作成されるホストのホスト名が一意になるように、少なくとも 1 つのローレベルディスカバリーマクロを含める必要があります。
表示名	ローレベルディスカバリーマクロがサポートされています。
グループプロトタイプ	ローレベルディスカバリーマクロを使用してホストグループプロトタイプを指定できます。 指定されたグループプロトタイプに基づいて、ホストグループがディスカバリされ、作成され、作成されたホストにリンクされます。すでに他のローレベルディスカバリールールによって作成されているディスカバリ済みグループも、作成されたホストにリンクされます。ただし、手動で作成されたホストグループと一致するディスカバリ済みホストグループは、作成されたホストにリンクされません。
インターフェース	ディスカバリされたホストが、ディスカバリールールが属するホストから IP を継承するか（デフォルト）、カスタムインターフェースを取得するかを設定します。 ローレベルディスカバリーマクロおよびユーザーマクロがサポートされています。
監視サーバー	このフィールドは、ホスト設定の該当フィールドと自動的に同期されます。
有効で作成	ディスカバリされたホストのステータスを設定します。チェックを外すと、ホストは無効として作成されます。
ディスカバリ	ホストプロトタイプからホストを作成するかどうかを設定します。チェックを外すと、ローレベルディスカバリールールでこの設定が上書きされない限り、ホストプロトタイプからホストは作成されません。

Note:
ローレベルディスカバリーマクロは、タグ値やホストプロトタイプのユーザーマクロ値にも対応しています。
 値のマッピングはホストプロトタイプではサポートされていません。

ホストプロトタイプの設定例については、仮想マシンの監視を参照してください。

ホストインターフェース

カスタムインターフェースを追加するには、インターフェースセレクターを「継承」から「カスタム」に切り替えます。Add をクリックし、インターフェースの種類（Zabbix エージェント、SNMP、JMX、IPMI）を選択します。

Note:
カスタムが選択されているがインターフェースが設定されていない場合、ホストはインターフェースなしで作成されます。
 継承が選択されていて、ホストプロトタイプがテンプレートに属している場合、検出されたすべてのホストは、テンプレートがリンクされているホストからホストインターフェースを継承します。

複数のカスタムインターフェースが指定されている場合、デフォルト列でプライマリインターフェースを設定できます。

カスタムホストインターフェースの設定例については、*VMware* 監視セットアップ例を参照してください。

Warning:
ホストインターフェースに正しいデータが含まれている場合のみ、ホストが作成されます。

検出されたホスト

ホストリストでは、検出されたホストの前に、それらを作成したディスカバリールールの名前が付けられます。

検出されたホストは、ほとんどのパラメータをホストプロトタイプから読み取り専用として継承します。次の検出されたホストのパラメータのみが設定可能です:

- テンプレート - 新しいテンプレートのリンクや手動で追加したテンプレートのリンク解除。ホストプロトタイプから継承されたテンプレートはリンク解除できません。
- 説明 - ホストの説明を追加または編集します。
- ステータス - ホストの有効化または無効化。
- タグ - 新しいタグの追加や手動で追加したタグの削除。ホストプロトタイプから継承されたタグは削除できません。タグは重複（同じ名前と値のタグ）できないことに注意してください。ホストプロトタイプに新しいタグが追加され、それが検出されたホストの手動で追加されたタグと一致する場合、ディスカバリー時に手動で追加されたタグは継承されたタグに置き換えられます。
- マクロ - 新しいホストマクロの追加や手動で追加したもの、ホストプロトタイプから継承したものの削除、マクロ値やタイプの変更。

Note:
検出されたホストは、ディスカバリールールが設定されているホストからユーザーマクロも継承します。これらの継承されたマクロは削除でき、値やタイプも変更できます。

検出されたホストが他のホストを検出することもできます。たとえば、ホストプロトタイプ（ハイパーバイザーの検出に使用）がリンクされているテンプレートに、独自のホストプロトタイプ（仮想マシンの検出に使用）を含む低レベルディスカバリールールが含まれている場合、Zabbix はハイパーバイザーとその仮想マシンの両方を検出します。これらの VM 上のコンテナも追加で検出するには、検出された VM に低レベルディスカバリールールを作成するか、ホストプロトタイプを含むテンプレートをリンクするか、またはホストプロトタイプにあらかじめテンプレートをリンクしておくことができます。

検出されたホストは手動で削除できます。ただし、ディスカバリーが有効になっている場合は再度検出されることに注意してください。

検出されなくなったホストは、次のいずれかになります:

- 自動的に無効化（ディスカバリールールの失われたリソースの無効化値に基づく）
- 自動的に削除（ディスカバリールールの失われたリソースの削除値に基づく）

5 ディスカバリのプロトタイプ

概要

ディスカバリプロトタイプは、「親」ディスカバリルール内にあるネストされた低レベルディスカバリルールであり、独自のアイテム、トリガーなどを持つオブジェクトの多層ディスカバリを作成できます。例えば、データベースサーバー上のすべてのデータベースインスタンスをディスカバリし、次に各インスタンスのテーブルスペースをディスカバリし、次に各テーブルスペースのテーブルをディスカバリしたい場合などです。

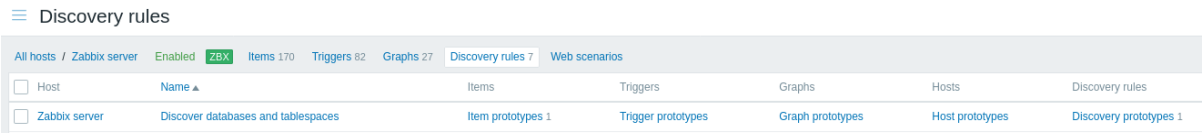
ディスカバリプロトタイプには、独自のアイテム、トリガー、グラフ、ホスト、ディスカバリプロトタイプがあります。ネストタイプを指定した場合、ネストされたディスカバリプロトタイプは親ルールと同じ JSON 値を使用します。

ディスカバリプロトタイプのネストレベルに制限はありません。

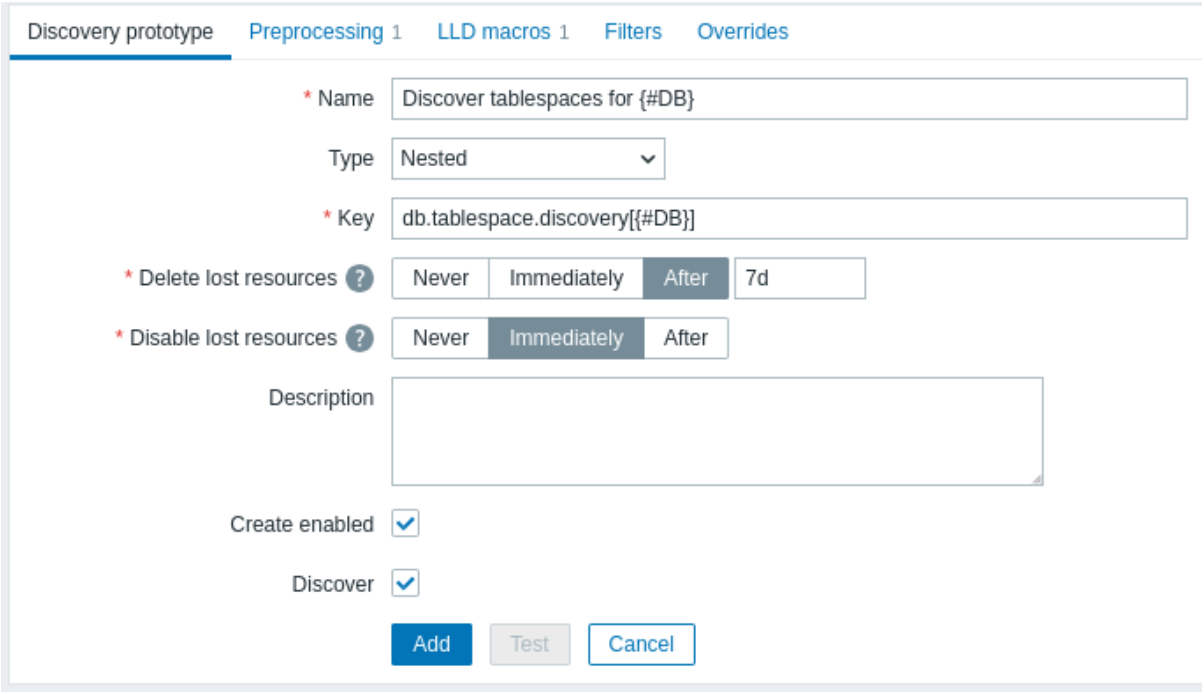
設定

ディスカバリープロトタイプを作成するには:

- 既存のディスカバリールールの行でディスカバリープロトタイプをクリックします



- ディスカバリープロトタイプの作成をクリックします



このフォームの設定フィールドは、通常のローレベルディスカバリと共通です。

開いたディスカバリープロトタイプフォームでタイプに「ネスト」を選択すると、ディスカバリールール（ディスカバリープロトタイプから）は、親ディスカバリールールと同じ JSON 値からの JSON オブジェクトに基づいて生成されます。たとえば、元の JSON が [<object A>, <object B>] で、1 つのネストされたディスカバリールールプロトタイプがある場合、オブジェクト A とオブジェクト B のデータに基づいて 2 つのディスカバリールールがそれぞれ生成されます。

この場合、ディスカバリープロトタイプは親ルールと同時に有効化されます。したがって、ネストされたルールは前処理を使用して、親によってすでに取得された同じデータの別の「スライス」に対して動作できます。

親 LLD ルールの LLD マクロは、ネストされたディスカバリールールでも利用できます。

検出されたホストでのネストされた LLD ルール

ネストされた低レベルディスカバリールールは、ホストプロトタイプに割り当てられたホストテンプレートで使用できます。検出されたホストにネストされたディスカバリールールが存在する場合、ホストの検出に使用された JSON オブジェクトは、このホストのすべてのネスト型 LLD ルールにも送信されます。詳細は例を参照してください。

ホストを作成したディスカバリールールの LLD マクロは、ネストされたディスカバリールールで利用できます。

例

以下のマルチレベル JSON を受信した場合を例に、ディスカバリプロトタイプの適用例を示します。

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
  },
  {
```

```
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```

ケース 1

データベースサーバー上のデータベースインスタンスを検出し、各インスタンスのテーブルスペースを検出します。

1. データベースサーバーの検出に関連するホストが少なくとも 1 つあります。
2. このホストのために、データベースとテーブルスペースの検出という名前のLLD ルールを作成します。
3. このルールの *LLD* マクロタブに切り替え、マクロ `{#DB}=$.database` を追加します。
4. このルールのアイテムプロトタイプを *{#DB}* のアクティブ接続数(タイプ：エージェント、キー：`db.connections[{#DB}]`）という名前で追加します。
5. 各データベースに関連するアイテムが検出されます：

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. このルールのディスカバリープロトタイプを *{#DB}* のテーブルスペースの検出（タイプ：ネスト、キー：`db.tablespace.discovery[{#DB}]`）という名前で作成します。
7. このディスカバリープロトタイプの前処理タブに切り替え、ステップ `JSONPath=$.tablespaces` を追加します。
8. このディスカバリープロトタイプの *LLD* マクロタブに切り替え、マクロ `{#TSNAME}=$.name` を追加します。
9. このディスカバリープロトタイプのアイテムプロトタイプを *{#DB}* の *{#TSNAME}* テーブルスペースのサイズ（タイプ：エージェント、キー：`db.ts.size[{#DB}, {#TSNAME}]`）という名前で作成します。
10. 各データベースの各テーブルスペースに関連するアイテムが検出されます：

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

キーは `db.ts.size[db1,ts1]`、`db.ts.size[db1,ts2]`、...、`db.ts.size[db3,ts3]` です。

ケース 2

データベースサーバー上のデータベースインスタンスを検出されたホストとして表現し、それぞれのインスタンスのテーブルスペースを検出します。

1. データベースサーバーの検出に関連する少なくとも 1 つのホスト（ルートホスト）が存在します。
2. 各データベースのテーブルスペースを検出するためのテンプレートを作成します。
3. このテンプレートに *Active connections to {#DB}* という名前のアイテム(タイプ: エージェント、キー: `db.connections[{#DB}]`）を作成します。
4. このテンプレートに *Discover tablespaces* という名前のLLD ルール（タイプ: ネスト）を作成します。
5. このルールのプリプロセスタブに切り替え、ステップ `JSONPath=$.tablespaces` を追加します。
6. このルールの *LLD* マクロタブに切り替え、マクロ `{#TSNAME}=$.name` を追加します。

7. このルールに *Size of tablespace {#TSNAME} for {#DB}* という名前のアイテムプロトタイプ（タイプ: エージェント、キー: `db.ts.size[{#DB}, {#TSNAME}]`）を作成します。
8. ルートホストに戻り、このホストに *Discover databases and tablespaces* という名前のLLD ルールを作成します。
9. このルールの *LLD* マクロタブに切り替え、マクロ `{#DB}=$.database` を追加します。
10. このルールに *Host for database {#DB}* という名前のホストプロトタイプを追加します。
11. このホストプロトタイプのマクロタブに切り替え、マクロ `{$DB}={#DB}`（ステップ 3 のアイテム名とキー用）を追加します。
12. ステップ 2 のテンプレートをこのホストプロトタイプにリンクします。
13. 検出されたホストには、それぞれのデータベースおよびそのテーブルスペースに関連する検出されたアイテムが含まれます:

ホスト	アイテム
Host for database db1	Active connections to db1
	Size of tablespace ts1 for db1
	Size of tablespace ts2 for db1
	Size of tablespace ts3 for db1
Host for database db2	Active connections to db2
	Size of tablespace ts1 for db2
	Size of tablespace ts2 for db2
	Size of tablespace ts3 for db2
Host for database db3	Active connections to db3
	Size of tablespace ts1 for db3
	Size of tablespace ts2 for db3
	Size of tablespace ts3 for db3

6 ローレベルディスカバリに関する注意点

ユーザーマクロコンテキストでの LLD マクロの使用

LLD マクロは、トリガープロトタイプなど、ユーザーマクロコンテキスト内で使用できます。

同じアイテムに対する複数の LLD ルール

同じディスカバリーアイテムで複数のローレベルディスカバリールールを定義することができます。

そのためには、エイリアスエージェントパラメータを定義し、異なるディスカバリールールで変更されたディスカバリーアイテムキーを使用できるようにします。例えば、`vfs.fs.discovery[foo]`、`vfs.fs.discovery[bar]` などです。

返り値のデータ制限

Zabbix サーバーが直接受信する場合、ローレベルディスカバリールールの JSON データには制限がありません。これは、返り値がデータベースに保存されることなく処理されるためです。

カスタムローレベルディスカバリールールにも制限はありません。ただし、カスタムローレベルディスカバリールールのデータがユーザーパラメータを使用して取得される場合は、ユーザーパラメータの返り値の制限が適用されます。

データが Zabbix プロキシを経由する必要がある場合、このデータをデータベースに保存する必要があります。その場合は、データベースの制限が適用されます。

7 ディスカバリールール

サイドバーを使用して、さまざまなケースのディスカバリールールの設定例を参照してください。

1 ファイルシステムマウントの検出

概要

マウントされたファイルシステムとそのプロパティを検出することができます:

- マウントポイント名
- ファイルシステムタイプ
- ファイルシステムサイズ
- inode 統計
- マウントオプション

これを行うには、以下の組み合わせを使用します:

- マスターアイテムとしてvfs.fs.getエージェントアイテム
- 従属のローレベルディスカバリールールとアイテムプロトタイプ

設定

マスターアイテム

以下のキーを使用して Zabbix エージェントアイテムを作成します。

vfs.fs.get

大きな JSON データになる可能性があるため、情報の型を「テキスト」に設定します。

このアイテムによって返されるデータは、マウントされたファイルシステムについて以下のような内容を含みます。

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

依存 LLD ルール

「依存アイテム」タイプとして低レベルディスカバリールールを作成します。

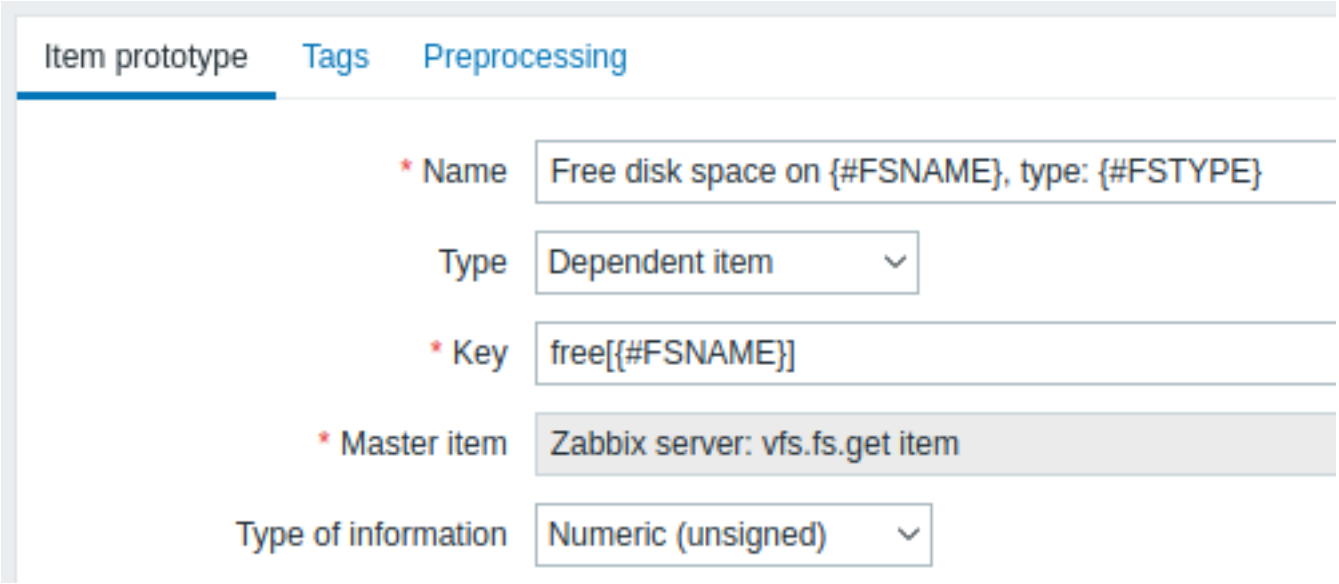
マスターアイテムとして、作成した vfs.fs.get アイテムを選択します。

「LLD マクロ」タブで、対応する JSONPath を持つカスタムマクロを定義します。

「フィルター」タブでは、読み書き可能なファイルシステムのみをフィルタリングする正規表現を追加できます。

依存アイテムのプロトタイプ

この LLD ルールで" 依存アイテム" タイプのアイテムプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した vfs.fs.get アイテムを選択します。



アイテムのプロトタイプ名とキーでカスタムマクロが使用されていることに注意してください。

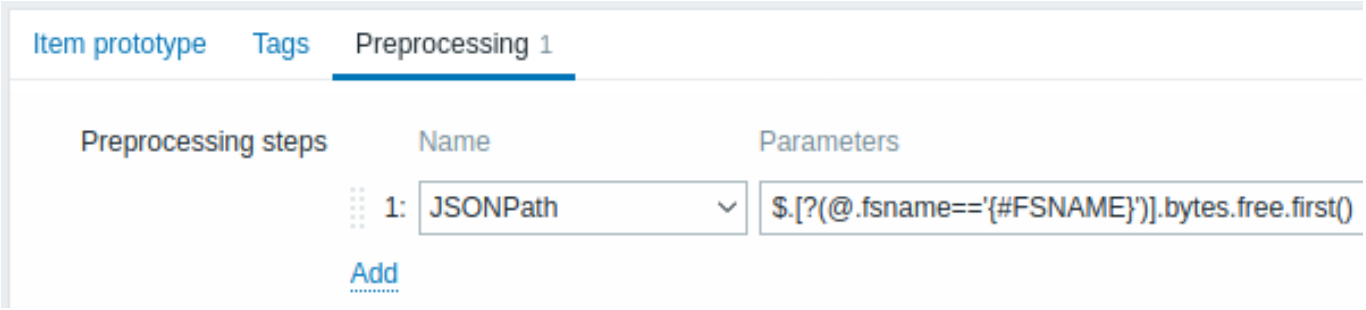
- *Name*: Free disk space on {#FSNAME}, type: {#FSTYPE}
- *Key*: Free[{#FSNAME}]

情報の種類として、次を使用します。

- 数値 *(整数)* 次のメトリックの場合'free', 'total', 'used'
- 数値 *(浮動小数)* 次のメトリックの場合'pfree', 'pused' (percentage)

アイテムのプロトタイプの" 保存前処理" タブで、JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



検出されるとマウントポイントごとに 1 つのアイテムが作成されます。この項目は、指定されたマウントポイントの空きバイト数を返します。

2 ネットワークインターフェースの検出

ファイルシステムが検出されるのと同様の方法で、ネットワークインターフェイスも検出できます。

アイテムキー

ディスカバリルールで使用するアイテムキーは次のとおりです。

```
net.if.discovery
```

サポートされているマクロ

ディスカバリルールフィルターで {#IFNAME} マクロを使用したり、アイテム、トリガー、グラフのプロトタイプを使用したりできます。

"net.if.discovery" に基づいて作成する可能性のあるアイテムのプロトタイプの例：

- "net.if.in[{#IFNAME},bytes]",
- "net.if.out[{#IFNAME},bytes]".

Windows では {#IFGUID} も返されることに注意してください。

3 CPU と CPU コアの検出

ファイルシステムが検出されるのと同様の方法で、CPU と CPU コアも検出できます。

アイテムキー

ディスカバリールールで使用するアイテムキーは

`system.cpu.discovery`

サポートされているマクロ

このディスカバリーキーは、CPU の順序番号とステータスをそれぞれ識別する {#CPU.NUMBER} と {#CPU.STATUS} の 2 つのマクロを返します。プロセッサは 0 から N-1 まで列挙され、N はプロセッサの総数です。

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown"は Windows のみ */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

実際の物理プロセッサ、コア、ハイパースレッドの間で明確な区別ができないことに注意してください。Linux、UNIX、BSD システムでは、{#CPU.STATUS} はプロセッサのステータスを返し、その値は”online” または”offline” のいずれかです。Windows システムでは、このマクロは”unknown” という 3 番目の値を表す場合があり、これはプロセッサが検出されたが、まだ情報が収集されていないことを示します。

CPU ディスカバリーは、エージェントのコレクタープロセスがコレクターによって提供されるデータと一貫性を保ち、データ取得のリソースを節約することに依存しています。このため、このアイテムキーはエージェントバイナリの test（-t）コマンドラインフラグでは動作せず、NOT_SUPPORTED ステータスと、コレクタープロセスが開始されていないことを示すメッセージが返されます。

CPU ディスカバリーに基づいて作成できるアイテムプロトタイプの例は以下の通りです。

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

アイテムキーの詳細な説明については、Zabbix エージェントのアイテムキーを参照してください。

NUMA システムで検出されたパフォーマンスカウンターアイテムに関する注意事項

一部のWindows パフォーマンスカウンターは、NUMA システムの論理プロセッサによっては利用できない場合があります。

たとえば、以下のキーを持つアイテムプロトタイプで検出されたアイテムは、最初の NUMA ノードでのみ機能する場合があります。他の NUMA ノードのアイテムは、サポートされていない状態になる場合があります。

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

また、キー「system.cpu.discovery」を持つディスカバリールールは、以下のようなパフォーマンスカウンターに代入するための低レベルディスカバリーマクロ値を提供しません。

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

system.cpu.utilアイテムは、プロセッサグループ数やプロセッサ数に応じて、CPU 使用率の監視に異なるパフォーマンスカウンターを利用します。また、Windows のプロセッサグループに関連するまれなバグも存在します。

4 SNMP OIDs の検出

概要

このセクションでは、SNMP デバイスでローレベルディスカバリを実行します。

SNMP OID のこのディスカバリ方法は、Zabbix サーバー/プロキシ 6.4 からサポートされています。

サンプル設定

1. 次のようなキーで SNMP エージェントアイテムを作成します:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```

このアイテムは、単一の SNMP テーブルウォークを実行し、すべてのテーブルエントリを 1 回のリクエストで返します。フォーマットは、`snmpwalk` ユーティリティのフォーマットオプション`-Oe -Ot -On` の出力に対応しています。

次のような複数行のテキスト値が返されます:

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. ディスカバリルールを作成します:

- 名前フィールドに、説明的なディスカバリルール名（例: ”Discover sensors”）を入力します。
- タイプフィールドで「従属アイテム」を選択します。
- キーフィールドに説明的なキー（例: ”net.if.discovery”）を入力します。
- マスターアイテムフィールドで「SNMP walk item」を選択します。

3. 前処理タブで、名前ドロップダウンで「SNMP walk to JSON」を選択し、3 つのパラメータで前処理ステップを追加します:

- フィールド名: ”{#SENSORNAME}”; *OID* プレフィックス: ”.1.3.6.1.4.1.9999.1.1.1.1.1”: フォーマット: ” 変更なし”
- フィールド名: ”{#SENSORTYPE}”; *OID* プレフィックス: ”.1.3.6.1.4.1.9999.1.1.1.1.2”: フォーマット: ” 変更なし”
- フィールド名: ”{#SENSORVALUE}”; *OID* プレフィックス: ”.1.3.6.1.4.1.9999.1.1.1.1.3”: フォーマット: ” 変更なし”

前処理後、ディスカバリルールはマクロセットの JSON 配列を返します。

例:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

各オブジェクトは 1 つの検出されたセンサーを表し、`{#SNMPINDEX}`、`{#SENSORNAME}`、`{#SENSORTYPE}`、`{#SENSORVALUE}` などのマクロを提供します。

これらは SNMP インデックスごとにグループ化されます。これは各 OID の末尾にある数値サフィックス（例: .1, .2）であり、このインデックスは SNMP テーブル内の各行を一意に識別し、自動的に `{#SNMPINDEX}` として抽出されます。

4. ディスカバリルールの下に、1 つ以上のアイテムプロトタイプ（ディスカバリルールをマスターアイテムとして）を作成します。

例: センサー値の従属アイテム

- 名前フィールドに「Sensor {#SNMPINDEX}: {#SENSORNAME}」と入力します。
- タイプフィールドで「従属アイテム」を選択します。
- キーフィールドに「sensor.value[{#SNMPINDEX}]」と入力します。
- マスターアイテムフィールドで「SNMP walk item」を選択します。

前処理タブで、パラメータフィールドに”.1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}” OID を指定し、「SNMP walk value」名の前処理ステップを追加します。フォーマット: ” 変更なし”

次のアイテムが検出されます:

名前	キー	値を抽出する OID	アイテム値
Sensor 1: Temperature Sensor	sensor.value[1]	1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	1.3.6.1.4.1.9999.1.1.1.1.3.2	200

ディスカバリルールが実行されると、sensor.value[1]、sensor.value[2] のようなアイテムが作成されます。

各従属アイテムは、前処理を使用してマスターアイテムの SNMP walk 結果から値を抽出し、個別の SNMP リクエストは実行しません。

5. ディスカバリルールの同じマクロを使用して、トリガープロトタイプで従属アイテムプロトタイプを参照します。例:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

これにより、検出された各センサー（例: sensor.value[1]、sensor.value[2]）にトリガーが作成され、最新値（温度または湿度）が 75 を超えた場合に発火します。

6. 検出された各エンティティに対して従属アイテムを含めます。グラフアイテムキーの例:

```
sensor.value[{#SNMPINDEX}]
```

{#SNMPINDEX} ごとに 1 つのグラフが作成され、温度と湿度の経時変化がプロットされます。

この設定では、検出されたアイテム数に関係なく、ポーリングサイクルごとに 1 回だけ SNMP walk リクエストが実行されます。すべての従属アイテムは、前処理を使用してマスター SNMP walk 結果から値を抽出し、SNMP トラフィックと負荷を大幅に削減します。

walk[] による動的インデックス

動的インデックス（たとえば、インターフェースインデックス）は、ハードウェアの再構成時に変更されることがあります。この動作に対応するために、次のようなキーを持つマスター SNMP walk ディスカバリールールが作成されます。

```
walk[1.3.6.1.2.1.2.2.1.10]
```

SNMP walk to JSON のプリプロセス後、結果は次のようになります。

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

従属アイテムプロトタイプは、{#SNMPINDEX} マクロを使用してキーを構築します。

```
net.if.in[{#SNMPINDEX}]
```

このプロトタイプのプリプロセスには、「SNMP walk value」名と「1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}」OID をパラメータフィールドに指定します。フォーマット:「変更しない」。

実行時には、net.if.in[2] や net.if.in[3] などの実際のアイテムが作成されます。特定のインターフェースインデックスが変更された場合（たとえば、SNMP テーブル内のインデックス 2 が 5 に置き換えられた場合）、次回のディスカバリールール実行時に以下の動作となります。

- 古い従属アイテム net.if.in[2] は「失われた」とマークされるか削除され、そのアイテムの新しいデータは収集されません。
- 新しい従属アイテム net.if.in[5] が作成され、履歴は空の状態から開始されます。
- net.if.in[2] の履歴データは自動的に net.if.in[5] に移動されません。

トリガープロトタイプの例:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

グラフプロトタイプの例（アイテムを含む）:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

この構成により、動的インデックスを持つテーブルの信頼性の高い監視が保証され、SNMP トラフィックも最小限に抑えられます。ポーリングサイクルごとに 1 回の SNMP walk だけが必要で、従属アイテムプロトタイプが必要な値を抽出します。

検出されたエンティティ

サーバーが実行されると、SNMP ディスカバリルールが返す値に基づいて、実際の従属アイテム、トリガー、グラフが作成されます。

5 SNMP OIDs の検出 **(legacy)**

概要

このセクションでは、スイッチで SNMP の検出を実行します。

アイテムキー

ファイルシステムやネットワークインターフェースのディスカバリとは異なり、アイテムは必ずしも”snmp.discovery” キーを持つ必要はありません - SNMP エージェントのアイテムタイプで十分です。

ディスカバリルールを設定するには、以下の手順を実行します。

- データ収集 → テンプレートに移動します
- 適切なテンプレートの行で ディスカバリをクリックします
- 画面右上の ディスカバリルールの作成をクリックします
- 下記のスクリーンショットのように、必要な詳細でディスカバリルールフォームに入力します

すべての必須入力フィールドには赤いアスタリスクが付いています。

ディスカバリする OID は、次の形式で SNMP OID フィールドに定義します: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

ここで *{#MACRO1}*, *{#MACRO2}* ... は有効な lld マクロ名、*oid1*, *oid2*... はこれらのマクロに対して意味のある値を生成できる OID です。組み込みマクロ *{#SNMPINDEX}* には、検出された OID のインデックスが含まれ、検出されたエンティティに適用されます。検出されたエンティティは *{#SNMPINDEX}* マクロ値でグループ化されます。

以下は、スイッチでいくつかの snmpwalk を実行した例です。

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

そして SNMP OID は次のように設定します: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

このルールでは、{#IFDESCR} マクロが **WAN**、**LAN1**、**LAN2**、{#IFPHYSADDRESS} マクロが **8:0:27:90:7a:75**、**8:0:27:90:7a:76**、**8:0:27:2b:af:9e**、{#SNMPINDEX} マクロが検出された OID のインデックス **1**、**2**、**3** に設定されたエンティティを検出します。

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
```

```
    }
]
```

エンティティに指定された OID がない場合、そのエンティティの対応するマクロは省略されます。たとえば、次のようなデータがある場合:

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

この場合、SNMP ディスカバリ `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` は次の構造を返します。

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

アイテムプロトタイプ

次のスクリーンショットは、アイテムプロトタイプでこれらのマクロをどのように使用できるかを示しています。

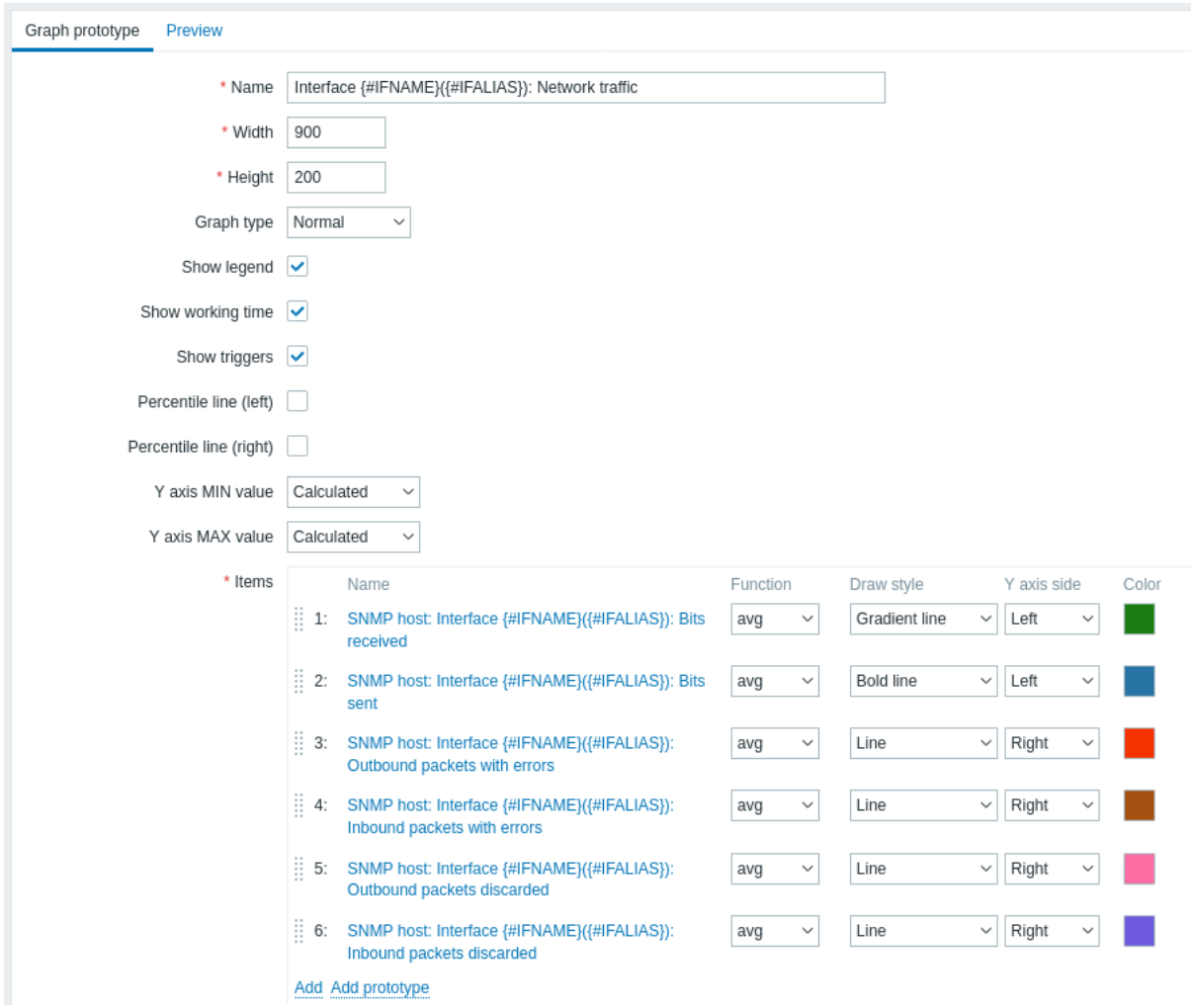
必要に応じて、いくつでもアイテムプロトタイプを作成できます。

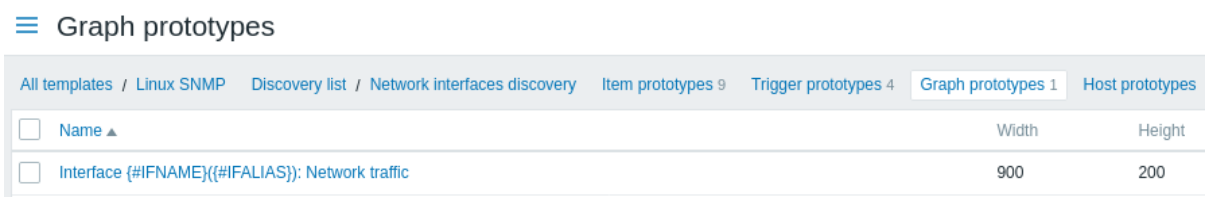
トリガープロトタイプ

次のスクリーンショットは、これらのマクロをトリガープロトタイプでどのように使用できるかを示しています。

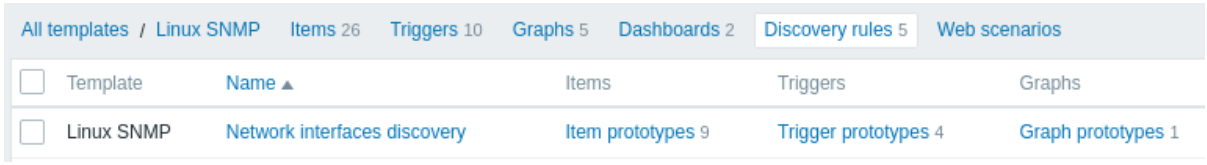
グラフプロトタイプ

次のスクリーンショットは、グラフプロトタイプでこれらのマクロをどのように使用できるかを示しています。





ディスカバリルールの概要:



検出されたエンティティ

サーバーが実行されると、SNMP ディスカバリールールが返す値に基づいて、実際のアイテム、トリガー、グラフが作成されます。ホストの設定では、それらがどのディスカバリールールから来たかを示すオレンジ色のリンクが先頭に付いています。

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 JMX オブジェクトの検出

概要

すべての JMX MBean または MBean 属性を検出 したり、これらのオブジェクトの検出パターンを指定したりできます。

ディスカバリルールを設定するには、MBean と MBean 属性の違いを理解しておく必要があります。MBean は、管理が必要なデバイス、アプリケーション、または任意のリソースを表すことができるオブジェクトです。

たとえば、Web サーバーを表す MBean があります。その属性は、接続数、スレッド数、リクエストタイムアウト、http ファイルキャッシュ、メモリ使用量などです。この考えを人間の包括的な言語で表現すると、コーヒーマシンを豆と水の補充時間などを監視対象の属性として持つ MBean として定義できます。

アイテムキー

ディスカバリールールの設定で、タイプフィールドに **JMX** エージェントを選択します。

JMX オブジェクトディスカバリ用に 2 つのアイテムキー（jmx.discovery[] と jmx.get[]）がサポートされています。

アイテムキー			
	戻り値	パラメータ	コメント
jmx.discovery[<ディスカバリーモード>,<オブジェクト名>,<一意な短い説明>]			

アイテムキー			
jmx.get[<ディスカバリーモード>,<オブジェクト名>,<一意な短い説明>]	このアイテムは、MBean オブジェクトまたはその属性を記述する LLD マクロを含む JSON 配列を返します。	ディスカバリーモード - 次のいずれか: *attributes*（JMX MBean 属性の取得、デフォルト）または *beans*（JMX MBean の取得） オブジェクト名 - 取得する MBean 名を識別するオブジェクト名パターン（ドキュメントを参照）（デフォルトは空、すべての登録済み Bean を取得） 一意な短い説明 - ホスト上で同じディスカバリーモードとオブジェクト名を持つ複数の JMX アイテムを許可する一意な説明（省略可能）	例: → jmx.discovery - すべての JMX MBean 属性を取得 → jmx.discovery[beans] - すべての JMX MBean を取得 → jmx.discovery[attributes,"*:type=GarbageCollector,name - すべてのガベージコレクタ属性を取得 → jmx.discovery[beans,"*:type=GarbageCollector,name=* - すべてのガベージコレクタを取得 このアイテムが返すことができる MBean プロパティには、マクロ名生成でサポートされる文字が制限されているため、制限事項があります（サポートされる文字は次の正規表現で表現できます: `A-Z0-9_\.`）。したがって、例えばハイフン付きの単語や非 ASCII 文字を含む MBean プロパティをディスカバリするには、`jmx.get[]` を使用する必要があります。

アイテムキー			
	このアイテムは、MBean オブジェクトまたはその属性を含む JSON 配列を返します。 `jmx.discovery[]` と比較して、LLD マクロは定義しません。	ディスカバリーモード - 次のいずれか: *attributes*（JMX MBean 属性の取得、デフォルト）または *beans*（JMX MBean の取得） オブジェクト名 - 取得する MBean 名を識別するオブジェクト名パターン（ドキュメントを参照）（デフォルトは空、すべての登録済み Bean を取得） 一意な短い説明 - ホスト上で同じディスカバリーモードとオブジェクト名を持つ複数の JMX アイテムを許可する一意な説明（省略可能）	このアイテムを使用する場合は、返された JSON から JSONPath を使用して抽出した値を指すカスタム低レベルディスカバリーマクロを定義する必要があります。

Attention:
パラメータを指定しない場合、JMX からすべての MBean 属性が要求されます。JMX ディスカバリでパラメータを指定しない、または `*:type=*,name=*` のような広範囲のすべての属性を取得しようとすると、パフォーマンス上の問題が発生する可能性があります。

jmx.discovery の使用

このアイテムは、MBean オブジェクトまたは属性を記述する低レベルディスカバリーマクロを含む JSON オブジェクトを返します。例えば、MBean 属性のディスカバリーの場合（見やすく整形）:

```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
```

```
        "{#JMXTYPE}":"javax.management.ObjectName",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

MBean のディスカバリーの場合（見やすく整形）：

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

サポートされているマクロ

次のマクロはディスカバリルールフィルター およびアイテム、トリガー、グラフのプロトタイプでの使用がサポートされています。

マクロ	説明
MBean 属性の検出	
{#JMXVALUE}	属性値
{#JMXTYPE}	属性タイプ
{#JMXOBJ}	オブジェクト名
{#JMXDESC}	属性名を含むオブジェクト名
{#JMXATTR}	属性名
MBean の検出	
{#JMXDOMAIN}	MBean ドメイン. (*Zabbix* 予約名)
{#JMXOBJ}	オブジェクト名. (*Zabbix* 予約名)
{#JMX<key property>}	MBean プロパティ (like {#JMXTYPE}, {#JMXNAME}) (参照Limitations below).

制限事項

MBean プロパティ名から LLD マクロ名を作成するアルゴリズムにはいくつかの制限があります:

- 属性名は大文字に変換されます
- 属性名が LLD マクロ名としてサポートされていない文字で構成されている場合は無視されます（LLD マクロは生成されません）。サポートされている文字は次の正規表現で表現できます: `A-Z0-9_\.`
- 属性名が”obj” または”domain” の場合、Zabbix の予約プロパティ {#JMXOBJ} および {#JMXDOMAIN} の値と重複するため無視されます

この jmx.discovery（”beans” モード）例を考えてみてください。MBean には以下のプロパティが定義されています（一部は無視されます。下記参照）：

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

JMX ディスカバリの結果、以下の LLD マクロが生成されます:

- {#JMXDOMAIN} - Zabbix 内部用、MBean のドメインを表す
- {#JMXOBJ} - Zabbix 内部用、MBean オブジェクトを表す
- {#JMXNAME} - ”name” プロパティから作成

無視されるプロパティは以下の通りです:

- тип：名前にサポートされていない文字（非 ASCII）が含まれている
- attributes[]：名前にサポートされていない文字（角括弧）が含まれている
- Name：すでに定義されている（name=test）
- domAin：Zabbix の予約名である

例

MBean を使用した LLD ルール作成のより実践的な例を 2 つ見てみましょう。MBeans を収集する LLD ルールと MBean 属性を収集する LLD ルールの違いをよりよく理解するために、次の表をご覧ください。

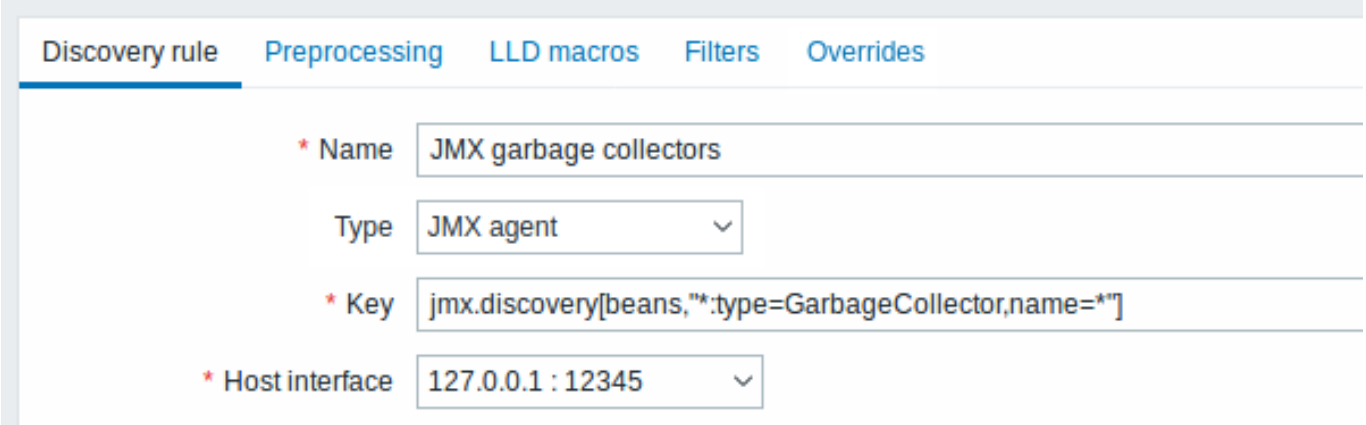
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

例 1: MBeans のディスカバリ

このルールは 3 つのオブジェクトを返します: 列の最上行: MBean1, MBean2, MBean3。

オブジェクトの詳細については、サポートされているマクロテーブルの「MBeans のディスカバリ」セクションを参照してください。

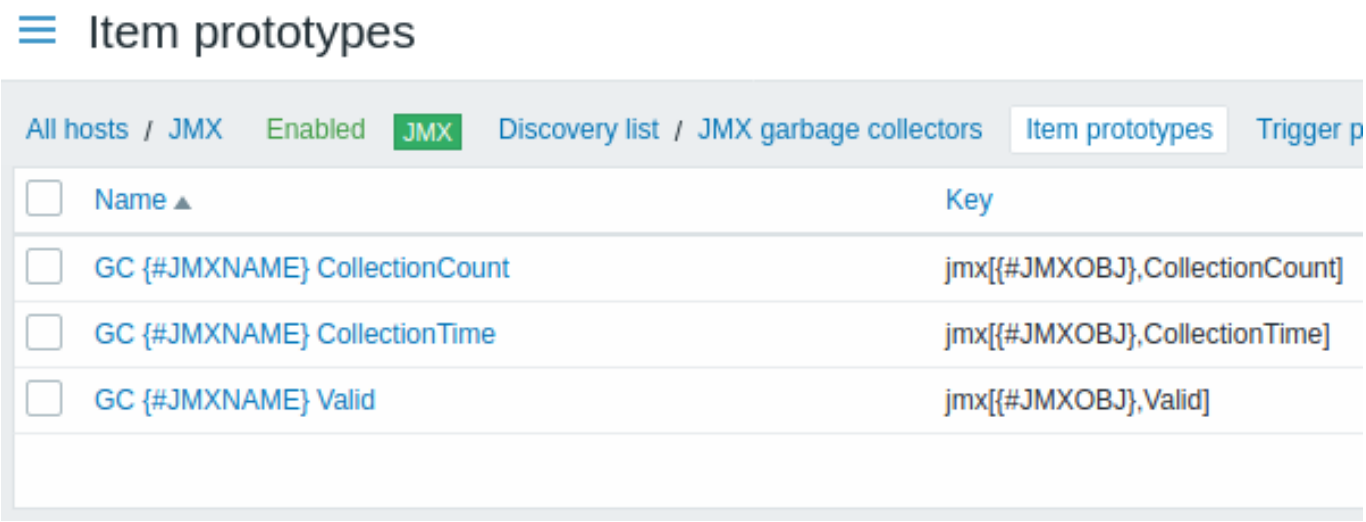
MBeans（属性なし）を収集するディスカバリルールの設定は次のようになります。



ここで使用されているキー:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

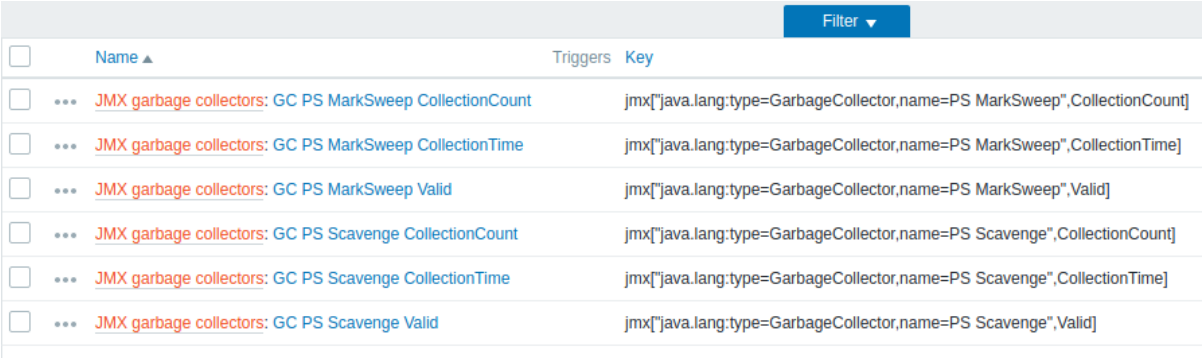
すべてのガベージコレクタが属性なしでディスカバリされます。ガベージコレクタは同じ属性セットを持っているため、アイテムプロトタイプで必要な属性を次のように使用できます。



ここで使用されているキー:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

LLD ディスカバリルールの結果は次のようになります（2 つのガベージコレクタのアイテムがディスカバリされます）。

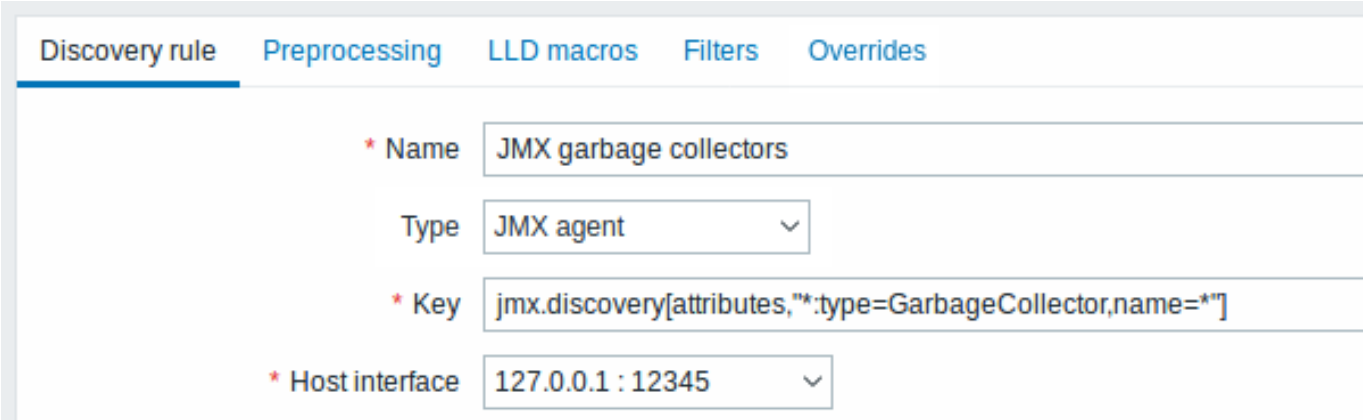


例 2: MBean 属性のディスカバリ

このルールは、以下のフィールドを持つ 9 つのオブジェクトを返します: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attrib MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3。

オブジェクトの詳細については、サポートされているマクロの表の「*MBean* 属性のディスカバリ」セクションを参照してください。

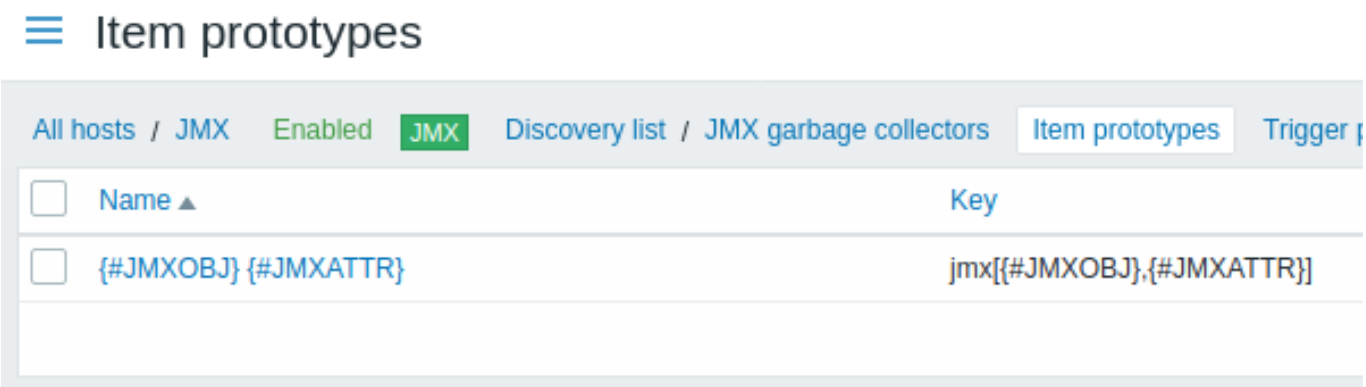
MBean 属性を収集するディスカバリルールの設定は以下のようになります:



ここで使用されているキー:

`jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]`

単一のアイテム属性を持つすべてのガベージコレクタがディスカバリされます。



この特定のケースでは、すべての MBean 属性ごとにプロトタイプからアイテムが作成されます。この設定の主な欠点は、すべての属性に対してアイテムプロトタイプが 1 つしかないため、トリガープロトタイプからトリガーを作成することができないことです。そのため、このセットアップはデータ収集には使用できますが、自動監視には推奨されません。

jmx.get の使用

`jmx.get[]` は `jmx.discovery[]` アイテムと似ていますが、Java オブジェクトのプロパティを低レベルディスカバリーマクロ名に変換しないため、ハイフンや非 ASCII 文字など、LLD マクロ名生成に関連する制限なしに値を返すことができます。

ディスカバリーに `jmx.get[]` を使用する場合、低レベルディスカバリーマクロは、ディスカバリールールのカスタムLLD マクロタブで個別に定義でき、必要な値を指すために JSONPath を使用します。

MBeans の検出

検出アイテム: `jmx.get[beans,"com.example:type=*,*"]`

レスポンス:

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

MBean 属性の検出

検出アイテム: `jmx.get[attributes,"com.example:type=*,*"]`

レスポンス:

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 IPMI センサーの検出

概要

IPMI センサーを自動的に検出することが可能です。

これを行うには次の組み合わせを使用します。

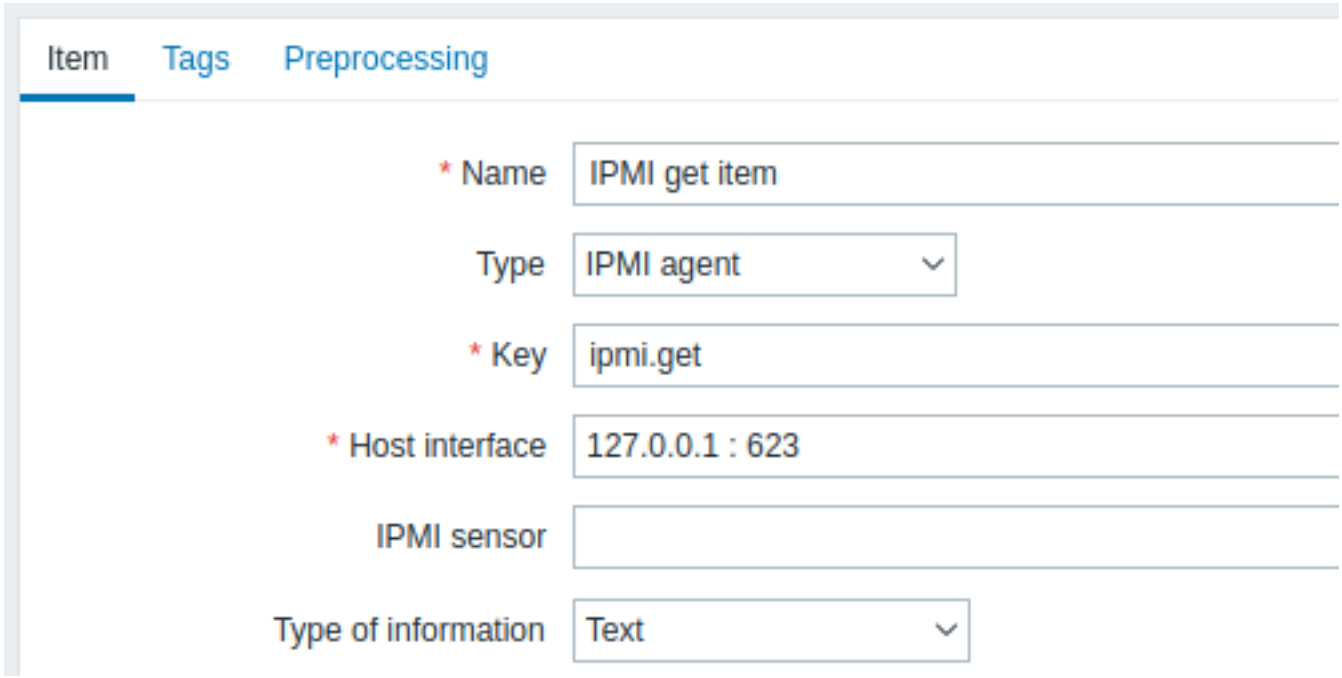
- マスターアイテムとしての `ipmi.get`IPMI アイテム
- 依存するローレベルディスカバリルールとアイテムのプロトタイプ

設定

マスターアイテム

次のキーを使用して IPMI アイテムを作成します。

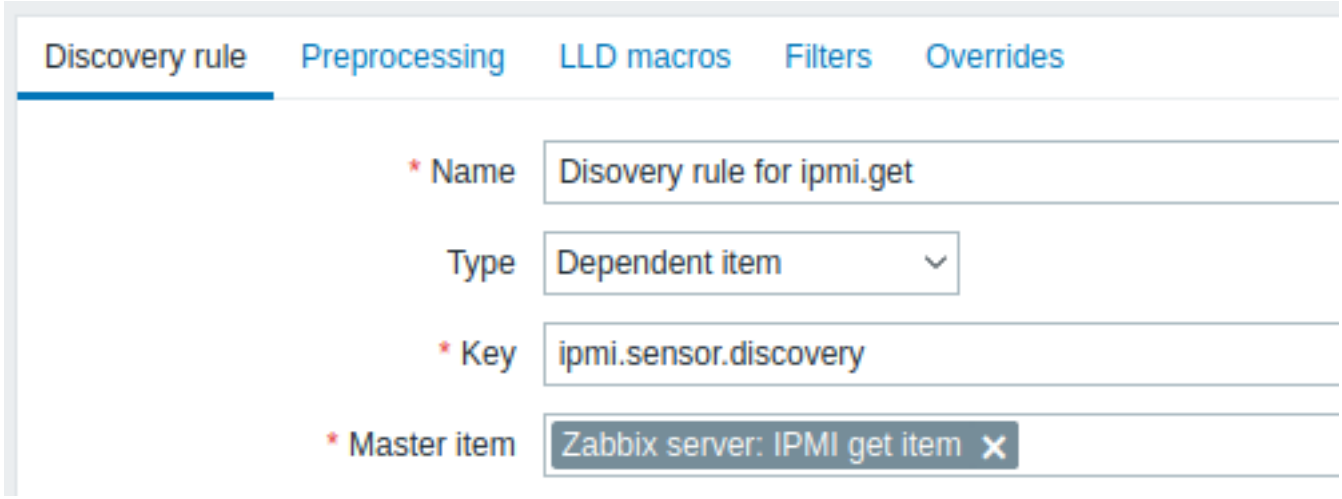
`ipmi.get`



JSON データが大きくなる可能性がある場合は、データ型を” テキスト ” に設定します。

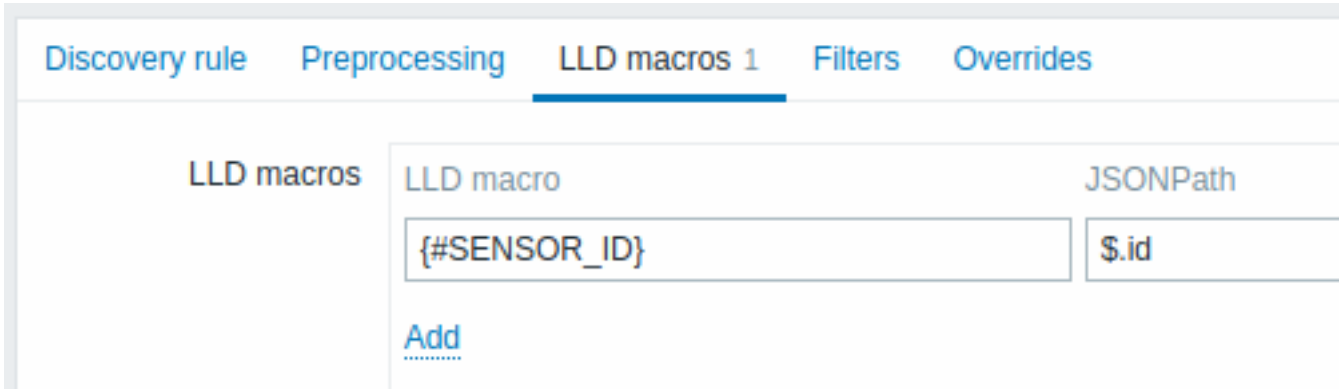
依存 LLD ルール

” 依存アイテム ” タイプとしてローレベルディスカバリルールを作成します。



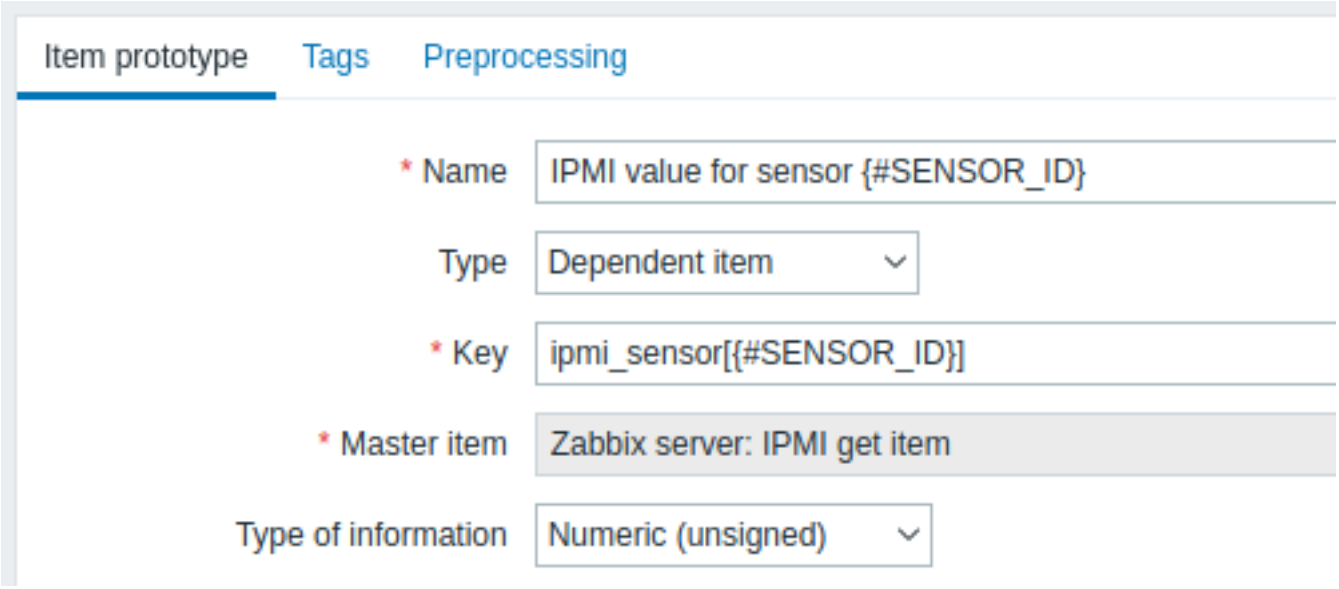
マスターアイテムとして作成した `ipmi.get` アイテムを選択します。

”LLD マクロ ” タブで対応する JSONPath を使用してカスタムマクロを定義します。



依存アイテムのプロトタイプ

この LLD ルールで” 依存アイテム ” タイプのアイテムのプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した `ipmi.get` アイテムを選択します。



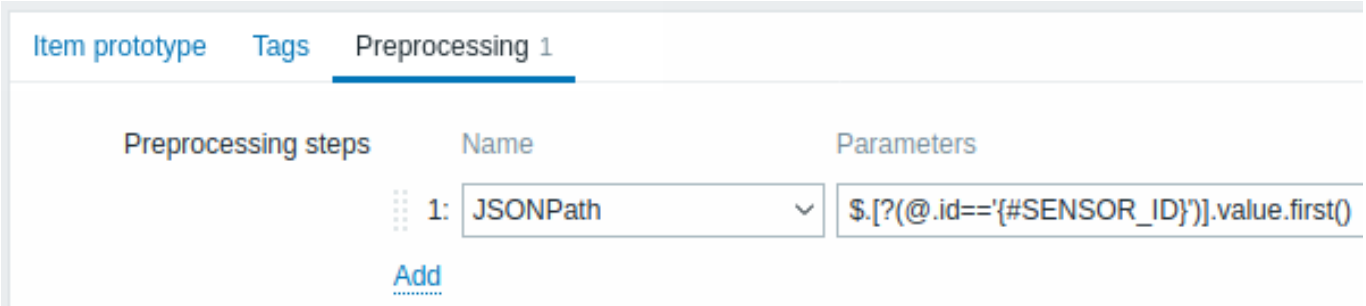
アイテムのプロトタイプ名とキーで {#SENSOR_ID} マクロが使用されていることに注意してください。

- 名前: IPMI value for sensor {#SENSOR_ID}
- キー: ipmi_sensor[{#SENSOR_ID}]

データ型として数値（整数）

アイテムのプロトタイプの” 保存前処理” タブで JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



ディスカバリが開始されると、IPMI センサーごとに 1 つのアイテムが作成されます。このアイテムは、指定されたセンサーの整数値を返します。

8 systemd サービスの検出

概要

Zabbix を使用して systemd ユニット（デフォルトではサービス）を検出することができます。

アイテムキー

ディスカバリールールで使用するアイテムは

```
systemd.unit.discovery
```

> **Attention:**
> このitemキーは Zabbix agent 2 でのみサポートされています。

このアイテムは systemd ユニットに関する情報を含む JSON を返します。例:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
```

```
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

無効な systemd ユニットのディスカバリ

無効な systemd ユニットをディスカバリすることも可能です。この場合、結果の JSON には 3 つのマクロが返されます:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}

Attention:
無効な systemd ユニットのプロトタイプからアイテムやトリガーを作成するには、{#UNIT.ACTIVESTATE} および {#UNIT.UNITFILESTATE} に対する LLD フィルタの禁止条件を調整（または削除）してください。

サポートされているマクロ

以下のマクロは、ディスカバリールールのフィルター およびアイテム、トリガー、グラフのプロトタイプで使用できます。

マクロ	説明
{#UNIT.NAME}	プライマリユニット名
{#UNIT.DESCRIPTION}	人間が読める説明
{#UNIT.LOADSTATE}	ロード状態（ユニットファイルが正常にロードされたかどうか）
{#UNIT.ACTIVESTATE}	アクティブ状態（現在ユニットが開始されているかどうか）
{#UNIT.SUBSTATE}	サブ状態（アクティブ状態のより細かいバージョンで、ユニットタイプに固有。アクティブ状態はそうではない）
{#UNIT.FOLLOWED}	このユニットが状態を追従しているユニットがあればそのユニット名。なければ空文字列
{#UNIT.PATH}	ユニットオブジェクトパス
{#UNIT.JOBID}	ジョブユニットにキューイングされているジョブがあれば数値のジョブ ID。なければ 0
{#UNIT.JOBTYPE}	ジョブタイプ
{#UNIT.JOBPATH}	ジョブオブジェクトパス
{#UNIT.UNITFILESTATE}	ユニットファイルのインストール状態
{#UNIT.SERVICETYPE}	サービスユニットのタイプ（例：simple、forking、oneshot、idle など）。このマクロはユニットがサービスの場合のみ返されます。

アイテムプロトタイプ

systemd サービスディスカバリに基づいて作成できるアイテムプロトタイプには、例えば以下のものがあります。

- アイテム名: {#UNIT.DESCRIPTION} active state info; アイテムキー: systemd.unit.info["{#UNIT.NAME}"]
- アイテム名: {#UNIT.DESCRIPTION} load state info; アイテムキー: systemd.unit.info["{#UNIT.NAME}",LoadState]

9 Windows サービスの検出

概要

ファイルシステムが検出されるのと同様の方法で、Windows サービスも検出できます。

アイテムキー

ディスカバリールールで使用するアイテムは

`service.discovery`

サポートされているマクロ

以下のマクロは、ディスカバリールールのフィルター およびアイテム、トリガー、グラフのプロトタイプで使用できます。

マクロ	説明
{#SERVICE.NAME}	サービス名
{#SERVICE.DISPLAYNAME}	表示されるサービス名
{#SERVICE.DESCRIPTION}	サービスの説明
{#SERVICE.STATE}	サービス状態の数値 詳細はservice.infoアイテムを参照してください。
{#SERVICE.STATENAME}	サービス状態の名前 詳細はservice.infoアイテムを参照してください。
{#SERVICE.PATH}	サービスパス
{#SERVICE.USER}	サービスユーザー
{#SERVICE.STARTUP}	サービスの起動タイプの数値 詳細はservice.infoアイテムを参照してください。
{#SERVICE.STARTUPNAME}	サービスの起動タイプの名前 詳細はservice.infoアイテムを参照してください。
{#SERVICE.STARTUPTRIGGER}	サービスの起動タイプにトリガーがあるかどうかを示す数値 0 - 起動トリガーなし 1 - 起動トリガーあり 自動 (トリガー開始)、自動遅延 (トリガー開始)、手動 (トリガー開始) などのサービス起動タイプを検出するのに便利です。

Windows サービスディスカバリーに基づいて、以下のようなアイテムプロトタイプを作成できます。

`service.info[{#SERVICE.NAME},<param>]`

ここで、param には *state*、*displayname*、*path*、*user*、*startup*、*description* のいずれかの値を指定できます。

例えば、サービスの表示名を取得するには”service.info[{#SERVICE.NAME},displayname]” アイテムを使用できます。param 値が指定されていない場合 (”service.info[{#SERVICE.NAME}]”)、デフォルトで *state* パラメータが使用されます。

10 Windows パフォーマンスカウンターインスタンスの検出

概要

Windows パフォーマンスカウンターのオブジェクトインスタンスを検出することができます。これは、マルチインスタンスのパフォーマンスカウンターに便利です。

アイテムキー

ディスカバリルールを設定するには、次のアイテムを使用します。

- `perf_instance.discovery[object]`

オブジェクト名はローカライズされている場合があることに注意してください。例:

```
perf_instance.discovery[Processor] # オブジェクト名は英語です。
perf_instance.discovery[Processador] # オブジェクト名はポルトガル語です。
```

または、OS のローカライズに関係なく、オブジェクト名が英語で提供されることを保証するには、次のアイテムを使用します。

- `perf_instance_en.discovery[object]`

例:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

サポートされているマクロ

ディスカバリープロセスは、{#INSTANCE} マクロで指定されたオブジェクトのすべてのインスタンスを返します。

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

このマクロは、perf_counter[] および perf_counter_en[] アイテムのプロトタイプで使用できます。

たとえば、ディスカバリールールで使用されるアイテムキーが perf_instance.discovery[Processor] の場合、次のアイテムプロトタイプを作成できます。

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

注意:

- 指定されたオブジェクトが見つからない場合、または可変インスタンスをサポートしていない場合、ディスカバリーアイテムは NOTSUPPORTED になります。
- 指定されたオブジェクトが可変インスタンスをサポートしているが、現在インスタンスが存在しない場合は、空の JSON 配列が返されます。
- 重複するインスタンスはスキップされます。

11 WMI クエリを使用した検出

概要

WMIは Windows の強力なインターフェイスであり、インストールされている Windows コンポーネント、サービス、状態、およびソフトウェアに関するさまざまな情報を取得するために使用できます。

物理ディスクの検出とそのパフォーマンスデータの収集、ネットワークインターフェイスの検出、Hyper-V ゲストの検出、Windows サービスの監視など WindowsOS の他の多くのことに使用できます。

このタイプのローレベルディスカバリは、結果がローレベルディスカバリに適した JSON オブジェクトに自動的に変換される WQL クエリを使用して実行されます。

アイテムキー

ディスカバリールールで使用するアイテムは

wmi.getall[<namespace>,<query>]

このアイテムは、クエリ結果を JSON 配列に変換します。例えば:

```
select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'
```

は、次のような結果を返す場合があります:

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
```

```
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```


ローレベルディスカバリマクロ

返された JSON にローレベルディスカバリマクロは作成されませんが、これらのマクロは、返された JSON で検出された値を指す JSONPath でカスタム LLD マクロ機能を使用して、追加の手順としてユーザーが定義できます。

その後、マクロを使用して、アイテム、トリガーなどのプロトタイプを作成できます。

12 ODBC SQL クエリを使用した検出

概要

このタイプのローレベルディスカバリは、SQL クエリを使用して実行され、その結果はローレベルディスカバリに適した JSON オブジェクトに自動的に変換されます。

アイテムキー

SQL クエリは「Database monitor」アイテムタイプを使用して実行されます。したがって、動作する「Database monitor」ディスカバリールールを取得するには、ODBC 監視ページのほとんどの手順が適用されます。

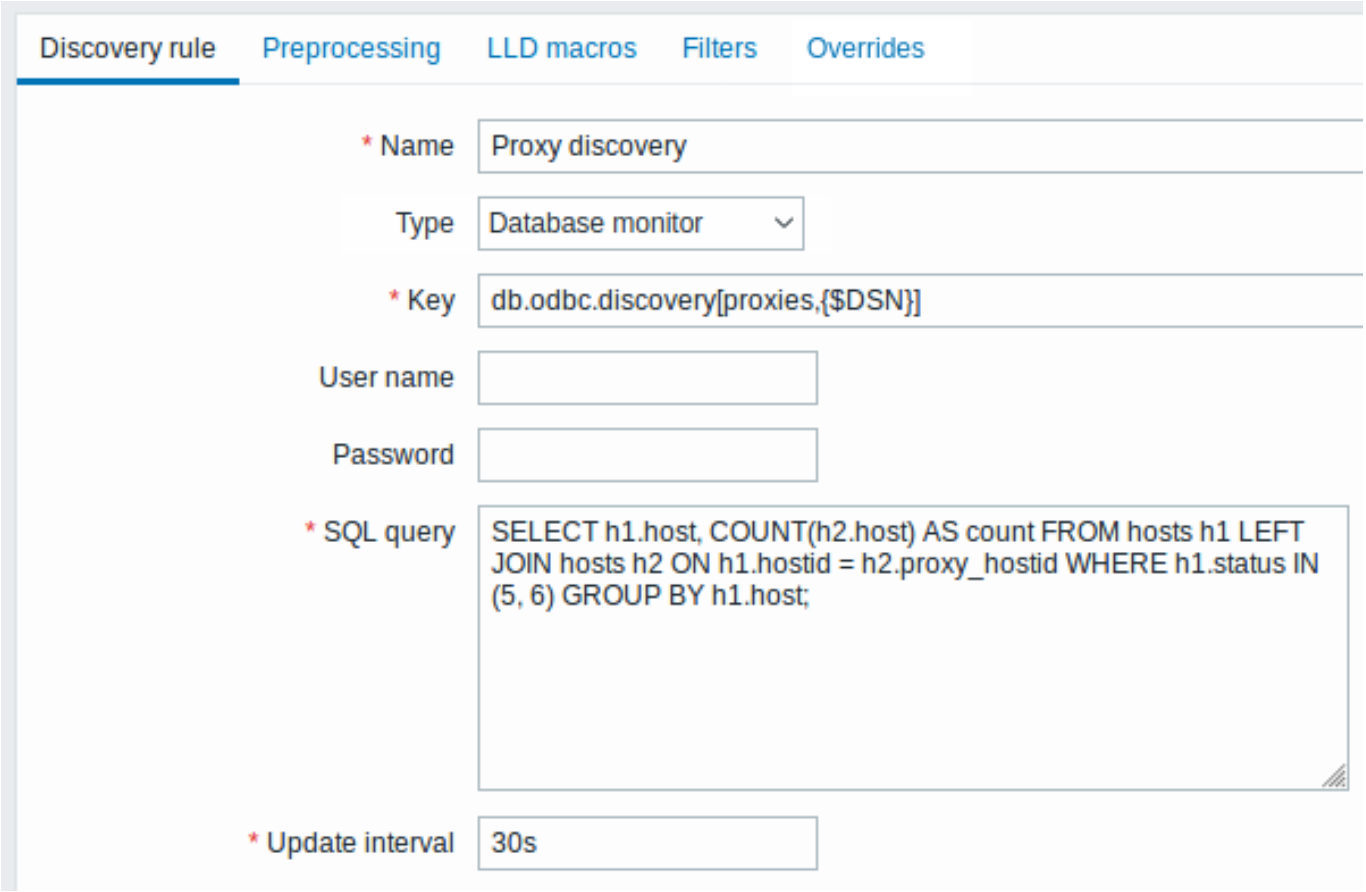
「Database monitor」ディスカバリールールでは、2 つのアイテムキーを使用できます。

- **db.odbc.discovery**[< 一意の短い説明 >,<dsn>,< 接続文字列 >] - このアイテムは、SQL クエリの結果を JSON 配列に変換し、クエリ結果の列名を低レベルディスカバリーマクロ名に変換し、検出されたフィールド値とペアにします。これらのマクロは、アイテム、トリガーなどのプロトタイプ作成時に使用できます。詳細はdb.odbc.discovery の使用も参照してください。
- **db.odbc.get**[< 一意の短い説明 >,<dsn>,< 接続文字列 >] - このアイテムは、SQL クエリの結果を JSON 配列に変換し、クエリ結果の元の列名を JSON のフィールド名として保持し、検出された値とペアにします。`db.odbc.discovery[]` と比較して、このアイテムは返される JSON に低レベルディスカバリーマクロを作成しないため、列名が有効なマクロ名になり得るかどうかを確認する必要はありません。低レベルディスカバリーマクロは、必要に応じてカスタム LLD マクロ機能を使用し、返された JSON 内の検出された値を指す JSONPath を指定して追加ステップとして定義できます。詳細はdb.odbc.get の使用も参照してください。

db.odbc.discovery の使用

次の例は、Zabbix データベース上の ODBC クエリに基づいて、Zabbix プロキシのローレベルディスカバリを使用して SQL クエリが JSON に変換される方法を示しています。これは、どのプロキシが生きているかを監視するために「zabbix[proxy,<name>,lastaccess]」内部アイテムを自動作成するのに便利です。

まずはディスカバリルールの設定から始めます。



必須入力フィールドはすべて赤いアスタリスクでマークされています。

ここでは、Zabbix データベース上で直接次のクエリを使用して、すべての Zabbix プロキシと、それらが監視しているホスト数を選択します。たとえば、ホスト数を使用して空のプロキシを除外することができます。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

”db.odbc.discovery[,{$DSN}]” アイテムの内部動作により、このクエリの結果は自動的に次の JSON に変換されます。

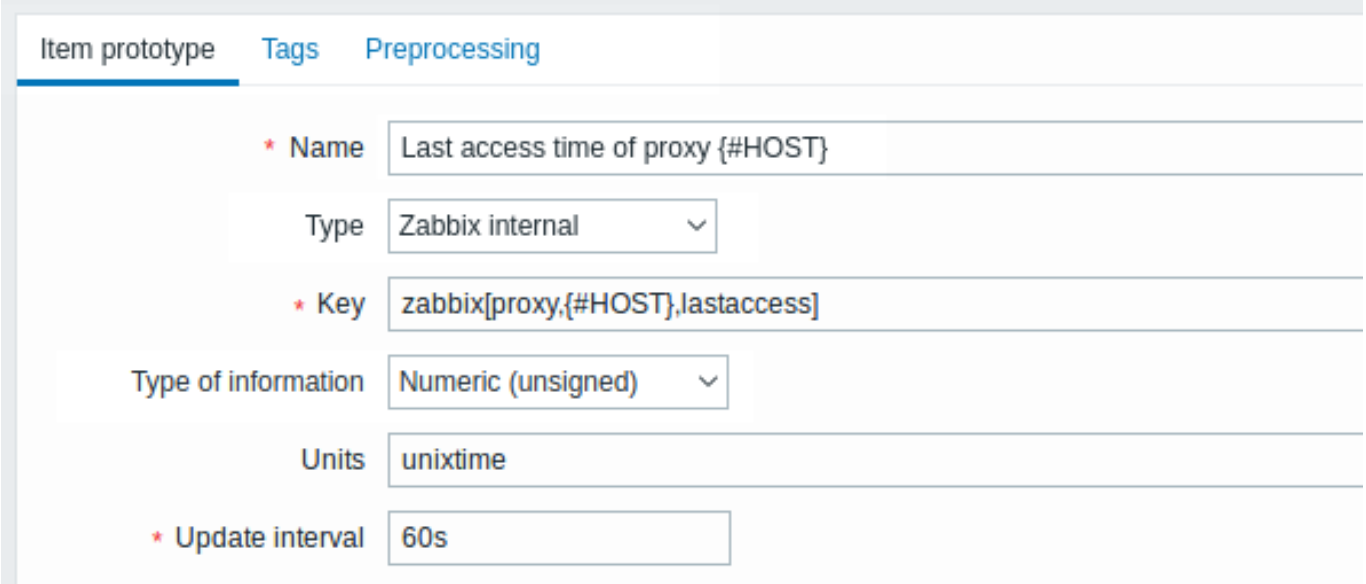
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

カラム名がマクロ名になり、選択された行がこれらのマクロの値になることがわかります。

Note:
カラム名がどのようにマクロ名に変換されるかが明確でない場合は、上記の例のように「COUNT(h2.host) AS count」のようなカラムエイリアスを使用することをお勧めします。
カラム名が有効なマクロ名に変換できない場合、ディスカバリルールはサポートされなくなり、エラーメッセージで問題のあるカラム番号が詳述されます。追加のヘルプが必要な場合は、取得したカラム名が Zabbix サーバーログファイルの DebugLevel=4 で提供されます。

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

SQL クエリがどのように JSON オブジェクトに変換されるかが理解できたので、アイテムプロトタイプで {#HOST} マクロを使用できます。



ディスカバリが実行されると、各プロキシに対してアイテムが作成されます。



db.odbc.get の使用

db.odbc.get[,{$DSN}] と以下の SQL 例を使用します。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

このような JSON が返されます。

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

ご覧の通り、ここには低レベルディスカバリマクロはありません。ただし、LLD マクロタブで、JSONPath を使用してカスタムの低レベルディスカバリマクロを作成できます。例えば:

```
{#HOST} → $.host
```

この {#HOST} マクロはアイテムプロトタイプで使用できます。

item_prototype_odbc.png

13 Prometheus データを使用した検出

概要

Prometheus ラインフォーマットで提供されるデータは、ローレベルディスカバリに使用できます。

Zabbix での Prometheus データクエリの実装方法の詳細についてはPrometheus チェックを参照してください。

設定

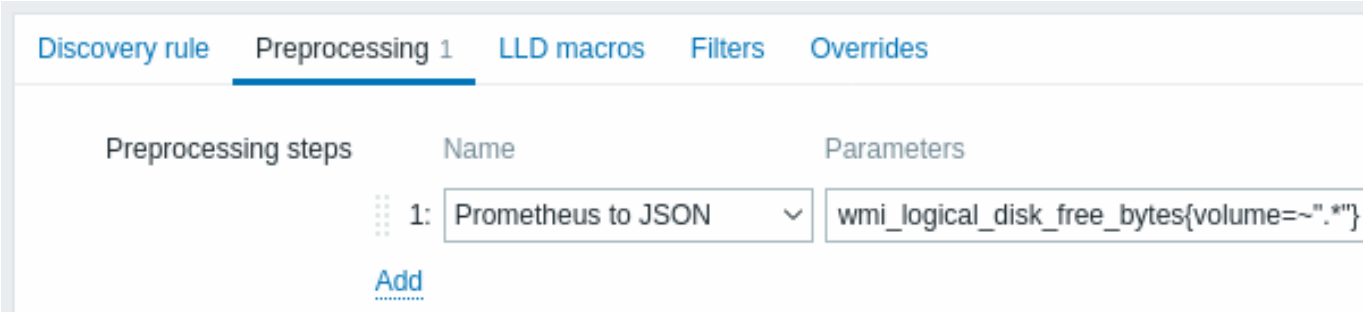
ローレベルディスカバリルールは、Prometheus データを収集する HTTP マスターアイテムへの依存アイテムとして作成する必要があります。

プロメテウスから JSON

ディスカバリルールで [保存前処理] タブに移動し、*Prometheus から JSON* オプションを選択します。ディスカバリには JSON 形式のデータが必要であり、*Prometheus から JSON* オプションは、次の属性を使用してそれを返します。

- metric name
- metric value
- help (if present)
- type (if present)
- labels (if present)
- raw line

例 : `wmi_logical_disk_free_bytes` をクエリします。



これらのプロメテウスラインから :

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

以下のように返します


```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

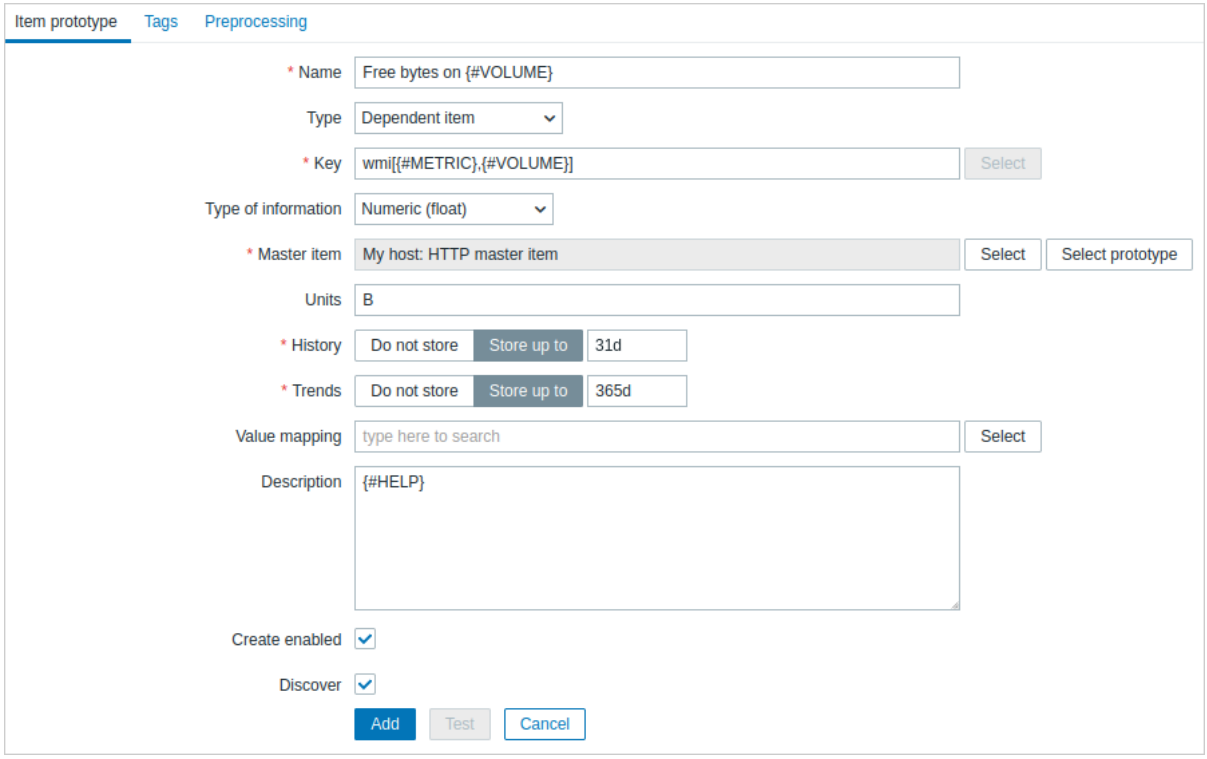

LLD マクロマッピング

次に LLD マクロタブに移動して次のマッピングを行います。

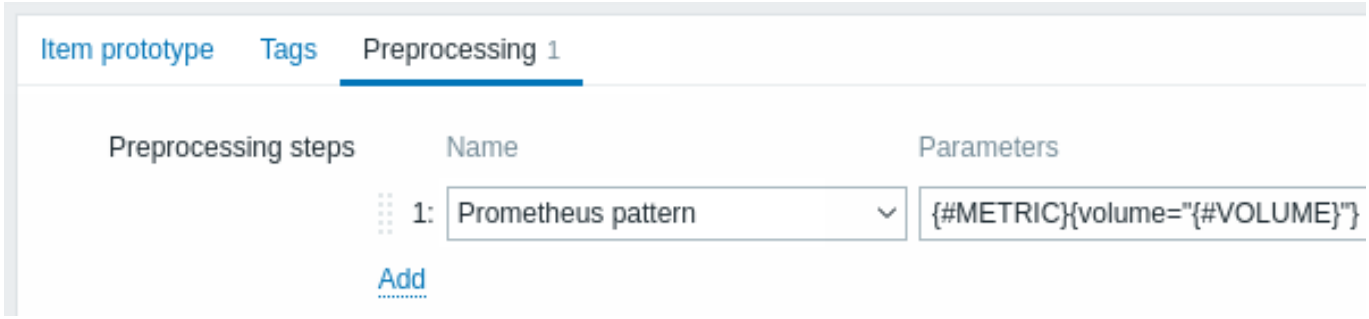
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

アイテムのプロトタイプ

次のようなアイテムのプロトタイプを作成することをお勧めします。



保存前処理オプション:



14 ブロックデバイスの検出

ファイルシステムが検出されるのと同様の方法で、ブロックデバイスとそのタイプも検出できます。

アイテムキー

ディスカバリールールで使用するアイテムキーは

`vfs.dev.discovery`

このアイテムは Linux プラットフォームでのみサポートされています。

このディスカバリーアイテムを使用してディスカバリールールを作成し、以下のようにフィルタリングできます:

- フィルタ: **{#DEVNAME}** が `sd[\D]$` に一致 - ”sd0”、”sd1”、”sd2” などのデバイスを検出する
- フィルタ: **{#DEVTYPE}** が `disk` に一致し、かつ **{#DEVNAME}** が `^loop.*` に一致しない - 名前が”loop” で始まらないディスクタイプのデバイスを検出する

サポートされているマクロ

このディスカバリキーは、ブロックデバイスの名前とタイプをそれぞれ識別する {#DEVNAME} と {#DEVTYPE} の 2 つのマクロを返します。次に例を示します。

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
```

```
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

ブロックデバイスの検出では、vfs.dev.read[] および vfs.dev.write[] アイテムと {#DEVNAME} マクロを使用してアイテムのプロトタイプを作成できます。次に例を示します。

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} は、デバイスのフィルタリングを目的としています。

15 Zabbix でのホストインターフェース検出

概要

ホストの Zabbix フロントエンドで設定されているすべてのインターフェースを検出することができます。

アイテムキー

ディスカバリールールで使用するアイテムは

zabbix[host,discovery,interfaces]

という内部アイテムです。

このアイテムは、インターフェースの説明を含む JSON を返します。内容は以下の通りです。

- IP アドレスまたは DNS ホスト名（ホストの「接続先」設定による）
- ポート番号
- インターフェースタイプ（Zabbix エージェント、SNMP、JMX、IPMI）
- デフォルトインターフェースかどうか
- バルクリクエスト機能が有効かどうか（SNMP インターフェースのみ）

例:

```
[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG
```

複数のインターフェースがある場合、JSON 内のレコードは以下の順序で並びます。

- インターフェースタイプ
- デフォルト - デフォルトインターフェースが非デフォルトインターフェースより前
- インターフェース ID（昇順）

サポートされているマクロ

以下のマクロは、ディスカバリールールのフィルター およびアイテム、トリガー、グラフのプロトタイプで使用できます。

マクロ	説明
{#IF.CONN}	インターフェースの IP アドレスまたは DNS ホスト名
{#IF.IP}	インターフェースの IP アドレス
{#IF.DNS}	インターフェースの DNS ホスト名
{#IF.PORT}	インターフェースのポート番号
{#IF.TYPE}	インターフェースのタイプ（"AGENT"、"SNMP"、"JMX"、"IPMI" のいずれか）
{#IF.DEFAULT}	インターフェースのデフォルト状態: 0 - デフォルトインターフェースではない 1 - デフォルトインターフェース
{#IF.SNMP.BULK}	インターフェースの SNMP バルク処理状態: 0 - 無効 1 - 有効 このマクロはインターフェースタイプが"SNMP" の場合のみ返されます。

概要

完全にカスタムな LLD ルールを作成し、任意のタイプのエンティティ（例えば、データベースサーバー上のデータベースなど）をディスカバリすることも可能です。

そのためには、見つかったオブジェクトとオプションでそのプロパティを指定する JSON 文字列を返すカスタムアイテムを作成する必要があります。エンティティごとのマクロの数に制限はありません。組み込みのディスカバリルールでは 1 つまたは 2 つのマクロ（例えば、ファイルシステムディスカバリでは 2 つ）が返されますが、より多くのマクロを返すことも可能です。

例

必要な JSON 文字列のフォーマットは、例で示すのが最も分かりやすいです。たとえば、古い Zabbix 1.8 エージェント（vfs.fs.discoveryキーをサポートしていないもの）を実行しているが、それでもファイルシステムをディスカバリする必要があるとします。以下は、Linux でマウントされているファイルシステムをディスカバリし、ファイルシステム名とタイプの両方を含む JSON で出力する簡単な Perl スクリプトです。これを「vfs.fs.discovery_perl」というキーの UserParameter として使用する方法があります。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"{#FSNAME}\":\"$fsname\",\n";
    print "\t\t\"{#FSTYPE}\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Attention:
LLD マクロ名で使用できる記号は **0-9**、**A-Z**、_、**.** です。名前に小文字はサポートされていません。

その出力例（見やすく整形済み）を以下に示します。カスタムディスカバリチェック用の JSON は、同じフォーマットに従う必要があります。

```
[
    { "{#FSNAME}":"/",                          "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                       "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                      "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                       "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",                   "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",                   "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                      "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

前述の例では、キーがプロトタイプで使用される LLD マクロ名と一致する必要があります。代わりに、JSONPath を使用して LLD マクロ値を抽出することもできます。たとえば、{#FSNAME} → $.fsname、{#FSTYPE} → $.fstype とし、以下のようなスクリプトも可能です。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\":\"$fsname\",\n";
    print "\t\t\"fstype\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

その出力例（見やすく整形済み）を以下に示します。

```
[
    { "fsname":"/",                          "fstype":"rootfs"   },
    { "fsname":"/sys",                       "fstype":"sysfs"    },
    { "fsname":"/proc",                      "fstype":"proc"     },
    { "fsname":"/dev",                       "fstype":"devtmpfs" },
    { "fsname":"/dev/pts",                   "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",               "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                   "fstype":"tmpfs"    },
    { "fsname":"/home",                      "fstype":"ext3"     },
    { "fsname":"/tmp",                       "fstype":"ext3"     },
    { "fsname":"/usr",                       "fstype":"ext3"     },
    { "fsname":"/var",                       "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",    "fstype":"fusectl"  }
]
```

その後、ディスカバリルールのフィルターフィールドで、”{#FSTYPE}” をマクロとして、”rootfs|ext3” を正規表現として指定できます。

Note:
カスタム LLD ルールでは、FSNAME/FSTYPE のようなマクロ名を使用する必要はありません。好きな名前を自由に使用できます。JSONPath を使用する場合、LLD 行はオブジェクトである配列要素になりますが、他の配列や値でもかまいません。

ユーザーパラメータを使用する場合、返り値は 16MB に制限されることに注意してください。詳細はLLD 返り値のデータ制限を参照してください。

14 分散監視

概要　Zabbix は、Zabbixプロキシを使用して分散 IT インフラストラクチャを監視する効果的で信頼性の高い方法を提供します。

プロキシは一元化された Zabbix サーバーに代わってローカルでデータを収集し、サーバーにデータを報告することができます。

プロキシの特徴

プロキシを使用するかどうかを選択する際には、いくつかの考慮事項があります。

	プロキシ
軽量	はい
GUI	いいえ

	プロキシ
独立して動作	はい
簡単なメンテナンス	はい
自動 *DB* 作成	はい[1]
ローカル管理	いいえ
組み込みハードウェア対応	はい
一方向 *TCP* 接続	はい
集中管理	はい
通知の生成	いいえ

[1] 自動 DB 作成機能は SQLite でのみ動作します。他のサポートされているデータベースでは手動セットアップが必要です。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

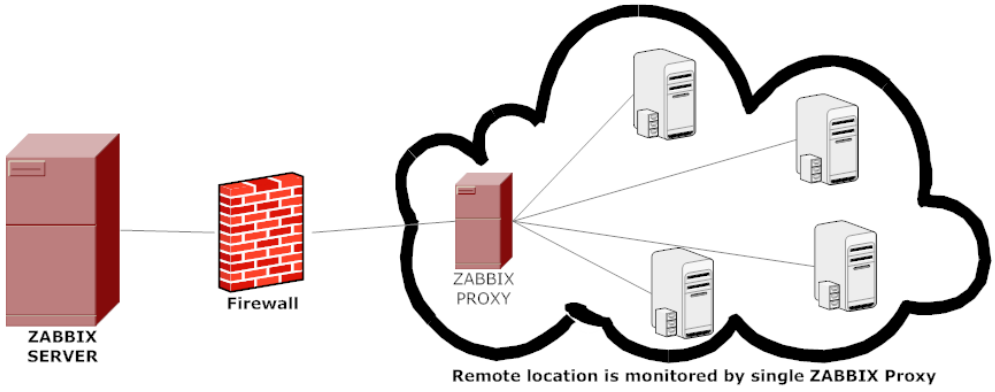
1 プロキシ

概要　Zabbix プロキシは、Zabbix サーバーの代わりにパフォーマンスおよび可用性データを収集できます。このようにして、プロキシはデータ収集の負荷の一部を引き受け、Zabbix サーバーの負荷を軽減できます。

また、プロキシを使用することで、すべてのエージェントとプロキシが 1 つの Zabbix サーバーに報告し、すべてのデータが中央で収集される集中型および分散型監視を最も簡単に実装できます。

Zabbix プロキシは以下の用途に使用できます:

- リモート拠点の監視
- 通信が不安定な拠点の監視
- 数千台のデバイスを監視する際の Zabbix サーバーの負荷軽減
- 分散監視の保守の簡素化



プロキシは Zabbix サーバーへの TCP 接続が 1 つだけ必要です。このため、ファイアウォールを回避するのが容易になり、ファイアウォールルールを 1 つだけ設定すれば済みます。

Attention:
Zabbix プロキシは別のデータベースを使用する必要があります。Zabbix サーバーのデータベースを指定すると、設定が壊れます。

プロキシによって収集されたすべてのデータは、サーバーに送信される前にローカルに保存されます。このため、サーバーとの一時的な通信障害によってデータが失われることはありません。プロキシ設定ファイルの *ProxyLocalBuffer* および *ProxyOfflineBuffer* パラメータで、データをローカルに保持する期間を制御します。

Attention:
Zabbix サーバーのデータベースから直接最新の設定変更を受け取るプロキシが、CacheUpdateFrequencyの値により設定がすぐに更新されない Zabbix サーバーよりも新しい設定を持つ場合があります。その結果、プロキシがデータ収集を開始し、それらを Zabbix サーバーに送信しても、サーバーがそれらのデータを無視することがあります。

Zabbix プロキシはデータ収集専用です。トリガーの計算、イベントの処理、アラートの送信は行いません。プロキシの機能概要については、以下の表を参照してください:

機能		プロキシによるサポート
アイテム		
	Zabbix エージェントチェック	はい
	Zabbix エージェントチェック（アクティブ）	はい [1]
	シンプルチェック	はい
	トラッパーアイテム	はい
	SNMP チェック	はい
	SNMP トラップ	はい
	IPMI チェック	はい
	JMX チェック	はい
	ログファイル監視	はい
	内部チェック	はい
	SSH チェック	はい
	Telnet チェック	はい
	外部チェック	はい
	依存アイテム	はい
	スクリプトアイテム	はい
	ブラウザーアイテム	はい
組み込み Web 監視		はい
アイテム値の事前処理		はい
ネットワークディスカバリ		はい
アクティブエージェントの自動登録		はい
ローレベルディスカバリ		はい [2]
リモートコマンド		はい
トリガーの計算		いいえ
イベントの処理		いいえ
イベントの相関		いいえ
アラートの送信		いいえ

Note:
[1] エージェントがアクティブチェックの問い合わせ先としてサーバーではなくプロキシに問い合わせるようにするには、エージェント設定ファイルの **ServerActive** パラメータにプロキシを指定する必要があります。
 [2] LLD の場合、Zabbix プロキシはデータの収集と事前処理のみを行い、その後 Zabbix サーバーに送信してさらに処理します。

過負荷からの保護

Zabbix サーバーがしばらくダウンしていて、プロキシが大量のデータを収集し、その後サーバーが起動すると、サーバーが過負荷になる可能性があります（ヒストリキャッシュの使用率がしばらくの間 95-100% のままになる）。この過負荷はパフォーマンスの低下を引き起こし、チェックの処理が本来よりも遅くなります。このシナリオによるヒストリキャッシュの過負荷による問題を回避するために、保護機能が実装されています。

Zabbix サーバーのヒストリキャッシュが満杯になると、ヒストリキャッシュへの書き込みアクセスが制限され、サーバーのデータ収集プロセスが停止します。最も一般的なヒストリキャッシュの過負荷のケースは、サーバーダウン後にプロキシが収集したデータをアップロードする場合です。これを回避するためにプロキシのスロットリングが追加されました（現在は無効化できません）。

ヒストリキャッシュの使用率が 80% に達すると、Zabbix サーバーはスロットリングモードに入ります。スロットリングモードでは、サーバーはヒストリキャッシュの使用率が 60% 未満の場合にのみプロキシデータを受け入れ、受け入れるプロキシをローテーションします。ヒストリキャッシュの使用率が 20% 未満になると、サーバーは通常モードに戻ります。

さらに、通常モードでは、Zabbix サーバーはヒストリキャッシュの使用率が 60% を超える場合、非常に大きなパケット（10,000 件以上のレコード）を送信する個々のプロキシをスロットリングします。この判断は、サーバーがプロキシのアップロードを評価する瞬間に適用されるため、ヒストリキャッシュの使用率グラフ（`zabbix[wcache,history,pused]` 内部アイテムとその更新間隔）は短いスパイクを見逃す場合があります。

このスロットリングモードは、キャッシュの使用率が再び 80% に達するか、20% まで下がるか、またはスロットリングリストが空になるまで続きます。最初の場合、サーバーは再びプロキシデータの受け入れを停止します。他の 2 つの場合、サーバーは通常通り動作し、すべてのプロキシからデータを受け入れます。

上記の情報は、次の表で説明できます。

ヒストリ書き込みキャッシュ使用率	Zabbix サーバーモード	Zabbix サーバーの動作
80% に達する	待機	プロキシデータの受け入れを停止し、スロットリングリスト（後で連絡するプロキシの優先リスト）を維持します。
60% に達する	通常だがスロットリングの準備	データ受け入れを判断する際に非常に大きなプロキシアップロード（10k レコード超）を拒否する場合があり、他のプロキシデータの受け入れは継続します。
20% まで下がる	通常	スロットリングリストを破棄し、プロキシデータの受け入れを通常通り開始します。

この Zabbix サーバーの動作をメトリックと相関させるには、zabbix[wcache,history,pused]内部アイテムを使用できます。

設定　プロキシをインストールし、設定したら、Zabbix フロントエンドでプロキシを設定します。

プロキシの追加

Zabbix フロントエンドでプロキシを設定するには:

- 管理 > プロキシに移動します
- プロキシの作成をクリックします

パラメータ		説明
プロキシ名		プロキシ名を入力します。プロキシの設定ファイルの *Hostname* パラメータと同じ名前である必要があります。
プロキシグループ		プロキシの負荷分散/高可用性のために 1 つのプロキシグループを選択します。
アクティブエージェントのアドレス		監視対象のアクティブエージェントまたは送信元が接続する必要があるアドレスを入力します。**Zabbix 7.0** 以降のエージェントのみサポートされています。 このアドレスは、アクティブプロキシとパッシブプロキシの両方への接続に使用されます。このフィールドは、プロキシグループフィールドでプロキシグループが選択されている場合にのみ利用できます。
	アドレス	接続する IP アドレス/DNS 名。
	ポート	接続する TCP ポート番号（デフォルトは 10051）。ユーザーマクロがサポートされています。

パラメータ		説明
プロキシモード		プロキシモードを選択します。 アクティブ - プロキシが Zabbix サーバーに接続し、設定データを要求します パッシブ - Zabbix サーバーがプロキシに接続します 注意: 暗号化されていない通信（機密性の高い）では、アクティブプロキシを使用する場合、Zabbix サーバートラッパーポートにアクセスできる第三者がプロキシの設定データにアクセスできる可能性があります。これは、認証が行われない場合や、プロキシアドレスフィールドでプロキシアドレスが制限されていない場合、誰でもアクティブプロキシを装って設定データを要求できるためです。
プロキシアドレス		指定した場合、アクティブプロキシからのリクエストは、このカンマ区切りの IP アドレス (CIDR 表記も可）またはアクティブ Zabbix プロキシの DNS 名のリストからのみ受け付けます。このフィールドは、プロキシモードフィールドでアクティブプロキシが選択されている場合にのみ利用できます。マクロはサポートされていません。
インターフェース		パッシブプロキシのインターフェース情報を入力します。 このフィールドは、プロキシモードフィールドでパッシブプロキシが選択されている場合にのみ利用できます。
	アドレス	パッシブプロキシの IP アドレス/DNS 名。
	ポート	パッシブプロキシの TCP ポート番号（デフォルトは 10051)。ユーザーマクロがサポートされています。
概要		プロキシの説明を入力します。

暗号化タブでは、プロキシとの暗号化された接続を必須にすることができます。

パラメータ	説明
プロキシへの接続	サーバーがパッシブプロキシに接続する方法: 暗号化なし（デフォルト)、PSK（事前共有鍵）または証明書を使用。
プロキシからの接続	アクティブプロキシから許可する接続タイプを選択します。複数の接続タイプを同時に選択できます（テストや他の接続タイプへの切り替えに便利です)。デフォルトは「暗号化なし」です。
発行者	証明書の許可された発行者。証明書はまず CA（認証局）で検証されます。有効で CA によって署名されている場合、発行者フィールドを使用して許可された CA をさらに制限できます。このフィールドはオプションであり、Zabbix インストールで複数の CA からの証明書を使用している場合に使用します。
サブジェクト	証明書の許可されたサブジェクト。証明書はまず CA で検証されます。有効で CA によって署名されている場合、サブジェクトフィールドを使用してサブジェクト文字列の 1 つの値のみを許可できます。このフィールドが空の場合、設定された CA によって署名された有効な証明書はすべて受け入れられます。
PSK 識別子	事前共有鍵の識別文字列。 PSK 識別子には機密情報を入力しないでください。どの PSK を使用するかを受信側に通知するためにネットワーク上で暗号化されずに送信されます。
PSK	事前共有鍵（16 進文字列)。最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用している場合は 512 桁（256 バイト PSK)、mbed TLS（PolarSSL）ライブラリを使用している場合は 64 桁 (32 バイト PSK)。例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

タイムアウトタブでは、対応するアイテムタイプのグローバルタイムアウトを上書きできます。

パラメータ	説明
アイテムタイプごとのタイムアウト	アイテムのタイムアウト（タイプに基づく）を設定します: グローバル - グローバルタイムアウトを使用（アイテムタイプごとにグレー表示されたタイムアウトフィールドに表示）; 上書き - カスタムタイムアウトを設定（各アイテムタイプのタイムアウトフィールドで）。許容範囲: 1 - 600 秒（デフォルト: グローバルタイムアウトから継承）。時間のサフィックス（例: 30s, 1m）やユーザーマクロがサポートされています。 グローバルタイムアウトリンクをクリックすると、グローバルタイムアウトを設定できます。グローバルタイムアウトリンクは、管理 > 一般フロントエンドセクションへの権限を持つスーパ管理者タイプのユーザーにのみ表示されます。 プロキシレベルのタイムアウトはグローバルタイムアウトを上書きしますが、個別に設定されている場合は、アイテムごとのタイムアウトでさらに上書きされることに注意してください。

Note:
プロキシのメジャーバージョンがサーバーのメジャーバージョンと一致しない場合、アイテムタイプごとのタイムアウトの横に アイコンが表示され、ホバーメッセージ「プロキシとサーバーのバージョンが一致しないため、タイムアウトは無効です」と表示されます。このような場合、プロキシはプロキシ設定ファイルの`Timeout`パラメータを使用します。

既存のプロキシの編集フォームには、以下の追加ボタンがあります:

- 設定の更新 - プロキシの設定を更新
- クローン - 既存のプロキシのプロパティを元に新しいプロキシを作成
- 削除 - プロキシを削除

ホストの設定

ホスト設定フォームの監視方法フィールドで、個々のホストをプロキシまたはプロキシグループで監視するように指定できます。



ホストの一括更新も、ホストをプロキシまたはプロキシグループで監視するように指定する別の方法です。

1 監視設定の同期

概要

このページでは、プロキシの監視設定の更新、つまりサーバーで行われた監視設定の変更がプロキシにどのように同期されるかについて説明します。

インクリメンタルアップデート

プロキシの設定更新はインクリメンタルです。設定の同期時には、変更されたエンティティのみが更新されます（したがって、エンティティが変更されていない場合は何も送信されません）。このアプローチにより、リソースを節約し、プロキシ設定の更新間隔を短く（ほぼ即時）設定することができます。

プロキシ設定の変更はリビジョン番号で追跡されます。プロキシ設定リビジョンより大きいリビジョンを持つエンティティのみが、プロキシに送信される設定データに含まれます。

設定同期の対象となるエンティティは以下の通りです。

エンティティ	詳細
自動登録 *TLS* データ	すべての自動登録 TLS データ。
式	すべての式（正規表現、式テーブル）。
グローバル設定	「config」テーブルで定義されたグローバル設定
ホスト	ホストのすべてのプロパティ、インターフェース、インベントリ、アイテム、アイテムの事前処理、アイテム
ホストマクロ	ホストで定義されたすべてのマクロと、それにリンクされたすべてのテンプレート ID。
プロキシディスカバリールール	プロキシに割り当てられたディスカバリールールとチェック。

つまり、

- ホストのアイテムが変更された場合、そのホストのすべての設定が同期されます。
- 正規表現が変更された場合、すべての正規表現が同期されます。

例外として、ホストマクロは、ホスト上の何かが変更された場合にも送信されます。

プロキシで `-R config_cache_reload` コマンドを実行すると、インクリメンタルアップデートも開始されます。

プロキシの起動/再起動、HA フェイルオーバー、セッショントークンの変更、または設定の受信中に接続が切断されるなど、プロキシで設定の更新に失敗した場合には、完全な設定同期が行われることに注意してください。

設定パラメータ

ProxyConfigFrequency パラメータは、プロキシ設定がサーバーと同期される頻度を決定します（デフォルトは 10 秒）。

ProxyConfigFrequency は以下の通りです:

- パッシブプロキシの場合はサーバーパラメータ
- アクティブプロキシの場合はプロキシパラメータ

アクティブプロキシでは、ProxyConfigFrequency は Zabbix 6.4 からの新しいパラメータであり、現在は非推奨となった ConfigFrequency の代わりに使用する必要があります。

Attention:
ProxyConfigFrequency と ConfigFrequency の両方が使用されている場合、プロキシはエラーを記録して終了します。

メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

2 プロキシの負荷分散と高可用性

概要

Zabbix プロキシは、プロキシグループに編成して、プロキシの負荷分散と高可用性を実現できます。

プロキシの負荷分散と高可用性とは、プロキシグループ内のプロキシ間でホストを自動的に再分配することです。

- プロキシがオフラインになると、そのホストは他のプロキシに移動され、プロキシの高可用性が維持されます。
- あるプロキシのホスト数が他のプロキシよりも大幅に多い/少ない場合、そのホストは他のプロキシに移動され、プロキシの負荷が分散されます。

ホストの再分配は、次の条件を満たすグループ内のプロキシ間でのみ機能します。

- プロキシは Zabbix 7.0 以降を実行している。
- プロキシのバージョンが Zabbix サーバーのバージョンと一致している。Zabbix エージェント（パッシブ）を使用する場合、プロキシのバージョンはエージェントのバージョンと一致している必要があります。アクティブエージェントは Zabbix 7.0 以降であれば十分です。
- プロキシグループがオンライン状態である。
- ホストは、個々のプロキシではなく、プロキシグループによって監視されるように設定されている。

Note:
プロキシグループの健全性は、プロキシグループに割り当てられた任意のホストによって内部チェックで監視できます。ただし、グループ内の単一のプロキシの健全性を監視するには、そのホストをそのプロキシに割り当ててください。そうしないと、結果が一貫しない場合があります。

ホストの再分配

プロキシの負荷分散と高可用性は、プロキシグループマネージャによって Zabbix サーバーが管理します。これは、各プロキシグループ内のすべてのプロキシの状態とホストの分配を継続的に監視します。

グループ内のプロキシの高可用性は、プロキシのフェイルオーバーによって保証されます。プロキシがオフラインになると、そのホストは直ちに他のプロキシに再分配されます。また、ホストが最も少ないプロキシに再割り当てされるため、プロキシの負荷分散も行われます。

さらに、プロキシのホスト数がグループの平均値から少なくとも 10 台かつ 2 倍以上（ホスト過剰またはホスト不足）異なる場合にも、プロキシの負荷分散がトリガーされます。猶予期間（10 x フェイルオーバー遅延）後も不均衡が続く場合、プロキシグループはホストの再分配のためにキューに入れられます。

プロキシグループマネージャは、次のロジックでホストを再分配します。

1. プロキシごとのホストの平均数を計算します。
2. ホストが過剰なプロキシについては、過剰なホストをプロキシの未割り当てプールに移動します。
3. ホストが不足しているプロキシについては、バランスを取るために必要なホスト数を計算します。
4. 最も多くのホストを持つプロキシから必要な数のホストを削除します。
5. 未割り当てのホストを、ホスト数が最も少ないプロキシに移動します。

ホストの再分配の例：

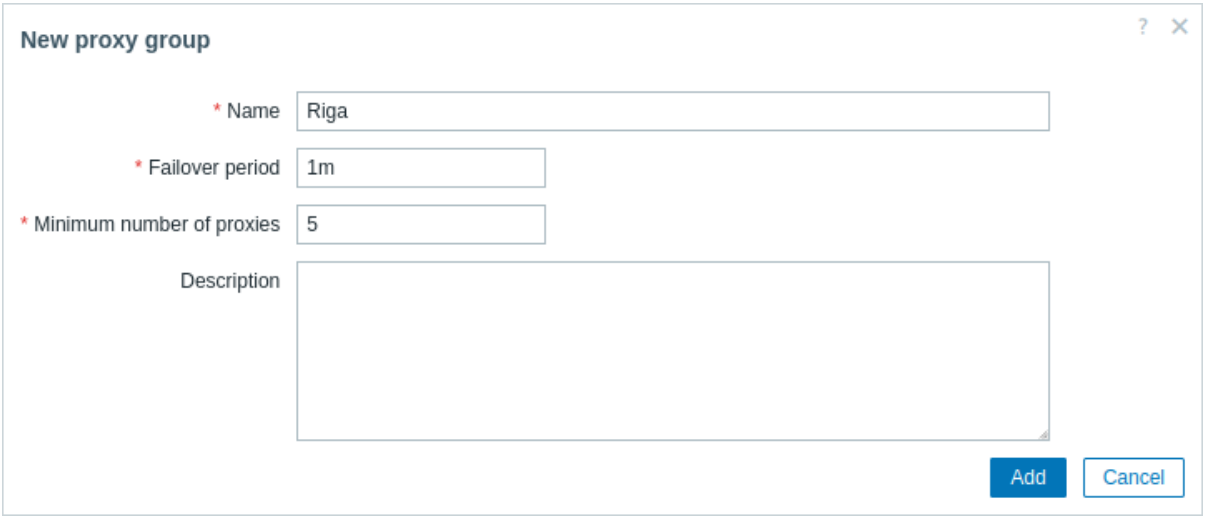
プロキシ上のホスト数	グループ平均	ホストの再割り当て
100	50	はい
60	50	いいえ
40	50	いいえ
25	50	はい
15	5	はい
10	5	いいえ

Attention:
プロキシグループで監視されているホストが 10 台未満の場合、グループ内のプロキシ間でホストの分配が不均等になる可能性があります。

プロキシグループの設定

Zabbix フロントエンドでプロキシグループを設定するには:

1. 管理 > プロキシグループに移動します
2. プロキシグループの作成をクリックします



パラメータ	説明
名前	プロキシグループの名前。
フェイルオーバー期間	プロキシグループ内のプロキシがオンラインと見なされるために Zabbix サーバーと通信しなければならない秒数（デフォルト: 1m; 範囲: 10s–15m）。この期間内にプロキシが通信しない場合、プロキシの状態はオフラインに変更され、そのホストは直ちに他のプロキシに再配分されます。プロキシの負荷分散はこの期間の 10 倍の時間が経過した後に開始されます。 時間のサフィックス（例: 30s, 1m）およびユーザーマクロをサポートします。
最小プロキシ数	プロキシグループをオンラインに保つために必要なオンラインプロキシの最小数（デフォルト: 1; 範囲: 1–1000）。 ユーザーマクロをサポートします。 この値はグループ内のプロキシの総数より少なくする必要があります。例えば、10 台のプロキシのグループで最小値を 10 に設定すると、いずれかのプロキシが障害を起こした場合にグループがオフラインになります。オフライングループ内のオンラインプロキシは通常通り動作し続けますが、負荷分散や高可用性は行われません。
説明	プロキシグループの説明。

パラメータ	説明
プロキシ	プロキシを持つグループを編集する際に、最大 5 つのプロキシのリスト（ユーザーのプロキシに対する権限に応じてリンクまたはプレーンテキスト）を表示します。

プロキシのロードバランシングの設定

プロキシのロードバランシングを使用するには、Zabbix フロントエンドでプロキシグループを設定（上記参照）し、ホストが個々のプロキシではなくプロキシグループによって監視されていることを確認する必要があります（ホストをプロキシからプロキシグループに移動するには、ホストの一括更新を使用できます）。

Zabbix エージェントを使用する場合は、以下のように設定してください。

- パッシブチェックの場合、Serverパラメータにプロキシグループのすべてのプロキシを列挙します。
- アクティブチェックの場合、ServerActiveパラメータにプロキシグループのすべてのプロキシまたは Zabbix サーバーを列挙することを推奨します。アクティブモードでプロキシグループと連携できるのは、Zabbix エージェント 7.0 以降のみです。

`ServerActive` パラメータにプロキシグループのプロキシ（または Zabbix サーバー）が 1 つだけ含まれている場合でも、エージェントは正しいプロキシに接続できます。エージェントサービスが起動して指定したプロキシに接続すると、エージェントはグループ内のプロキシ IP の完全なリストと現在の負荷を受信してキャッシュします。その後、アクティブチェックは、プロキシグループ内の現在のプロキシ-ホスト割り当てに基づいて、ホストに対して正しいオンラインプロキシにリダイレクトされます。

Warning:
Zabbix エージェントの `ServerActive` パラメータに単一のプロキシのみを指定している場合、指定したプロキシがオフラインの状態でエージェントが起動/再起動すると、監視データが失われる可能性があります。

Note:
Zabbix senderを使用する場合も、データリクエストはプロキシグループ内の現在のプロキシ-ホスト割り当てに基づいて、ホストに対して正しいオンラインプロキシにリダイレクトされます。ただし、入力ファイルから複数ホストの値を送信する場合は、`-g` オプションを使用して、誤ったプロキシへのデータ送信を防止してください。

Zabbix エージェントは、ファイアウォールを介してプロキシグループ内のすべてのプロキシに接続できる必要があります。そうでない場合、アクティブチェックのリダイレクトやフェイルオーバー時にハングや失敗が発生する可能性があります。例：

- アクティブチェック中に、プロキシがエージェントを別のプロキシにリダイレクトする場合があります。そのプロキシがファイアウォールでブロックされていると、応答待ちの間に通信がハングします。
- 最近リバランスされていない安定した高可用性構成では、エージェントがバックアッププロキシに一度も接続しない場合があります。ファイアウォールルールが変更されてテストされていない場合、フェイルオーバーが失敗する可能性があります。

プロキシの負荷分散のテスト

プロキシの負荷分散をテストするには:

1. プロキシグループを設定します。
2. プロキシグループがオンライン状態であることを確認します。
3. ホストが個々のプロキシではなく、プロキシグループによって監視されていることを確認します（ホストをプロキシからプロキシグループに移動するには、ホストの一括更新を使用できます）。
4. 設定の更新とプロキシグループ内のプロキシ間でのホストの分散のために数秒待ちます。管理 > プロキシでホストリストを更新して変更を確認します。

重要な注意事項

- SNMP トラップは、プロキシグループ内のプロキシではサポートされていません。
- 外部設定に依存するチェック（外部チェックのスクリプトや、データベースチェックの ODBC 設定など）は、プロキシグループ内のすべてのプロキシで同じ設定にする必要があります。
- データベースチェックには、データベースオブジェクト/サーバーに対する拡張権限が必要です。
- プロキシグループによって監視されているVMware ホストは、グループ内のプロキシ間でランダムに分散されます。これにより、各プロキシがすべての VMware データをキャッシュすることになり、vCenter への追加負荷が発生します。
- プロキシグループ内のプロキシからの自動登録データに基づいて作成されたホストは、そのプロキシグループによって監視されるように設定されます。ただし、プロキシグループ内のプロキシからのネットワークディスカバリデータに基づいて作成されたホストは、そのプロキシによって監視されるように設定されます。

15 暗号化

概要　Zabbix は、暗号化ライブラリに応じて、トランスポート・レイヤー・セキュリティ（TLS）プロトコル v1.2 および 1.3 を使用した Zabbix コンポーネント間の暗号化通信をサポートしています。証明書ベースおよび事前共有鍵ベースの暗号化がサポートされています。

暗号化は以下の接続で設定できます:

- Zabbix サーバー、Zabbix プロキシ、Zabbix エージェント、Zabbix Web サービス、zabbix_sender および zabbix_get ユーティリティ間
- Zabbix フロントエンドおよびサーバー/プロキシから Zabbix データベースへの接続
- Zabbix フロントエンドと Zabbix サーバー間

暗号化はオプションであり、個々のコンポーネントごとに設定できます:

- 一部のプロキシやエージェントは、サーバーとの証明書ベースの暗号化を使用するように設定でき、他のものは事前共有鍵ベースの暗号化を使用し、さらに他のものは（従来通り）暗号化されていない通信を継続できます。
- サーバー（プロキシ）は、異なるホストに対して異なる暗号化設定を使用できます。

Zabbix デーモンプログラムは、暗号化された接続と暗号化されていない接続の両方に対して 1 つのリスニングポートを使用します。暗号化を追加しても、ファイアウォールで新しいポートを開く必要はありません。

制限事項

- 秘密鍵は、起動時に Zabbix コンポーネントが読み取れるファイルに平文で保存されます。
- 事前共有鍵は Zabbix フロントエンドで入力され、Zabbix データベースに平文で保存されます。
- 組み込みの暗号化は、Zabbix フロントエンドを実行している Web サーバーとユーザーの Web ブラウザー間の通信を保護しません。
- 現在、各暗号化接続は完全な TLS ハンドシェイクで開始され、セッションキャッシュやチケットは実装されていません。
- 暗号化を追加すると、ネットワーク遅延に応じてアイテムチェックやアクションの時間が増加します:
  - 例えば、パケット遅延が 100ms の場合、TCP 接続を開いて暗号化されていないリクエストを送信するのに約 200ms かかります。暗号化を使用すると、TLS 接続の確立に約 1000ms が追加されます。
  - タイムアウトを増やす必要がある場合があります。そうしないと、エージェントでリモートスクリプトを実行する一部のアイテムやアクションは、暗号化されていない接続では動作しますが、暗号化された接続ではタイムアウトで失敗する可能性があります。
- ネットワークディスカバリでは暗号化はサポートされていません。ネットワークディスカバリによって実行される Zabbix エージェントチェックは暗号化されず、Zabbix エージェントが暗号化されていない接続を拒否するように設定されている場合、そのようなチェックは成功しません。

暗号化サポート付きで **Zabbix** をコンパイルする　暗号化をサポートするには、Zabbix をサポートされているいずれかの暗号ライブラリでコンパイルおよびリンクする必要があります。

- GnuTLS - バージョン 3.1.18 以降
- OpenSSL - バージョン 1.0.1、1.0.2、1.1.0、1.1.1、3.0.x
- LibreSSL - バージョン 2.7.4、2.8.2 でテスト済み:
  - LibreSSL 2.6.x はサポートされていません
  - LibreSSL は OpenSSL の互換代替としてサポートされています。新しい `tls_*()`LibreSSL 固有の API 関数は使用されません。LibreSSL でコンパイルされた Zabbix コンポーネントは PSK を使用できず、証明書のみが使用可能です。

Note:
Zabbix フロントエンドの SSL 設定については、ベストプラクティスを参照してください。

ライブラリは、”configure” スクリプトにそれぞれのオプションを指定して選択します。

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]`（LibreSSL にも使用）

例えば、*OpenSSL* でサーバーとエージェントのソースを設定するには、次のようにします。

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

異なる Zabbix コンポーネントは、異なる暗号ライブラリでコンパイルできます (例：サーバーは *OpenSSL*、エージェントは *GnuTLS*)。

Attention:
事前共有鍵（PSK）を使用する予定の場合は、PSK を使用する Zabbix コンポーネントで *GnuTLS* または *OpenSSL 1.1.0*（またはそれ以降）のライブラリを使用することを検討してください。*GnuTLS* および *OpenSSL 1.1.0* ライブラリは、Perfect Forward Secrecyを備えた PSK 暗号スイートをサポートしています。*OpenSSL* ライブラリの古いバージョン（1.0.1、1.0.2c）も PSK をサポートしていますが、利用可能な PSK 暗号スイートは Perfect Forward Secrecy を提供しません。

接続の暗号化管理　Zabbix の接続では以下を使用できます:

- 暗号化なし（デフォルト）
- RSA 証明書ベースの暗号化
- PSK ベースの暗号化

Zabbix コンポーネント間の暗号化を指定するために使用される重要なパラメータが 2 つあります:

- TLSConnect - 発信接続にどの暗号化を使用するかを指定します（暗号化なし、PSK または証明書）
- TLSAccept - 着信接続に許可される接続タイプを指定します（暗号化なし、PSK または証明書）。1 つ以上の値を指定できます。

`TLSConnect` は、Zabbix プロキシの設定ファイル（アクティブモードではサーバーへの接続のみを指定）および Zabbix エージェント（アクティブチェック用）で使用されます。Zabbix フロントエンドでは、TLSConnect に相当するものがデータ収集 → ホスト → < あるホスト > → 暗号化タブのホストへの接続フィールド、および管理 → プロキシ → < あるプロキシ > → 暗号化タブのプロキシへの接続フィールドです。接続のために設定された暗号化タイプで失敗した場合、他の暗号化タイプは試行されません。

`TLSAccept` は、Zabbix プロキシの設定ファイル（パッシブモードではサーバーからの接続のみを指定）および Zabbix エージェント（パッシブチェック用）で使用されます。Zabbix フロントエンドでは、TLSAccept に相当するものがデータ収集 → ホスト → < あるホスト > → 暗号化タブのホストからの接続フィールド、および管理 → プロキシ → < あるプロキシ > → 暗号化タブのプロキシからの接続フィールドです。

通常、着信接続には 1 種類の暗号化のみを設定します。しかし、例えば暗号化なしから証明書ベースに切り替えるなど、ダウンタイムを最小限に抑え、ロールバックの可能性を残したまま暗号化タイプを切り替えたい場合があります。そのためには:

- エージェントの設定ファイルで `TLSAccept=unencrypted,cert` を設定し、Zabbix エージェントを再起動します
- 証明書を使用して zabbix_get でエージェントへの接続をテストします。正常に動作すれば、Zabbix フロントエンドのデータ収集 → ホスト → < あるホスト > → 暗号化タブで、そのエージェントの暗号化をホストへの接続を「証明書」に設定して再設定できます。
- サーバーの設定キャッシュが更新され（ホストがプロキシで監視されている場合はプロキシの設定も更新される）、そのエージェントへの接続が暗号化されます
- すべてが期待通りに動作すれば、エージェントの設定ファイルで `TLSAccept=cert` を設定し、Zabbix エージェントを再起動できます。これでエージェントは証明書ベースの暗号化接続のみを受け付けるようになります。暗号化なしおよび PSK ベースの接続は拒否されます。

同様の方法でサーバーやプロキシでも動作します。Zabbix フロントエンドのホスト設定でホストからの接続が「証明書」に設定されている場合、エージェント（アクティブチェック）および zabbix_sender（トラッパーアイテム）からの証明書ベースの暗号化接続のみが受け付けられます。

ほとんどの場合、着信および発信接続の両方で同じ暗号化タイプ、または暗号化なしを設定します。しかし、技術的には非対称に設定することも可能です。例えば、着信は証明書ベース、発信は PSK ベースの暗号化などです。

各ホストの暗号化設定は、Zabbix フロントエンドのデータ収集 → ホストのエージェント暗号化列に表示されます。例:

例	ホストへの接続	ホストからの許可された接続	ホストからの拒否された接
NONE	暗号化なし	暗号化なし	暗号化、証明書および PSK
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化、証明書ベース	暗号化なしおよび PSK ベー
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化、PSK ベース	暗号化なしおよび証明書ベ
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化なしおよび PSK ベースの暗号化	証明書ベースの暗号化
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化なし、PSK または証明書ベースの暗号化	-

Attention:
接続はデフォルトで暗号化されていません。暗号化は各ホストおよびプロキシごとに個別に設定する必要があります。

zabbix_get と **zabbix_sender** の暗号化機能　zabbix_get とzabbix_sender manpages のマニュアルを参照してください。

暗号スイート　暗号スイートはデフォルトで Zabbix の起動時に内部的に設定されます。

また、GnuTLS および OpenSSL ではユーザーが設定した暗号スイートもサポートされています。ユーザーは設定でセキュリティポリシーに従って暗号スイートを設定できます。この機能の使用は任意です（組み込みのデフォルト暗号スイートも引き続き動作します）。

デフォルト設定でコンパイルされた暗号ライブラリの場合、Zabbix の組み込みルールは通常、以下の暗号スイート（優先度の高い順）となります。

ライブラリ	証明書暗号スイート	PSK 暗号スイート
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

ユーザー設定の暗号スイート　組み込みの暗号スイート選択基準は、ユーザー設定の暗号スイートで上書きできます。

Attention:
ユーザー設定の暗号スイートは、TLS 暗号スイート、そのセキュリティおよびミスの影響を理解し、TLS のトラブルシューティングに慣れている上級ユーザー向けの機能です。

組み込みの暗号スイート選択基準は、以下のパラメータを使用して上書きできます。

上書き範囲	パラメータ	値	説明
証明書用の暗号スイート選択	TLSCipherCert13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 cipher strings（値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の証明書ベースの暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherCert	TLS 1.2 用の有効な OpenSSL cipher strings または有効な GnuTLS priority strings。値はそれぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の証明書ベースの暗号スイート選択基準
PSK 用の暗号スイート選択	TLSCipherPSK13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 cipher strings（値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の PSK ベースの暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。

上書き範囲	パラメータ	値	説明
	TLSCipherPSK	TLS 1.2 用の有効な OpenSSL cipher strings または有効な GnuTLS priority strings。値はそれぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の PSK ベースの暗号スイート選択基準
証明書と PSK の両方の暗号スイートリスト	TLSCipherAll13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 cipher strings（値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherAll	TLS 1.2 用の有効な OpenSSL cipher strings または有効な GnuTLS priority strings。値はそれぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の暗号スイート選択基準

zabbix_getおよびzabbix_senderユーティリティで暗号スイート選択を上書きするには、コマンドラインパラメータを使用します。

- `--tls-cipher13`
- `--tls-cipher`

新しいパラメータはオプションです。パラメータが指定されていない場合は、内部のデフォルト値が使用されます。パラメータが定義されている場合は空にできません。

暗号ライブラリで TLSCipher* 値の設定に失敗した場合、サーバー、プロキシ、またはエージェントは起動せず、エラーが記録されます。

各パラメータがいつ適用されるかを理解することが重要です。

外部接続

最も単純なケースは外部接続です:

- 証明書を使用した外部接続の場合は、TLSCipherCert13 または TLSCipherCert を使用します
- PSK を使用した外部接続の場合は、TLSCipherPSK13 または TLSCipherPSK を使用します
- zabbix_get および zabbix_sender ユーティリティの場合、コマンドラインパラメータ `--tls-cipher13` または `--tls-cipher` を使用できます（暗号化は `--tls-connect` パラメータで明確に指定されます）

着信接続

着信接続の場合は、ルールがコンポーネントや設定に固有であるため、少し複雑です。

Zabbix エージェントの場合:

エージェント接続設定	暗号スイート設定
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix サーバーおよびプロキシの場合:

接続設定	暗号スイート設定
PSK を使用した発信接続	TLSCipherPSK, TLSCipherPSK13
証明書を使用した着信接続	TLSCipherAll, TLSCipherAll13
サーバーに証明書がない場合の PSK を使用した着信接続	TLSCipherPSK, TLSCipherPSK13
サーバーに証明書がある場合の PSK を使用した着信接続	TLSCipherAll, TLSCipherAll13

上記 2 つの表には、いくつかのパターンが見られます:

- TLSCipherAll および TLSCipherAll13 は、証明書ベースおよび PSK ベースの暗号スイートの組み合わせリストが使用される場合のみ指定できます。これが発生するのは 2 つの場合です: 証明書が設定されたサーバー（暗号ライブラリが PSK をサポートしている場合、PSK 暗号スイートは常にサーバーやプロキシで設定されます）、証明書ベースおよび PSK ベースの着信接続の両方を受け入れるように設定されたエージェント
- その他の場合は、TLSCipherCert* および/または TLSCipherPSK* で十分です

以下の表は、TLSCipher* の組み込みデフォルト値を示しています。これらは独自のカスタム値を設定する際の良い出発点となります。

パラメータ	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

パラメータ	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] デフォルト値は、古い OpenSSL バージョン（1.0.1、1.0.2、1.1.0）、LibreSSL、および OpenSSL が PSK サポートなしでコンパイルされている場合は異なります。

ユーザー設定の暗号スイートの例

以下に、ユーザー設定の暗号スイートの例を示します:

- 暗号文字列のテストと PFS 暗号スイートのみの許可
- AES128 から AES256 への切り替え

暗号文字列のテストと PFS 暗号スイートのみの許可

どの暗号スイートが選択されたかを確認するには、設定ファイルで'DebugLevel=4' を設定するか、zabbix_sender に-vv オプションを使用する必要があります。

目的の暗号スイートを得るには、TLSCipher* パラメータでいくつか試行錯誤が必要な場合があります。TLSCipher* パラメータを調整するためだけに Zabbix サーバー、プロキシ、エージェントを何度も再起動するのは不便です。より便利な方法は、zabbix_sender や openssl コマンドを使用することです。両方の方法を紹介します。

1. zabbix_sender を使用する場合。

例えば/home/zabbix/test.conf というテスト用設定ファイルを作成します。zabbix_agentd.conf ファイルの構文で記述します:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

この例では有効な CA 証明書とエージェント証明書、PSK が必要です。証明書や PSK ファイルのパスや名前は環境に合わせて調整してください。

証明書を使用せず、PSK のみを使用する場合は、より簡単なテストファイルを作成できます:

```
Hostname=nonexisting
ServerActive=nonexisting
```

```
TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

選択された暗号スイートは、zabbix_sender を実行することで確認できます（OpenSSL 1.1.d でコンパイルされた例）:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

ここでデフォルトで選択されている暗号スイートが確認できます。これらのデフォルト値は、古い OpenSSL バージョン（1.0.1 以降）で動作する Zabbix エージェントとの相互運用性を確保するために選択されています。

新しいシステムでは、PFS（Perfect Forward Secrecy）付きの暗号スイートのみを許可するなど、セキュリティを強化することができます。TLSCipher* パラメータを使用して PFS 付きの暗号スイートのみを許可してみましょう。

Attention:
PSK を使用する場合、OpenSSL 1.0.1 および 1.0.2 を使用するシステムとの相互運用性は失われます。証明書ベースの暗号化は動作するはずです。

test.conf 設定ファイルに 2 行追加します:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

そして再度テストします:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

”certificate ciphersuites” と”PSK ciphersuites” のリストが変わっていることがわかります。以前より短くなり、期待通り TLS 1.3 の暗号スイートと TLS 1.2 の ECDHE-* 暗号スイートのみが含まれています。

2. TLSCipherAll および TLSCipherAll13 は zabbix_sender ではテストできません。これらは上記の例で表示される”certificate and PSK ciphersuites” の値には影響しません。TLSCipherAll および TLSCipherAll13 を調整するには、エージェント、プロキシ、サーバーで試行する必要があります。

したがって、PFS 暗号スイートのみを許可するには、最大 3 つのパラメータを

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

zabbix_agentd.conf、zabbix_proxy.conf、zabbix_server_conf に追加する必要があります。それぞれに証明書が設定されており、エージェントにも PSK がある場合です。

Zabbix 環境が PSK ベースの暗号化のみを使用し、証明書を使用しない場合は、1 つだけで十分です:

```
TLSCipherPSK=kECDHEPSK+AES128
```

仕組みが理解できたので、Zabbix 以外でも openssl コマンドで暗号スイートの選択をテストできます。3 つの TLSCipher* パラメータ値すべてをテストしてみましょう:

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

より詳細な出力が必要な場合は、openssl ciphers に-V オプションを付けて使用することもできます:

```
$ openssl ciphers -V EECDH+aRSA+AES128:kECDHEPSK+AES128
           0x13,0x02 - TLS_AES_256_GCM_SHA384  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(256) Mac=AEAD
           0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256
           0x13,0x01 - TLS_AES_128_GCM_SHA256  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(128) Mac=AEAD
           0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
```

```
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA256
0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA1
```

同様に、GnuTLS の優先度文字列もテストできます:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

AES128 から AES256 への切り替え

Zabbix はデータ用のビルトインデフォルトとして AES128 を使用しています。証明書を使用していて、OpenSSL 1.1.1 の
 AES256 に切り替えたい場合を考えてみましょう。

これは、zabbix_server.conf に各パラメータを追加することで実現可能です:

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
証明書関連の暗号スイートのみが使用されますが、より広い相互運用性のために、安全性の低い暗号を含むデフォルト値を避けるために、TLSCipherPSK* パラメータも定義されています。PSK 暗号スイートは、server / proxy で完全に無効にすることはできません。

zabbix_agentd.conf でも:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

1 証明書の利用

概要

Zabbix は、公開または社内の認証局（CA）によって署名された PEM 形式の RSA 証明書を使用できます。

証明書の検証は、事前に設定された CA 証明書に対して実行されます。オプションで、証明書失効リスト（CRL）を使用できます。

各 Zabbix コンポーネントには、1 つの証明書のみを設定できます。

内部 CA のセットアップと運用、証明書要求の生成と署名、証明書の失効については、OpenSSL PKI チュートリアル v2.0などのチュートリアルを参照してください。

証明書の拡張については慎重に検討し、テストしてください。詳細については、X.509 v3 証明書拡張の使用に関する制限を参照してください。

証明書の設定パラメータ

Zabbix コンポーネントで証明書を設定するために、以下の設定パラメータがサポートされています。

パラメータ	必須	説明
TLSCAFile	はい	ピア証明書の検証のための最上位 CA 証明書を含むファイルのフルパス名。 複数のメンバーを持つ証明書チェーンを使用する場合は、下位 CA 証明書を先に、上位 CA 証明書を後に並べてください。 複数の CA の証明書を 1 つのファイルに含めることができます。
TLSCRLFile	いいえ	証明書失効リスト（CRL）を含むファイルのフルパス名。
TLSCertFile	はい	証明書を含むファイルのフルパス名。 複数のメンバーを持つ証明書チェーンを使用する場合は、サーバー、プロキシ、またはエージェント証明書を先に、下位 CA 証明書を次に、上位 CA 証明書を最後に並べてください。
TLSKeyFile	はい	秘密鍵を含むファイルのフルパス名。 このファイルがZabbix ユーザーのみ読み取り可能であることを、適切なアクセス権を設定して確認してください。
TLSServerCertIssuer	いいえ	許可されるサーバー証明書の発行者。
TLSServerCertSubject	いいえ	許可されるサーバー証明書のサブジェクト。

設定例

必要な証明書を設定した後、Zabbix コンポーネントを証明書ベースの暗号化を使用するように設定します。

以下は、設定の詳細な手順です:

- Zabbix サーバー
- Zabbix プロキシ
- Zabbix エージェント

Zabbix サーバー

1. CA 証明書ファイルを準備します。

ピア証明書を検証するために、Zabbix サーバーは最上位の自己署名ルート CA 証明書を含むファイルにアクセスできる必要があります。たとえば、2 つの独立したルート CA からの証明書が必要な場合、それらを/home/zabbix/zabbix_ca_file.crt のファイルに配置します:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
```

```
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Zabbix サーバー証明書/証明書チェーンをファイルに配置します。たとえば、/home/zabbix/zabbix_server.crt に配置します。最初の証明書が Zabbix サーバー証明書で、その後に中間 CA 証明書が続きます:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
```

```
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJMOHw==
-----END CERTIFICATE-----
```

Note:
証明書の検証プロセスに影響を与えないように、クライアント証明書とサーバー証明書の両方で上記の属性のみを使用してください。たとえば、*X509v3 Subject Alternative Name* や *Netscape Cert Type* 拡張が使用されている場合、OpenSSL は暗号化接続の確立に失敗する可能性があります。詳細については、X.509 v3 証明書拡張の使用制限を参照してください。

3. Zabbix サーバーの秘密鍵をファイルに配置します。たとえば、/home/zabbix/zabbix_server.key に配置します:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Zabbix サーバーの設定ファイルで TLS 設定パラメータを編集します:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix プロキシ

1. Zabbix サーバーのセクションで説明されているように、最上位の CA 証明書、Zabbix プロキシ証明書/証明書チェーン、および秘密鍵のファイルを準備します。次に、Zabbix プロキシの設定ファイルで TLSCAFile、TLSCertFile、TLSKeyFile パラメータを適切に編集します。

2. Zabbix プロキシの設定ファイルで追加の TLS パラメータを編集します:

- アクティブプロキシの場合: TLSConnect=cert
- パッシブプロキシの場合: TLSAccept=cert

Note:
プロキシのセキュリティを向上させるために、TLSServerCertIssuer および TLSServerCertSubject パラメータも設定できます。詳細については、許可された証明書発行者およびサブジェクトの制限を参照してください。

最終的なプロキシ設定ファイルの TLS パラメータは次のようになります:

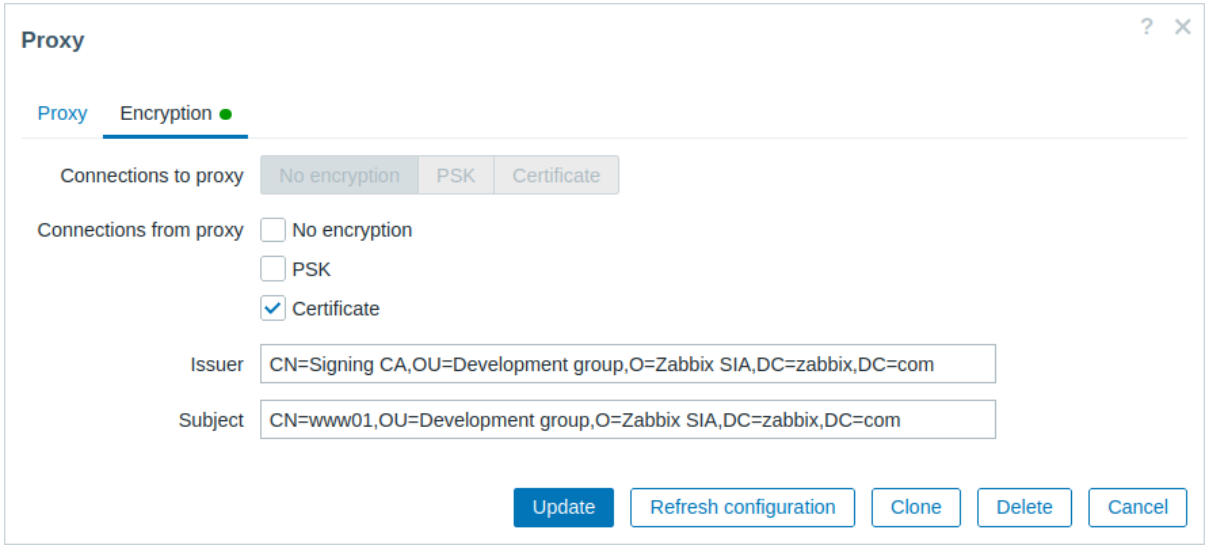
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Zabbix フロントエンドでこのプロキシの暗号化を設定します:

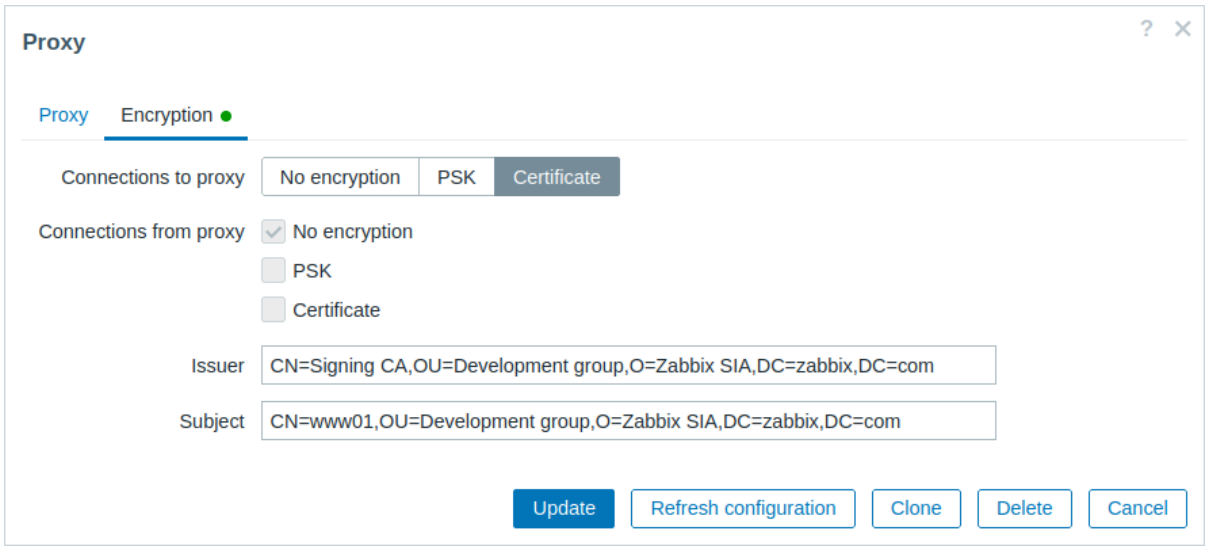
- 管理 → プロキシに移動します。
- プロキシを選択し、暗号化タブをクリックします。

以下の例では、発行者およびサブジェクトフィールドが入力されています。これらのフィールドを使用する理由と方法については、許可された証明書発行者およびサブジェクトの制限を参照してください。

アクティブプロキシの場合:



パッシブプロキシの場合:



Zabbix エージェント

1. Zabbix サーバーのセクションで説明されているように、トップレベルの CA 証明書、Zabbix エージェントの証明書/証明書チェーン、および秘密鍵のファイルを準備します。次に、Zabbix エージェントの設定ファイルで TLSCAFile、TLSCertFile、TLSKeyFile パラメータを適切に編集します。

2. Zabbix エージェントの設定ファイルで追加の TLS パラメータを編集します:

- アクティブエージェントの場合: `TLSConnect=cert`
- パッシブエージェントの場合: `TLSAccept=cert`

> **Note:**
> エージェントのセキュリティを向上させるために、TLSServerCertIssuer および TLSServerCertSubject パラメータを設定できます。詳細については、許可された証明書発行者およびサブジェクトの制限を参照してください。

最終的なエージェント設定ファイルの TLS パラメータは次のようになります。この例では、ホストがプロキシによって監視されていることを前提としているため、証明書の Subject として指定されています:

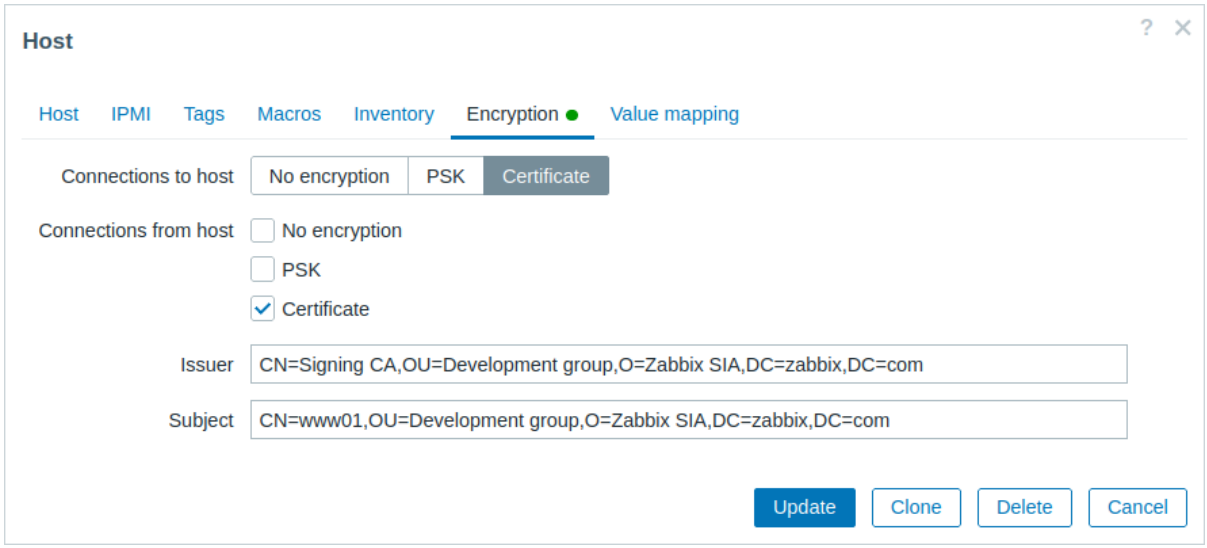
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. このエージェントによって監視されているホストの暗号化を Zabbix フロントエンドで設定します。

- データ収集 → ホストに移動します。

- ホストを選択し、暗号化タブをクリックします。

以下の例では、発行者およびサブジェクトフィールドが入力されています。これらのフィールドを使用する理由と方法の詳細については、許可された証明書発行者およびサブジェクトの制限を参照してください。



Zabbix Web サービス

1. Zabbix サーバーのセクションで説明されているように、トップレベルの CA 証明書、Zabbix Web サービスの証明書/証明書チェーン、および秘密鍵のファイルを準備します。次に、Zabbix Web サービスの設定ファイルの TLSCAFile、TLSCertFile、TLSKeyFile パラメータを適切に編集します。

2. Zabbix Web サービスの設定ファイルで追加の TLS パラメータを編集します: TLSAccept=cert

最終的な Web サービスの設定ファイルの TLS パラメータは次のようになります:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Zabbix サーバーの設定ファイルの WebServiceURL パラメータを編集して、TLS で設定された Zabbix Web サービスに接続するように Zabbix サーバーを設定します:

```
WebServiceURL=https://example.com:443/report
```

許可された証明書発行者およびサブジェクトの制限

2 つの Zabbix コンポーネント（たとえば、サーバーとエージェント）が TLS 接続を確立すると、お互いの証明書を検証します。ピア証明書が信頼された CA（TLSCAFile に事前設定されたトップレベル証明書）によって署名されており、有効であり、有効期限が切れておらず、他のチェックにも合格した場合、コンポーネント間の通信は続行できます。この最も単純なケースでは、証明書の発行者およびサブジェクトは検証されません。

ただし、これにはリスクがあります。つまり、有効な証明書を持つ誰でも他の誰かになりすますことができます（たとえば、ホスト証明書を使用してサーバーになりすますことができます）。証明書が専用の社内 CA によって署名され、なりすましのリスクが低い小規模な環境ではこれで十分かもしれませんが、より大規模またはセキュリティに敏感な環境では不十分な場合があります。

トップレベル CA が Zabbix で受け入れるべきでない証明書を発行する場合、またはなりすましのリスクを減らしたい場合は、発行者およびサブジェクトを指定して許可される証明書を制限できます。

たとえば、Zabbix プロキシの設定ファイルで次のように指定できます。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

これらの設定により、アクティブプロキシは、証明書の発行者またはサブジェクトが異なる Zabbix サーバーとは通信しません。同様に、パッシブプロキシはそのようなサーバーからのリクエストを受け付けません。

Issuer および Subject 文字列のマッチングルール

Issuer および Subject 文字列のマッチングルールは以下の通りです。

- Issuer および Subject 文字列は独立してチェックされます。どちらも省略可能です。
- 文字列が指定されていない場合は、任意の文字列が受け入れられます。
- 文字列はそのまま比較され、完全一致する必要があります。

- UTF-8 文字がサポートされています。ただし、ワイルドカード（*）や正規表現はサポートされていません。
- 以下のRFC 4514要件が実装されています - エスケープが必要な文字（バックスラッシュ'\'、U+005C）：
  - 文字列内の任意の場所: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - 文字列の先頭: スペース（' ', U+0020）または番号記号（'#', U+0023）;
  - 文字列の末尾: スペース（' ', U+0020）。
- ヌル文字（U+0000）はサポートされていません。ヌル文字が検出された場合、マッチングは失敗します。
- RFC 4517およびRFC 4518標準はサポートされていません。

例えば、Issuer および Subject の組織（O）文字列に末尾スペースが含まれ、Subject の組織単位（OU）文字列に二重引用符が含まれている場合、これらの文字はエスケープする必要があります。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development head,O=\ Example SIA\ ,DC=example,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group \"5\",O=\ Example SIA\ ,DC=example,DC=com
```

フィールドの順序とフォーマット

Zabbix はRFC 4514の推奨に従い、これらのフィールドに「逆順」を指定します。つまり、最も下位のフィールド（CN）から始まり、中間レベルのフィールド（OU、O）を経て、最上位のフィールド（DC）で終わります。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

一方、OpenSSL はデフォルトで Issuer および Subject 文字列を上位から下位の順序で表示します。次の例では、Issuer および Subject フィールドは最上位（DC）から始まり、最下位（CN）で終わります。スペースやフィールド区切り文字のフォーマットも使用するオプションによって異なり、Zabbix で必要なフォーマットとは一致しません。

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Issuer および *Subject* 文字列を Zabbix 用に正しくフォーマットするには、OpenSSL を次のオプションで実行します。

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname\
    -in /home/zabbix/zabbix_proxy.crt
```

この出力は逆順でカンマ区切りとなり、Zabbix の設定ファイルやフロントエンドで利用できます。

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

X.509 v3 証明書拡張の使用に関する制限

Zabbix で X.509 v3 証明書を実装する場合、特定の拡張が完全にサポートされていなかったり、一貫性のない動作を引き起こす可能性があります。

Subject Alternative Name 拡張

Zabbix は、IP アドレスやメールアドレスなどの代替 DNS 名を指定するために使用される *Subject Alternative Name* 拡張をサポートしていません。Zabbix は証明書の *Subject* フィールドの値のみを検証できます（許可された証明書発行者およびサブジェクトの制限を参照）。証明書に subjectAltName フィールドが含まれている場合、証明書検証の結果は、Zabbix コンポーネントのコンパイルに使用された暗号化ツールキットによって異なる場合があります。その結果、Zabbix はこれらの組み合わせに基づいて証明書を受け入れたり拒否したりする場合があります。

Extended Key Usage 拡張

Zabbix は *Extended Key Usage* 拡張をサポートしています。ただし、使用する場合は、一般的に *clientAuth*（TLS WWW クライアント認証用）と *serverAuth*（TLS WWW サーバー認証用）の両方の属性を指定する必要があります。例：

- パッシブチェックでは、Zabbix エージェントが TLS サーバーとして動作するため、*serverAuth* 属性をエージェントの証明書に含める必要があります。
- アクティブチェックでは、エージェントが TLS クライアントとして動作するため、*clientAuth* 属性をエージェントの証明書に含める必要があります。

GnuTLS はキー使用違反に対して警告を発する場合がありますが、通常はこれらの警告にもかかわらず通信を許可します。

Name Constraints 拡張

Name Constraints 拡張のサポートは暗号化ツールキットによって異なります。選択したツールキットがこの拡張をサポートしていることを確認してください。この拡張は、使用している特定のツールキットによっては、このセクションがクリティカルとしてマークされている場合に、Zabbix が CA 証明書をロードできなくなる可能性があります。

証明書失効リスト (CRL)

証明書が危険にさらされた場合、認証局 (CA) は証明書失効リスト (CRL) に証明書を含めることで証明書を失効させることができます。CRL は設定ファイルで管理され、サーバー、プロキシ、エージェントの設定ファイルで TLSCRLFile パラメータを使用して指定できます。例:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

この場合、zabbix_crl_file.crt には複数の CA からの CRL を含めることができ、次のようになります:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

CRL ファイルは Zabbix の起動時のみ読み込まれます。CRL を更新するには、Zabbix を再起動してください。

Attention:
Zabbix コンポーネントが OpenSSL でコンパイルされており CRL が使用されている場合、証明書チェーン内の各トップレベルおよび中間 CA に対応する CRL (空でも可) が TLSCRLFile に含まれていることを確認してください。

2 事前共有鍵の使用

概要

Zabbix の各事前共有キー (PSK) は、次のペアです。

- 非秘匿 PSK ID 文字列、
- 秘匿 PSK 文字列値。

PSK ID 文字列は、空ではない UTF-8 の文字列です。たとえば”PSK ID 001 Zabbix agentd” などです。この特定の PSK が Zabbix コンポーネントによって参照される一意の名前です。暗号化されずにネットワーク経由で送信されますので、機密情報を PSK ID 文字列に含めないでください。

PSK 文字列値は推測しにくい 16 進数の文字列です。たとえば、”e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” などです。

サイズ制限

Zabbix には PSK ID と値のサイズ制限があります。場合によっては、暗号化ライブラリの制限がより低くなることがあります。

コンポーネント	PSK ID 最大サイズ	PSK 値最小サイズ	PSK 値最大サイズ
Zabbix	128 UTF-8 文字列	128-bit (16-bytes PSK, 16 進数 32 桁)	2048-bit (256-bytes PSK, 16 進数 512 桁)
GnuTLS	128 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.0.x, 1.1.0	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.1.1	127 bytes (UTF-8 文字列を含む)	-	512-bit (64-byte PSK, 16 進数 128 桁)
OpenSSL 1.1.1a 以降	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)

Attention:
Zabbix フロントエンドでは使用する暗号化ライブラリに関係なく、最大 128 文字の PSK ID 文字列と 2048 ビットの PSK を設定できます。\一部の Zabbix コンポーネントが下限をサポートしている場合、これらのコンポーネントに許可された長さで PSK ID 文字列と PSK 値を設定するのはユーザーの責任です。\長さの制限を超えると、Zabbix コンポーネント間で通信障害が発生します。

Zabbix サーバーが PSK を使用してエージェントに接続する前に、サーバーはそのエージェント用に構成された PSK ID と PSK 値をデータベース (実際には構成キャッシュ) で検索します。接続を受信すると、エージェントは設定ファイルから PSK ID と PSK 値を使用します。双方が同じ PSK ID 文字列と PSK 値を持っている場合、接続が成功します。

Attention:
各 PSK ID は、1 つの値のみとペアにする必要があります。ID 文字列が同じで値が異なる 2 つの PSK が存在しないことを確認するのは、ユーザーの責任です。そうしないと、予測できないエラーが発生したり、この PSK ID 文字列を持つ PSK を使用する Zabbix コンポーネント間の通信が中断したりする可能性があります。

PSK の生成

たとえば、256 ビット (32 バイト) の PSK は、次のコマンドを使用して生成できます。

- with *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- with *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

上記の「psktool」は、PSK ID とそれに関連付けられた PSK を含むデータベースファイルを生成することに注意してください。Zabbix は PSK ファイルに PSK のみを想定しているため、識別文字列とコロン (':') をファイルから削除する必要があります。

サーバーエージェント通信用の PSK 設定 (例)

エージェントホストで PSK 値をファイルに書き込みます (例: `/home/zabbix/zabbix_agentd.psk`)。ファイルには、最初のテキスト文字列に PSK が含まれている必要があります。例えば、

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

エージェント設定ファイル `zabbix_agentd.conf` で TLS パラメーターを編集します。たとえば次のように設定します。

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

エージェントはサーバーに接続し、PSK を使用したサーバー (アクティブチェック) と `zabbix_get` からの接続のみを受けつけます。PSK ID は"PSK 001" です。

エージェント再起動後、`zabbix_get` を使用して接続をテストできます。例えば、

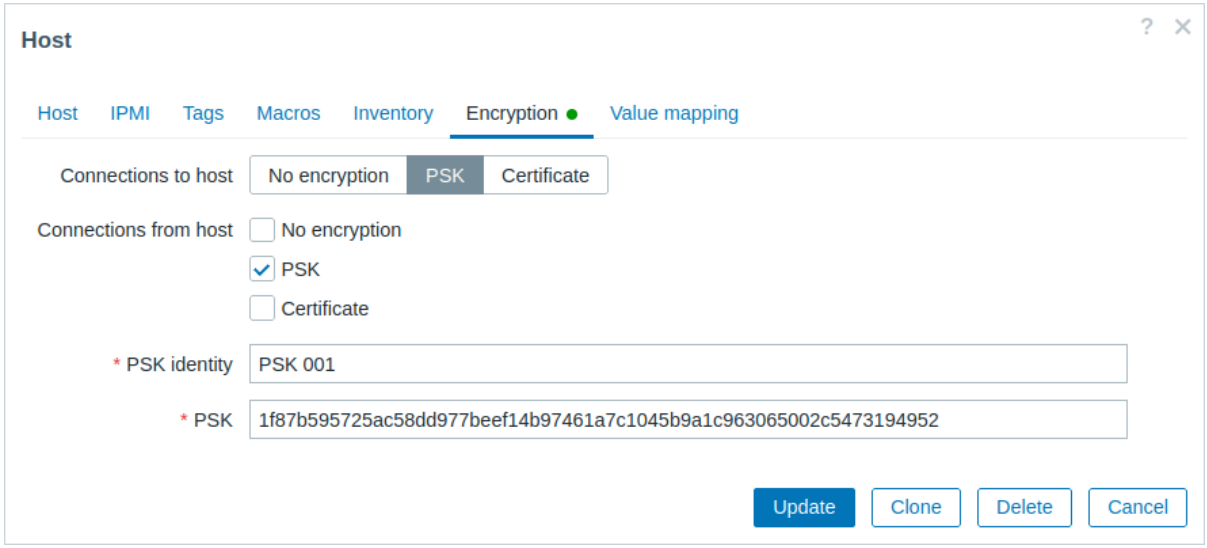
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(ダウンタイムを最小限に抑えるには、接続暗号化の管理の接続タイプを変更する方法を参照してください。)

Zabbix フロントエンドでこのエージェントの PSK 暗号化を設定します。

- データ収集 → ホストに移動
- ホストを選択して暗号化タブをクリック

例:



すべての必須入力フィールドには赤いアスタリスクが付いています。

設定キャッシュがデータベースと同期されると、新しい接続は PSK を使用します。サーバーとエージェントのログファイルでエラーメッセージを確認してください。

サーバーの PSK の構成 - アクティブなプロキシ通信 (例)

プロキシで PSK 値をファイルに書き込みます (例: `/home/zabbix/zabbix_proxy.psk`)。ファイルの最初のテキスト文字列に PSK が含まれている必要があります。次に例を示します。

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

プロキシ構成ファイル `zabbix_proxy.conf` で TLS パラメータを編集します。たとえば、次のように設定します。

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

プロキシは PSK を使用してサーバーに接続します。PSK ID は”PSK 002” になります。

(ダウンタイムを最小限に抑えるには、接続暗号化の管理 で接続タイプを変更する方法を参照してください)。

Zabbix フロントエンドでこのプロキシの PSK を構成します。*[*管理*]*→プロキシ に移動し、プロキシを選択して、暗号化 タブに移動します。” プロキシからの接続” で PSK にチェックします。”PSK アイデンティティ” フィールドに”PSK 002” を貼り付け、”e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” を”PSK” フィールドに入力して、” 更新” をクリックします。

プロキシを再起動して、サーバーへの PSK ベースの暗号化された接続の使用を開始します。サーバーとプロキシのログファイルでエラーメッセージを確認してください。

パッシブプロキシの場合、手順は非常に似ています。唯一の違いは、プロキシ構成ファイルで `TLSAccept=psk` を設定し、Zabbix フロントエンドで” プロキシへの接続” を PSK に設定することです。

3 トラブルシューティング

一般的な推奨事項

- 問題が発生した場合、どのコンポーネントが TLS クライアントとして機能し、どのコンポーネントが TLS サーバーとして機能するかを理解することから始めます。\Zabbix サーバー、プロキシ、およびエージェントは、それらの間の相互作用に応じて、すべて TLS サーバーおよびクライアントとして機能します。\たとえば、Zabbix パッシブチェックのためにエージェントに接続するサーバーは、TLS クライアントとして機能します。エージェントは TLS サーバーの役割を果たします。\Zabbix エージェントは、プロキシからアクティブなチェックのリストを要求し、TLS クライアントとして機能します。プロキシは TLS サーバーの役割を果たします。‘zabbix_get および zabbix_sender‘ ユーティリティは常に TLS クライアントとして機能します。
- Zabbix は相互認証を使用します。 それぞれの側がピアを検証し、接続を拒否する場合があります。 たとえば、エージェントに接続している Zabbix サーバーは、エージェントの証明書が無効な場合、すぐに接続を閉じることができます。逆もまた同様です - サーバーがエージェントによって信頼されていない場合、サーバーからの接続を受け入れる Zabbix エージェントは接続を閉じることができます。

- TLS クライアントと TLS サーバーの両方でログファイルを調べます。\接続を拒否する側は、拒否された正確な理由をログに記録する場合があります。反対側は、かなり一般的なエラーを報告することがよくあります (たとえば、「ピアによって接続が閉じられました」、「接続が正しく終了されませんでした」など)。
- 暗号化の設定を誤ると、実際の原因を示すものではなく、混乱を招くエラーメッセージが表示されることがあります。 以下のサブセクションでは、トラブルシューティングに役立つ可能性のあるメッセージと考えられる原因の (完全ではない) コレクションを提供しようとしています. さまざまな暗号化ツールキット (OpenSSL、GnuTLS) は、同じ問題の状況で異なるエラーメッセージを生成することがよくあります。 場合によっては、エラーメッセージが両側の暗号化ツールキットの特定の組み合わせに依存することもあります。

1 接続タイプまたは権限の問題

サーバーは PSK を使用してエージェントに接続するように設定されているが、エージェントは暗号化されていない接続のみを受け入れる場合

サーバーまたはプロキシログ (*GnuTLS* 3.3.16 を使用)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

サーバーまたはプロキシログ (*OpenSSL* 1.0.2c を使用)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

片側は証明書で接続するが、反対側は PSK のみを受け入れる、またはその逆

任意のログ (*GnuTLS* を使用):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

任意のログ (*OpenSSL* 1.0.2c を使用):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

TLS サポートでコンパイルされた Zabbix センダーを使用して、TLS なしでコンパイルされた Zabbix サーバー/プロキシにデータを送信した場合

接続元ログ:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a  20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

受付側ログ:

```
...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in
```

一方は PSK で接続するが、もう一方は LibreSSL を使用しているか、暗号化サポートなしでコンパイルされている場合

LibreSSL は PSK をサポートしていません。

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

受付側ログ :

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Zabbix フロントエンド:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

一方は PSK で接続しますが、もう一方は PSK サポートを無効にして OpenSSL を使用している場合

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

受付側ログ:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 証明書の問題

CRL で使用される OpenSSL と証明書チェーンの一部の CA では、CRL が TLSCRLFile に含まれていない

OpenSSL ピアの場合の TLS サーバーログ:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

GnuTLS ピアの場合の TLS サーバーログ:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL が期限切れになった、またはサーバー操作中に期限切れになった

OpenSSL サーバーログ:

- 期限切れ前:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- 期限切れ後:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

ここでのポイントは、有効な CRL を使用すると、取り消された証明書が「証明書が取り消された」として報告されることです。CRL の有効期限が切れると、エラーメッセージが「証明書の有効期限が切れました」に変わります。これは誤解を招く可能性があります。

GnuTLS サーバーログ:

- 有効期限の前後で同じ:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

自己署名証明書や不明な CA

OpenSSL ログ:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

これはサーバー証明書が CA によって署名されているにもかかわらず、間違って発行者とサブジェクトの文字列が同じである場合に発生します。最上位の CA 証明書では発行者とサブジェクトは同じですが、サーバー証明書では同じにすることはできません。(プロキシ証明書とエージェント証明書も同様です。)

証明書に同じ発行者とサブジェクトのエントリが含まれているかどうかを確認するには、次を実行します。

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

ルート (トップレベル) 証明書が発行者とサブジェクトの同一の値を持つことは許容されます。

3 PSK の問題

PSK に奇数個の 16 進数が含まれている

プロキシまたはエージェントが起動しない、プロキシまたはエージェントログのメッセージ:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

128 バイトを超える PSK ID 文字列が GnuTLS に渡された

TLS クライアント側のログ:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

TLS サーバー側のログ:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

OpenSSL 1.1.1 で使用される PSK 値が長すぎる

接続側ログ: ...OpenSSL library (version OpenSSL 1.1.1 11 Sep 2018) initialized ... ...In zbx_tls_connect(): psk_identity:"PSK 1" ...zbx_psk_client_cb() requested PSK identity "PSK 1" ...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\extensions_clnt.c line 801: error:14212044:SSL routines:tls_construct_ctos_early_data:internal error: TLS write fatal alert "internal error"'

受付側ログ:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

この問題は通常、OpenSSL を 1.0.x または 1.1.0 から 1.1.1 にアップグレードするときに、PSK 値が 512 ビット (64 バイト PSK、128 の 16 進数として入力) よりも長い場合に発生します。

参照: 値のサイズ制限

16 Web インターフェース

概要　どこからでも、どのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。

> **Note:**
> 複数のフロントエンドインスタンスを使用する場合は、言語や地域の設定とライブラリ (LDAP、SAML など) がすべてのフロントエンドにインストールされ、同じように構成されていることを確認してください。

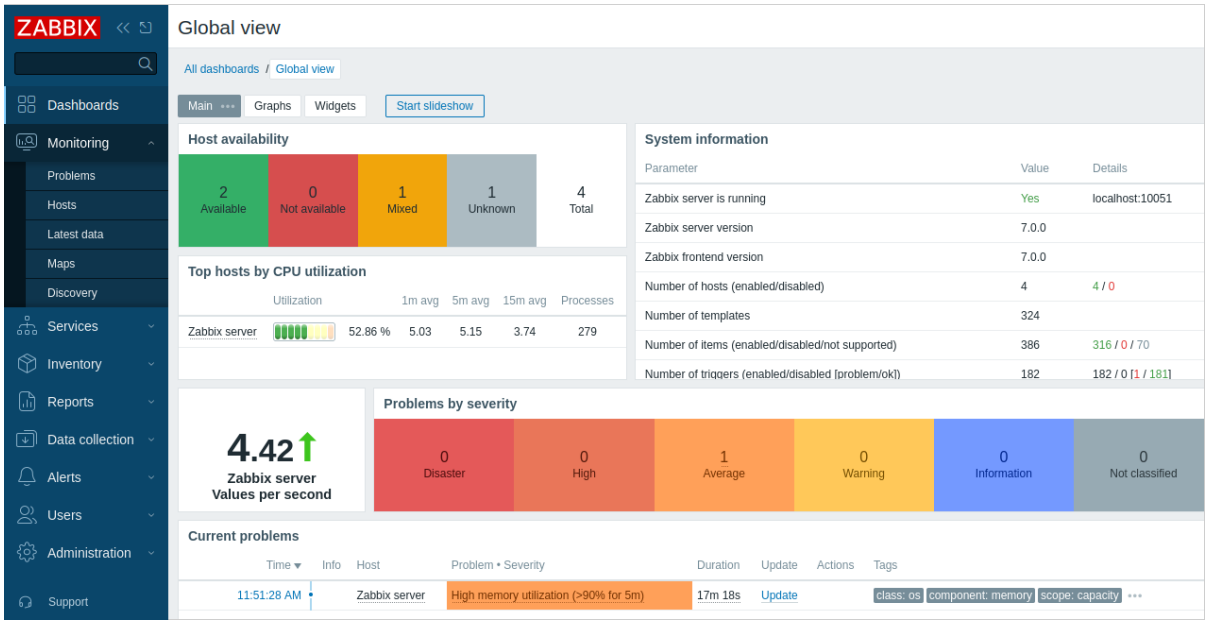
フロントエンドヘルプ　Zabbix フロントエンドフォームにはヘルプリンク ? があり、ドキュメントの対応する部分への直接リンクが提供されます。

1 メニュー

概要

サイドバーのメニューから、さまざまな Zabbix フロントエンドセクションにアクセスできます。

デフォルトのテーマでは、メニューは紺色です。

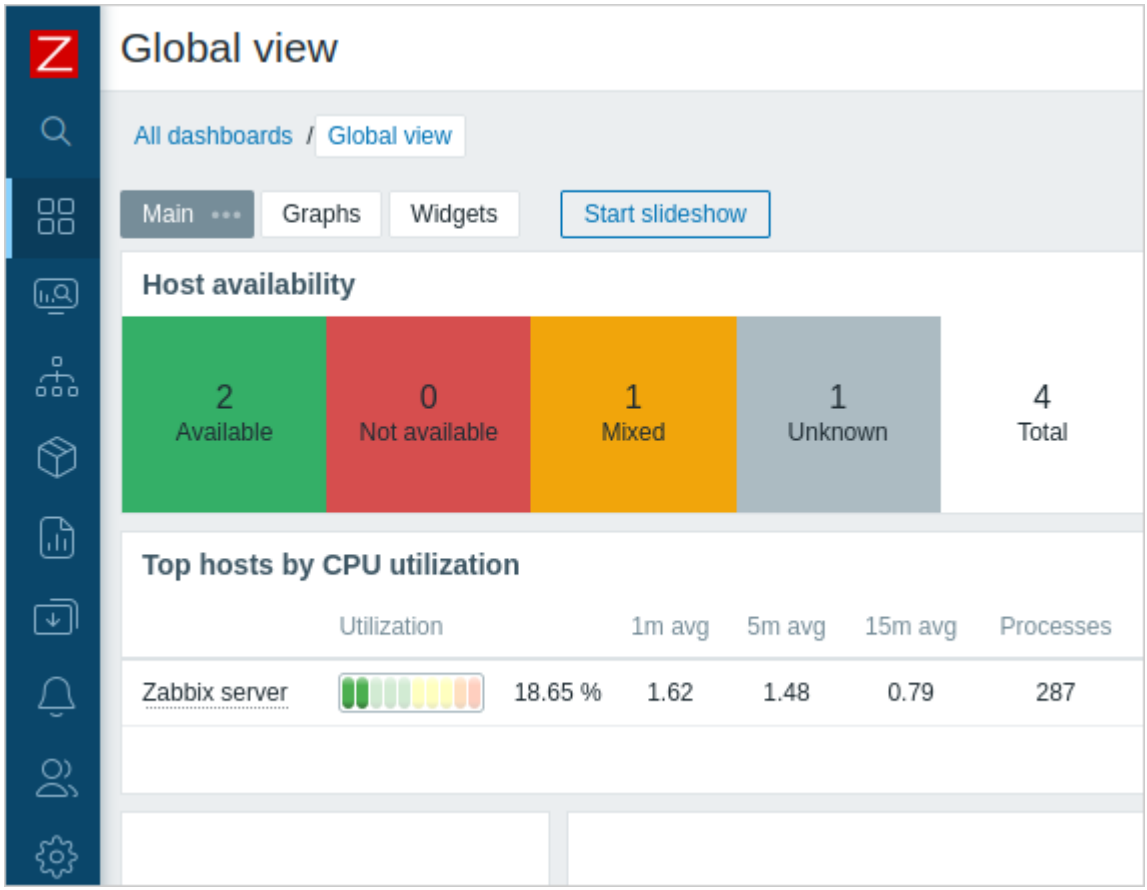


メニューの操作

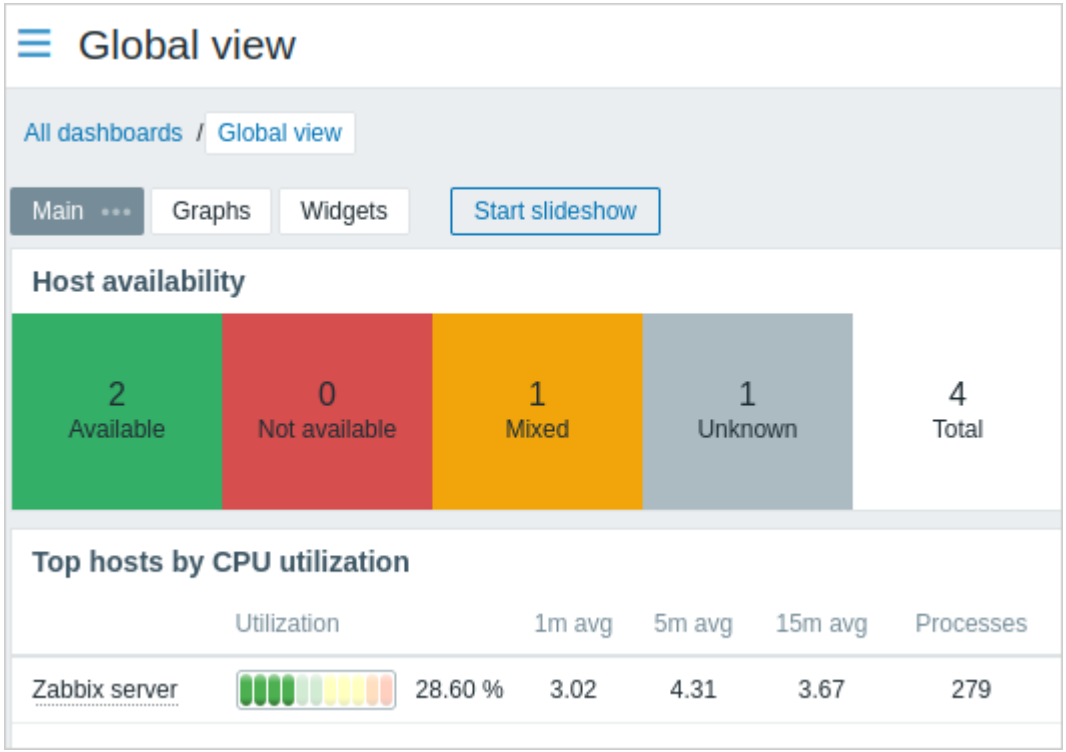
Zabbix ロゴの下にグローバル検索ボックスがあります

メニューは折りたたんだり、完全に非表示にしたりできます。

- 折りたたむには、Zabbix ロゴの横にある  をクリックします。折りたたまれたメニューでは、アイコンのみが表示されます。



- 非表示にするには、Zabbix ロゴの横にある  をクリックします。非表示メニューではすべてが非表示になります。



折りたたまれたメニュー

メニューがアイコンのみに折りたたまれている場合、マウスカーソルをメニュー上に置くと、完全なメニューが再表示されます。再表示されるメニューはページコンテンツの上に表示されます。ページコンテンツを右に移動するには、展開ボタンをクリックする必要があります。マウスカーソルが再び完全なメニューの外に置かれると、2 秒後にメニューは再び折りたたまれます。

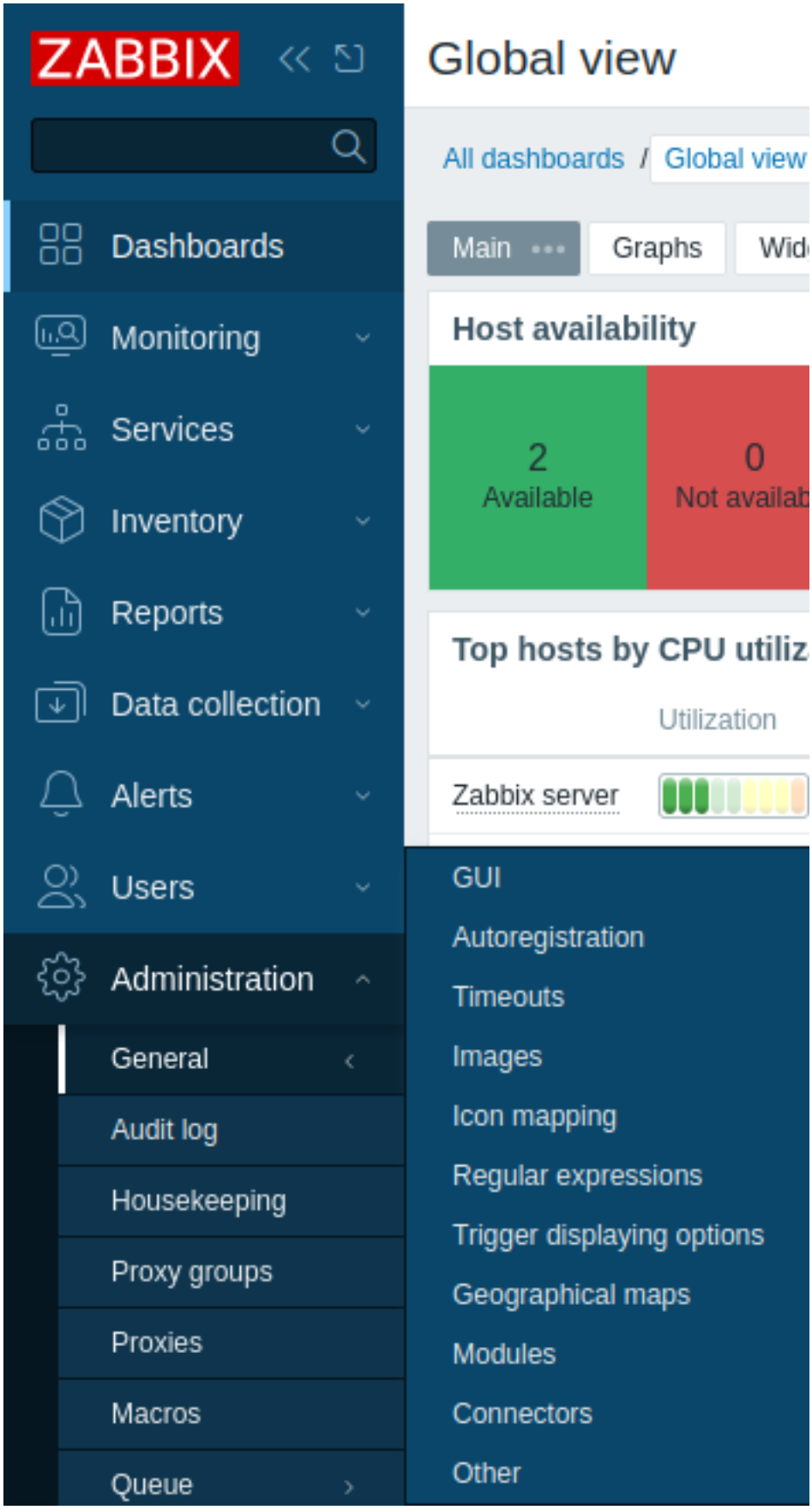
また、Tab キーを押すことで折りたたまれたメニューを完全に再表示することもできます。Tab キーを繰り返し押すことで、次のメニュー要素にフォーカスを移すことができます。

メニュー非表示

メニューが完全に非表示になっている場合でも、ハンバーガーアイコン を一回クリックするだけで完全なメニューを表示できます。ページコンテンツの上に再表示されることに注意してください。ページコンテンツを右に移動するには、サイドバーの表示ボタンをクリックしてメニューを再表示する必要があります。

メニューの階層

メニューには最大 3 つの階層があります。



コンテキストメニュー

メインメニューに加えて、Zabbix はホスト、アイテム、イベントのコンテキストメニューを提供し、最新の値、シンプルグラフ、設定フォーム、関連スクリプト、外部リンクなど、頻繁に使用されるエンティティにすばやくアクセスできます。

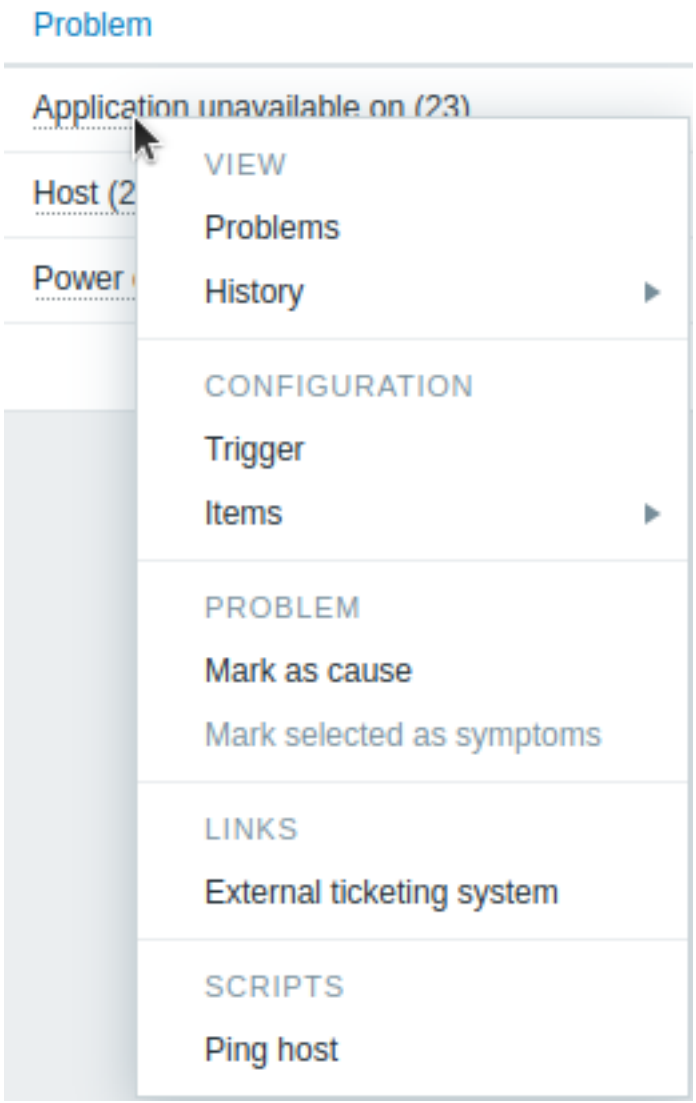
コンテキストメニューには、サポートされている場所でホスト、アイテム、または障害/トリガー名をクリックするとアクセスできます。

1 イベントメニュー

概要

イベントメニューには、イベントで頻繁に必要となるアクションやフロントエンドセクションへのショートカットが含まれています。

イベントメニューは、イベント名をクリックすると開きます。



コンテンツ

イベントコンテキストメニューには、表示、アクション、設定、障害、リンク、スクリプトの 6 つのセクションがあります。設定されていないエンティティの場合、リンクは無効になり、灰色で表示されます。スクリプトセクションとリンクセクションは、エンティティが設定されている場合に表示されます。

表示セクションには、次へのリンクが含まれています。

- 障害 - 基礎となるトリガーの未解決の障害のリストを開きます。
- ヒストリ - l 基礎となるアイテムの最新データグラフ/アイテムヒストリにつながります。トリガーが複数のアイテムを使用する場合、各アイテムのリンクが使用可能になります。

アクションセクションは、トリガー概要ウィジェットでのみ使用できます。以下へのリンクが含まれています。

- 障害更新 - 障害更新画面を開きます。

設定セクションには、次の設定へのリンクが含まれています。

- 障害を引き起こしたトリガー。
- トリガー式で使用されるアイテム。

Note:
設定セクションは、admin および Super admin ユーザーのみが使用できることに注意してください。

障害セクションには、次のオプションが含まれています。

- 副次的な障害から削除 - 障害を原因としてマークします。
- 選択したものを副次的な障害として追加 - 選択した障害をこの障害の症状としてマークします。

リンクセクションには、次のリンクが含まれています。

- 設定されたトリガー URLにアクセスします
- グローバルスクリプトで設定されたカスタムリンクにアクセスします (' 手動イベント処理' スコープに'URL' タイプで)
- 障害に対して設定された外部チケットにアクセスします (Webhookを設定するときにイベントメニュー項目を含めるオプションを参照してください)。

スクリプトセクションには、グローバルスクリプトを実行するためのリンクが含まれています (手動イベントアクションスコープで)。この機能は、外部システムで障害チケットを管理するために使用されるスクリプトを実行する場合に便利です。

サポートされている場所

イベントコンテキストメニューには、さまざまなフロントエンドセクションで障害またはイベント名をクリックするとアクセスできます。次に例を示します。

- ダッシュボードウィジェット、障害ウィジェット、トリガー概要ウィジェットなど。
- 監視 → 障害
- 監視 → 障害 → イベント詳細
- レポート → 障害発生数上位 100 項目 (グローバルスクリプトと外部チケットへのアクセスは、この場所ではサポートされていません)

2 ホストメニュー

概要

ホストメニューには、ホストに対して頻繁に必要とされるアクションやフロントエンドセクションへのショートカットが含まれています。

ホストメニューは、フロントエンドセクションによって、ホスト名または三点リーダーアイコンをクリックすることで開くことができます。例:

監視 → 最新データ　　データ収集 → ホスト

内容

ホストのコンテキストメニューには、表示、設定、リンク、スクリプトの 4 つのセクションがあります。設定されていないエンティティについては、リンクが無効化され、グレーで表示されます。スクリプトとリンクのセクションは、それぞれのエンティティが設定されている場合に表示されます。

表示セクションには以下のリンクがあります:

- ダッシュボード - ウィジェットとグラフを開きます。
- 障害 - 対象のトリガーの未解決の障害のリストを持つ障害セクションを開きます。
- 最新データ - 現在のホストのすべての最新データのリストを持つ最新データセクションを開きます。
- グラフ - 現在のホストのシンプルグラフを開きます。
- **Web** - 設定された Web シナリオへのリンクを開きます。
- インベントリ - 現在のホストのインベントリへのリンクを開きます。

設定セクションには以下のリンクがあります:

- ホストウィザード - 現在のホストのホストウィザードを開きます（検出されたホストでは無効)。
- ホスト - 現在のホストの設定フォーム。
- アイテム - 現在のホストのアイテムのリスト。
- トリガー - 現在のホストのトリガーのリスト。
- グラフ - 現在のホストのシンプルグラフ。
- ディスカバリ - 現在のホストのローレベルディスカバリルールのリスト。
- **Web** - 現在のホストの Web シナリオのリスト。

> **Note:**
> 設定セクションは管理者およびスーパ管理者ユーザーのみ利用可能です。

リンクセクションには以下のリンクがあります:

- 設定されたトリガー URLへのアクセス。
- グローバルスクリプトで設定されたカスタムリンクへのアクセス（スコープが手動ホストアクション、タイプが'URL' のもの）。

スクリプトセクションでは、現在のホストに設定されたグローバルスクリプトを実行できます。これらのスクリプトは、ホストメニューで利用できるようにスコープを手動ホストアクションとして定義する必要があります。

サポートされている場所

ホストメニューは、さまざまなフロントエンドセクションでホスト名をクリックすることでアクセスできます。例:

- ダッシュボードのウィジェット（問題、トップアイテム、トリガー概要など）
- 監視 → 問題
- 監視 → 問題 → イベントの詳細
- 監視 → ホスト
- 監視 → ホスト → Web 監視
- 監視 → 最新データ
- 監視 → マップ
- インベントリ → ホスト
- レポート → トップ 100 トリガー

ホストメニューは、データ収集 → ホストで三点リーダーアイコンをクリックすることでアクセスできます。

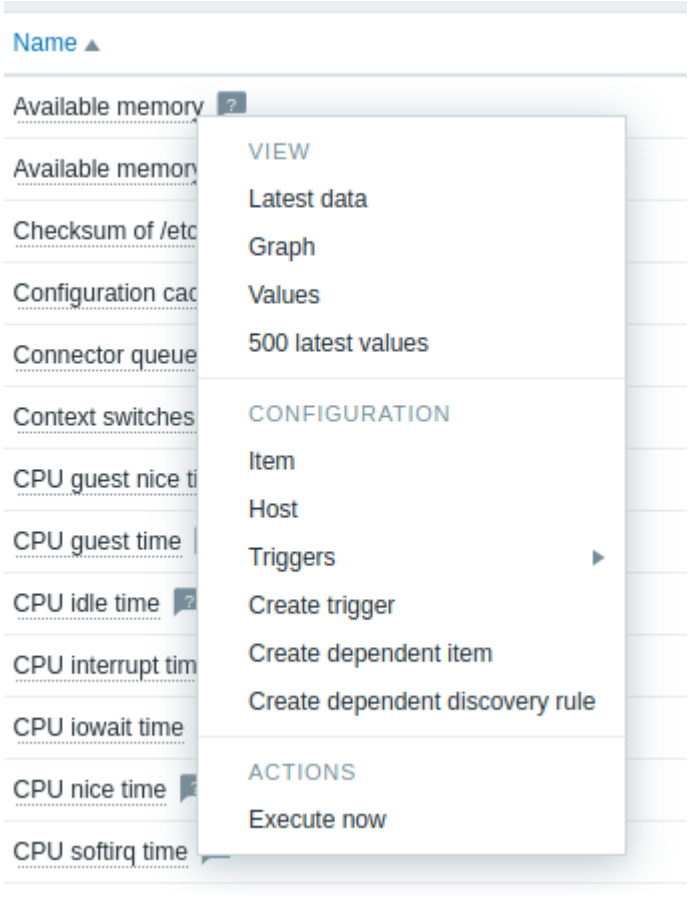
3 アイテムメニュー

概要

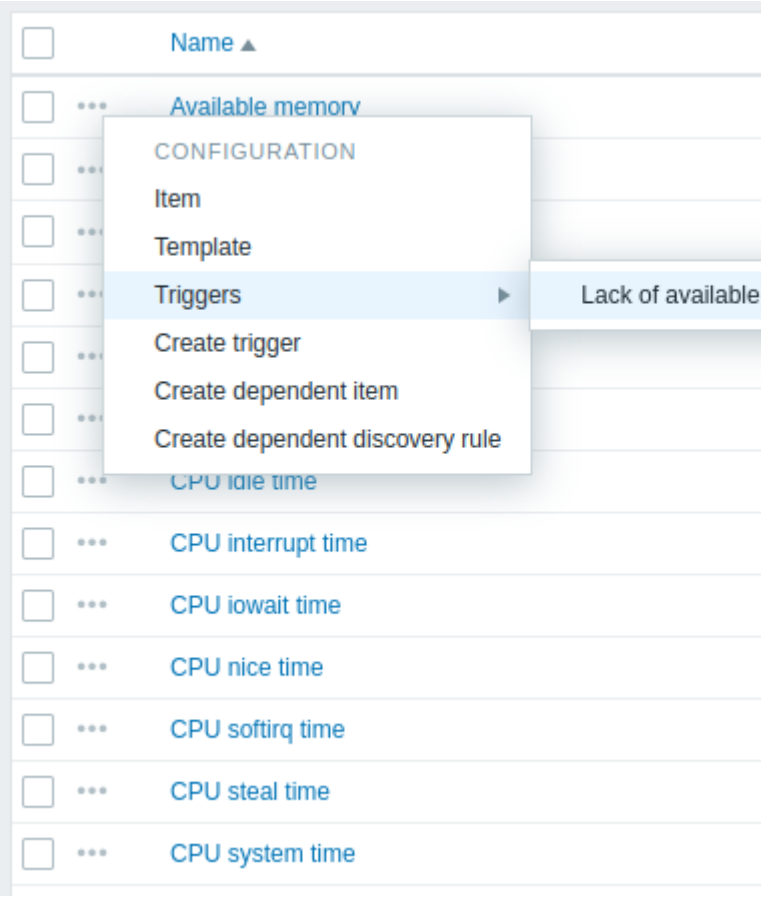
アイテムメニューには、アイテムで頻繁に必要となるアクションまたはフロントエンドセクションへのショートカットが含まれています。

アイテムメニューは、フロントエンドセクションに応じて、アイテム名または 3 つのドットのアイコンをクリックして開くことができます。例:

監視データ → 最新データ



データ収集 → テンプレート → アイテム



内容

アイテムメニューには、表示、設定、アクションの 3 つのセクションがあります。

表示セクションには、次のオプションがあります。

- 最新データ - 現在のホストとアイテムでフィルタリングされた最新データセクションを開きます。
- グラフ - 現在のアイテムのシンプルグラフを開きます。
- 値 - 過去 60 分間に現在のアイテムで受信したすべての値のリストを開きます。
- 最新 **500** 件の値 - 現在のアイテムの最新 500 件の値のリストを開きます。

設定セクション（*Admin* および *Super admin* タイプのユーザーのみ利用可能）には、次のオプションがあります。

- アイテム - 現在のアイテムのアイテム設定フォームを開きます。
- テンプレート - アイテムが属するテンプレートのテンプレート設定フォームを開きます（データ収集 → テンプレート → アイテムからアイテムメニューにアクセスした場合のみ利用可能）。
- ホスト - アイテムが属するホストのホスト設定フォームを開きます。
- トリガー - マウスオーバーでアイテムのトリガーのリストを開きます（存在する場合）。トリガーをクリックすると、そのトリガー設定フォームを開きます。
- トリガーの作成 - このアイテムのトリガーを作成するためのトリガー設定フォームを開きます。
- 従属アイテムの作成 - 現在のアイテムをマスターアイテムとして従属アイテムを作成するためのアイテム設定フォームを開きます。
- 従属ディスカバリールールの作成 - 現在のアイテムをマスターアイテムとして従属ディスカバリールールを作成するためのディスカバリールール設定フォームを開きます。

アクションセクションには、次のオプションがあります。

- 今すぐ実行 - 新しいアイテム値のチェックを即時実行します。

サポートされている場所

アイテムメニューは、以下のようなフロントエンドのさまざまなセクションでアイテム名をクリックすることでアクセスできます:

- 監視 → 最新データ
- データ収集 → ホスト → アイテム
- データ収集 → ホスト → ディスカバリルール → アイテムプロトタイプ

アイテムメニューは、トップアイテムダッシュボードウィジェットでアイテム値をクリックすることでアクセスできます。

2 Web インターフェースセクション

メニュー構成　Zabbix Web インターフェースのメニューは次の構造になっています。

- ダッシュボード
- 監視
  - 障害
  - ホスト
  - 最新データ
  - マップ
  - ディスカバリ
- サービス
  - サービス
  - SLA
  - SLA レポート
- インベントリ
  - 概要
  - ホスト
- レポート
  - システム情報
  - 定期レポート
  - 稼働レポート
  - 障害発生数上位 100 項目
  - 監査ログ
  - アクションログ
  - 通知レポート
- データ収集
  - テンプレートグループ
  - ホストグループ
  - テンプレート

    - ホスト
    - メンテナンス
    - イベント相関関係
    - ディスカバリ
- 通知
    - アクション
        * トリガーアクション
        * サービスアクション
        * ディスカバリアクション
        * 自動登録アクション
        * 内部イベントアクション
    - メディアタイプ
    - スクリプト
- ユーザー
    - ユーザーグループ
    - ユーザーの役割
    - ユーザー
    - API トークン
    - 認証
- 管理
    - 一般
        * GUI
        * 自動登録
        * 画像
        * アイコンのマッピング
        * 正規表現
        * トリガー表示オプション
        * 地理マップ
        * モジュール
        * その他
    - 監査ログ
    - ハウスキーピング
    - プロキシグループ
    - プロキシ
    - マクロ
    - キュー
        * キュー概要
        * プロキシごとのキュー概要
        * キュー詳細

1 ダッシュボード

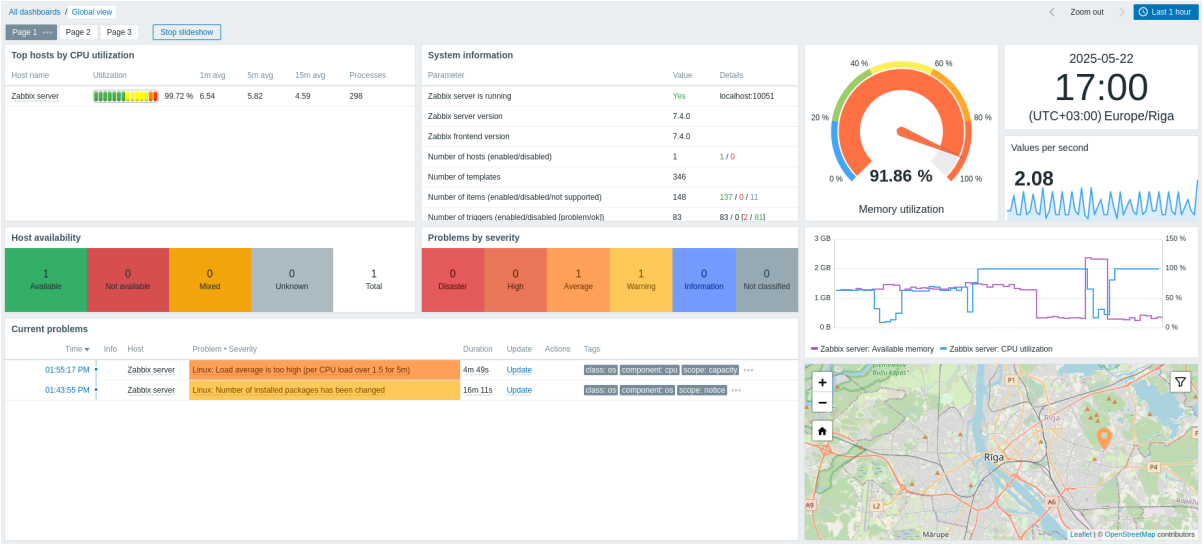
概要

ダッシュボードセクションは、ダッシュボード内のすべての重要な情報の概要を表示するために設計されています。

一度に表示できるダッシュボードは 1 つだけですが、複数のダッシュボードを設定することが可能です。各ダッシュボードには 1 つまたは複数のページを含めることができ、スライドショーで回転させることができます。

ダッシュボードページはウィジェットで構成されており、各ウィジェットは特定の種類やソースの情報（サマリー、マップ、グラフ、時計など）を表示するように設計されています。

ウィジェット内のホストへのアクセスは、ホストの権限に依存します。



ページやウィジェットは、ダッシュボードの編集モードでダッシュボードに追加・編集します。ページはダッシュボードの表示モードで閲覧・回転できます。

グラフウィジェットに表示される期間は、ウィジェット上部にある期間セレクターで制御されます。そのラベルには現在選択されている期間が表示され、ラベルをクリックすると期間セレクターを展開・折りたたみできます。

ダッシュボードがキオスクモードで表示され、ウィジェットのみが表示されている場合、グラフ内をダブルクリックすることでグラフの期間をズームアウトできることに注意してください。

ダッシュボードのサイズ

ダッシュボードの最小幅は 1200 ピクセルです。ダッシュボードはこの幅より小さくはなりません。ブラウザウィンドウがこれより小さい場合は、水平スクロールバーが表示されます。

ダッシュボードの最大幅はブラウザウィンドウの幅です。ダッシュボードウィジェットはウィンドウに合わせて水平方向に伸縮します。同時に、ダッシュボードウィジェットはウィンドウの制限を超えて水平方向に伸ばすことはできません。

水平方向では、ダッシュボードは常に等しい幅の 72 列で構成されており、動的に伸縮します（ただし合計 1200 ピクセル未満にはなりません）。

垂直方向では、ダッシュボードには最大 64 行まで含めることができ、各行の高さは 70 ピクセルで固定されています。

したがって、ウィジェットは最大で 72 列の幅と 64 行の高さにすることができます。

ダッシュボードの表示

設定済みのすべてのダッシュボードを表示するには、セクションタイトルのすぐ下にあるすべてのダッシュボードをクリックします。



ダッシュボードは共有タグ付きで表示されます。

- 自分 - 現在のユーザーが所有するダッシュボードを示します

- 共有 - パブリックダッシュボードまたは他のユーザーやユーザーグループと共有されたプライベートダッシュボードを示します

共有タグは、現在のユーザーが所有するダッシュボードにのみ表示されることに注意してください。他のユーザーが所有するダッシュボードは、共有タグなしで一覧表示されます。

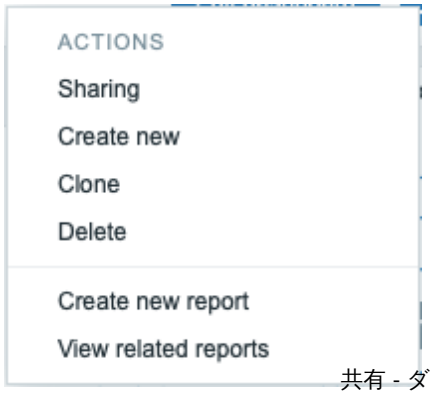
リストの右上にあるフィルターで、ダッシュボード名や現在のユーザーが作成したダッシュボードで絞り込むことができます。

1 つまたは複数のダッシュボードを削除するには、該当するダッシュボードのチェックボックスをオンにし、リスト下部の削除をクリックします。

ダッシュボードの表示

単一のダッシュボードを表示するには、ダッシュボードの一覧でその名前をクリックします。

ダッシュボードを表示しているときは、次のオプションが利用できます。

Edit dashboard	ダッシュボードの編集モードに切り替えます。 編集モードは、新しいダッシュボードを作成する場合や、ウィジェットの 編集ボタンをクリックした場合にも開きます。
	アクションメニューを開きます（下記のアクション説明を参照）。
	共有 - ダッシュボードの共有設定を編集します。 新規作成 - 新しいダッシュボードを作成します。 クローン - 既存のダッシュボードのプロパティをコピーして新しいダッシュボードを作成します。最初にダッシュボードのパラメータ入力が求められます。その後、元のダッシュボードのすべてのウィジェットを持つ新しいダッシュボードが編集モードで開きます。 削除 - ダッシュボードを削除します。 新しいレポートの作成 - レポートの設定フォームを持つポップアップウィンドウを開きます。ユーザーにスケジュールレポートの管理権限がない場合は無効になります。 関連レポートの表示 - 現在のダッシュボードを元にした既存のレポート一覧を持つポップアップウィンドウを開きます。関連レポートがない場合や、ユーザーにスケジュールレポートの閲覧権限がない場合は無効になります。
	ページの内容のみを表示します（キオスクモード）。 キオスクモードは、次の URL パラメータでもアクセスできます： `/zabbix.php?action=dashboard.view&kiosk=1`。 通常モードに戻るには：`/zabbix.php?action=dashboard.view&kiosk=0`

共有

ダッシュボードは公開または非公開にすることができます。

公開ダッシュボードはすべてのユーザーに表示されます。非公開ダッシュボードは、その所有者と共有設定で追加されたユーザー/ユーザーグループにのみ表示されます。

ダッシュボードの共有ステータスを編集するには、単一のダッシュボードを表示しているときにアクションメニューの共有オプションをクリックします。

パラメータ	説明
タイプ	ダッシュボードのタイプを選択: 非公開 - ダッシュボードは選択したユーザーグループとユーザーにのみ表示されます 公開 - ダッシュボードはすべてのユーザーに表示されます

パラメータ	説明
ユーザーグループ共有のリスト	ダッシュボードにアクセスできるユーザーグループを選択します。 読み取り専用または読み書きアクセスを許可できます。
ユーザー共有のリスト	ダッシュボードにアクセスできるユーザーを選択します。 読み取り専用または読み書きアクセスを許可できます。

共有オプションがユーザーに利用可能なアクションをどのように制限するかについては、ダッシュボードへの権限を参照してください。

ダッシュボードの編集

ダッシュボードの編集時には、以下のオプションが利用できます。

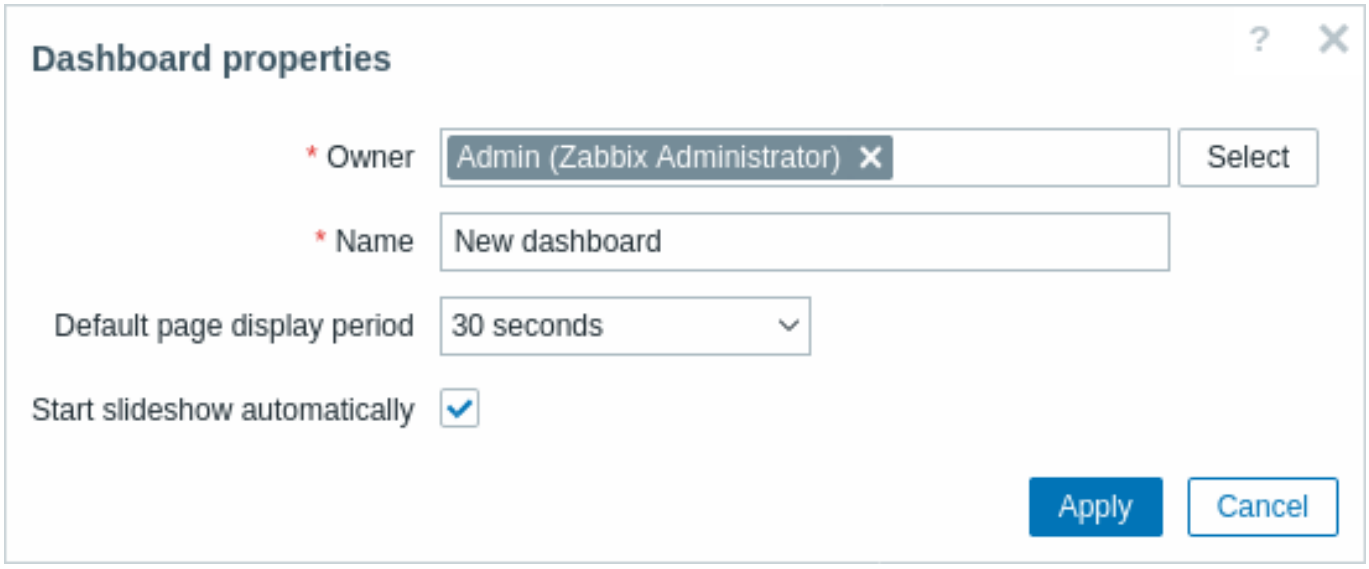
	ダッシュボードのパラメータを編集します。 新しいウィジェットを追加します。 矢印ボタンをクリックするとアクションメニューが開きます（アクションの説明は下記参照）。 ウィジェットを追加 - 新しいウィジェットを追加します ページを追加 - 新しいページを追加します ウィジェットを貼り付け - コピーしたウィジェットを貼り付けます。このオプションは、ウィジェットがコピーされていない場合はグレー表示されます。一度にコピーできるのは 1 つのエンティティ（ウィジェットまたはページ）のみです。 ページを貼り付け - コピーしたページを貼り付けます。このオプションは、ページがコピーされていない場合はグレー表示されます。 ダッシュボードの変更を保存します。 ダッシュボードの変更をキャンセルします。

ダッシュボードの作成

新しいダッシュボードを作成するには、2 つの方法があります。

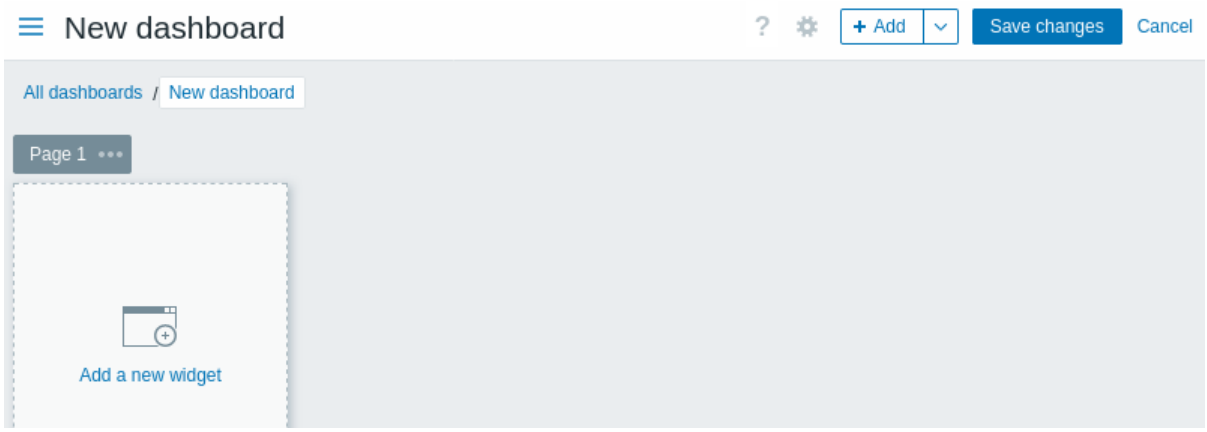
- すべてのダッシュボードを表示しているときにダッシュボードの作成をクリックします
- 単一のダッシュボードを表示しているときに、アクションメニューから新規作成を選択します

最初に、一般的なダッシュボードパラメータの入力を求められます。



パラメータ	説明
所有者	ダッシュボードの所有者となるシステムユーザーを選択します。
名前	ダッシュボード名を入力します。
デフォルトページ表示期間	スライドショーで次のページに切り替わるまで、ダッシュボードページが表示される期間を選択します。
スライドショーを自動的に開始	複数のダッシュボードページが存在する場合に、スライドショーを自動的に実行するには、このチェックボックスをオンにします。

適用をクリックすると、空のダッシュボードが開きます。



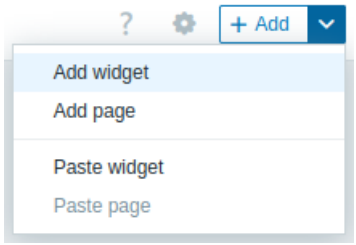
ダッシュボードにウィジェットやページを追加して内容を充実させることができます。

変更を保存ボタンをクリックすると、ダッシュボードが保存されます。キャンセルをクリックすると、ダッシュボードは作成されません。

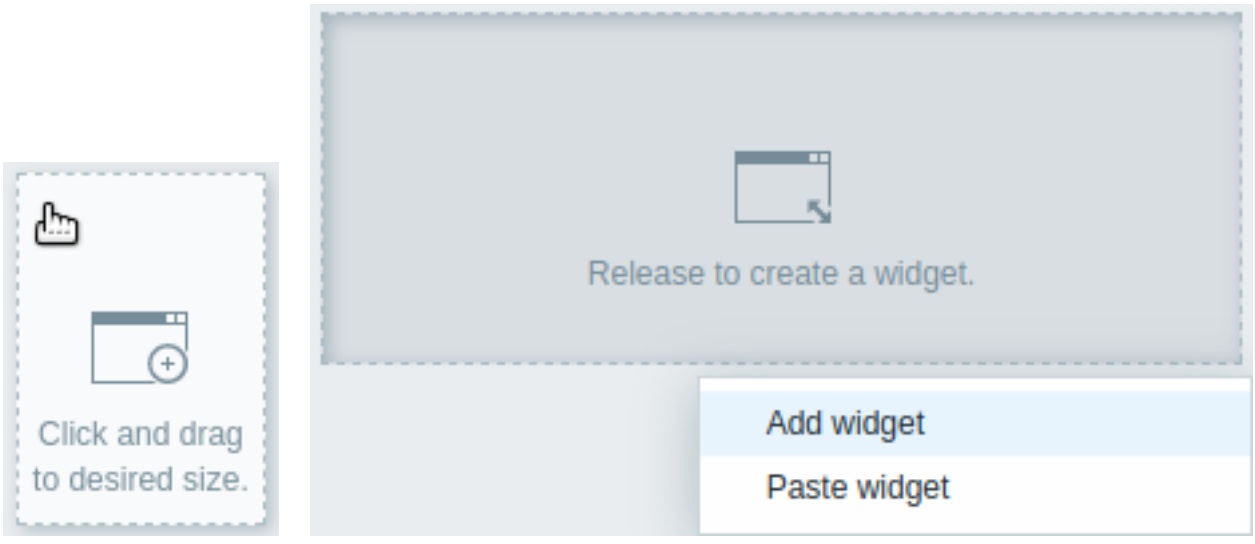
ウィジェットの追加

ダッシュボードには、さまざまなウィジェット（アクションログ、時計、ディスカバリーステータスなど）を追加できます。ウィジェットは 2 つの方法で追加できます:

- 追加ボタンまたはその横の矢印をクリックし、アクションメニューからウィジェットの追加を選択します。ウィジェットを設定すると、デフォルトサイズで追加され、既存のウィジェットの後に配置されます。

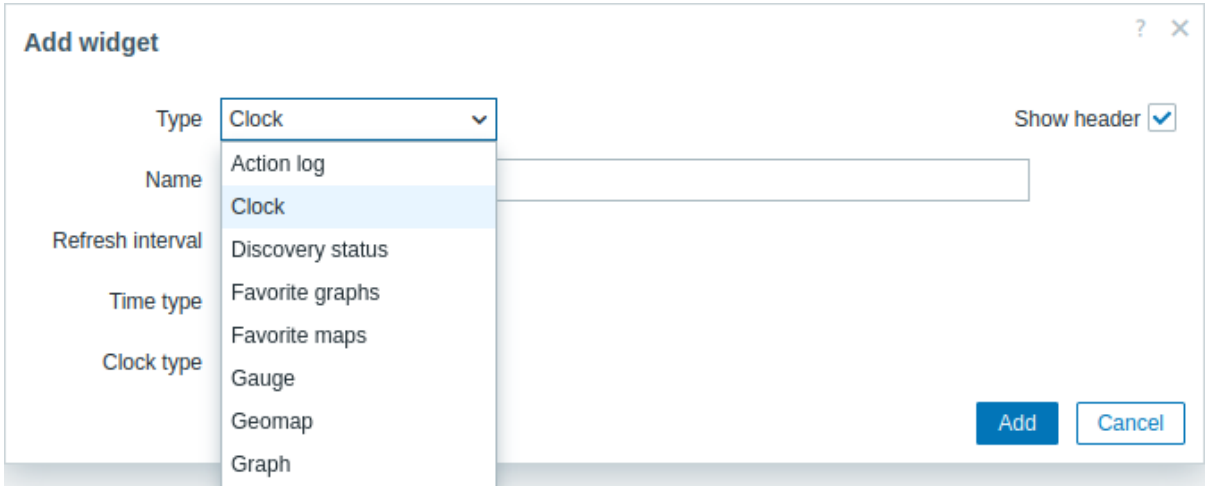


- ダッシュボードの空いている場所にマウスを移動します。プレースホルダーが表示されたら、それをクリックしてウィジェット設定フォームを開きます。ウィジェットを設定すると、デフォルトサイズで追加されるか、利用可能なスペースに合わせてリサイズされます。または、プレースホルダーをドラッグしてウィジェットのサイズを決定し、リリースしてウィジェット設定フォームを開きます（ウィジェットがクリップボードにコピーされている場合は、最初にウィジェットの追加またはウィジェットの貼り付けの選択を求められます）。



ウィジェット設定フォームでは:

1. ウィジェットタイプを選択します。
2. ウィジェットパラメータを設定します。
3. 追加をクリックしてウィジェットをダッシュボードに追加します。



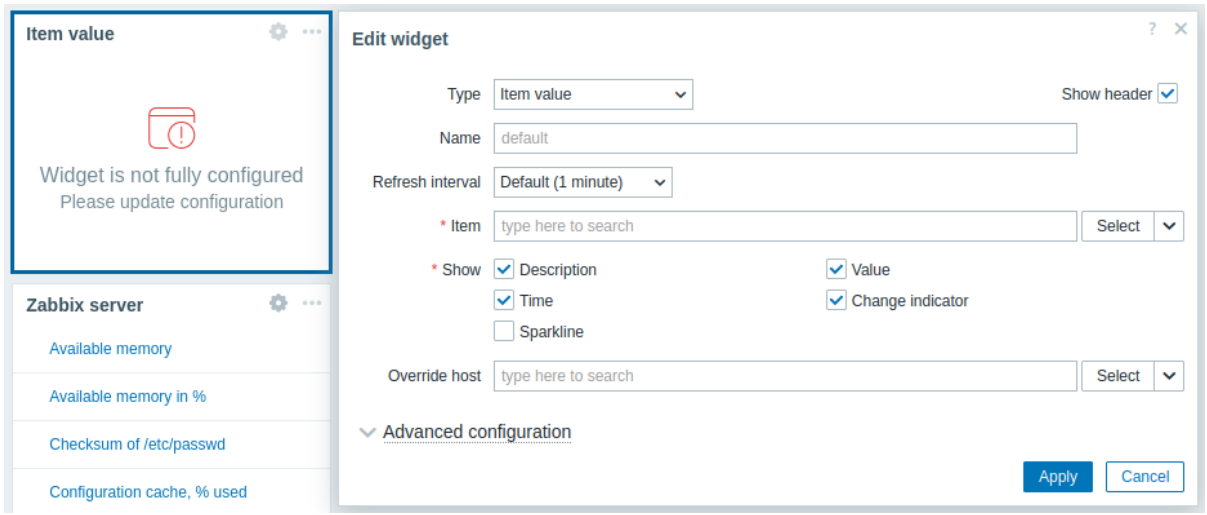
ウィジェットの編集

ウィジェットは、ダッシュボードの編集モードでタイトルバーをドラッグすることで、サイズ変更や再配置ができます。また、ウィジェット設定フォームもタイトルバーをドラッグして再配置できます。

各ウィジェットの右上隅には以下のコントロールがあります:

- - ウィジェットの編集
- - ウィジェットメニューを開く

編集中はウィジェットがハイライトされ、変更がリアルタイムでプレビューされます。必須のウィジェットパラメータが設定されていない場合、ウィジェットはウィジェットが完全に設定されていませんの状態になります。



編集後は、ウィジェット内の適用をクリックし、ダッシュボード右上の変更を保存をクリックして変更を適用します。

ウィジェットのコピー/貼り付け

ダッシュボードウィジェットをコピーして貼り付けることで、同じ設定の新しいウィジェットをすばやく作成できます。ウィジェットは、同じダッシュボード内または異なるタブで開いたダッシュボード間でコピー & ペーストできます。

ウィジェットをコピーするには、ウィジェットメニューを使用します。コピーしたウィジェットを貼り付けるには (ダッシュボードの編集モードで利用可能):

- 追加ボタンの横の矢印をクリックし、ウィジェットを貼り付けを選択します。
- または、ダッシュボードの空いている領域をクリックし、ウィジェットを貼り付けを選択します。

また、コピーしたウィジェットを既存のウィジェットの上に貼り付けることもできます。その場合は、ウィジェットメニューの貼り付けオプションを使用します。

スライドショーの作成

ダッシュボードに 2 ページ以上が含まれている場合（ページの追加を参照）、以下のいずれかの条件を満たすとスライドショーが自動的に実行されます。

- ダッシュボードのプロパティでスライドショーを自動的に開始オプションがマークされている
- ダッシュボードの URL に `slideshow=1` パラメータが含まれている

ページは、ダッシュボードおよび個々のページのプロパティで指定された間隔に従って切り替わります。以下をクリックします。

- スライドショーを停止 - スライドショーを停止します
- スライドショーを開始 - スライドショーを開始します



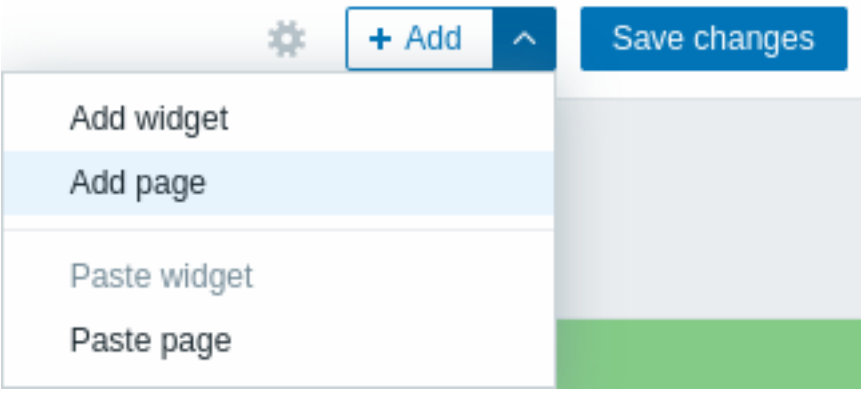
スライドショー関連のコントロールは、キオスクモード（ページの内容のみが表示されるモード）でも利用できます。

- - スライドショーを停止
- - スライドショーを開始
- - 1 ページ戻る
- - 次のページへ進む

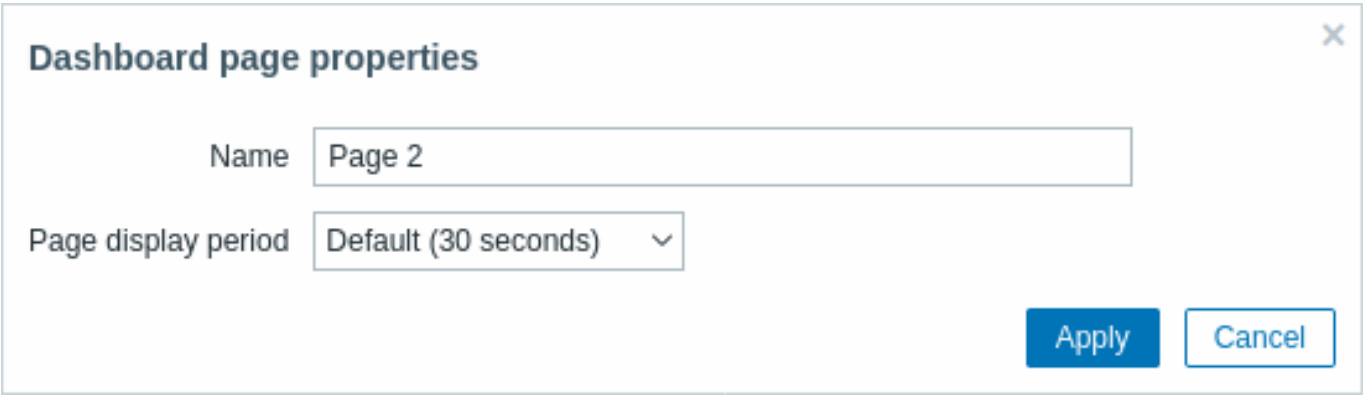
ページの追加

ダッシュボードに新しいページを追加するには:

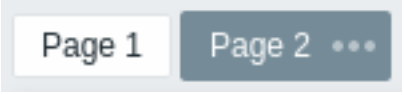
- ダッシュボードが編集モードであることを確認します
- 追加ボタンの横にある矢印をクリックし、ページを追加オプションを選択します



- 一般的なページパラメータを入力し、適用をクリックします。名前を空欄のままにすると、ページはページ N という名前で追加されます。ここで'N' はページの連番です。ページの表示期間は、スライドショーでページが表示される時間をカスタマイズできます。



新しいページが追加され、新しいタブ (ページ *2*) で表示されます。



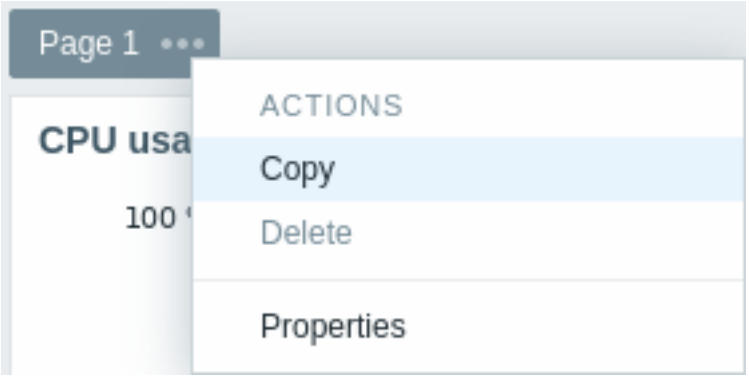
ページは、ページタブをドラッグアンドドロップすることで並べ替えることができます。並べ替えても元のページ名は維持されます。各ページのタブをクリックすることで、いつでもそのページに移動できます。

新しいページを追加すると、そのページは空です。上記の説明のようにウィジェットを追加できます。

ページのコピー/貼り付け

ダッシュボードのページはコピーして貼り付けることができ、既存のページのプロパティを持つ新しいページを作成できます。同じダッシュボードまたは別のダッシュボードから貼り付けることができます。

既存のページをダッシュボードに貼り付けるには、まずページメニューを使用してコピーします:

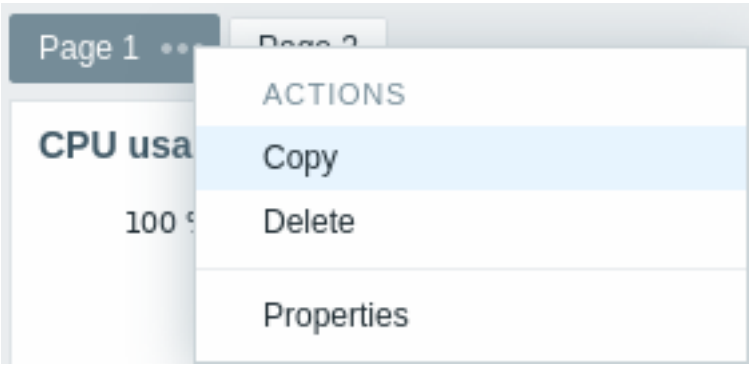


コピーしたページを貼り付けるには:

- ダッシュボードが編集モードであることを確認します
- 追加ボタンの横の矢印をクリックし、ページを貼り付けオプションを選択します

ページメニュー

ページメニューは、ページ名の横にある三点リーダーをクリックすると開きます。

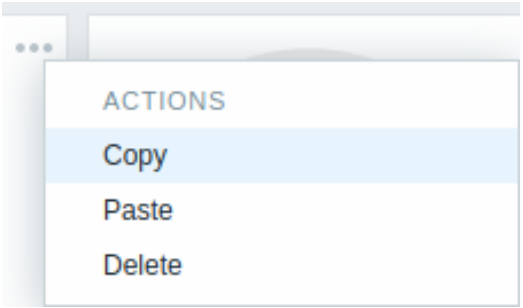


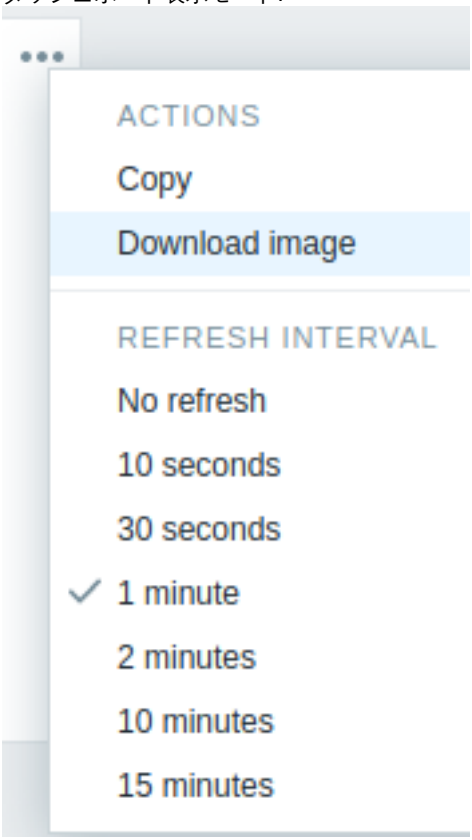
以下のオプションが含まれています。

- コピー - ページをコピーします
- 削除 - ページを削除します（ページはダッシュボード編集モードでのみ削除できます）
- プロパティ - ページのパラメータ（名前やスライドショーでのページ表示期間）をカスタマイズします

ウィジェットメニュー

ウィジェットメニューには、ダッシュボードが編集モードか表示モードかによって異なるオプションが表示されます。

ウィジェットメニュー	オプション
ダッシュボード編集モード: 	コピー - ウィジェットをコピー 貼り付け - コピーしたウィジェットをこのウィジェットに貼り付け ウィジェットがコピーされていない場合、このオプションはグレー表示されます。 削除 - ウィジェットを削除

ウィジェットメニュー	オプション
ダッシュボード表示モード: 	コピー - ウィジェットをコピー 画像のダウンロード - ウィジェットを PNG 画像としてダウンロード (グラフ/クラシックグラフウィジェットのみ利用可能) 更新間隔 - ウィジェット内容の更新頻度を選択

ダッシュボードへの権限

通常のユーザーおよび「Admin」タイプのユーザー（ダッシュボードへのアクセスとダッシュボードの作成と編集オプションが、割り当てられたユーザーロールで有効になっている場合）のダッシュボードへの権限は、以下のように制限されています。

- ダッシュボードの共有設定でユーザーやユーザーグループが追加されていなくても、公開ダッシュボードを表示およびクローンできます。
- 共有設定で読み取り権限以上が設定されていれば、プライベートダッシュボードを表示およびクローンできます。
- 共有設定で読み書き権限が設定されている場合のみ、ダッシュボードを編集および削除できます。
- ダッシュボードの所有者を変更することはできません。

1 ダッシュボードウィジェット

概要

Zabbix は、監視データを可視化し要約するさまざまなダッシュボードウィジェットを提供しており、システムのパフォーマンスと可用性を明確に把握できます。

以下は、主な目的ごとにグループ化したすべてのウィジェットの概要です。

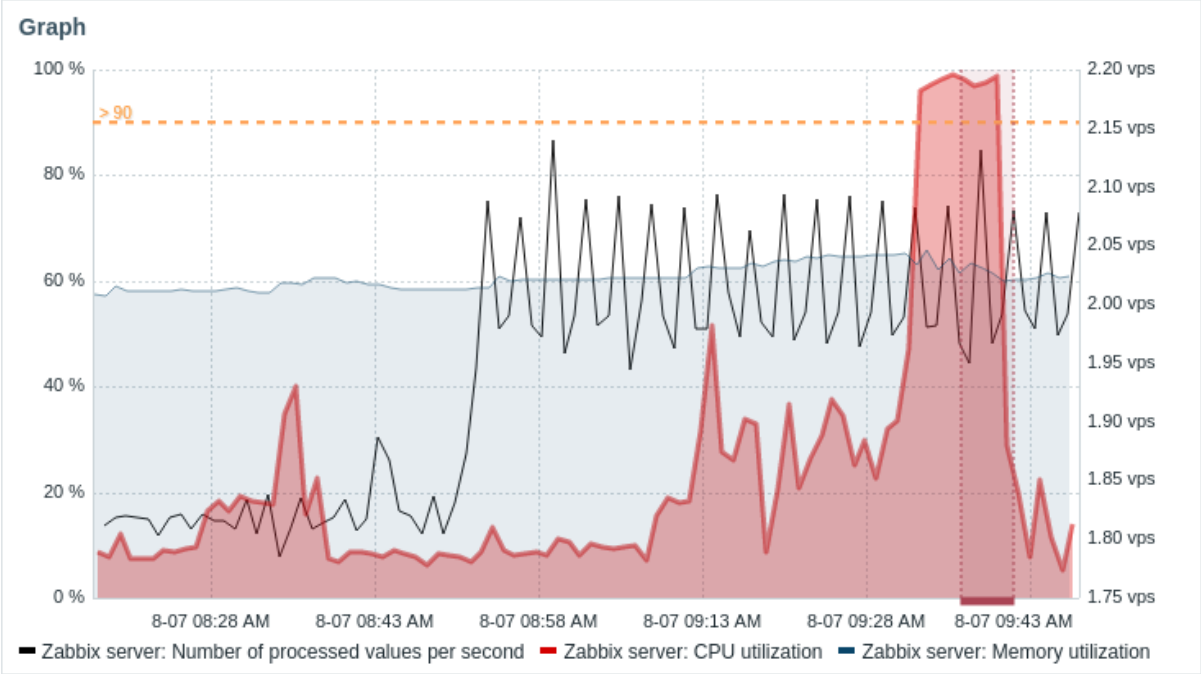
ウィジェットの設定の詳細については、ダッシュボードウィジェットのパラメータを参照してください。

データの可視化

これらのウィジェットは、さまざまな視覚的な形式でデータを表示し、さまざまな方法でデータや問題を監視するのに役立ちます。

グラフ

グラフウィジェットは、数値アイテムデータをベクターグラフとして表示します。これにより、メトリクスの追跡、問題の特定、時間やホスト間での値の比較が容易になります。



関連情報:

- グラフ（クラシック） - 数値アイテムデータを画像ベースのグラフとして表示する従来のグラフウィジェットです。
- グラフプロトタイプ - LLD ルールのグラフプロトタイプやアイテムプロトタイプに基づいて自動作成されたカスタムグラフのグリッドを表示します。
- お気に入りグラフ - 現在のユーザーがお気に入りとしてマークしたグラフへのショートカットのアルファベット順リストを表示します。

散布図

散布図ウィジェットは、X 軸と Y 軸に沿って個々のデータポイントをプロットすることで、2 つのメトリクス間の関係を表示します。これにより、データセット内のパターン、クラスター、相関関係、外れ値を明らかにするのに役立ちます。

円グラフ

円グラフウィジェットは、数値アイテムデータをベクターベースの円グラフまたはドーナツグラフとして表示します。
 このウィジェットは、アイテムやホストが全体のデータセットにどのように貢献しているかを視覚化するのに役立ちます。

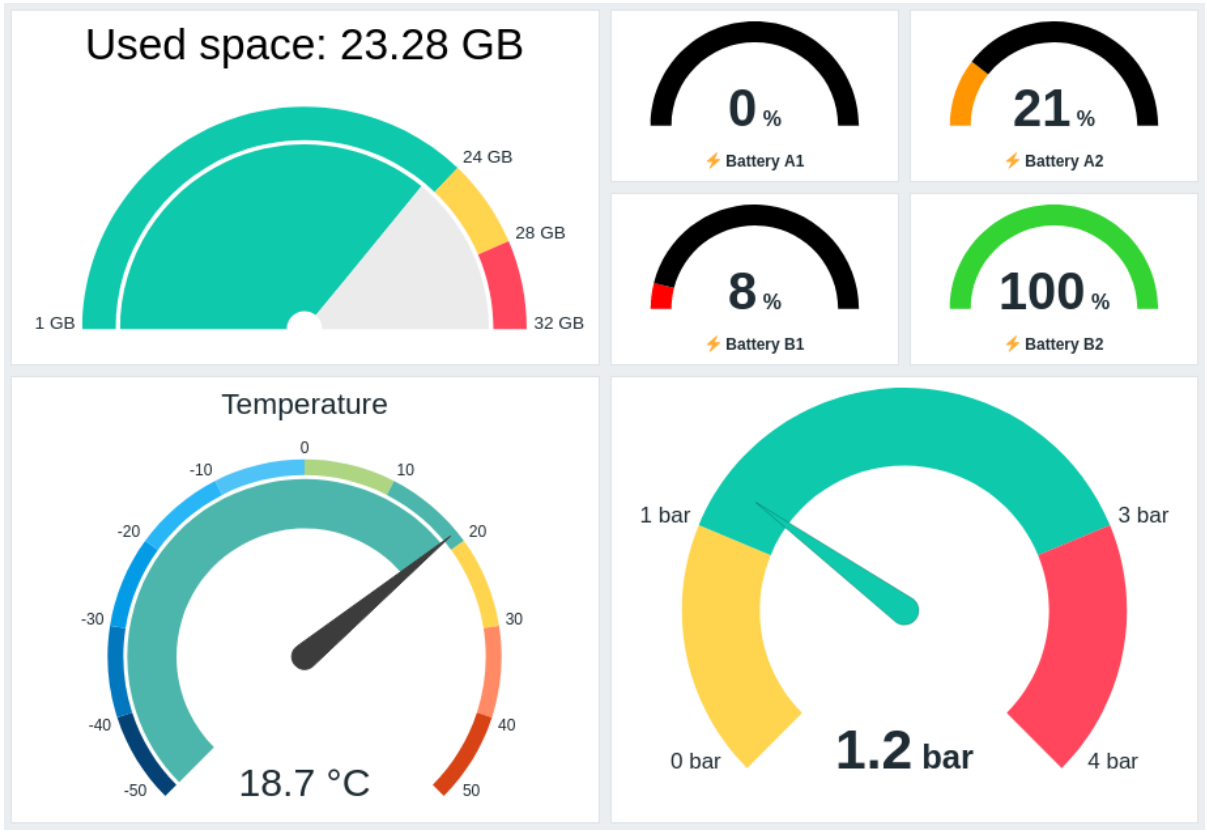
—

—

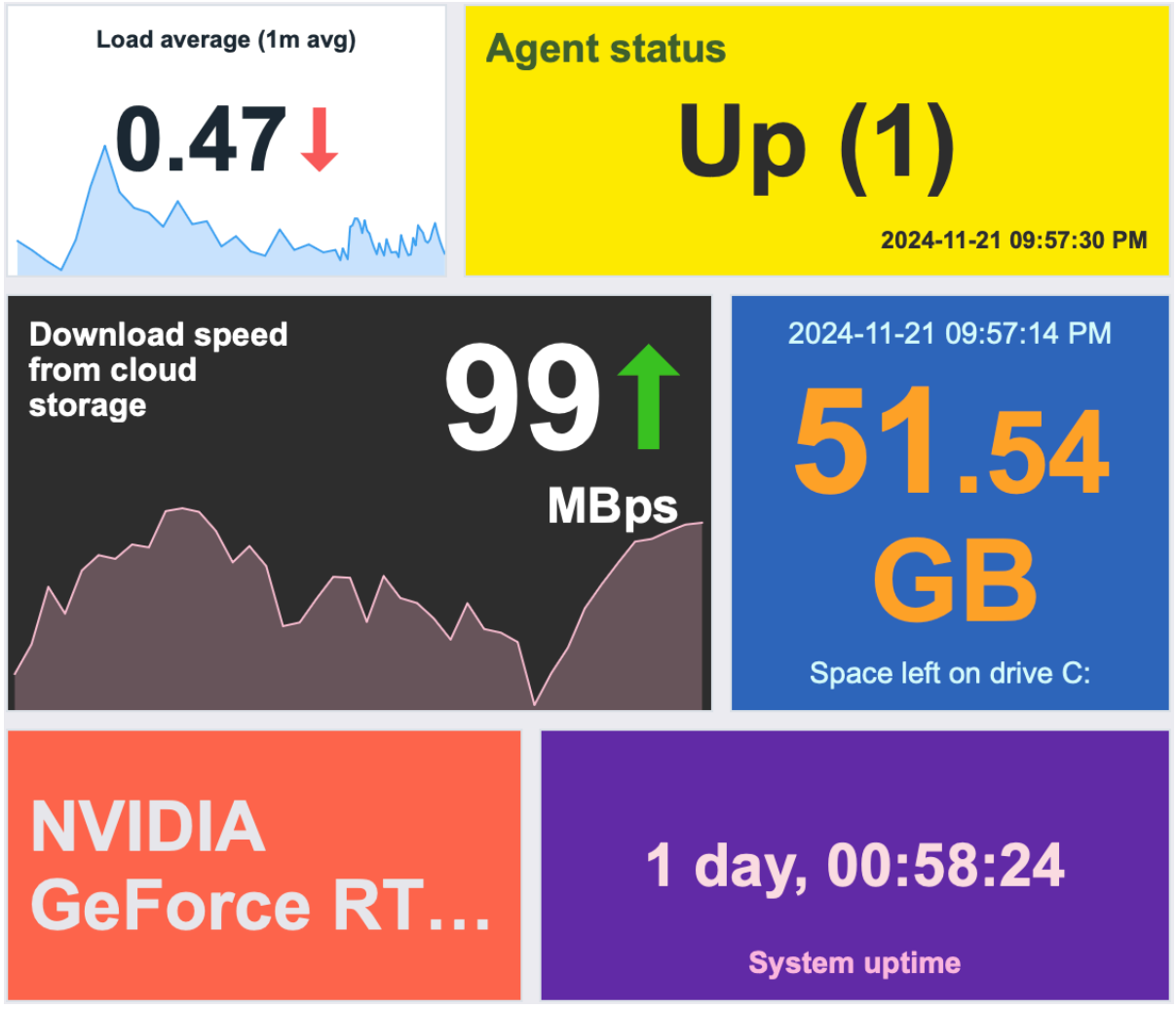
ゲージ/アイテム値

ゲージ/アイテム値ウィジェットは、単一のアイテムの値を表示します。これは、重要なメトリクスを監視したり、しきい値を可視化したり、データの急激な変化を検出したりするのに便利です。

ゲージウィジェット:



アイテム値ウィジェット:

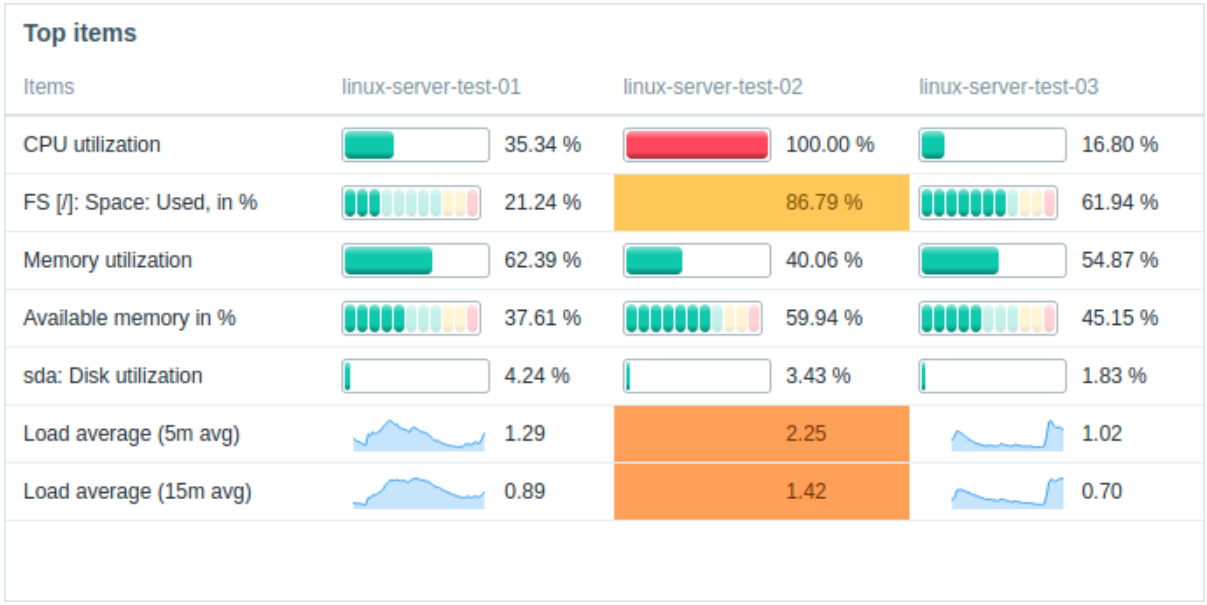


アイテム履歴

アイテム履歴ウィジェットは、すべてのアイテムタイプ（数値、テキストなど）の最新データを表形式で表示します。プログレスバーや画像（ブラウザーアイテムに便利）を表示したり、値をハイライト表示したり（ログファイル監視に便利）することができます。

トップアイテム

トップアイテムウィジェットは、選択したアイテムのトップ/ボトム値を表示し、そのパフォーマンスの概要を迅速に把握できます。

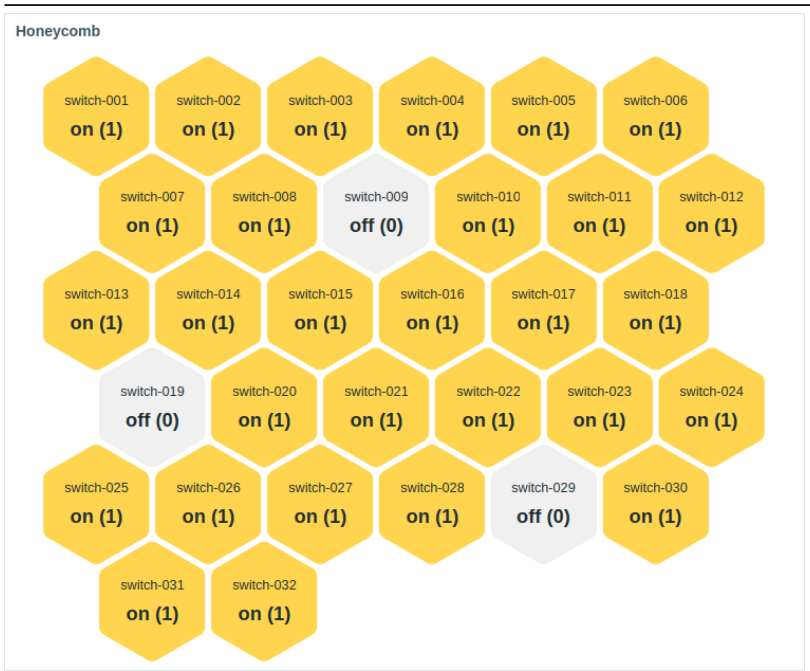


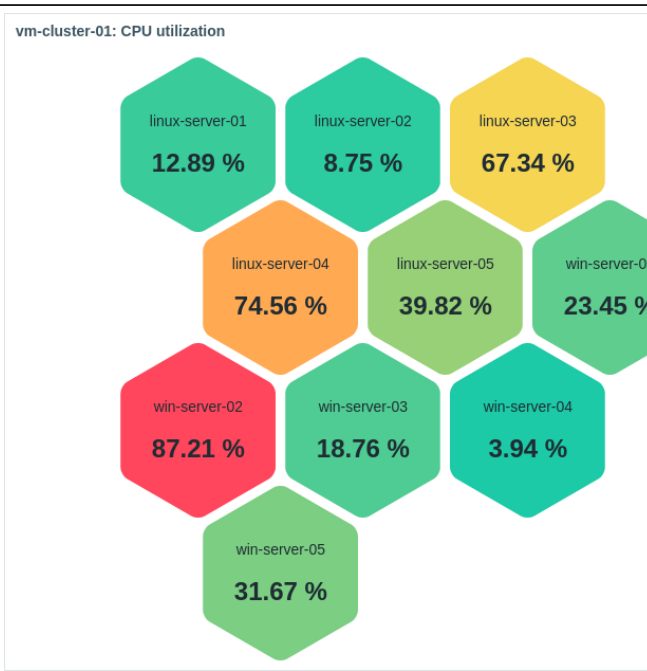
関連情報:

- トップホスト - 複数ホストの選択したアイテムのトップ/ボトム値を表示します。
- トップトリガー - 問題数が最も多いトリガーを表示します。

ハニカム

ハニカムウィジェットは、ホストまたはアイテムを六角形のセルとしてグリッドに表示します。このレイアウトにより、問題のあるホスト/アイテムの検出、問題のクラスターの特定、グループの比較などが容易になります。



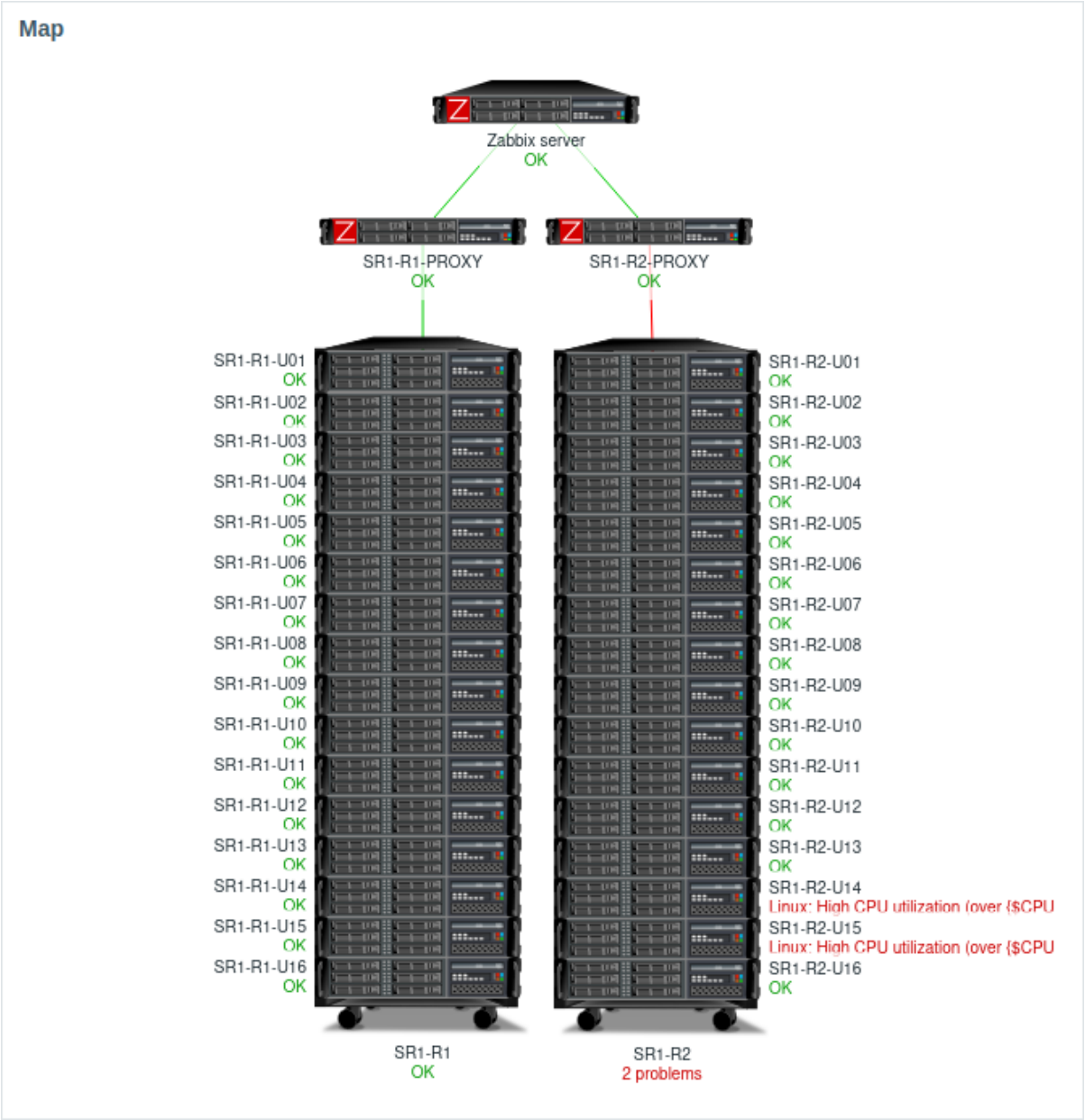


マップ

マップウィジェットは、ネットワークトポロジーやホストの位置を可視化し、全体および場所ごとの健全性を評価するのに役立ちます。

マップ

マップウィジェットは、ネットワークマップや類似のビジュアライゼーションを表示し、ネットワークの動的な概要を提供します。



関連情報:

- マップナビゲーションツリー - 既存のマップの階層と、各マップおよびマップグループの問題数を表示します。
- お気に入りマップ - 現在のユーザーがお気に入りとしてマークしたマップへのショートカットのリストをアルファベット順に表示します。

ジオマップ

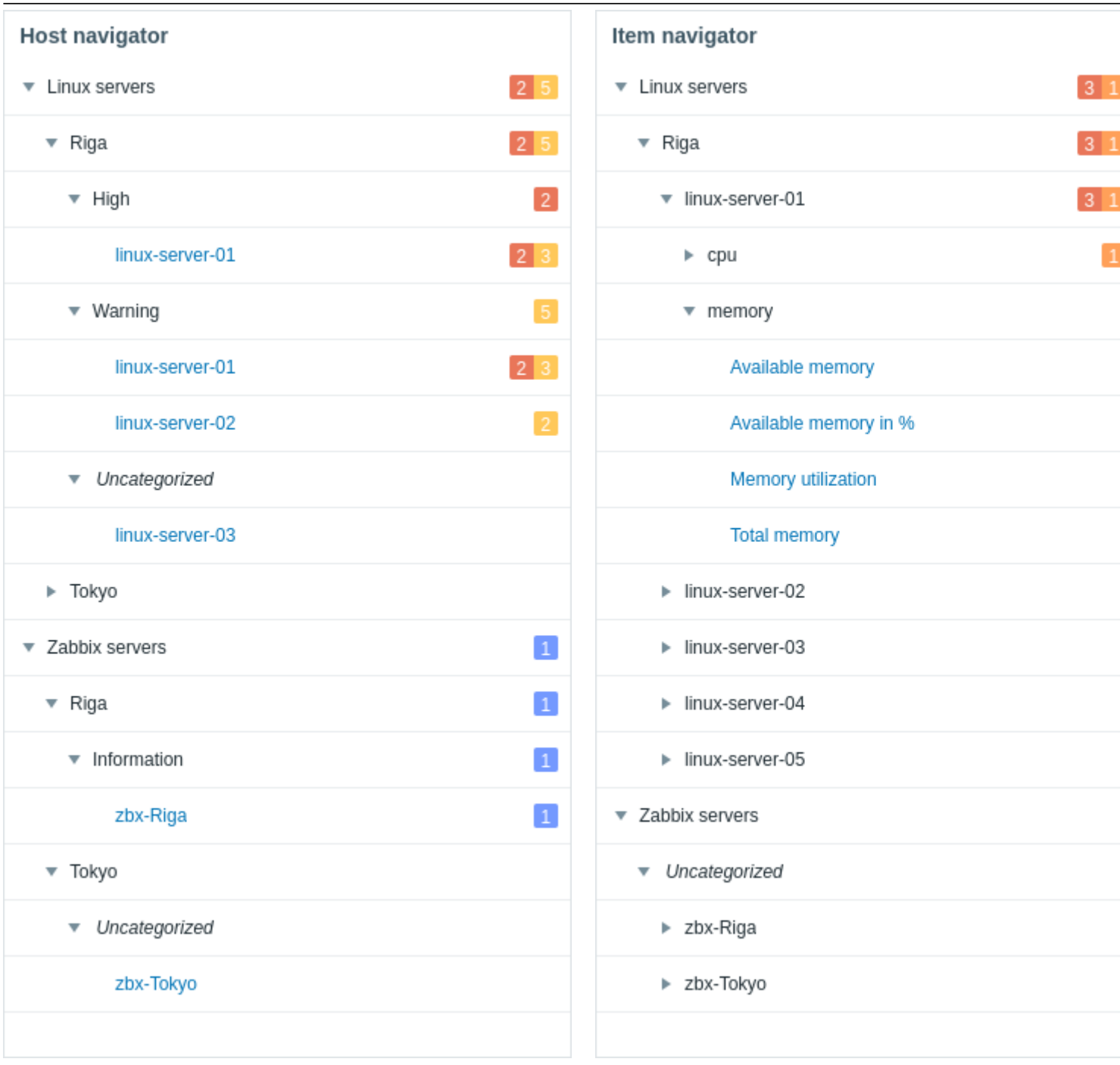
ジオマップウィジェットは、インタラクティブな Leaflet ベースのマップ上にホストをマーカーとして表示し、マーカーの色で各ホストの最も深刻な問題を示します。

ホスト/アイテムの詳細

これらのウィジェットは、ホストまたはアイテムのリストと詳細ビューを提供します。

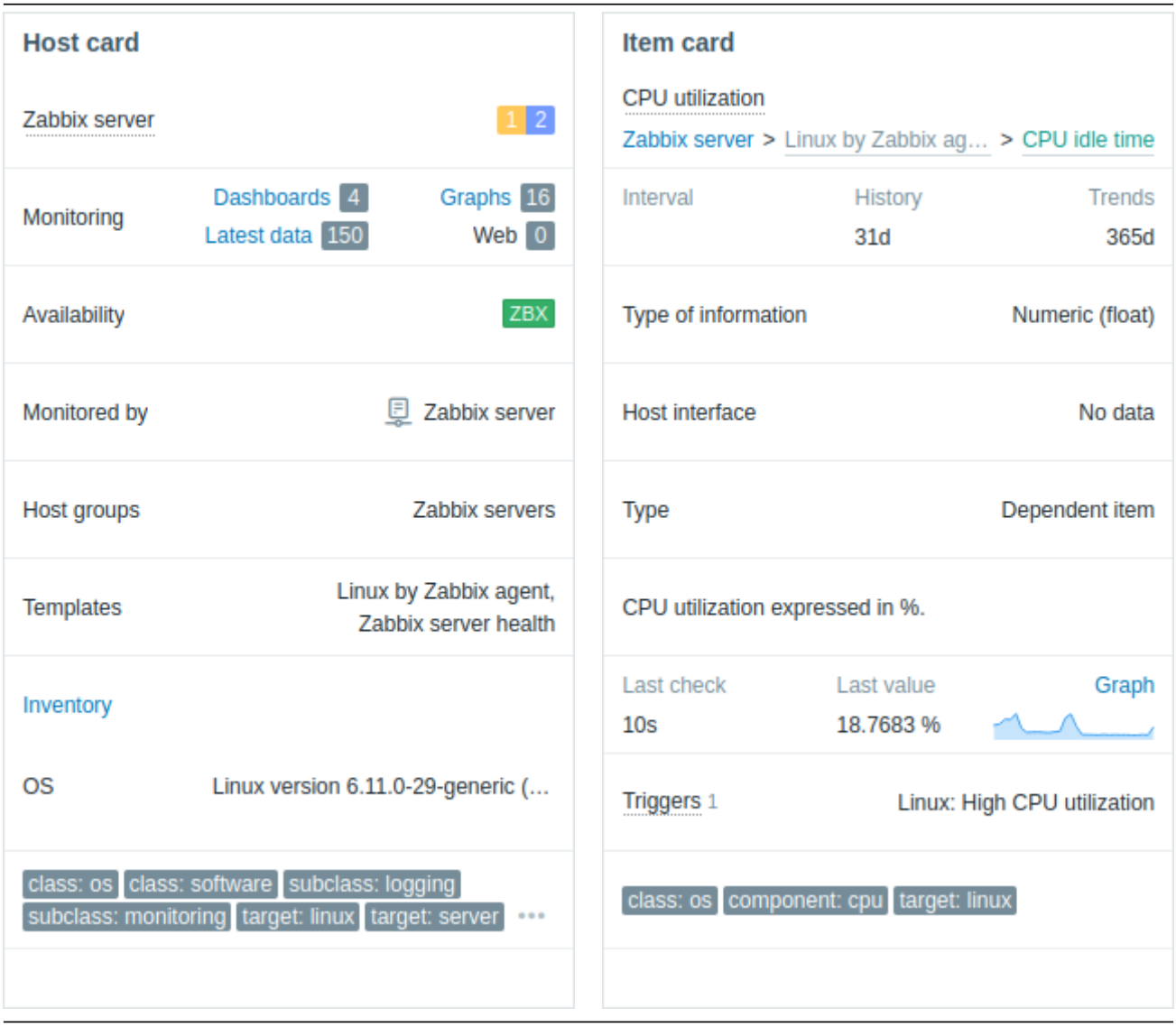
ホスト/アイテムナビゲーター

ホストナビゲーター/アイテムナビゲーターウィジェットは、さまざまなフィルタリングやグループ化オプションに基づいてホストやアイテムのリストを表示します。これらのウィジェットは、選択したホストやアイテムに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。



ホスト/アイテムカード

ホストカード/アイテムカードウィジェットは、単一のホスト/アイテムの詳細を表示し、ホストまたはアイテムのステータスや設定（ホストの可用性、問題、インベントリ、アイテムの最新データ、エラー、説明など）をすばやく確認できます。



ステータスとレポート

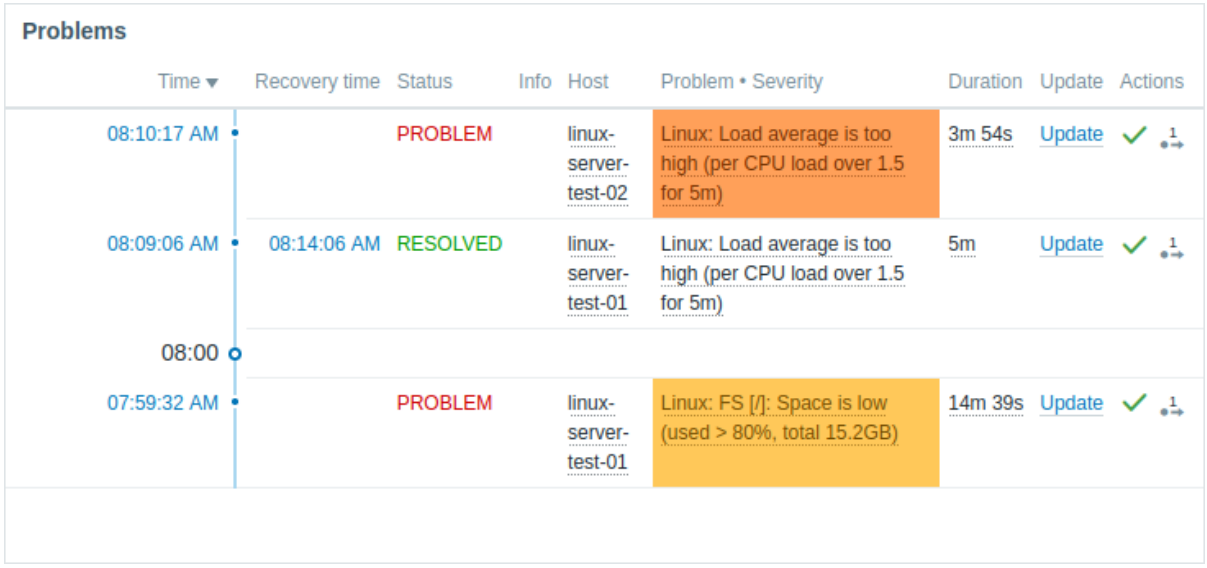
これらのウィジェットは、ステータスデータ（可用性、トリガー、問題など）やレポートを表示します。

ホストの可用性

ホストの可用性ウィジェットは、選択したホストグループ全体のホストの可用性を表示し、稼働中または停止中のホスト数を監視できます。

問題

問題ウィジェットは、現在の問題をさまざまなパラメータ（ホストグループ、ホスト、問題名など）でフィルタリングして表示し、注意が必要な内容を明確に把握できます。



関連情報:

- 深刻度別の問題 - 深刻度ごとにグループ化された原因問題の数を表示します。
- 問題ホスト - ホストグループごとの原因問題の数を表示し、各グループの最大の問題の深刻度を表示します。

トリガー概要

トリガー概要ウィジェットは、ホストトリガーの現在の状態を色付きブロックのテーブルとして表示します。異なるホストのトリガーの健全性とアクティビティの迅速な視覚的スナップショットを提供します。



アクションログ

アクションログウィジェットは、アクション内で実行された操作（通知、リモートコマンド）の詳細を表示します。詳細なログは、監査、トラブルシューティング、および各アクションの実行の監視に役立ちます。

SLA レポート

SLA レポートウィジェットは、SLA レポートを表示し、定義された目標に対するサービスのパフォーマンスを監視するのに役立ちます。

SLA report

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	99.6863	23h 55m 29s	4m 31s	-3m 5s	
2025-08-30	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-08-29	99.9%	99.7095	23h 55m 49s	4m 11s	-2m 45s	
2025-08-28	99.9%	99.9421	23h 59m 10s	50s	36s	
2025-08-27	99.9%	99.7188	23h 55m 57s	3m 15s	-2m 37s	
2025-08-26	99.9%	99.7755	23h 56m 46s	3m 14s	-1m 48s	
2025-08-25	99.9%	99.9094	22h 58m 45s	1h 1m 15s	8s	2025-08-25 01:00 AM Maintenance: 1h

Web 監視

Web 監視ウィジェットは、アクティブな Web シナリオのステータスを表示します。

Web monitoring

Host group ▲	Ok	Failed	Unknown
Applications	1	1	1
Applications/External	1		
Applications/Internal			1
Applications/Test		1	

ディスカバリーステータス

ディスカバリーステータスウィジェットは、有効なネットワークディスカバリールールによって検出されたデバイスのステータスを表示します。ネットワークディスカバリーが正しく機能しているかどうか、および稼働中または停止中のデバイスの数を監視するのに役立ちます。

ダッシュボードのユーティリティとシステム情報

これらのウィジェットは、ダッシュボードを強化するための補足情報やツールを提供します。

システム情報

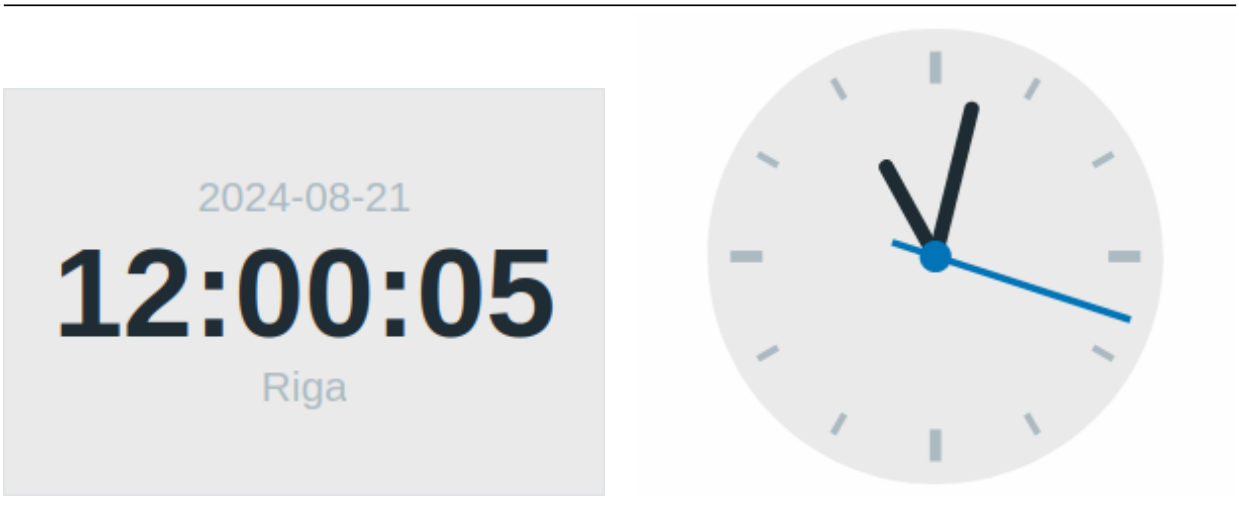
システム情報ウィジェットは、Zabbix サーバーおよびシステムデータの主要な概要、または高可用性ノードの詳細を表示します。

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	8.0.0	Up to date
Zabbix frontend version	8.0.0	Up to date
Software update last checked	2025-08-30	
Latest release	8.0.0	Release notes
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	312	
Number of items (enabled/disabled/not supported)	176	165 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	89	89 / 0 [4 / 85]
Number of users (online)	2	1
Required server performance, new values per second	2.68	
Global scripts on Zabbix server	Disabled	
High availability cluster	Disabled	

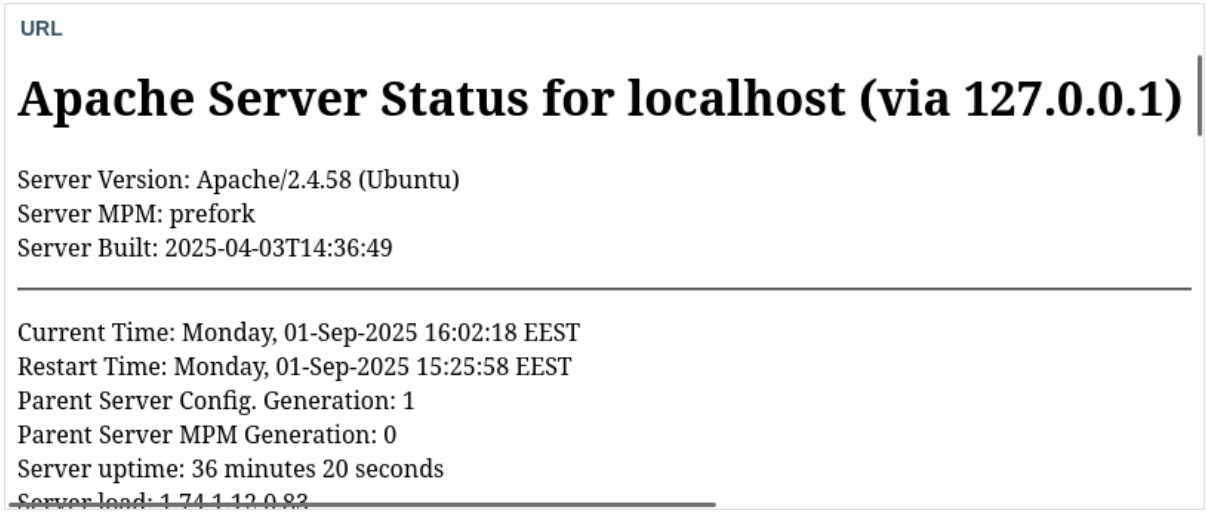
時計

時計ウィジェットは、ローカル、サーバー、またはホストの時刻をアナログまたはデジタル形式で表示します。



URL

URLウィジェットは、URL から取得したコンテンツを表示します。



2 ダッシュボードウィジェットのパラメータ

概要

このページでは、すべてのダッシュボードウィジェットに共通するパラメーターと、複数のウィジェットが互いに、またはダッシュボードと構成データを交換できる動的パラメーターについて説明します。

各ウィジェットの固有のパラメーターについては、サイドバーから該当するウィジェットのページに移動してください。

共通パラメータ

以下のパラメータはすべてのウィジェットに共通です。

名前	ウィジェット名を入力します。
更新間隔	デフォルトの更新間隔を設定します。 ウィジェットのデフォルトの更新間隔は、ウィジェットの種類によって更新なしから *15* 分までさまざまです。例： - *URL* ウィジェットの場合は更新なし; - アクションログウィジェットの場合は *1* 分; - 時計ウィジェットの場合は *15* 分。 更新間隔は、すべてのユーザーに対してデフォルト値に設定できます。ダッシュボードを編集モードに切り替え、ウィジェットの編集ボタンをクリックし、ドロップダウンリストから希望の更新間隔を選択します。 各ユーザーは独自のウィジェット更新間隔を設定することもできます。ダッシュボードの表示モードで、ウィジェットの三点リーダー ボタンをクリックし、ドロップダウンリストから希望の更新間隔を選択します。ユーザー固有の更新間隔はウィジェットの設定よりも優先され、ウィジェットの設定が変更されても保持されることに注意してください。
ヘッダーを表示	チェックボックスをオンにすると、ウィジェットのヘッダーが常に表示されます。 オフの場合、スペースを節約するためにヘッダーは非表示になり、ウィジェットにマウスオーバーしたときのみ表示されます（表示モードと編集モードの両方）。ウィジェットを新しい場所にドラッグするときも、ヘッダーは半透明で表示されます。

動的パラメータ

複数のウィジェットパラメータ（例：ホスト、ホストの上書き、期間）は動的データソースをサポートしています。静的な値を設定する代わりに、これらのパラメータを他のウィジェットやダッシュボード自体にリンクできます。ウィジェットは、ダッシュボードの他の部分からの選択やブロードキャストに基づいて自動的に更新されます。

ウィジェットの互換性

一部のウィジェットは他のウィジェットに設定データをブロードキャストでき、一部はデータをリッスンでき、また一部は両方が可能です。例えば:

- アクションログウィジェットは、グラフ、グラフ (クラシック)、グラフプロトタイプウィジェットから期間データのみを取得できます。

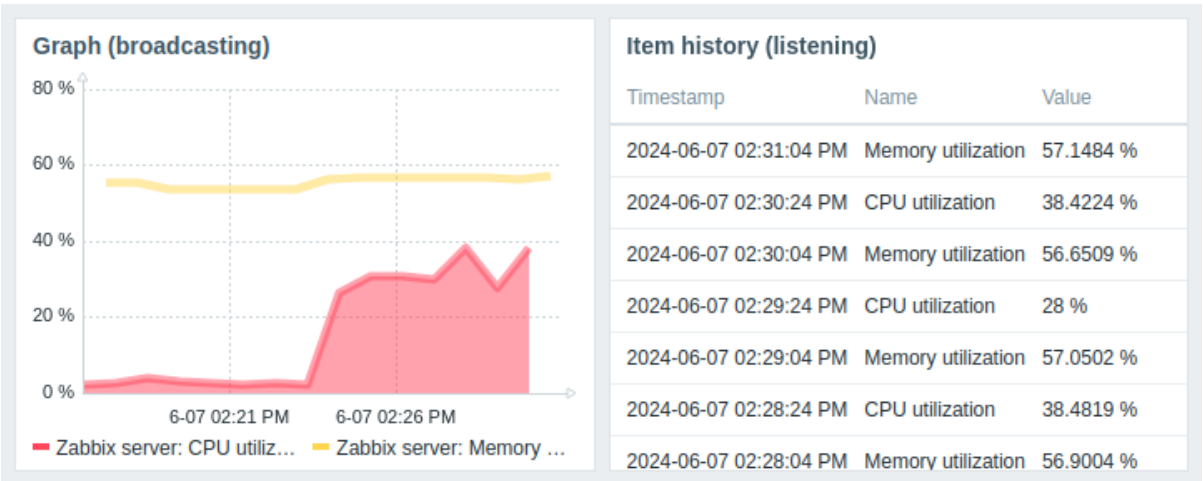
- ジオマップウィジェットは、ホストデータをリッスンするウィジェット (ハニカム、トップアイテムなど) にブロードキャストでき、また、ブロードキャストするウィジェット (ハニカム、問題ホストなど) からホストグループやホストデータをリッスンすることもできます。
- 時計ウィジェットは、いかなるデータもブロードキャストもリッスンもできません。

以下の表は、各ウィジェットのブロードキャストおよびリッスンの機能を示しています。

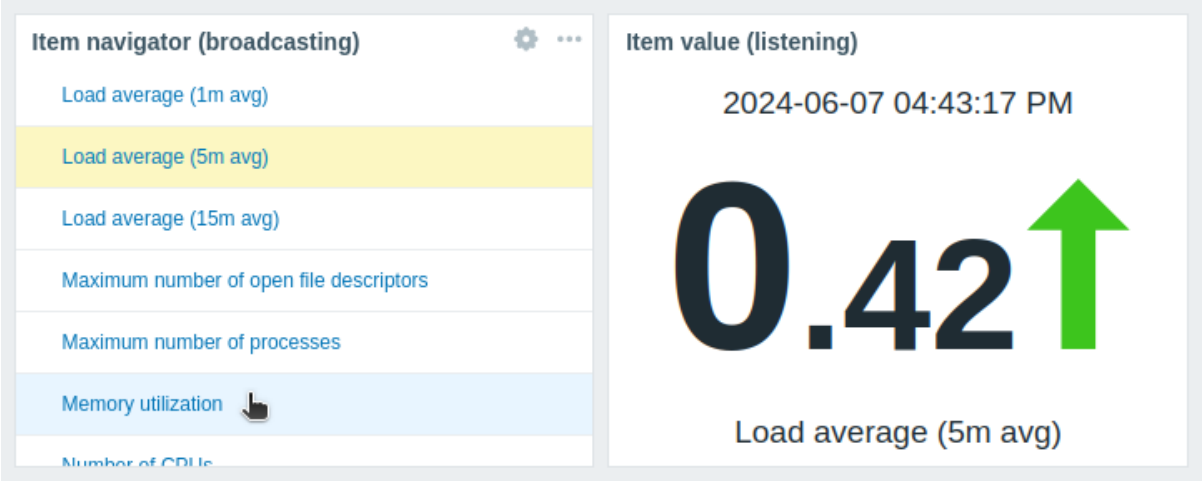
ウィジェット	ブロードキャスト	リッスン
アクションログ	-	期間
時計	-	-
ディスカバリーステータス	-	-
お気に入りグラフ	-	-
お気に入りマップ	-	-
ゲージ	-	ホスト、アイテム
ジオマップ	ホスト	ホストグループ、ホスト
グラフ	期間	期間、アイテム、ホスト
グラフ (クラシック)	期間	ホスト、アイテム、グラフ、期間
グラフプロトタイプ	期間	ホスト、期間
ハニカム	ホスト、アイテム	ホストグループ、ホスト
ホスト可用性	-	ホストグループ
ホストカード	-	ホスト
ホストナビゲーター	ホスト	ホストグループ
アイテムカード	-	ホスト、アイテム、期間
アイテム履歴	アイテム	ホスト、期間
アイテムナビゲーター	アイテム	ホストグループ、ホスト
アイテム値	-	ホスト、アイテム、期間
マップ	ホストグループ、ホスト	マップ
マップナビゲーションツリー	マップ	-
円グラフ	-	期間、アイテム
問題ホスト	ホストグループ	ホストグループ、ホスト
問題	イベント	ホストグループ、ホスト
深刻度別問題	ホストグループ	ホストグループ、ホスト
散布図	期間	ホスト、アイテム、期間
SLA レポート	-	-
システム情報	-	-
トップホスト	ホスト	ホストグループ、ホスト
トップアイテム	-	ホストグループ、ホスト
トップトリガー	-	期間
トリガー概要	-	ホストグループ、ホスト
URL	-	ホスト
ウェブ監視	ホストグループ	ホストグループ、ホスト

ウィジェットは、他のウィジェットにデータをブロードキャストする方法が異なります。

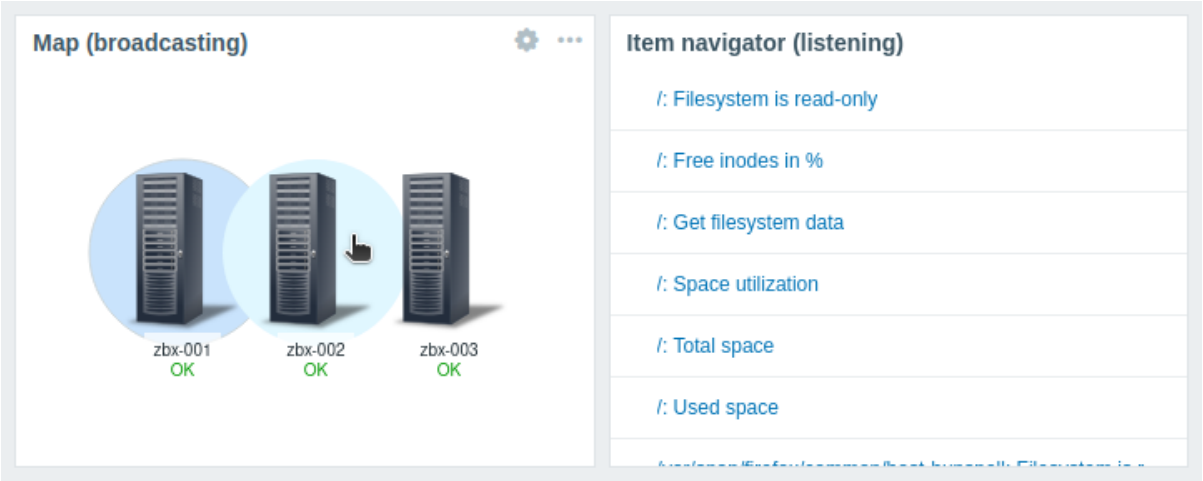
ブロードキャストをサポートするウィジェットは、作成時に自動的にブロードキャストを開始します。たとえば、グラフウィジェットは、リスニングウィジェットに対して期間データを即座にブロードキャストします。



エンティティ選択をサポートするウィジェットは、最初に利用可能なエンティティのデータを自動的にブロードキャストします。たとえば、アイテムナビゲーターウィジェットは、アイテムリスト内の最初のアイテムのデータをブロードキャストします。別のアイテムが選択されると、そのアイテムのデータをブロードキャストします。マウスオーバー時はアイテムが薄い青色でハイライトされ、選択時は黄色でハイライトされます。

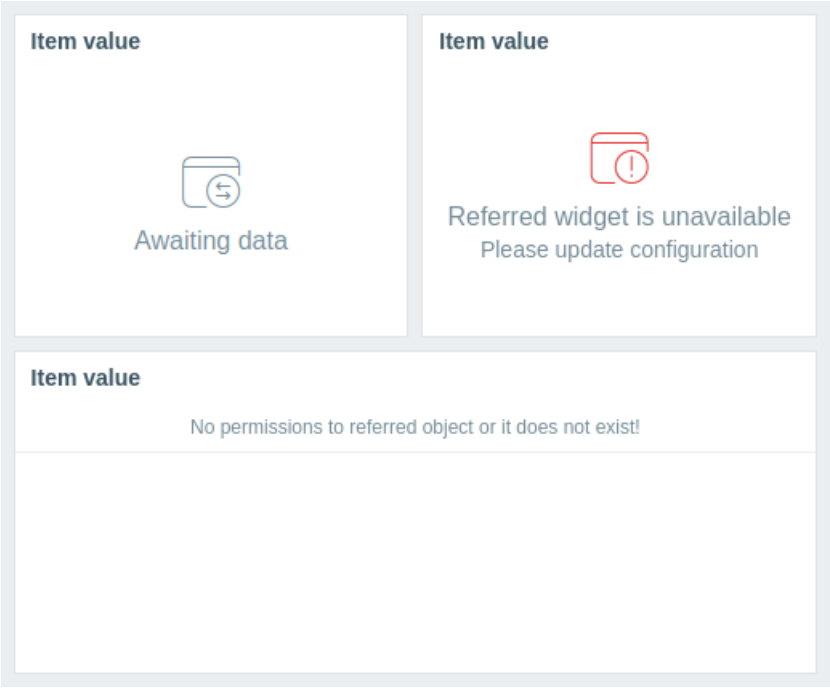


マップウィジェットも同様に動作し、ウィジェットの表示領域の左上隅に最も近い要素（ジオマップウィジェットの場合は中央に最も近い要素）のデータをブロードキャストします。別の要素が選択されると、その要素のデータをブロードキャストします。マウスオーバー時は要素が薄い青色でハイライトされ、選択時は濃い青色でハイライトされます。



ウィジェットは、他のウィジェットからのデータをリッスンする際にも特定の動作をします:

- データソースウィジェットがデータをブロードキャストしていない場合、リスニングウィジェットはデータ待機中状態になります。
- データソースウィジェットが削除された場合、互換性のないウィジェットに置き換えられた場合、または別のダッシュボードページに移動された場合、リスニングウィジェットは参照ウィジェットが利用できません状態になります。
- データソース（ウィジェットまたはダッシュボード）で指定されたホストに、リスニングウィジェットで設定されたエンティティ（アイテム、グラフ、マップなど）が存在しない場合、またはユーザーにホストへのアクセス権限がない場合、リスニングウィジェットには次のメッセージが表示されます: ”参照オブジェクトへの権限がないか、存在しません！”



1 アクションログ

概要

アクションログウィジェットは、アクション内で実行されたオペレーション（通知、リモートコマンド）の詳細を表示します。詳細なログは、各アクションの実行の監査、トラブルシューティング、監視に役立ちます。

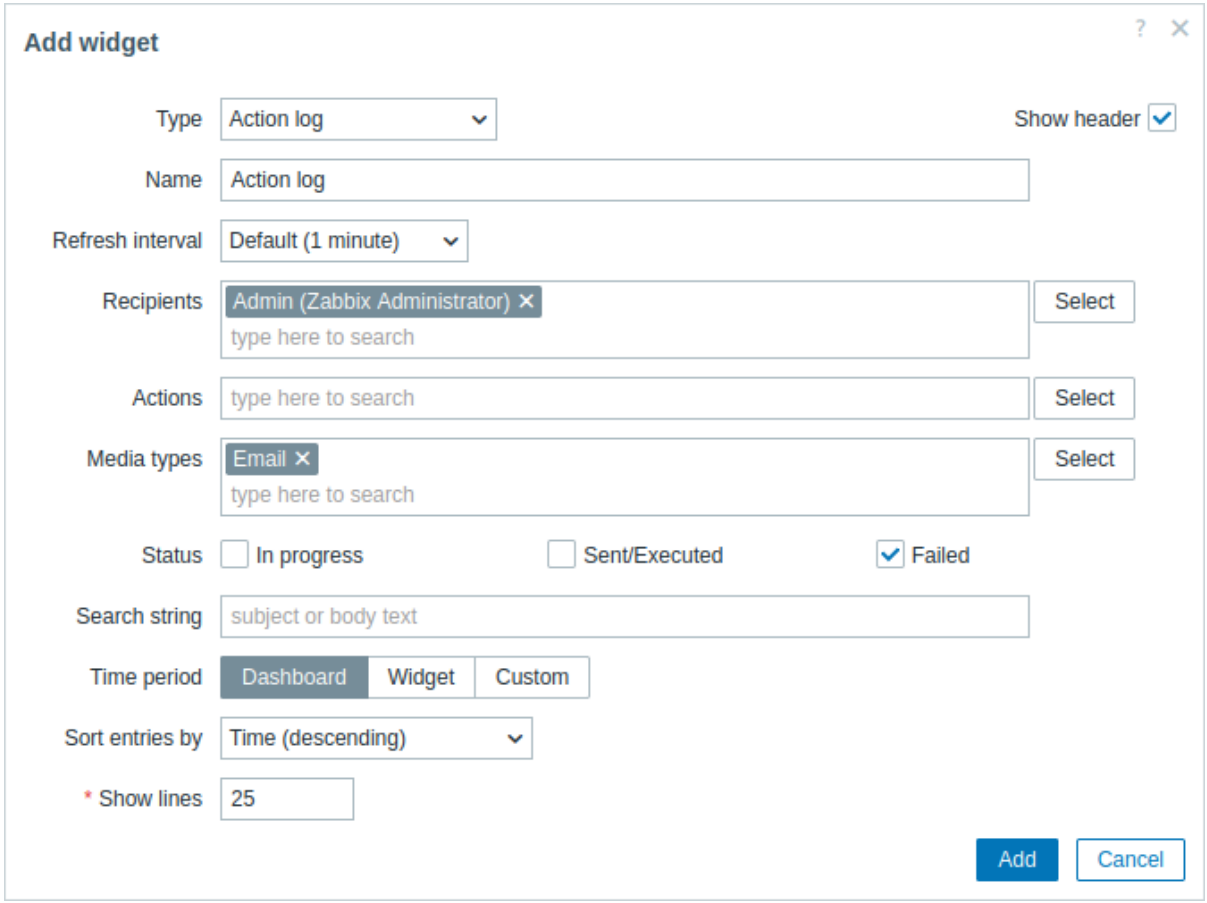
Action log

Time ▾	Action	Media type	Recipient	Message	Status	Info
2025-08-15 07:58:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Resolved in 1m 59s: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem has been resolved at 10:58:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Problem duration: 1m 59s Host: Linux server Severity: Average Original problem ID: 45	Sent	
2025-08-15 07:56:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Problem: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem started at 10:56:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Host: Linux server Severity: Average Operational data: Load averages(1m 5m 15m): (6.256836 4.808105 3.398438), # of CPUs: 2 Original problem ID: 45	Sent	
2025-08-15 07:42:30 AM	Report problems to Zabbix administrators		Admin (Zabbix Administrator)		Failed	i

レポート > アクションログと同じデータを表示し、最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして アクションログを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

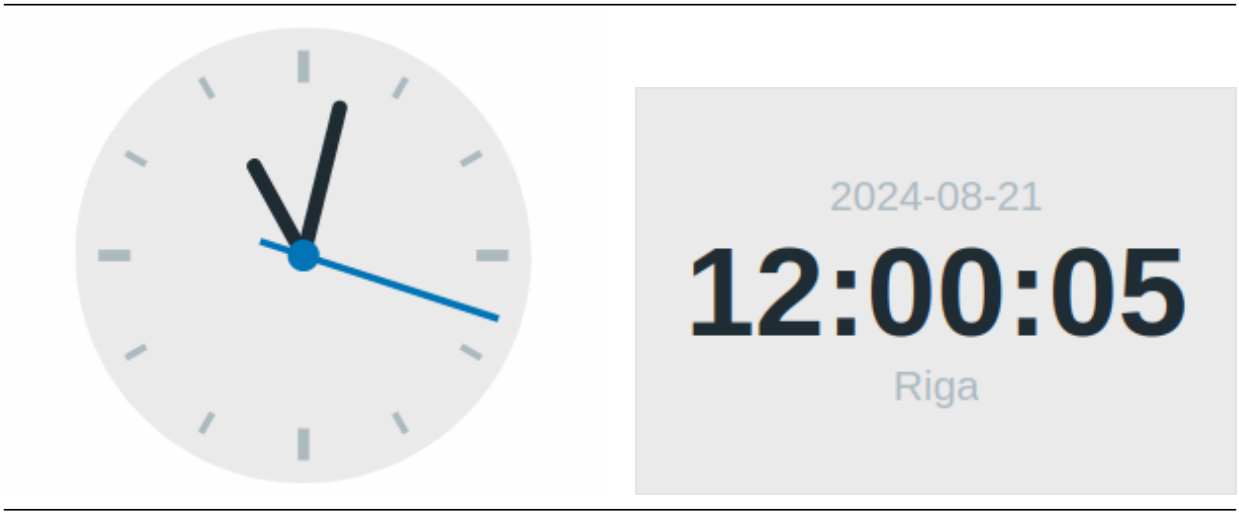
受信者	受信者でエントリをフィルタリングします。このフィールドはオートコンプリート対応で、受信者名の入力を始めると一致する受信者のドロップダウンが表示されます。受信者が選択されていない場合、すべての受信者のアクション操作の詳細が表示されます。
アクション	アクションでエントリをフィルタリングします。このフィールドはオートコンプリート対応で、アクション名の入力を始めると一致するアクションのドロップダウンが表示されます。アクションが選択されていない場合、すべてのアクションのアクション操作の詳細が表示されます。
メディアタイプ	メディアタイプでエントリをフィルタリングします。このフィールドはオートコンプリート対応で、メディアタイプ名の入力を始めると一致するメディアタイプのドロップダウンが表示されます。メディアタイプが選択されていない場合、すべてのメディアタイプのアクション操作の詳細が表示されます。
ステータス	それぞれのステータスでエントリをフィルタリングするにはチェックボックスをマークします: 進行中 - 進行中のアクション操作が表示されます; 送信済み/実行済み - 通知が送信された、または実行されたアクション操作が表示されます; 失敗 - 失敗したアクション操作が表示されます。
検索文字列	メッセージ/リモートコマンドの内容でエントリをフィルタリングします。ここに文字列を入力すると、その文字列を含むメッセージ/リモートコマンドのアクション操作のみが表示されます。マクロは展開されません。
期間	期間でエントリをフィルタリングします。期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できることに注意してください。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。

エントリの並び順	エントリを以下で並び替えます: 時刻（降順または昇順）; タイプ（降順または昇順）; ステータス（降順または昇順）; 受信者（降順または昇順）。
表示行数	ウィジェットに表示するアクションログの行数を設定します。

2 時計

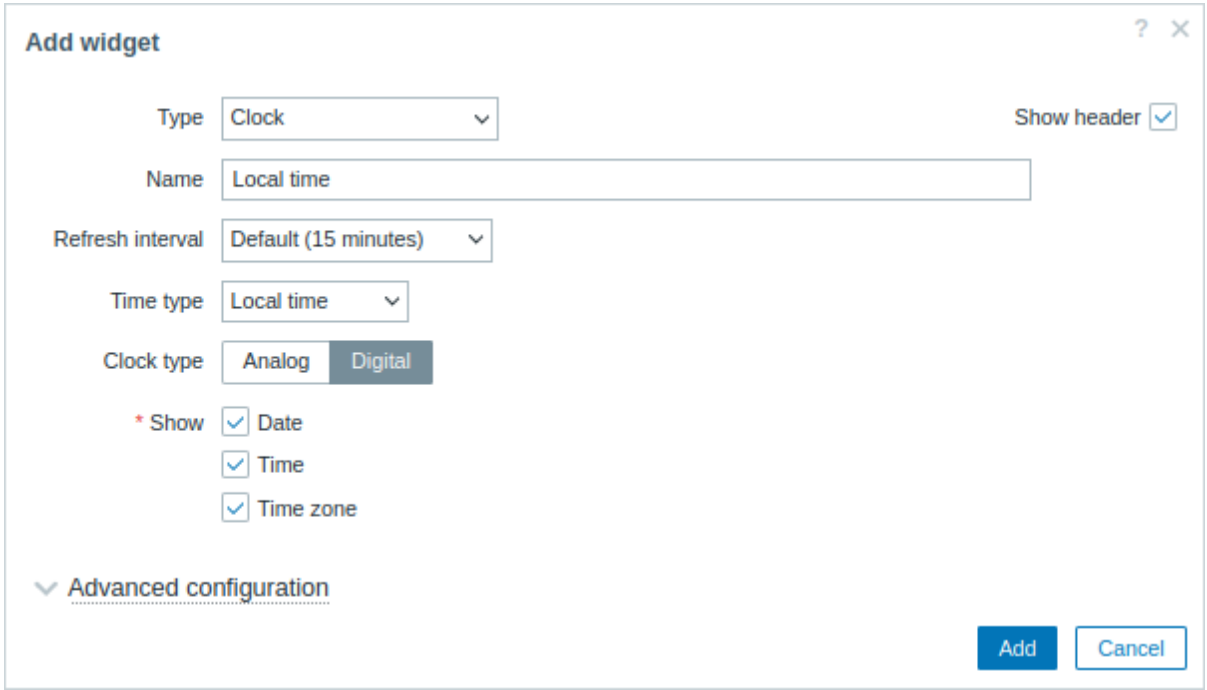
概要

時計ウィジェットは、アナログまたはデジタル形式でローカル、サーバー、またはホストの時刻を表示します。



設定

設定するには、タイプとして *Clock* を選択します:



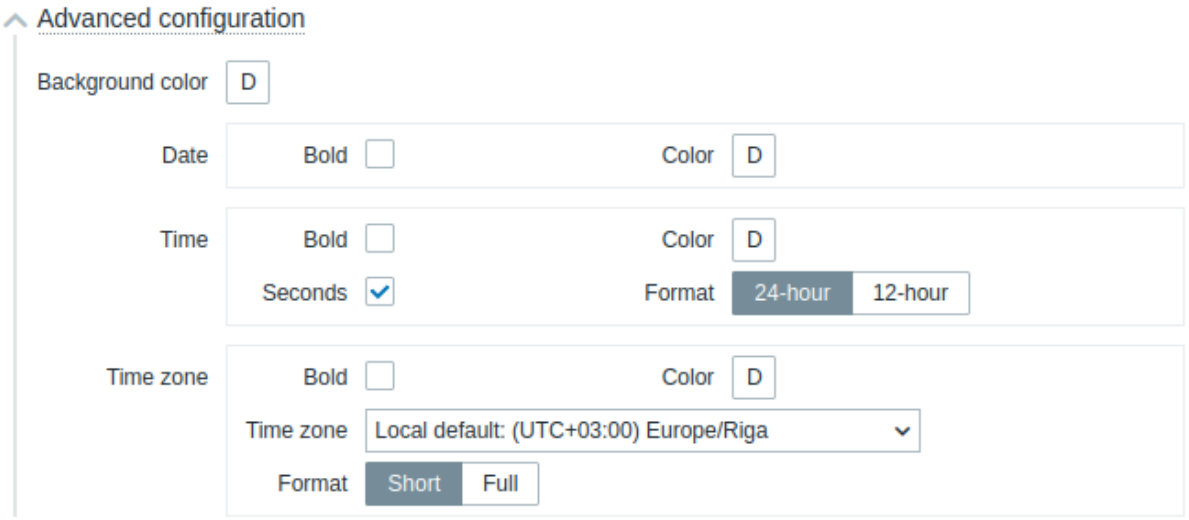
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

時刻タイプ	時刻のソースを選択します: ローカル - ブラウザを実行しているデバイスのシステム時刻; サーバー - グローバルまたは Zabbix ユーザーに設定されたタイムゾーン; ホスト - ホストアイテムから取得した時刻。
アイテム	ホスト時刻のアイテムを選択します（system.localtime[local]アイテムを使用）。 このパラメータは、時刻タイプが「ホスト時刻」に設定されている場合に利用できます。
時計タイプ	時計のタイプ（アナログまたはデジタル）を選択します。
表示	デジタル時計に表示する情報単位（日付、時刻、タイムゾーン）を 1 つ以上選択します。 単位サイズは、表示する単位、ウィジェットサイズ、表示スケーリングに自動的に適応します。 このパラメータは、時計タイプが「デジタル」に設定されている場合に利用できます。
詳細設定	詳細設定ラベルをクリックすると、デジタル時計の詳細設定オプションが表示されます。 このセクションは、時計タイプが「デジタル」に設定されている場合に利用できます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用でき、表示フィールドで選択された要素にのみ適用されます(上記参照)。

さらに、詳細設定ではウィジェット全体の背景色を変更することができます。

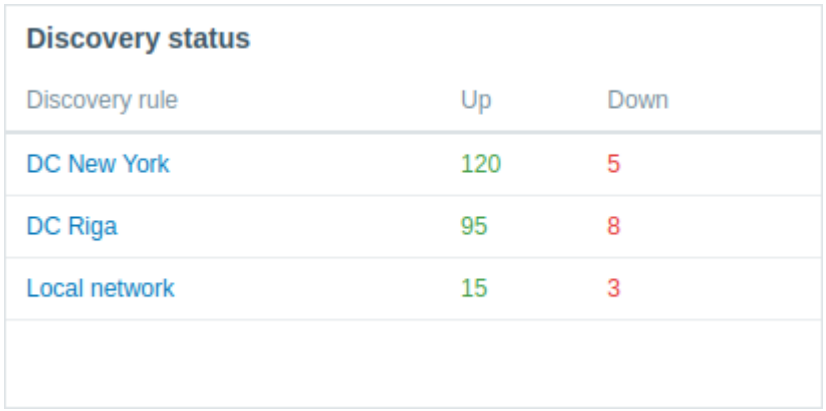


背景色	カラーピッカーから背景色を選択します。 D はデフォルト色（フロントエンドのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
日付	
太字	チェックボックスをオンにすると、日付が太字で表示されます。
色	カラーピッカーから日付の色を選択します。 D はデフォルト色（フロントエンドのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
時刻	
太字	チェックボックスをオンにすると、時刻が太字で表示されます。
色	カラーピッカーから時刻の色を選択します。 D はデフォルト色（フロントエンドのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
秒	チェックボックスをオンにすると、秒も表示されます。オフの場合は時と分のみが表示されます。
フォーマット	24 時間表示または 12 時間表示を選択します。
タイムゾーン	
太字	チェックボックスをオンにすると、タイムゾーンが太字で表示されます。
色	カラーピッカーからタイムゾーンの色を選択します。 D はデフォルト色（フロントエンドのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
タイムゾーン	タイムゾーンを選択します。
フォーマット	タイムゾーンを短い形式（例：New York）または完全な形式（例：(UTC-04:00) America/New York）で表示するか選択します。

3 ディスカバリーステータス

概要

ディスカバリーステータスウィジェットは、有効なネットワークディスカバリールールによって検出されたデバイスのステータスを表示します。ネットワークディスカバリーが正しく動作しているかどうか、また、稼働中または停止中のデバイスの数を監視するのに役立ちます。



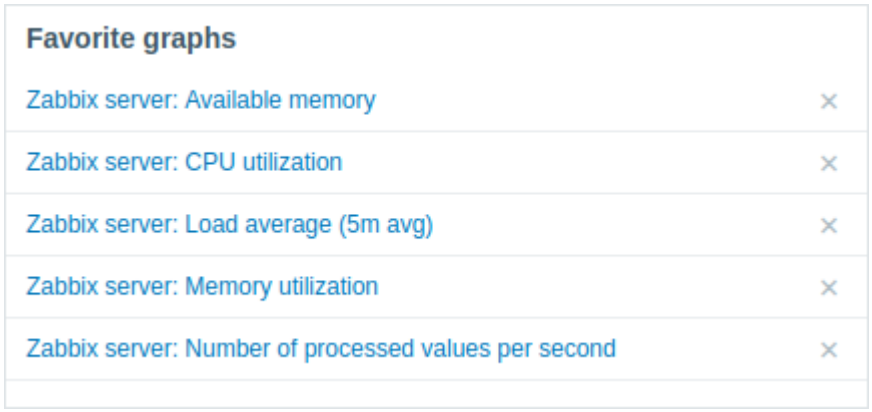
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

4 お気に入りグラフ

概要

お気に入りグラフウィジェットは、現在のユーザーがお気に入りとしてマークしたグラフへのショートカットのアルファベット順リストを表示します。



x をクリックすると、リストとお気に入りからグラフが削除されます。

グラフは、監視 > 最新データでアイテムグラフを表示しているときに、  お気に入りに追加ボタンをクリックすることで、お気に入りとしてマークされます。



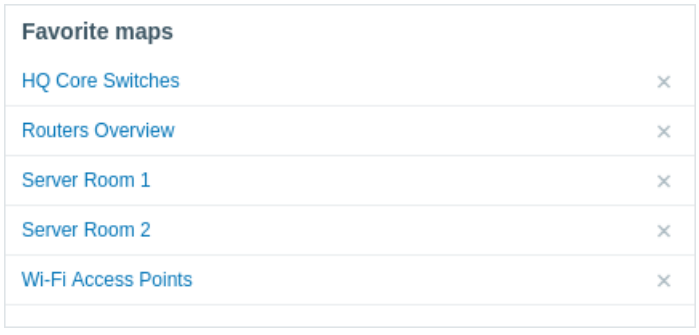
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

5 お気に入りのマップ

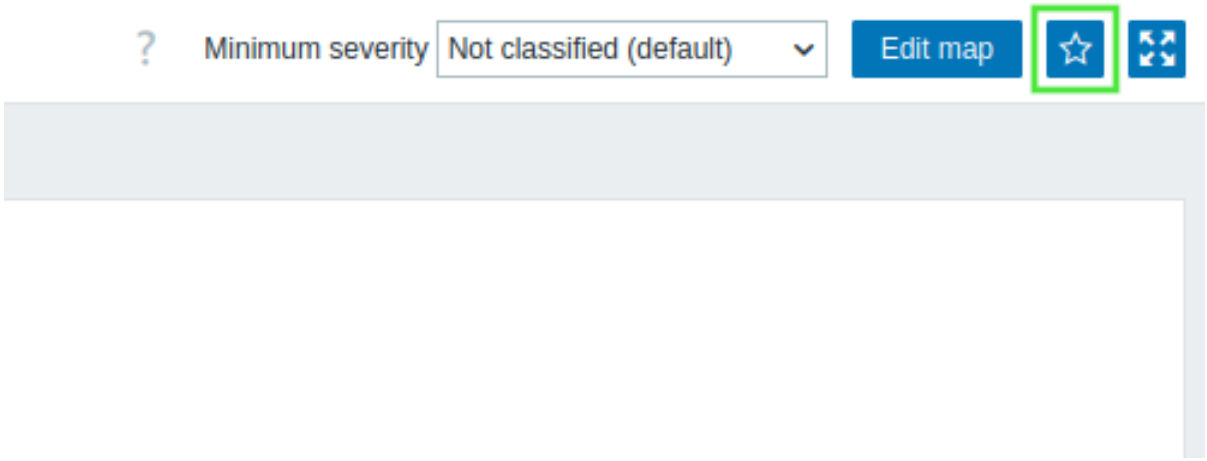
概要

お気に入りのマップウィジェットは、現在のユーザーがお気に入りに登録したマップへのショートカットのリストをアルファベット順に表示します。



x をクリックすると、お気に入りからマップが削除されます。

マップは、監視 > マップでマップを表示しているときに、  お気に入りに追加ボタンをクリックすることでお気に入りに登録されます。



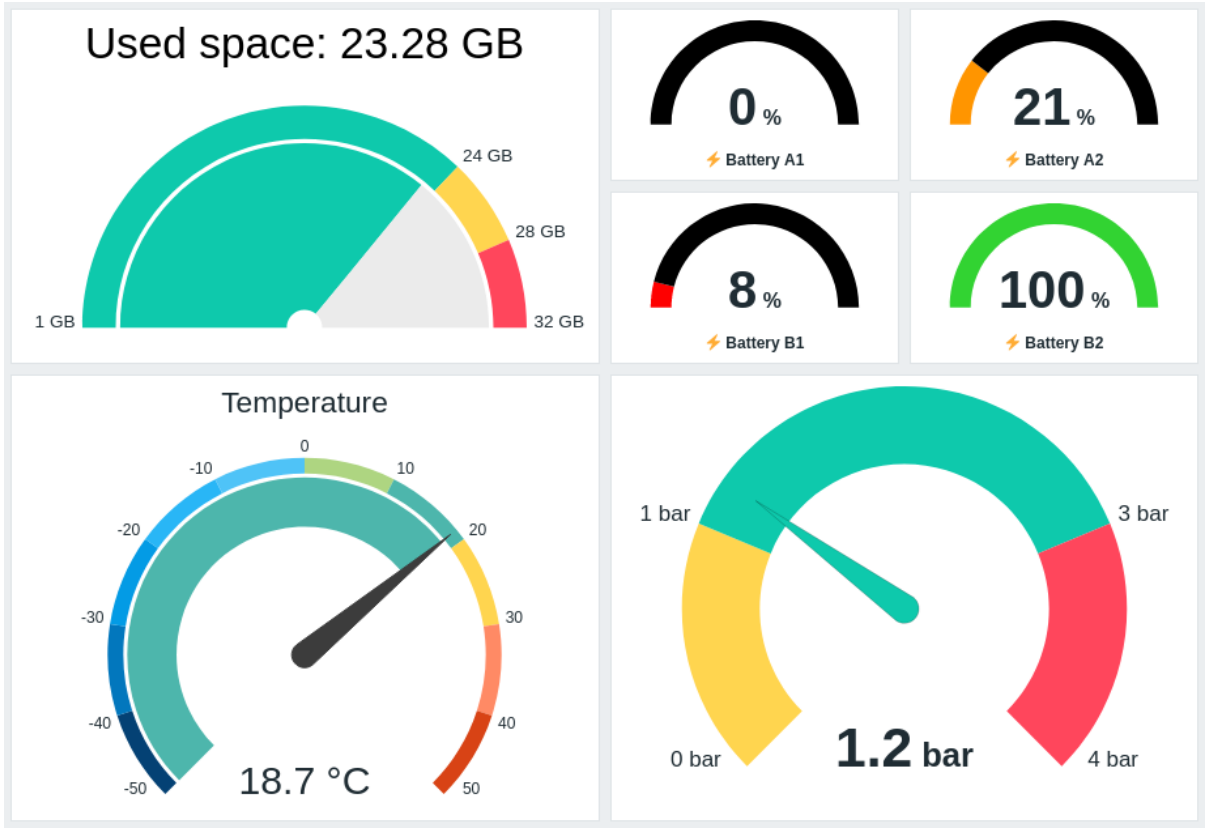
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

6 ゲージ

概要

ゲージウィジェットは、単一の数値アイテムの値をゲージとして表示します。重要なメトリクスを監視したり、しきい値を可視化したり、急激な変化を検出したりするのに便利です。



ウィジェットで以下を表示するように設定できます:

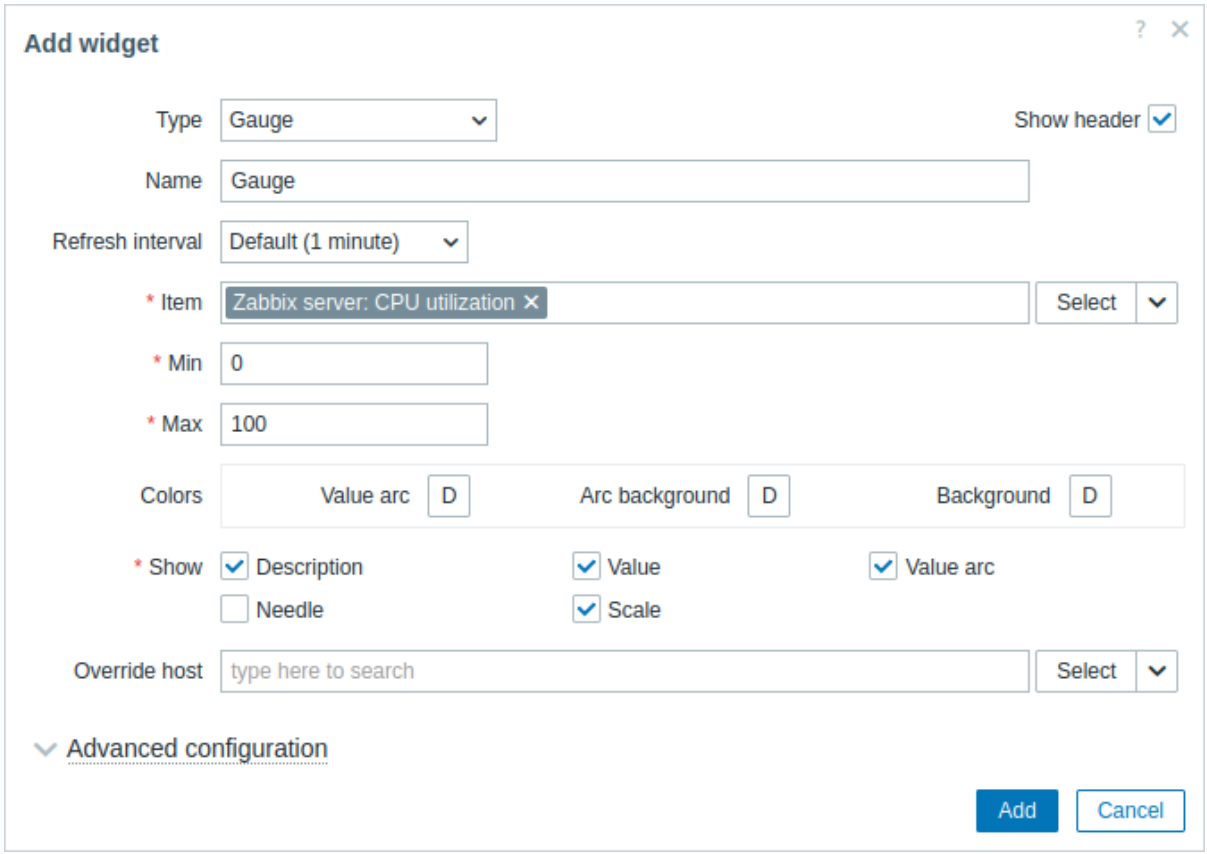
- アイテムの説明（Used space: {ITEM.LASTVALUE}, Temperature）
- アイテムの値と単位（21 %, 18.7 °C, 1.2 bar）
- ゲージの目盛り（1GB/24GB/28GB/32GB, -50/-40/-30/etc.）
- ゲージのアーク（ゲージ値のアークとゲージしきい値のアーク）
- ゲージの針

ウィジェットをクリックすると、そのアイテムのシンプルグラフが開きます。

また、レンダリングされたウィジェットを.png 画像としてエクスポートすることもできます。

設定

設定するには、タイプとしてゲージを選択します:



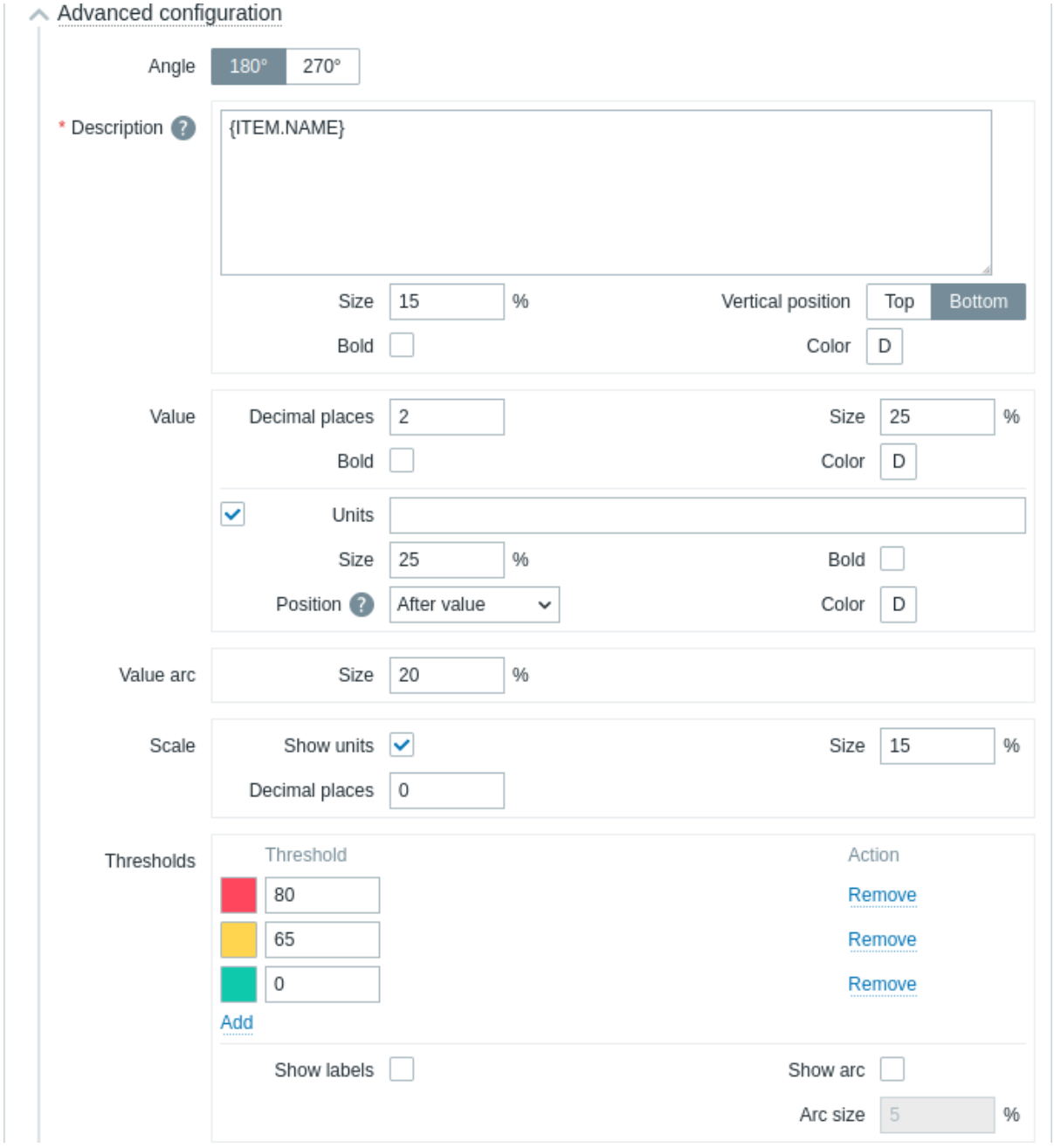
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

アイテム	アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、アイテム名の入力を始めると一致するアイテムのドロップダウンが表示されます。 数値データ（バイナリデータを除く）を返すアイテムのみ選択できることに注意してください。
最小値	ゲージの最小値を入力します。 サフィックス（例: ”1d”, ”2w”, ”4K”, ”8G”）がサポートされています。値のマッピングもサポートされています。
最大値	ゲージの最大値を入力します。 サフィックス（例: ”1d”, ”2w”, ”4K”, ”8G”）がサポートされています。値のマッピングもサポートされています。
色	カラーピッカーから色を選択します: 値アーク - ゲージ値アークの色を選択します; アーク背景 - ゲージ値アークおよびゲージしきい値アークの背景色を選択します; 背景 - ウィジェットの背景色を選択します。 ”D” はデフォルト色を表し、これはフロントエンドのテーマによって異なります。しきい値が設定されている場合、値アークのデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
表示	それぞれのゲージ要素（説明、値、値アーク、針、スケール（ゲージアークの始点と終点の最小値と最大値））を表示するにはチェックボックスをオンにします。非表示にするにはオフにします。少なくとも 1 つの要素を選択する必要があります。 ゲージの針とスケールは、ゲージ値アークまたはゲージしきい値アーク（詳細設定オプション参照）が表示されている場合に表示できることに注意してください。また、ゲージの針が表示されている場合は値が針の下に配置され、針が非表示の場合は値がゲージアークの下部に揃えられることにも注意してください。
ホストの上書き	ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。ここでは、表示フィールドで選択したゲージ要素も調整できます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用できます。



角度	ゲージの角度（180° または 270°）を選択します。
説明	
説明	アイテムの説明を入力します。この説明はデフォルトのアイテム名を上書きする場合があります。 複数行の説明がサポートされています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*} およびユーザーマクロがサポートされています。
サイズ	アイテムの説明のフォントサイズの高さを入力します（パーセント、ウィジェット全体の高さに対する割合）。
垂直位置	アイテムの説明の垂直位置（ゲージアークに対して上または下）を選択します。
太字	チェックボックスをオンにすると、アイテムの説明が太字で表示されます。
色	カラーピッカーからアイテムの説明の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

値	
小数点以下の桁数	値とともに表示する小数点以下の桁数を入力します。 このオプションは、数値（浮動小数点）データを返すアイテムにのみ影響します。
サイズ	値のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
太字	チェックボックスをオンにすると、値が太字で表示されます。
色	カラーピッカーから値の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
単位	
単位	チェックボックスをオンにすると、アイテム値とともに単位が表示されます。 単位名を入力すると、アイテム設定で設定された単位を上書きします。
サイズ	アイテム単位のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
太字	チェックボックスをオンにすると、アイテム単位が太字で表示されます。
位置	アイテム単位の位置（値に対して上、下、前、後）を選択します。 このオプションは、次の時間関連の単位（unixtime、uptime、s）には無視されます。
色	カラーピッカーからアイテム単位の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
値アーク	
アークサイズ	ゲージ値アークのサイズの高さを入力します（パーセント、ゲージアーク半径に対する割合）。
針	
色	カラーピッカーからゲージ針の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。しきい値が設定されている場合、針のデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
目盛り	
単位を表示	チェックボックスをオンにすると、ゲージの最小値と最大値に単位が表示されます。
サイズ	ゲージの最小値と最大値のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
小数点以下の桁数	ゲージの最小値と最大値とともに表示する小数点以下の桁数を入力します。 このオプションは、数値（浮動小数点）データを返すアイテムにのみ影響します。
しきい値	
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値リストは保存時に昇順でソートされます。 しきい値として設定された色は、数値アイテムに対してのみ正しく表示されることに注意してください。 サフィックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています。値のマッピングもサポートされています。
ラベルを表示	チェックボックスをオンにすると、しきい値がゲージ目盛りのラベルとして表示されます。
アークを表示	チェックボックスをオンにすると、ゲージしきい値アークが表示されます。
アークサイズ	ゲージしきい値アークのサイズの高さを入力します（パーセント、ゲージアーク半径に対する割合）。

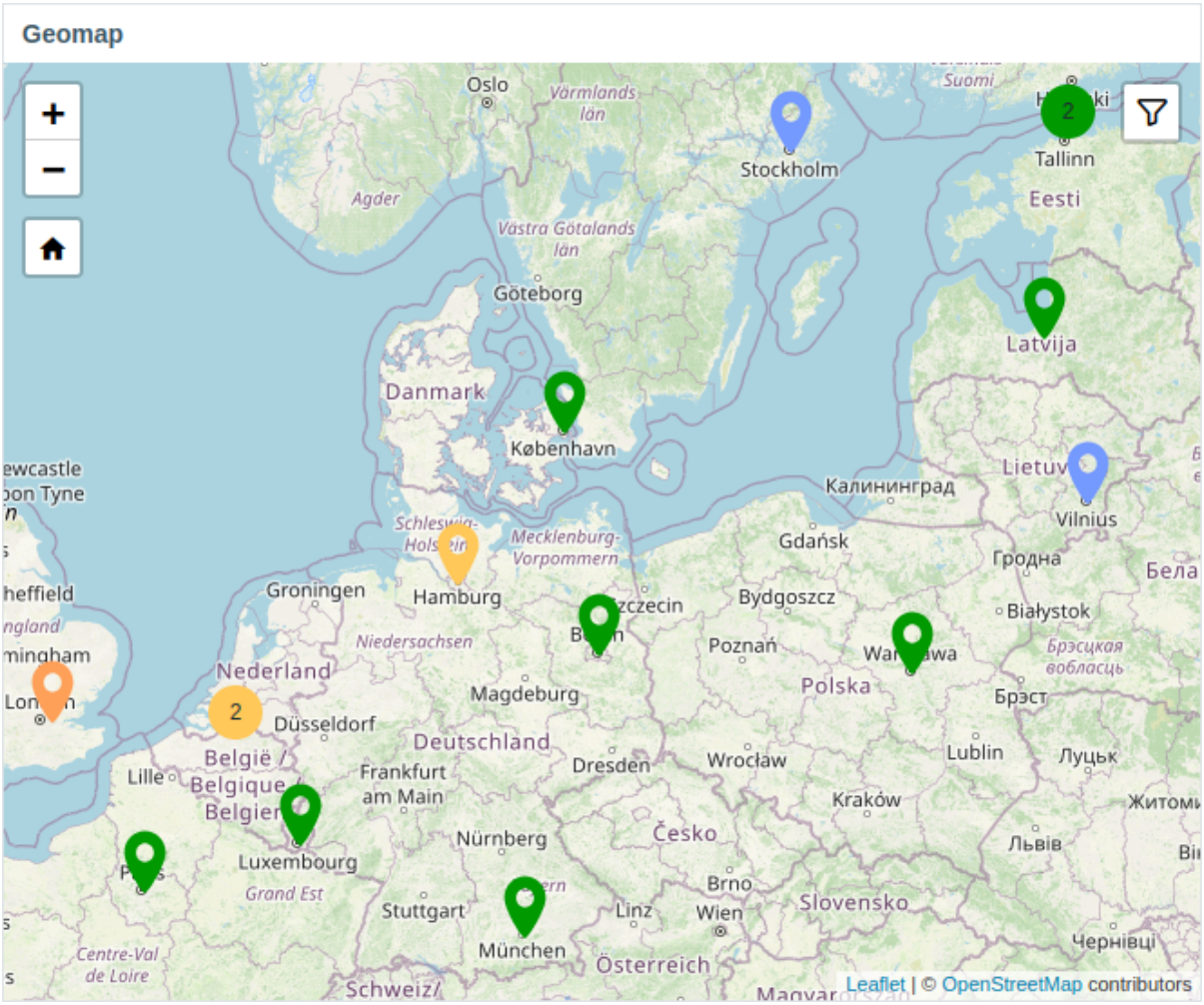
ゲージウィジェットで表示される情報は、ウィジェットメニューを使用して*.png* 画像としてダウンロードできます。

ウィジェットのスクリーンショットはダウンロードフォルダーに保存されます。

7 Geomap

概要

Geomap ウィジェットは、インタラクティブな Leaflet ベースの地図上にホストをマーカーとして表示し、マーカーの色で各ホストの最も深刻な問題を示します。

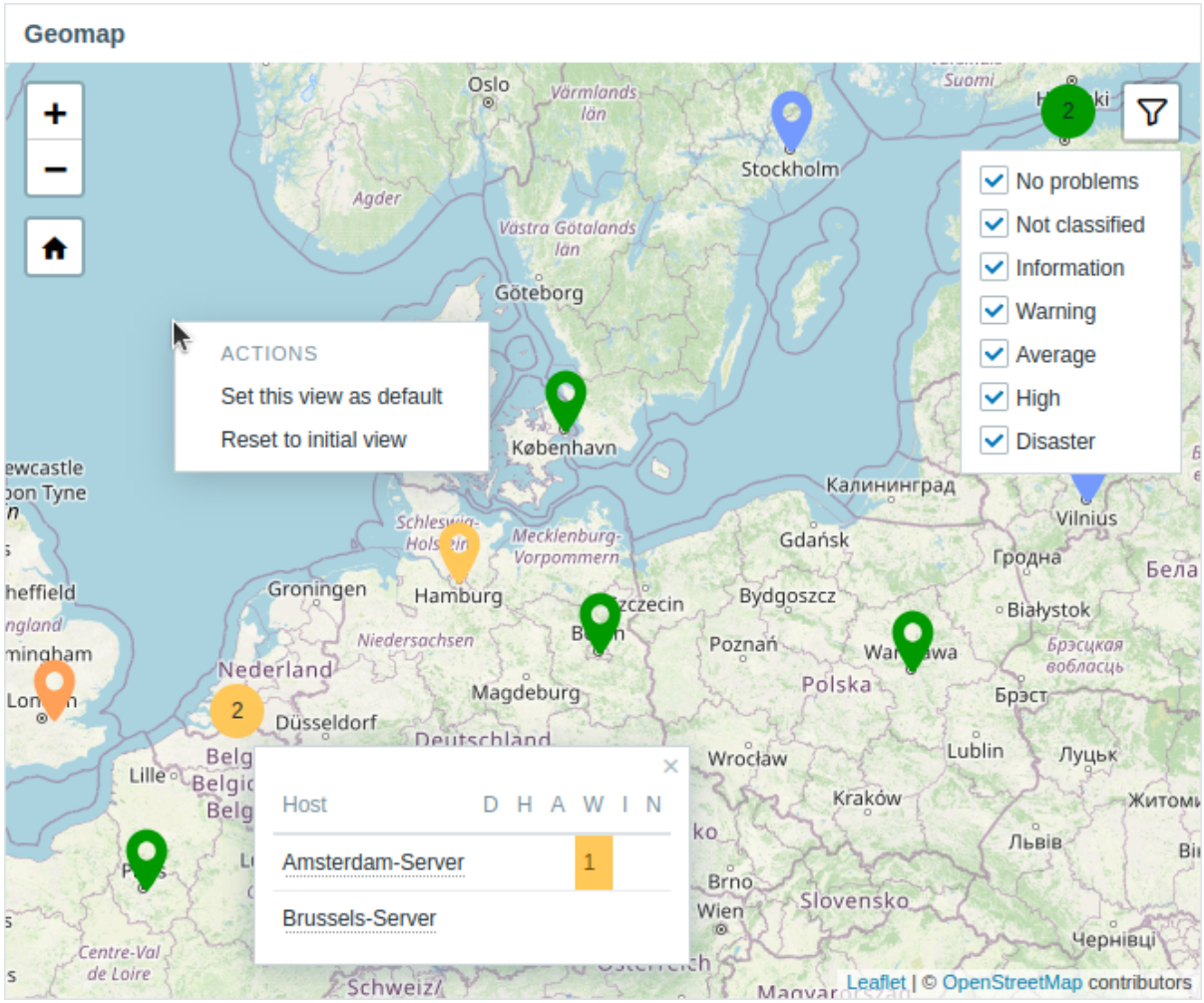


Note:
Zabbix は複数の定義済み地図タイルプロバイダーを提供しており、管理 > 一般 > 地理的地図でカスタムプロバイダーを追加できます。

デフォルトでは、ウィジェットはインベントリに有効な座標（緯度-90～90、経度-180～180）が設定されている有効なホストをすべて表示します。

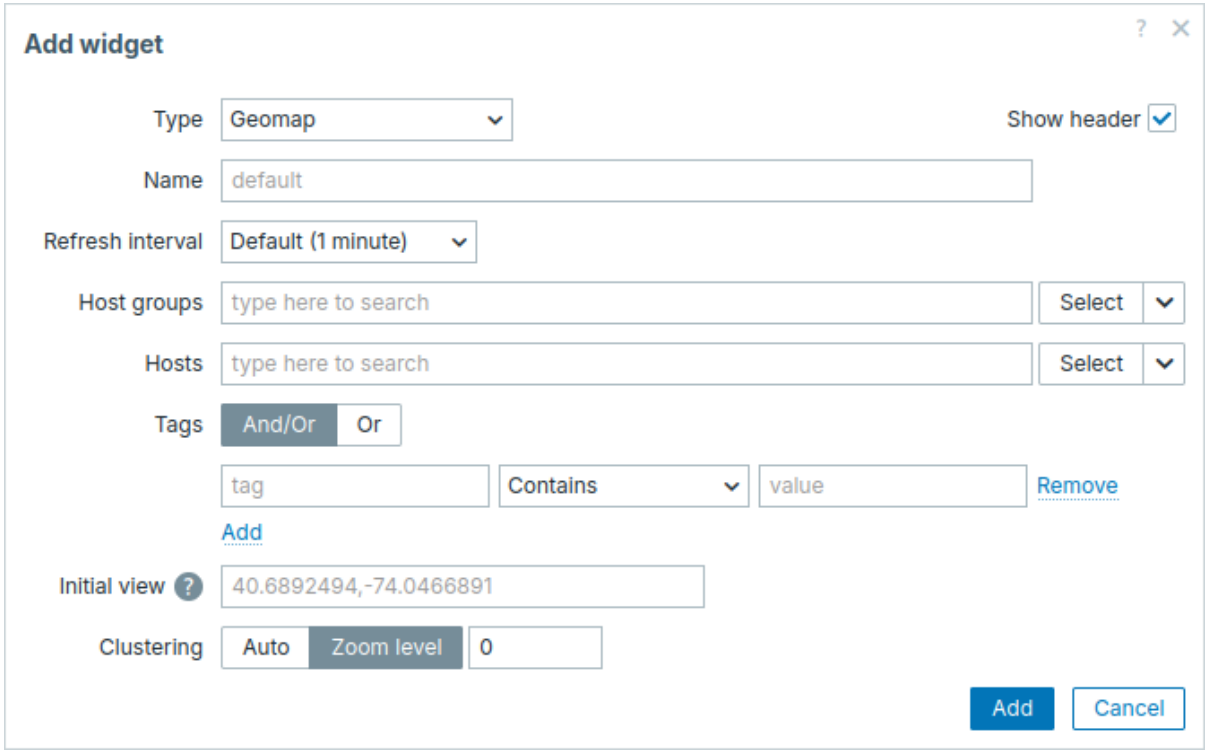
ウィジェットと対話するには、以下のオプションを使用します：

- マーカーをクリックするとホスト名と未解決の問題が表示され、ホスト名をクリックするとホストメニューが開きます。
- 右上のフィルターアイコンで問題の深刻度でホストをフィルタリングできます。
- 地図を右クリックしてデフォルトビューを設定またはリセットし、ホームアイコンでデフォルトビューに戻ります。



設定

ウィジェットを追加するには、タイプとして *Geomap* を選択します。



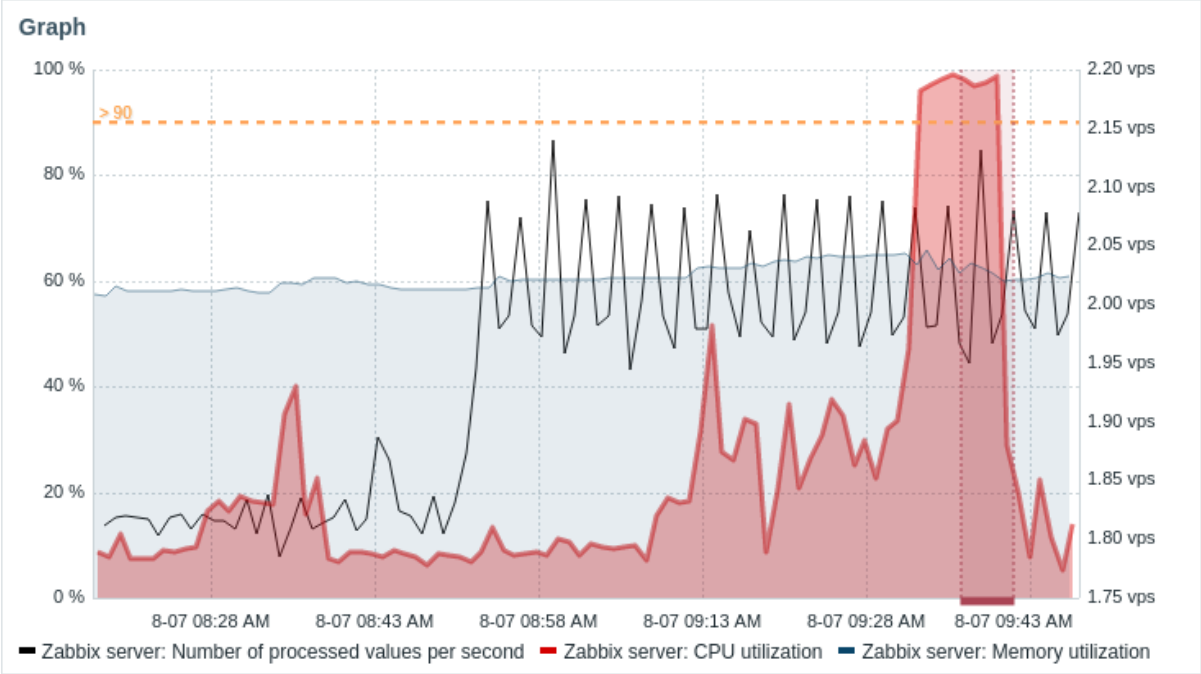
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	マップ上に表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 ホストグループとホストの両方のフィールドで何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	マップ上に表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応のため、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストグループとホストの両方のフィールドで何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
タグ	ウィジェットに表示されるホスト数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を含める; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を除外する。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
初期ビュー	このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。 ウィジェットが最初に読み込まれたときに表示する中心座標とオプションのズームレベルをカンマ区切りで < 緯度 >,< 経度 >,< ズーム > の形式で指定します。 初期ズームが指定されている場合、Geomap ウィジェットは指定されたズームレベルで読み込まれます。指定されていない場合、初期ズームはタイルプロバイダーごとの最大ズームの半分として計算されます。 デフォルトビューが設定されている場合（下記参照）、初期ビューは無視されます。 例: 40.6892494,-74.0466891 40.6892494,-122.0466891
クラスタリング	近接するホストマーカーをカウント付きの単一マーカーにまとめる方法を指定します: 自動 - マーカーは自動的にまとめられます; ズームレベル - 指定したズームレベルに基づいてマーカーがまとめられます（0 - 最大ズームアウト、30 - 最大ズームイン）。たとえば、パラメータが 0 の場合、どのズームレベルでもマーカーはまとめられません。15 の場合、最大ズームアウトからズームレベル 15 までマーカーがまとめられます。30 の場合、すべてのズームレベルでマーカーがまとめられます。 最大ズームインレベルはマップタイルプロバイダーによって決まることに注意してください。

8 グラフ

概要

グラフウィジェットは、数値アイテムデータをベクターグラフとして表示します。これにより、傾向の追跡、問題の発見、時間やホスト間での値の比較が可能になります。

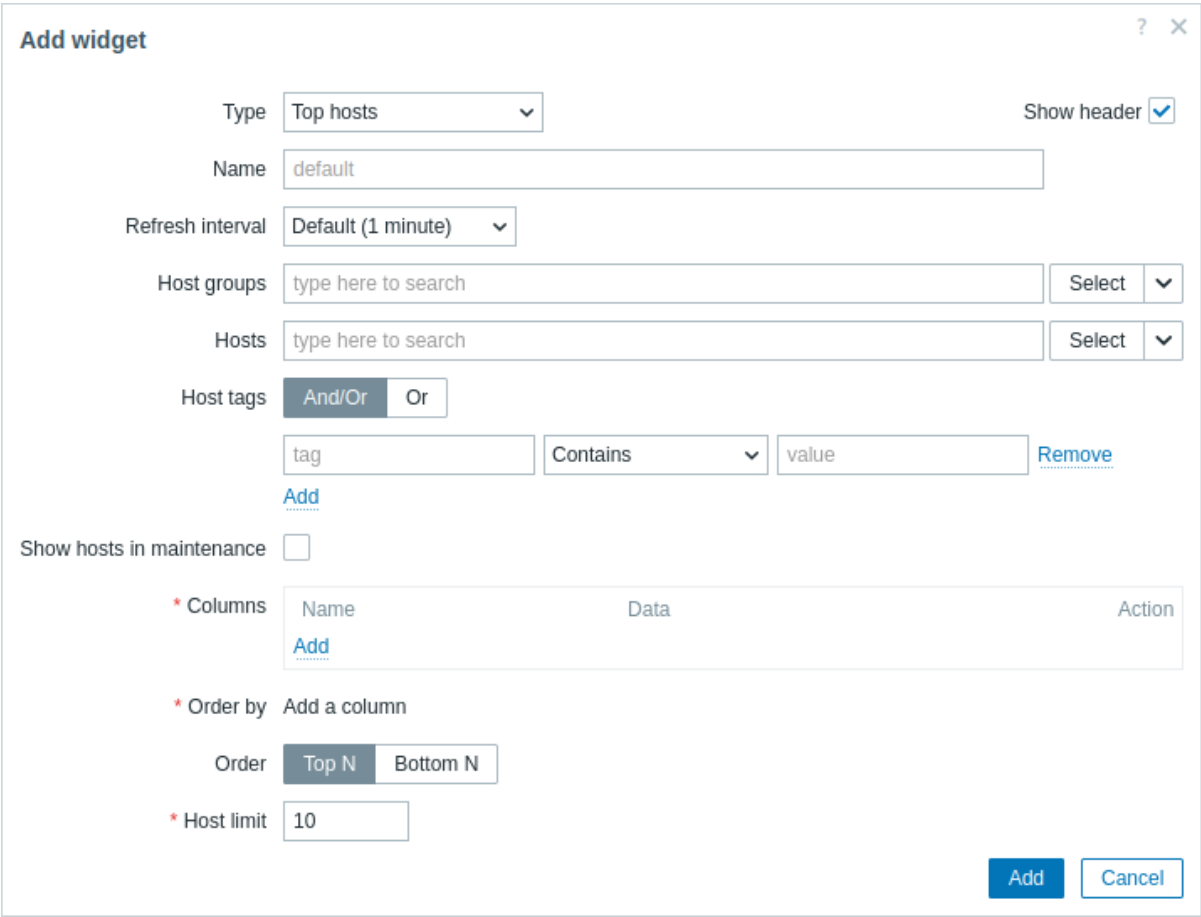


描画されたグラフは.png 画像としてエクスポートできます。

従来のグラフ（クラシック）ウィジェットも利用可能です。

設定

設定するにはタイプとしてグラフを選択します。



データセット

データセットタブでは、データセットを追加してグラフのデータを選択できます。2 種類のデータセットを追加できます:

- アイテムパターン - 一致するアイテムのデータが表示されます。単一の基本色を選択するか、パレットの行を選択して一致する各アイテムに個別の色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータが表示されます。各アイテムの色をピッカーから個別に選択できます。

デフォルトでは、アイテムパターンのデータセットが追加されます。

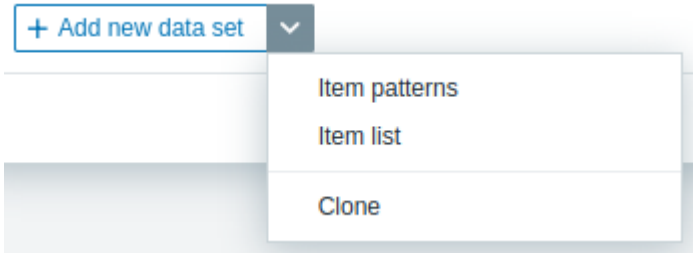
データセット		アイテムパターンデータセットの場合: ホストとアイテムのパターンを選択または入力します。これらのパターンに一致するアイテムのデータがグラフに表示されます。最大 50 個のアイテムが表示されます。 ワイルドカードパターンを使用して選択できます（例: * は 0 文字以上に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。 ワイルドカード記号は常に解釈されるため、他に一致するアイテム（例: *item2*、*item3*）がある場合、*item** という名前のアイテムを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストとアイテムのパターンの指定が必須です。 詳細はデータセット設定の詳細も参照してください。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックしてグラフのアイテムを選択します。 また、ウィジェットを追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択することもできます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットの指定が必須です。 詳細はデータセット設定の詳細も参照してください。 数値型アイテムのみが許可されていることに注意してください。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは利用できず、アイテムリストを指定するパラメータではテンプレートで設定されたアイテムのみ選択できます。
	描画	メトリックの描画タイプを選択します。 選択可能な描画タイプ: 線（デフォルト）、ポイント、ステアケース、バー。 線/ステアケースグラフでデータポイントが 1 つだけの場合、描画タイプに関係なくポイントとして描画されます。ポイントサイズは線の幅から計算されますが、線の幅が小さくても 3 ピクセル未満にはなりません。
	積み上げ	チェックボックスをオンにすると、データが積み上げ（塗りつぶし領域表示）で表示されます。 このオプションはポイント描画タイプを選択した場合は無効です。
	幅	線の幅を設定します。 このオプションは線またはステアケース描画タイプを選択した場合に利用できます。
	ポイントサイズ	ポイントサイズを設定します。 このオプションはポイント描画タイプを選択した場合に利用できます。
	透明度	透明度レベルを設定します。
	塗りつぶし	塗りつぶしレベルを設定します。 このオプションは線またはステアケース描画タイプを選択した場合に利用できます。
	欠損データ	欠損データの表示方法を選択します: なし - ギャップは空白のまま; 接続 - 2 つの境界値を接続; **0** として扱う - 欠損データを 0 値として表示; 最後の値 - 欠損データを最後の既知の値と同じ値で表示; ポイントおよびバー描画タイプには適用されません。
	ホストの上書き	ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
	Y 軸	Y 軸を表示するグラフの側を選択します。
	時間シフト	必要に応じて時間シフトを指定します。 このフィールドでは時間サフィックスを使用できます。負の値も許可されます。

集計関数	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **sum** - 合計値を表示; **count** - 値の個数を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; **none** - すべての値を表示（集計なし）。 集計を使用すると、すべての値の代わりに選択した間隔（5 分、1 時間、1 日など）の集計値を表示できます。詳細はグラフの集計も参照してください。
集計間隔	値を集計する間隔を指定します。 このフィールドでは時間サフィックスを使用できます。サフィックスなしの数値は秒として扱われます。 ウィジェットがトレンドに基づいて履歴データを表示するように設定されている場合（履歴データの選択がトレンドまたは自動に設定）、1 時間の倍数の集計間隔（例: 3600、60m、1h、3h など）を使用することを推奨します。トレンドは 1 時間ごとに集計された値を保存するため、1 時間の倍数でない集計間隔（例: 100s、7min、15min、90min など）を使用すると、解釈が難しい結果になる場合があります。
集計	集計方法を指定します: 各アイテム - データセット内の各アイテムを個別に集計して表示; データセット - データセット内のすべてのアイテムを集計して 1 つの値として表示。
近似値	垂直グラフピクセルごとに複数の値が存在する場合に表示する値を指定します: すべて - 最小値、最大値、平均値を表示; **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示。 この設定は、10 分ごとに収集された 1 年分の値など、更新間隔が頻繁な長期間のグラフを表示する場合に便利です。
データセットラベル	グラフのデータセット設定およびグラフの凡例（集計データセットの場合）に表示されるデータセットラベルを指定します。 すべてのデータセットは、データセットラベルが指定されている場合も含めて番号が付けられます。ラベルが指定されていない場合、データセットは自動的に番号（例:「データセット #2」、「データセット #3」など）でラベル付けされます。データセットの並べ替えやドラッグ後にデータセットの番号付けが再計算されます。 ラベルが長すぎる場合は、表示される場所に合わせて短縮されます（例:「Number of proc...」）。

データセットの設定詳細

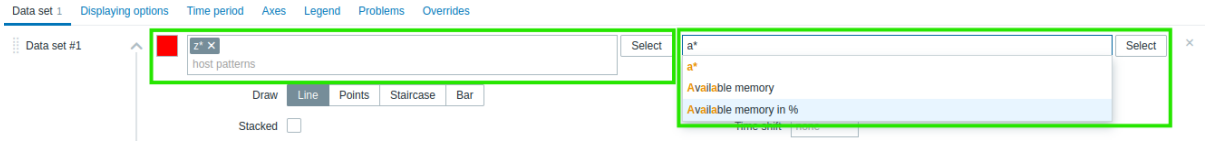
既存のデータセットはリストに表示されます。以下の操作が可能です:

-  移動アイコンをクリックし、データセットをリスト内の新しい場所にドラッグします。
- 展開アイコンをクリックして、データセットの詳細を展開します。展開時、このアイコンは 折りたたみアイコンに変わります。
- カラーアイコンをクリックしてカラーピッカーを開きます。16 進数コードを入力するか、単色スウォッチを選択するか、パレットタブに切り替えて定義済みの色の行を選択できます。選択した色は、アイテムリストデータセットには直接適用され、アイテムパターンでは生成されたシェードのベースとして適用されます。Tab でダイアログコントロール間を移動し、矢印キーでスウォッチやパレットの行を移動し、Enter で選択、Esc でキャンセルします。
- 新しいデータセットを追加ボタンをクリックして、ホストやアイテムパターンを選択できる空のデータセットを追加します。新しいデータセットを追加ボタンの横にある下向きアイコンをクリックすると、ドロップダウンメニューが表示され、新しいアイテムパターンまたはアイテムリストデータセットを追加したり、現在開いているデータセットを複製したりできます。すべてのデータセットが折りたたまれている場合、複製オプションは利用できません。

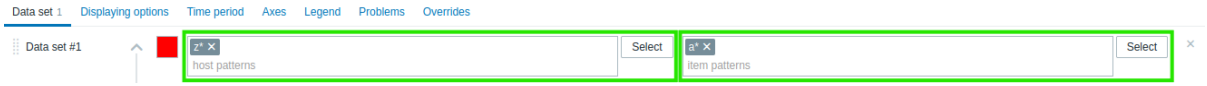


アイテムパターンデータセットには、ホストパターンとアイテムパターンのフィールドがあり、どちらもワイルドカード記号（*）を含む完全な名前またはパターンを認識します。この機能により、選択したパターンを含むすべてのホスト名とアイテム名を選択できます。アイテムパターンフィールドにアイテム名またはアイテムパターンを入力すると、選択したホスト名に属するアイテムのみがドロップダウンリストに表示されます。

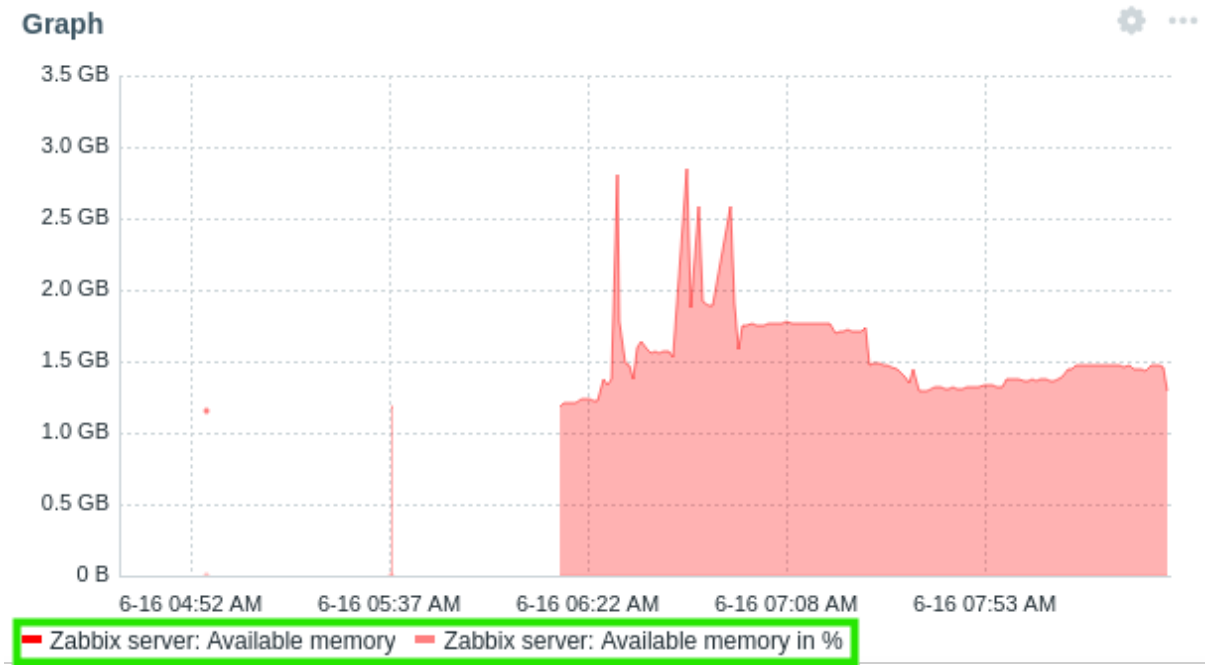
たとえば、ホストパターンフィールドにパターン **z*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのホスト名が表示されます：**z*, Zabbix server, Zabbix proxy**。*Enter* を押すと、このパターンが受け入れられ、**z*** として表示されます。同様に、アイテムパターンフィールドにパターン **a*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのアイテム名が表示されます：**a*, Available memory, Available memory in %**。



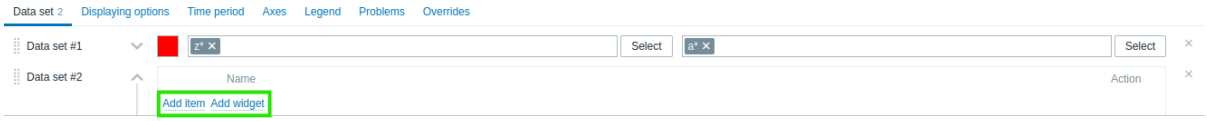
Enter を押すと、パターンが受け入れられ、**a*** として表示されます。



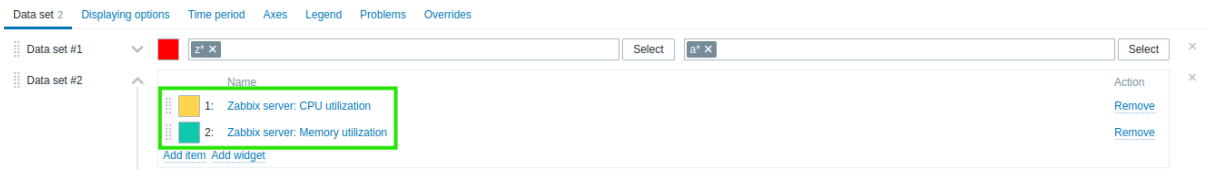
その後、グラフには選択したホスト名に属するすべてのアイテムが表示されます。



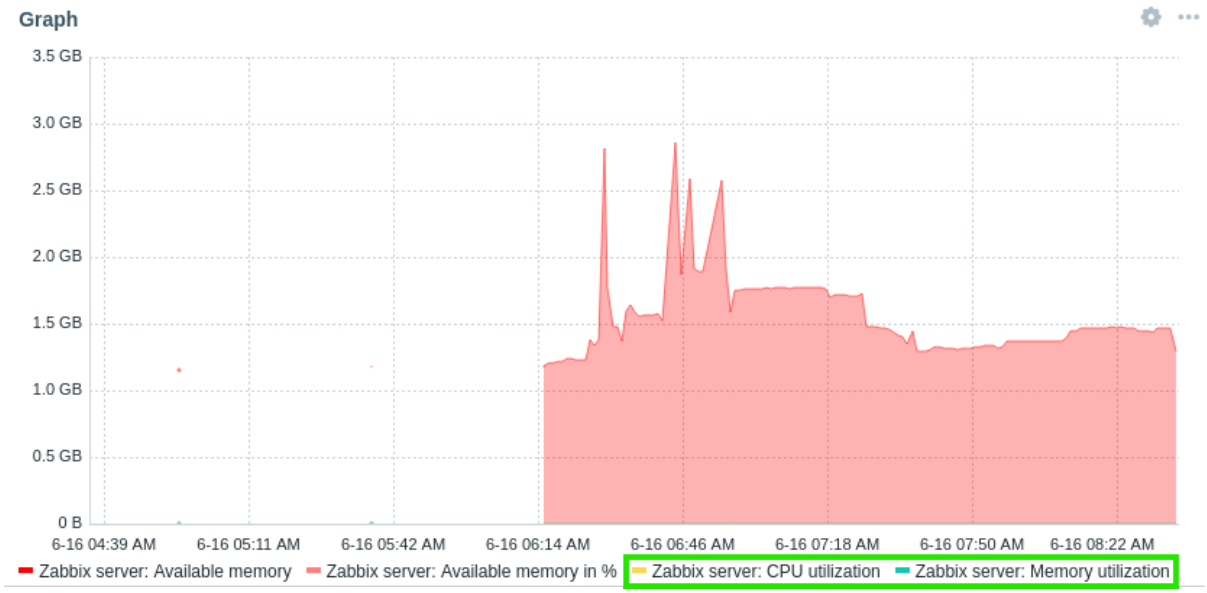
アイテムリストデータセットには、グラフに表示するアイテムを追加できるアイテムの追加ボタンがあります。また、ウィジェットの追加ボタンをクリックすることで、アイテムのデータソースとして互換性のあるウィジェットを追加することもできます。



例えば、アイテムの追加ボタンをクリックすると、ホストパラメータを含むポップアップウィンドウが開きます。ホストを選択すると、そのホストの選択可能なすべてのアイテムがリストに表示されます。

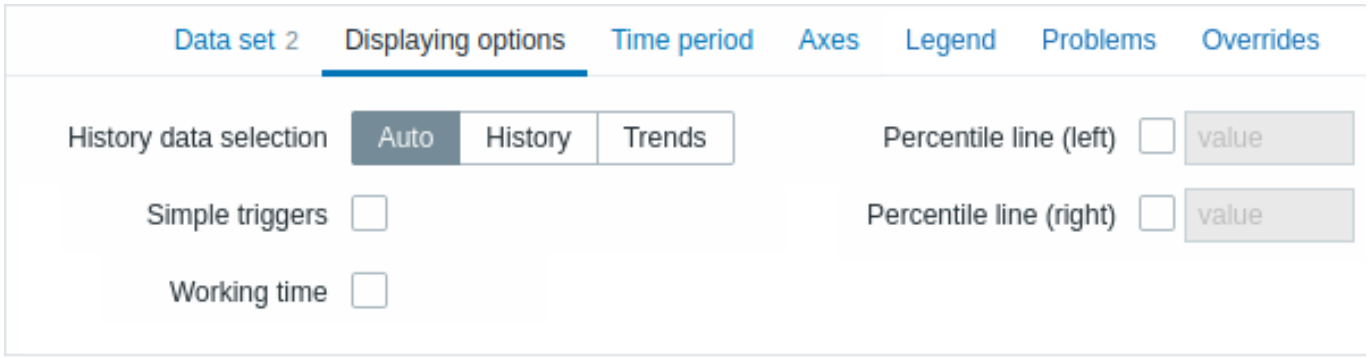


1 つまたは複数のアイテムを選択すると、それらがデータセットのアイテムリストとグラフに表示されます。



表示オプション

表示オプションタブでは、履歴データの選択を定義できます。

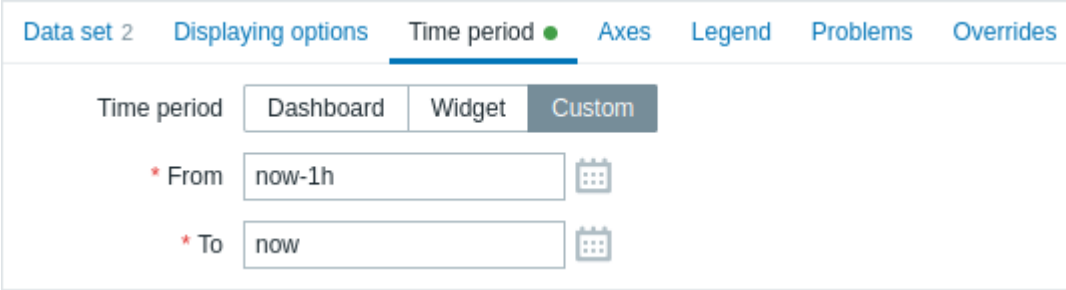


履歴データの選択	グラフデータのソースを設定します: 自動 - クラシックグラフのアルゴリズムに従ってデータが取得されます（デフォルト）; 履歴 - 履歴からデータを取得; トレンド - トレンドからデータを取得。
シンプルトリガー	チェックボックスをオンにすると、シンプルトリガーのしきい値が表示されます。しきい値は、トリガーの深刻度の色で点線として描画されます。 シンプルトリガーとは、1 つのアイテムに対して 1 つの関数（`last`、`max`、`min`、`avg` のみ）を使用したトリガーです。 最大 3 つのトリガーを描画できます。トリガーが描画範囲内にある場合のみ表示されることに注意してください。
稼働時間	チェックボックスをオンにすると、グラフ上に稼働時間が表示されます。稼働時間（稼働日）はグラフ上で白色の背景で表示され、非稼働時間はグレー（デフォルトの *Original blue* フロントエンドテーマの場合）で表示されます。
パーセンタイルライン（左）	チェックボックスをオンにし、パーセンタイル値を入力すると、指定したパーセンタイルがグラフの左 Y 軸にラインとして表示されます。 例えば、95% パーセンタイルを設定した場合、パーセンタイルラインは値の 95% がその下に収まるレベルに表示されます。

パーセンタイルライン（右）	チェックボックスをオンにし、パーセンタイル値を入力すると、指定したパーセンタイルがグラフの右 Y 軸にラインとして表示されます。 例えば、95% パーセンタイルを設定した場合、パーセンタイルラインは値の 95% がその下に収まるレベルに表示されます。

期間

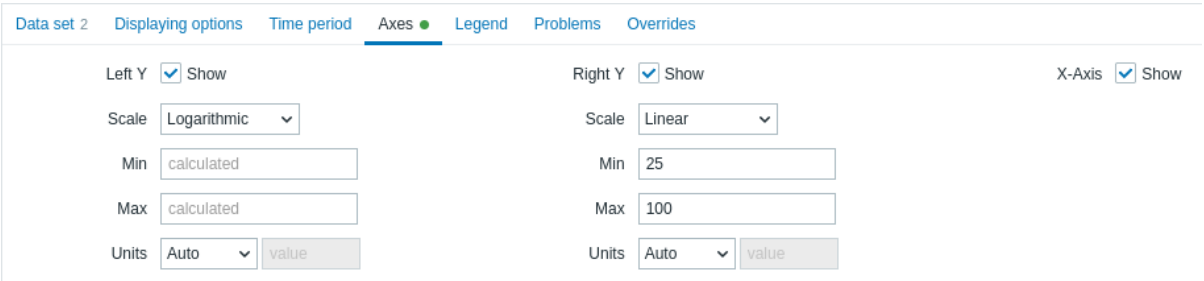
期間タブでは、グラフにデータを表示する期間を設定できます。



期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用します; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメーターで設定）を使用します; カスタム - *From* および *To* パラメーターで設定したカスタム期間を使用します。設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できることに注意してください。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメーターは、期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。

軸

軸タブでは、軸の表示方法をカスタマイズできます。

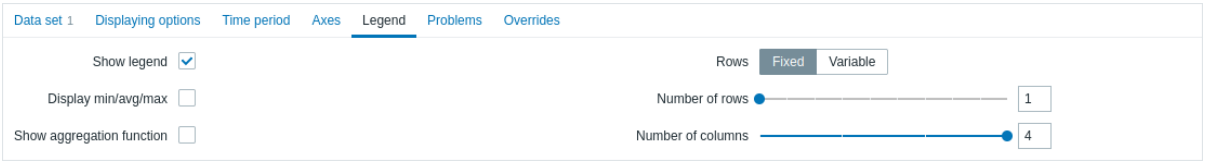


左 Y	左 Y 軸を表示するには、このチェックボックスをオンにします。 このチェックボックスは、データセットまたはオーバーライドタブで選択解除されている場合は無効になります。
右 Y	右 Y 軸を表示するには、このチェックボックスをオンにします。 このチェックボックスは、データセットまたはオーバーライドタブで選択解除されている場合は無効になります。
X 軸	このチェックボックスをオフにすると、X 軸が非表示になります（デフォルトでオン）。
スケール	グラフ軸の値のスケールをドロップダウンから選択します： リニア - 軸の値が一定の量で増加します（例：10、20、30）。データが安定して変化する場合や、範囲が小～中程度の場合に適しています。 対数 - 軸の値が指数関数的に増加します（例：10、100、1000）。データが急激に変化する場合や、範囲が大きい場合に適しています。

最小	対応する軸の最小値を設定します。 Y 軸の表示範囲の最小値を指定します。
最大	対応する軸の最大値を設定します。 Y 軸の表示範囲の最大値を指定します。
単位	グラフ軸の値の単位をドロップダウンから選択します： 自動 - 軸の値はデータセット内の最初のアイテムの単位で表示されます。 静的 - 軸の値は値入力フィールドで指定した単位で表示されます。フィールドが空白の場合は、数値のみが表示されます。

凡例

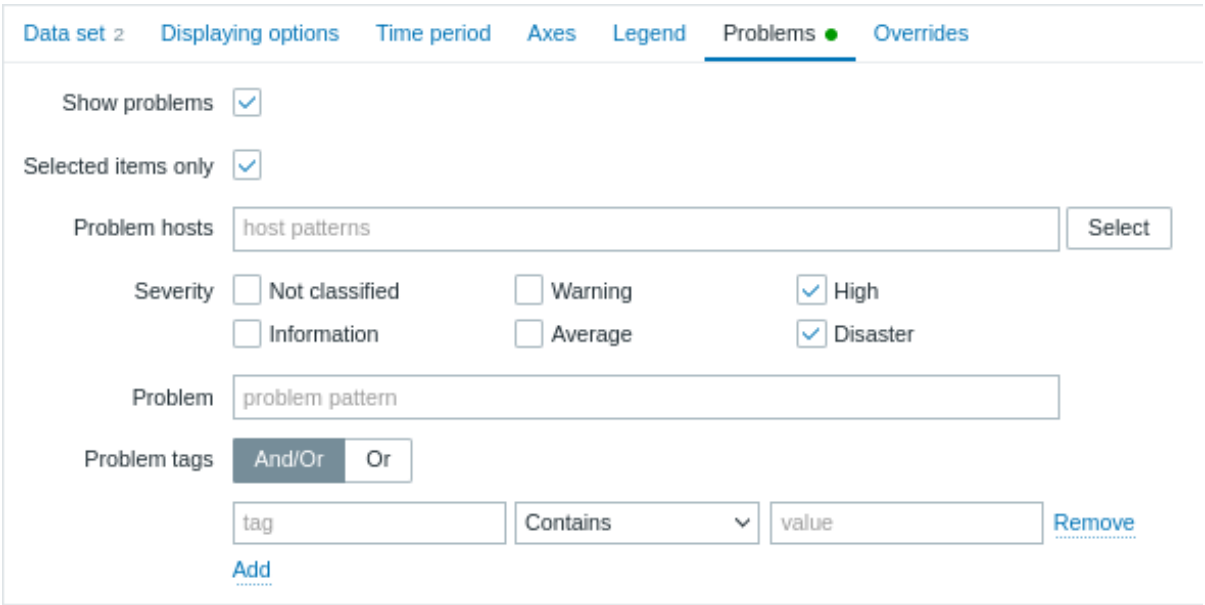
凡例タブでは、グラフの凡例をカスタマイズできます。



凡例を表示	このチェックボックスのチェックを外すと、グラフの凡例が非表示になります（デフォルトでチェックされています）。
最小/平均/最大値を表示	このチェックボックスをオンにすると、凡例にアイテムの最小値、平均値、最大値が表示されます。
集約関数を表示	このチェックボックスをオンにすると、凡例に集約関数が表示されます。
行数	凡例の行の表示モードを選択します： 固定 - 表示される行数は行数パラメータの値で決まります。 可変 - 表示される行数は設定されたアイテム数によって決まり、最大行数パラメータの値を超えません。
行数/ 最大行数	行数が「固定」の場合、表示する凡例の行数を設定します（1～10）。 行数が「可変」の場合、表示する凡例の最大行数を設定します（1～10）。
列数	表示する凡例の列数を設定します（1～4）。 このパラメータは、最小/平均/最大値を表示がオフの場合に利用できます。

問題

問題タブでは、問題の表示をカスタマイズできます。

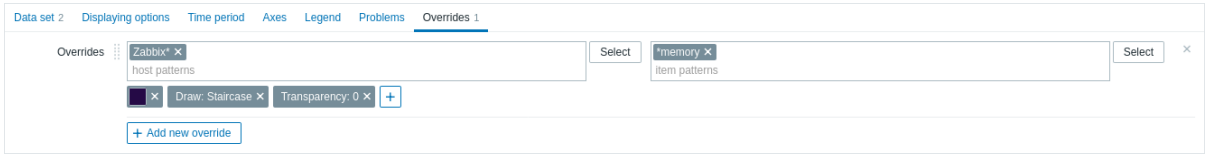


問題を表示	このチェックボックスをオンにすると、グラフ上に問題が表示されます（デフォルトではオフ）。
選択したアイテムのみ	このチェックボックスをオンにすると、選択したアイテムの問題のみがグラフに表示されます。

問題のホスト	グラフに表示する問題のホストを選択します。 ワイルドカードパターンを使用できます（例：* は 0 文字以上に一致します）。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。 入力中に、すべての一致するホストがドロップダウンに表示されることに注意してください。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
深刻度	問題の深刻度を選択して、グラフに表示する問題をフィルタリングします。 深刻度が選択されていない場合は、すべての問題が表示されます。
問題	グラフに表示する問題名を指定します。
問題タグ	ウィジェットに表示される問題の数を制限するために問題タグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります： 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります： **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

オーバーライド

オーバーライドタブでは、データセットに対してカスタムオーバーライドを追加できます。



オーバーライドは、* ワイルドカードを使用してデータセットに複数のアイテムを選択し、デフォルトでアイテムの表示方法 (例: デフォルトのベースカラーやその他のプロパティ) を変更したい場合に便利です。

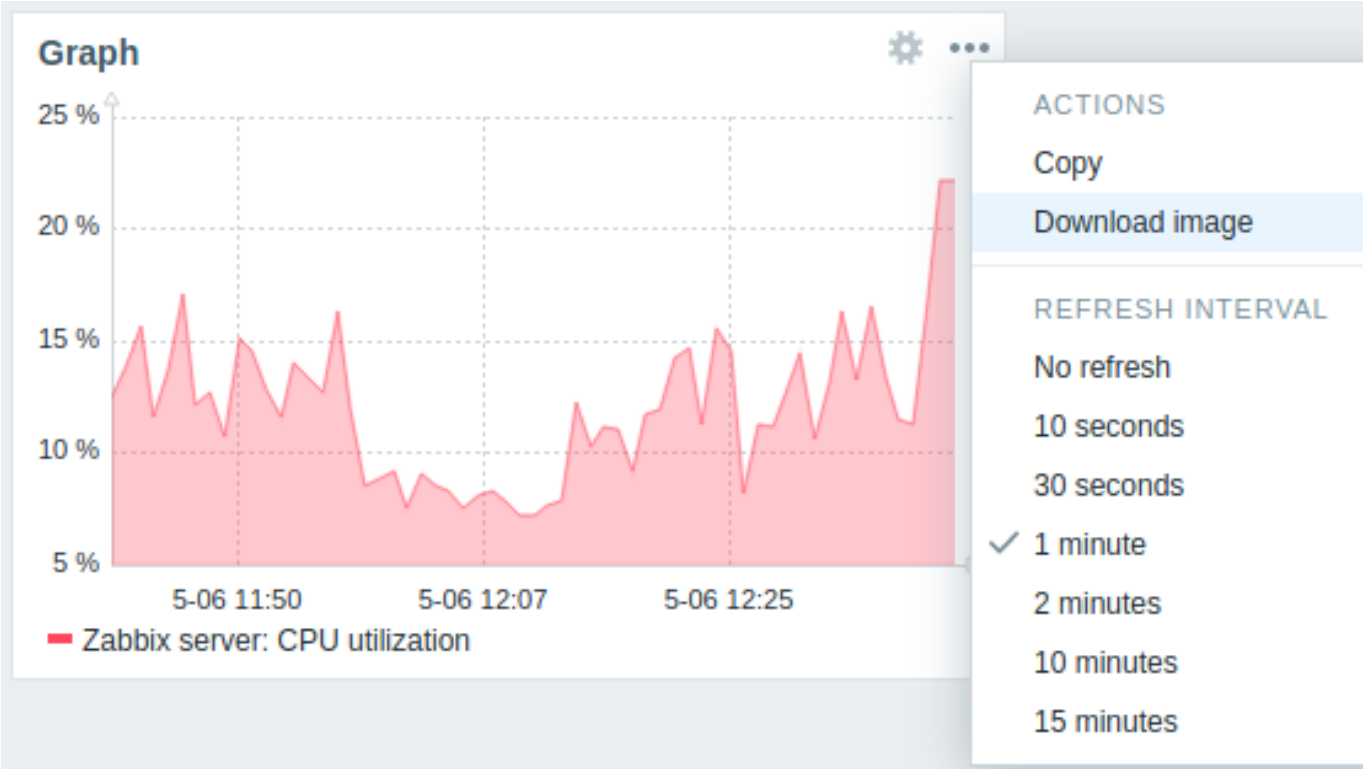
既存のオーバーライド (ある場合) はリストで表示されます。新しいオーバーライドを追加するには:

- 

ボタンをクリックします
- オーバーライドするホストとアイテムを選択します。あるいは、ホストとアイテムのパターンを入力することもできます。ワイルドカードパターンを使用できます (例: * は 0 文字以上に一致する結果を返します)。ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力中に、すべての一致するホストがドロップダウンに表示されることに注意してください。ワイルドカード記号は常に解釈されるため、他に一致するアイテム (例: item2, item3) がある場合、”item*” という名前のアイテムを個別に追加することはできません。ホストパターンとアイテムパターンのパラメータは必須です。テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは利用できません。テンプレートダッシュボードでウィジェットを設定する場合、アイテムリストを指定するパラメータではテンプレートで設定されたアイテムのみ選択できます。
- 

をクリックして、オーバーライドパラメータを選択します。少なくとも 1 つのオーバーライドパラメータを選択する必要があります。パラメータの説明については、上記のデータセットタブを参照してください。

グラフウィジェットで表示される情報は、ウィジェットメニューを使用して.png 画像としてダウンロードできます。

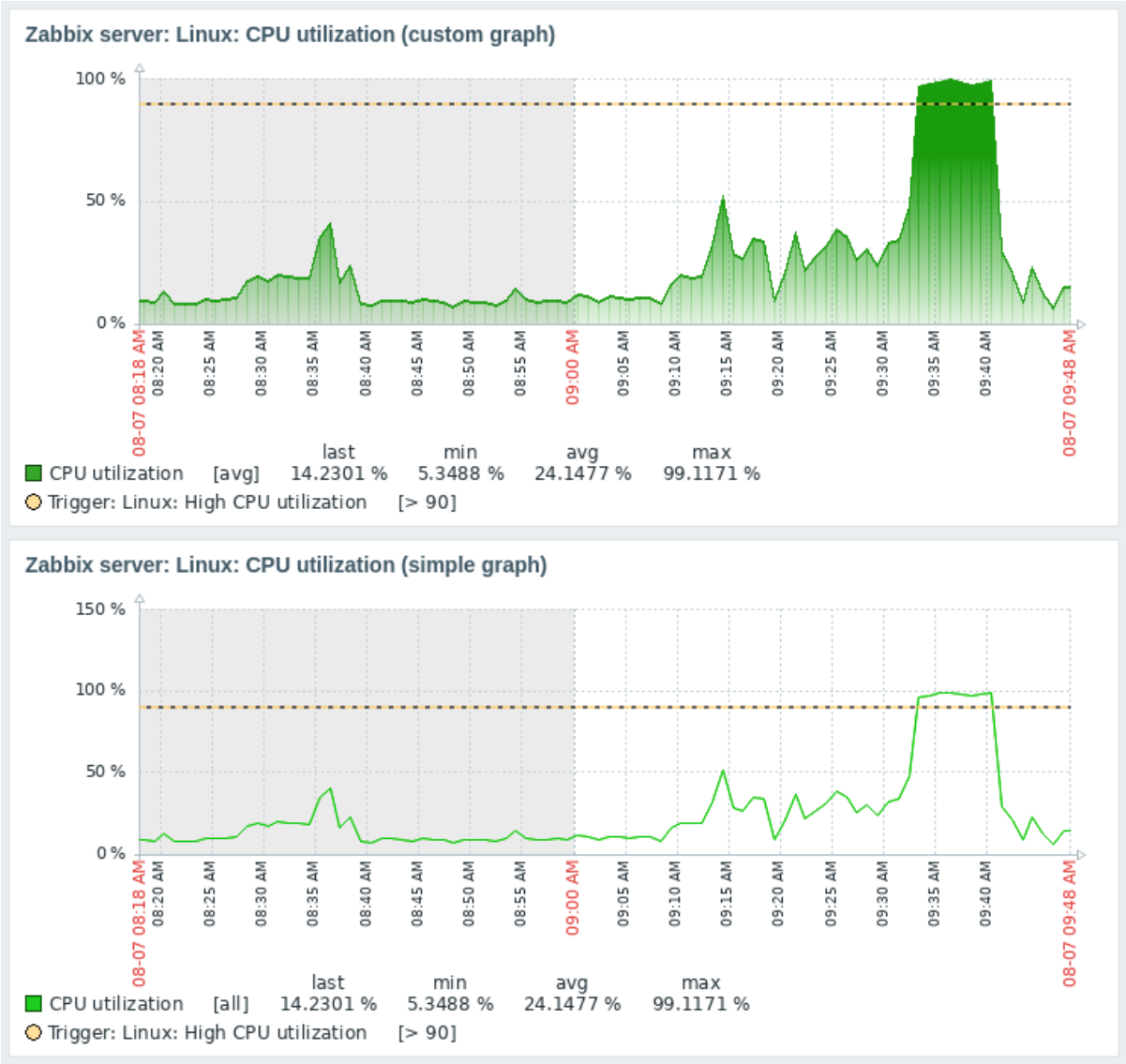


ウィジェットのスクリーンショットはダウンロードフォルダに保存されます。

9 グラフ（クラシック）

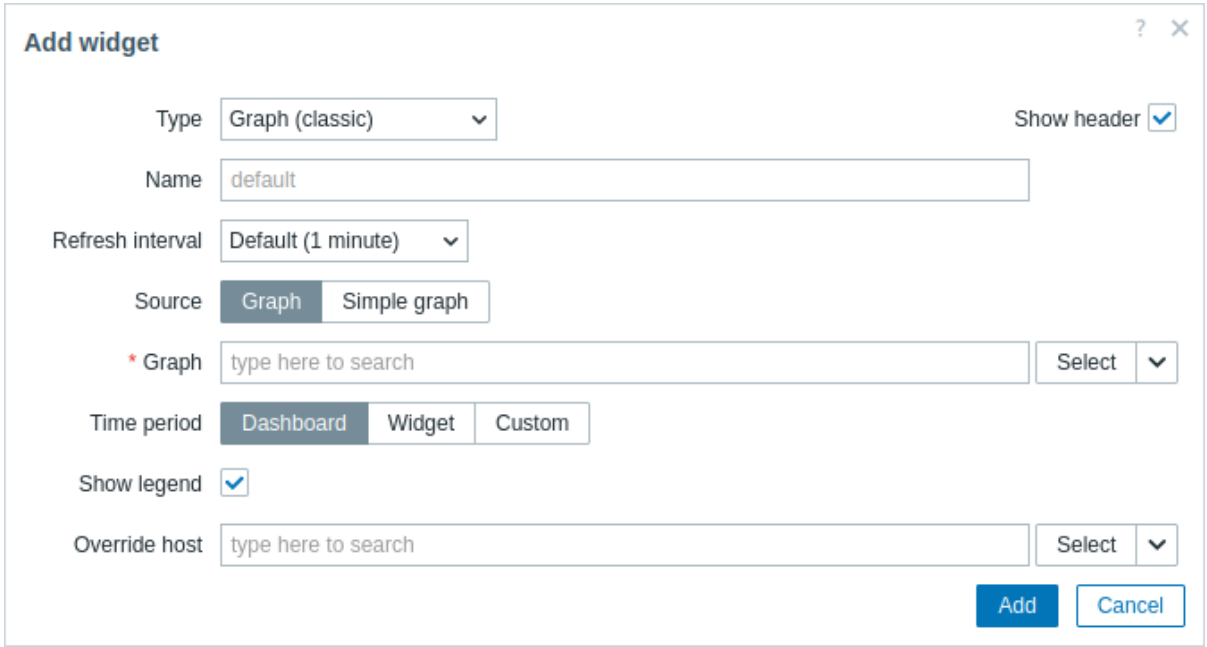
概要

グラフ（クラシック）ウィジェットは、数値アイテムデータを画像ベースのカスタムグラフまたはシンプルグラフとして表示します。
期間選択、トリガーライン、オプションの凡例表示をサポートしています。



設定

設定するには、タイプとして グラフ（クラシック）を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ソース	グラフタイプを選択します： グラフ - カスタムグラフ; シンプルグラフ - シンプルグラフ。
グラフ	表示するカスタムグラフを選択します。 または、グラフのデータソースとして互換性のあるウィジェットを選択します。 このパラメータは、ソースが「グラフ」に設定されている場合に利用できます。
アイテム	シンプルグラフで表示するアイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 このパラメータは、ソースが「シンプルグラフ」に設定されている場合に利用できます。
期間	グラフにデータを表示する期間を設定します。期間のデータソースを選択します： ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
凡例を表示	このチェックボックスのチェックを外すと、グラフの凡例が非表示になります（デフォルトでチェックされています）。
ホストの上書き	互換性のあるウィジェットまたはダッシュボードのホストセレクタを、ホストのデータソースとして選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

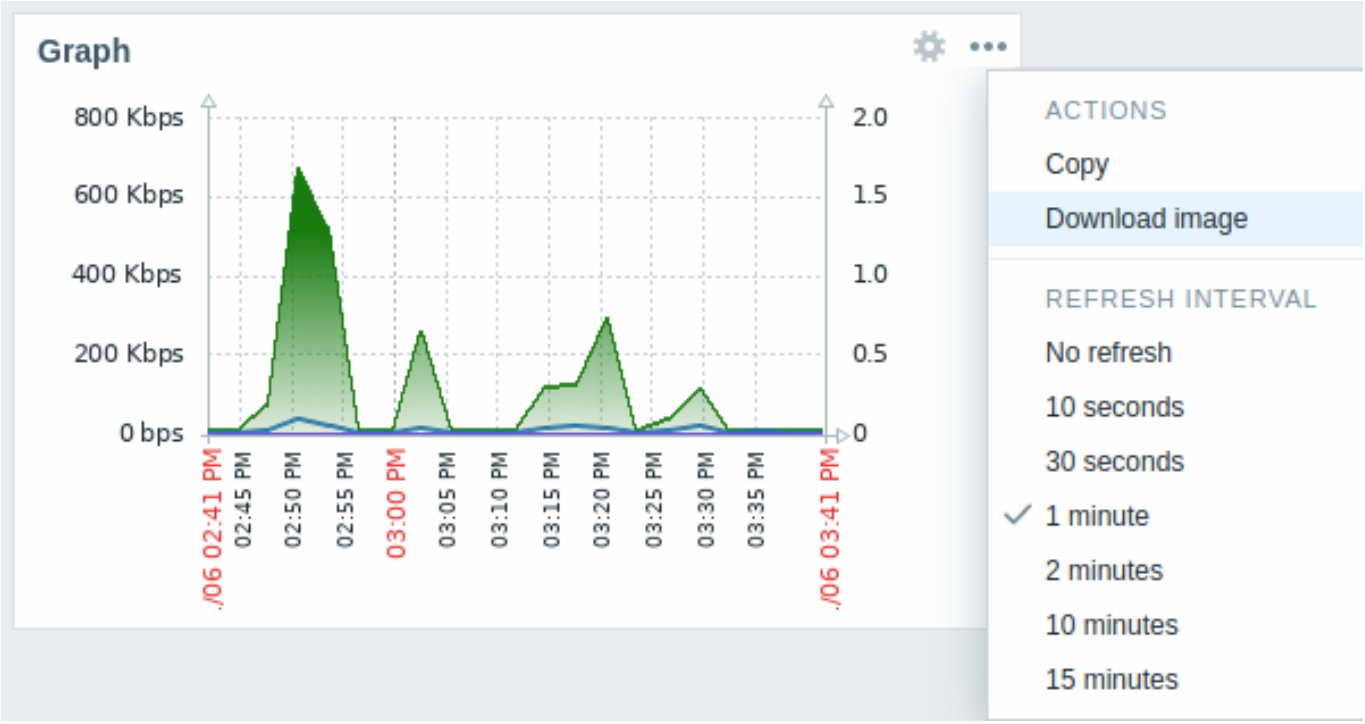
グラフの凡例は 3 つのセクションで構成されています。

- アイテムとその集計値
- パーセンタイル（設定されている場合）
- トリガー（表示されているアイテムに関連付けられている場合）

ウィジェット内のグラフの高さが不十分な場合、凡例が表示されないか、部分的にしか表示されない場合があります。トリガーとパーセンタイルが最初に非表示になり、次にアイテムの凡例が非表示になります。完全な凡例を表示するには、ウィジェットの垂直サイズを大きくしてください。

Note:
トリガーラインは最大 3 本まで表示できます。トリガーがそれ以上ある場合は、重大度の低いトリガーが優先して表示されます。

クラシックグラフウィジェットで表示される情報は、ウィジェットメニューから*.png* 画像としてダウンロードできます。



ウィジェットのスクリーンショットは *Downloads* フォルダに保存されます。

10 グラフのプロトタイプ

概要

グラフプロトタイプウィジェットは、ローレベルディスカバリールールのグラフプロトタイプまたはアイテムプロトタイプに基づいて自動的に作成されたカスタムグラフのグリッドを表示します。

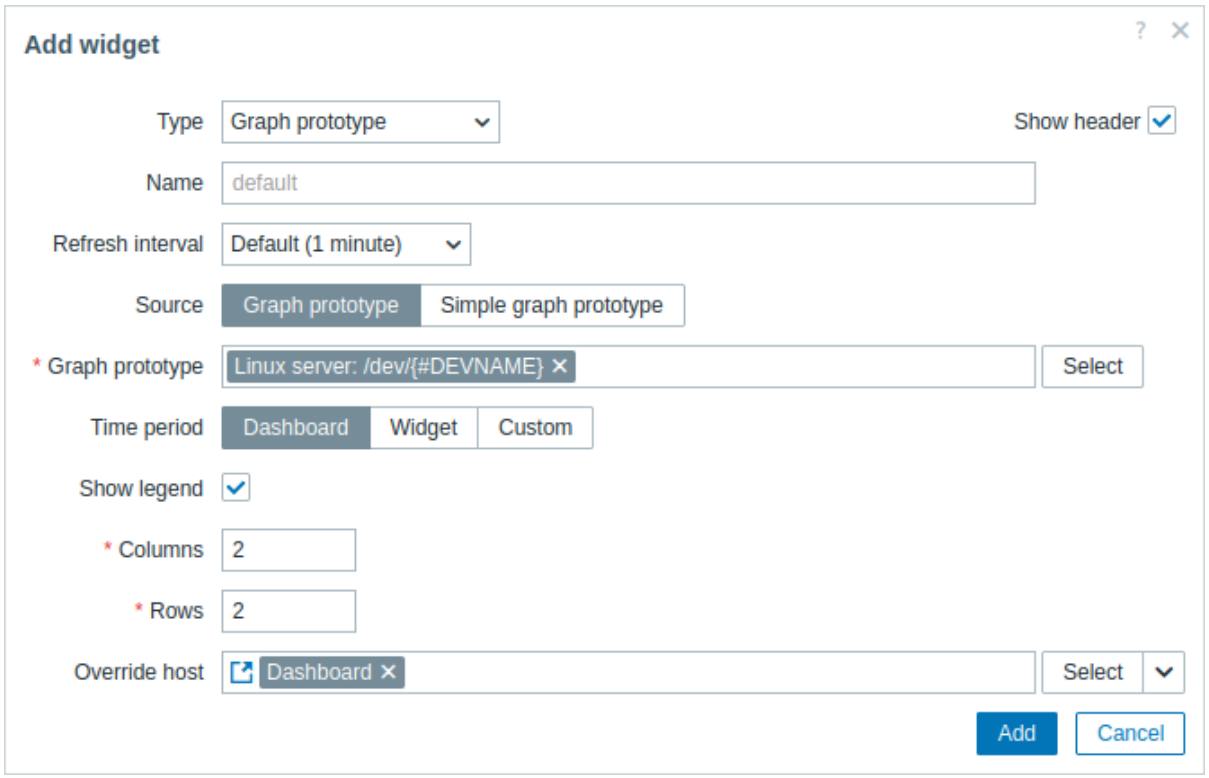


グラフは設定可能なグリッドレイアウトで配置されます。グラフの数がレイアウトを超える場合、ウィジェットでページングが有効

になります。

設定

設定するには、ウィジェットタイプとしてグラフプロトタイプを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ソース	グラフのソースを選択します: グラフプロトタイプまたはシンプルグラフプロトタイプ。
グラフプロトタイプ	グラフプロトタイプで検出されたグラフを表示するグラフプロトタイプを選択します。 このパラメータはソースが「グラフプロトタイプ」に設定されている場合に利用できます。
アイテムプロトタイプ	アイテムプロトタイプで検出されたアイテムのシンプルグラフを表示するアイテムプロトタイプを選択します。 このパラメータはソースが「シンプルグラフプロトタイプ」に設定されている場合に利用できます。
期間	グラフにデータを表示する期間を設定します。期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
凡例を表示	このチェックボックスのチェックを外すと、グラフの凡例が非表示になります（デフォルトでチェックされています）。
ホストを上書き	互換性のあるウィジェットまたはダッシュボードのホストセレクタを、ホストのデータソースとして選択します。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
列	グラフプロトタイプウィジェット内に表示するグラフの列数を入力します。
行	グラフプロトタイプウィジェット内に表示するグラフの行数を入力します。

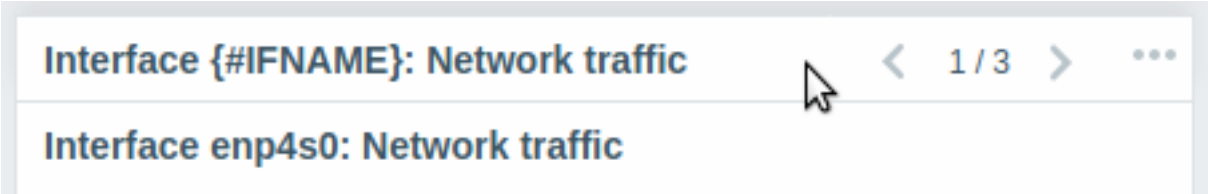
グラフの凡例は 3 つのセクションで構成されています。

- アイテムとその集計値
- パーセンタイル（設定されている場合）
- トリガー（表示されているアイテムに関連付けられている場合）

ウィジェット内のグラフの高さが不十分な場合、凡例が表示されないか、部分的にしか表示されない場合があります。トリガーとパーセンタイルが最初に非表示になり、次にアイテムの凡例が非表示になります。完全な凡例を表示するには、ウィジェットの縦サイズを大きくしてください。

Note:
トリガーラインは最大 3 本まで表示できます。トリガーがそれ以上ある場合は、重大度の低いトリガーが優先的に表示されます。

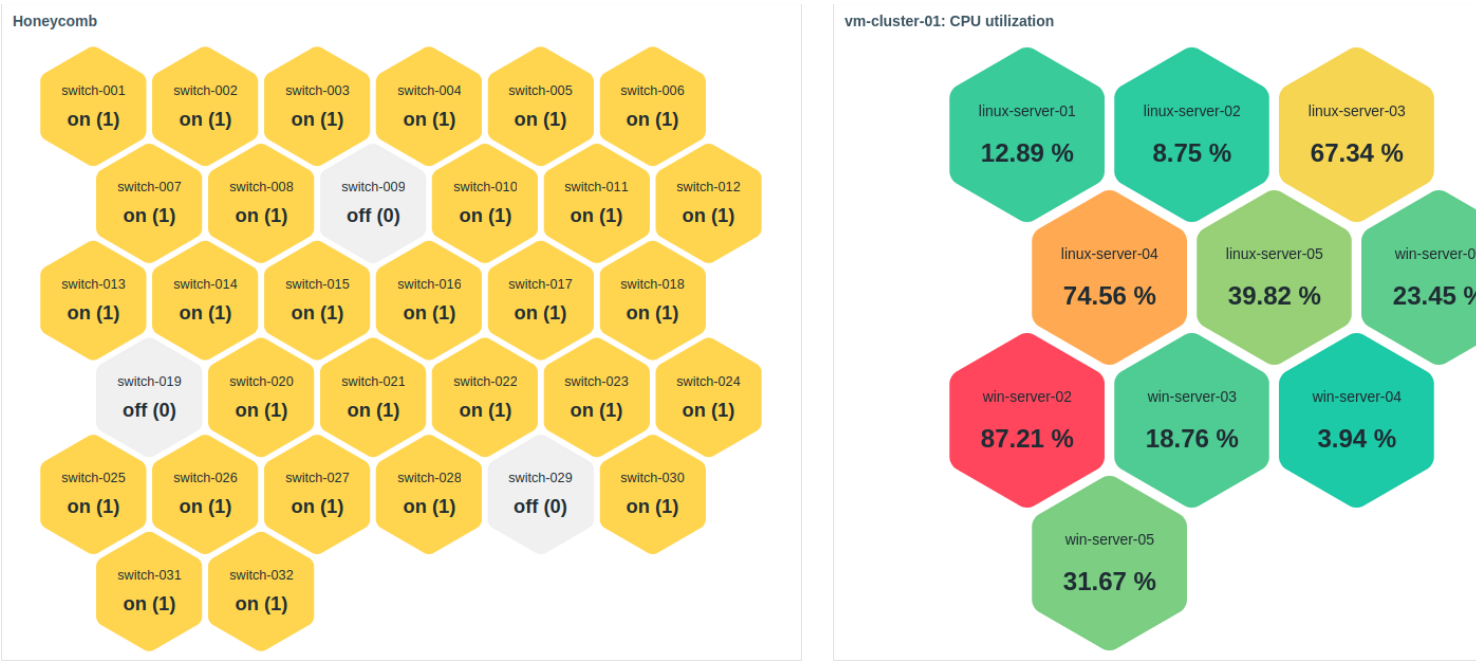
列および行パラメータでウィジェット内に複数のグラフを収めることができますが、ウィジェット内の列/行数よりも多くのグラフが検出される場合があります。この場合、ウィジェットでページングが利用可能になり、スライドアップヘッダーで左右の矢印を使ってページを切り替えることができます。



11 ハニカム

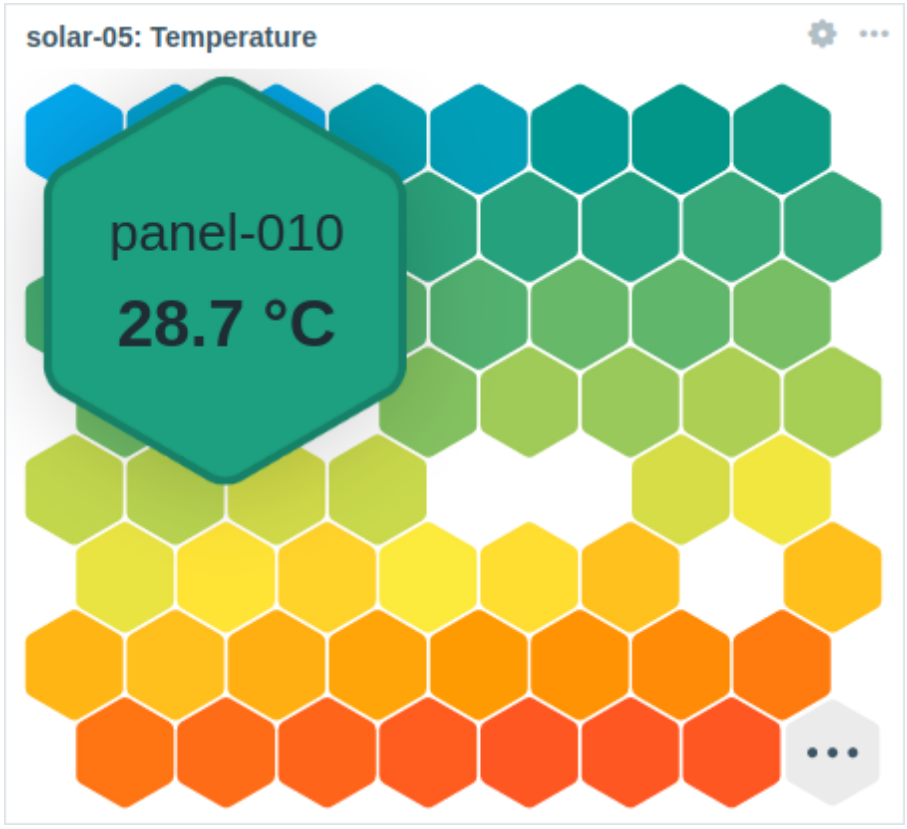
概要

Honeycomb ウィジェットは、ホストまたはアイテムを六角形のセルとしてグリッドに表示します。このレイアウトにより、問題のあるホスト/アイテムの検出、問題のクラスターの特定、グループの比較などが容易になります。



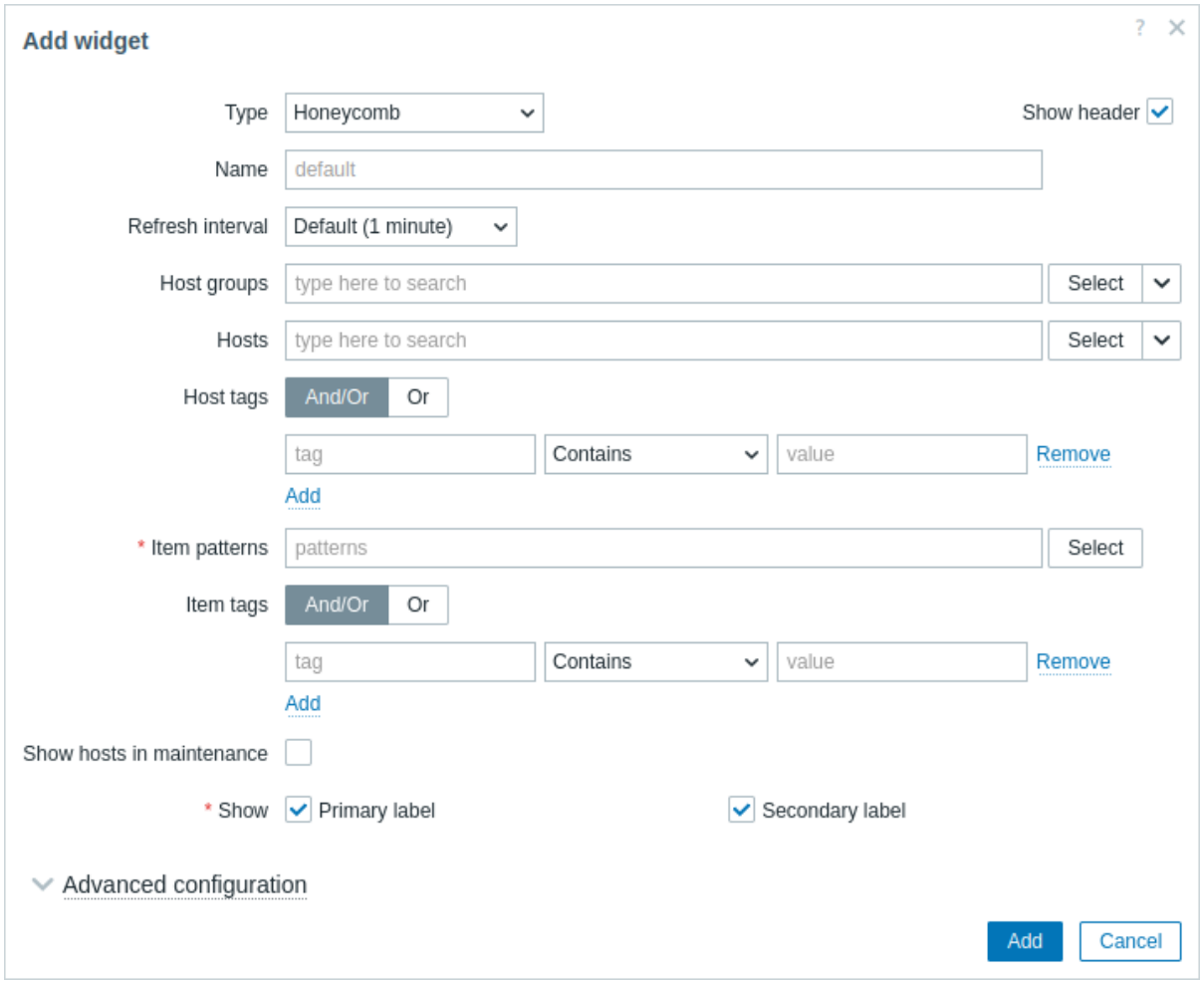
表示されるセルの数は、ウィジェットのサイズと最小セルサイズ（32px）によって制限されます。すべてのセルが収まらない場合、最後のセルに省略記号が表示されます。ウィジェットはリサイズ可能で、セルの数、サイズ、位置は動的に調整されます。

マウスオーバー時には、フォーカスされたセルが拡大されて視認性が向上します。セルをクリックすると、そのセルの枠線がハイライトされ、別のセルが選択されるまで維持されます。



設定

タイプとして *Honeycomb* を選択します:

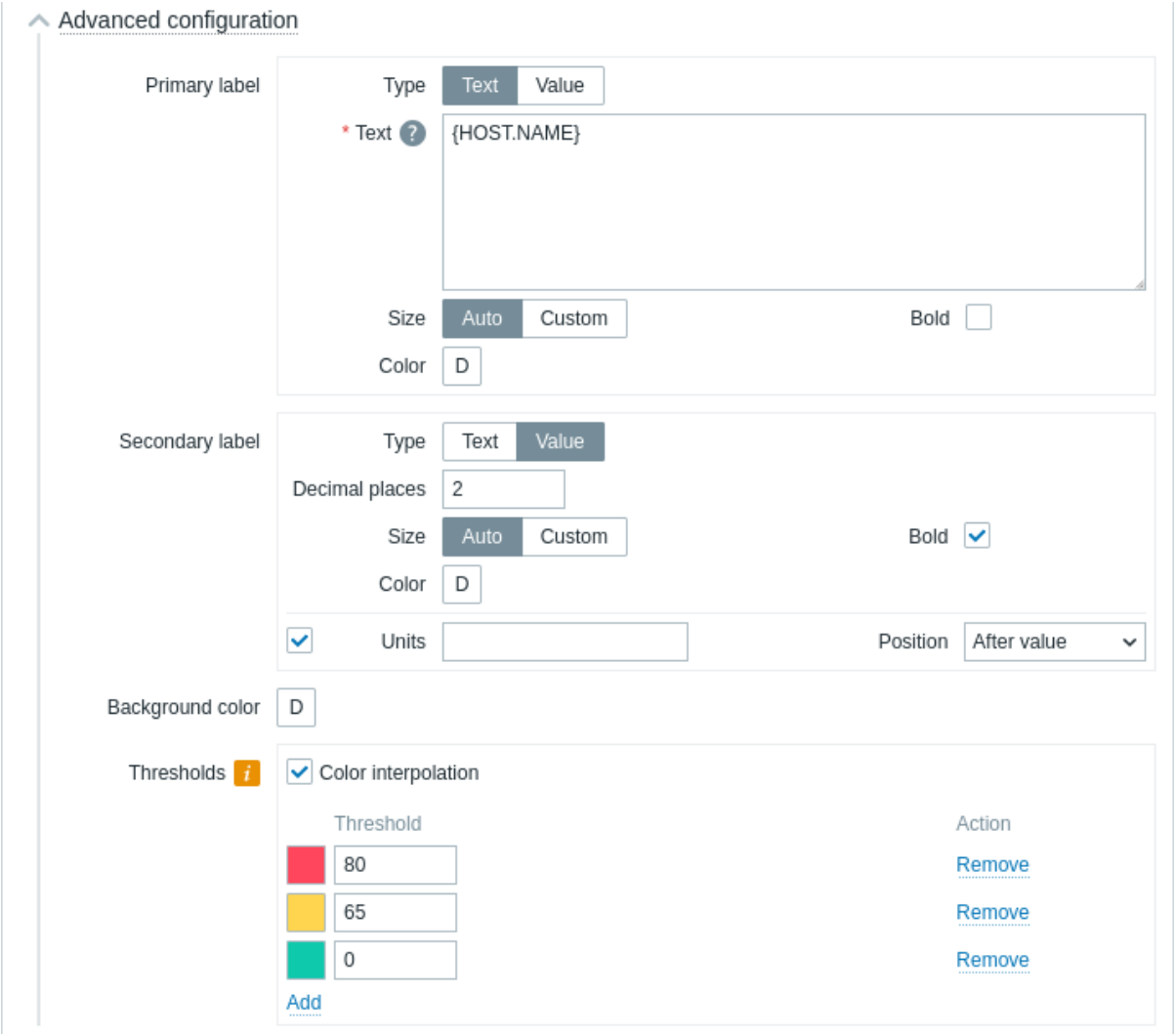


すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、すべてのネストされたホストグループが暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットは選択したアイテムパターン（下記）に一致するアイテムを持つホストを含むすべてのホストグループを表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットは選択したアイテムパターン（下記）に一致するアイテムを持つすべてのホストを表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストタグ	ウィジェットに表示するホスト数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件の計算タイプは 2 つあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
アイテムパターン	アイテムパターンを入力するか、既存のアイテムをアイテムパターンとして選択します。入力または選択したパターンに一致するアイテムのデータがハニカムに表示されます。アイテムパターンパラメータは必須です。 ワイルドカードパターンを選択に使用できます（例: `*` は 0 文字以上に一致、`Zabbix*` は”Zabbix”で始まるアイテムに一致）。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。入力を始めると、選択したホストまたは選択したホストグループ内のホストに属するアイテムに限定した一致するアイテムのドロップダウンリストが表示されます。ワイルドカード記号は常に解釈されるため、例えば *item** という名前のアイテムを個別に追加することはできません（他に一致するアイテムがある場合、例: item2, item3）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータではテンプレートで設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示するアイテム数を制限するタグを指定します。詳細は上記ホストタグを参照してください。
メンテナンス中のホストを表示	このチェックボックスをオンにすると、メンテナンス中のホストが表示されます（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはメンテナンス中のデータを表示と表示されます。
表示	このチェックボックスをオンにすると、該当するハニカムセル要素（一次ラベル、二次ラベル）が表示されます。 少なくとも 1 つの要素を選択する必要があります。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用でき、表示フィールド (上記参照) で選択した要素およびハニカムセルの背景色やしきい値にのみ適用されます。

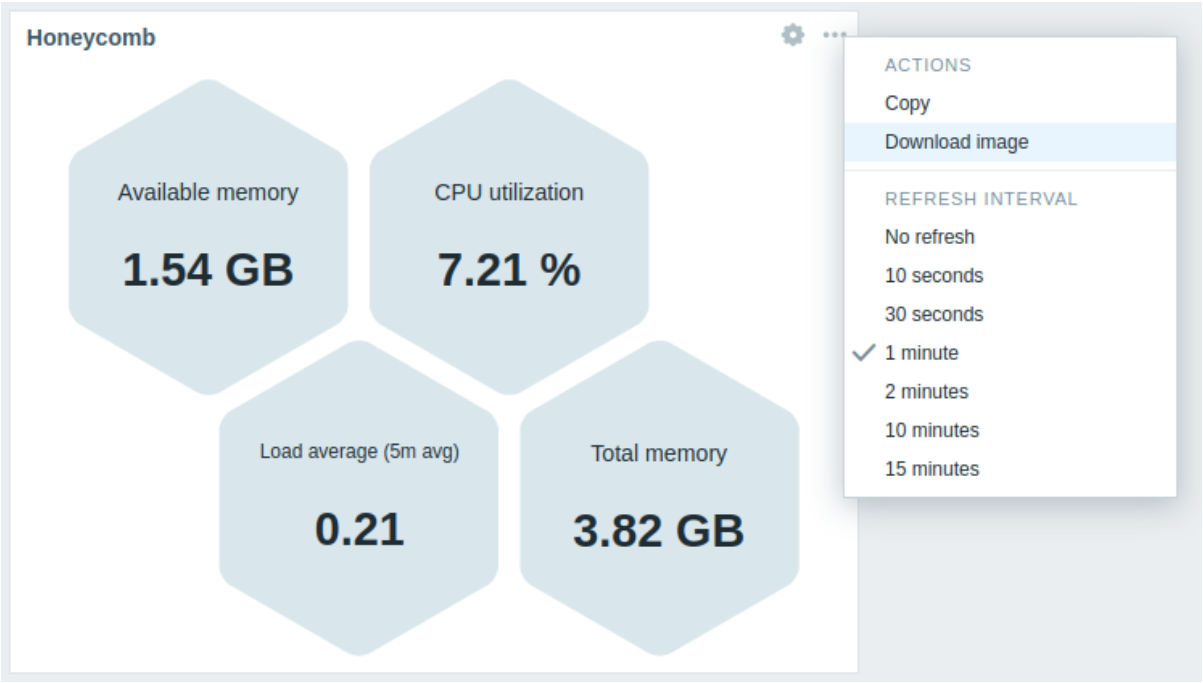


主/副ラベル

タイプ	ラベルのタイプを選択します: テキスト - テキストパラメータで指定したテキストをラベルに表示します; 値 - 小数点以下の桁数パラメータで指定した小数点以下の桁数でアイテム値をラベルに表示します。
テキスト	ラベルのテキストを入力します。このテキストはデフォルトのアイテム名を上書きする場合があります。 複数行のテキストがサポートされています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*}、およびユーザーマクロがサポートされています。 ハニカムセルはホスト名でアルファベット順に、各ホスト内ではアイテム名でアルファベット順に並べられます。 このパラメータはタイプが「テキスト」に設定されている場合に利用できます。
小数点以下の桁数	値とともに表示する小数点以下の桁数を入力します。 このパラメータはタイプが「値」に設定されている場合に利用でき、数値 (浮動小数点)データを返すアイテムのみに影響します。
サイズ	ラベルのサイズを選択します: 自動 - 自動調整されたラベルサイズを使用します; カスタム - カスタムラベルサイズ (ハニカムセルサイズに対するパーセント) を入力します。 ラベルがハニカムセルサイズに収まらない場合は切り捨てられることに注意してください。
太字	チェックボックスをオンにすると、アイテムの単位を太字で表示します。
色	カラーピッカーからアイテム単位の色を選択します。 「D」はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
単位	

単位	チェックボックスをオンにすると、アイテム値とともに単位を表示します。 単位名を入力すると、アイテム設定で設定された単位を上書きします。 このパラメータはタイプが「テキスト」に設定されている場合に利用できます。
位置	アイテム単位の位置 (アイテム値の前または後) を選択します。 このパラメータは、次の時間関連の単位には無視されます: unixtime、uptime、s。 このパラメータはタイプが「テキスト」に設定されている場合に利用できます。
背景色	
背景色	カラーピッカーからハニカムセルの背景色を選択します。 「D」はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
しきい値	
色の補間	チェックボックスをオンにすると、ハニカムセルのしきい値色間のスムーズな遷移が有効になります。 このパラメータは、2 つ以上のしきい値が設定されている場合に利用できます。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値リストは保存時に昇順でソートされます。 しきい値として設定された色は、数値アイテムに対してのみ正しく表示されることに注意してください。 サフィックス(例:「1d」「2w」「4K」「8G」) がサポートされています。値のマッピングもサポートされています。

ハニカムウィジェットで表示される情報は、ウィジェットメニューを使用して*.png* 画像としてダウンロードできます。

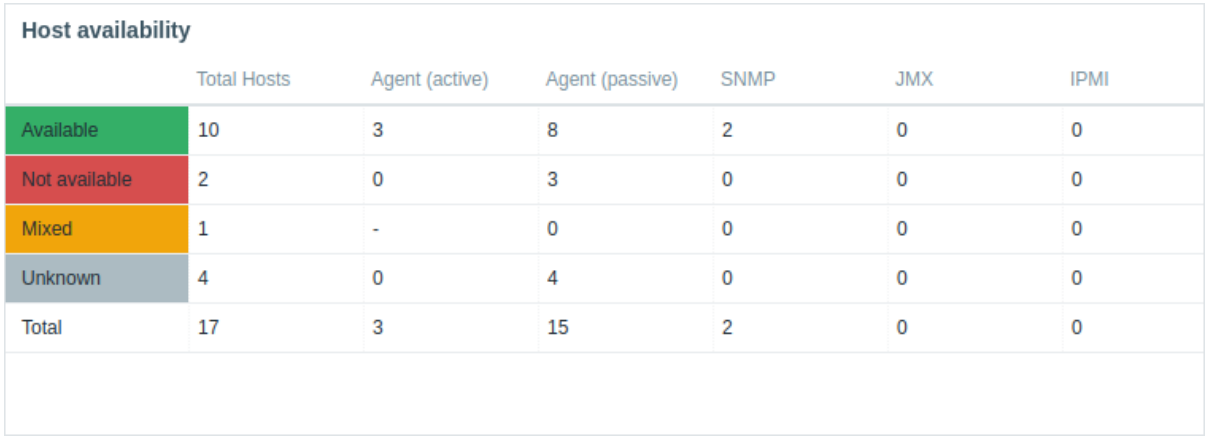


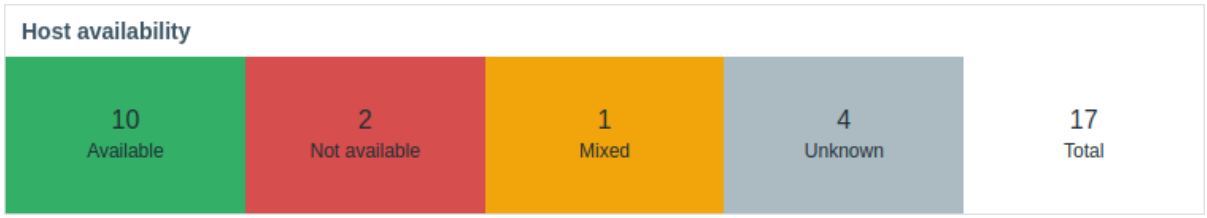
ウィジェットのスクリーンショットはダウンロードフォルダに保存されます。

12 ホスト稼働状況

概要

ホスト可用性ウィジェットは、選択したホストグループ全体の可用性を表示し、稼働中または停止中のホスト数を監視できます。



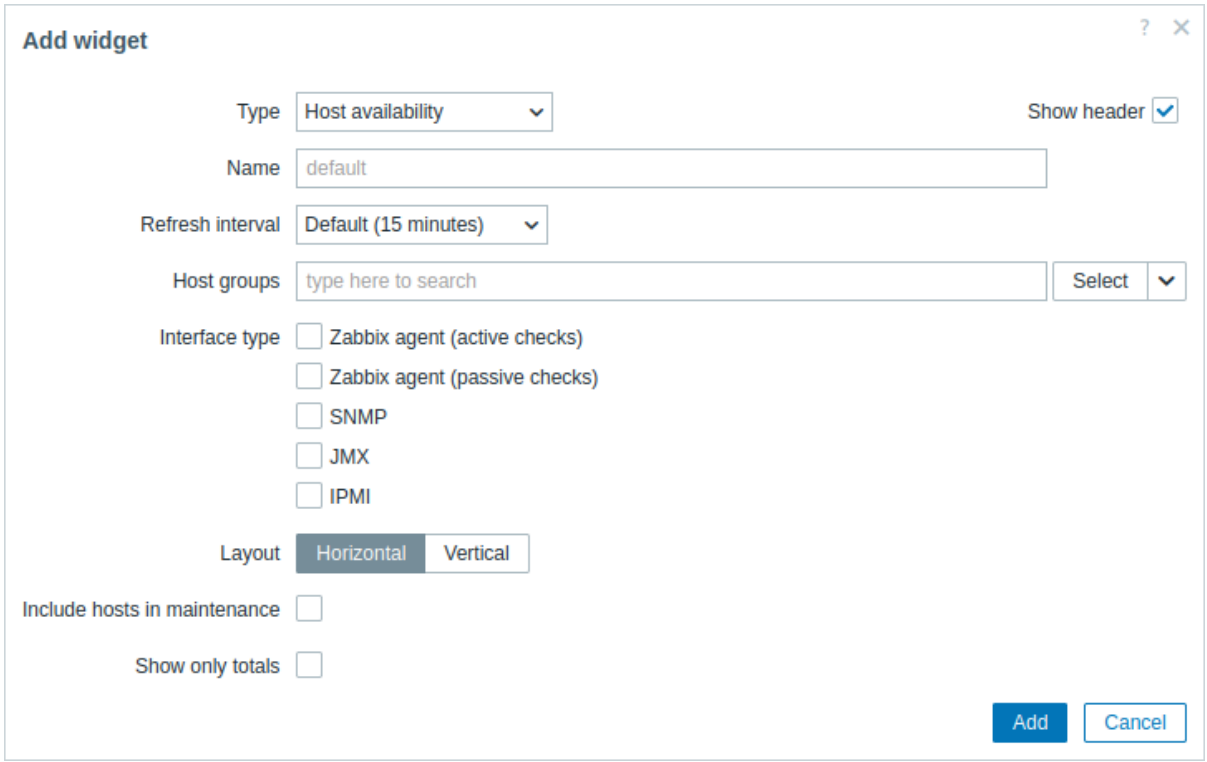


ホストの可用性は次のようにカウントされます:

- 利用可能 - すべてのインターフェースが利用可能なホスト
- 利用不可 - すべてのインターフェースが利用不可なホスト
- 混在 - 少なくとも 1 つのインターフェースが利用不可で、少なくとも 1 つが利用可能または不明なホスト
- 不明 - 少なくとも 1 つのインターフェースが不明で、利用不可なものがないホスト
- 合計 - すべてのホストの合計

設定

設定するには、タイプとして ホストの可用性を選択します。



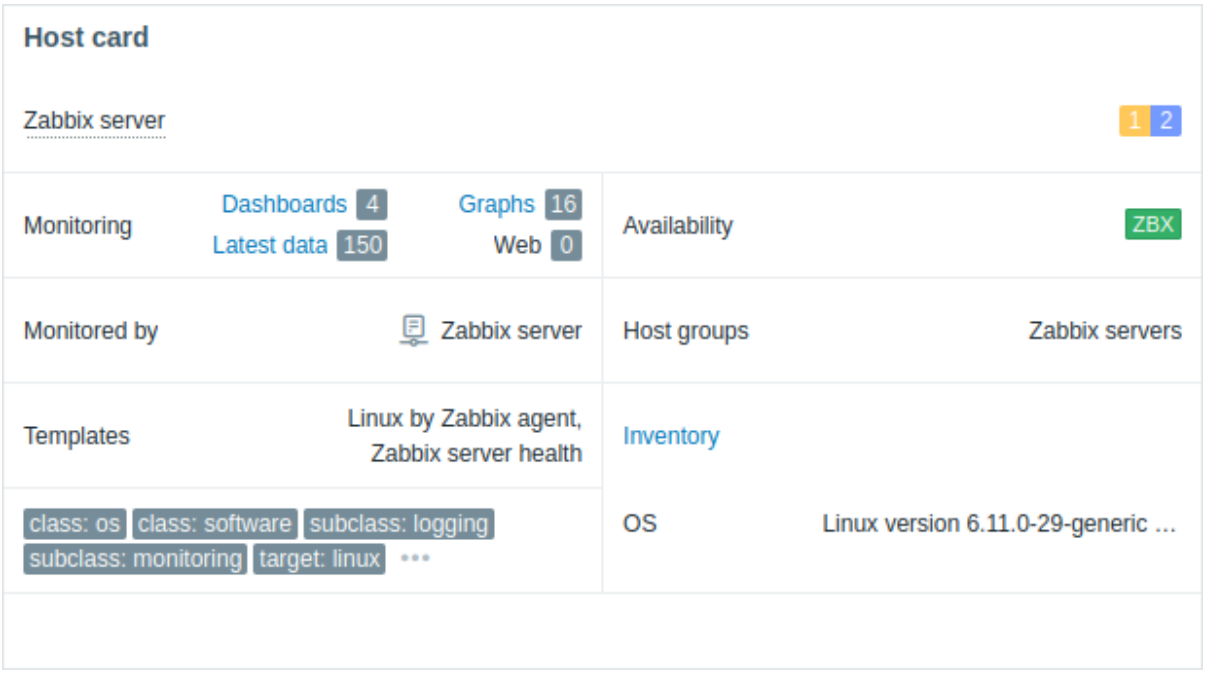
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートなので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
インターフェースタイプ	可用性データを表示するホストインターフェースを選択します—*Zabbix* エージェント（アクティブチェック）、*Zabbix* エージェント（パッシブチェック）、*SNMP*、*JMX*、*IPMI*。 *Zabbix* エージェント（アクティブチェック）の場合、このタイプのアイテムは複数のインターフェースを持つことができないため、混在セルは常に空になります。 何も選択しない場合は、すべてのインターフェースの可用性がデフォルトで表示されます。
レイアウト	横表示（列）または縦表示（行）を選択します。
メンテナンス中のホストを含める	統計にメンテナンス中のホストを含めます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はメンテナンス中のデータを表示と表示されます。
合計のみ表示	チェックを入れると、インターフェースごとの内訳なしでホストの合計が表示されます。このオプションは、1 つのインターフェースのみが選択されている場合は無効になります。

13 ホストカード

概要

ホストカードウィジェットは、単一のホストの詳細を表示し、ホストのステータスや設定 (可用性、問題、インベントリなど) をすばやく評価できます。

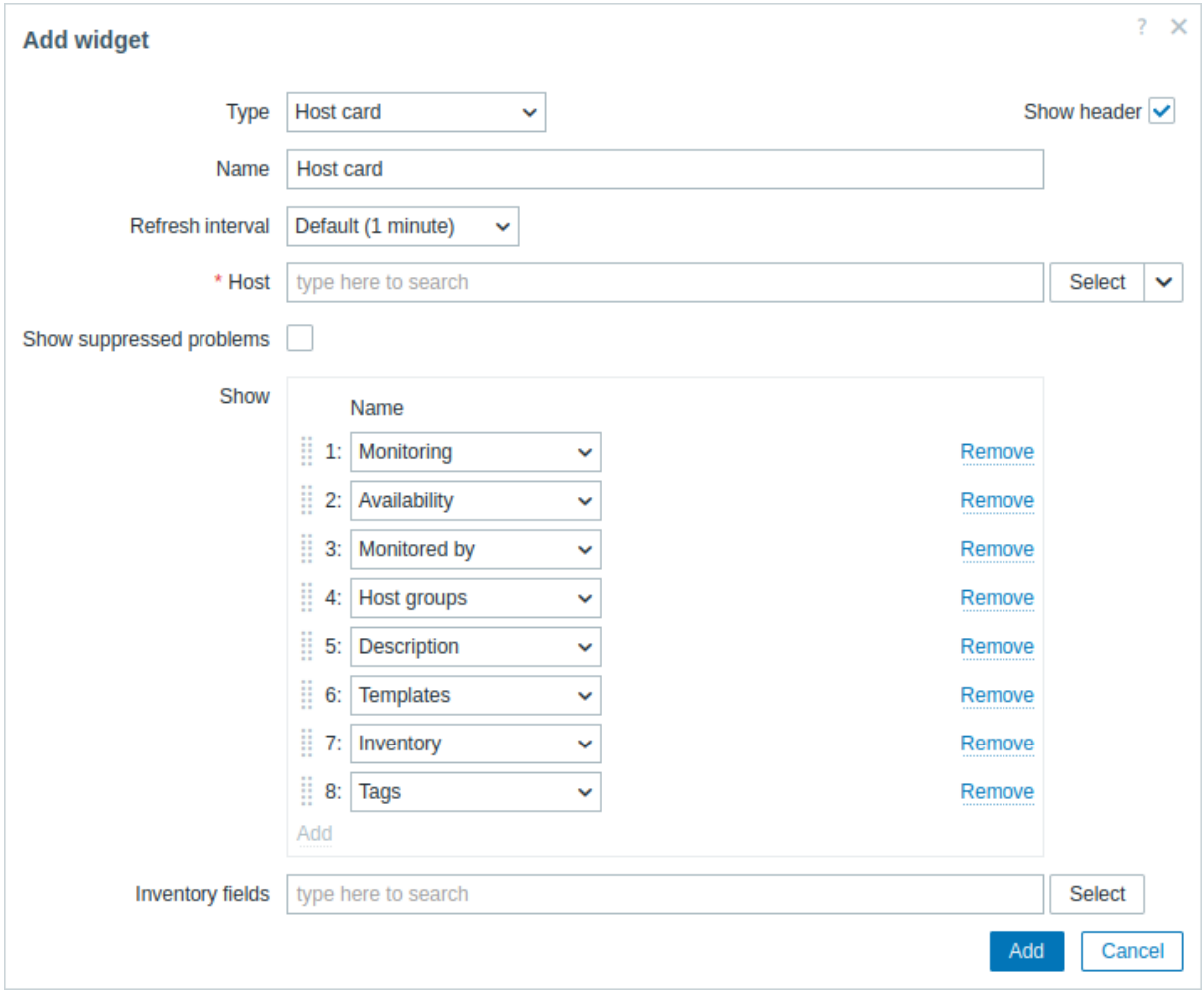


ホストはウィジェットの設定で直接指定するか、互換性のあるウィジェットまたはダッシュボードのホストセレクターから選択できます。

ウィジェットは複数のセクションを表示でき、それぞれが異なる情報を示します。セクションのレイアウトはウィジェットの幅に応じて自動的に調整されます。横に広げると、セクションは自動的に複数の列に再配置されます。

設定

設定するには、タイプとして ホストカードを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有パラメータを設定できます。

ホスト		ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 入力中に一致する候補が表示されます。 ウィジェットの上部セクションには常にホスト名が表示されます（クリックするとホストメニューが開きます）: - ホストが無効の場合、ホスト名の横に赤い無効ラベルが表示されます。 - ホストがメンテナンス中の場合、ホスト名の横にメンテナンスアイコンが表示されます。 ウィジェットの上部セクションには、ホストの障害数も重大度ごとにグループ化して表示されます（クリックするとホストの障害が開きます）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
抑制された障害を表示		このチェックボックスをオンにすると、抑制された障害も障害計算に含めます。
表示		ウィジェットに表示するセクションを追加します。セクションはドラッグ＆ドロップで並べ替えることができます。
	ホストグループ	ホストが所属するホストグループを表示します。
	説明	ホストの説明を表示します。

	監視	ホストのダッシュボード、最新データ、グラフ、Web シナリオへのクイックリンクを表示します。各リンクの横の数字は、それぞれのエンティティの数を示します。
	可用性	インターフェースごとのホストの可用性を表示します。
	監視元	ホストが Zabbix サーバー、特定のプロキシ、またはプロキシグループによって監視されているかどうかを表示します。
	テンプレート	ホストにリンクされているテンプレートを表示します。
	インベントリ	ホストのインベントリフィールドを表示します。
	タグ	ホストのタグを表示します。
インベントリフィールド		表示するインベントリフィールドを選択します。 入力中に一致する候補が表示されます。 インベントリフィールドが指定されていない場合、すべての入力済みインベントリフィールドが表示されます。 このパラメータは、表示にインベントリが含まれている場合に利用できます。

14 ホストナビゲーター

概要

ホストナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてホストのリストを表示します。

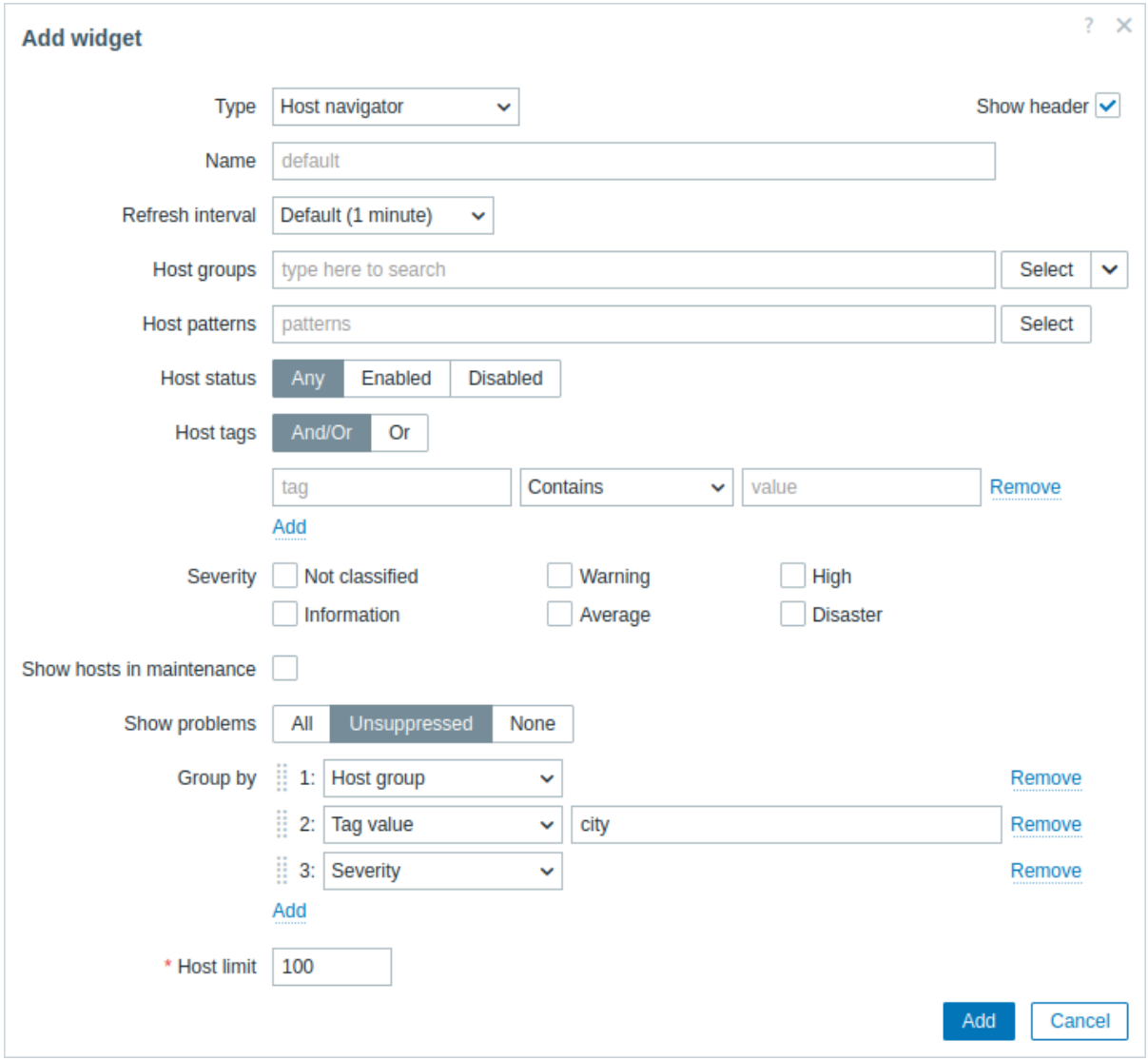
ホストが整理されているグループは、展開または折りたたむことができます。

グループ、障害、メンテナンス中のホストについては、マウスオーバーのヒントで追加の詳細情報にアクセスできます。

このウィジェットは、選択したホストに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。

設定

設定するには、タイプとして ホストナビゲーターを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、すべてのネストされたホストグループが暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットにはすべてのホストグループのすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストパターン	ホストパターンを入力するか、既存のホストをホストパターンとして選択します。指定したパターンに一致するホストがホストナビゲーターに表示されます。 このフィールドはオートコンプリート対応のため、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットにはすべてのホストが表示されます。 ワイルドカードパターンを選択に使用できます（例：`*` は 0 文字以上に一致、`Zabbix*` は”Zabbix”で始まるホストに一致）。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。入力を始めると、ドロップダウンリストには選択したホストグループ内のホストに限定して一致するホストが表示されます（該当する場合）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

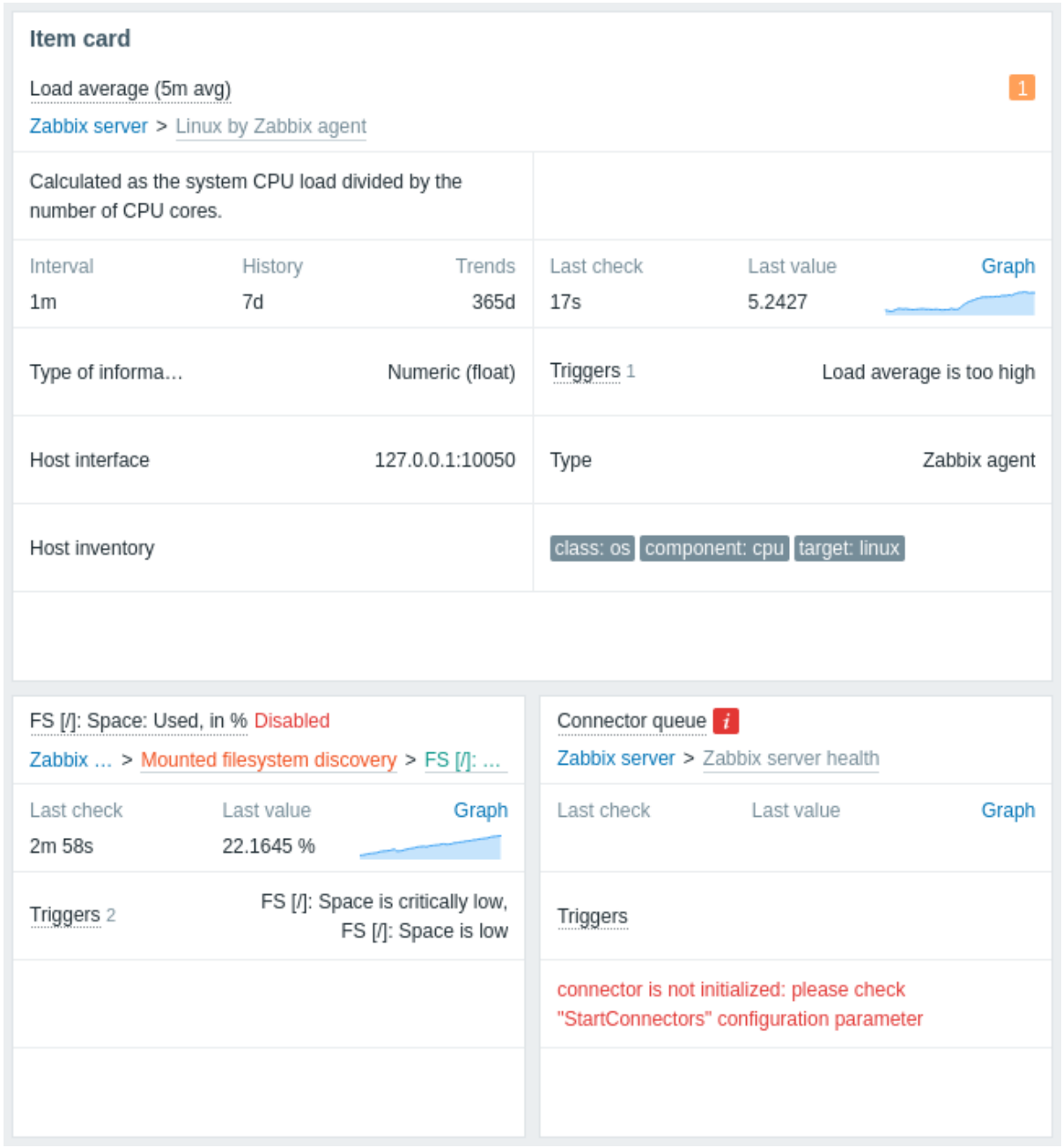
ホストステータス	ステータス（すべて、有効、無効）に基づいて表示するホストをフィルタリングします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストタグ	ウィジェットに表示するホストをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります： 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）； 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）； 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）； 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件の計算タイプは 2 つあります： **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
深刻度	問題の深刻度をマークして、ウィジェットに表示する問題を持つホストをフィルタリングします。 深刻度がマークされていない場合、すべての問題を持つすべてのホストが表示されます。
メンテナンス中のホストを表示	このチェックボックスをマークすると、メンテナンス中のホストが表示されます（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はメンテナンス中のデータを表示と表示されます。
問題を表示	
グループ化	選択したホストをグループ化する属性を追加します： ホストグループ - ホストグループでホストをグループ化; タグ値 - タグ名を入力して、そのタグの値でホストをグループ化（例：”city” と入力すると”Riga”、”Tokyo” などの値でグループ化）； 深刻度 - 問題の深刻度でホストをグループ化。 問題を表示が問題を表示するように設定されている場合、次のように表示されます： - 各深刻度グループには対応する問題数のみが表示されます; - 各ホストにはすべての問題数が表示されます。 ホストは深刻度パラメータでマークされた深刻度でのみグループ化されます。深刻度がマークされていない場合、すべてのホストがすべての深刻度でグループ化されます。 グループ化属性は、グループ名の前のハンドルで上下にドラッグして並べ替えできます。グループ化属性の順序がグループのネスト順序を決定します。たとえば、複数のタグ名（1: color、2: city）を指定すると、color（red、blue など）でグループ化され、次に city（Riga、Tokyo など）でグループ化されます。 ホストは、設定されたグループ化属性によって複数のグループに表示される場合があります（例：ホストグループでグループ化し、ホストが複数のホストグループに属している場合）。そのようなホストをクリックすると、すべてのグループで選択されてハイライトされます。 設定されたグループ化属性に一致しないホストは未分類グループに表示されます。 最大 10 個のグループ化属性を指定でき、すべて一意である必要があります。 グループ化属性が指定されていない場合、ホストはグループ化されません。

ホスト上限	表示するホストの最大数を入力します。1～9999 の値が可能です。 表示可能なホストが設定した上限より多い場合、表示されたホストの下に対応するメッセージが表示されます（例：”100 of 100+ hosts are shown”）。 設定したホスト上限は、設定したグループの表示にも影響します。たとえば、ホスト上限が 100 に設定され、ホストがタグ値でグループ化されている場合（200 以上ある場合）、ウィジェットには最初の 100 個のタグ値と対応するホストのみが表示されます。 このパラメータは、管理 → 一般 → *GUI*の検索およびフィルター結果の上限パラメータの影響を受けません。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

15 アイテムカード

概要

アイテムカードウィジェットは、単一のアイテムの詳細を表示し、アイテムの状態や設定（最新データ、エラー、説明など）をすばやく評価できるようにします。

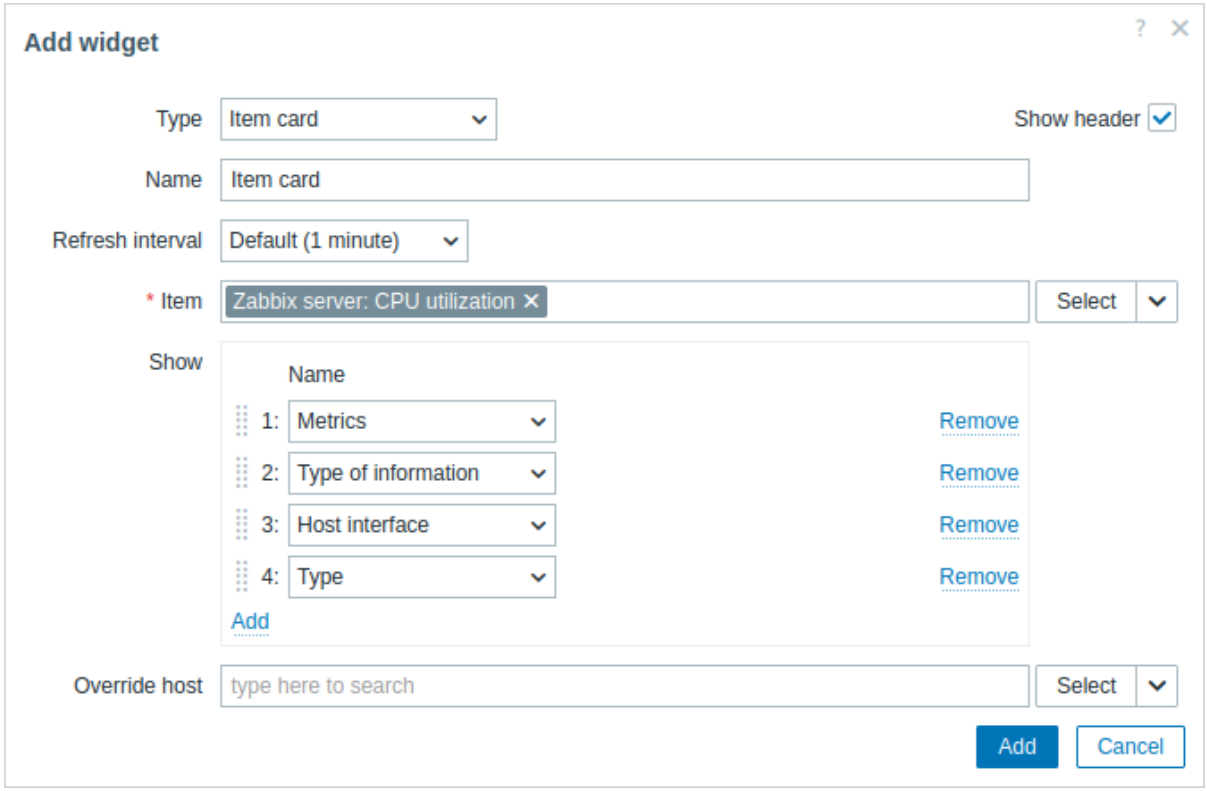


アイテムはウィジェットの設定で直接指定するか、ダッシュボード上の互換性のあるウィジェットから選択できます。

ウィジェットは複数のセクションを表示でき、それぞれが異なる情報を示します。セクションのレイアウトはウィジェットの幅に応じて自動的に調整され、水平方向に拡張すると、セクションは自動的に複数の列に再配置されます。

設定

設定するには、タイプとしてアイテムカードを選択します:



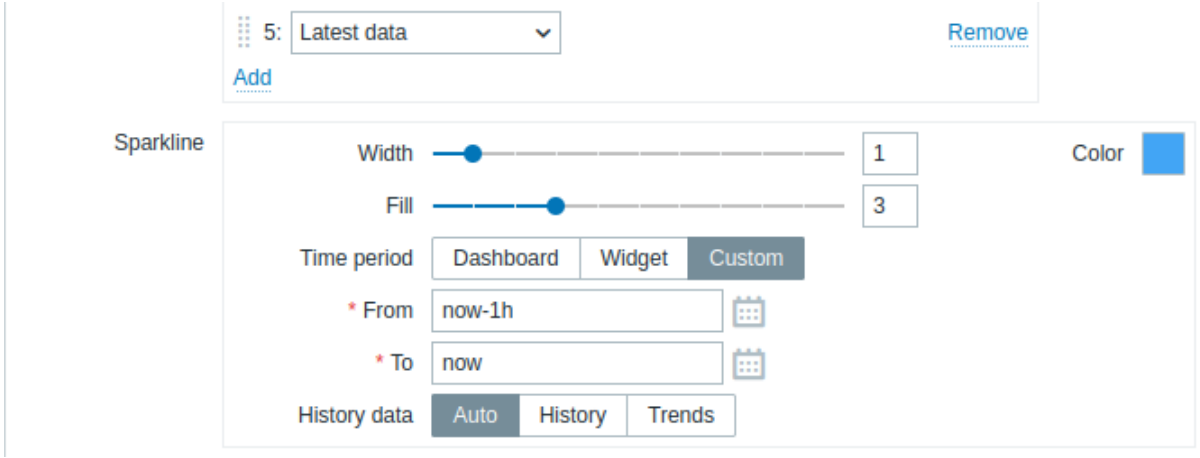
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

アイテム		アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 入力中に一致する候補が表示されます。 ウィジェットの上部セクションには常にアイテム名が表示されます（クリックするとアイテムメニューが開きます）: - アイテムが無効の場合、アイテム名の横に赤い無効ラベルが表示されます。 - エラーがある場合は、四角い「i」アイコンが表示されます。問題がある場合は、問題の深刻度ごとに問題数が表示された四角いアイコンが表示されます。アイコンにカーソルを合わせると詳細のポップアップが表示されます。 - テンプレートダッシュボードでは、アイテムの表示名の代わりに技術名が表示されます。 ウィジェットの上部セクションにはアイテムパス（ホスト名、および該当する場合はテンプレート、低レベルディスカバリ（LLD）ルール、またはマスターアイテム）が表示されます（エンティティをクリックするとその設定が開きます）。 アイテムパスは、該当するエンティティへの権限を持つユーザーにのみ表示されます。
表示		ウィジェットに表示するセクションを追加します。セクションはドラッグ＆ドロップで並べ替えできます。
	説明	アイテムの説明を表示します。マクロが展開されます。
	エラーテキスト	アイテムのエラー（存在する場合）を表示します。
	メトリクス	以下の情報を表示します： - アイテムの更新間隔（トラッパーアイテムを除く）; - アイテムの履歴設定（存在する場合）; - アイテムのトレンド設定（存在する場合）。 マクロが展開されます。 アイテムがカスタム間隔を使用している場合、アイテムの更新間隔の横にヘルプアイコンが表示されます。アイコンにカーソルを合わせるとカスタム間隔の詳細のポップアップが表示されます。

	最新データ	以下の情報を表示します： - 最後のアイテムチェックからの経過時間； - アイテムの最新値； - アイテム値のシンプルグラフ/履歴へのリンク（履歴またはトレンドが保存されている場合）； - 設定可能なスパークラインチャート（アイテムが数値型の場合）。 バイナリ型アイテムの場合、最新値の代わりにサムネイルまたは表示オプションが表示されます。サムネイルにカーソルを合わせると画像のポップアップが表示されます。表示にカーソルを合わせるとアイテム値（Base64 文字列）のポップアップが表示されます。 テンプレートダッシュボードでは、アイテムの最新値の代わりにデータなしが表示されます。
	情報の型	アイテムの情報の型を表示します。
	トリガー	以下の情報を表示します： - トリガー数付きのトリガーラベル（トリガーにカーソルを合わせるとトリガーの詳細のポップアップが表示されます）； - トリガー名のリスト（マクロが展開されます）。
	ホストインターフェース	アイテムで使用されているホストインターフェース、またはホストインターフェースを持たないアイテム（依存アイテム、シンプルチェックなど）の場合はデータなしを表示します。 テンプレートダッシュボードではデータなしが表示されます。
	タイプ	アイテムのタイプを表示します。
	ホストインベントリ	アイテム値が入力されるホストインベントリフィールドを表示します。
	タグ	アイテムのタグを表示します。
ホストの上書き		ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

スパークライン

スパークラインチャートの設定オプションは、最新データセクションがウィジェットに追加されている場合のみ利用できます。



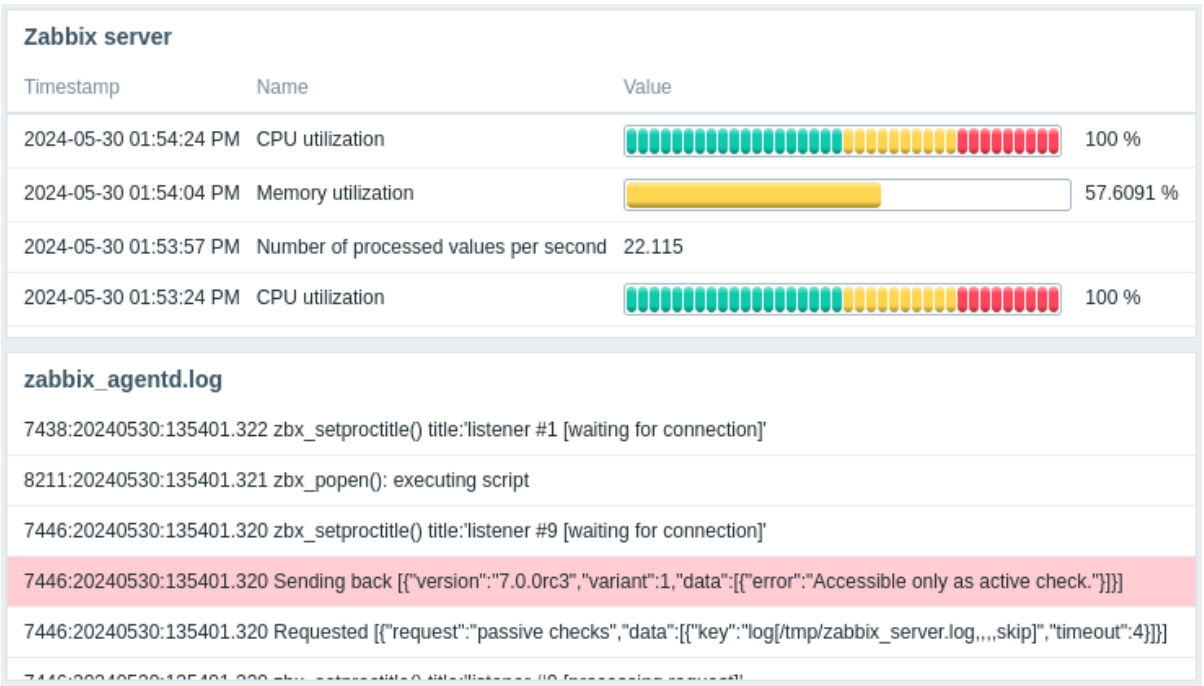
幅	スライダーを使用するか、0 から 10 の範囲で値を手動入力してグラフ線の太さを設定します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーを使用するか、0 から 10 の範囲で値を手動入力して塗りつぶし色の透明度を設定します。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）； カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。

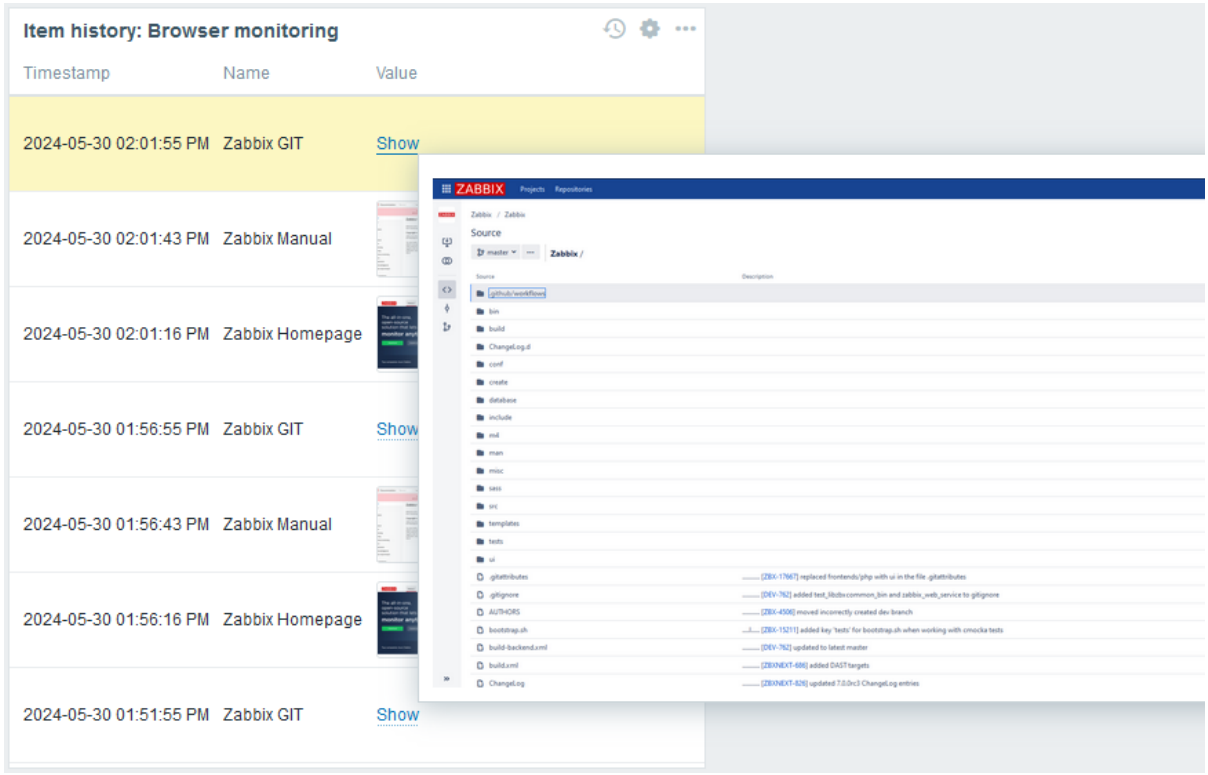
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。 このパラメータは数値データにのみ適用されます。非数値データは常に履歴から取得されます。

16 アイテム履歴

概要

アイテム履歴ウィジェットは、すべてのアイテムタイプ（数値、テキストなど）の最新データをテーブル形式で表示します。進捗バーや画像（ブラウザーアイテムに便利）を表示したり、値をハイライト表示（ログファイル監視に便利）することができます。

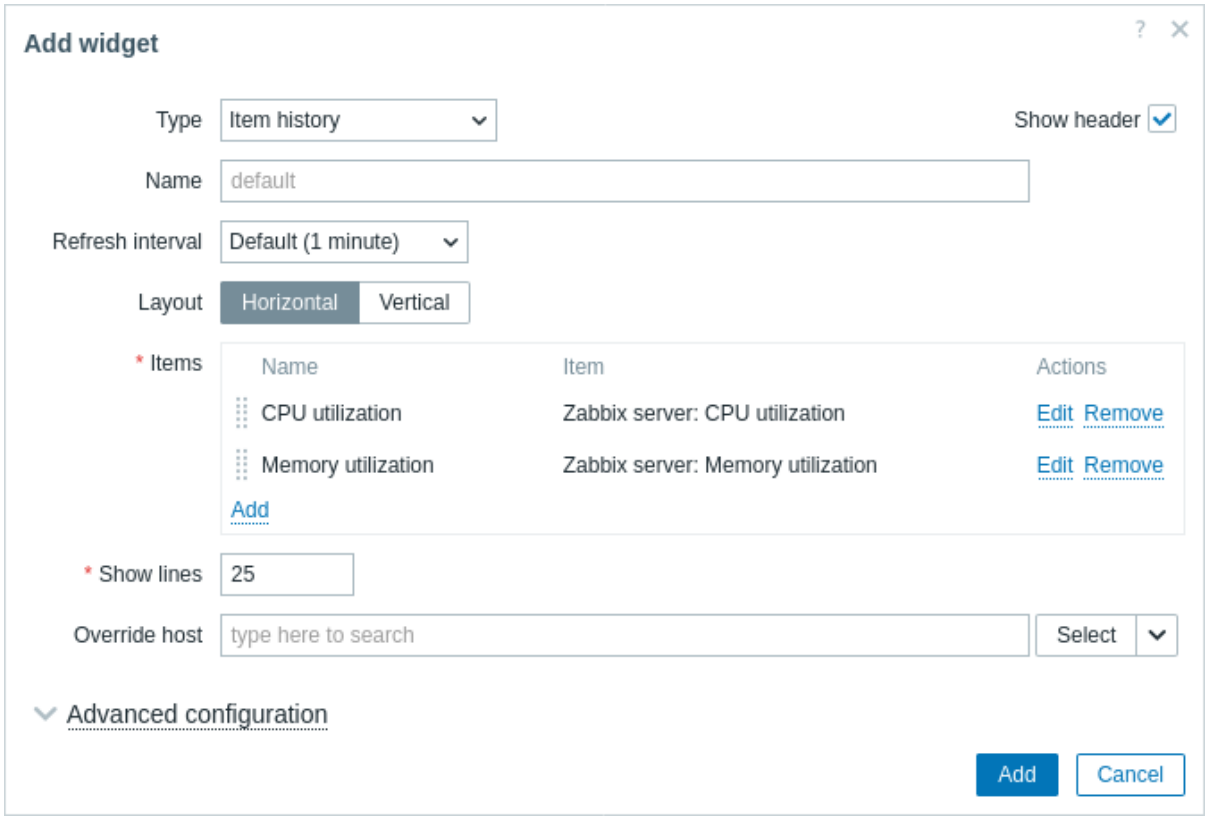




最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして アイテム履歴を選択します:



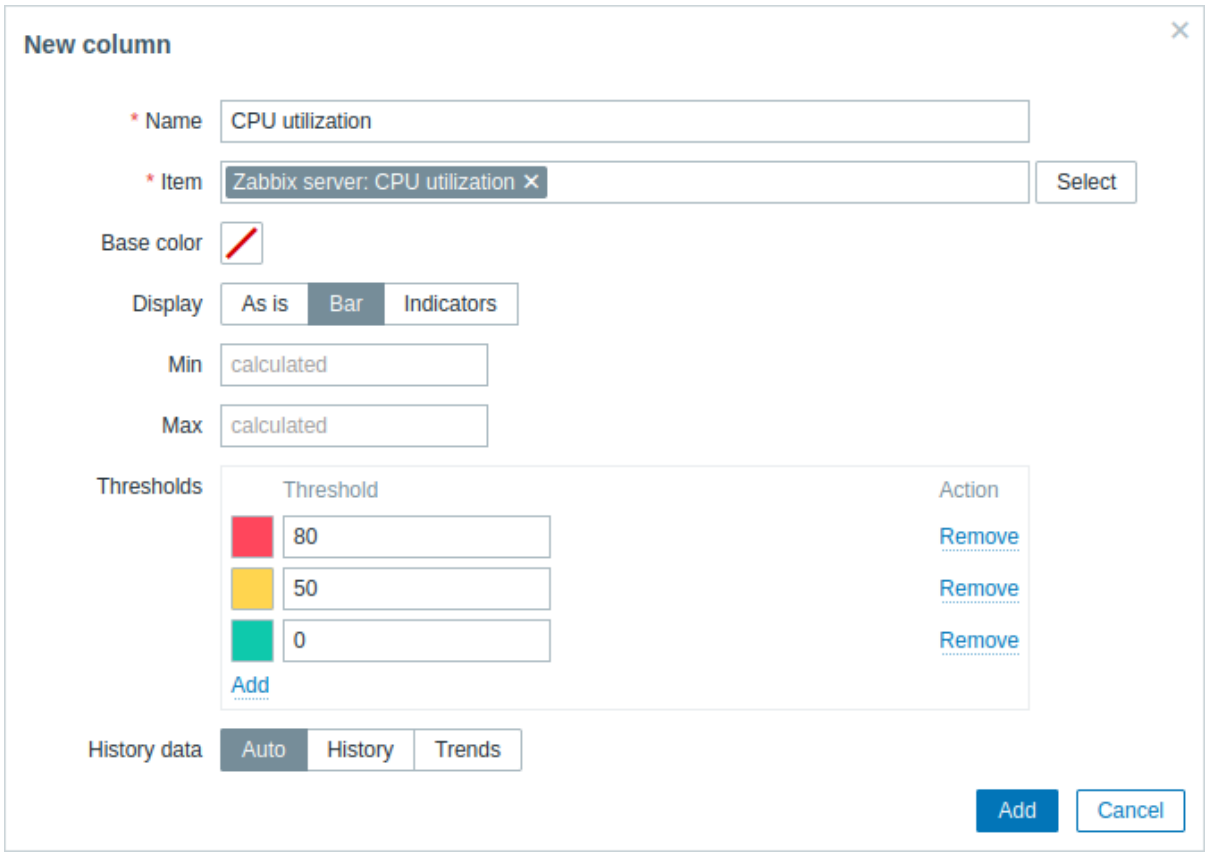
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

レイアウト	アイテム列のレイアウトオプションを選択します: 水平 - アイテムが水平に、値が垂直に表示されます; 垂直 - アイテムが垂直に、値が水平に表示されます。
アイテム	表示する列アイテムを追加します。 アイテムの順序が表示順序を決定します。 アイテム名の前のハンドルで上下にドラッグして並べ替えることができます。

行数の表示	表示するアイテム値の行数を指定します。
ホストの上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのホストセレクタを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。

カラムの設定

アイテムカラムを設定するには、アイテムパラメータで追加をクリックします。



一般的なカラムパラメータ:

名前	カラムの名前を入力します。 空欄の場合、アイテムパラメータのアイテム名が使用されます。
アイテム	アイテムを選択します。 選択したアイテムの情報タイプによってカラムの設定パラメータが異なることに注意してください。詳細は下記の個別パラメータを参照してください。 テンプレートのダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
ベースカラー	カラムの背景色、または表示が「バー」や「インジケーター」の場合は塗りつぶし色を選択します。 ベースカラーはしきい値やハイライトカラーで上書きされる場合があることに注意してください。

数値タイプのアイテムに固有のカラムパラメータ:

表示	アイテム値の表示方法を選択します: そのまま - 通常のテキストとして; バー - ソリッドな色付きバーとして; インジケータ - セグメント化された色付きバーとして。
最小	バー/インジケータの最小値を入力します。 空欄の場合、ウィジェットはアイテムの最小値を使用します。 このパラメータは表示が「バー」または「インジケータ」に設定されている場合のみ利用可能です。

最大	バー/インジケータの最大値を入力します。 空欄の場合、ウィジェットはアイテムの最大値を使用します。 このパラメータは表示が「バー」または「インジケータ」に設定されている場合のみ利用可能です。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値リストは保存時に昇順でソートされます。 サフィックス（例: ”1d”, ”2w”, ”4K”, ”8G”）がサポートされています。値のマッピングもサポートされています。
履歴データ	履歴またはトレンドからデータを取得するか選択します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。

文字、テキスト、ログタイプのアイテムに固有のカラムパラメータ：

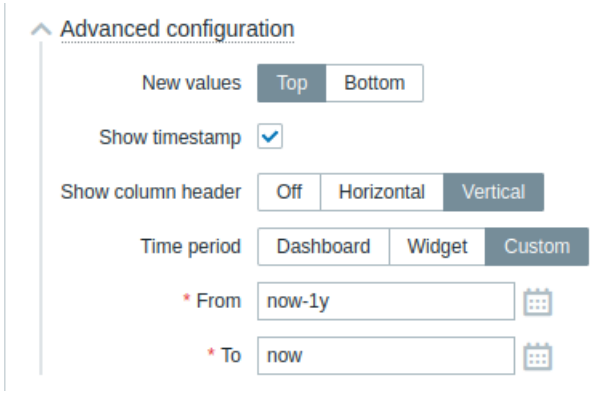
ハイライト	追加をクリックしてハイライトを追加し、カラーピッカーからハイライト色を選択し、正規表現を指定します。 指定した正規表現がテキストに一致するアイテム値の背景色として選択した色が使用されます。
表示	アイテム値の表示方法を選択します： そのまま - そのまま表示： - 入力値に改行がある場合は改行が反映されます； - 水平レイアウトまたは 1 カラムの垂直レイアウトではワードラップなし；2 カラム以上の垂直レイアウトではワードラップあり。 値にマウスオーバーまたはクリックすると、値のポップアップが表示されます。 **HTML** - HTML テキストとして表示； **1** 行 - 指定した長さ（1～500 文字）に切り詰めて 1 行で表示。 切り詰められた値にマウスオーバーまたはクリックすると、完全な値のポップアップが表示されます。
等幅フォントを使用	このチェックボックスをオンにすると、アイテム値が等幅フォントで表示されます（デフォルトはオフ）。
ローカル時刻を表示	このチェックボックスをオンにすると、タイムスタンプカラムにタイムスタンプの代わりにローカル時刻が表示されます。 詳細設定のタイムスタンプを表示チェックボックスもオンにする必要があります。 このパラメータはログタイプのアイテムでのみ利用可能です。

バイナリタイプのアイテムに固有のカラムパラメータ:

サムネイルを表示	画像バイナリの場合はサムネイルを、非画像バイナリの場合は「表示」オプションを表示するには、このチェックボックスをオンにします。 このチェックボックスをオフにすると、すべてのバイナリアイテム値に「表示」オプションが表示されます。 「表示」オプションにマウスを合わせるかクリックすると、アイテム値（画像または Base64 文字列）がポップアップウィンドウで開きます。 アイテム値が空文字列の場合、「表示」オプションが表示され、それにマウスを合わせるかクリックすると「空文字列」と表示されたポップアップが開きます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用できます。



新しい値	新しいアイテム値をどこに追加するかを選択します: 上 - 列の一番上に追加; 下 - 列の一番下に追加。
タイムスタンプを表示	このチェックボックスをオンにすると、タイムスタンプ列が表示されます（デフォルトではオフ）。
列ヘッダーを表示	列ヘッダーの表示方法を選択します: オフ - ヘッダーを非表示; 水平 - ヘッダーを水平に表示; 垂直 - ヘッダーを垂直に表示。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメーターで設定）を使用; カスタム - 開始および終了パラメーターで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフプロトタイプ）を入力または選択します。 このパラメーターは、期間が「ウィジェット」に設定されている場合に利用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。

17 アイテムナビゲーター

概要

アイテムナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてアイテムのリストを表示します。

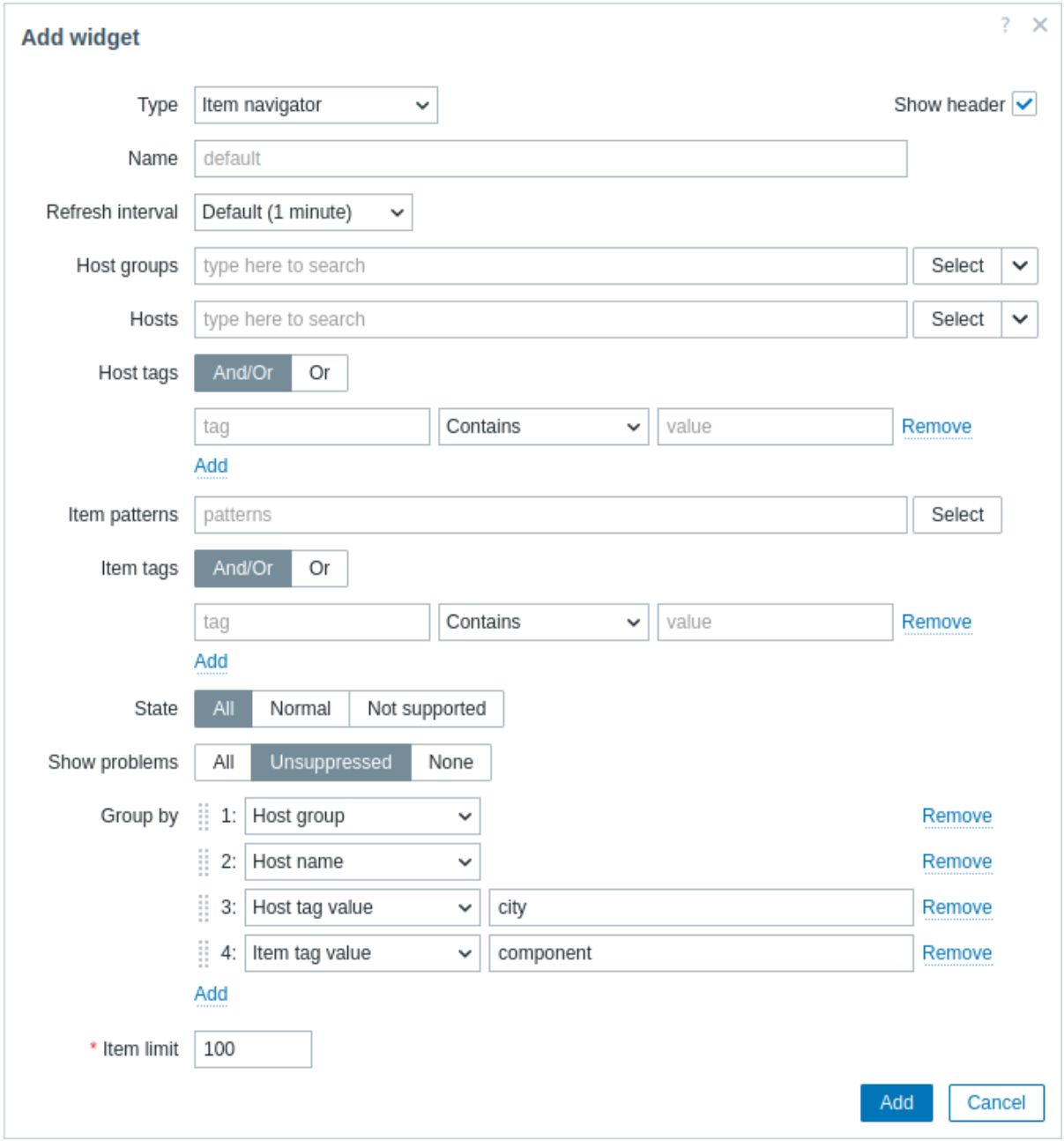
アイテムが整理されているグループは、展開または折りたたむことができます。

グループや問題については、マウスオーバーのヒントで追加の詳細情報にアクセスできます。

このウィジェットは、選択したアイテムに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。

設定

設定するには、タイプとして アイテムナビゲーターを選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、すべてのネストされたホストグループが暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットはすべてのホストグループのすべてのホストに属するアイテムを表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応なので、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットはすべてのホストに属するアイテムを表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

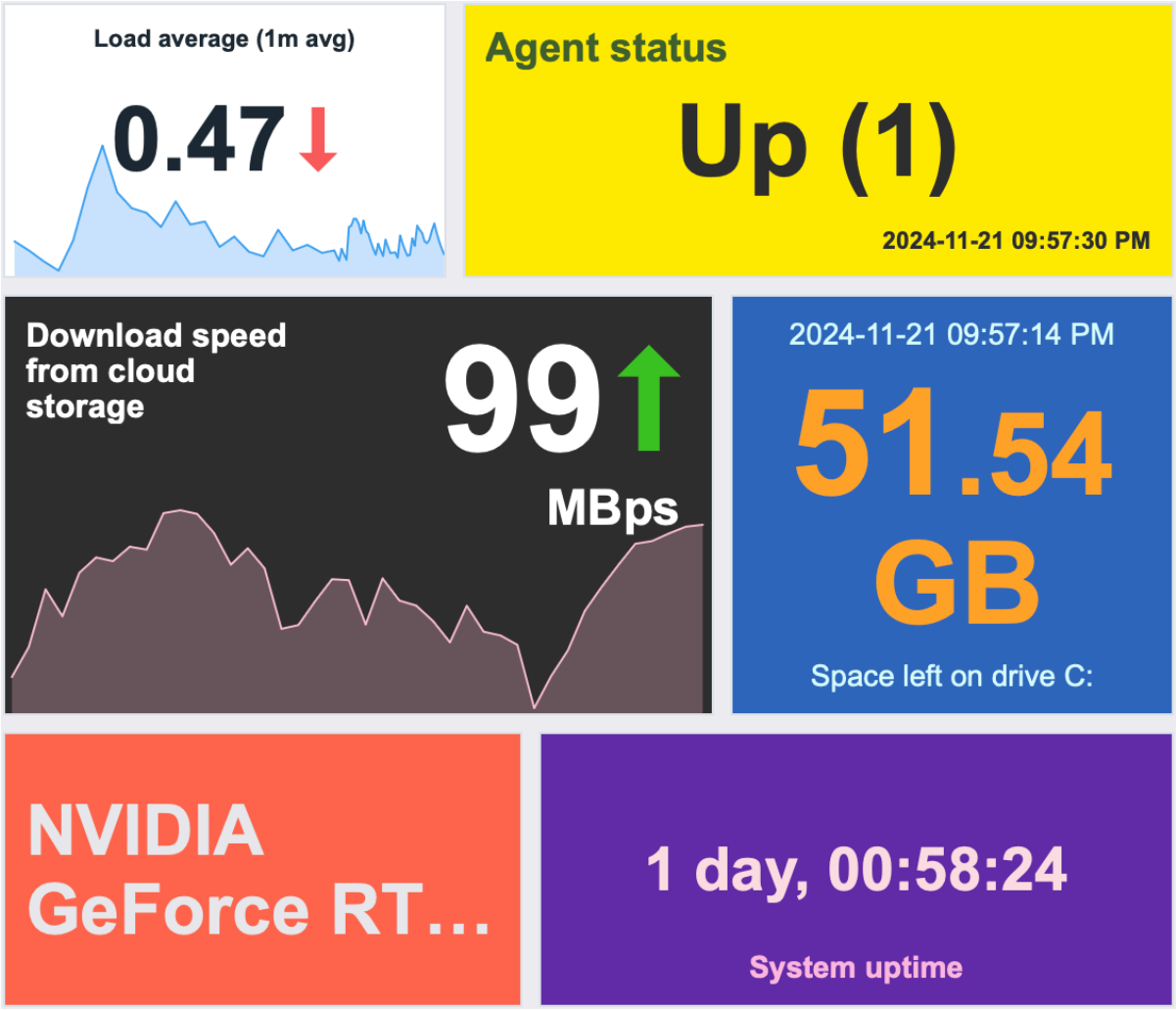
ホストタグ	ウィジェットに表示するアイテムをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - 指定したタグ名でタグ値が入力した文字列を含むものを含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - 指定したタグ名でタグ値が入力した文字列を含むものを除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
アイテムパターン	アイテムパターンを入力するか、既存のアイテムをアイテムパターンとして選択します。指定したパターンに一致するアイテムがアイテムナビゲーターに表示されます。 ワイルドカードパターンを選択に使用できます（例: `*` は 0 文字以上に一致、`Zabbix*` は”Zabbix”で始まるアイテムを返す）。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。入力を始めると、選択したホストまたは選択したホストグループ内のホストに属するアイテムに限定した一致するアイテムのドロップダウンリストが表示されます。ワイルドカード記号は常に解釈されるため、例えば *item** という名前のアイテムを個別に追加することはできません（他に一致するアイテムがある場合、例: item2, item3）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータではテンプレートで設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示するアイテムをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - 指定したタグ名でタグ値が入力した文字列を含むものを含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - 指定したタグ名でタグ値が入力した文字列を含むものを除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
状態	状態（すべて、通常、未サポート）に基づいて表示するアイテムをフィルタリングします。
障害を表示	ステータス（すべて、抑制されていない、なし）に基づいて表示する障害をフィルタリングします。 障害はアイテム行の末尾に色付きブロックで表示されます。色は障害の深刻度に基づき、管理 > 一般 > トリガー表示オプションでカスタマイズできます。障害の深刻度は障害の更新時に変更できます。

グループ化	アイテムをグループ化する属性を追加します: ホストグループ - アイテムのホストのホストグループでグループ化; ホスト名 - アイテムのホストでグループ化; ホストタグ値 - ホストタグの値でアイテムをグループ化するタグ名を入力（例: ”city” と入力すると”Riga”、”Tokyo” などの値でグループ化）; アイテムタグ値 - アイテムタグの値でアイテムをグループ化するタグ名を入力（例: ”component” と入力すると”cpu”、”memory” などの値でグループ化）。 グループ化属性は、グループ名の前のハンドルで上下にドラッグして並べ替えできます。グループ化属性の順序がグループのネスト順序を決定します。例えば、複数のホストタグ名（1: color、2: city）を指定すると、色（red、blue など）でグループ化され、その後都市（Riga、Tokyo など）でグループ化されます。 設定したグループ化属性によっては、1 つのアイテムが複数のグループに表示される場合があります（例: ホストグループでグループ化し、アイテムのホストが複数のホストグループに属している場合）。そのようなアイテムをクリックすると、すべてのグループで選択・ハイライトされます。 設定したグループ化属性に一致しないアイテムは未分類グループに表示されます。 障害を表示が障害を表示するように設定されている場合、障害は以下のように表示されます: - 各グループには、すべての下位アイテムの障害数が表示されます; - 各アイテムには、そのアイテムの障害数のみが表示されます。 最大 10 個のグループ化属性を指定でき、すべて一意である必要があります。 グループ化属性が指定されていない場合、アイテムはグループ化されません。
アイテムの上限	表示するアイテムの最大数を入力します。1～9999 の値が設定可能です。 表示可能なアイテム数が設定した上限を超える場合、表示されたアイテムの下に対応するメッセージが表示されます（例: ”100 of 100+ items are shown”）。 設定したアイテムの上限は、設定したグループの表示にも影響します。例えば、アイテムの上限が 100 に設定され、アイテムがホストごとにグループ化されていて（各ホストに 200 アイテムがある場合）、ウィジェットには最初のホストとその 100 アイテムのみが表示されます。 このパラメータは、管理 → 一般 → *GUI*の検索およびフィルター結果の上限パラメータの影響を受けません。

18 アイテム値

概要

アイテム値ウィジェットは、単一の数値または文字列アイテムの値を表示します。重要なメトリクスを監視したり、しきい値を可視化したり、データの急激な変化を検出したりするのに便利です。



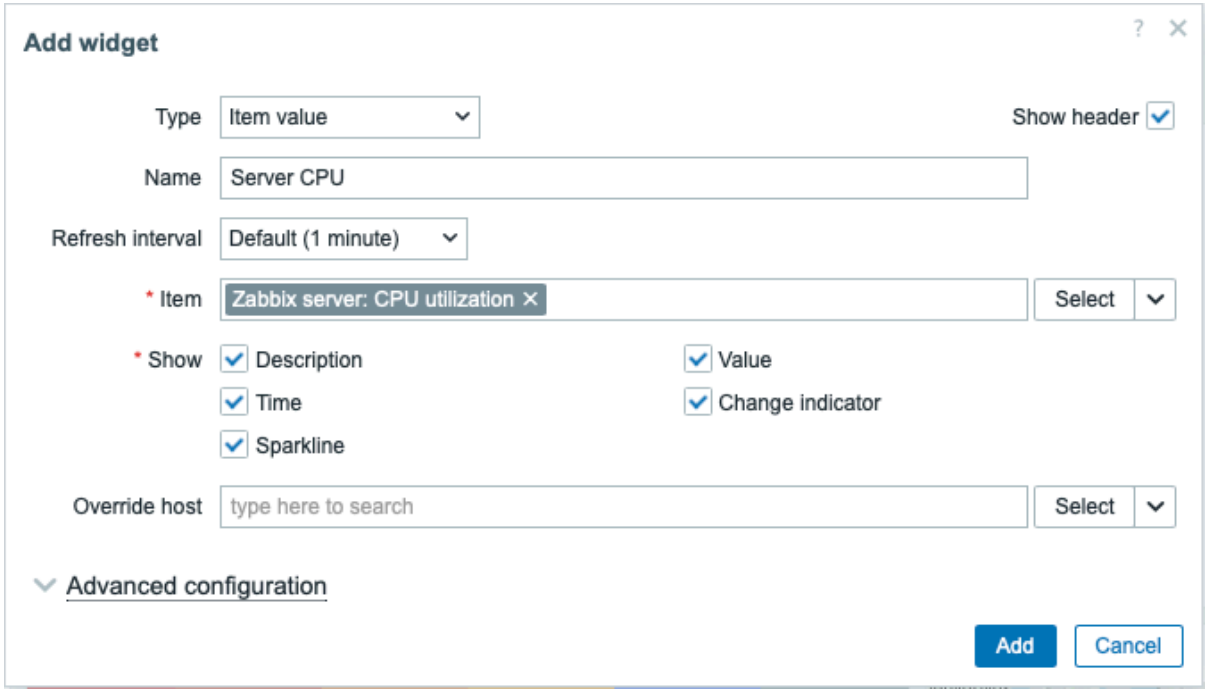
ウィジェットで以下を表示するように設定できます:

- メトリクスのタイムスタンプ (2024-11-21 09:57:30)
- アイテムの説明 (Load average (1m), Agent status)
- アイテムの値、変化の指標、単位 (0.47 ↓, 99 ↑ Mbps)
- 背景色
- 指定した期間の値のスパークラインチャート

ウィジェットをクリックすると、数値アイテムの場合はアドホックグラフ、文字列アイテムの場合は最新データが開きます。

設定

設定するには、ウィジェットタイプとしてアイテム値を選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます:

アイテム	アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 バイナリデータを返すアイテムはサポートされていません。
表示	それぞれの要素（説明、値、時刻、変化インジケーター、スパークライン）を表示するにはチェックボックスをマークします。非表示にするにはチェックを外します。少なくとも 1 つの要素を選択する必要があります。
ホストの上書き	ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。

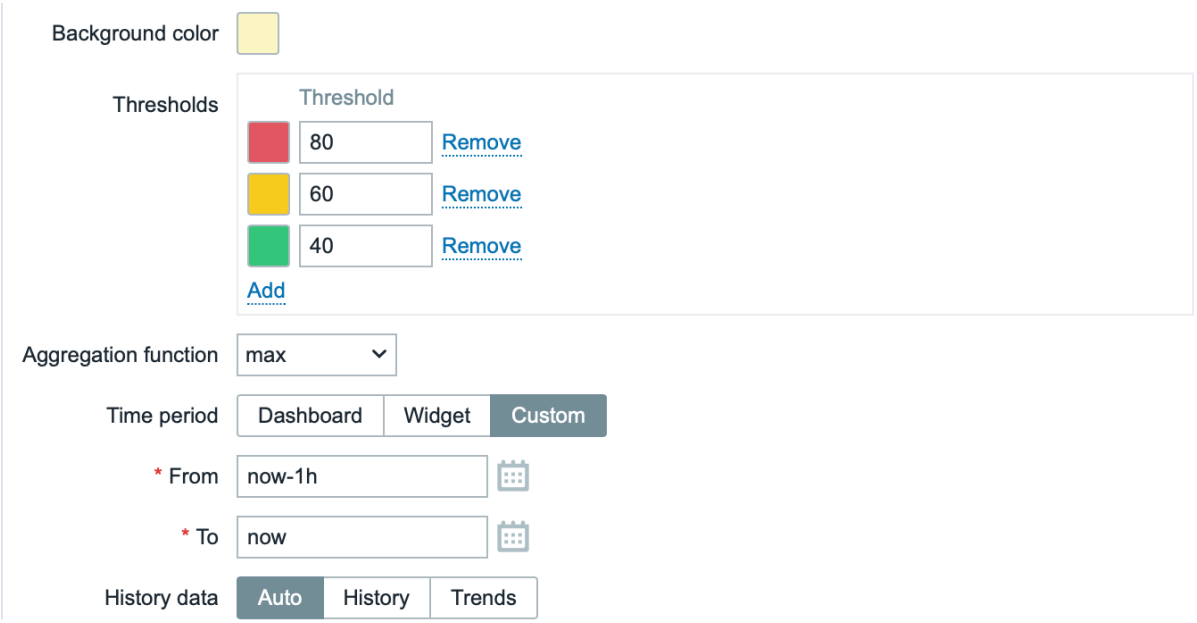
詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用でき、表示フィールドで選択された要素にのみ適用されます (上記参照)。

複数の要素が同じスペースを占有することはできません。同じスペースに配置した場合はエラーメッセージが表示されますのでご注意ください。

一般的なウィジェットパラメータ

これらのパラメータは、ウィジェット全体の背景色（静的または動的）と値を表示するための集計関数を決定します。

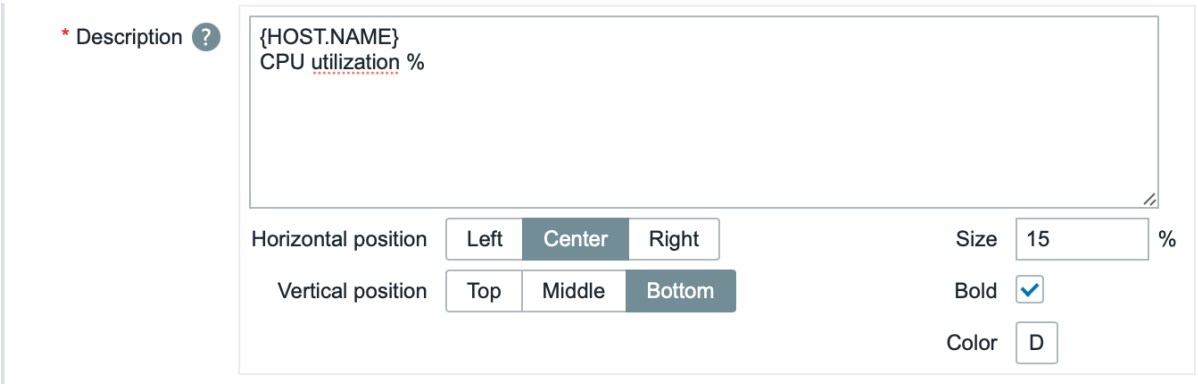


背景色	カラーピッカーからウィジェット全体の背景色を選択します。 D はデフォルト色（フロントエンドのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
しきい値	ウィジェット全体の動的な背景色を設定します。追加をクリックしてしきい値を追加し、カラーピッカーから背景色を選択し、数値を指定します。アイテム値がしきい値と等しいかそれ以上になると、背景色が変わります。 リストは保存時に昇順でソートされます。 動的な背景色は数値アイテムに対してのみ正しく表示されることに注意してください。
集計関数	使用する集計関数を指定します： **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; 使用しない - 最新の値を表示（集計なし）。 集計を使用すると、最新の値の代わりに選択した間隔（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では非数値データが数値に変換されます。
期間	値を集計するために使用する期間。期間のデータソースとしてデータソースを選択します： ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - 開始および終了パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。 このパラメータは、集計関数が「使用しない」に設定されている場合は利用できません。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に利用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。

履歴データ	履歴またはトレンドからデータを取得します： 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。 この設定は数値データにのみ適用されます。非数値データは常に履歴から取得されます。

説明

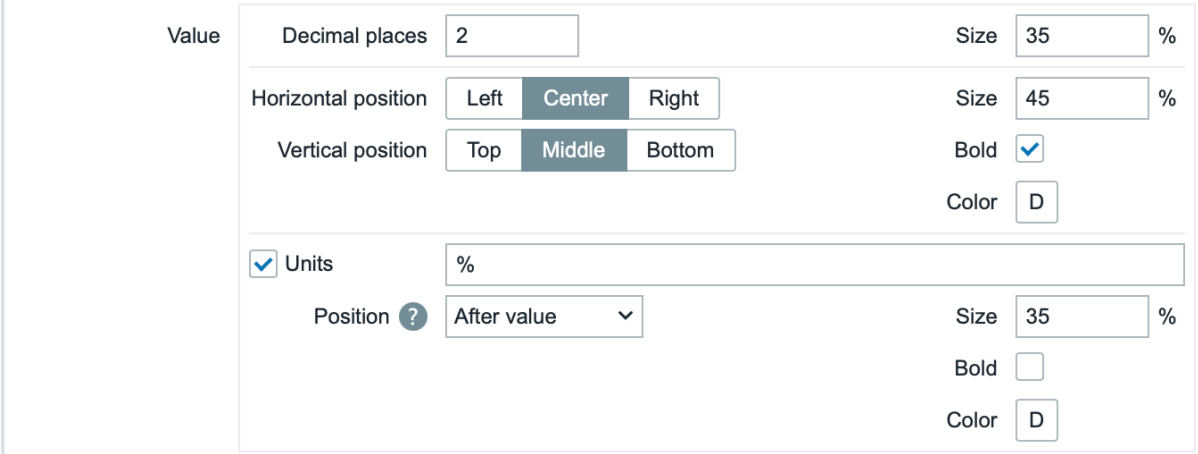
これらのパラメータは、アイテムの説明をどのように表示するかを決定します。



説明	アイテムの説明を入力します。この説明はデフォルトのアイテム名を上書きする場合があります。 複数行の説明がサポートされています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*} およびユーザーマクロがサポートされています。
水平位置	アイテム説明の水平位置を選択します（左、右、中央）。
垂直位置	アイテム説明の垂直位置を選択します（上、下、中央）。
サイズ	アイテム説明のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
太字	チェックボックスをオンにすると、アイテム説明が太字で表示されます。
色	カラーピッカーからアイテム説明の色を選択します。 D はデフォルト色（フロントエンドテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

値

これらのパラメータは、アイテム値の表示方法を決定します。

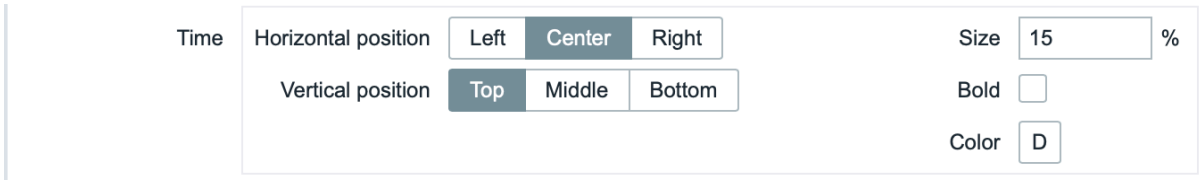


小数点以下の桁数	値に表示する小数点以下の桁数を選択します。この値は float 型のアイテムにのみ影響します。単位が「s」に設定されているアイテムの場合、この値が 0 より大きい場合（デフォルトは 2）、ウィジェットは最も重要な時間単位を四捨五入し、値を完全な時間文字列に変換する代わりに数値（例：「10.43m」）を表示します。0 に設定すると、秒が人間が読みやすい時間文字列（例：「4h 56m 30s」）に変換されます。

サイズ	小数点以下の桁数のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
水平位置	アイテム値の水平位置を選択します（左、右、中央）。
垂直位置	アイテム値の垂直位置を選択します（上、下、中央）。
サイズ	アイテム値のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。 アイテム値のサイズが優先されることに注意してください。他の要素は値のためにスペースを譲る必要があります。ただし、変化インジケーターがある場合、値が大きすぎると変化インジケーターを表示するために切り捨てられます。
太字	チェックボックスをオンにすると、アイテム値が太字で表示されます。
色	カラーピッカーからアイテム値の色を選択します。 D はデフォルト色（フロントエンドテーマによる）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
単位	
単位	チェックボックスをオンにすると、アイテム値に単位が表示されます。単位名を入力すると、アイテム設定の単位を上書きします。
位置	アイテム単位の位置を選択します（値の上、下、前、後）。
サイズ	アイテム単位のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
太字	チェックボックスをオンにすると、アイテム単位が太字で表示されます。
色	カラーピッカーからアイテム単位の色を選択します。 D はデフォルト色（フロントエンドテーマによる）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

時刻

これらのパラメータは、時刻 (アイテム履歴のクロック値) をどのように表示するかを決定します。



水平位置	時刻の水平位置を選択します - 左、右、中央。
垂直位置	時刻の垂直位置を選択します - 上、下、中央。
サイズ	時刻のフォントサイズの高さを入力します (ウィジェット全体の高さに対するパーセント)。
太字	チェックボックスをオンにすると、時刻が太字で表示されます。
色	カラーピッカーから時刻の色を選択します。 D はデフォルト色 (フロントエンドのテーマによって異なります) を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

変化インジケーター

このセクションでは、カラーピッカーから変化インジケーターの色を選択できます。



変化インジケーターは以下の通りです:

- ↑ - アイテム値が上昇（数値アイテムの場合）
- ↓ - アイテム値が下降（数値アイテムの場合）
- ↕ - アイテム値が変化（文字列アイテムおよび値のマッピングがあるアイテムの場合）

注意:

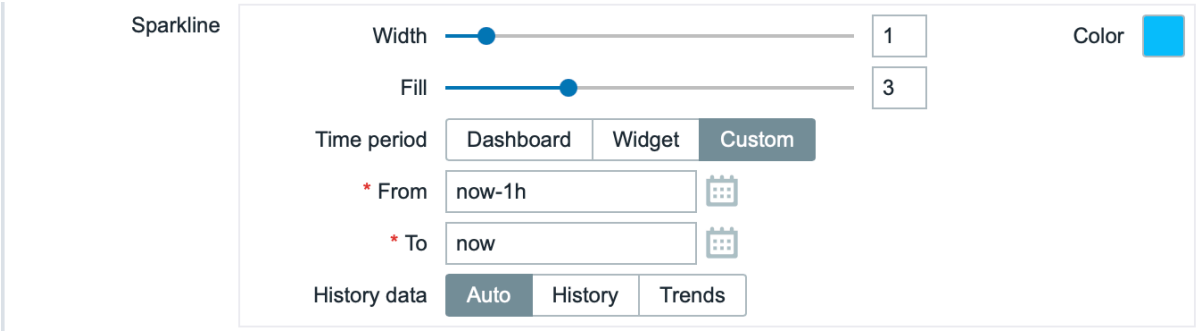
- 値が 1 つだけの場合、上昇（↑）および下降（↓）インジケーターは表示されません。
- 値は常に同じスコープ内で比較されます。例:
  - 最新の値は前の値と比較されます。
  - 月次の値は前月と比較されます。
  - 集計の場合、前の期間は選択した期間と同じ期間で、選択した期間が始まる直前に終了します。

D はデフォルト色（フロントエンドテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

変化インジケーターの縦サイズは値のサイズ（数値アイテムの場合は値の整数部分）と同じです。

スパークライン

これらのパラメータは、スパークラインチャートの表示方法を決定します。

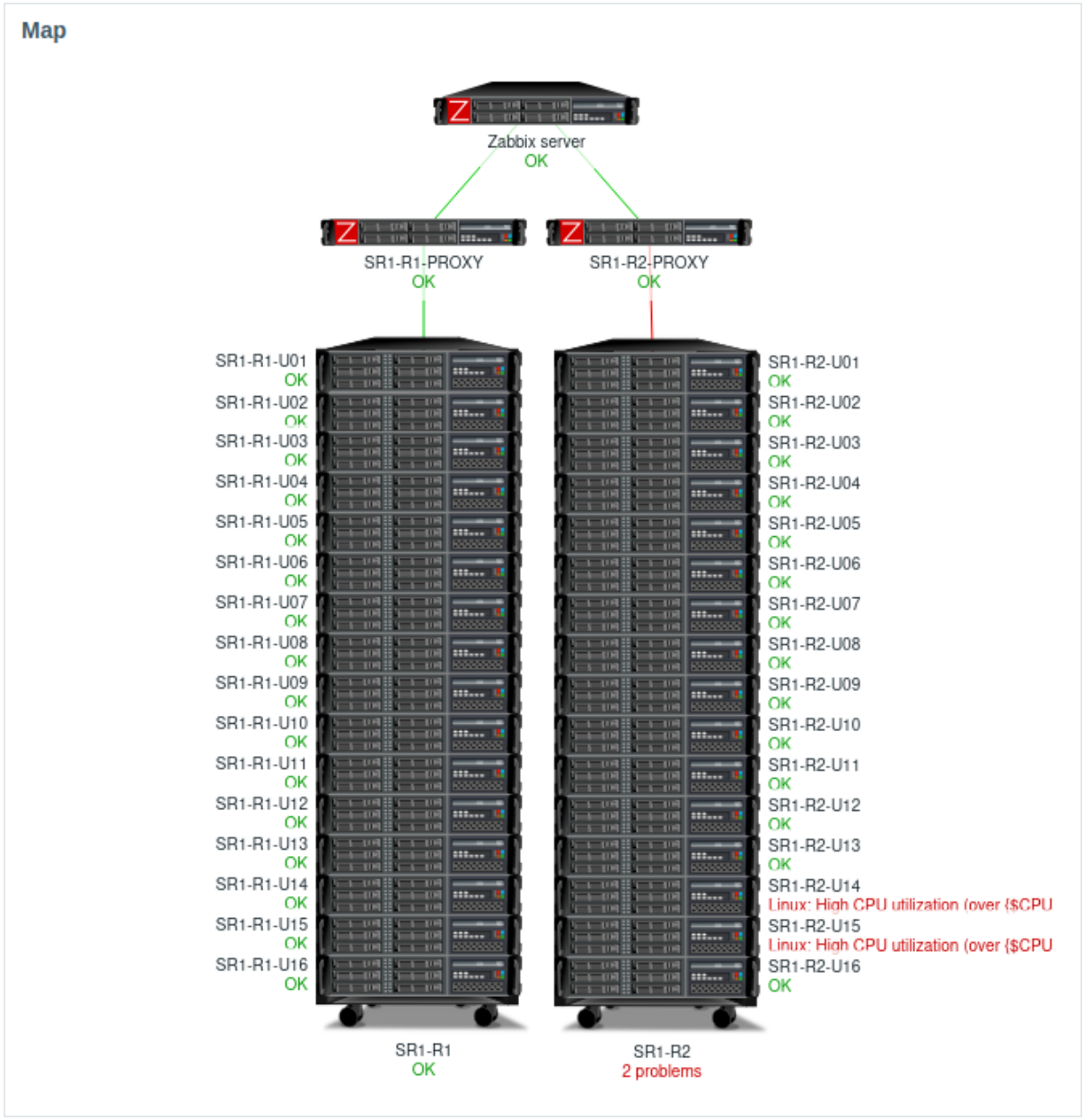


幅	グラフの線の太さをスライダーで設定するか、0 から 10 の範囲で値を手動で入力します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーで塗りつぶしの透明度を設定するか、0 から 10 の範囲で値を手動で入力します。
期間	スパークラインチャートに含める値の期間を指定します。期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）; カスタム - 開始および終了パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。 このパラメータは、集約関数が「使用しない」に設定されている場合は利用できません。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に利用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得;

19 マップ

概要

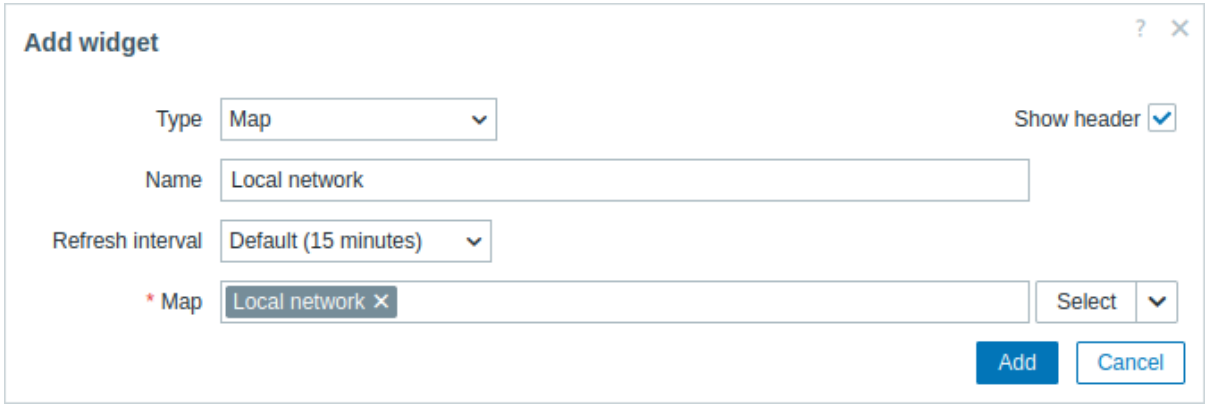
マップウィジェットは、ネットワークマップや類似のビジュアライゼーションを表示し、ネットワークの動的な概要を提供します。



マップナビゲーションツリーウィジェットと組み合わせて、ツリーで選択したマップを動的に表示することもできます。

設定

設定するには、タイプとしてマップを選択します:



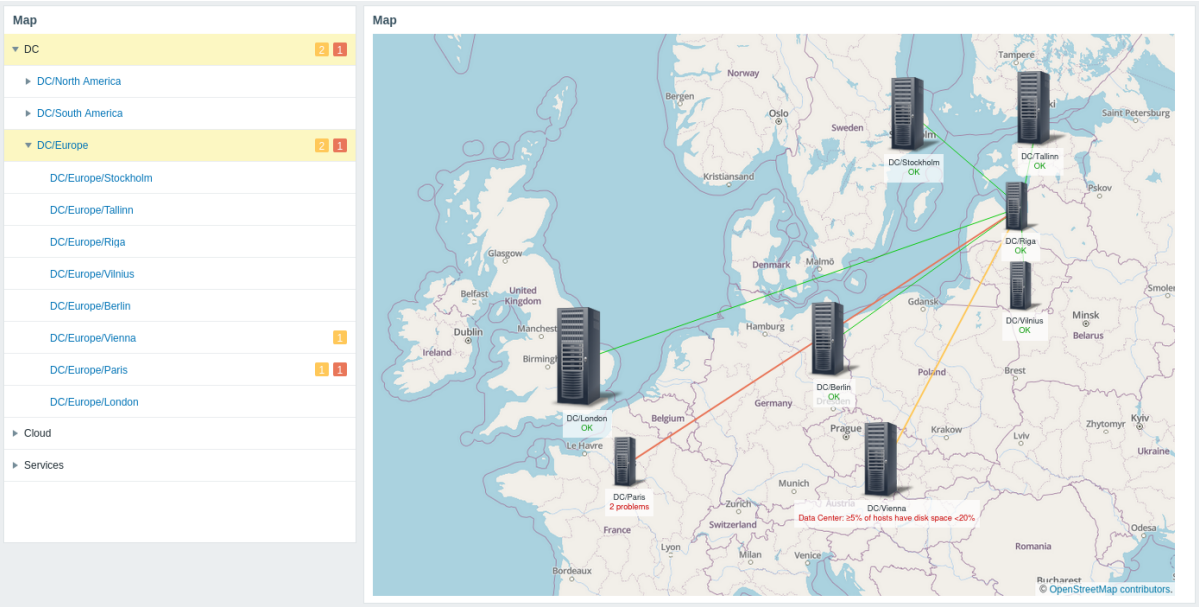
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

マップ	表示するマップを設定します。 または、マップを表示するためのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、マップやウィジェットの名前を入力し始めると、一致するマップやウィジェットのドロップダウンが表示されます。

20 マップナビゲーションツリー

概要

マップナビゲーションツリーウィジェットは、既存のマップの階層と、各マップおよびマップグループの問題数を表示します。

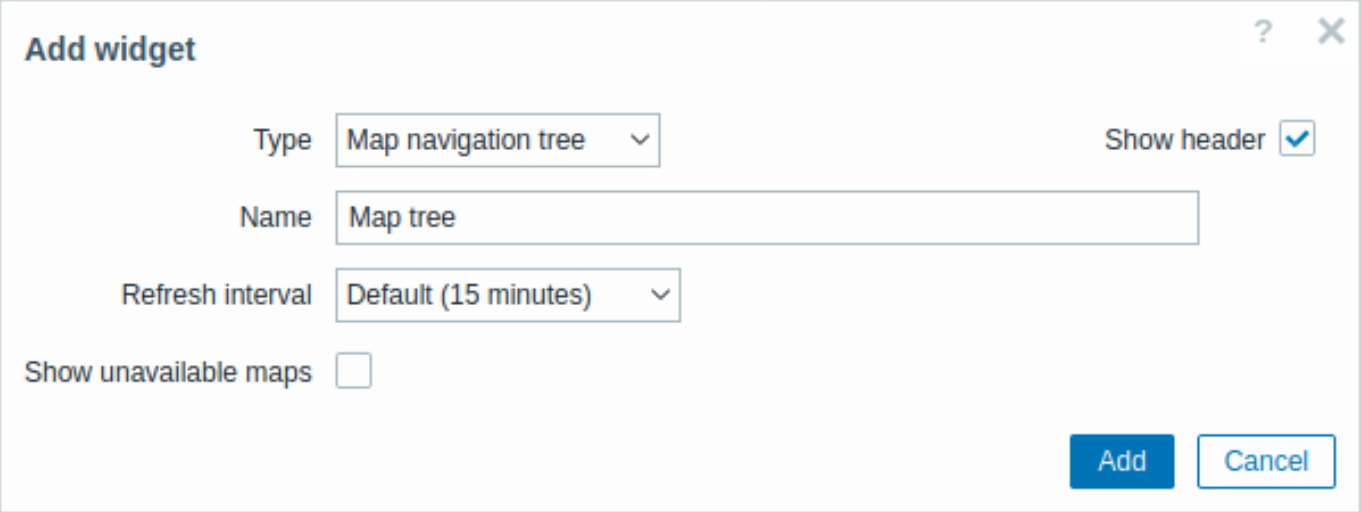


マップウィジェットと組み合わせて、ツリーで選択したマップを動的に表示することができます。

階層内の最上位マップの問題数には、すべてのサブマップの問題と自身の問題が含まれます。

設定

設定するには、タイプとしてマップナビゲーションツリーを選択します:

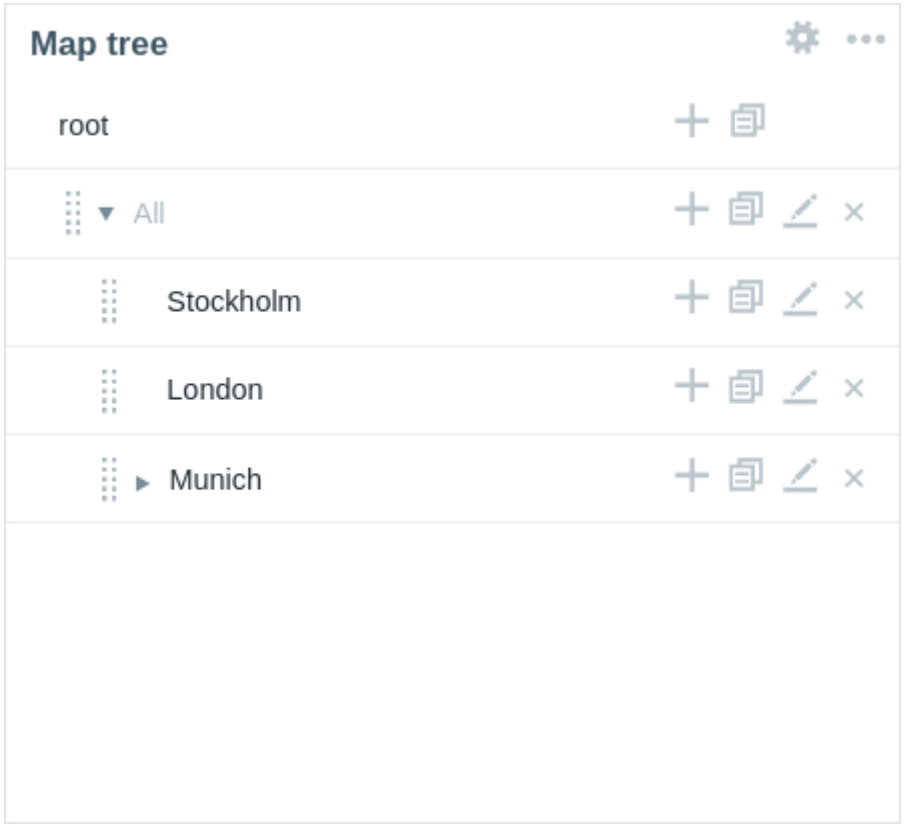


すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

利用できないマップを表示	このチェックボックスをオンにすると、ユーザーが読み取り権限を持たないマップも表示されます。 ナビゲーションツリー内の利用できないマップは、グレーアウトしたアイコンで表示されます。 このチェックボックスをオンにした場合、親レベルのマップが利用できなくても、利用可能なサブマップは表示されます。オフにした場合、利用できない親マップの利用可能なサブマップはまったく表示されません。 問題数は、利用可能なマップおよび利用可能なマップ要素に基づいて計算されます。

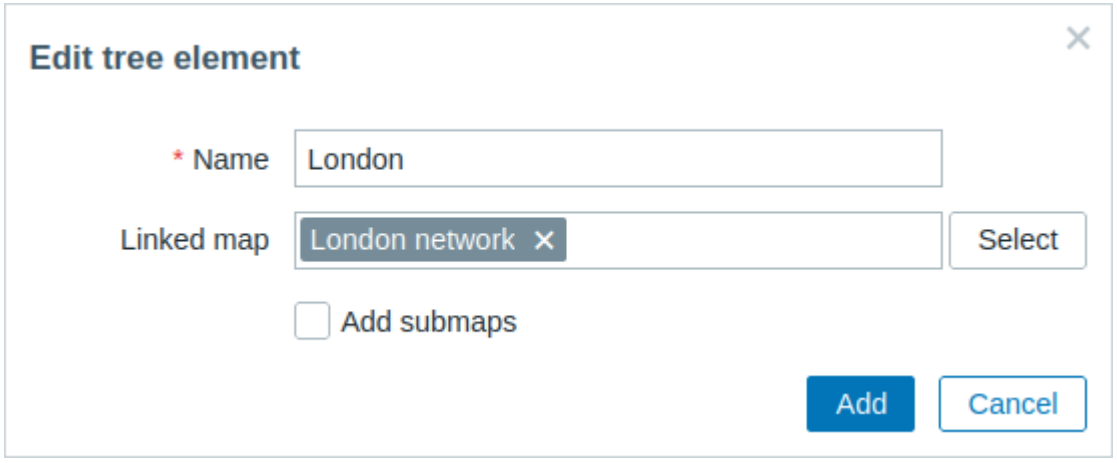
ナビゲーションツリーの要素はリストで表示されます。以下の操作が可能です。

- 要素（子要素を含む）をリスト内の新しい場所にドラッグする
- 要素を展開または折りたたんで、子要素を表示または非表示にする
- 要素に子要素（リンクされたマップの有無にかかわらず）を追加する
- 要素に複数の子要素（リンクされたマップ付き）を追加する
- 要素を編集する
- 要素（子要素を含む）を削除する



要素の設定

ナビゲーションツリー要素を設定するには、新しい要素を追加するか、既存の要素を編集します。



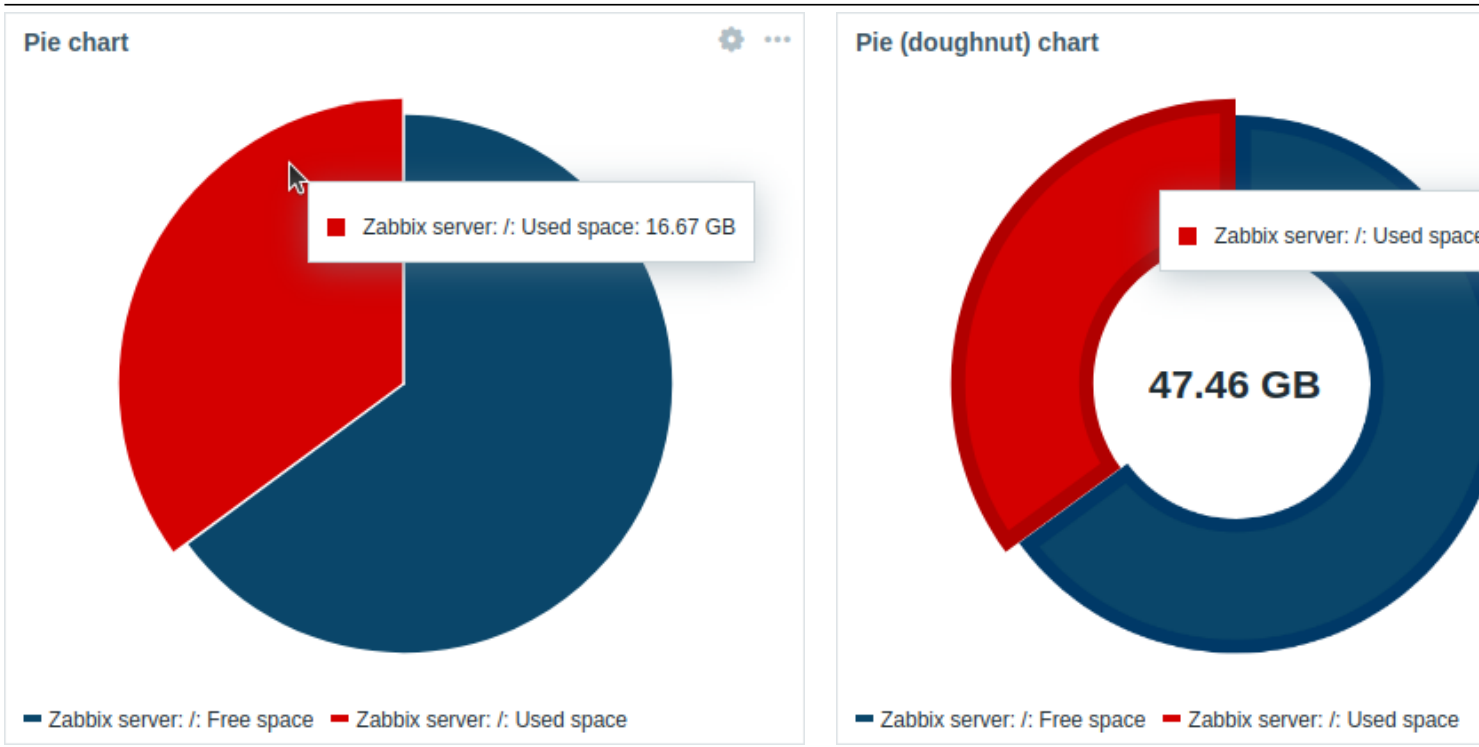
ナビゲーションツリー要素の設定パラメータは以下の通りです。

名前	ナビゲーションツリー要素の名前を入力します。
リンクするマップ	ナビゲーションツリー要素にリンクするマップを選択します。 このフィールドはオートコンプリート対応なので、マップ名の入力を始めると一致するマップのドロップダウンが表示されます。
サブマップを追加	このチェックボックスをオンにすると、リンクされたマップのサブマップをナビゲーションツリー要素の子要素として追加します。

21 円グラフ

概要

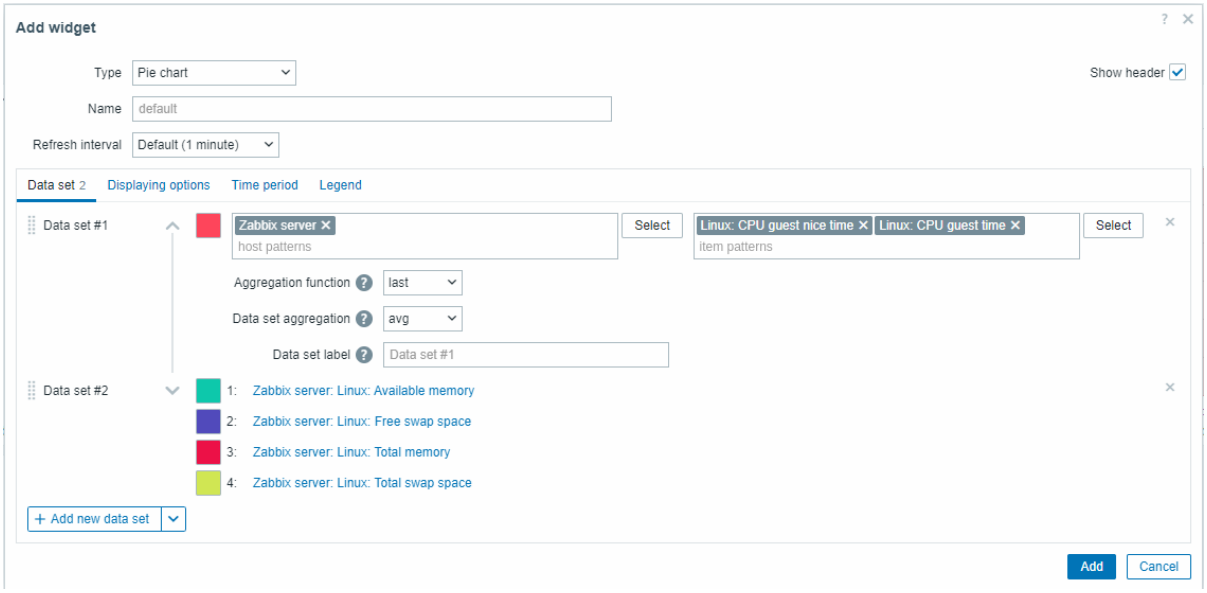
円グラフウィジェットは、数値アイテムデータをベクターベースの円グラフまたはドーナツグラフとして表示します。このウィジェットは、アイテムやホストが全体のデータセットにどのように貢献しているかを視覚化するのに役立ちます。



マウスオーバーすると、ホバーしたセクターが拡大し、その凡例が表示されます。セクターをクリックすると、閉じるまで拡大したまま凡例が表示されます。

設定

設定するには、タイプとして 円グラフを選択します:



データセット

データセットタブでは、データセットを追加して円グラフに表示するデータを選択できます。2 種類のデータセットを追加できます:

- アイテムパターン - 一致するアイテムのデータが表示されます。単一の基本色を選択するか、パレットの行を選択して一致した各アイテムに個別の色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータが表示されます。各アイテムの色をピッカーから個別に選択できます。

デフォルトでは、アイテムパターンのデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストおよびアイテムパターンを選択または入力します。入力したパターンに一致するアイテムのデータが円グラフに表示されます。最大 50 個のアイテムが表示されます。 選択にはワイルドカードパターンを使用できます (例: * は 0 文字以上に一致する結果を返します)。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。 ワイルドカード記号は常に解釈されるため、他に一致するアイテム (例: *item2*、*item3*) がある場合、*item** という名前のアイテムを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストおよびアイテムパターンの指定が必須です。 詳細はデータセットの設定詳細も参照してください。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックして、円グラフに表示するアイテムを選択します。 アイテム名の後のタイプドロップダウンで、各アイテムの表示タイプを選択できます: 通常 - アイテム値が円グラフ上で割合で表示されます (デフォルト); 合計 - アイテム値が円グラフ全体を占めます。円グラフごとに「合計」アイテムは 1 つだけ存在でき、円グラフの凡例の最初に配置されます。アイテムが「合計」に設定されている場合、データセット集計パラメータ (下記参照) は無効になり、「未使用」に設定されます。 また、ウィジェットを追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択できます。ウィジェットにも個別アイテムと同じオプションが適用されます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットの指定が必須です。 詳細はデータセットの設定詳細も参照してください。 数値型アイテムのみが許可されていることに注意してください。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターン指定用のパラメータは利用できず、アイテムリスト指定用のパラメータではテンプレートで設定されたアイテムのみ選択できます。

集計関数	データセット内の各アイテムに使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **sum** - 合計値を表示; **count** - 値の個数を表示; **first** - 最初の値を表示; **last** - 最後の値を表示 (デフォルト)。 集計を使用すると、期間タブで選択した (またはダッシュボード全体で使用される) 間隔 (5 分、1 時間、1 日) の集計値を表示できます。
データセット集計	データセット全体に使用する集計関数を指定します: 未使用 - 集計しない、アイテムを個別に表示 (デフォルト); **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **sum** - 合計値を表示; **count** - 値の個数を表示。 集計を使用すると、期間タブで選択した (またはダッシュボード全体で使用される) 間隔 (5 分、1 時間、1 日) の集計値を表示できます。
データセットラベル	データセットのカスタムラベルを指定します。 ラベルはデータセットの設定および円グラフの凡例 (集計データセットの場合) に表示されます。 すべてのデータセットは、データセットラベルが指定されている場合も含めて番号が付けられます。ラベルが指定されていない場合、データセットには番号に応じて自動的にラベルが付けられます (例:「データセット #2」、「データセット #3」など)。データセットの並べ替え/ドラッグ後に番号付けが再計算されます。 ラベルが長すぎる場合は、表示される場所に合わせて短縮されます (例:「プロセス数...」)。

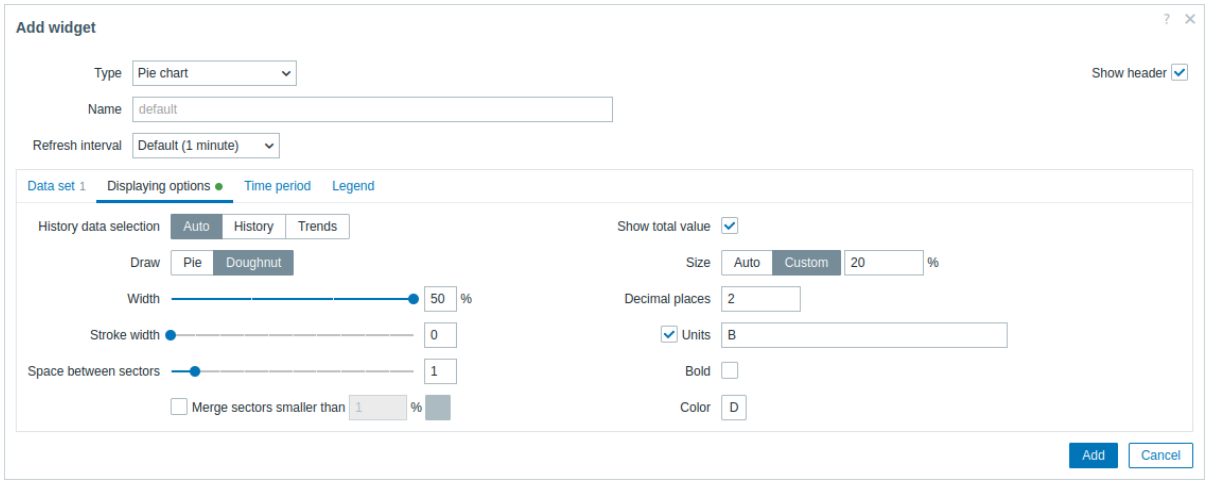
データセットの設定詳細

既存のデータセットはリストで表示されます。これらのデータセットは、並べ替え、展開/折りたたみ、色の変更、クローン作成が可能です。

詳細については、グラフウィジェットのデータセット設定の詳細を参照してください。これらの詳細は円グラフウィジェットにも適用されます。

表示オプション

表示オプションタブでは、円グラフの履歴データの選択と可視化オプションを定義できます。

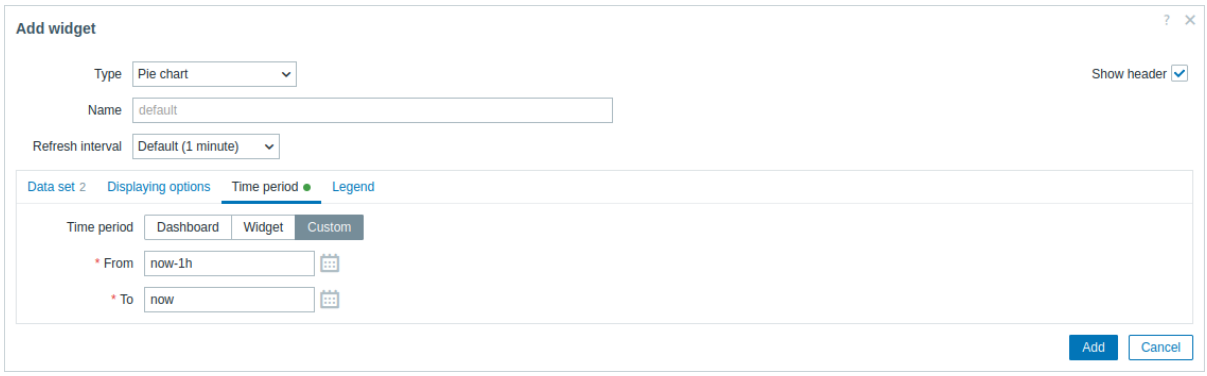


履歴データの選択		データソースを選択: 自動 - 従来のアルゴリズムに従ってデータを取得します（デフォルト）; 履歴 - 履歴からデータを取得; トレンド - トレンドからデータを取得。
描画		円グラフの可視化スタイルを選択: 円グラフ - フルパイ（セクターが半径の 100% を占める）; ドーナツ - 中央が空いた円グラフ（セクターが半径の最大 50% を使用）。
セクター間のスペース		セクター間のスペースサイズ（0-10 の単位）を選択（デフォルトは「1」）。
N% 未満のセクターを結合		N% 未満のセクターを結合するにはチェックボックスをオンにします。 有効にした場合、結合されたセクターの色と、小さいセクターを結合するためのパーセンテージ閾値（N）を選択します。
描画スタイル: ドーナツ		
	幅	ドーナツの幅を選択：半径の 20、30、40 または 50%（デフォルト）。
	ストローク幅	ドーナツセクターの境界線の幅（0-10）を選択。
	合計値を表示	ドーナツグラフの中央に合計値を表示するにはチェックボックスをオンにします。
	サイズ	合計値のサイズオプションを選択： 自動 - テキストはドーナツの中央に読みやすく収まるよう自動的にサイズ調整されます； カスタム - テキストサイズをウィジェット全体の高さに対するパーセンテージで指定します。
	小数点以下の桁数	合計値の小数点以下の桁数（0-6）を指定します。
	単位	合計値の単位を指定します。
	太字	合計値を太字で表示するにはチェックボックスをオンにします。

色	合計値の色を選択します。

期間

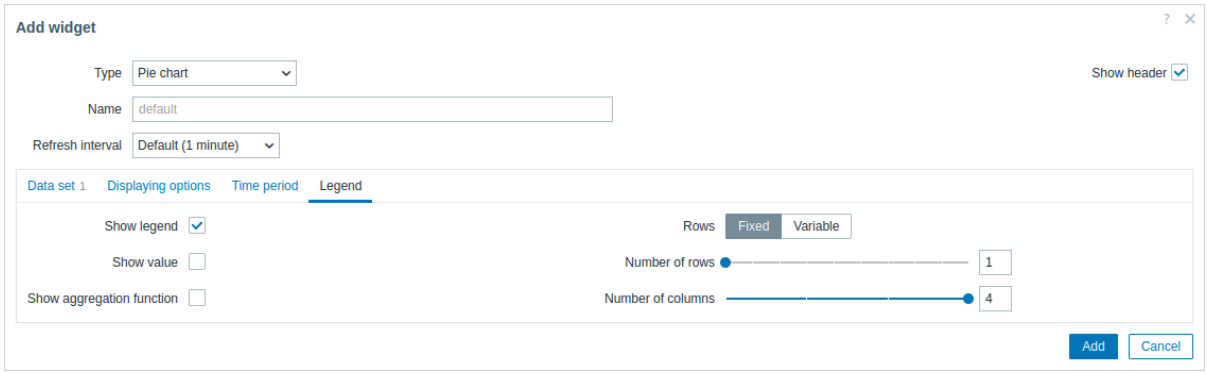
期間タブでは、円グラフの集計設定にカスタム期間を設定できます。



期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用します; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメーターで設定）を使用します; カスタム - *From* および *To* パラメーターで指定されたカスタム期間を使用します; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できることに注意してください。
ウィジェット	期間のデータソースとして互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフプロトタイプ）を入力または選択します。 このパラメーターは、期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。

凡例

凡例タブでは、円グラフの凡例をカスタマイズできます。



凡例を表示	このチェックボックスのチェックを外すと、円グラフの凡例が非表示になります（デフォルトでチェックされています）。
値を表示	このチェックボックスをオンにすると、凡例にアイテムの値が表示されます。
集計関数を表示	このチェックボックスをオンにすると、凡例に集計関数が表示されます。
行数	凡例の行の表示モードを選択します： 固定 - 表示される行数は行数パラメータの値で決まります。 可変 - 表示される行数は設定されたアイテム数で決まり、最大行数パラメータの値を超えません。
行数/ 最大行数	行数が「固定」の場合、表示する凡例の行数を設定します（1～10）。 行数が「可変」の場合、表示する凡例の最大行数を設定します（1～10）。

列数	表示する凡例の列数を設定します（1～4）。 このパラメータは値を表示がオフの場合に利用できます。

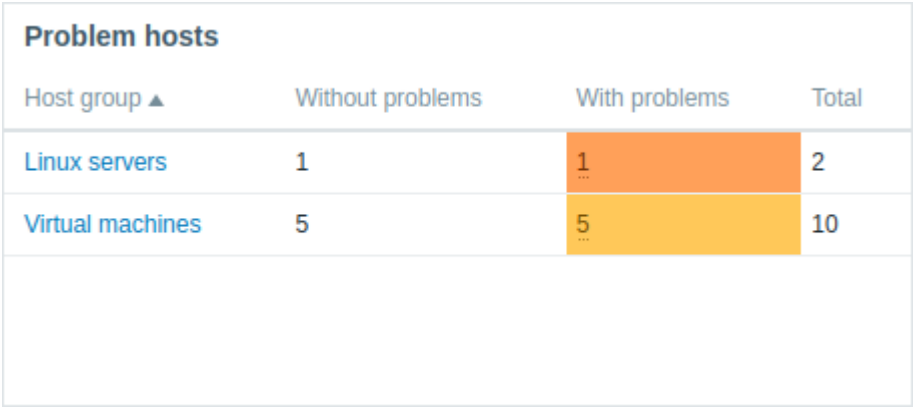
円グラフウィジェットで表示される情報は、ウィジェットメニューから.*png* 画像としてダウンロードできます。

ウィジェットのスクリーンショットはダウンロードフォルダに保存されます。

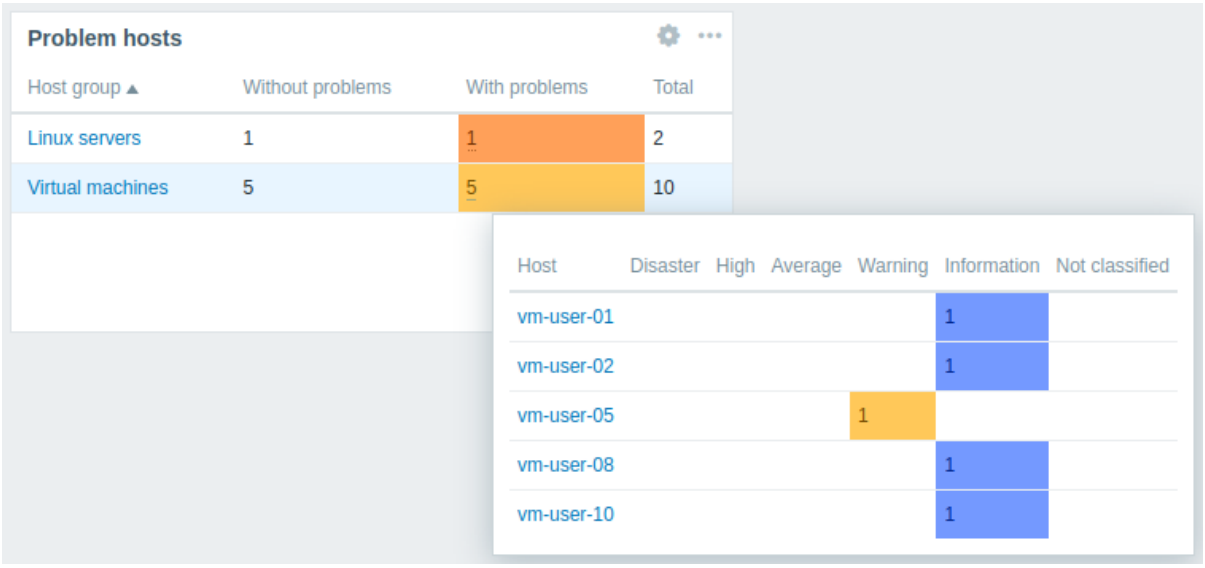
22 問題のあるホスト

概要

問題ホストウィジェットは、ホストグループごとの原因となる障害の数を表示し、各グループで最も高い障害の深刻度を示します。さまざまなパラメータ（ホストグループ、ホスト、障害名など）で障害をフィルタリングできます。

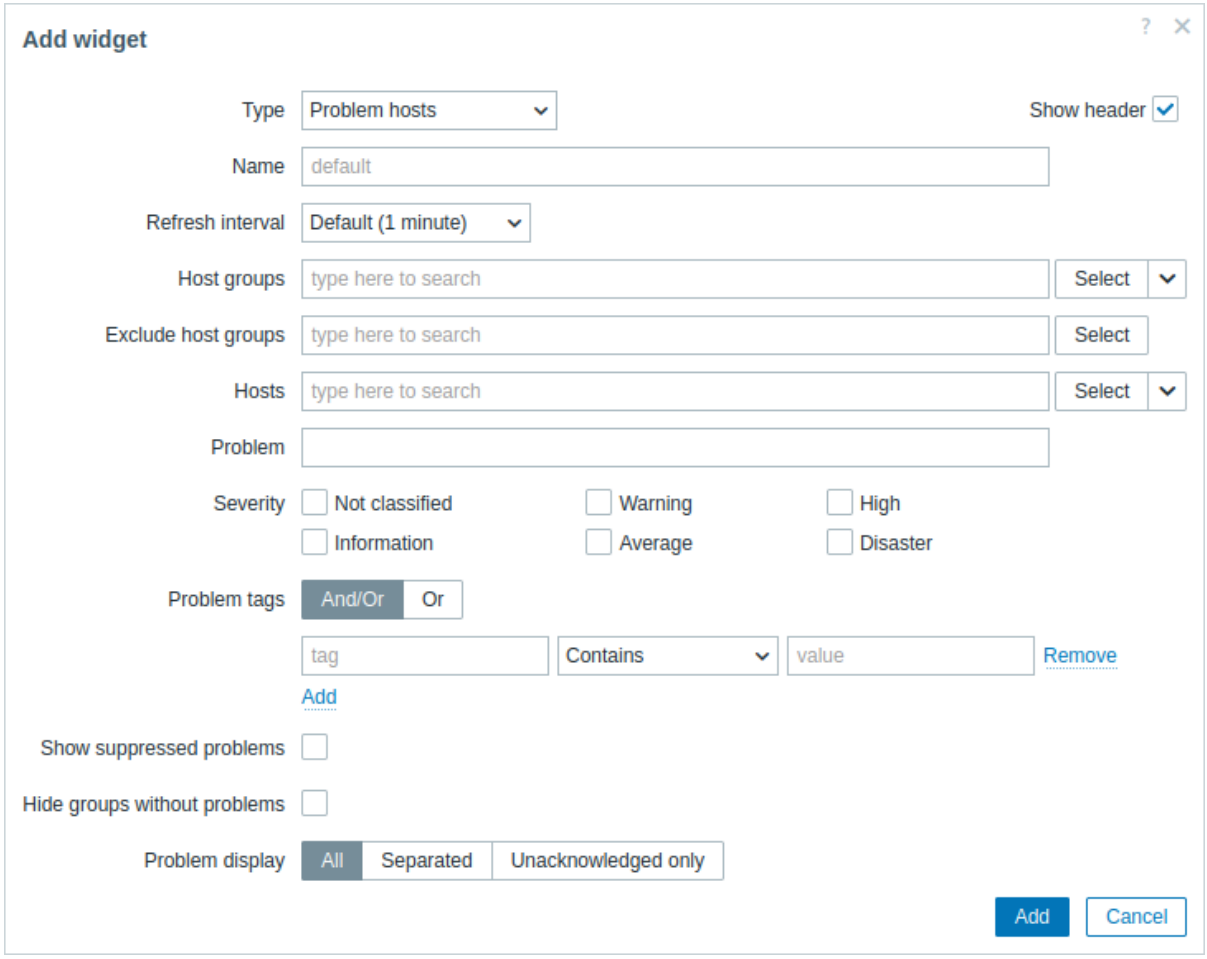


障害数にカーソルを合わせると、障害が発生しているホストのリストが表示されます:



設定

設定するには、タイプとして 問題のあるホストを選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

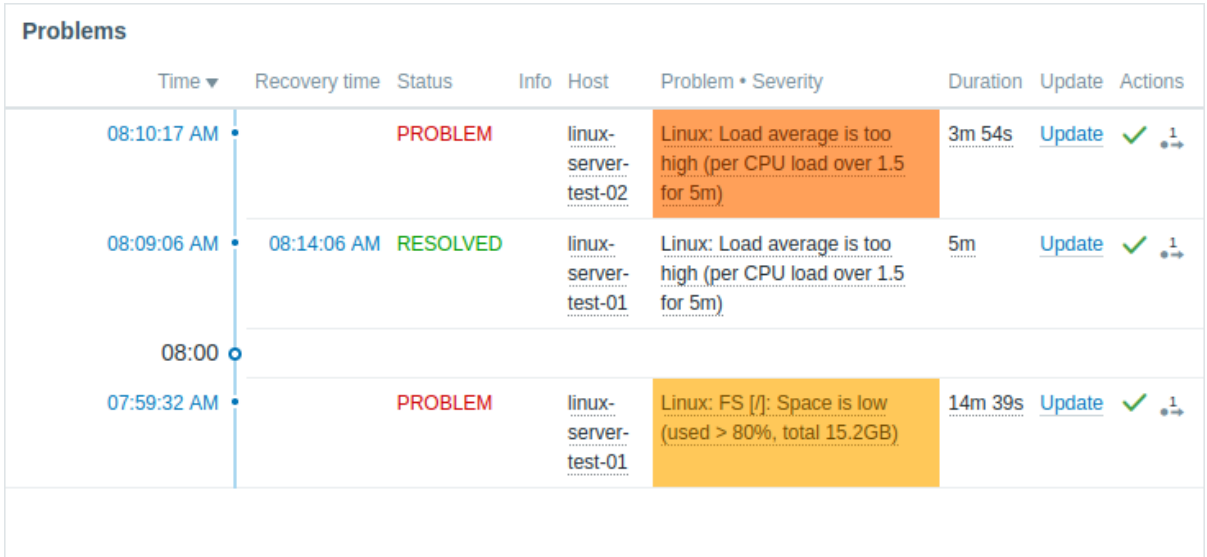
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストグループの除外	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータはウィジェットに表示されません。例えば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

問題	問題名で表示する問題ホストの数を制限できます。 ここに文字列を入力すると、名前に入力した文字列が含まれる問題を持つホストのみが表示されます。 マクロは展開されません。
深刻度	ウィジェットに表示する問題をフィルタリングするために、問題の深刻度を選択します。 深刻度を選択しない場合は、すべての問題が表示されます。
問題タグ	ウィジェットに表示する問題の数を制限するために、問題タグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
抑制された問題を表示	ホストのメンテナンスのために抑制（非表示）される問題も表示するには、チェックボックスをオンにします。
問題のないグループを非表示	問題のないグループを非表示オプションを選択すると、ウィジェットで問題のないホストグループのデータが非表示になります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
問題の表示	問題数の表示方法: すべて - 問題の総数を表示; 分離 - 未確認の問題数を、問題の総数の数値として分離して表示; 未確認のみ - 未確認の問題数のみを表示;

23 問題

概要

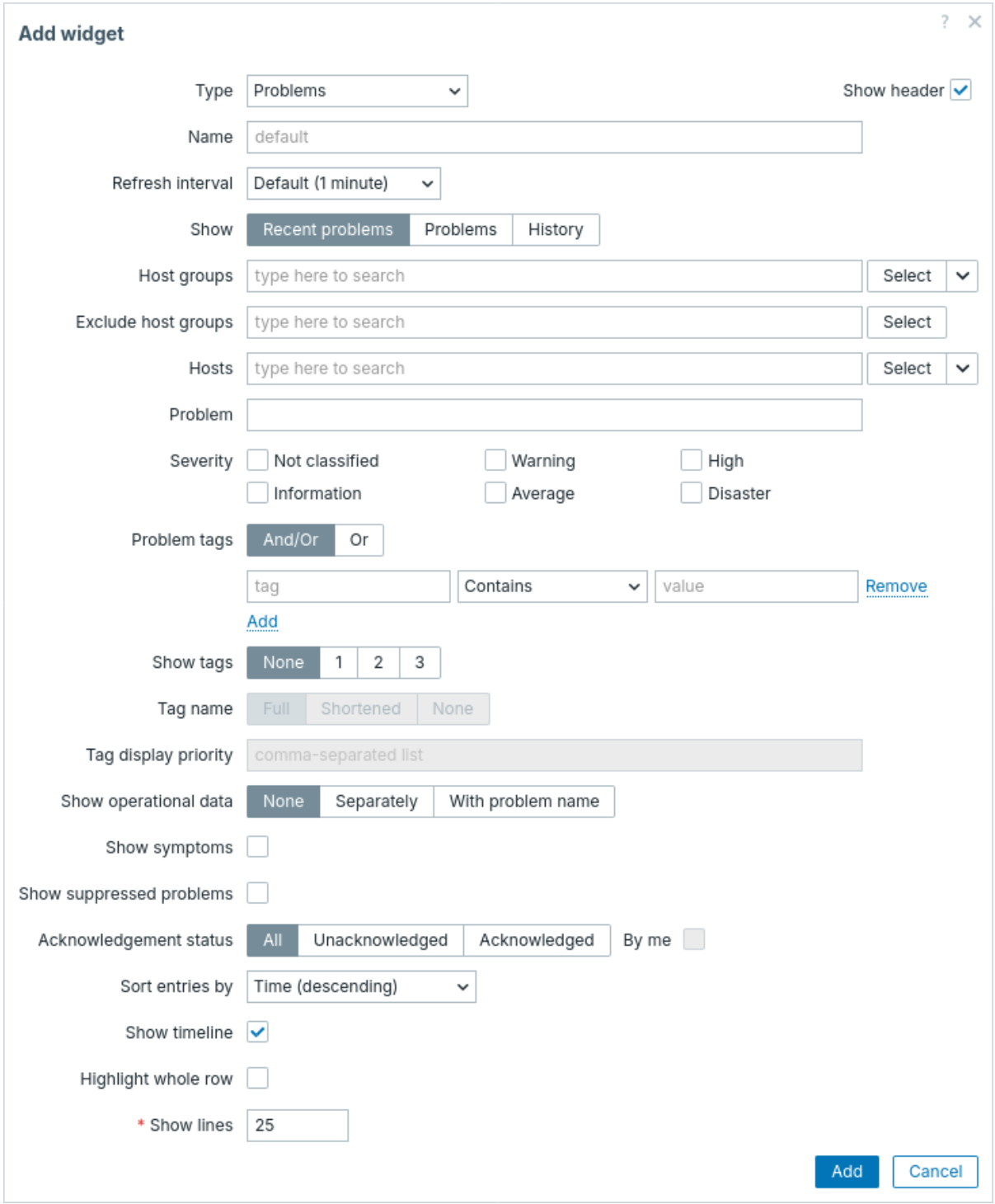
問題ウィジェットは、さまざまなパラメータ（ホストグループ、ホスト、問題名など）でフィルタリングされた現在の問題を表示し、注意が必要な内容を明確に把握できます。



監視 > 問題と同じデータを表示し、最大 1000 件まで表示できます。

設定

設定するには、タイプとして 障害を選択します:



ウィジェットに表示される障害の数は、障害のステータス、障害名、深刻度、ホストグループ、ホスト、イベントタグ、確認ステータスなど、さまざまな方法で制限できます。

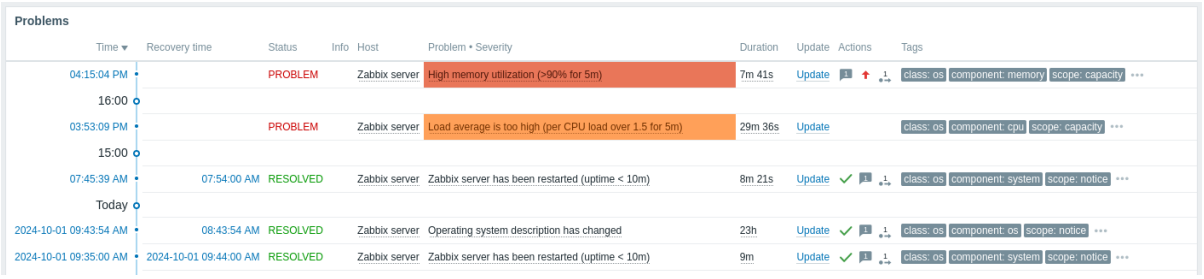
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

表示	障害のステータスでフィルタ: 最近の障害 - 未解決および最近解決された障害が表示されます（デフォルト）; 障害 - 未解決の障害が表示されます; 履歴 - すべてのイベントの履歴が表示されます。

ホストグループ	ウィジェットに表示する障害のホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループの障害がウィジェットに表示されます。ホストグループが入力されていない場合、すべてのホストグループの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストグループの除外	ウィジェットから障害を非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループの障害はウィジェットに表示されません。例えば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、表示にグループ A、除外にグループ B を同時に選択すると、ホスト 001 の障害のみがウィジェットに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示する障害のホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応なので、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストが入力されていない場合、すべてのホストの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
障害	障害名で表示される障害の数を制限できます。 ここに文字列を入力すると、入力した文字列を含む障害名のみが表示されます。 マクロは展開されません。
深刻度	障害の深刻度をマークして、ウィジェットに表示する障害をフィルタします。 深刻度がマークされていない場合、すべての障害が表示されます。
障害タグ	ウィジェットに表示される障害の数を制限するために障害タグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 フィルタリング時、ここで指定したタグは、タグ表示優先度（下記）リストで上書きされない限り、障害とともに最初に表示されます。
タグの表示	表示するタグの数を選択します: なし - タグ列なし; **1** - タグ列に 1 つのタグ; **2** - タグ列に 2 つのタグ; **3** - タグ列に 3 つのタグ。 障害のすべてのタグを表示するには、三点リーダーアイコンにマウスを重ねてください。
タグ名	タグ名の表示モードを選択します: フル - タグ名と値をすべて表示; 短縮 - タグ名を 3 文字に短縮して表示し、タグ値はすべて表示; なし - タグ値のみを表示し、名前は表示しません。

タグ表示優先度	障害のタグ表示優先度をカンマ区切りのタグリストとして入力します。 値は使用せず、タグ名のみを使用してください。 例: `Services,Applications,Application` このリストのタグは常に最初に表示され、アルファベット順の自然な並び順を上書きします。
運用データの表示	運用データの表示モードを選択します: なし - 運用データは表示されません; 別列 - 運用データを別の列に表示; 障害名と一緒に - 運用データを障害名に括弧付きで追加表示。
症状の表示	症状として分類された障害を独自の行で表示するには、チェックボックスをマークします。
抑制された障害の表示	ホストのメンテナンスや単一の障害抑制のために抑制（非表示）される障害を表示するには、チェックボックスをマークします。
確認ステータス	すべての障害、未確認の障害のみ、または確認済みの障害のみを表示するようにフィルタします。 追加のチェックボックスをマークすると、自分が一度でも確認した障害を除外できます。
並び替え	以下で並び替え: 時刻（降順または昇順）; 深刻度（降順または昇順）; 障害名（降順または昇順）; ホスト（降順または昇順）。 ホスト（降順または昇順）での並び替えは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
タイムラインの表示	ビジュアルタイムラインを表示するには、チェックボックスをマークします。
行全体のハイライト	未解決の障害の行全体をハイライトするには、チェックボックスをマークします。障害の深刻度の色がハイライトに使用されます。 行全体のハイライトはハイコントラストテーマでは利用できません。
表示行数	表示する障害行数を指定します。

ウィジェットの使用

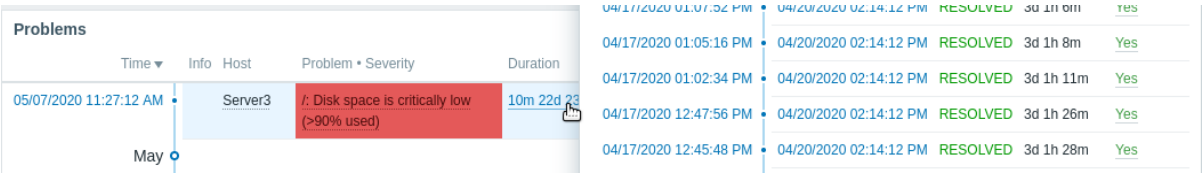


障害ウィジェットは追加情報へのクイックアクセスを提供します:

- 障害発生日時 (Time 列) または復旧日時 (Recovery Time 列) をクリックすると、イベントの詳細を表示します。
- Info フィールドが空でない場合、表示されたアイコンにマウスを重ねると追加の詳細が表示されます。
- ホスト名をクリックすると、ホストメニューが開きます。
- 障害名をクリックすると、イベントメニューが開きます。
- 障害の継続時間にマウスを重ねるかクリックすると、障害イベントポップアップが表示されます。
- Update を押すと、障害の更新ウィンドウが開きます。
- Actions 列のグレーの矢印アイコンにマウスを重ねるか押すと、実行されたアクションの一覧が表示されます。

障害イベントポップアップ

障害イベントポップアップには、このトリガーの障害イベントのリストと、定義されていればトリガーの説明やクリック可能な URL が含まれます。



障害イベントポップアップを表示するには:

- *Problems* ウィジェットの *Duration* 列の障害継続時間にマウスを重ねます。マウスを継続時間から外すとポップアップは消えます。

- *Problems* ウィジェットの *Duration* 列の継続時間をクリックします。再度継続時間をクリックしない限り、ポップアップは消えません。

24 問題の深刻度別

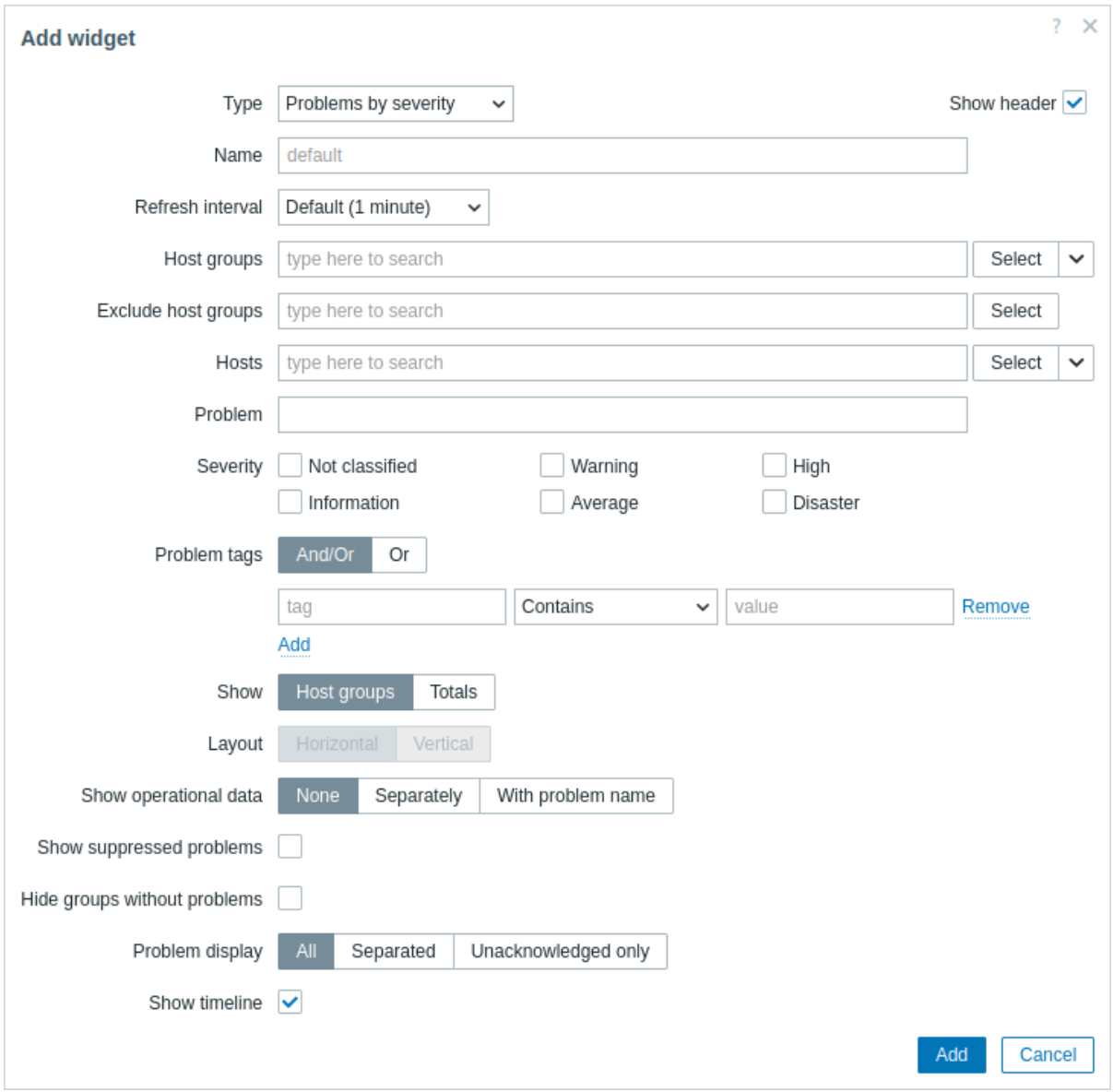
概要

重大度別の問題ウィジェットは、重大度ごとにグループ化された原因となる問題の数を表示します。さまざまなパラメータ（ホストグループ、ホスト、問題名など）で問題をフィルタリングできます。

問題数にカーソルを合わせると、関連する問題のリストが表示されます:

設定

設定するには、タイプとして 深刻度別の障害を選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

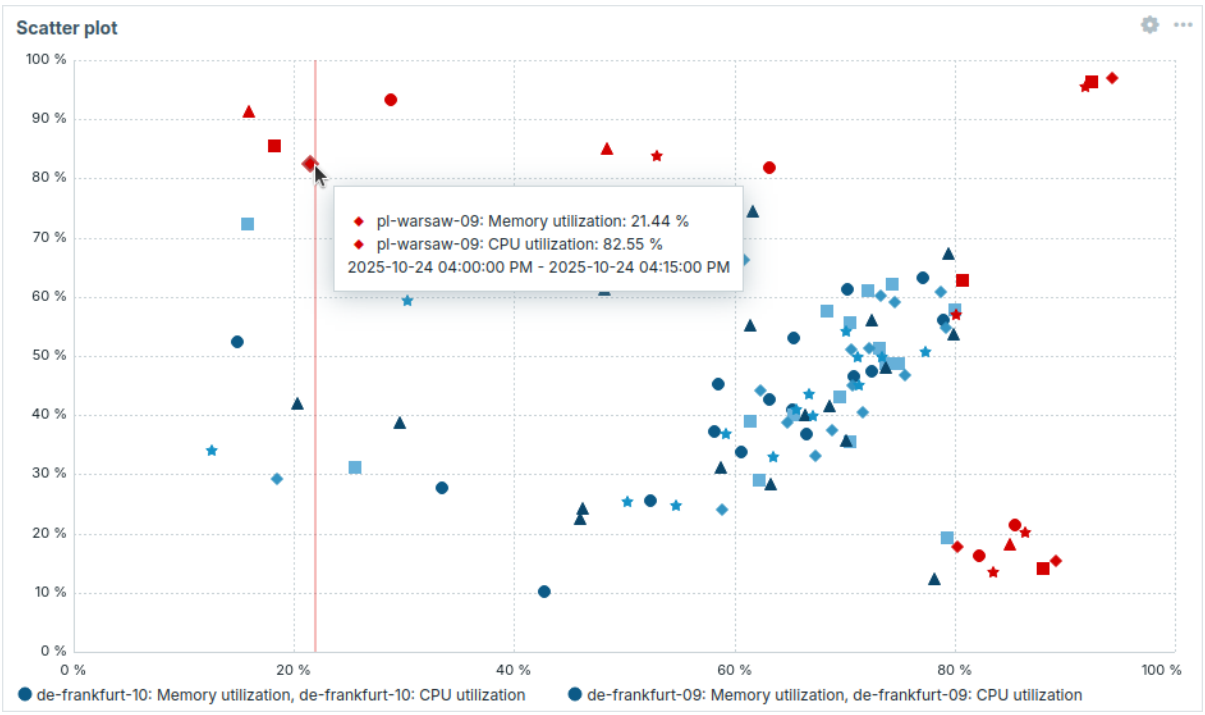
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストグループの除外	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループのホストデータはウィジェットに表示されません。例えば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
障害	障害名で表示する障害ホストの数を制限できます。 ここに文字列を入力すると、入力した文字列を名前に含む障害を持つホストのみが表示されます。 マクロは展開されません。
深刻度	ウィジェットに表示する障害をフィルタリングするために障害の深刻度をマークします。 深刻度をマークしない場合は、すべての障害が表示されます。
障害タグ	ウィジェットに表示する障害の数を制限するための障害タグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
表示	表示オプションを選択します: ホストグループ - ホストグループごとに障害を表示; 合計 - 問題の深刻度に対応する色付きブロックで、選択したすべてのホストグループの障害の合計を表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できず、障害の合計のみが表示されます。
レイアウト	レイアウトオプションを選択します: 水平 - 合計の色付きブロックが水平に表示されます; 垂直 - 合計の色付きブロックが垂直に表示されます。 このパラメータは表示が「合計」に設定されている場合に利用できます。
運用データの表示	運用データ（監視 > 障害の運用データの説明を参照）を以下のように表示します: なし - 運用データは表示されません 個別に - 運用データが個別の行として表示されます 障害名と一緒に - 運用データが障害名と一緒に表示されます。

抑制された障害の表示	ホストのメンテナンスにより抑制（非表示）される障害も表示するには、チェックボックスをマークします。
障害のないグループを非表示	ウィジェットで障害のないホストグループのデータを非表示にするには、障害のないグループを非表示オプションをマークします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
障害の表示	障害数を以下のように表示します: すべて - 全障害数を表示; 分離 - 未確認の障害数を全障害数の数として分離して表示; 未確認のみ - 未確認の障害数のみを表示。
タイムラインの表示	視覚的なタイムラインを表示するには、チェックボックスをマークします。

25 散布図

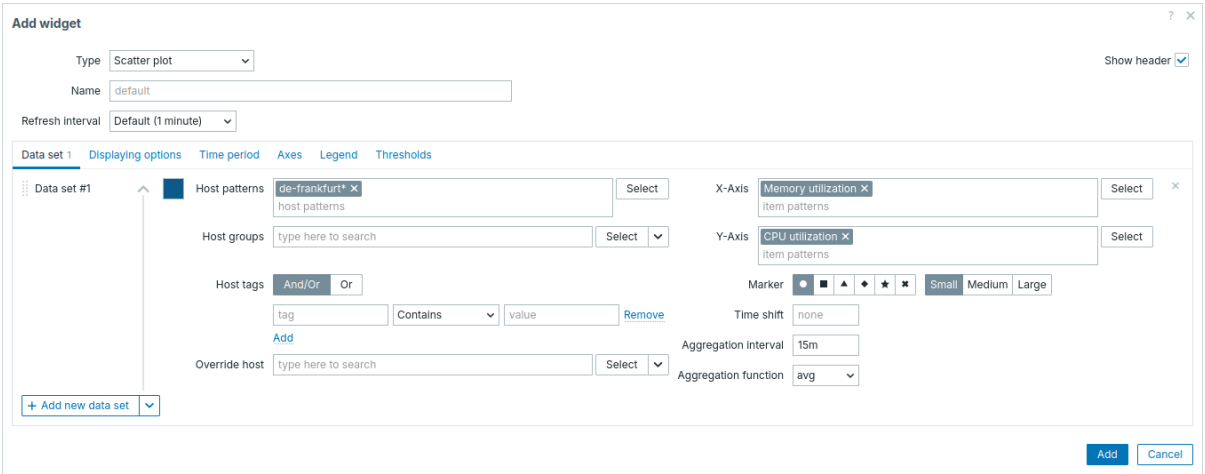
概要

散布図ウィジェットは、X 軸と Y 軸に沿って個々のデータポイントをプロットすることで、2 つのメトリクス間の関係を表示します。これにより、データセット内のパターン、クラスター、相関関係、および外れ値を明らかにするのに役立ちます。



設定

設定するには、タイプとして散布図を選択します:



データセット

データセットタブでは、データセットを追加して散布図のデータを選択できます。2 種類のデータセットを追加できます:

- アイテムパターン - 一致するアイテムを持つホストがデータポイントとして表示されます。単一の基本色を選択するか、パレット行を選択して一致するアイテムを持つ各ホストに個別の色を割り当てることができます。
- アイテムリスト - 選択したアイテムがデータポイントとして表示され、各ポイントはアイテムのペア (X と Y) を表します。各データセットの色はピッカーから個別に選択できます。

両軸に少なくとも 1 つの値があるホスト/アイテムのペアに対してのみ、データポイントが作成されます。

デフォルトでは、アイテムパターンのデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストとアイテムのパターンを選択または入力します。これらのパターンに一致するアイテムのデータが散布図に表示されます。最大 100 個のアイテムが表示されます (各軸 50 個、合計 50 ホスト)。 ワイルドカードパターンを選択に使用できます (例: * は 0 文字以上に一致する結果を返します)。 ワイルドカードパターンを指定するには、文字列を手動で入力し、*Enter* を押します。 ワイルドカード記号は常に解釈されるため、例えば *item2* や *item3* など他に一致するアイテムがある場合、*item** という名前のアイテムを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストとアイテムのパターンの指定が必須です。 詳細はデータセット設定の詳細も参照してください。 アイテムリストデータセットの場合: アイテムの追加ボタンをクリックして、散布図のアイテムを選択します。 また、ウィジェットの追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択することもできます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットの指定が必須です。 詳細はデータセット設定の詳細も参照してください。 数値型のアイテムのみが許可されていることに注意してください。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターン指定用のパラメータは利用できず、アイテムリスト指定用のパラメータではテンプレートで設定されたアイテムのみ選択できます。
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータはアイテムパターンデータセットでのみ利用可能で、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

ホストタグ	ウィジェットに表示するホストをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子が利用できます: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める (大文字小文字を区別); 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字小文字を区別しない); 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する (大文字小文字を区別); 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字小文字を区別しない)。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化される; **Or** - いずれか 1 つの条件を満たせば十分。 このパラメータはアイテムパターンデータセットでのみ利用可能で、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストの上書き	ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクタを選択します。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
マーカー	データセット内のホストデータポイントのマーカータイプとサイズを選択します。
タイムシフト	必要に応じてタイムシフトを指定します。 このフィールドでは時間のサフィックスを使用できます。負の値も許可されます。
集計関数	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **sum** - 合計値を表示; **count** - 値の個数を表示 (両軸に少なくとも 1 つの値がある場合); **first** - 最初の値を表示; **last** - 最後の値を表示; **none** - すべての値を表示 (集計なし)。 集計を使用すると、すべての値の代わりに選択した間隔 (5 分、1 時間、1 日) の集計値を表示できます。詳細はグラフの集計も参照してください。
集計間隔	値を集計する間隔を指定します。 このフィールドでは時間のサフィックスを使用できます。サフィックスのない数値は秒と見なされます。 ウィジェットがトレンドに基づいて履歴データを表示するように設定されている場合 (履歴データの選択がトレンドまたは自動に設定されている場合)、集計間隔は 1 時間の倍数 (例: 3600, 60m, 1h, 3h など) を使用することを推奨します。トレンドは 1 時間ごとの集計値を保存するため、1 時間の倍数でない集計間隔 (例: 100s, 7min, 15min, 90min など) を使用すると、解釈が難しい結果になる場合があります。 集計間隔の指定は必須です。 集計間隔はデータポイントの時間間隔と座標を決定します。例えば、X 軸アイテムに 4 つの値、Y 軸アイテムに 1 つの値がある場合、座標は指定した間隔で X を 4 つ、Y を 1 つ集計したものになります。

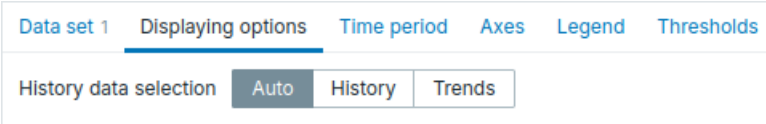
データセットの設定の詳細

既存のデータセットはリストに表示されます。これらのデータセットは、並べ替え、展開/折りたたみ、色の変更、クローン作成が可能です。

詳細については、グラフウィジェットのデータセット設定の詳細を参照してください。これらの詳細は散布図ウィジェットにも適用されます。

表示オプション

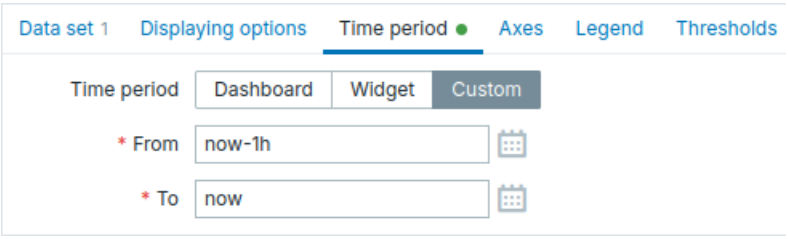
表示オプションタブでは、履歴データの選択を定義できます。



履歴データの選択	散布図データのソースを設定します: 自動 - クラシックグラフのアルゴリズムに従ってデータが取得されます（デフォルト）; 履歴 - 履歴からデータを取得; トレンド - トレンドからデータを取得。

期間

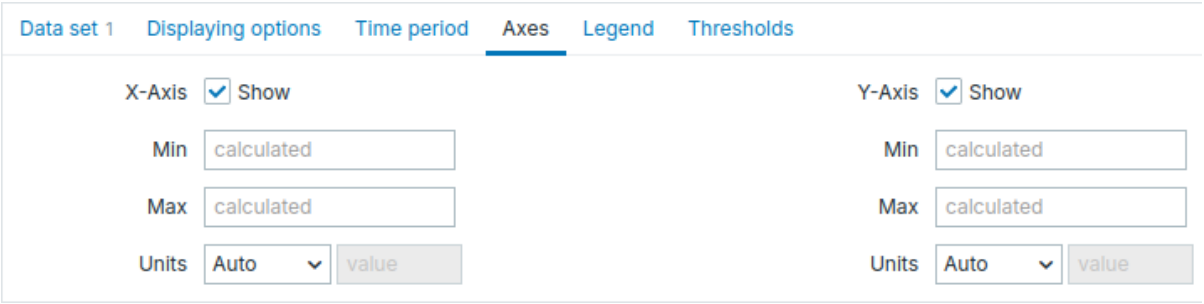
期間タブでは、散布図でデータを表示する期間を設定できます。



期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用します; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します; カスタム - *From* および *To* パラメータで設定されたカスタム期間を使用します。設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。

軸

軸タブでは、軸の表示方法をカスタマイズできます。

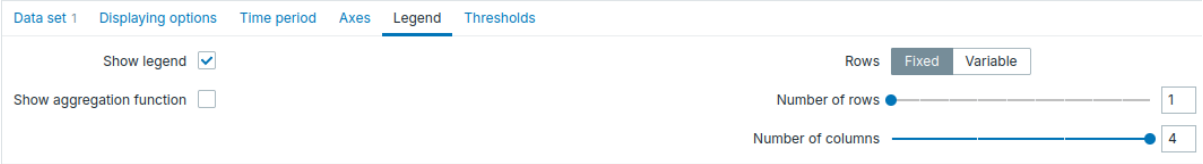


X 軸/*Y* 軸	このチェックボックスのチェックを外すと、X 軸/Y 軸が非表示になります（デフォルトでチェックされています）。
最小	対応する軸の最小値を設定します。
最大	対応する軸の最大値を設定します。

単位	散布図の軸値の単位をドロップダウンから選択します： 自動 - 軸値はデータセット内の最初のアイテムの単位で表示されます； 静的 - 軸値は値入力フィールドで指定された単位で表示されます。フィールドが空白の場合は数値のみが表示されます。

凡例

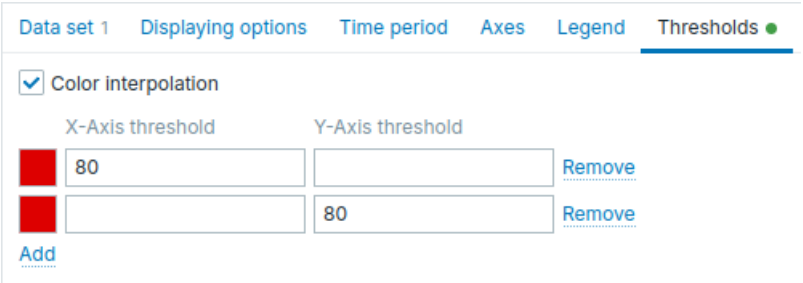
凡例タブでは、散布図の凡例をカスタマイズできます。



凡例を表示	このチェックボックスのチェックを外すと、散布図の凡例が非表示になります（デフォルトでチェックされています）。
集計関数を表示	このチェックボックスをオンにすると、凡例に集計関数が表示されます。
行数	凡例の行の表示モードを選択します： 固定 - 表示される行数は行数パラメータの値で決まります。 可変 - 表示される行数は設定されたアイテム数によって決まり、最大行数パラメータの値を超えません。
行数/ 最大行数	行数が「固定」の場合、表示する凡例の行数を設定します（1～10）。 行数が「可変」の場合、表示する凡例の最大行数を設定します（1～10）。
列数	表示する凡例の列数を設定します（1～4）。 このパラメータは、最小/平均/最大を表示がオフの場合に利用できます。

しきい値

しきい値タブでは、ホストマーカーの色を決定するしきい値を設定できます。



色の補間	チェックボックスをオンにすると、ホストマーカーのしきい値の色間で滑らかに遷移します。 このパラメータは、X 軸、Y 軸、または両方の軸に 2 つ以上のしきい値が設定されている場合に適用されます。
X 軸/*Y* 軸のしきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値は、単一の軸、両方の軸それぞれ、または両方の軸同時に適用できます。 たとえば、X=80、Y=80 の単一のしきい値は、両方の条件が満たされた場合にマーカーに色を付けます。 しきい値のリストは、保存時に昇順でソートされます。 サフィックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています。値のマッピングもサポートされています。

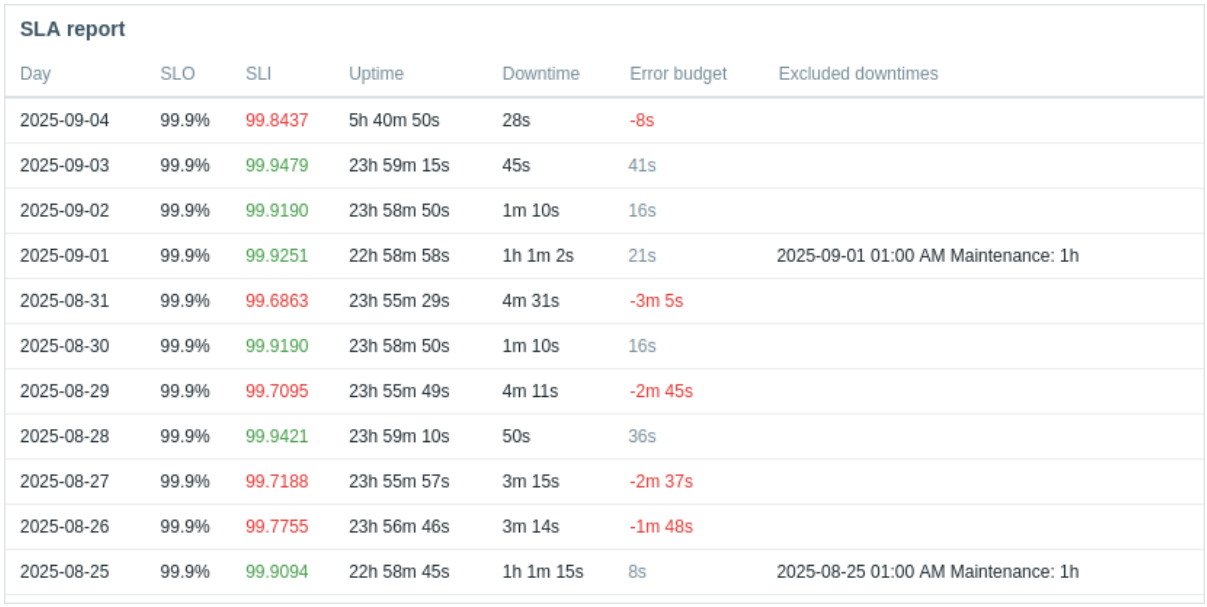
散布図ウィジェットで表示される情報は、ウィジェットメニューを使用して*.png* 画像としてダウンロードできます。

ウィジェットのスクリーンショットは *Downloads* フォルダーに保存されます。

26 SLA レポート

概要

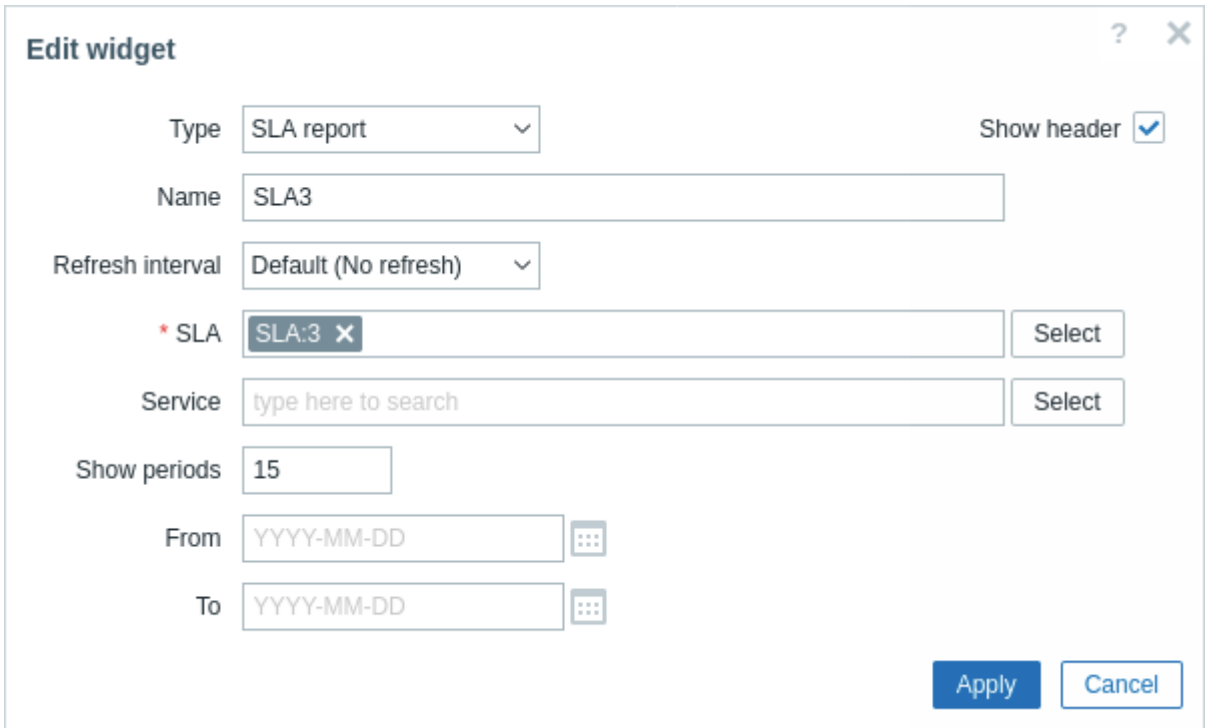
SLA レポートウィジェットは、SLA レポートを表示し、定義された目標に対するサービスのパフォーマンスを監視するのに役立ちます。



サービス > *SLA* レポートと同じデータを表示します。

設定

設定するには、タイプとして *SLA* レポートを選択します:



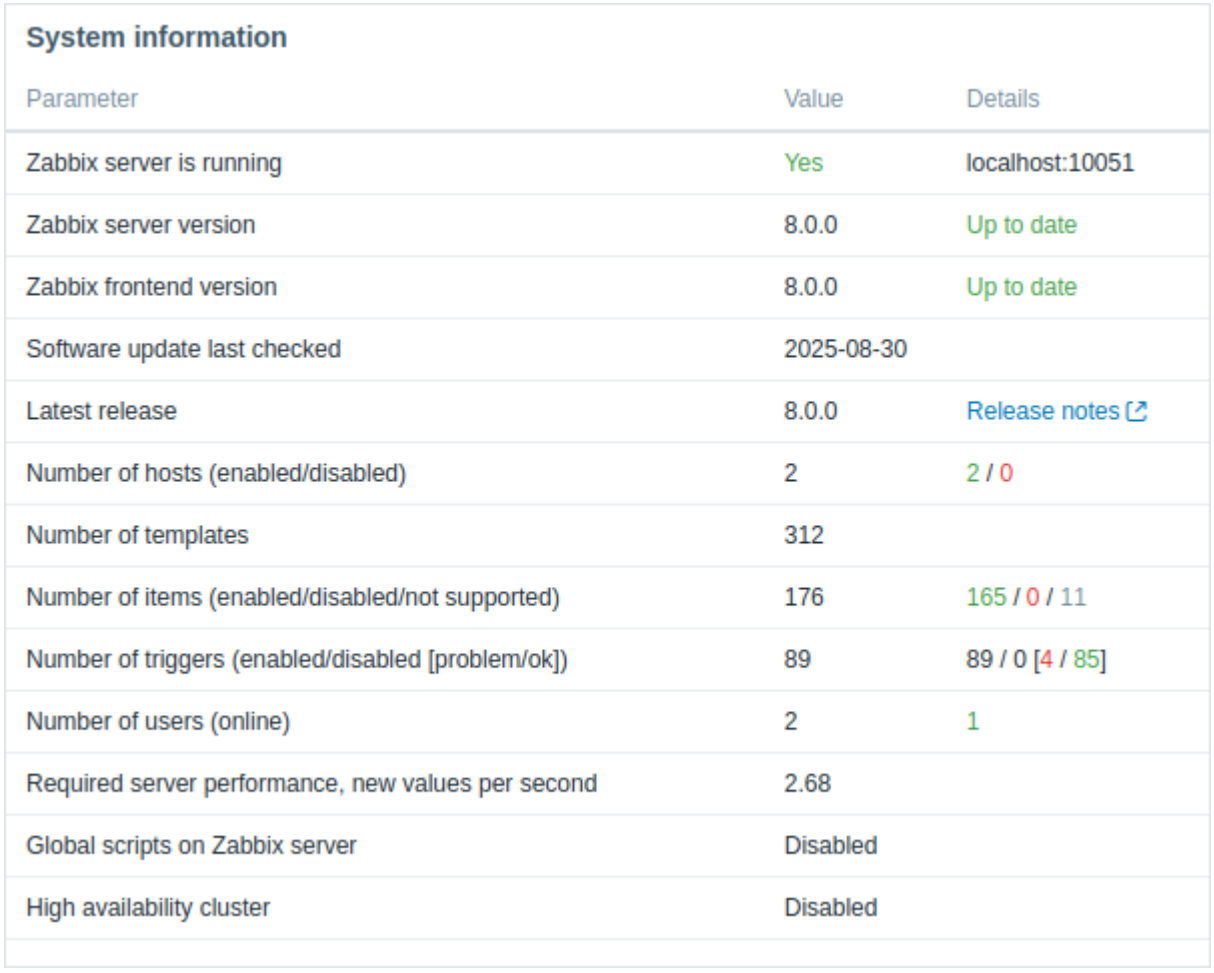
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

SLA	レポートの SLA を選択します。
サービス	レポートのサービスを選択します。
期間の表示	ウィジェットに表示する期間の数を設定します (デフォルトは 20、最大 100)。
開始	レポートの開始日を選択します。 相対日付がサポートされています: `now`, `now/d`, `now/w-1w` など; サポートされている日付修飾子: d, w, M, y。
終了	レポートの終了日を選択します。 相対日付がサポートされています: `now`, `now/d`, `now/w-1w` など; サポートされている日付修飾子: d, w, M, y。

27 システム情報

概要

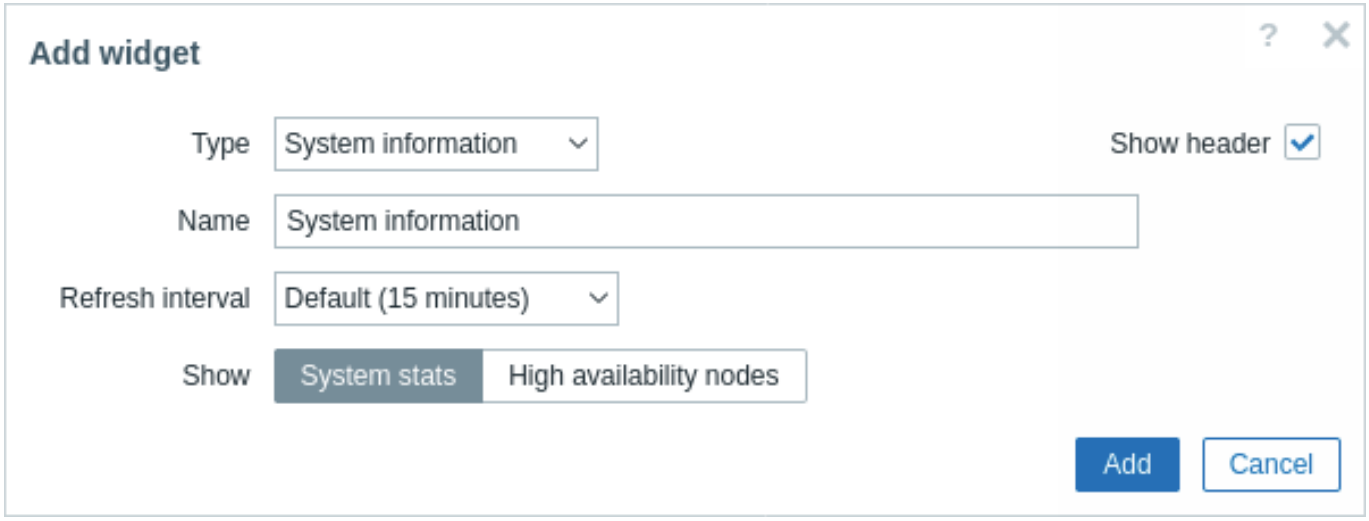
システム情報ウィジェットは、主要な Zabbix サーバーおよびシステムデータの概要、または高可用性ノードの詳細を表示します。



レポート > システム情報と同じデータが表示されます。

設定

設定するにはタイプとしてシステム情報を選択します。



すべてのウィジェットの共通パラメーターに加えて、次の特定のオプションを設定できます。

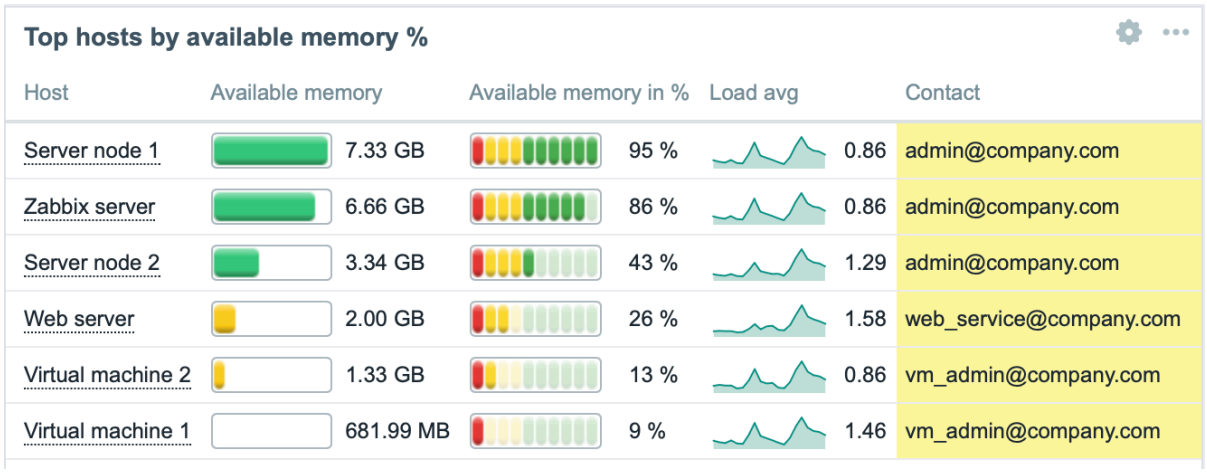
表示	表示する内容を選択します: システムステータス - 主要な Zabbix サーバーとシステムデータの概要を表示します。 **HA** ノード - 高可用性ノードのステータスを表示します (高可用性クラスターが有効な場合)。

ソフトウェアアップデート確認の詳細を表示	Zabbix ソフトウェアアップデート確認の詳細を表示するには、チェックボックスをオンにします。このオプションは、Zabbix サーバー設定でソフトウェアアップデート確認が有効になっており、表示フィールドで” システム統計ステータス” が選択されている場合にのみ使用できます。

28 トップホスト

概要

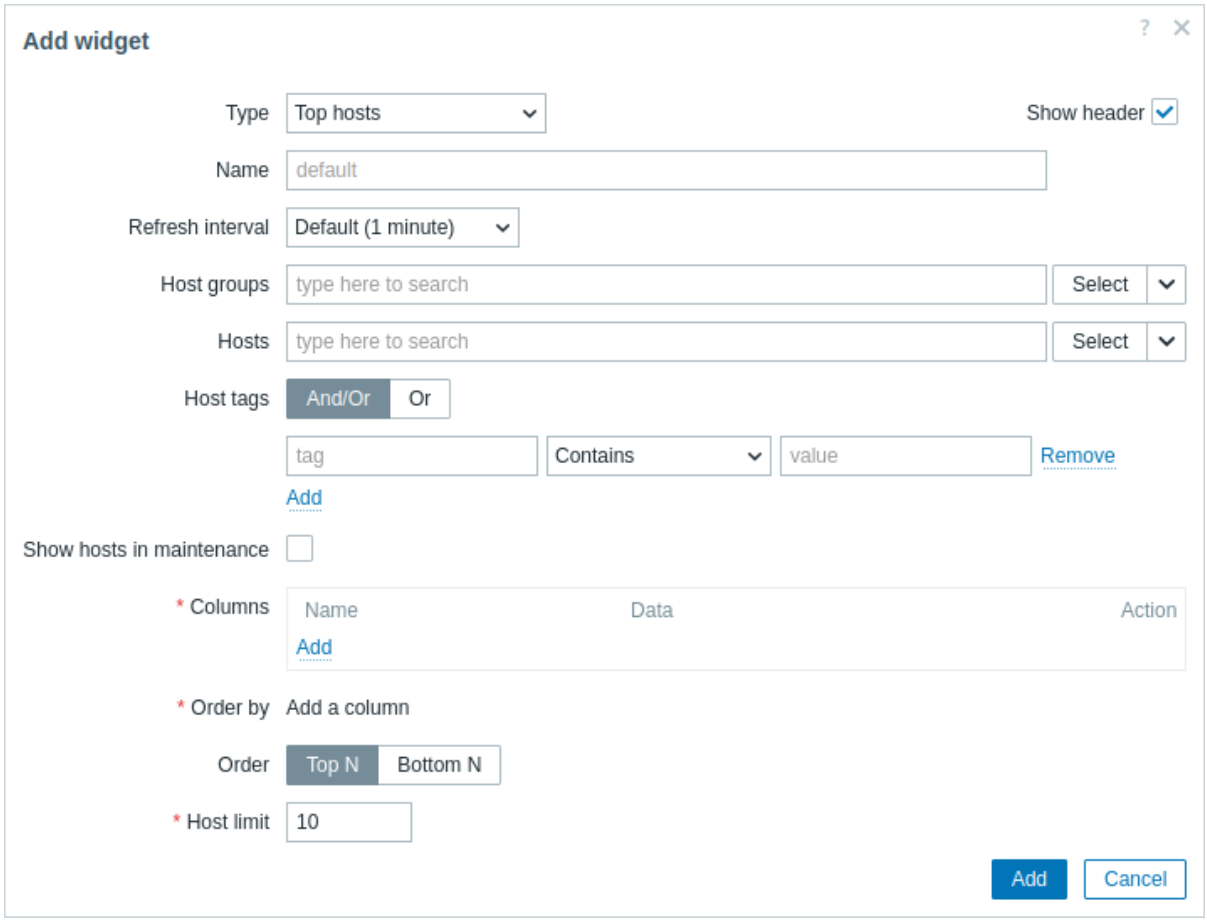
Top hosts ウィジェットは、複数のホストから選択したアイテムのトップ/ボトムの値を表示します。



このウィジェットは、最大 1000 台のホストを同時に表示することができます。

設定

設定するにはタイプとして上位ホストを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

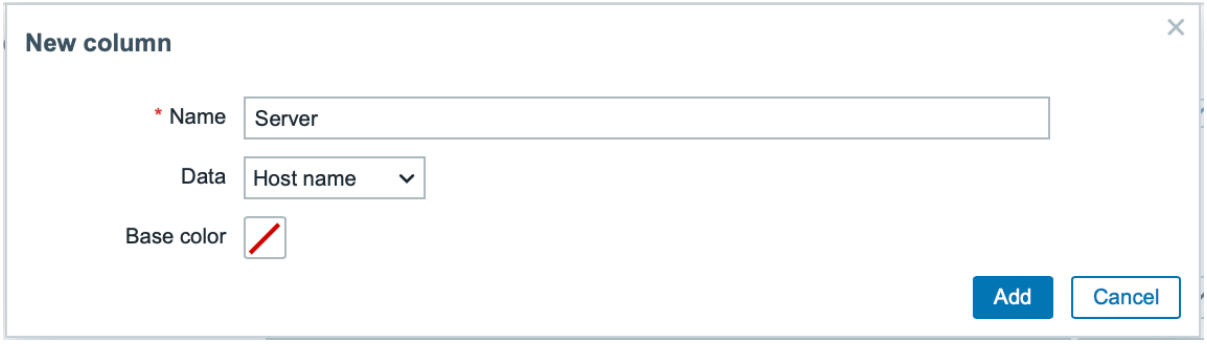
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストタグ	ウィジェットに表示するホスト数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字・小文字を区別します。 各条件にはいくつかの演算子が利用できます: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を含める; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を除外する。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
メンテナンス中のホストを表示	メンテナンス中のホストも表示する場合はこのチェックボックスをオンにします（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。デフォルトではオフです。
カラム	表示するデータのカラムを追加します。 カラムの順序は、左から右への表示順を決定します。 カラムはカラム名の前のハンドルをドラッグして上下に並べ替えることができます。
並び替え	*Top N* または *Bottom N* の並び替えに使用するカラムリストからカラムを指定します。
順序	行の並び順を指定します: **Top N** - 並び替えで集計した値の降順; **Bottom N** - 並び替えで集計した値の昇順。
ホストの上限	表示するホスト行数（1～1000）。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。

カラムの設定

3 つのデータ型のカラムを追加できます: ホスト名、アイテム値、またはテキストです。利用可能なカラムパラメータのリストは、カラムのデータ型、およびアイテム値タイプの場合は値のフォーマットによって異なります。

ホスト名カラム

ホスト名カラムは、ホストの名前を表示するために使用されます。



サポートされているパラメータ:

名前	カラムの名前。
データ	カラムに表示するデータタイプ。ホスト名を選択します。
基本色	カラムの背景色。

テキストカラム

テキストカラムは、指定したテキスト文字列を表示するために使用します。

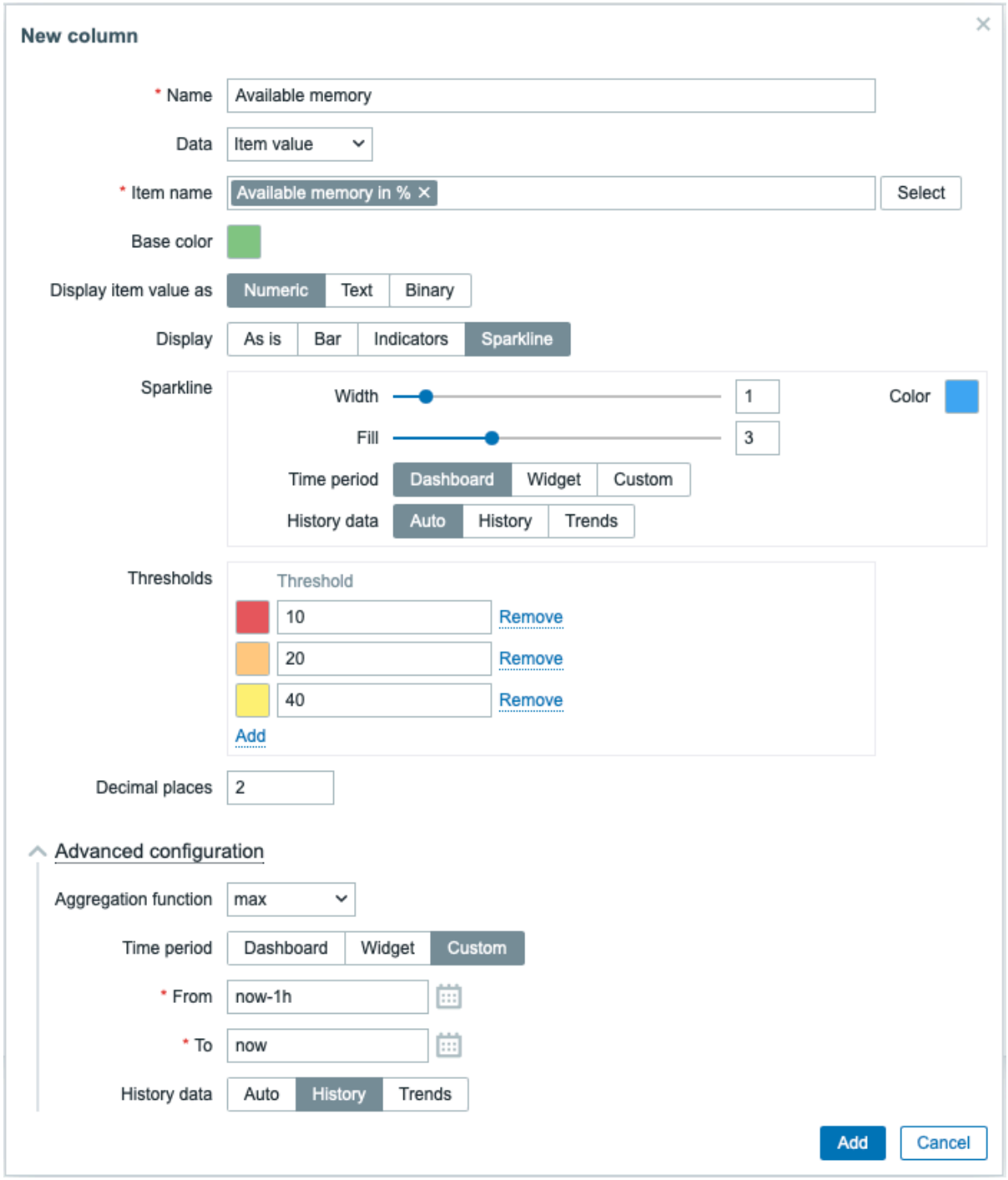


サポートされているパラメータ:

名前	カラムの名前。
データ	カラムに表示するデータ型。テキストを選択します。
基本色	カラムの背景色。
テキスト	表示する文字列を入力します。 ホストやインベントリのマクロを含めることができます。

アイテム値カラム

アイテム値カラムは、指定したアイテムの値を表示するために使用されます。複数の値表示形式とオプションをサポートしています。



サポートされているパラメータ:

Name	カラム名。
Data	カラムに表示するデータタイプ。**Item value** を選択します。
Item name	アイテムを選択します。その名前を使用して、選択したすべてのホストで同じ名前のアイテムを一致させて表示します。ホストに複数の該当アイテムがある場合、ウィジェットはキーがアルファベット順で最初のアイテムを表示します（例: `system.cpu.util` ではなく `proc.cpu.util` キーの *CPU utilization*）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
Base color	カラムの背景色。*Item value* データがバー/インジケーターとして表示される場合は塗りつぶし色。 デフォルト色は、アイテム値が指定したしきい値を超えた場合、カスタム色で上書きできます。
Display item value as	アイテム値の表示形式: **Numeric**、**Text**、**Binary**。 選択したオプションによって、利用可能な追加パラメータが決まります。各形式のパラメータリストは以下を参照してください。
Advanced configuration	

Aggregation function	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; **not used** - 最新値を表示（集計なし）。 集計を使用すると、最新値の代わりに選択した間隔（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* には数値データのみ表示できます。*count* の場合、非数値データは数値に変換されます。
Time period	値を集計する期間を指定します。期間のデータソースを選択します: **Dashboard** - ダッシュボードの期間セレクターを使用; **Widget** - 互換性のあるウィジェット（*Widget* パラメータで設定）を使用; **Custom** - *From* および *To* パラメータで設定したカスタム期間を使用。設定した場合、ウィジェットの右上に時計アイコンが表示され、マウスオーバーで設定期間が表示されます。 ウィジェットの *Time period* 設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。 このパラメータは、*Aggregation function* が”not used” の場合は利用できません。
Widget	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、*Time period* が”Widget” に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、*Time period* が”Custom” に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、*Time period* が”Custom” に設定されている場合に利用できます。
History data	ヒストリまたはトレンドからデータを取得します: **Auto** - 自動選択; **History** - ヒストリデータを取得; **Trends** - トレンドデータを取得。 このパラメータは数値アイテム値のみ利用可能です。テキストおよびバイナリ値の場合、常にヒストリからデータが取得されます。

数値アイテム値のパラメータ:

Display	値の表示方法を定義します: **As is** - 通常のテキストとして; **Bar** - ソリッドな色付きバーとして; **Indicators** - セグメント化された色付きバーとして; **Sparkline** - ミニ折れ線グラフとして。
Thresholds	背景/塗りつぶし色を変更するしきい値を指定します。 保存時にリストは昇順にソートされます。 スパークラインの場合、しきい値はアイテムの最新値のみに適用されます。
Decimal places	値に表示する小数点以下の桁数を指定します。
Bar/Indicators	
Min	最小値。
Max	最大値。
Sparkline	
Width	スライダーを使用するか、0～10 の範囲で値を手動入力してグラフ線の太さを設定します。
Color	線と塗りつぶしの色を選択します。
Fill	スライダーを使用するか、0～10 の範囲で値を手動入力して塗りつぶし色の透明度を設定します。

Time period	スパークラインチャートに含める値の期間を指定します。期間のデータソースを選択します: **Dashboard** - ダッシュボードの期間セレクターを使用; **Widget** - 互換性のあるウィジェット（*Widget* パラメータで設定）を使用; **Custom** - *From* および *To* パラメータで設定したカスタム期間を使用。設定した場合、ウィジェットの右上に時計アイコンが表示され、マウスオーバーで設定期間が表示されます。 ウィジェットの *Time period* 設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
Widget	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、*Time period* が”Widget” に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、*Time period* が”Custom” に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、*Time period* が”Custom” に設定されている場合に利用できます。
History data	ヒストリまたはトレンドからデータを取得します: **Auto** - 自動選択; **History** - ヒストリデータを取得; **Trends** - トレンドデータを取得。

テキストアイテム値のパラメータ:

Highlights	背景/塗りつぶし色を変更するために一致させる正規表現を指定します。

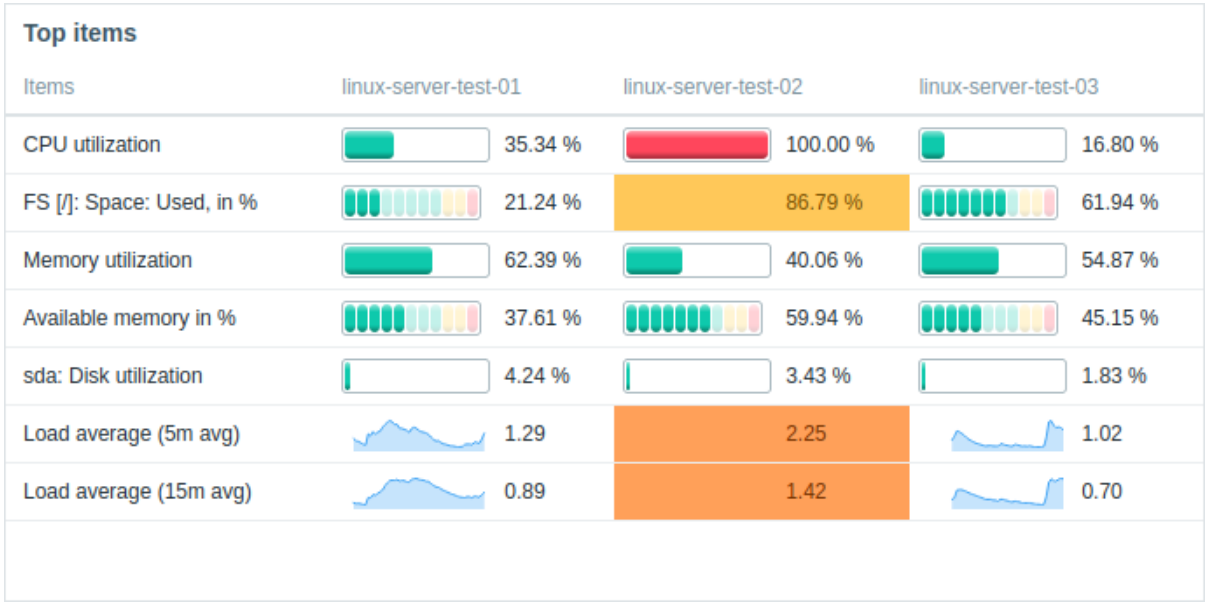
バイナリアイテム値のパラメータ:

Show thumbnail	バイナリデータを含む画像のサムネイルを作成して表示するか、値カラムにフルサイズ画像へのハイパーリンク *Show* を表示するかを指定します。

29 トップアイテム

概要

Top items ウィジェットは、選択したアイテムのトップ/ボトム値を表示し、そのパフォーマンスの概要を素早く把握できます。

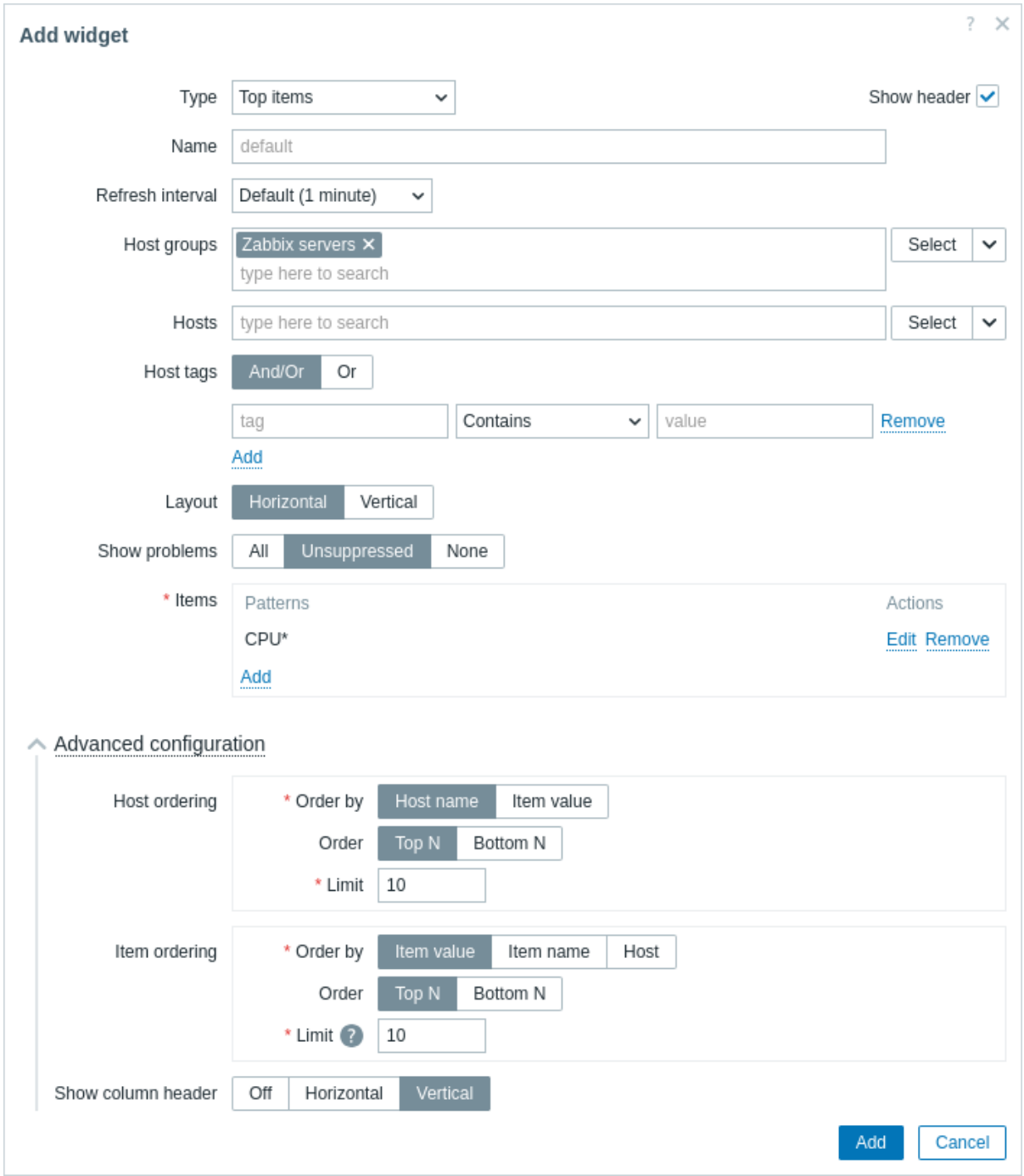


デフォルトでは、ウィジェットは直近 24 時間の値を表示します。この期間は、管理 > 一般 > *GUI*（最大履歴表示期間）で変更できます。

アイテム値をクリックすると、アイテムメニューが開きます。

設定

設定するには、タイプとして *Top items* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

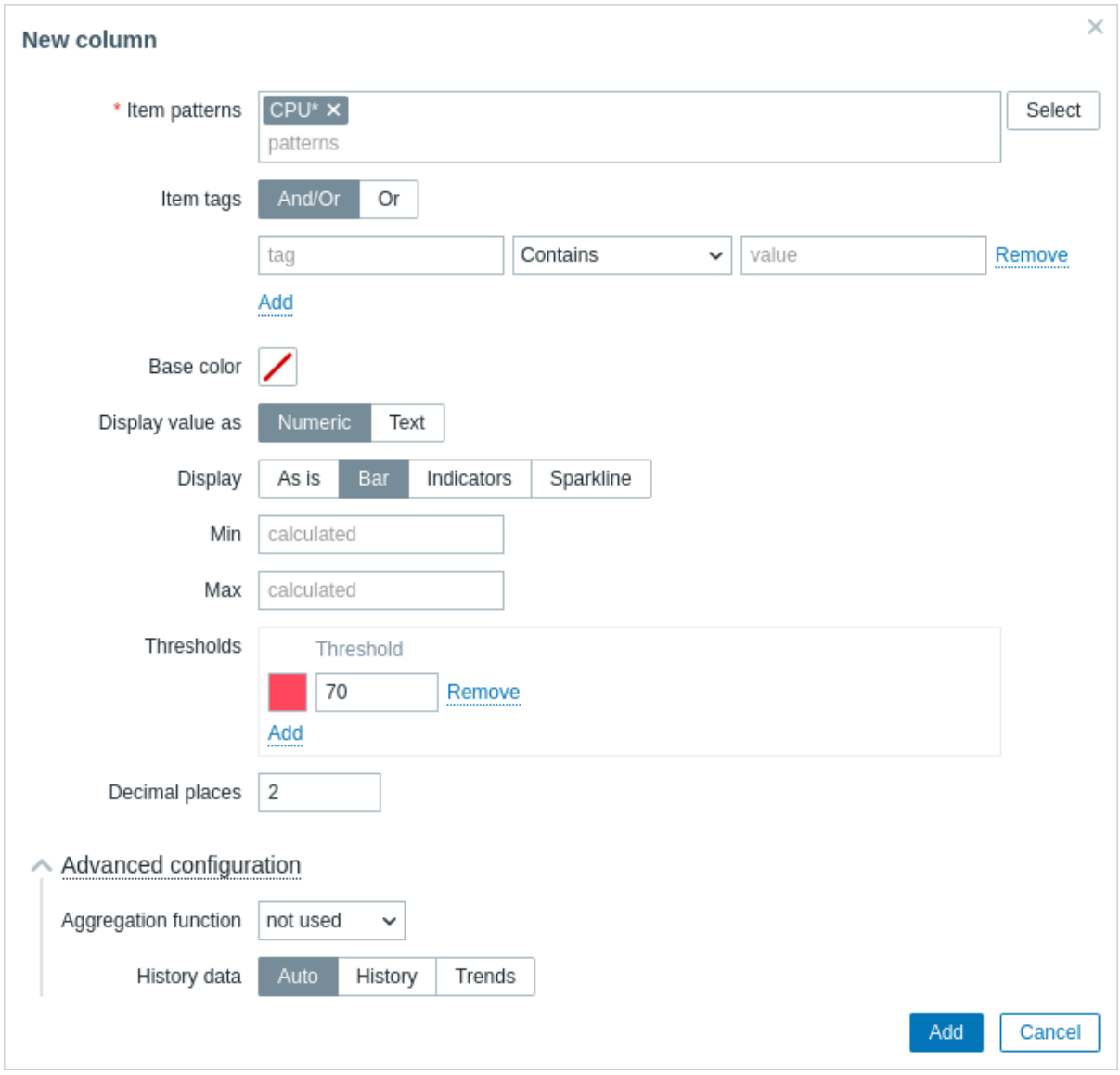
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートなので、グループ名の入力を開始すると一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートなので、ホスト名の入力を開始すると一致するホストのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

ホストタグ		ウィジェットに表示されるホストデータの数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
レイアウト		レイアウトオプションを選択します: 水平 - ホスト名が左側に表示されます; 垂直 - ホスト名が上部に表示されます。
障害の表示アイテム詳細設定		障害のステータス（すべて、抑制されていない、なし）に基づいて表示する障害をフィルタリングします。 障害は、アイテム値の色で示され、アイテム列の表示設定を上書きします。色は障害の深刻度に基づいており、管理 > 一般 > トリガー表示オプションでカスタマイズできます。障害の深刻度は、障害の更新時に変更できます。 表示するアイテムパターンまたは特定のアイテムを追加します（列の設定を参照）。
ホストの並び順		ホスト列/行の並び順オプションを選択します。
	並び替え基準	ホストの並び順: ホスト名 - ホスト名で並び替え; アイテム値 - 選択したアイテムの値で並び替え。
	順序	最も高い値または最も低い値のいずれかを表示するかを選択します: **TopN** - 上位 N 件; **BottomN** - 下位 N 件。 N の値は制限フィールドで選択します。
	制限	表示可能なホストの制限値を入力します（範囲 1-1000、デフォルト 10）。 この値が順序フィールドの N の値になります。
アイテムの並び順		アイテム列/行の並び順オプションを選択します。

	並び替え基準	アイテムの並び順: アイテム値 - アイテム値で並び替え; アイテム名 - アイテム名で並び替え; ホスト - 選択したホストパターンで並び替え。
	順序	最も高い値または最も低い値のいずれかを表示するかを選択します: **TopN** - 上位 N 件; **BottomN** - 下位 N 件。 N の値は制限フィールドで選択します。
	制限	表示可能なアイテムの制限値を入力します（範囲 1-1000、デフォルト 10）。 この値が順序フィールドの N の値になります。
列ヘッダーの表示		列ヘッダーの表示オプションを選択します: オフ - 列ヘッダーを表示しない; 水平 - ヘッダーにテキストを水平に表示; 垂直 - ヘッダーにテキストを垂直に表示。

カラムの設定

アイテムカラムを設定するには、アイテムパラメータで追加をクリックします。



共通のカラムパラメータ:

アイテムパターン	ワイルドカード文字を使用して、1 つまたは複数のアイテムパターンを指定します。 または、アイテムを選択します。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示されるアイテムデータの数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める (大文字・小文字を区別); 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字・小文字を区別しない); 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する (大文字・小文字を区別); 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字・小文字を区別しない)。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
ベースカラー	表示が「バー」または「インジケーター」に設定されている場合、カラムの背景色または塗りつぶし色を選択します。 ベースカラーはしきい値やハイライトカラーで上書きされる場合があります。
値の表示形式	アイテム値の表示形式: 数値またはテキスト。 選択したオプションによって、利用可能な追加パラメータが決まります。各形式のパラメータリストを参照してください。

数値値のカラムパラメータ:

表示	値をどのように表示するかを定義します: そのまま - 通常のテキストとして; バー - ソリッドで色付きのバーとして; インジケーター - セグメント化された色付きのバーとして; スパークライン - ミニ折れ線グラフ。
最小	バー/インジケーター表示の最小値。
最大	バー/インジケーター表示の最大値。
幅	スライダーを使用するか、0 から 10 の範囲で値を手動入力してグラフ線の太さを設定します。 このパラメータはスパークライン表示専用です。
塗りつぶし	スライダーを使用するか、0 から 10 の範囲で値を手動入力して塗りつぶし色の透明度レベルを設定します。 このパラメータはスパークライン表示専用です。
色	線と塗りつぶしの色を選択します。 このパラメータはスパークライン表示専用です。
時間期間と履歴データ選択に関連するフィールドの説明については詳細設定を参照してください。	
しきい値	背景/塗りつぶし色を変更するしきい値を指定します。 リストは保存時に昇順でソートされます。 スパークラインの場合、しきい値はアイテムの最新値のみに適用されます。
小数点以下の桁数	値とともに表示される小数点以下の桁数を指定します。
詳細設定	

集計関数	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; 未使用 - 最新値を表示（集計なし）。 集計を使用すると、最新値の代わりに選択した間隔（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* には数値データのみ表示できます。*count* の場合、非数値データは数値に変換されます。
時間期間	値を集計するために使用する時間期間を指定します。時間期間のデータソースを選択します: ダッシュボード - ダッシュボードの時間期間セレクターを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - *From* および *To* パラメータで設定されたカスタム時間期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの時間期間設定に関係なく、互換性のあるウィジェットは引き続き時間期間のデータソースとして使用できます。 このパラメータは集計関数が「未使用」に設定されている場合は利用できません。
ウィジェット	時間期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは時間期間が「ウィジェット」に設定されている場合に利用できます。
From	時間期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは時間期間が「カスタム」に設定されている場合に利用できます。
To	時間期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは時間期間が「カスタム」に設定されている場合に利用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。 このパラメータは数値アイテム値にのみ利用できます。テキスト値の場合、データは常に履歴から取得されます。

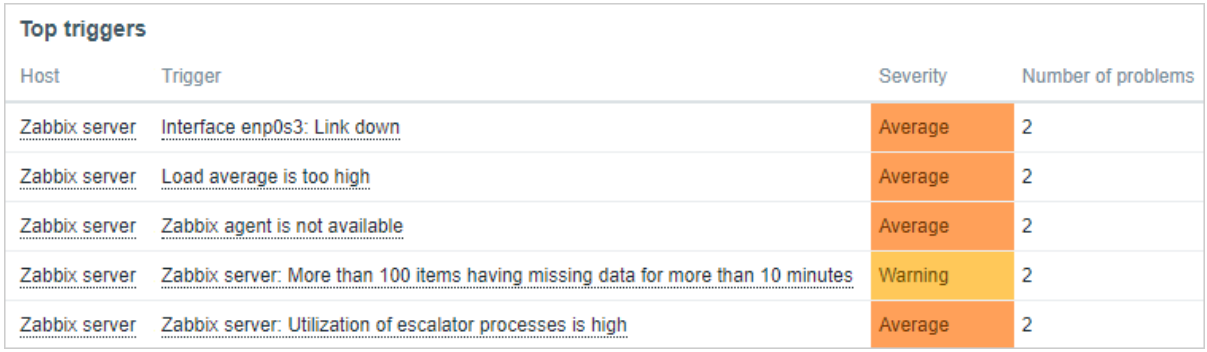
テキスト値のカラムパラメータ:

ハイライト	背景/塗りつぶし色を変更するために一致させる正規表現を指定します。

30 トップトリガー

概要

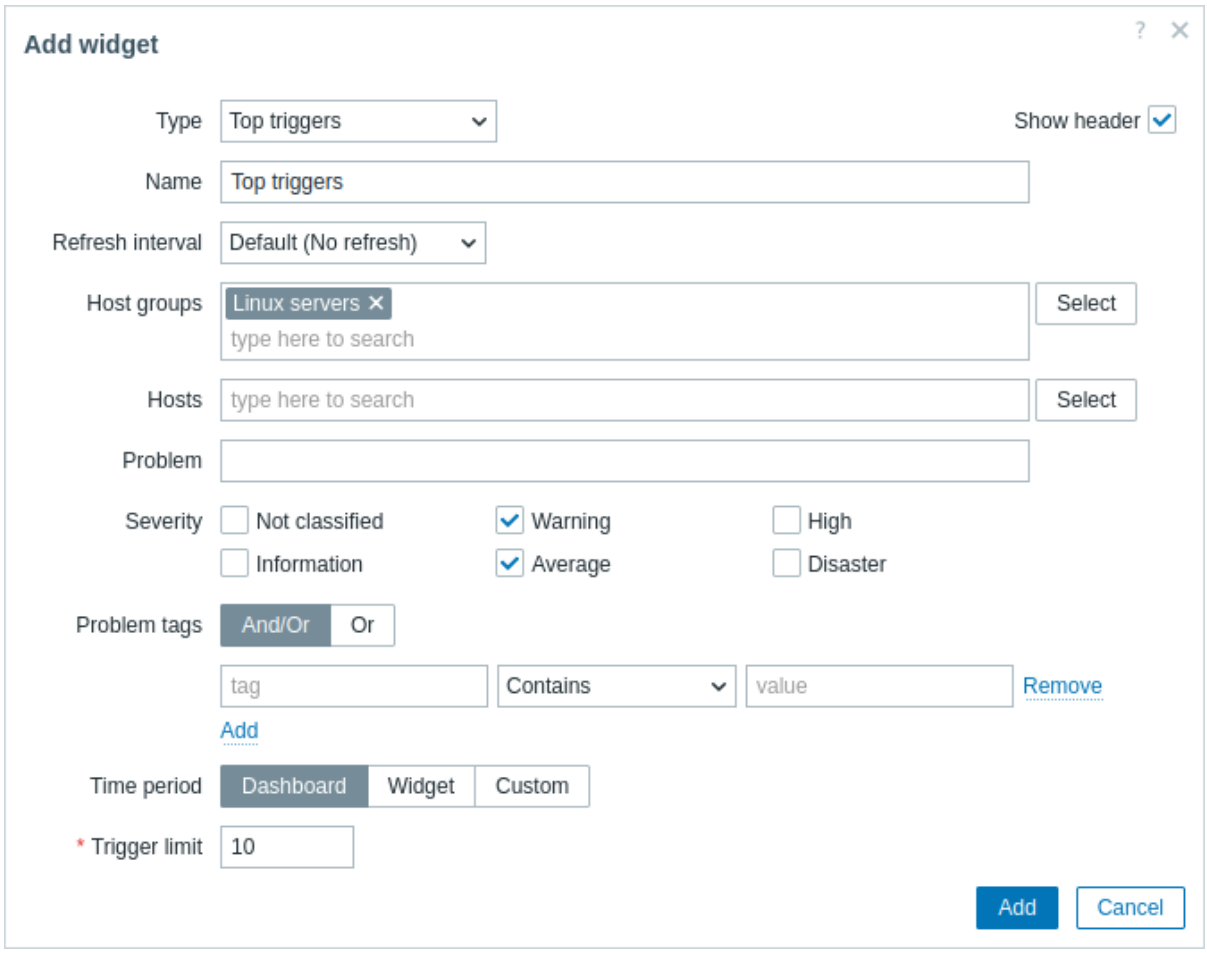
Top triggers ウィジェットは、最も多くの問題が発生しているトリガーを表示します。



* レポート > Top 100 triggers*と同じデータを表示し、最大 1000 件まで表示できます。

設定

設定するには、ウィジェットタイプとして トップトリガーを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ウィジェットに表示する問題のホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループの問題がウィジェットに表示されます。ホストグループを入力しない場合、すべてのホストグループの問題が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示する問題のホストを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合、すべてのホストの問題が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
問題	特定の問題のみのトリガーを表示できます。そのためには、問題名に一致する文字列を入力します。 マクロは展開されません。
深刻度	ウィジェットに表示するトリガーをフィルタリングするために、トリガーの深刻度をマークします。 深刻度をマークしない場合、すべてのトリガーが表示されます。

問題タグ	ウィジェットに表示する問題のタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字・小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定した時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
トリガー数の上限	表示するトリガー数を設定します。設定可能な値の範囲: 1-1000。

31 トリガーの概要

概要

トリガー概要ウィジェットは、ホストトリガーの現在の状態を色付きブロックのテーブルとして表示します。これにより、異なるホストのトリガーの健全性とアクティビティの視覚的なスナップショットを素早く把握できます。



このウィジェットは、トリガーの詳細を伝えるために複数の視覚要素を使用します:

- PROBLEM トリガーのブロック色は、問題の深刻度に一致します。
- 点滅するブロックは、最近のトリガー状態の変化（過去 2 分以内）を示します。
- グレーの矢印は依存関係のあるトリガーを示します（詳細を見るにはマウスオーバーしてください）。
- チェックボックスアイコンは、問題が認識済みであることを示します。トリガーのすべてのアクティブまたは解決済みの問題が認識されている場合にのみ、このアイコンが表示されます。

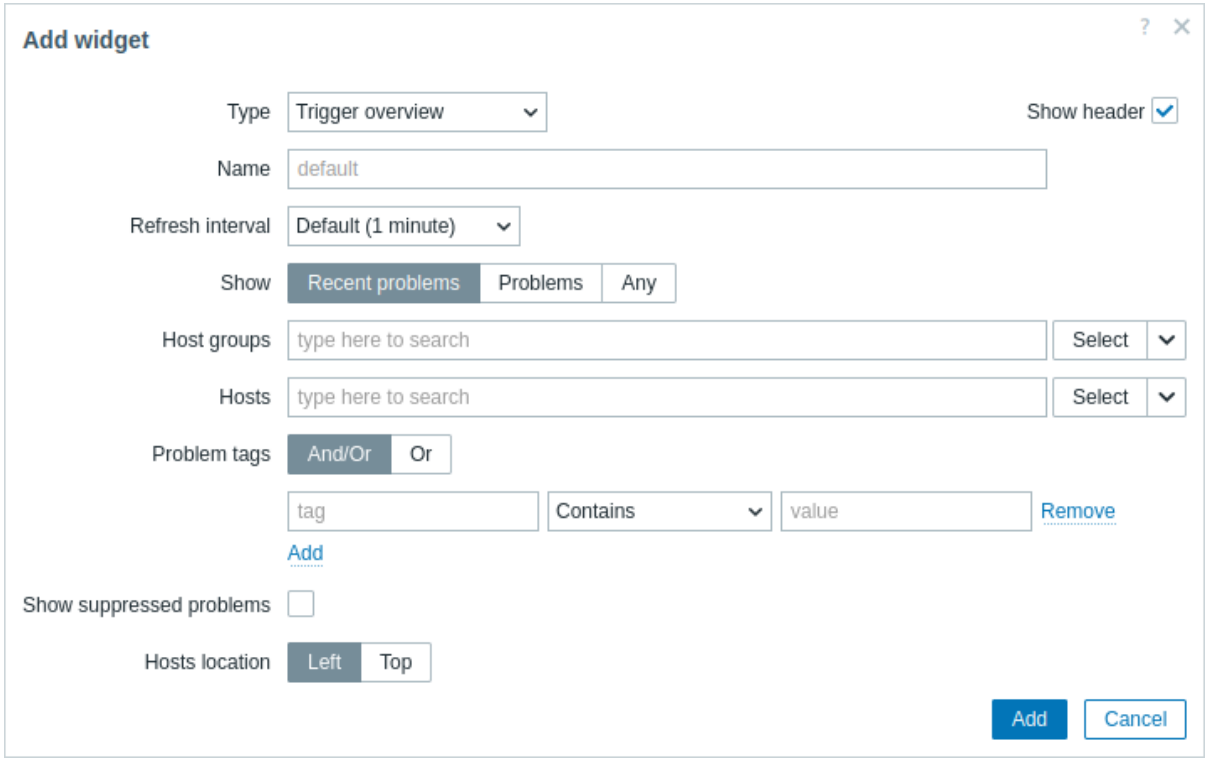
Note:
トリガーの深刻度の色は、管理 > 一般で調整できます。トリガーの深刻度は、問題の更新時に変更できます。

ブロックをクリックすると、トリガーに関連する情報（問題、トリガー設定など）へのリンクが表示されます。

デフォルトでは最大 50 件のレコードが表示されます（管理 > 一般 > *GUI* の 概要テーブルの最大列数と行数で設定可能）。トリガーがそれ以上存在する場合は、フィルタリング条件を絞り込むようメッセージが表示されます。この制限は、追加のフィルタリング（タグによるフィルタリングなど）の前に適用されます。

設定

設定するには、タイプとして トリガーの概要を選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

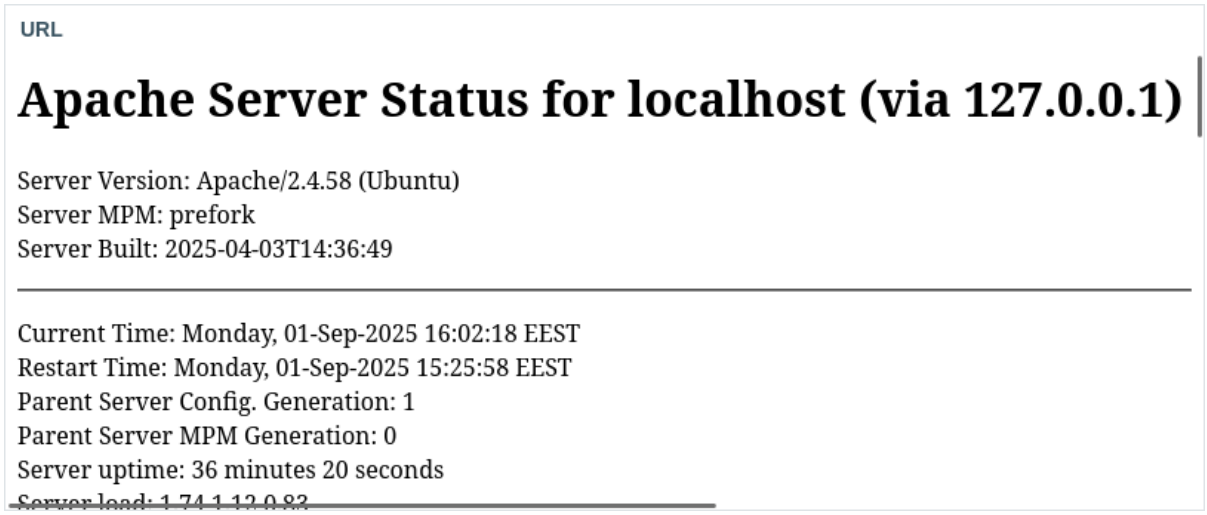
表示	トリガーの状態でフィルタリング: 最近の障害 - *(デフォルト)* 最近障害状態になった、または現在も障害状態のトリガー (解決済みと未解決) を表示; 障害 - 障害状態 (未解決) のトリガーを表示; すべて - すべてのトリガーを表示。
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応なので、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
障害タグ	ウィジェットに表示するトリガーをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 注意**:** パラメータ 表示が「すべて」に設定されている場合、タグを指定してもすべてのトリガーが表示されます。ただし、最近のトリガー状態の変更 (点滅ブロックとして表示) はすべてのトリガーで更新されますが、トリガー状態の詳細 (障害の深刻度の色や障害が認識済みかどうか) は、指定したタグに一致するトリガーのみ更新されます。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める (大文字小文字を区別); 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字小文字を区別しない); 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する (大文字小文字を区別); 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字小文字を区別しない)。 条件の計算タイプは 2 つあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

抑制された障害を表示	ホストのメンテナンスによって抑制 (非表示) される障害も表示するにはチェックボックスをオンにします。
レイアウト	レイアウトオプションを選択: 横 - ホスト名が左側に表示されます; 縦 - ホスト名が上部に表示されます。

32 URL

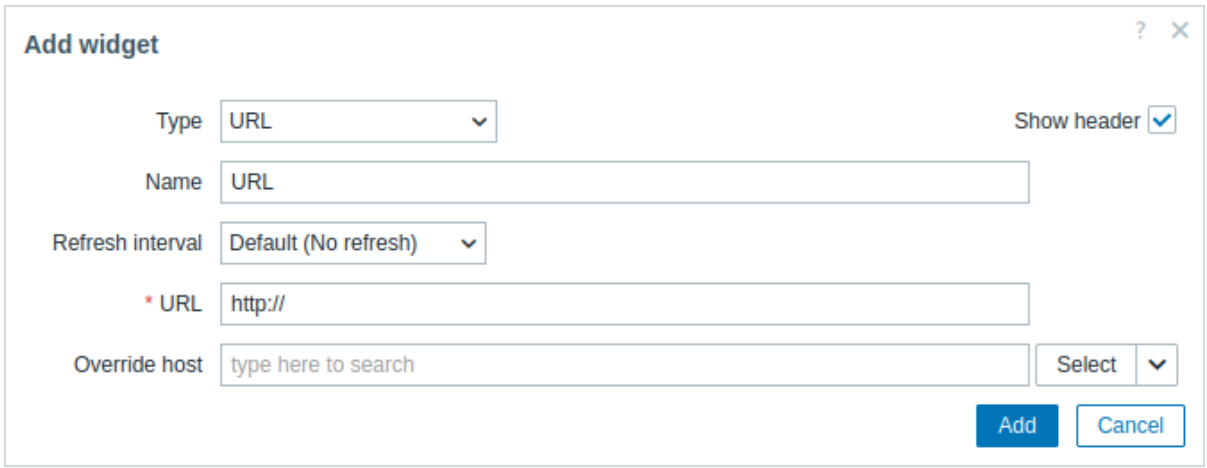
概要

URL ウィジェットは、URL から取得したコンテンツを表示します。



設定

設定するには、タイプとして *URL* を選択します:



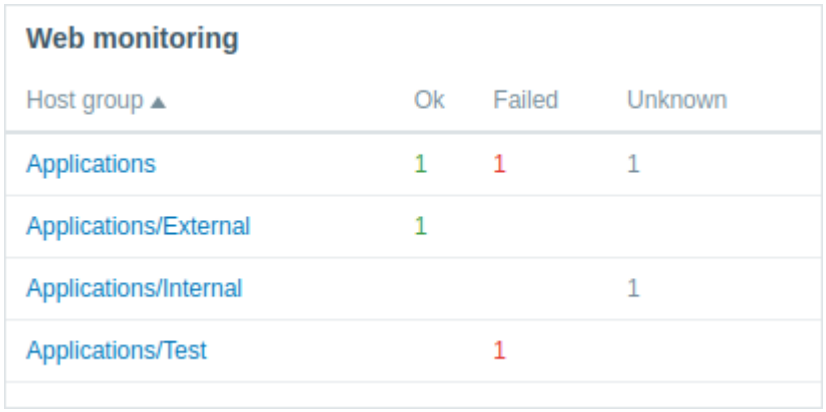
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

URL	表示する URL を入力します（最大 2048 文字）。 外部 URL は `http://`または `https://`で始める必要があります。 内部 URL は相対パスをサポートします（例: `zabbix.php?action=report.status`）。 {HOST.*} マクロがサポートされています。
ホストの上書き	互換性のあるウィジェットまたはダッシュボードのホストセレクタを、ホストのデータソースとして選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

Attention:
Zabbix フロントエンドに HTTPS 経由でアクセスしている場合、ウィジェットで設定した HTTP ページがブラウザで読み込まれないことがあります。また、Zabbix フロントエンド外のコンテンツを表示するには、X-Frame-Options HTTP レスポンスヘッダーを変更する必要がある場合があります。

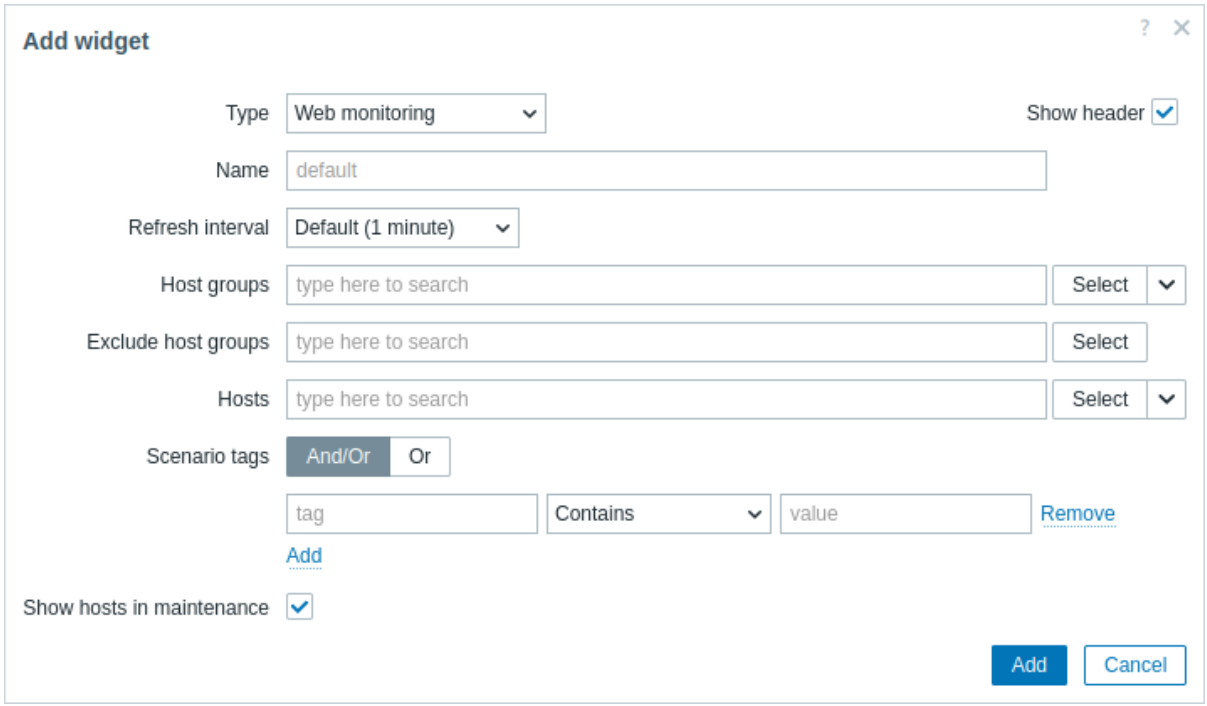
概要

Web 監視ウィジェットは、アクティブな Web シナリオのステータスを表示します。



ホストグループ名をクリックすると、監視データ > ホスト > *Web* シナリオが開き、シナリオの詳細を確認できます。

設定



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

ホストグループの除外	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループのホストデータはウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
シナリオタグ	ウィジェットに表示する Web シナリオの数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
メンテナンス中のホストを表示	メンテナンス中のホストを統計に含めます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はメンテナンス中のデータを表示と表示されます。

2 監視データ

概要

監視データメニューは、すべてデータの表示に関するものです。Zabbix が収集、視覚化、操作するために設定されている情報が何であれ、それは監視データメニューのさまざまなセクションに表示されます。

ビューモードボタン

右上隅にある次のボタンは、すべてのセクションで共通です。

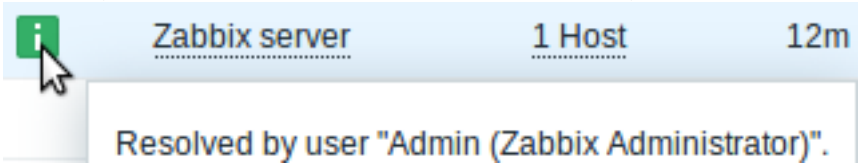
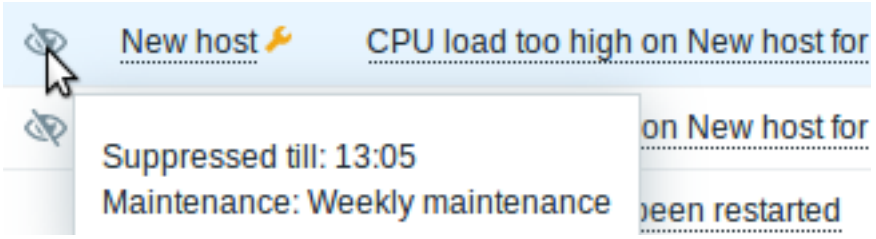
	ページをキオスクモードで表示します。このモードではページの内容のみが表示されます。 キオスクモードを終了するには、マウスカーソルを移動して  終了ボタンが表示されたら、それをクリックします。通常モードに戻ります。

1 障害

概要

監視 > 障害では、現在発生している障害を確認できます。障害とは、「障害」状態にあるトリガーのことです。

デフォルトでは、すべての新しい障害は原因障害として分類されます。特定の障害を手動で原因障害の症状障害として再分類することも可能です。詳細については、原因イベントと症状イベントを参照してください。

列	説明
チェックボックス	障害選択用のチェックボックスが表示されます。 チェックボックスの横のアイコンには以下の意味があります:  - 原因障害に対する症状イベントの数;  - 症状イベントを表示するために展開;  - 症状イベントを非表示にするために折りたたみ;  - これは症状イベントです。
時刻	障害の発生時刻が表示されます。
深刻度	障害の深刻度が表示されます。 障害の深刻度は、元々は基礎となる障害トリガーの深刻度に基づいていますが、イベント発生後に障害の更新画面で更新することができます。障害発生中は、障害の深刻度の色がセルの背景色として使用されます。
復旧時刻	障害の復旧時刻が表示されます。
ステータス	障害のステータスが表示されます: 障害 - 未解決の障害 復旧 - 最近復旧した障害。フィルタを使用して最近復旧した障害を非表示にできます。 新規および最近復旧した障害は 2 分間点滅します。復旧した障害は合計 5 分間表示されます。これらの値は管理 > 一般 > トリガー表示オプションで設定できます。
情報	障害がグローバル相関または障害の更新時に手動でクローズされた場合、緑色の情報アイコンが表示されます。アイコンにマウスを重ねると詳細が表示されます:  抑制された障害が表示されている場合は、次のアイコンが表示されます（フィルタの抑制された障害を表示オプションを参照）。アイコンにマウスを重ねると詳細が表示されます: 
ホスト	障害が発生したホストが表示されます。 ホスト名をクリックするとホストメニューが表示されます。
障害	障害名が表示されます。 障害名は、基礎となる障害トリガーの名前に基づいています。 トリガー名のマクロは障害発生時に解決され、その後は値が更新されません。 運用データを障害名に追加して、最新のアイテム値を表示することも可能です。 障害名をクリックするとイベントメニューが表示されます。 障害名の後ろの  アイコンにマウスを重ねると、トリガーの説明（設定されている場合）が表示されます。
運用データ	運用データが表示され、最新のアイテム値が含まれます。 運用データは、トリガーレベルで設定されている場合はテキストとアイテム値マクロの組み合わせになります。トリガーレベルで運用データが設定されていない場合は、式に含まれるすべてのアイテムの最新値が表示されます。 この列は、フィルタの運用データの表示で個別を選択した場合のみ表示されます。
継続時間	障害の継続時間が表示されます。 参考: 負の障害継続時間
更新	更新リンクをクリックすると、障害の更新画面に移動し、障害に対してコメントや認識などのさまざまな操作を行うことができます。

列	説明
アクション	障害に関するアクティビティの履歴が記号アイコンで表示されます:  - 障害が認識されました。このアイコンは常に最初に表示されます。  - コメントが追加されました。コメント数も表示されます。  - 障害の深刻度が上がりました（例: 情報 → 警告）  - 障害の深刻度が下がりました（例: 警告 → 情報）  - 障害の深刻度が変更されましたが、元のレベルに戻りました（例: 警告 → 情報 → 警告）  - アクションが実行されました。アクション数も表示されます。  - アクションが実行され、少なくとも 1 つが進行中です。アクション数も表示されます。  - アクションが実行され、少なくとも 1 つが失敗しました。アクション数も表示されます。 アイコンにマウスを重ねると、アクティビティの詳細を含むポップアップが表示されます。実行されたアクションのポップアップで使用されるアイコンについては、詳細の表示を参照してください。
タグ	タグが表示されます（存在する場合）。 さらに、外部チケッティングシステムからのタグも表示される場合があります（Webhook設定時のタグの処理オプションを参照）。

障害の運用データ

現在の障害に対して運用データ、すなわち障害発生時のアイテム値ではなく最新のアイテム値を表示することができます。

運用データの表示は、監視データ > 障害のフィルタや、該当するダッシュボードウィジェットの設定で、次の 3 つのオプションのいずれかを選択して設定できます。

- なし - 運用データは表示されません
- 別列 - 運用データが別の列に表示されます
- 障害名に追加 - 運用データが障害名の後ろに括弧付きで追加されます。運用データは、トリガー設定の運用データフィールドが空でない場合のみ障害名に追加されます。

運用データの内容は、各トリガーの運用データフィールドで設定できます。このフィールドには、任意の文字列とマクロを指定できます。特に重要なのは {ITEM.LASTVALUE<1-9>} マクロです。

このフィールド内の {ITEM.LASTVALUE<1-9>} は、常にトリガー式内のアイテムの最新値に展開されます。このフィールド内の {ITEM.VALUE<1-9>} は、トリガーの状態が変化した瞬間（障害への変化、OK への変化、ユーザーによる手動クローズ、または相関によるクローズ）でのアイテム値に展開されます。

障害を手動でクローズしても新しい値は生成されないため、{ITEM.LASTVALUE<1-9>} や {ITEM.VALUE<1-9>} の展開値は障害発生時の値のままになります。

最新の履歴値が「最大履歴表示期間」よりも前に収集された場合（「管理」>「一般」* を参照）、{ITEM.LASTVALUE<1-9>} または {ITEM.VALUE<1-9>} は *UNKNOWN* に展開されます。

負の障害継続時間

実際には、障害の解決時刻が障害の発生時刻よりも早い場合、つまり負の障害継続時間になることが、いくつかの一般的な状況で発生する可能性があります。例えば:

- あるホストがプロキシによって監視されていて、ネットワークエラーが発生し、しばらくの間プロキシからデータが受信されなくなった場合、nodata(/host/key) トリガーがサーバーによって発火されます。接続が回復すると、サーバーは過去の時刻を持つアイテムデータをプロキシから受信します。その後、nodata(/host/key) の障害が解決され、負の障害継続時間となります。
- 障害イベントを解決するアイテムデータが Zabbix sender によって送信され、そのタイムスタンプが障害発生時刻よりも早い場合も、負の障害継続時間が表示されます。

Note:
負の障害継続時間は、特定のトリガーのSLA 計算や可用性レポートには一切影響しません。障害時間を短縮も延長もしません。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

- 一括更新 - 選択した障害を障害の更新画面で更新します

このオプションを使用するには、該当する障害の前のチェックボックスをオンにして、一括更新ボタンをクリックします。

ボタン

右側のボタンには、次のオプションがあります。

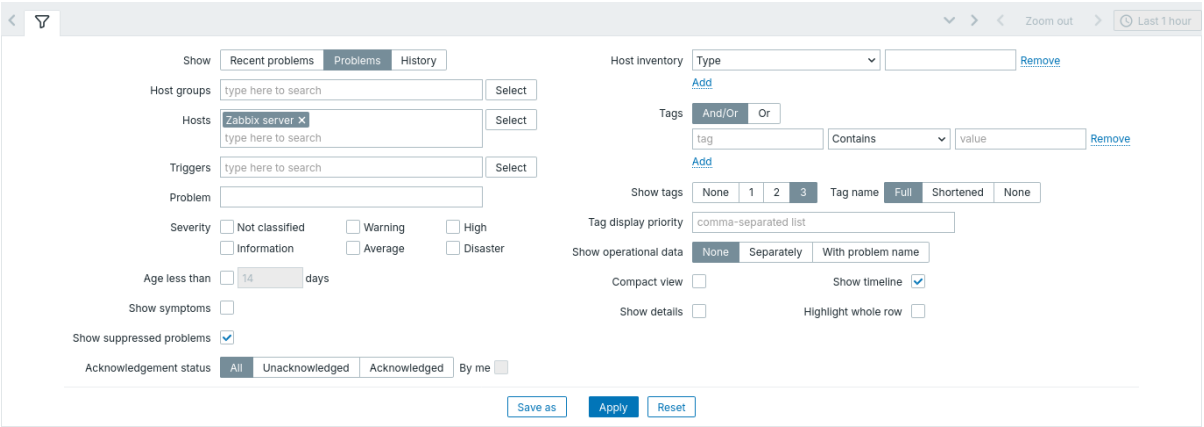
Export to CSV	すべてのページの内容を CSV ファイルにエクスポートします。

表示モードのボタンはすべてのセクションで共通であり、監視ページで説明されています。

フィルターの使用

フィルターを使用して、関心のある問題のみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決のまま検索されます。

フィルターはテーブルの上にあります。お気に入りのフィルター設定はタブとして保存でき、フィルター上部のタブをクリックすることで素早くアクセスできます。



パラメータ	説明
表示	問題のステータスでフィルター: 最近の問題 - 未解決および最近解決された問題が表示されます（デフォルト） 問題 - 未解決の問題が表示されます 履歴 - すべてのイベントの履歴が表示されます
ホストグループ	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。
ホスト	1 つ以上のホストでフィルターします。
トリガー	1 つ以上のトリガーでフィルターします。
問題	問題名でフィルターします。
深刻度	トリガー（問題）の深刻度でフィルターします。
経過時間が次より小さい	問題が発生してからの経過時間でフィルターします。
症状を表示	チェックボックスをオンにすると、症状として分類された問題が独自の行で表示されます。
抑制された問題を表示	チェックボックスをオンにすると、ホストのメンテナンスや単一の問題抑制のために抑制（非表示）される問題が表示されます。
認識ステータス	すべての問題、未認識の問題のみ、または認識済みの問題のみを表示するようにフィルターします。 追加のチェックボックスをオンにすると、自分が一度でも認識した問題を除外できます。
ホストインベントリ	インベントリのタイプと値でフィルターします。

パラメータ	説明
タグ	イベントタグの名前と値でフィルターします。特定のタグやタグ値を含めたり除外したりできます。 複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です フィルター後、ここで指定したタグはタグ表示優先度（下記参照）リストで上書きされない限り、問題とともに最初に表示されます。
タグの表示	表示するタグの数を選択します: なし - 監視 > 問題にタグ列がありません **1** - タグ列に 1 つのタグが表示されます **2** - タグ列に 2 つのタグが表示されます **3** - タグ列に 3 つのタグが表示されます 問題のすべてのタグを表示するには、三点リーダーアイコンにマウスを重ねてください。
タグ名	タグ名の表示モードを選択します: フル - タグ名と値がすべて表示されます 短縮 - タグ名が 3 文字に短縮され、タグ値はすべて表示されます なし - タグ値のみが表示され、名前は表示されません
タグ表示優先度	カンマ区切りのタグリスト（例：`customer,scope,component`）として、問題のタグ表示優先度を入力します。値ではなくタグ名のみを使用してください。このリストのタグは常に最初に表示され、アルファベット順の自然な並び順を上書きします。
運用データの表示	運用データの表示モードを選択します: なし - 運用データは表示されません 別列 - 運用データが別の列に表示されます 問題名と一緒 - 運用データを問題名に括弧付きで追加します
コンパクトビュー	チェックボックスをオンにするとコンパクトビューが有効になります。
詳細を表示	チェックボックスをオンにすると、問題の基礎となるトリガー式が表示されます。コンパクトビューのチェックボックスがオンの場合は無効です。
タイムラインを表示	チェックボックスをオンにすると、ビジュアルタイムラインとグループ化が表示されます。コンパクトビューのチェックボックスがオンの場合は無効です。
行全体をハイライト	チェックボックスをオンにすると、未解決の問題の行全体がハイライトされます。ハイライトには問題の深刻度の色が使用されます。 行全体をハイライトはハイコントラストテーマでは利用できません。

お気に入りフィルターのタブ

よく使うフィルターパラメータのセットは、タブに保存できます。

新しいフィルターパラメータのセットを保存するには、メインタブを開き、フィルター設定を構成してから、名前を付けて保存ボタンを押します。新しいポップアップウィンドウで、フィルターのプロパティを定義します。

problem_filter0.png

パラメータ	説明
名前	タブリストに表示するフィルターの名前。
レコード数を表示	タブ名の横に問題の数を表示したい場合はチェックします。
期間セレクターを上書き	このフィルターセットに特定のデフォルト期間を設定する場合はチェックします。設定した場合、このタブの期間はフィルター設定を更新することでのみ変更できます。カスタム期間のないタブの場合は、右上隅の期間セレクターボタン（ボタン名は選択した期間によって異なります：今週、過去 30 分、昨日など）を押すことで期間を変更できます。 このオプションは監視 > 障害のフィルターでのみ利用可能です。
開始/終了	期間の開始と終了を絶対値（`Y-m-d H:i:s`）または相対値（`now-1d`）で指定します。カスタム期間を設定にチェックした場合に利用可能です。

保存すると、フィルターは名前付きフィルタータブとして作成され、すぐに有効になります。

既存のフィルターのプロパティを編集するには、アクティブなタブ名の横にある歯車のマークを押します。

problem_filter2.png

注意:

- フィルターエリアを非表示にするには、現在のタブ名をクリックします。もう一度アクティブなタブ名をクリックすると、フィルターエリアが再び開きます。
- キーボードナビゲーションに対応しています：矢印キーでタブを切り替え、*Enter* で開きます。
- フィルター上部の左右ボタンで保存済みフィルターを切り替えられます。または、下向きボタンでドロップダウンメニューが開き、すべての保存済みフィルターから必要なものをクリックできます。
- フィルタータブはドラッグ＆ドロップで並べ替え可能です。
- 保存済みフィルターの設定が変更された（まだ保存されていない）場合、フィルター名の後ろに緑色の点が表示されます。新しい設定でフィルターを更新するには、名前を付けて保存ボタンの代わりに表示される更新ボタンをクリックします。
- 現在のフィルター設定はユーザープロファイルに記憶されます。ユーザーが再度ページを開いたとき、フィルター設定はそのまま維持されます。

Note:
フィルターを共有するには、アクティブなフィルターの URL をコピーして他の人に送信します。この URL を開くと、他のユーザーはこのパラメータセットを自分の Zabbix アカウントの永続的なフィルターとして保存できます。
参考: ページパラメータ。

フィルターのボタン

Apply	指定されたフィルタリングを（保存せずに）適用します
Reset	現在のフィルターをリセットし、現在のタブに保存されたパラメーターに戻ります。メインタブの場合はフィルターがクリアされます。
Save as	現在のフィルタリングパラメーターを新しいタブに保存します。メインタブでのみ使用できます。
Update	タブに保存されているパラメータを現在指定されているパラメータに置き換えます。メインタブでは使用できません。

詳細の表示

監視 > 障害の障害発生時刻と復旧時刻はリンクになっています。クリックするとイベントの詳細が表示されます。

障害イベントの障害の更新画面で障害イベントの深刻度が更新された場合、トリガーと障害イベントの深刻度が異なる場合があることに注意してください。

アクションリストでは、以下のアイコンがアクティビティタイプを示すために使用されます:

- icon_generated.png - 障害イベントが生成された
- icon_message.png - メッセージが送信された
- icon_acknowledged.png - 障害イベントが確認された
- icon_unacknowledged.png - 障害イベントの確認が解除された
- icon_comment2.png - コメントが追加された
- icon_sev_up1.png - 障害の深刻度が上がった（例: 情報 → 警告）
- icon_sev_down1.png - 障害の深刻度が下がった（例: 警告 → 情報）
- icon_severity_back.png - 障害の深刻度が変更されたが、元のレベルに戻った（例: 警告 → 情報 → 警告）
- icon_remote.png - リモートコマンドが実行された
- icon_recovery.png - 障害イベントが復旧した
- icon_closed.png - 障害が手動でクローズされた
- icon_suppression.png - 障害が抑制された
- icon_unsuppressed.png - 障害の抑制が解除された
- icon_symptom.png - 障害がシンプトム障害に変換された
- icon_cause.png - 障害が原因障害に変換された

1 原因と症状の障害

概要

デフォルトでは、すべての新しい問題は原因問題として分類されます。特定の問題を手動で原因問題の症状問題として再分類することが可能です。

たとえば、停電が実際の根本原因であり、あるホストに到達できない、またはサービスがダウンしている理由である場合があります。この場合、「ホストに到達できない」や「サービスがダウンしている」という問題は、「停電」という原因問題の症状問題として分類する必要があります。

原因-症状の階層は 2 レベルのみをサポートします。すでに症状となっている問題には「従属」する症状問題を割り当てることはできません。症状問題に割り当てられた問題は、すべて同じ原因問題の症状となります。

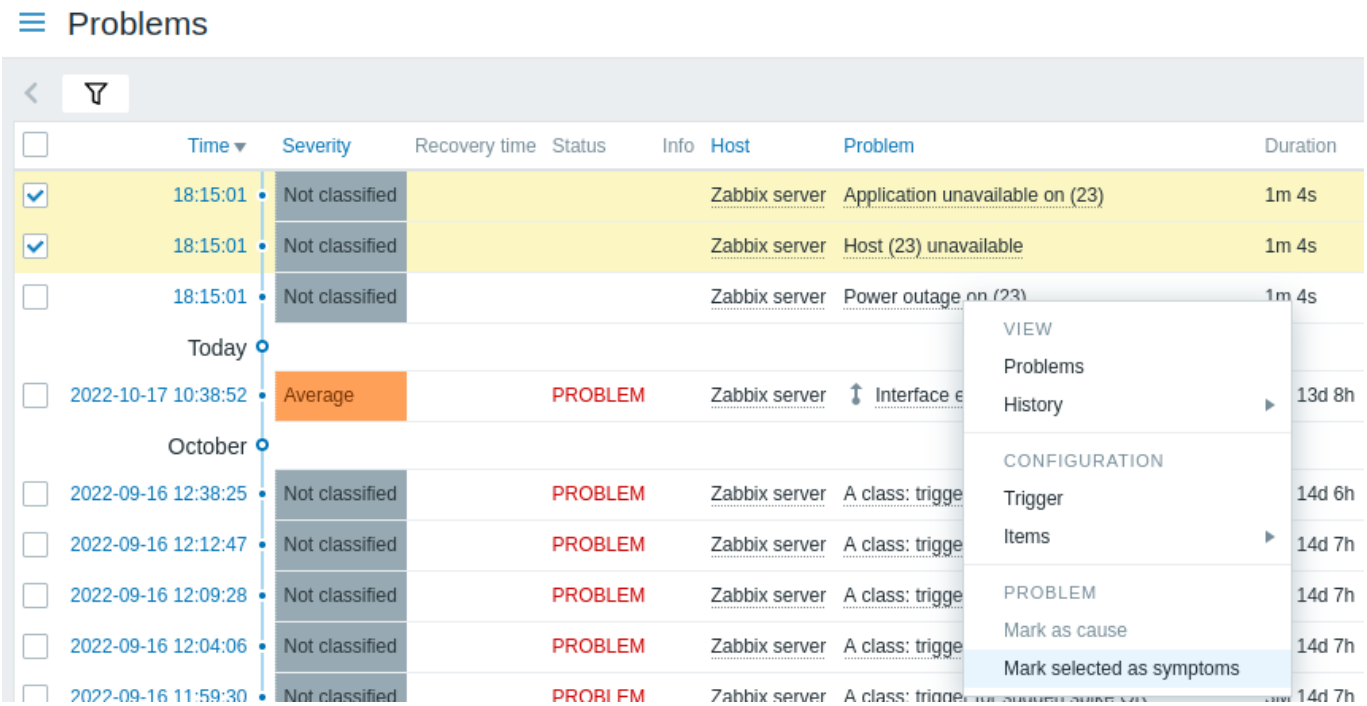
原因問題のみが、マップや深刻度別の問題や問題のあるホストなどのダッシュボードウィジェットの問題合計にカウントされます。ただし、問題のランク付けはサービスには影響しません。

症状問題は 1 つの原因問題のみにリンクできます。原因問題が解決またはクローズされても、症状問題は自動的には解決されません。

設定

問題を症状問題として再分類するには、まず問題のリストで選択します。1 つまたは複数の問題を選択できます。

次に、原因問題に移動し、そのコンテキストメニューで選択したものを症状としてマークオプションをクリックします。



その後、選択した問題はサーバーによって原因問題の症状問題として更新されます。

問題のステータスが更新されている間は、次のいずれかの方法で表示されます。

- ステータス列に点滅する「UPDATING」ステータスが表示されます。
- 情報列に点滅する または アイコンが表示されます（これは、フィルターで問題のみが選択されている場合に有効であり、その場合はステータス列は表示されません）。

表示

症状の問題は、監視 -> 問題（および問題ダッシュボードウィジェット）で、原因の問題の下に表示され、アイコン、小さいフォント、異なる背景でマークされます。

cause_symptom_display2.png

折りたたみ表示では、原因の問題のみが表示され、症状の問題が存在する場合は、行の先頭の数字と表示を展開するアイコンで示されます。

cause_symptom_display.png

また、症状の問題を通常のフォントで独自の行に追加表示することもできます。その場合は、フィルター設定またはウィジェット設定で症状を表示を選択します。

原因問題への戻し

症状問題は、原因問題に戻すことができます。そのためには、次のいずれかを行います。

- 症状問題のコンテキストメニューで原因としてマークオプションをクリックします。
- 問題の更新画面で原因に変換オプションにチェックを入れ、更新をクリックします (このオプションは複数の問題が選択されている場合にも有効です)。

2 ホスト

概要

監視 → ホストセクションには、監視対象ホストの一覧が表示され、ホストインターフェース、可用性、タグ、現在の問題、ステータス（有効/無効）、およびホストの最新データ、問題履歴、グラフ、ダッシュボード、Web シナリオへの簡単なナビゲーションリンクなどの詳細情報が表示されます。

列	説明
名前	表示されるホスト名です。名前をクリックするとホストメニューが表示されます。 名前の後にオレンジ色のレンチアイコン が表示されている場合は、このホストがメンテナンス中であることを示します。 列ヘッダーをクリックすると、ホスト名で昇順または降順に並べ替えることができます。
インターフェース	ホストのメインインターフェースが表示されます。

列	説明
可用性	設定されたインターフェースごとにホストの可用性が表示されます。 可用性アイコンは、Zabbix サーバー上のホストインターフェースの現在のステータスを表します。そのため、フロントエンドでホストを無効にした場合、Zabbix サーバーが設定変更を同期した後に可用性が更新されます。同様に、ホストを有効にした場合も、Zabbix サーバーが設定変更を同期し、ホストをポーリングした後に可用性が更新されます。 可用性アイコンは、設定されているインターフェースタイプ（Agent、SNMP、IPMI、JMX）のみを表します。 アイコンにカーソルを合わせると、同じタイプのすべてのインターフェースのリスト（詳細、ステータス、エラー付き）がポップアップで表示されます。Agent インターフェースの場合、ポップアップにはインターフェース（パッシブ）とアクティブチェックが表示されます。ホストがアクティブチェックのみの場合、Agent インターフェースが設定されていなくても Agent インターフェースアイコンが表示されます。 インターフェースがないホストの場合、この列は空白です。 単一のホストインターフェースのステータスは、そのインターフェースを使用する有効なアイテムとホスト間の接続によって決まります。ステータスは次のいずれかです。 利用可能 - ホストへの接続に成功した場合 利用不可 - ホストへの接続に失敗した場合（タイムアウト、ファイアウォールの問題など） 不明 - ホストへの接続が試行されていない、または結果が不明な場合 Zabbix サーバーがインターフェースのステータスをどのように判断するかの詳細については、不明なインターフェースステータスおよび到達不能/利用不可なホストインターフェースの設定を参照してください。 単一タイプ（Agent、SNMP、IPMI、JMX）のすべてのホストインターフェースのステータスは、少なくとも 1 つの有効なアイテムで使用されているインターフェースによって決まります。ステータスはアイコンの色で示されます。 緑 - すべてのインターフェースが利用可能 黄 - 少なくとも 1 つのインターフェースが利用不可で、少なくとも 1 つが利用可能または不明 赤 - すべてのインターフェースが利用不可 灰色 - 少なくとも 1 つのインターフェースが不明だが、利用不可はない アクティブチェックの可用性。ホストで少なくとも 1 つのアクティブチェックが有効になっている場合、アクティブチェックも上記のように Agent インターフェースの合計可用性に影響します。アクティブチェックの可用性を判断するために、エージェントのアクティブチェックスレッドでハートビートメッセージが送信されます。ハートビートメッセージの頻度は、Zabbixエージェントまたはエージェント 2の設定ファイルの HeartbeatFrequency パラメータで制御されます（デフォルト 60 秒、範囲 0-3600）。アクティブチェックのハートビートが 2 x HeartbeatFrequency 秒より古い場合、アクティブチェックは利用不可と見なされます。 注意**:** バージョン 6.2.x より古い Zabbix エージェントはアクティブチェックのハートビートを送信しないため、それらのホストの可用性は不明のままです。
タグ	ホストおよびすべてのリンクされたテンプレートのタグ（マクロは未解決）。
ステータス	ホストのステータス - 有効または 無効。 列ヘッダーをクリックすると、ステータスで昇順または降順に並べ替えることができます。
最新データ	リンクをクリックすると、そのホストから収集されたすべての最新データを含む監視 - 最新データページが開きます。 最新データのあるアイテム数がグレーで表示されます。
問題	ホストの未解決の問題数が深刻度別に表示されます。四角の色は問題の深刻度を示します。四角の数字はその深刻度の問題数を示します。 アイコンをクリックすると、そのホストの監視 - 問題ページが開きます。 ホストに問題がない場合は、そのホストの問題セクションへのリンクがテキストで表示されます。 フィルターを使用して、抑制された問題を含めるかどうかを選択できます（デフォルトでは含まれません）。
グラフ	リンクをクリックすると、ホストに設定されているグラフが表示されます。グラフの数がグレーで表示されます。 ホストにグラフがない場合は、リンクが無効（グレーのテキスト）になり、数は表示されません。
ダッシュボード	リンクをクリックすると、ホストに設定されているダッシュボードが表示されます。ダッシュボードの数がグレーで表示されます。 ホストにダッシュボードがない場合は、リンクが無効（グレーのテキスト）になり、数は表示されません。

列	説明
Web	リンクをクリックすると、ホストに設定されている Web シナリオが表示されます。Web シナリオの数がグレーで表示されます。 ホストに Web シナリオがない場合は、リンクが無効（グレーのテキスト）になり、数は表示されません。

ボタン

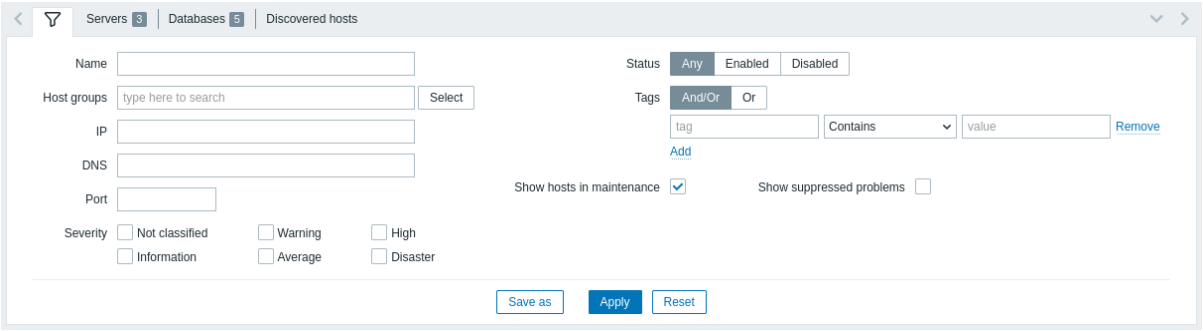
ホストの作成では、新しいホストを作成できます。このボタンは、管理者およびスーパー管理者ユーザーのみが使用できます。

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

フィルターの使用

フィルターを使用して、関心のあるホストのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルターはテーブルの上にあります。ホスト名、ホストグループ、IP または DNS、インターフェースポート、タグ、問題の深刻度、ステータス (有効/無効/すべて) でホストをフィルタリングできます。また、抑制された問題や現在メンテナンス中のホストを表示するかどうかも選択できます。



パラメータ	説明
名前	表示されるホスト名でフィルタリングします。
ホストグループ	1 つまたは複数のホストグループでフィルタリングします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。
IP	IP アドレスでフィルタリングします。
DNS	DNS 名でフィルタリングします。
ポート	ポート番号でフィルタリングします。
深刻度	問題の深刻度でフィルタリングします。デフォルトではすべての深刻度の問題が表示されます。抑制されていない場合に問題が表示されます。
ステータス	ホストのステータスでフィルタリングします。
タグ	ホストのタグ名と値でフィルタリングします。ホストレベルのタグだけでなく、すべてのリンクされたテンプレート (ネストされたテンプレートを含む) のタグでもフィルタリングできます。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める (大文字・小文字を区別) 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字・小文字を区別しない) 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する (大文字・小文字を区別) 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字・小文字を区別しない) 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
メンテナンス中のホストを表示	チェックボックスをオンにすると、メンテナンス中のホストが表示されます (デフォルトで表示)。
抑制された問題を表示	チェックボックスをオンにすると、ホストのメンテナンスや単一の問題の抑制により、通常は抑制されて表示されない問題が表示されます。

フィルターの保存

お気に入りのフィルター設定はタブとして保存でき、フィルター上部の該当するタブをクリックすることで素早くアクセスできます。

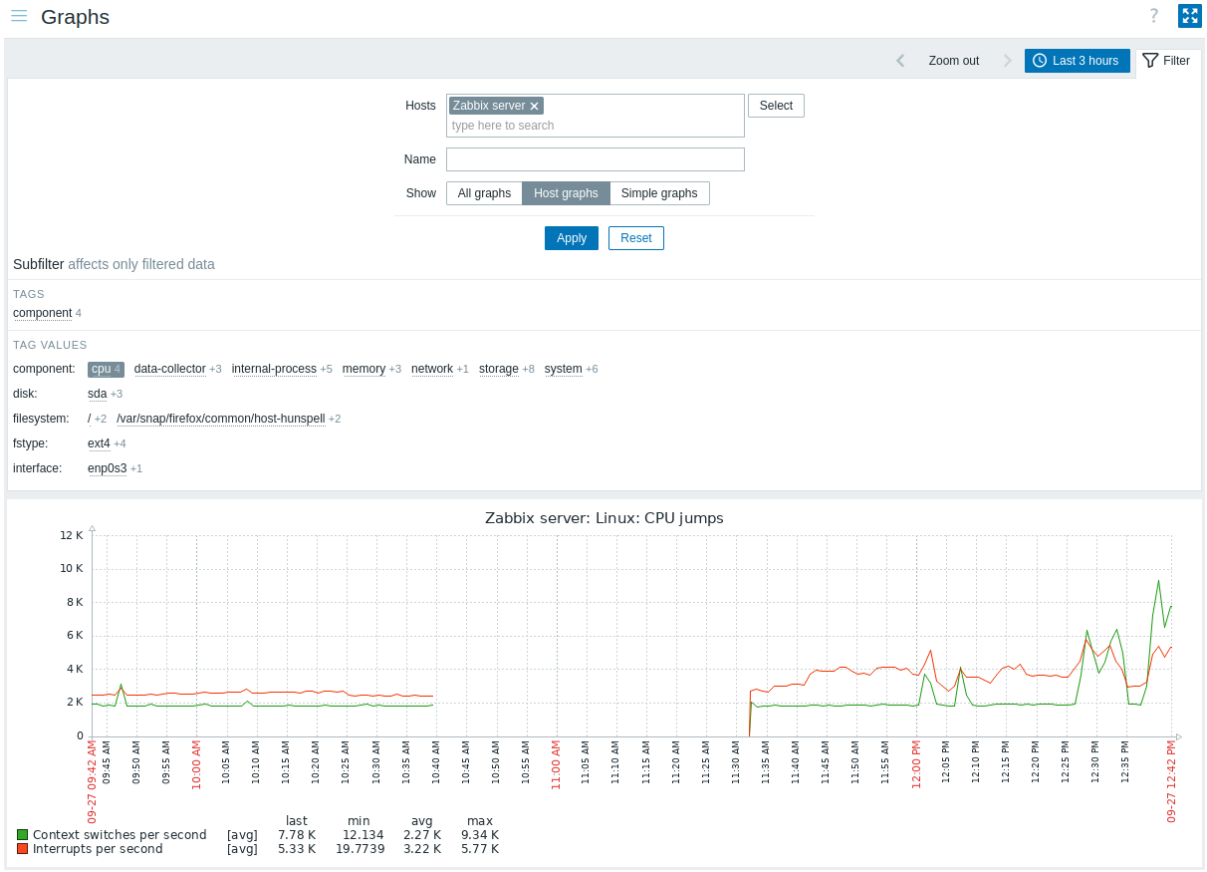
フィルターの保存の詳細をご覧ください。

1 グラフ

概要

ホストグラフには、監視 → ホストから該当ホストのグラフをクリックすることでアクセスできます。

ホストに設定されたカスタムグラフや、シンプルグラフを表示できます。



グラフは以下の順でソートされます。

- グラフ名（カスタムグラフ）
- アイテム名（シンプルグラフ）

無効化されたホストのグラフも参照できます。

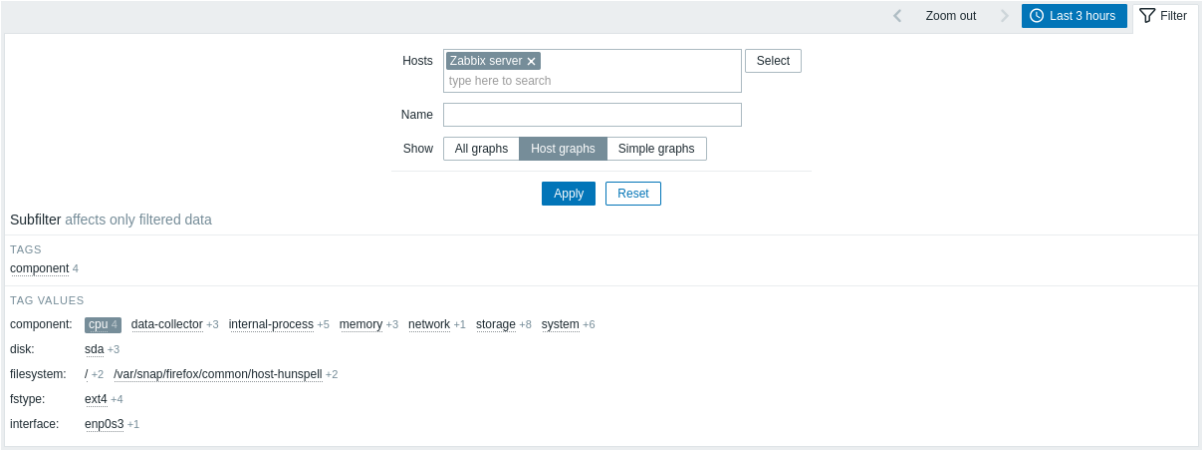
期間セレクター

グラフの上にある期間セレクターに注目してください。よく使う期間をワンクリックで選択できます。

参考: 期間セレクター

フィルターの使用

特定のグラフを表示するには、フィルターで選択します。フィルターでは、ホスト、グラフ名、表示オプション（すべて/ホストグラフ/シンプルグラフ）を指定できます。



フィルターでホストが選択されていない場合、グラフは表示されません。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターからさらにフィルタリングを変更できます。

サブフィルターには、関連するグラフへのクイックアクセス用のクリック可能なリンクが含まれています。グラフは共通のエンティティ（タグ名または値）によって関連付けられます。タグ名/値をクリックすると、グレーの背景でハイライト表示され、グラフが即座にフィルタリングされます（メインフィルターで適用をクリックする必要はありません）。別のタグ名/値をクリックすると、フィルタリングされた結果に追加されます。タグ名/値を再度クリックすると、フィルタリングが解除されます。

サブフィルターは、フィルタリングされたデータに基づいて生成されます。このデータは最大 1000 件に制限されています。サブフィルターでさらに多くのレコードを表示したい場合は、検索およびフィルター結果の制限パラメータ（管理 -> 一般 -> *GUI*）の値を増やす必要があります。

メインフィルターとは異なり、サブフィルターは各テーブルのリフレッシュリクエストとともに更新され、常に利用可能なフィルタリングオプションとそのカウンター数の最新情報を取得します。

表示されるエンティティの数は水平方向に 100 件に制限されています。それ以上の場合は、末尾に三点リーダーアイコンが表示されますが、クリックできません。垂直リスト（値付きのタグなど）は 20 件に制限されています。それ以上の場合は、三点リーダーアイコンが表示されますが、クリックできません。

各クリック可能なエンティティの横の数字は、メインフィルターの結果にそのエンティティが含まれるグラフの数を示します。

1 つのエンティティが選択されると、他の利用可能なエンティティの数字は、現在の選択に追加できるグラフの数を示すプラス記号付きで表示されます。

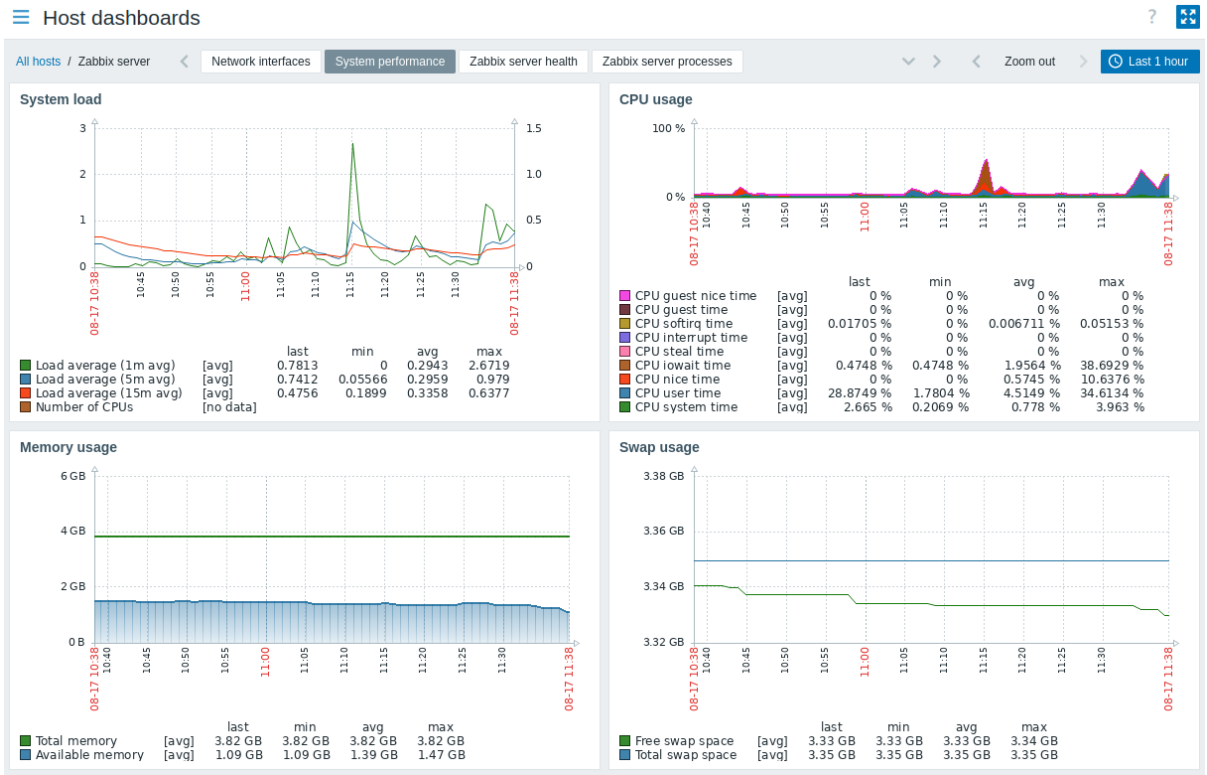
ボタン

表示モードのボタンはすべてのセクションで共通であり、監視ページで説明されています。

2 ホストダッシュボード

概要

ホストダッシュボードはグローバルダッシュボードと似ていますが、ホストダッシュボードには所有者がなく、選択したホストのデータのみが表示されます。



ホストダッシュボードを表示しているときは、以下をクリックすることで設定済みのダッシュボードを切り替えることができます。

- ダッシュボードのタブ
- ヘッダー下の  矢印ボタン
- ヘッダー下の  矢印ボタン（利用可能なホストダッシュボードの全リストが表示されます）

左上のヘッダー下にある *All hosts* ナビゲーションリンクをクリックすると、監視 → ホストセクションに切り替わります。

設定

ホストダッシュボードはテンプレートレベルで設定します。テンプレートがホストにリンクされると、そのホスト用のホストダッシュボードが生成されます。ホストダッシュボードはダッシュボードセクションで設定できないことに注意してください。このセクションはグローバルダッシュボード専用です。

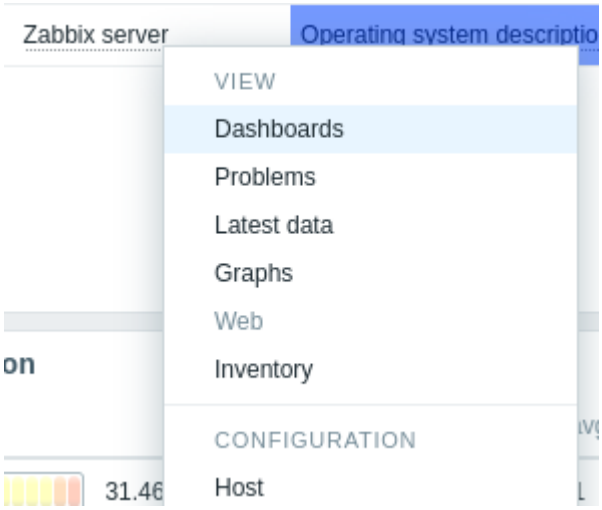
ホストダッシュボードのウィジェットも、テンプレートレベルでのみ設定できます（更新間隔の変更を除く）。さらに、ホストダッシュボードのウィジェットは、同じテンプレート内の他のホストダッシュボードにのみコピーできます。グローバルダッシュボードのウィジェットはホストダッシュボードにコピーできないことに注意してください。

Note:
Zabbix 5.2 以前では、ホストダッシュボードはホストスクリーンでした。スクリーンを含む古いテンプレートをインポートする場合、スクリーンのインポートは無視されます。

アクセス

ホストダッシュボードには、以下の方法でアクセスできます。

- グローバル検索でホスト名を検索した後（検索結果に表示されるダッシュボードリンクをクリック）
- インベントリ → ホストでホスト名をクリックした後（ホスト概要に表示されるダッシュボードリンクをクリック）
- ホストメニューからダッシュボードをクリック

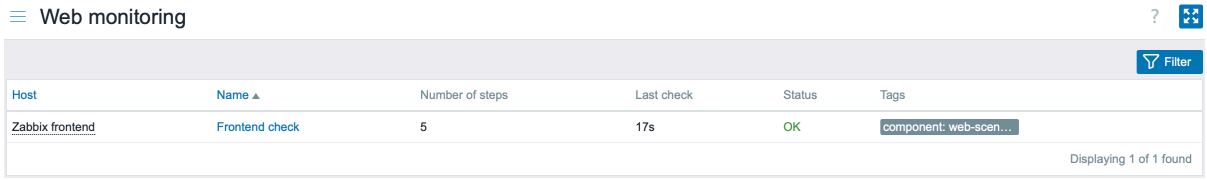


ホストダッシュボードはダッシュボードセクションから直接アクセスできません。このセクションはグローバルダッシュボード専用です。

3 Web シナリオ

概要

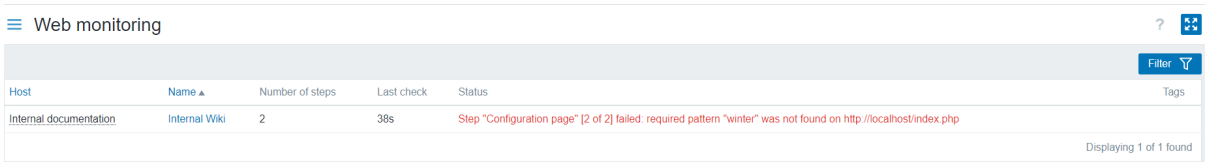
監視データ → ホストから、該当ホストの Web をクリックすることで、ホストのWeb シナリオ情報にアクセスできます。



ホスト名をクリックすると、ホストメニューが表示されます。無効化されたホストのデータにもアクセスできます。無効化されたホストの名前は赤で表示されます。

Web シナリオのステータスは以下のいずれかです:

- OK (緑) - すべてのステップが成功し、Zabbix がシナリオのデータを収集しました。
- エラー (赤) - 1 つ以上のステップが失敗しました（どのパラメータを修正する必要があるかのヒントはエラーメッセージを参照してください）。
- 不明 (灰色) - ステップのステータスがまだ利用できないか、データが収集されていません。



1 ページあたりに表示されるシナリオの最大数は、ユーザープロファイルの *[1* ページあたりの行数*]*設定によって決まります。

デフォルトでは、直近 24 時間以内の値のみが表示されます。この制限は、大量の最新データページの初期読み込み時間を改善する目的で導入されています。この期間を延長するには、管理 → 一般 → *GUI*メニューセクションの最大履歴表示期間パラメータの値を変更してください。

シナリオ名は、その詳細な統計情報へのリンクになっています:



フィルタの使用

このページには、選択したホストのすべての Web シナリオのリストが表示されます。監視データ → ホストページに戻らずに別のホストまたはホストグループの Web シナリオを表示するには、フィルターでそのホストまたはグループを選択します。タグに基づいてシナリオをフィルタリングすることもできます。

ボタン

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

3 最新データ

概要

監視 → 最新データセクションには、アイテムによって収集された最新の値が表示されます。

このセクションには、以下の要素が含まれます。

- フィルタ
- サブフィルタ
- アイテムリスト

Note:
サブフィルタとアイテムリストは、フィルタが設定され、表示する結果がある場合にのみ表示されます。

列	説明
ホスト	アイテムが属するホストの名前。 名前をクリックするとホストコンテキストメニューが表示されます。 ホストがメンテナンス中の場合、ホスト名の後にオレンジ色のレンチ アイコンが表示されます。 ホストが無効の場合、ホスト名は赤色で表示されます。無効なホストのデータ（グラフやアイテム値リストを含む）は最新データセクションでアクセス可能であることに注意してください。
名前	アイテムの名前。 名前をクリックするとアイテムメニューが表示されます。 説明があるアイテムには、アイテム名の横にクエスチョンマーク アイコンが表示されます。アイコンにマウスオーバーすると、アイテムの説明がツールチップで表示されます。
最終チェック	最後にアイテムをチェックしてからの経過時間。
最新の値	アイテムの最新値。 値は単位変換と値のマッピングが適用されて表示されます。値にマウスオーバーすると生データが表示されます。 デフォルトでは、過去 24 時間以内に受信した値のみが表示されます。この制限により、大量の最新データページの初期読み込み時間が短縮されます。制限を延長するには、管理 → 一般 → *GUI* の *Max history display period* パラメータの値を更新してください。
変化	前回の値と最新値の差。 更新間隔が 1 日以上のアイテムの場合、(デフォルト設定では）変化量は表示されません。この場合、最新値が 24 時間以上前に受信された場合は、最新値自体も表示されません。
タグ	アイテムに関連付けられているタグ。 アイテムリスト内のタグはクリック可能です。タグをクリックするとサブフィルタが有効になり、アイテムリストにはこのタグ（およびサブフィルタで以前に選択された他のタグ）を含むアイテムのみが表示されます。この方法でアイテムをフィルタリングした後は、アイテムリスト内のタグはクリックできなくなることに注意してください。タグに基づくさらなる変更（タグの削除や他のフィルタの指定など）は、サブフィルタで行う必要があります。
グラフ/履歴	アイテム値のシンプルグラフ/履歴へのリンク。
情報	アイテムに関する追加情報。 アイテムにエラーがある場合（たとえば、サポートされなくなった場合)、情報 アイコンが表示されます。アイコンにマウスオーバーすると詳細が表示されます。

ボタン

すべてのセクションで共通の表示モードボタンについては、監視のページで説明しています。

一括操作

リストの下のボタンは、選択した 1 つまたは複数のアイテムに対して一括操作を提供します。

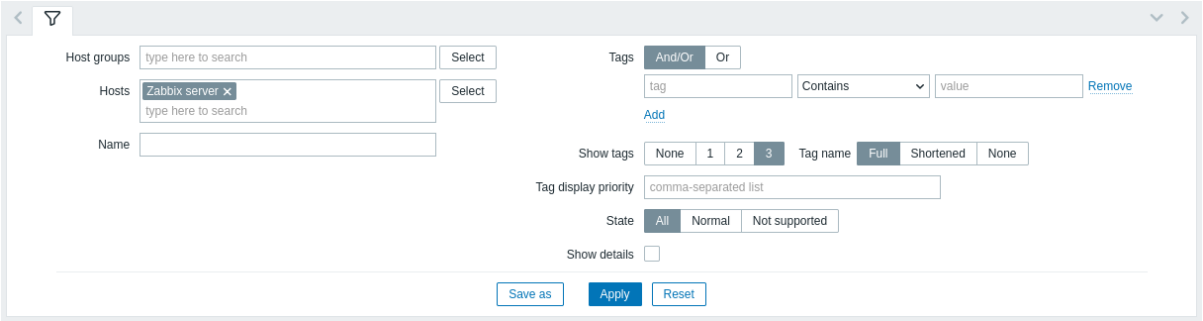
- 積み上げグラフを表示 - アドホックグラフの積み上げグラフを表示します。
- グラフを表示 - シンプルなアドホックグラフを表示します。
- 今すぐ実行 - 新しいアイテム値のチェックを即座に実行します。パッシブチェックのみサポートされています（詳細はこちら)。このオプションは、読み書きアクセス権を持つホストでのみ利用可能です。読み取り専用権限のホストでこのオプションにアクセスできるかどうかは、ユーザーロールの読み取り専用ホストで「今すぐ実行」を呼び出すオプションに依存します。

これらのオプションを使用するには、該当するアイテムの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるアイテムのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルター  アイコンは、アイテムリストとサブフィルターの上にあります。クリックするとフィルターが展開されます。



フィルターを使用すると、ホストグループ、ホスト、アイテム名、タグ、状態、その他の設定でアイテムリストを絞り込むことができます。フィルターで親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。タグによるフィルタリングの詳細については、監視 → 障害 を参照してください。

詳細を表示フィルターオプションを使用すると、アイテムに表示される情報を拡張できます。これをマークすると、アイテムの更新間隔、履歴とトレンドの設定、アイテムタイプ、アイテムエラー（正常/未サポート）などの詳細が表示されます。

フィルターの保存

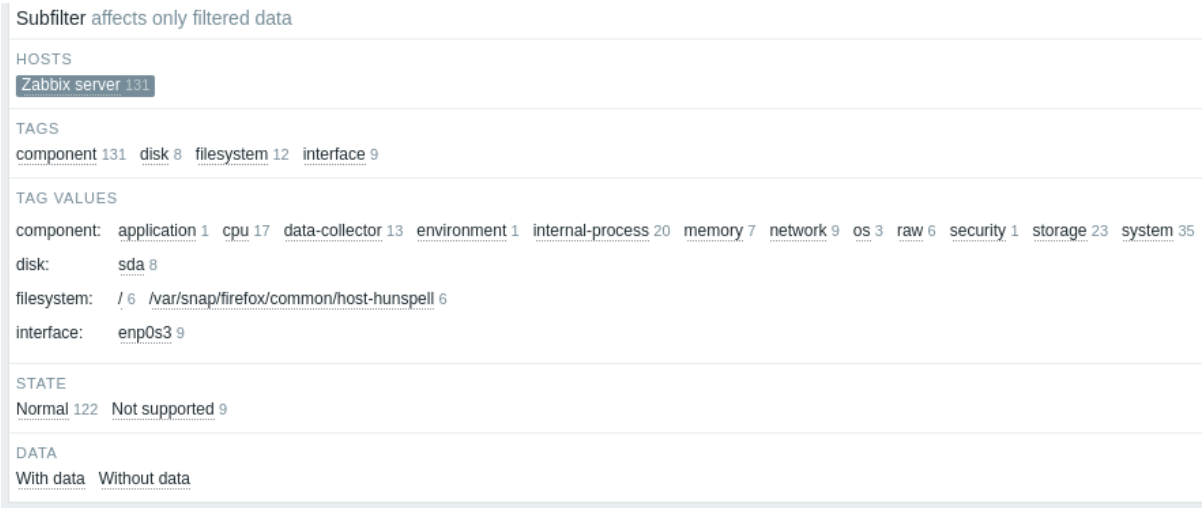
お気に入りのフィルター設定はタブとして保存でき、フィルターの上にある該当するタブをクリックすることで素早くアクセスできます。

フィルターの保存の詳細については、こちらをご覧ください。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターからのフィルタリングをさらに変更できます。

サブフィルターには、関連するアイテムへのクイックアクセス用のクリック可能なリンクが含まれています。アイテムは、ホスト、タグ名または値、アイテムの状態、データの状態などの共通のエンティティによって関連付けられています。エンティティをクリックすると、そのエンティティがグレーの背景でハイライトされ、アイテムが即座にフィルタリングされます（メインフィルターで適用をクリックする必要はありません）。別のエンティティをクリックすると、フィルタリングされた結果に追加されます。もう一度エンティティをクリックすると、フィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータ（最大 1000 レコードに制限）に基づいて生成されます。たとえば、20 台のホストにそれぞれ 100 個のアイテムがある場合（合計 2000 レコード）、サブフィルターにはホストの半分しか表示されません。サブフィルターでより多くのレコードを表示したい場合は、検索およびフィルター結果の制限パラメータ（管理 -> 一般 -> *GUI*）の値を増やす必要があります。

メインフィルターとは異なり、サブフィルターはテーブルのリフレッシュ要求ごとに更新され、利用可能なフィルタリングオプションとそのカウンター番号の最新情報が常に表示されます。

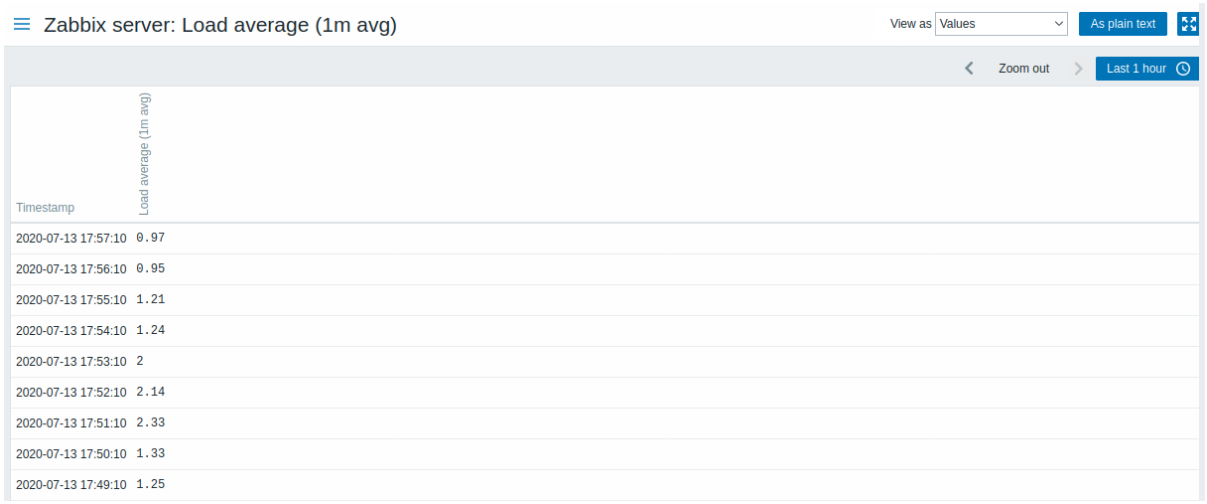
各エンティティグループ（ホスト、タグ、タグ値など）ごとに、最大 10 行のエンティティが表示されます。エンティティがさらに多い場合は、リストの最後にある三点リーダー ••• アイコンをクリックすることで、最大 1000 件（フロントエンド定義の SUBFILTER_VALUES_PER_GROUP の値）まで拡張できます。タグ値の場合は、最大 200 個のタグ名とそれに対応する値までリストを拡張できます。一度完全に展開すると、リストは折りたたむことができない点に注意してください。

各クリック可能なエンティティの横の数字は、それにグループ化されたアイテムの数（メインフィルターの結果に基づく）を示します。エンティティをクリックすると、他の利用可能なエンティティの数字がプラス記号付きで表示され、現在の選択に追加できるアイテムの数を示します。アイテムがないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

グラフと履歴

アイテムリストのグラフ/履歴列には、次のリンクがあります。

- 履歴 - すべてのテキストアイテムに対して、過去のアイテム値の履歴を表示するリスト (値/最新 *500* 件の値) に移動します。
- グラフ - すべての数値アイテムに対して、シンプルグラフに移動します。グラフが表示されている場合、右上のドロップダウンで値/最新 *500* 件の値に切り替えることもできます。



このリストに表示される値は生データであり、後処理は適用されません。

Note:
表示される値の合計数は、管理 → 一般 → *GUI*で設定されている検索およびフィルター結果の制限パラメータの値によって定義されます。

4 マップ

概要

監視データ → マップセクションではネットワークマップを構成、管理、および表示できます。

このセクションを開くと最後にアクセスしたマップ、またはアクセスできるすべてのマップのリストが表示されます。

すべてのマップは、パブリックまたはプライベートのいずれかになります。パブリックマップはすべてのユーザーが利用できますが、プライベートマップは、所有者とマップが共有されているユーザーのみがアクセスできます。

マップ一覧



表示されるデータ:

列	説明
名前	マップの名前。名前をクリックすると、マップを表示できます。
幅	マップの幅が表示されます。
高さ	マップの高さが表示されます。
操作	2 つの操作が利用可能です: プロパティ - 一般的なマップのプロパティを設定します 編集 - マップ要素を追加するためのグリッドにアクセスします

新しいマップを設定するには、右上のマップの作成ボタンをクリックします。YAML、XML、または JSON ファイルからマップをインポートするには、右上のインポートボタンをクリックします。マップをインポートしたユーザーが所有者として設定されます。

リストの下にある 2 つのボタンで一括編集オプションが利用できます:

- エクスポート - マップを YAML、XML、または JSON ファイルにエクスポートします
- 削除 - マップを削除します

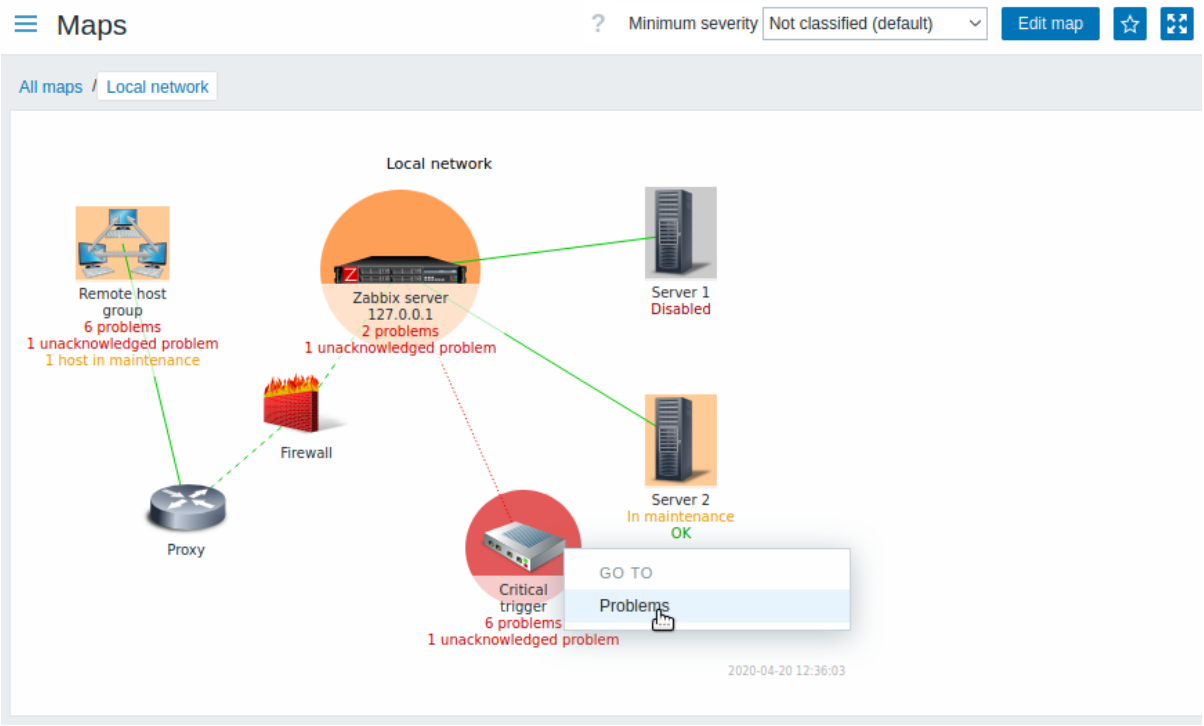
これらのオプションを使用するには、該当するマップの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のマップのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

マップの表示

マップを表示するには、マップのリストで表示したいマップの名前をクリックします。



マップのタイトルバーのドロップダウンを使用して、表示する問題トリガーの最低の重大度レベルを選択できます。*default* としてマークされた重大度は、マップ設定で設定されたレベルです。マップにサブマップが含まれている場合は、サブマップに移動すると上位レベルのマップの深刻度が保持されます（ただし、分類されていない場合を除き、深刻度はサブマップに渡されません）。

アイコンのハイライト

マップ要素が障害状態の場合、丸い円でハイライト表示されます。円の塗りつぶし色は、障害の深刻度の色に対応しています。選択した深刻度以上の障害のみが要素に表示されます。すべての障害が確認済みの場合、円の周囲に太い緑色の枠が表示されます。

さらに、以下のようなハイライト表示があります。

- メンテナンス中のホストは、オレンジ色の塗りつぶし四角でハイライトされます。なお、マップ要素がホストの場合、メンテナンスのハイライトは障害の深刻度のハイライトよりも優先されます。
- 無効（監視対象外）のホストは、グレーの塗りつぶし四角でハイライトされます。

ハイライト表示は、マップの設定でアイコンのハイライトチェックボックスをオンにした場合に表示されます。

最近の変更マーカー

要素の周りを内側に向けた赤い三角形は、最近のトリガーステータスの変更を示します。これは過去 30 分以内に発生したものです。これらの三角形は、マップ設定でトリガーステータス変更時に要素をマークするチェックボックスがマークされている場合に表示されます。

リンク

マップ要素をクリックすると、いくつかの利用可能なリンクが表示されたメニューが開きます。ホスト名をクリックすると、ホストメニューが表示されます。

ボタン

右側のボタンで以下のオプションを利用できます:

	マップの内容の編集に進みます。
	ダッシュボードの「お気に入り」ウィジェットにマップを追加します。
	マップはダッシュボードの「お気に入り」ウィジェットにあります。クリックすると「お気に入り」ウィジェットからマップを削除します。

表示モードのボタンはすべてのセクションで共通であり、監視ページで説明されています。

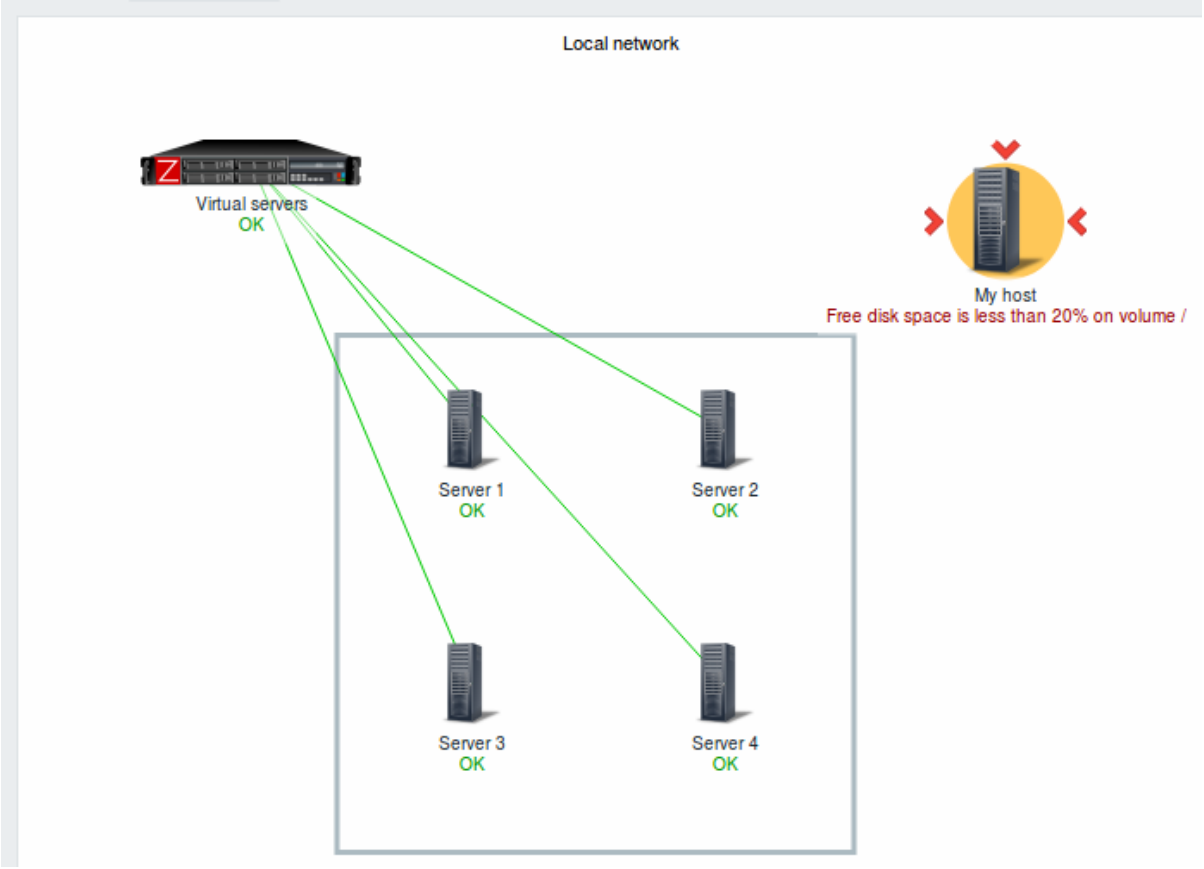
読みやすい要約地図

非表示の「aria-label」プロパティを使用すると、スクリーンリーダーで地図情報を読み取ることができます。一般的なマップの説明と個々の要素の説明の両方が、次の形式で利用できます。

- マップの説明：`<Map name>, <* of * items in problem state>, <* problems in total>.`
- 1 つの障害でで 1 つの要素を記述する：`<Element type>, Status <Element status>, <Element name>, <Problem description>.`
- 複数の障害がある 1 つの要素を記述する：`<Element type>, Status <Element status>, <Element name>, <* problems>.`
- 障害のない 1 つの要素を記述する：`<Element type>, Status <Element status>, <Element name>.`

たとえば、次のマップの場合以下のような説明が利用可能です。

```
'Local network, 1 of 6 elements in problem state, 1 problem in total. Host, Status problem, My host, Free
```



ネットワークマップの参照

ネットワークマップは sysmapid と mapname の両方の GET パラメータで参照できます。例えば、

```
http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network
```

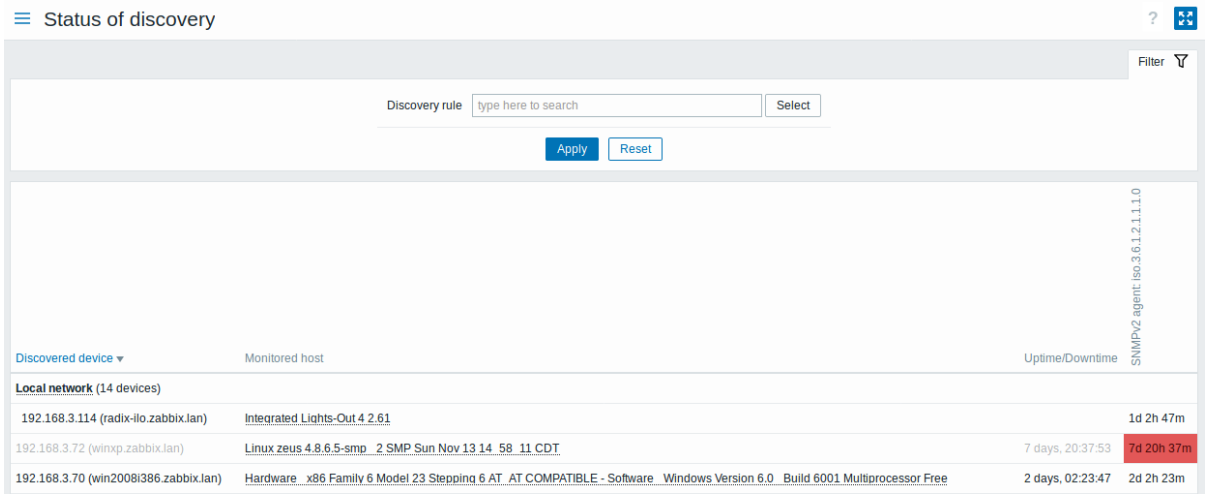
指定された名前のマップを開きます。(Local network)

sysmapid（マップ ID）と mapname（マップ名）の両方が指定されている場合、mapname の優先度が高くなります。

5 ディスカバリ

概要

監視 → 検出セクションでは、ネットワーク検出の結果が表示されます。検出されたデバイスは、検出ルールごとにソートされます。



表示されるデータ:

列	説明
検出されたデバイス	検出されたデバイスが、検出ルールごとにグループ化されて一覧表示されます。検出ルールをクリックすると、検出ルール設定フォームへのリンクを含むルールメニューが表示されます。
監視対象ホスト	すでに監視されているデバイスの場合、この列にホスト名が表示されます。 ホスト名をクリックすると、ホストメニューが表示されます。
稼働/停止時間	前回の検出以降にデバイスが検出または消失した期間がこの列に表示されます。
検出チェック	検出された各デバイスの個々のサービス (検出チェック) の状態が表示されます。赤いセルはサービスがダウンしていることを示します。サービスの稼働または停止時間はセル内に表示されます。 この列は、少なくとも 1 つの検出デバイスでサービスが見つかった場合にのみ表示されます。

ボタン

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

フィルタの使用

フィルタを使用して、任意のディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタで何も選択されていない場合、有効なすべてのディスカバリルールが表示されます。表示する特定のディスカバリルールを選択するには、フィルターにその名前を入力します。一致するすべての有効なディスカバリルールが選択のために一覧表示されます。複数のディスカバリルールを選択できます。

3 サービス

概要

サービスメニューは、Zabbix のサービスモニタリング機能です。

1 サービス

概要

このセクションでは、インフラストラクチャに基づいて Zabbix で設定されたすべてのサービスのハイレベルなステータスを確認できます。

サービスは、他の複数のレベルのサービス（「子」サービスと呼ばれる）で構成される階層である場合があり、これらはサービス全体のステータスに属性付けされます（サービス監視機能の概要も参照してください）。

サービスステータスの主なカテゴリは *OK* または問題であり、問題ステータスは対応する問題の深刻度名と色で表されます。

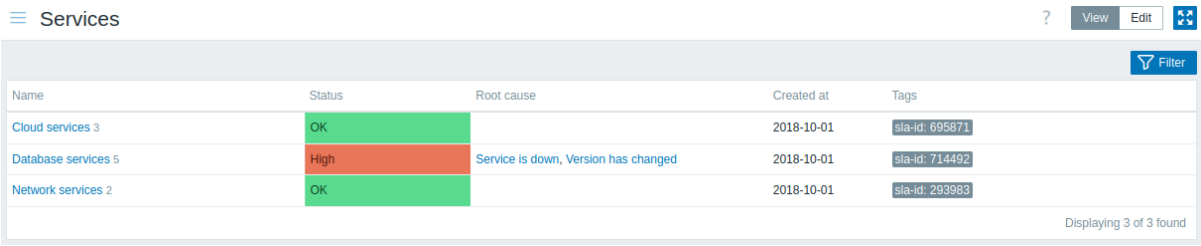
表示モードでは、サービスのステータスやその他の詳細を監視できますが、このセクションで編集モードに切り替えることで、サービス階層（サービス、子サービスの追加/編集）を設定することもできます。

表示モードから編集モード（およびその逆）に切り替えるには、右上隅の該当するボタンをクリックします。

- - サービスの表示
- - サービスおよび子サービスの追加/編集

編集へのアクセスはユーザーロールの設定に依存することに注意してください。

サービスの表示



既存のサービスのリストが表示されます。

表示されるデータ:

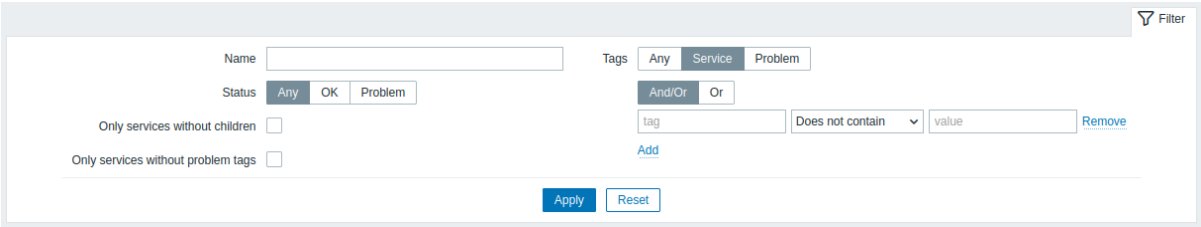
パラメータ	説明
名前	サービス名。 サービス名はサービスの詳細へのリンクです。 名前の後の数字は、そのサービスが持つ子サービスの数を示します。
ステータス	サービスのステータス: **OK** - 問題なし **<** 問題の色と深刻度 **>** - 問題とその深刻度を示します。複数の問題がある場合は、最も重大な問題の色と深刻度が表示されます。
根本原因	サービスのステータスに直接または間接的に影響を与える根本的な問題が一覧表示されます。 同じ問題が {SERVICE.ROOTCAUSE} マクロによって返されるものとして一覧表示されます。 問題名をクリックすると、監視データ → 障害で詳細を確認できます。 サービスのステータスに影響しない問題はリストに含まれません。
作成日時	サービスが作成された日時が表示されます。
タグ	サービスのタグが表示されます。タグは、サービスアクションやSLAでサービスを識別するために使用されます。

ボタン

すべてのセクションで共通の表示モードボタンについては、監視ページで説明しています。

フィルターの使用

フィルターを使用して、関心のあるサービスのみを表示できます。



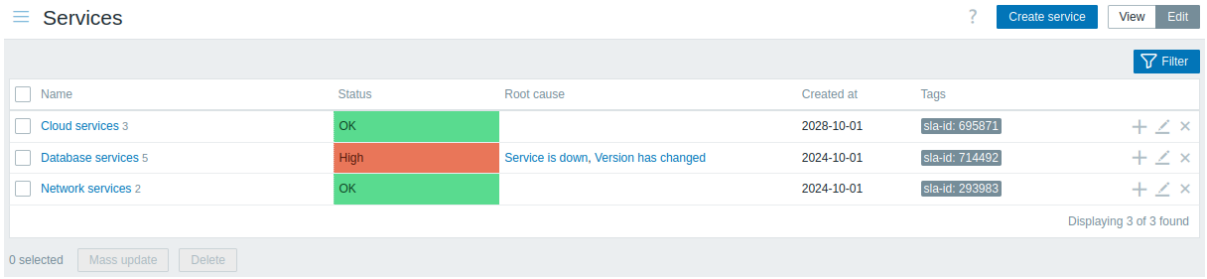
パラメータ	説明
名前	サービス名でフィルターします。
ステータス	サービスのステータスでフィルターします。
子サービスのないサービスのみ	チェックボックスをオンにすると、子サービスのないサービスのみが表示されます。このパラメータはサービスの編集モードでのみ利用可能です。
問題タグのないサービスのみ	チェックボックスをオンにすると、問題タグのないサービスのみが表示されます。このパラメータはサービスの編集モードでのみ利用可能です。

パラメータ	説明
タグ	サービスのタグ名と値、またはサービスの問題タグ名と値（サービスの編集モードの場合）でフィルターします。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります： 存在する - 指定したタグ名を含めます; 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）； 含む - タグ値が入力した文字列を含むタグ名を含めます（部分一致、大文字と小文字を区別しない）； 存在しない - 指定したタグ名を除外します; 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）； 含まない - タグ値が入力した文字列を含むタグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります： **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

サービスの編集

編集ボタンをクリックすると、編集モードに入ります。編集モードでは、リストの各エントリの前にチェックボックスが追加され、さらに以下のオプションが利用できます:

- - このサービスに子サービスを追加
- - このサービスを編集
- - このサービスを削除



新しいサービスを設定するには、右上のサービスの作成ボタンをクリックします。

サービスの詳細

サービスの詳細にアクセスするには、サービス名をクリックします。すべてのサービスのリストに戻るには、すべてのサービスをクリックします。

サービスの詳細には、情報ボックスと子サービスのリストが含まれます。

情報ボックスにアクセスするには、情報タブをクリックします。情報ボックスには以下の項目が含まれます。

- 親サービス名（存在する場合）
- このサービスの現在のステータス
- このサービスの現在の SLA（SLA 名: サービスレベル指標の形式）。'SLA 名' はこのサービスの SLA レポートへのリンクにもなっています。サービスレベル指標（SLI）の横にある情報ボックスにマウスを合わせると、SLI の詳細を示すポップアップ情報リストが表示されます。サービスレベル指標は、現在のサービスレベルをパーセンテージで表示します。
- サービスタグ

情報ボックスにはサービス設定へのリンクも含まれています。

子サービスのフィルターを使用するには、フィルタータブをクリックします。

編集モードでは、子サービスのリストに追加の編集オプションが表示されます。

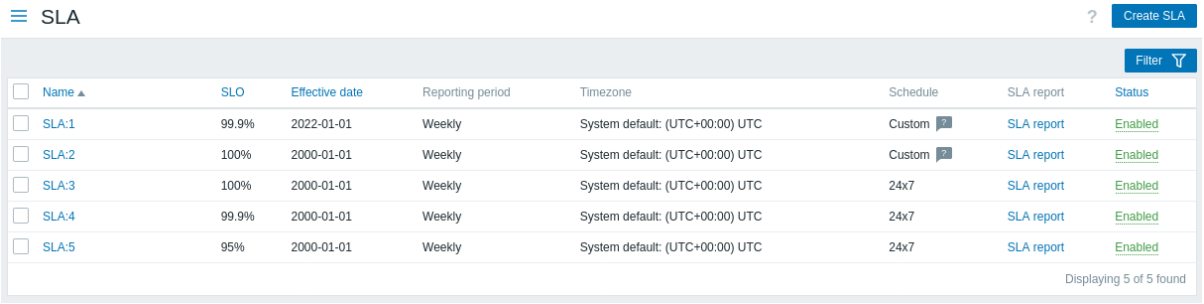
- - このサービスに子サービスを追加
- - このサービスを編集
- - このサービスを削除

2 SLA

概要

このセクションでは SLA を表示および設定することができます。

SLAs



設定された SLAs のリストが表示されます。注ユーザーがアクセスできるサービスに関連する SLAs のみが表示されることに注意してください。(ユーザーロールに対して * SLA の管理 * が有効になっていない限り、読み取り専用として表示されます)

表示情報:

パラメータ	説明
名前	SLA 名が表示されます。 名前はSLA の設定へのリンクです。
SLO	サービスレベル目標（SLO）が表示されます。
発行日	SLA 計算の開始日が表示されます
レポート期間	SLA レポートで使用される期間が表示されます - 毎日、毎週、毎月、四半期、または毎年
タイムゾーン	SLA タイムゾーンが表示されます
スケジュール	SLA スケジュールが表示されます - 24 時間 365 日またはカスタム
SLA レポート	リンクをクリックして、この SLA の SLA レポートを表示します
ステータス	SLA ステータスが表示されます - 有効または無効

3 SLA レポート

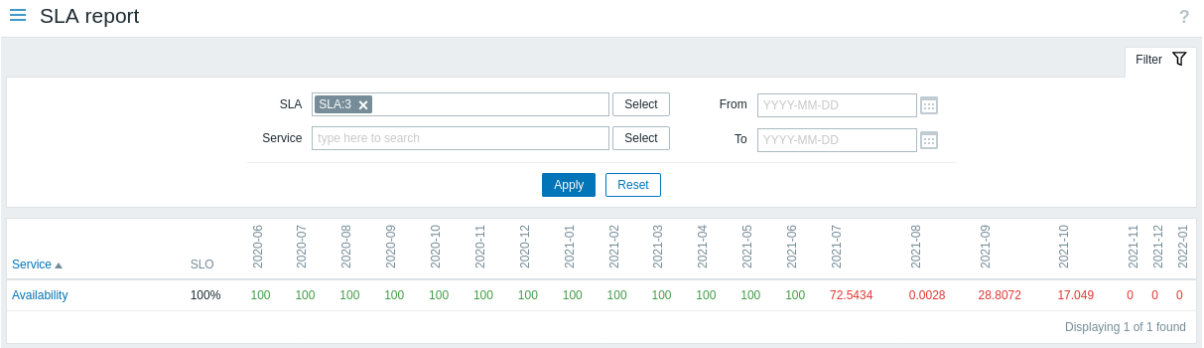
概要

このセクションでは、フィルタで選択した条件に基づいてSLAレポートを表示できます。

SLA レポートは、ダッシュボードウィジェットとしても表示できます。

レポート

フィルタを使用すると SLA 名とサービス名に基づいてレポートを選択できます。表示期間を絞り込む事も可能です。



各列（期間）には、その期間の SLI が表示されます。セット SLO に違反している SLI は、赤で強調表示されます。

レポートには 20 の期間が表示されます。開始日と終了日の両方を指定した場合、最大 100 の期間を表示できます。

レポートの詳細

レポート内のサービス名をクリックすると、より詳細なビューを表示する別のレポートにアクセスできます。

負の障害時間は、SLA の計算やレポートには影響しません。

4 インベントリ

概要

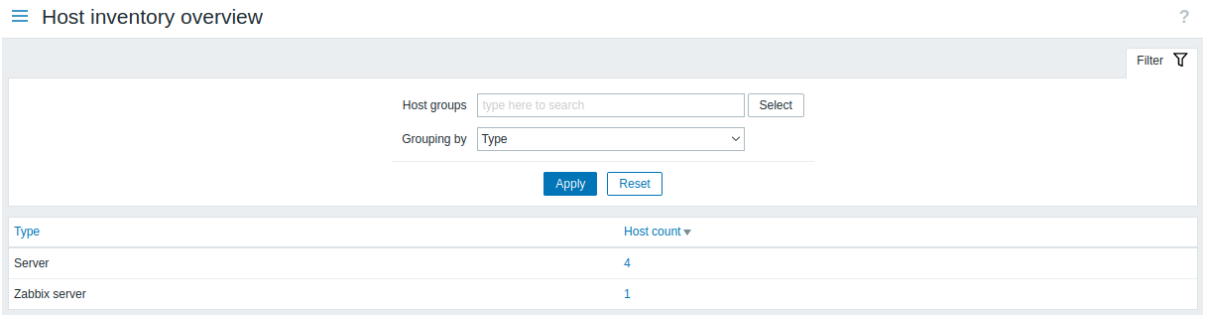
インベントリメニューには、選択したパラメータごとのホストインベントリデータの概要と、ホストインベントリの詳細を表示するセクションがあります。

1 概要

概要

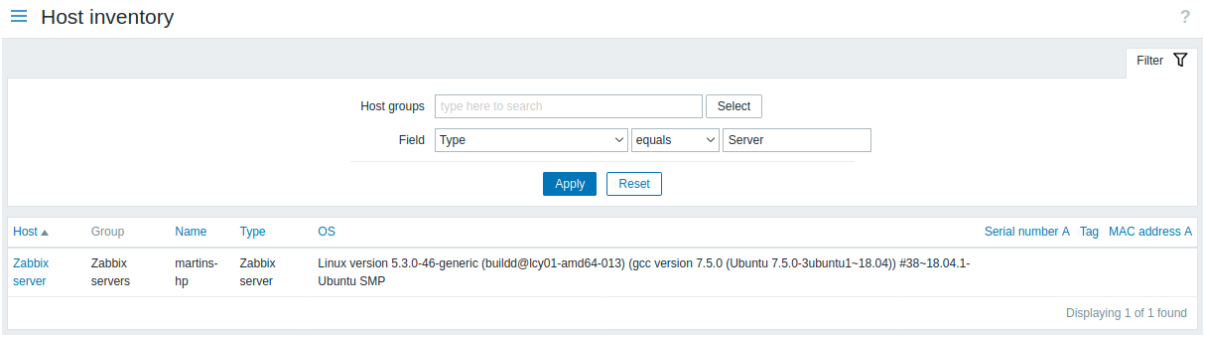
インベントリ → 概要セクションではホストインベントリデータの概要を確認する方法を提供します。

表示する概要については、ホストグループ（またはなし）とデータを表示するインベントリフィールドを選択します。選択したフィールドの各エントリに対応するホストの数が表示されます。



概要の完全性は、ホストで維持されるインベントリ情報の量によって異なります。

ホスト数列の数字はリンクです。これらのホストは *Host Inventory* テーブルで除外されます。

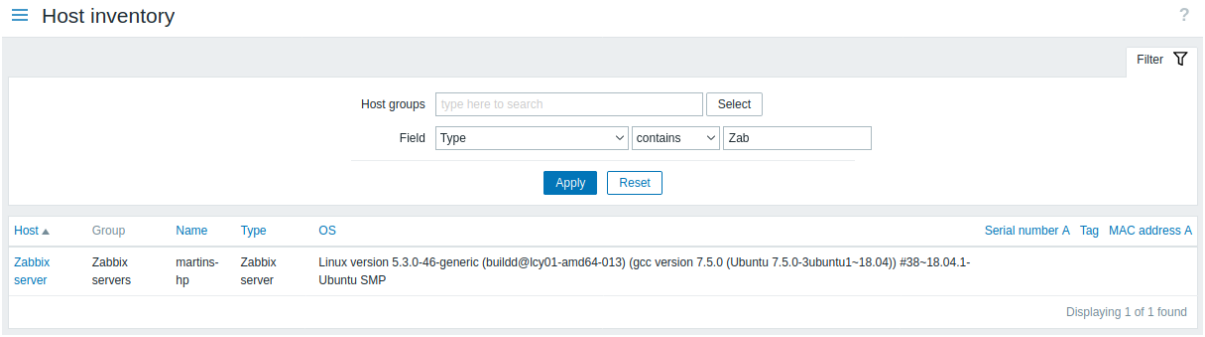


2 ホスト

概要

インベントリ → ホストセクションでは、ホストのインベントリデータが表示されます。

ホストグループや任意のインベントリフィールドでホストをフィルタリングし、関心のあるホストのみを表示できます。

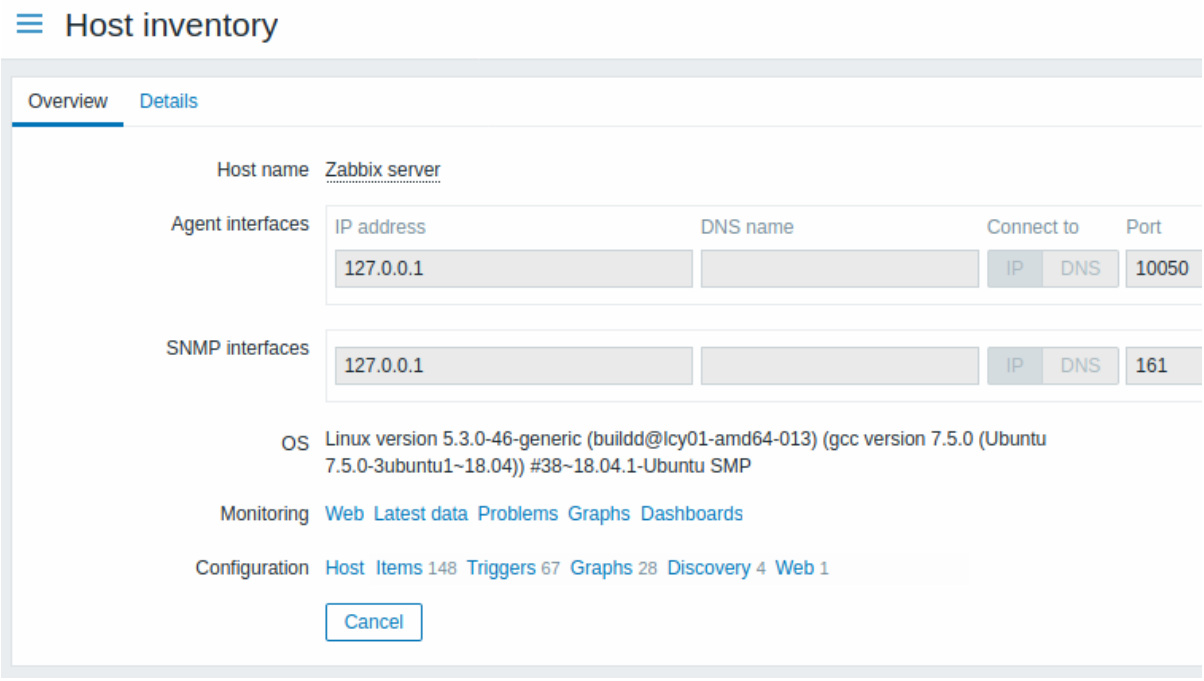


すべてのホストインベントリを表示するには、「リセット」ボタンを押します。

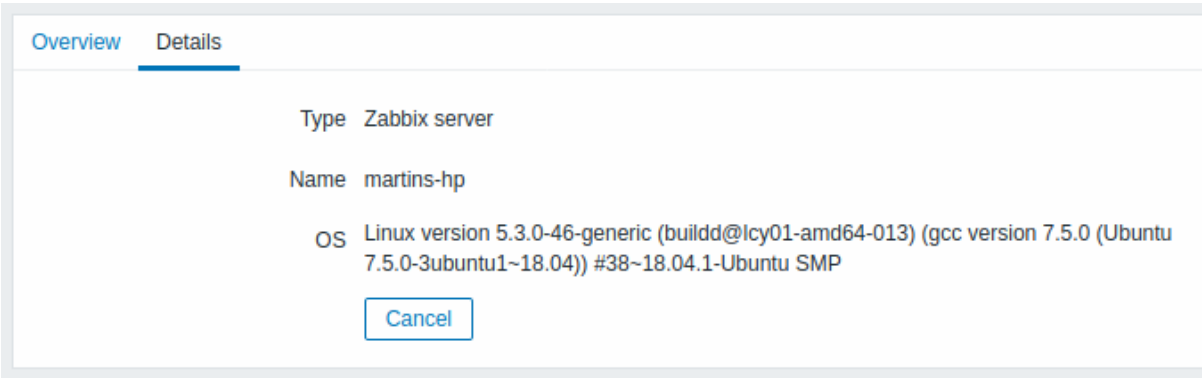
テーブルには主要なインベントリフィールドのみが表示されますが、そのホストのすべてのインベントリ情報を表示することもできます。その場合は、最初の列のホスト名をクリックしてください。

インベントリの詳細

概要タブには、ホストに関する一般的な情報、最新の監視データ、およびホストの設定オプションが含まれています。



詳細タブには、ホストのすべての利用可能なインベントリの詳細が含まれています。



インベントリデータの完全性は、ホストで管理されているインベントリ情報の量によって異なります。情報が管理されていない場合、詳細タブは無効になります。

5 レポート

概要

レポートメニューには、システム情報、トリガー、収集されたデータなどのパラメーターの表示に焦点を当てた、事前定義されたユーザーがカスタマイズ可能なさまざまなレポートを含むいくつかのセクションがあります。

1 システム情報

概要

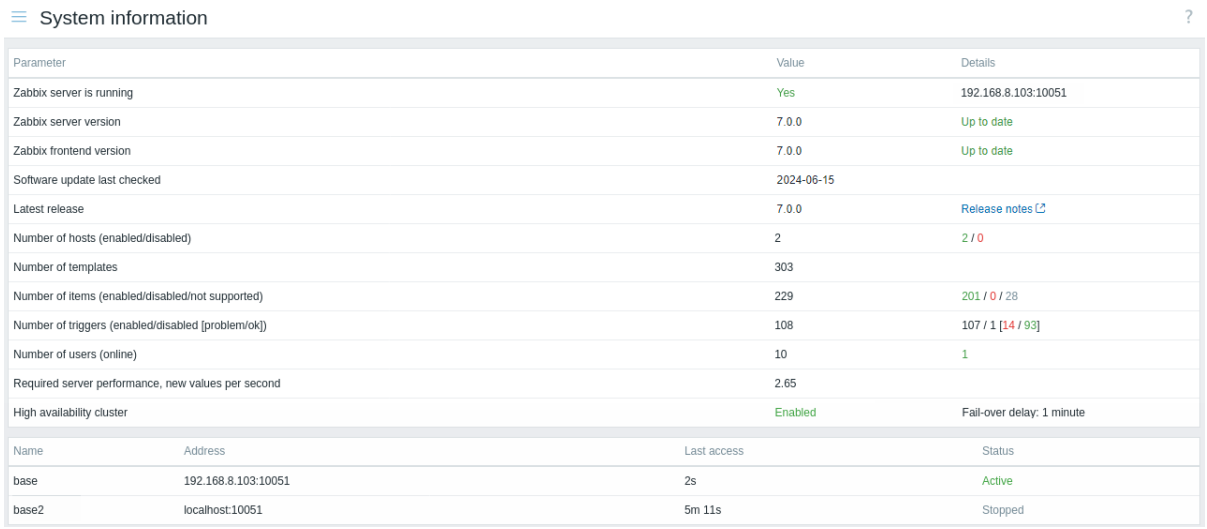
レポート → システム情報では、Zabbix サーバーとシステムの主要なデータの概要が表示されます。システムデータは内部アイテムを使用して収集されます。

高可用性構成の場合、システム情報のソース（サーバーインスタンス）をリダイレクトすることが可能です。そのためには、*zabbix.conf.php* ファイルを編集し、$ZBX_SERVER または $ZBX_SERVER と $ZBX_SERVER_PORT の両方を、アクティブとして表示されているもの以外のサーバーに設定します。$ZBX_SERVER のみを設定した場合、$ZBX_SERVER_PORT にはデフォルト値（10051）が使用されることに注意してください。

高可用性構成が有効になっている場合、システム統計情報の下に高可用性ノードの詳細を含む別のブロックが表示されます。このブロックは Zabbix のスーパ管理者ユーザーのみが表示できます。

システム情報は、ダッシュボードのウィジェットとしても利用できます。

システム統計



表示されるデータ:

パラメータ	値	詳細
Zabbix サーバーが稼働中	Zabbix サーバーのステータス: **Yes** - サーバーが稼働中 **No** - サーバーが稼働していない 注: 残りの情報を表示するには、Web フロントエンドがサーバーを稼働させている必要があり、サーバーで少なくとも 1 つのトラッパープロセスが起動している必要があります (zabbix_server.confファイルの StartTrappers パラメータ > 0)。	Zabbix サーバーの場所とポート。
Zabbix サーバーバージョン	現在のサーバーバージョン番号が表示されます。 注: Zabbix サーバーが稼働中の場合のみ表示されます。	サーバーバージョンのステータスが表示されます: **Up to date** - 最新バージョンを使用中; **New update available** - より新しいバージョンが利用可能; **Outdated** - このバージョンのフルサポート期間が終了しています。 この情報は、Zabbix サーバーの設定でソフトウェアアップデートチェックが有効になっている場合のみ利用可能です。最後のソフトウェアアップデートチェックが 1 週間以上前に実行された場合や、現在のバージョンに関するデータが存在しない場合は何も表示されません。
Zabbix フロントエンドバージョン	Zabbix フロントエンドのバージョン番号が表示されます。	Zabbix フロントエンドバージョンのステータスが表示されます: **Up to date** - 最新バージョンを使用中; **New update available** - より新しいバージョンが利用可能; **Outdated** - このバージョンのフルサポート期間が終了しています。 この情報は、Zabbix サーバーの設定でソフトウェアアップデートチェックが有効になっている場合のみ利用可能です。最後のソフトウェアアップデートチェックが 1 週間以上前に実行された場合や、現在のバージョンに関するデータが存在しない場合は何も表示されません。
ソフトウェアアップデートの最終チェック	Zabbix ソフトウェアアップデートの最終チェック日が表示されます。 この情報は、Zabbix サーバーの設定でソフトウェアアップデートチェックが有効になっている場合のみ利用可能です。	

パラメータ	値	詳細
最新リリース	現在の Zabbix バージョンに対して利用可能な新しいリリース番号が表示されます (存在する場合)。 この情報は、Zabbix サーバーの設定でソフトウェアアップデートチェックが有効になっている場合のみ利用可能です。最後のソフトウェアアップデートチェックが 1 週間以上前に実行された場合や、現在のバージョンに関するデータが存在しない場合は何も表示されません。	利用可能な最新の Zabbix リリースのリリースノートへのリンクが表示されます。
ホスト数	設定されているホストの総数が表示されます。	監視中/監視外のホスト数。
テンプレート数	テンプレートの総数が表示されます。	
アイテム数	アイテムの総数が表示されます。	監視中/無効/サポート外のホストレベルアイテム数。 無効なホスト上のアイテムは無効としてカウントされます。
トリガー数	トリガーの総数が表示されます。	有効/無効なホストレベルトリガー数; 有効なトリガーの「障害」/「正常」状態ごとの内訳。 「正常」状態にリストされているトリガーには、「不明」ステータスのトリガーも含まれます。 無効なアイテムに依存しているトリガーや、無効なホストに割り当てられているトリガーは無効としてカウントされます。
ユーザー数	設定されているユーザーの総数が表示されます。	オンラインユーザー数。
必要なサーバーパフォーマンス *(新規値/秒)*	Zabbix サーバーが 1 秒あたりに処理する新しい値の予想数が表示されます。	必要なサーバーパフォーマンスは推定値であり、ガイドラインとして役立ちます。実際に処理された値の正確な数値については、`zabbix[wcache,values,all]` 内部アイテムを使用してください。 監視中のホストの有効なアイテムが計算に含まれます。ログアイテムは、アイテムの更新間隔ごとに 1 つの値としてカウントされます。通常の間隔値がカウントされ、柔軟な間隔やスケジューリング間隔値はカウントされません。「nodata」メンテナンス期間中も計算は調整されません。トラッパーアイテムはカウントされません。
Zabbix サーバー上のグローバルスクリプト	サーバー設定で EnableGlobalScripts=0 を設定して Zabbix サーバー上のグローバルスクリプトが無効になっている場合、このフィールドに **Disabled** と表示されます。	
高可用性クラスタ	Zabbix サーバーの高可用性クラスタのステータス: **Disabled** - スタンドアロンのサーバー **Enabled** - 少なくとも 1 つの高可用性ノードが存在	有効な場合、フェイルオーバー遅延が表示されます。

システム情報は、以下の条件でもエラーメッセージを表示します:

- 使用しているデータベースが必要な文字セットまたは照合順序 (UTF-8) を持っていない場合。
- データベースのバージョンがサポートされている範囲より下または上の場合 (「スーパ管理者ロール」タイプのユーザーのみ利用可能)。
- TimescaleDBのハウスキーピングが正しく設定されていない場合 (ヒストリまたはトレンドテーブルに圧縮チャンクが含まれているが、アイテムヒストリ期間の上書きまたはアイテムトレンド期間の上書きオプションが無効になっている場合)。

高可用性ノード

高可用性クラスタが有効になっている場合、各高可用性ノードのステータスを表示する別のデータブロックが表示されます。



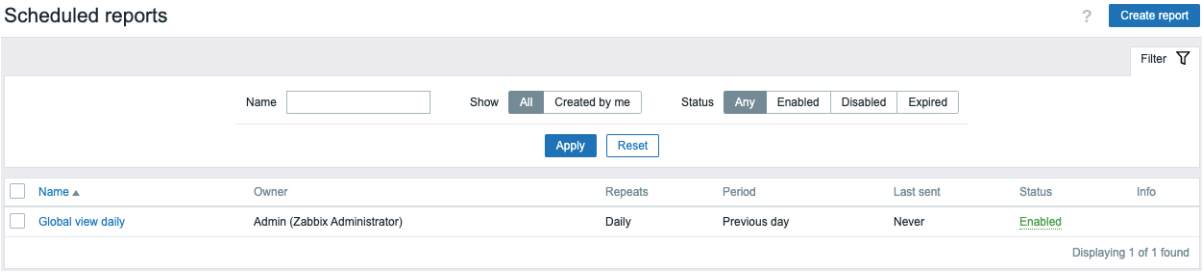
表示されるデータ:

列	説明
名前	サーバー設定で定義されたノード名。
アドレス	ノードの IP アドレスとポート。
最終アクセス	ノードの最終アクセス時刻。 セルにカーソルを合わせると、最終アクセスのタイムスタンプが長い形式で表示されます。
ステータス	ノードのステータス。テーブルの行は、以下の優先順位でこれらのステータスでソートされます: アクティブ - ノードが稼働中で動作している 利用不可 - ノードがフェイルオーバー遅延を超えて見つからない（原因を調査する必要があります） 停止 - ノードが停止しているか、起動できなかった（起動するか削除する必要があります） スタンバイ - ノードが稼働中で待機している

2 定期レポート

概要

レポート → スケジュールレポートでは、十分な権限を持つユーザーが、ダッシュボードの PDF バージョンを定期的に生成し、指定した受信者にメールで送信するように設定できます。



初期画面にはスケジュールレポートの情報が表示され、簡単にナビゲーションできるようにフィルタリングできます。詳細は下記のフィルターの使用セクションを参照してください。

表示されるデータ:

列	説明
名前	レポートの名前。クリックするとレポートの設定フォームが開きます。
所有者	レポートを作成したユーザー。
繰り返し	レポート生成の頻度（日次/週次/月次/年次）。
期間	レポートが作成される期間。
最終送信	最新のレポートが送信された日時。
ステータス	レポートの現在のステータス（有効/無効/期限切れ）。十分な権限を持つユーザーは、クリックしてステータスを変更できます -「有効」から「無効」へ（およびその逆）、「期限切れ」から「無効」へ（およびその逆）。権限が不十分なユーザーの場合、ステータスはクリックできません。
情報	情報アイコンを表示します: 赤いアイコンはレポートの生成に失敗したことを示します。マウスオーバーするとエラー情報のツールチップが表示されます。 黄色いアイコンはレポートは生成されたが、一部（またはすべて）の受信者への送信に失敗した、またはレポートが期限切れであることを示します。マウスオーバーすると追加情報のツールチップが表示されます。

フィルターの使用

フィルターを使用して、レポートのリストを絞り込むことができます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

利用可能なフィルターオプションは次のとおりです。

- 名前 - 部分一致が可能
- 表示 - 現在のユーザーが作成したレポートまたはすべてのレポート
- ステータス - 「すべて」(すべてのレポートを表示)、「有効」、「無効」、「期限切れ」から選択

フィルターは定期レポートセクション名の下にあります。右上のフィルタータブをクリックすることで、フィルターを開いたり折りたたんだりできます。

一括更新

複数のレポートのステータスを一度に削除または変更したい場合があります。その場合、各レポートを個別に編集する代わりに、一括更新機能を使用できます。

複数のレポートを一括更新するには、以下の手順を実行します。

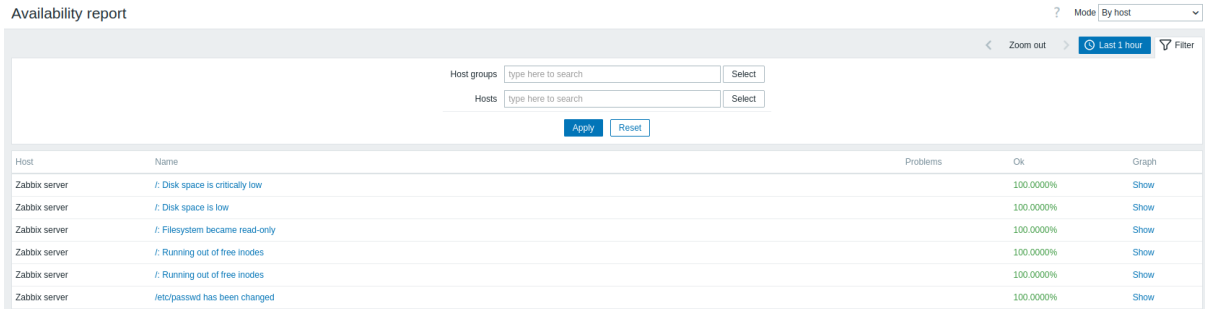
- 一覧で更新したいレポートのチェックボックスをオンにします
- 一覧の下にある必要なボタン（有効、無効、削除）をクリックして変更を行います

3 稼働レポート

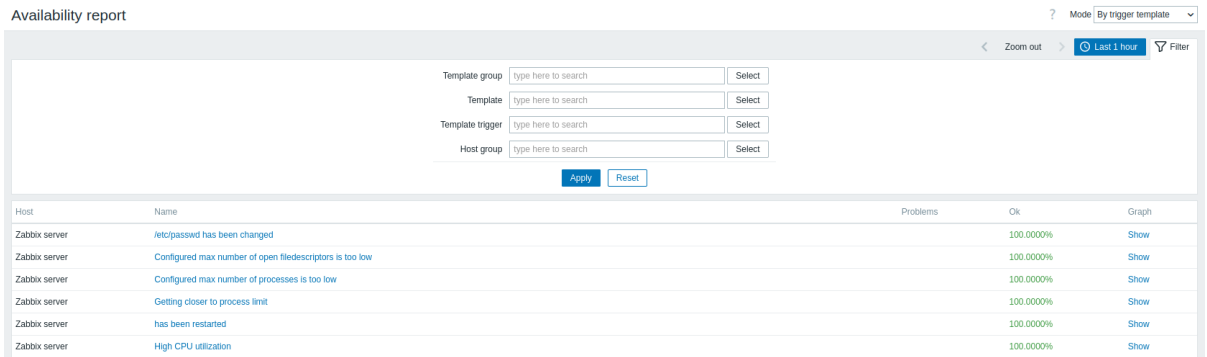
概要

レポート > 可用性レポートでは、各トリガーが問題/正常状態にあった時間の割合を確認できます。

各状態ごとに時間の割合が表示されるため、システム上のさまざまな要素の可用性を簡単に判断できます。



右上のドロップダウンで選択モードを選択できます。ホストごとにトリガーを表示するか、テンプレートに属するトリガーごとに表示するかを選択できます。



トリガー名は、そのトリガーの最新イベントへのリンクです。

フィルターの使用

フィルターは、表示されるホストやトリガーの数を絞り込むのに役立ちます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターは可用性レポートセクション名の下にあります。右側のフィルタータブをクリックすることで、フィルターを開いたり折りたたんだりできます。

トリガーテンプレートによるフィルタリング

トリガーテンプレートによるモードでは、以下のいずれかまたは複数のパラメータで結果をフィルタリングできます。

パラメータ	説明
テンプレートグループ	選択したテンプレートグループに属するテンプレートから継承されたトリガーによってホストをフィルタリングします。 親テンプレートグループを指定すると、すべてのネストされたテンプレートグループが暗黙的に選択されます。
テンプレート	選択したテンプレート（ネストされたテンプレートを含む）から継承されたトリガーによってホストをフィルタリングします。 ネストされたテンプレートに独自のトリガーがある場合、それらのトリガーは表示されません。
テンプレートトリガー	選択したトリガーによってホストをフィルタリングします。フィルタリングされたホストの他のトリガーは表示されません。
ホストグループ	選択したホストグループに属するホストをフィルタリングします。

ホストによるフィルタリング

ホストによるモードでは、ホストまたはホストグループで結果をフィルタリングできます。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。

期間セレクター

期間セレクターを使用すると、よく使う期間をワンクリックで選択できます。セレクターは、フィルターの横にある期間タブをクリックすることで展開および折りたたみができます。

グラフ列の表示をクリックすると、各バーが当年の経過した週を表す棒グラフで可用性情報が表示されます。

バーの緑色は OK の時間、赤色は問題の時間を表します。

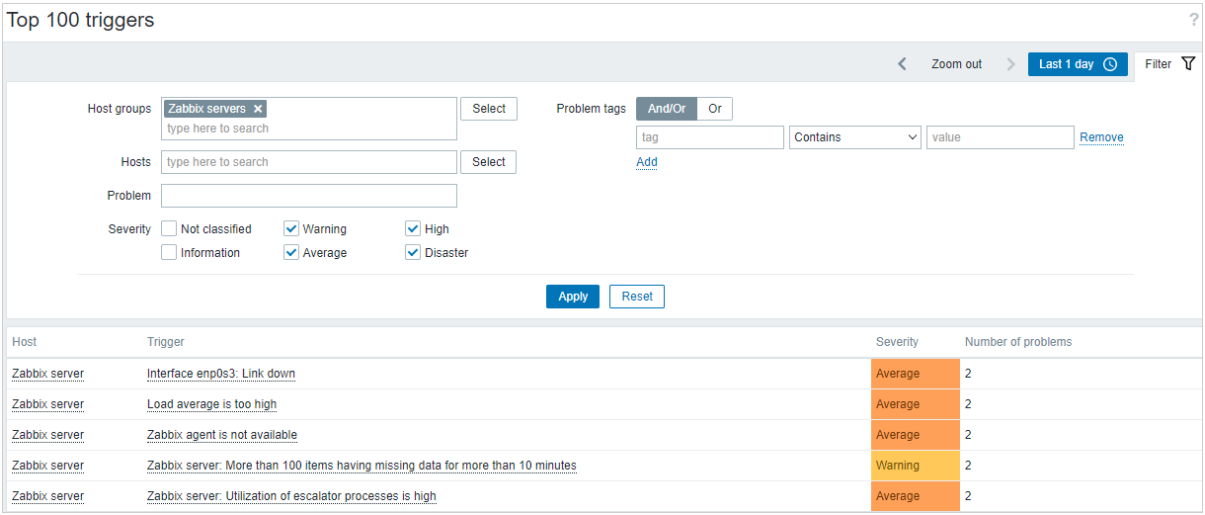
メンテナンス期間の影響

メンテナンスは、可用性レポートから自動的に時間を除外しません。データ収集なしとして設定されたメンテナンスのみがポーリングを停止し（そのため障害が生成されない）、影響を受けるトリガーの可用性パーセンテージが変更されません。

4 障害発生数上位 100 トリガー

概要

レポート → トリガートップ *100* では、選択した期間中に検出された問題の数が最も多いトリガーを確認できます。



ホストおよびトリガーの列のエントリは、いくつかの便利なオプションを提供するリンクになっています。

- ホストの場合 - ホスト名をクリックするとホストメニューが表示されます
- トリガーの場合 - トリガー名をクリックすると、最新のイベント、各トリガーアイテムのシンプルグラフ、トリガー自体および各トリガーアイテムの設定フォームへのリンクが表示されます

フィルターの使用

フィルターを使用して、ホストグループ、ホスト、問題名、タグ、またはトリガーの深刻度でトリガーを表示できます。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターはトリガートップ *100* セクション名の下にあります。右側のフィルタータブをクリックすると、フィルターを開いたり折りたたんだりできます。

期間セレクター

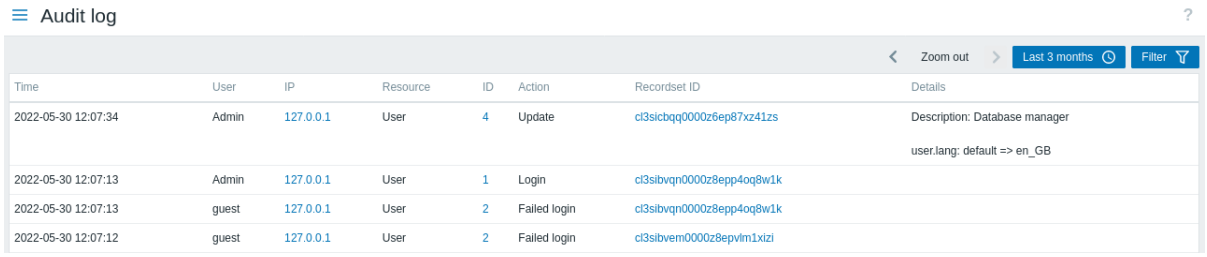
フィルターの横にある期間セレクターを使用すると、よく使用する期間をワンクリックで選択できます。詳細については、期間およびホストセレクターを参照してください。

5 監査ログ

概要

レポート → 監査ログセクションでは、ユーザーおよびシステムのアクティビティの記録を表示できます。

Note:
監査記録を収集して表示するには、管理 → 監査ログ セクションで 監査ログを有効にするチェックボックスをオンにする必要があります。この設定を有効にしないと、アクティビティの履歴はデータベースに記録されず、監査ログにも表示されません。



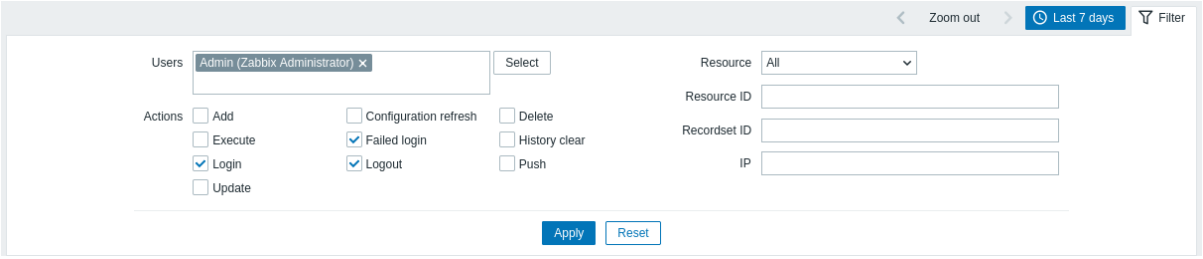
監査ログには以下のデータが表示されます:

列	説明
時刻	監査記録のタイムスタンプ。
ユーザー	アクティビティを実行したユーザー。
IP	アクティビティが開始された IP。 ハイパーリンクをクリックすると、この IP で監査ログ記録がフィルタリングされます。
リソース	影響を受けたリソースのタイプ（*API* トークン、アクション、認証、自動登録など）。
ID	影響を受けたリソースの ID。 ハイパーリンクをクリックすると、このリソース ID で監査ログ記録がフィルタリングされます。
アクション	アクティビティのタイプ（追加、設定の更新、削除、実行、ログイン失敗、履歴クリア、ログイン、ログアウト、プッシュ、更新）。
レコードセット *ID*	同じ操作の結果として作成されたすべての監査ログ記録に共通の ID。 たとえば、テンプレートをホストにリンクする場合、継承されたテンプレートエンティティ（アイテム、トリガーなど）ごとに個別の監査ログ記録が作成されます。これらすべての記録には同じレコードセット *ID* が付きます。 ハイパーリンクをクリックすると、このレコードセット *ID* で監査ログ記録がフィルタリングされます。
詳細	リソースの説明と実行されたアクティビティの詳細情報。 記録に 2 行を超える情報が含まれている場合は、追加の詳細リンクが表示されます。このリンクをクリックすると、変更の全リストを表示できます。

Note:
トラッパーアイテムまたは（トラップを有効にした）HTTP エージェントアイテムがデータを受信した場合、監査ログにエントリが追加されるのは、データが`history.push` API メソッドを使用して送信された場合のみであり、Zabbix senderユーティリティを使用した場合は追加されません。

フィルターの使用

フィルターは監査ログバーの下にあります。右上隅のフィルタータブをクリックすると、フィルターを開いたり折りたたんだりできます。



フィルターを使用して、ユーザー、影響を受けたリソース、リソース ID、実行された操作（レコードセット *ID*）、IP でレコードを絞り込むことができます。リソースによっては、フィルターで 1 つまたは複数の特定のアクションを選択できます。

検索パフォーマンスを向上させるために、すべてのデータはマクロを解決せずに検索されます。

期間セレクター

フィルターの横にある期間セレクターを使用すると、よく使用する期間をワンクリックで選択できます。詳細については、期間およびホストセレクターを参照してください。

6 アクションログ

概要

レポート → アクションログでは、アクション内で実行されたオペレーション（通知、リモートコマンド）の詳細を確認できます。

表示されるデータ:

列	説明
時刻	オペレーションのタイムスタンプ。
アクション	オペレーションを引き起こしたアクションの名前。
メディアタイプ	通知送信に使用されたメディアタイプ（例: Email、Jira など）。 リモートコマンドを実行したオペレーションの場合、この列は空になります。
送信先	通知の送信先に関する情報 - ユーザー名、氏名（括弧内）、およびメディアタイプに応じた追加情報（メールアドレス、ユーザー名など）。 リモートコマンドを実行したオペレーションの場合、この列は空になります。
メッセージ	メッセージの内容、リモートコマンドまたはグローバルスクリプト名。 リモートコマンドはターゲットホストとコロン記号で区切られます: `<host>:<command>`。たとえば、リモートコマンドが Zabbix サーバーで実行された場合、情報は次の形式になります: `Zabbix server:<command>`。
ステータス	オペレーションのステータス: 進行中 - 通知送信のオペレーションが進行中（サーバーが通知送信を試行する残り回数も表示） 送信済み - 通知が送信された 実行済み - リモートコマンドが実行された 失敗 - オペレーションが完了しなかった
情報	オペレーション実行に関するエラー情報（ある場合）。

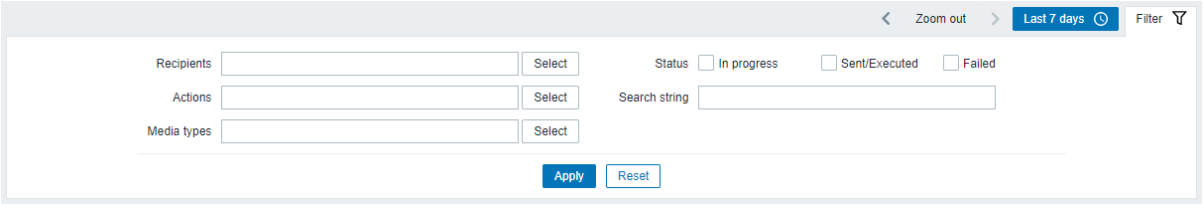
ボタン

ページの右上隅にあるボタンには、次のオプションがあります。

Export to CSV	現在のページのアクションログレコードを CSV ファイルにエクスポートします。フィルターが適用されている場合は、フィルターされたレコードのみがエクスポートされます。 エクスポートされた CSV ファイルでは、「Recipient」と「Message」列が複数の列に分割されます - 「Recipient's Zabbix username」、「Recipient's name」、「Recipient's surname」、「Recipient」、および「Subject」、「Message」、「Command」。

フィルターの使用

フィルターはアクションログバーの下にあります。ページの右上隅にあるフィルタータブをクリックすると、フィルターを開いたり折りたたんだりできます。



フィルターを使用して、通知の受信者、アクション、メディアタイプ、ステータス、またはメッセージ/リモートコマンドの内容 (検索文字列) でレコードを絞り込むことができます。検索パフォーマンスを向上させるために、データはマクロを展開せずに検索されます。

期間セレクター

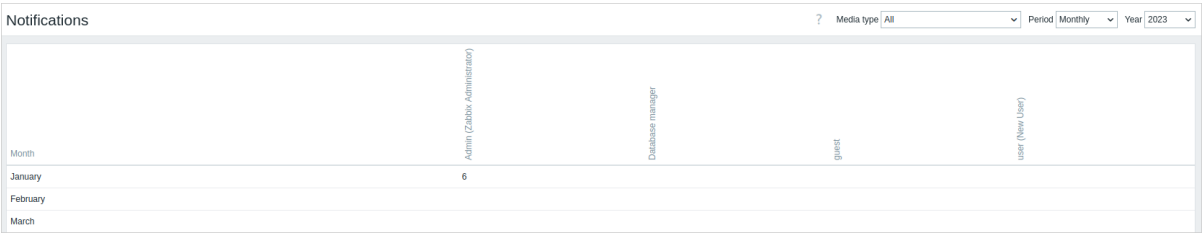
フィルターの横にある期間セレクターを使用すると、よく使用する期間をワンクリックで選択できます。
 詳細については、期間およびホストセレクターを参照してください。

7 通知レポート

概要

レポート → 通知セクションでは、各ユーザーに送信された通知の数に関するレポートが表示されます。

右上隅のドロップダウンから、通知に使用されたメディアタイプ（またはすべて）、期間（日/週/月/年ごとのデータ）、および年を選択できます。



各列には、1 人のシステムユーザーごとの合計が表示されます。

6 データ収集

概要

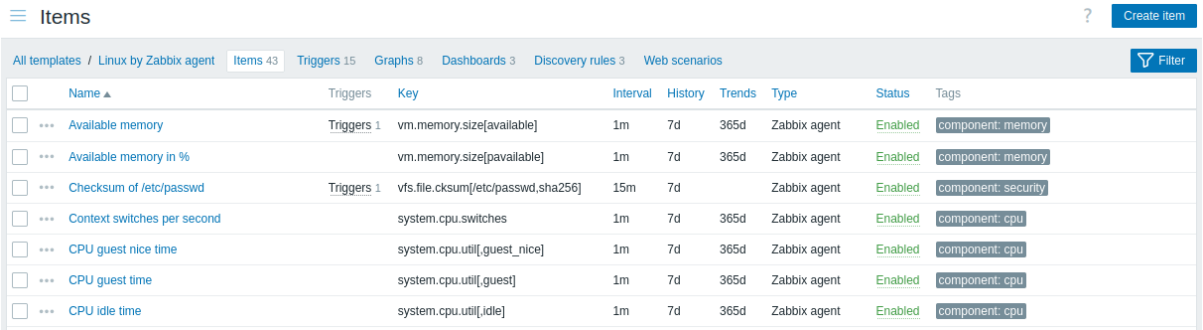
このメニューには、データ収集の設定に関連するセクションがあります。

1 アイテム

概要

テンプレートのアイテムリストには、データ収集 → テンプレートから該当するテンプレートの アイテムをクリックしてアクセスできます。

既存のアイテムのリストが表示されます。



表示されるデータ:

列	説明
アイテムメニュー	三点アイコンをクリックすると、アイテムメニューが開きます。
テンプレート	アイテムが属するテンプレート。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルタで複数のテンプレートまたはテンプレートが選択されていない場合のみ表示されます。
名前	アイテムの名前が青いリンクでアイテム詳細として表示されます。 アイテム名のリンクをクリックすると、アイテムの設定フォームが開きます。 アイテムが他のテンプレートから継承されている場合、テンプレート名がアイテム名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのアイテムリストが開きます。
トリガー	トリガーにマウスを移動すると、アイテムに関連付けられたトリガーを表示するインフォボックスが表示されます。 トリガーの数がグレーで表示されます。
キー	アイテムキーが表示されます。
間隔	チェックの頻度が表示されます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムのトレンドヒストリを保持する日数が表示されます。
タイプ	アイテムタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
ステータス	アイテムのステータスが表示されます - 有効または 無効。ステータスをクリックすると、有効から無効（またはその逆）に変更できます。
タグ	アイテムタグが表示されます。 最大 3 つのタグ（name:value ペア）が表示されます。タグが 3 つ以上ある場合は、「...」リンクが表示され、マウスオーバーで全タグを確認できます。

新しいアイテムを設定するには、右上隅の アイテムの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

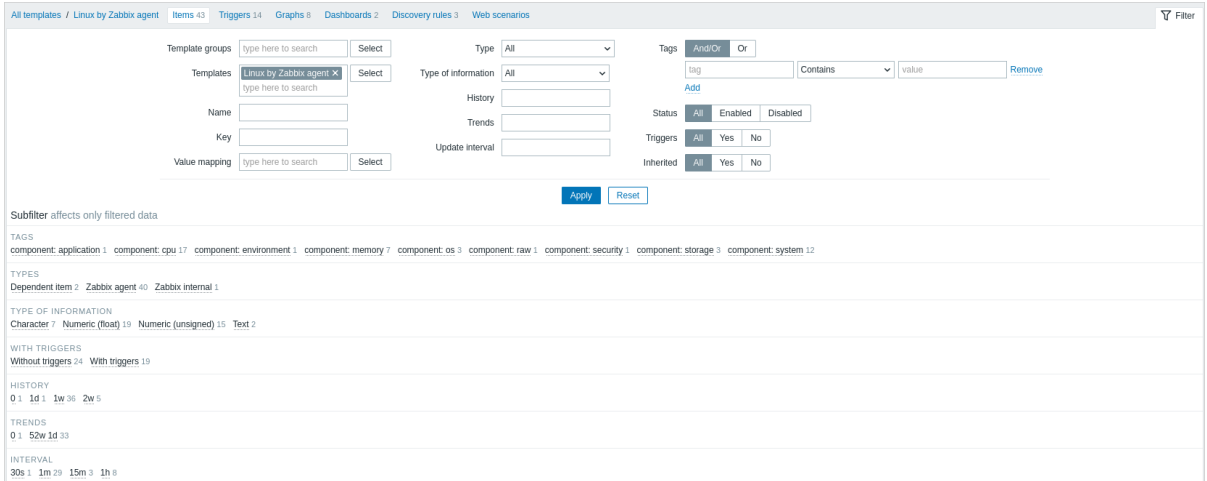
- 有効 - アイテムのステータスを有効に変更します。
- 無効 - アイテムのステータスを無効に変更します。
- コピー - アイテムを他のホストやテンプレートにコピーします。
- 一括更新 - 複数のアイテムに対して複数のプロパティを一度に更新します。
- 削除 - アイテムを削除します。

これらのオプションを使用するには、該当するアイテムの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

アイテムリストには多くのアイテムが含まれている場合があります。フィルターを使用することで、いくつかのアイテムを除外し、探しているアイテムをすばやく見つけることができます。より良い検索パフォーマンスのために、データはマクロを解決せずに検索されます。

フィルターアイコンは右上隅にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。

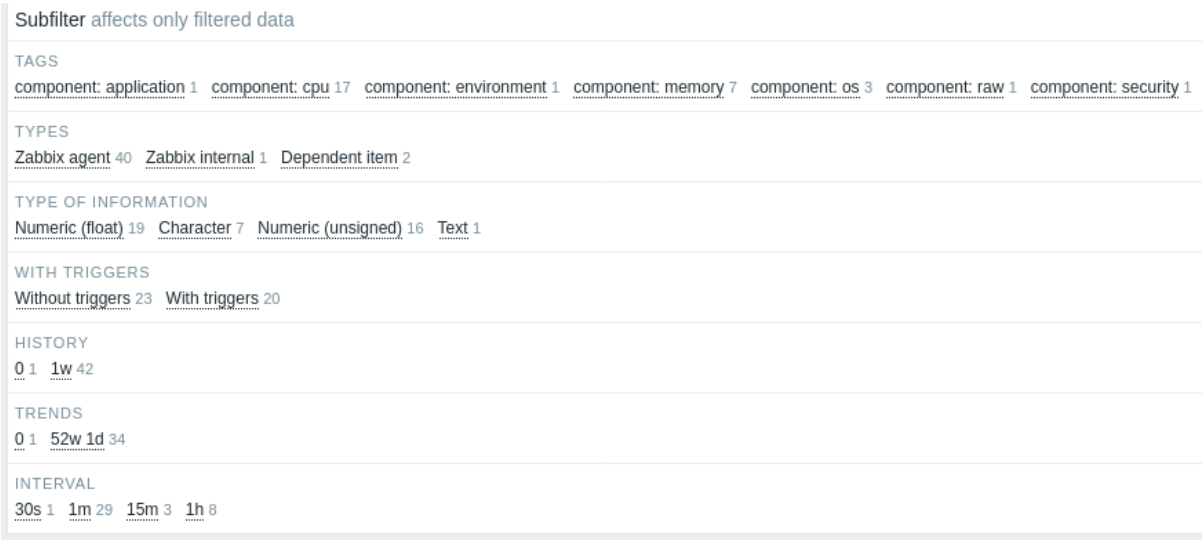


パラメータ	説明
テンプレートグループ	1 つ以上のテンプレートグループでフィルターします。 親テンプレートグループを指定すると、すべてのネストされたグループが暗黙的に選択されます。
テンプレート	1 つ以上のテンプレートでフィルターします。
名前	アイテム名でフィルターします。
キー	アイテムキーでフィルターします。
値のマッピング	使用されている値のマップでフィルターします。 このパラメータは、テンプレートオプションが空の場合は表示されません。
タイプ	アイテムタイプ（Zabbix エージェント、SNMP エージェントなど）でフィルターします。
情報のタイプ	情報のタイプ（数値（符号なし）、浮動小数点など）でフィルターします。
ヒストリ	アイテムのヒストリが保持される期間でフィルターします。
トレンド	アイテムのトレンドが保持される期間でフィルターします。
更新間隔	アイテムの更新間隔でフィルターします。
タグ	表示されるアイテム数を制限するタグを指定します。特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字・小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
ステータス	アイテムのステータス（有効または 無効）でフィルターします。
トリガー	トリガーがある（またはない）アイテムでフィルターします。
継承	リンクされたテンプレートから継承された（または継承されていない）アイテムでフィルターします。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターからさらにフィルタリングを変更できます。

サブフィルターには、関連するアイテムへのクイックアクセス用のクリック可能なリンクが含まれています。アイテムは、タグ、アイテムタイプ、アイテムの更新間隔などの共通エンティティによって関連付けられます。エンティティをクリックすると、そのエンティティがグレーの背景でハイライトされ、アイテムが即座にフィルタリングされます（メインフィルターで適用をクリックする必要はありません）。別のエンティティをクリックすると、フィルタリングされた結果に追加されます。もう一度エンティティをクリックすると、フィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータに基づいて生成されます。このデータは 1000 件に制限されています。サブフィルターでさらに多くのレコードを表示したい場合は、検索およびフィルター結果の制限パラメータ（管理 -> 一般 -> *GUI*）の値を増やす必要があります。

メインフィルターとは異なり、サブフィルターはテーブルの更新要求ごとに更新され、利用可能なフィルタリングオプションとその

カウンター番号の最新情報を常に保持します。

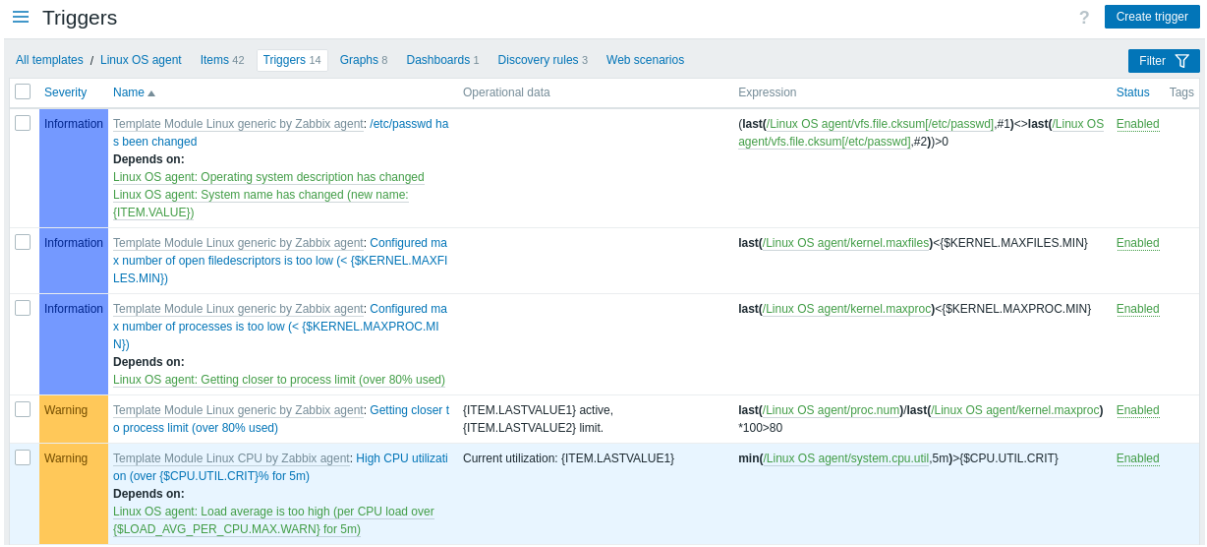
表示されるエンティティの数は、横方向に 100 個までに制限されています。それ以上の場合は、末尾に三点リーダーアイコンが表示されますが、クリックできません。

各クリック可能なエンティティの横の数字は、それにグループ化されたアイテムの数を示します（メインフィルターの結果に基づく）。エンティティをクリックすると、他の利用可能なエンティティの数字がプラス記号付きで表示され、現在の選択に追加できるアイテム数を示します。アイテムのないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

2 トリガー

概要

テンプレートのトリガーリストは、データ収集 → テンプレートから該当テンプレートの トリガーをクリックすることでアクセスできます。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が名前とセルの背景色の両方で表示されます。
テンプレート	トリガーが属するテンプレート。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルタで複数のテンプレートまたはテンプレートが選択されていない場合にのみ表示されます。
名前	トリガーの名前が青いリンクでトリガー詳細として表示されます。 トリガー名のリンクをクリックすると、トリガーの設定フォームが開きます。 トリガーが他のテンプレートから継承されている場合、テンプレート名がトリガー名の前にグレーのリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのトリガーリストが開きます。
運用データ	トリガーの運用データ定義。任意の文字列やマクロを含み、監視データ → 障害で動的に解決されます。
式	トリガーの式が表示されます。式のテンプレート-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動します。
ステータス	トリガーのステータスが表示されます - 有効または 無効。ステータスをクリックすると、有効から無効（またはその逆）に変更できます。
タグ	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上隅の トリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - トリガーのステータスを有効に
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします

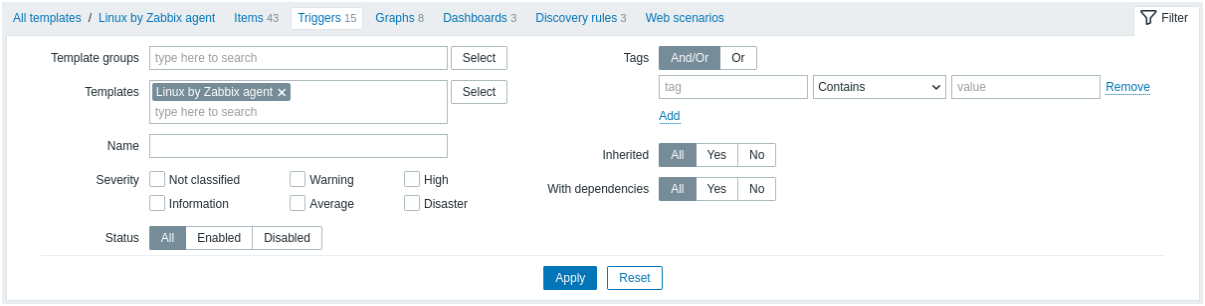
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるトリガーのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルターアイコンは右上隅にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。



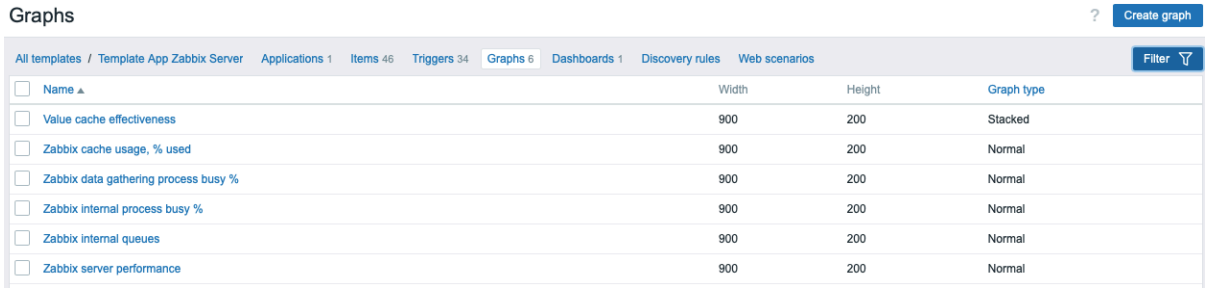
パラメータ	説明
テンプレートグループ	1 つ以上のテンプレートグループでフィルターします。 親テンプレートグループを指定すると、すべてのネストされたグループが暗黙的に選択されます。
テンプレート	1 つ以上のテンプレートでフィルターします。 上記ですでにテンプレートグループが選択されている場合、テンプレートの選択はそれらのグループに限定されます。
名前	トリガー名でフィルターします。
深刻度	1 つまたは複数のトリガーの深刻度でフィルターするように選択します。
ステータス	トリガーステータスでフィルターします。
タグ	トリガータグ名と値でフィルターします。特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字と小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字と小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字と小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字と小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名およびタグ値フィールドでは、マクロおよびマクロ関数がサポートされています。
継承	リンクされたテンプレートから継承された（または継承されていない）トリガーでフィルターします。
依存関係あり	依存関係のある（またはない）トリガーでフィルターします。

3 グラフ

概要

テンプレートのカスタムグラフリストには、データ収集 → テンプレートから該当するテンプレートの グラフをクリックしてアクセスできます。

既存のグラフのリストが表示されます。



表示されるデータ:

列	説明
テンプレート	グラフが属するテンプレート。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルタで複数のテンプレートまたはテンプレートが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前。グラフ詳細への青いリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフの設定フォームが開きます。 グラフが他のテンプレートから継承されている場合、テンプレート名がグラフ名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのグラフリストが開きます。
幅	グラフの幅が表示されます。
高さ	グラフの高さが表示されます。
グラフタイプ	グラフタイプが表示されます - 通常、積み上げ、円、分解円。

新しいグラフを設定するには、右上隅の グラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

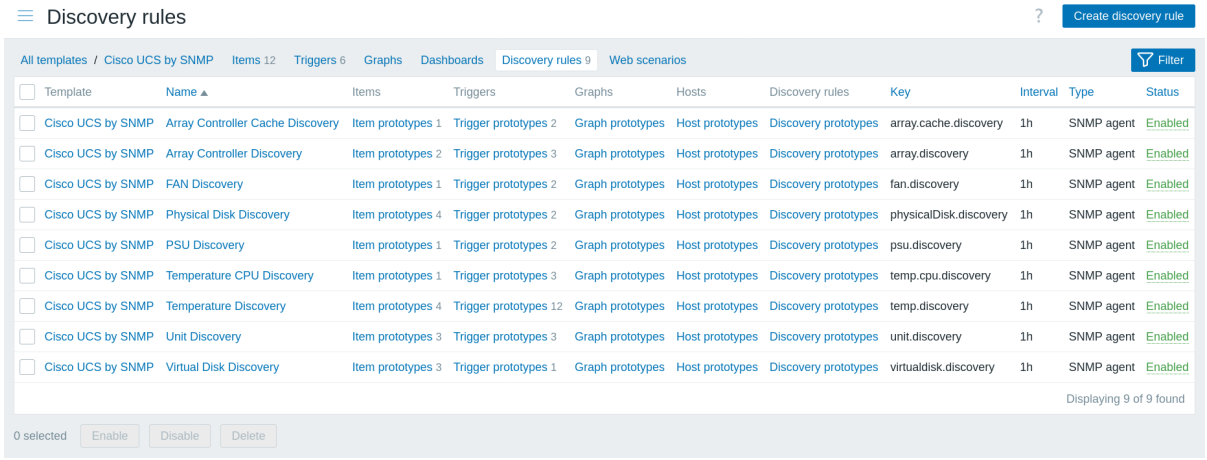
テンプレートグループおよびテンプレートでグラフをフィルタリングできます。検索パフォーマンスを向上させるために、データはマクロが解決されていない状態で検索されます。

4 ディスカバリルール

概要

テンプレートのローレベルディスカバリールールの一覧は、データ収集 → テンプレートから、該当するテンプレートの ディスカバリーをクリックすることでアクセスできます。

既存のローレベルディスカバリールールの一覧が表示されます。フィルター設定を変更することで、テンプレートに関係なくすべてのディスカバリールールや、特定のテンプレートグループのすべてのディスカバリールールを表示することも可能です。



表示されるデータ:

列	説明
テンプレート	テンプレートディスカバリールールが属するテンプレート。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。
名前	ルール名が青いリンクで表示されます。 ルール名をクリックすると、ローレベルディスカバリールールの設定フォームが開きます。 ディスカバリールールが他のテンプレートから継承されている場合、テンプレート名がルール名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのディスカバリールール一覧が開きます。
アイテム	アイテムプロトタイプの一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数がグレーで表示されます。
トリガー	トリガープロトタイプの一覧へのリンクが表示されます。 既存のトリガープロトタイプ数がグレーで表示されます。
グラフ	グラフプロトタイプの一覧へのリンクが表示されます。 既存のグラフプロトタイプ数がグレーで表示されます。
ホスト	ホストプロトタイプの一覧へのリンクが表示されます。 既存のホストプロトタイプ数がグレーで表示されます。
キー	ディスカバリーに使用されるアイテムキーが表示されます。
間隔	ディスカバリーの実行頻度が表示されます。
タイプ	ディスカバリーに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
ステータス	ディスカバリールールのステータスが表示されます - 有効または 無効。ステータスをクリックすることで、有効から無効（またはその逆）に変更できます。

新しいローレベルディスカバリールールを設定するには、右上の ディスカバリールールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

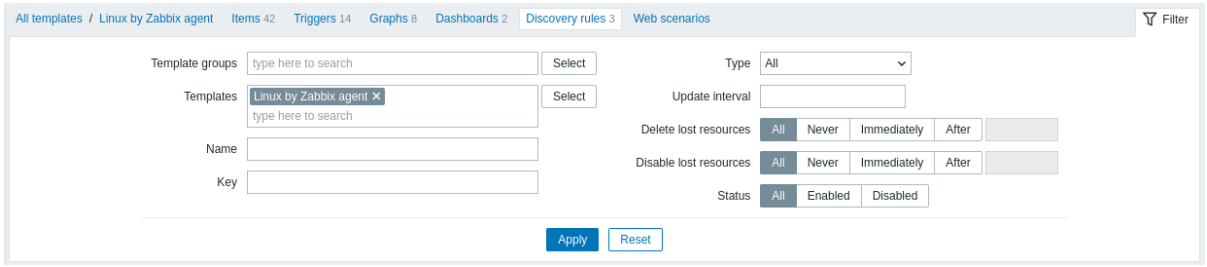
- 有効 - 低レベルのディスカバリルールのステータスを有効に変更します
- 無効 - 低レベルのディスカバリルールのステータスを無効に変更します
- 削除 - 低レベルのディスカバリルールを削除します

これらのオプションを使用するには、ディスカバリルール先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるディスカバリールールのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターアイコンは右上隅にあります。クリックすると、テンプレート、ディスカバリールール名、アイテムキー、アイテムタイプなど、希望するフィルタリング条件を指定できるフィルターが開きます。



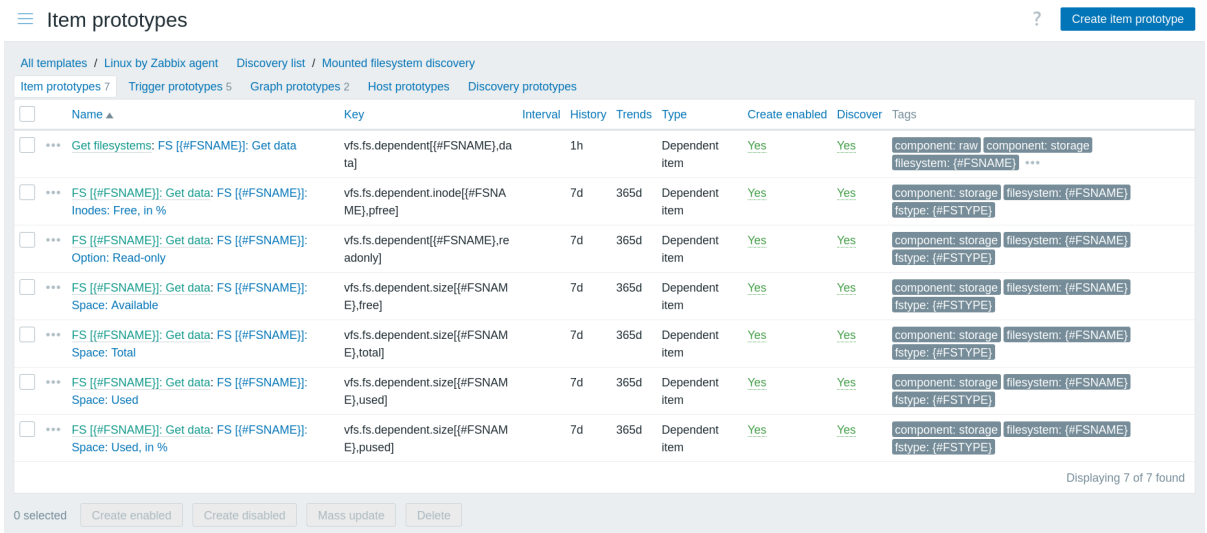
パラメータ	説明
テンプレートグループ	1 つ以上のテンプレートグループでフィルターします。 親テンプレートグループを指定すると、すべてのネストされたグループが暗黙的に選択されます。
テンプレート	1 つ以上のテンプレートでフィルターします。
名前	ディスカバリールール名でフィルターします。
キー	ディスカバリーアイテムキーでフィルターします。
タイプ	ディスカバリーアイテムタイプでフィルターします。
更新間隔	更新間隔でフィルターします。 Zabbix トラッパーおよび従属アイテムでは利用できません。
失われたリソースの削除	失われたリソースの削除期間でフィルターします。
失われたリソースの無効化	失われたリソースの無効化期間でフィルターします。
ステータス	ディスカバリールールのステータス（すべて/有効/無効）でフィルターします。

1 アイテムのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリルールに設定されたアイテムプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、アイテムプロトタイプは低レベルディスカバリ中に実際のホストアイテムを作成するための基礎となります。



表示されるデータ:

列	説明
名前	アイテムプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、アイテムプロトタイプの設定フォームが開きます。 アイテムプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がアイテムプロトタイプ名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのアイテムプロトタイプ一覧が開きます。 アイテムプロトタイプが「従属アイテム」タイプの場合、マスターアイテム名がアイテムプロトタイプ名の前に緑色で表示されます（ホストレベルのアイテムプロトタイプの場合と同様）。
キー	アイテムプロトタイプのキーが表示されます。

列	説明
間隔	チェックの頻度が表示されます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムトレンドのヒストリを保持する日数が表示されます。
タイプ	アイテムプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
有効で作成	このプロトタイプに基づいてアイテムを次の状態で作成します: はい - 有効 いいえ - 無効。’ はい’ と’ いいえ’ をクリックして切り替えることができます。
ディスカバリ	このプロトタイプに基づいてアイテムをディスカバリします: はい - ディスカバリする いいえ - ディスカバリしない。’ はい’ と’ いいえ’ をクリックして切り替えることができます。
タグ	アイテムプロトタイプのタグが表示されます。

新しいアイテムプロトタイプを設定するには、右上隅のアイテムプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

- 有効で作成 - これらのアイテムを有効で作成します
- 無効で作成 - これらのアイテムを無効で作成します
- 一括更新 - これらのアイテムプロトタイプを一括更新します
- 削除 - これらのアイテムプロトタイプを削除します

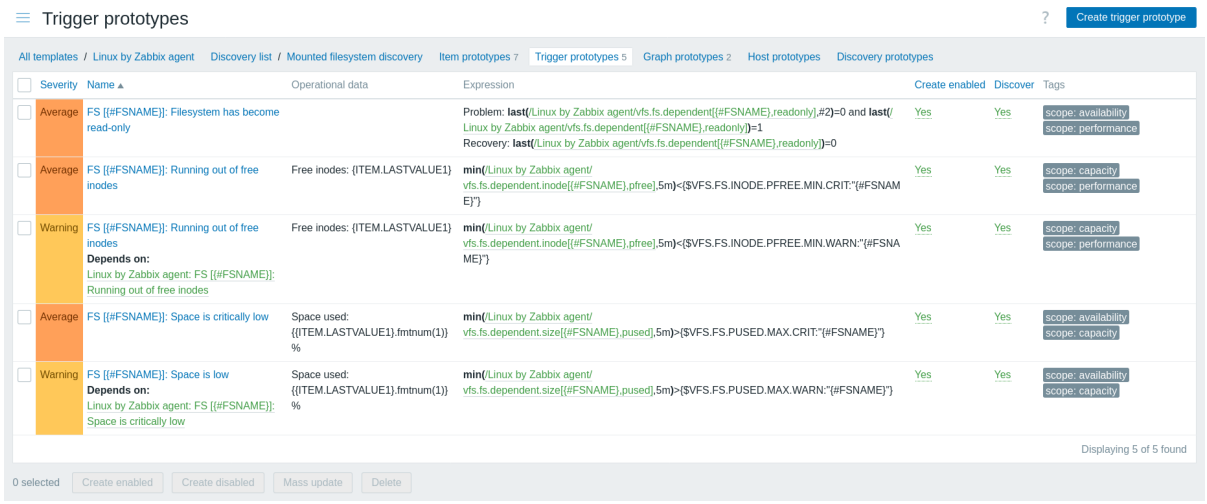
これらのオプションを使用するには、該当するアイテムプロトタイプの前のチェックボックスをオンにし、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、テンプレートのローレベルディスカバリールールに設定されたトリガープロトタイプが表示されます。

テンプレートがホストにリンクされている場合、トリガープロトタイプはローレベルディスカバリー中に実際のホストトリガー作成の基礎となります。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が名前とセルの背景色の両方で表示されます。
名前	トリガープロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、トリガープロトタイプの設定フォームが開きます。 トリガープロトタイプがリンクされたテンプレートに属している場合、テンプレート名がトリガー名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのトリガープロトタイプ一覧が開きます。

列	説明
式	トリガーの式が表示されます。式のテンプレート-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動します。
運用データ	トリガーの運用データのフォーマットが表示されます。任意の文字列やマクロを含み、監視 → 障害で動的に解決されます。
有効	このプロトタイプに基づいてトリガーを作成するかどうか: はい - 有効 いいえ - 無効。’ はい’ と’ いいえ’ をクリックして切り替えることができます。
ディスカバリ	このプロトタイプに基づいてトリガーをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。’ はい’ と’ いいえ’ をクリックして切り替えることができます。
タグ	トリガープロトタイプのタグが表示されます。

新しいトリガープロトタイプを設定するには、右上のトリガープロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションが利用できます。

- 有効で作成 - これらのトリガーを有効で作成します
- 無効で作成 - これらのトリガーを無効で作成します
- 一括更新 - これらのトリガープロトタイプを一括更新します
- 削除 - これらのトリガープロトタイプを削除します

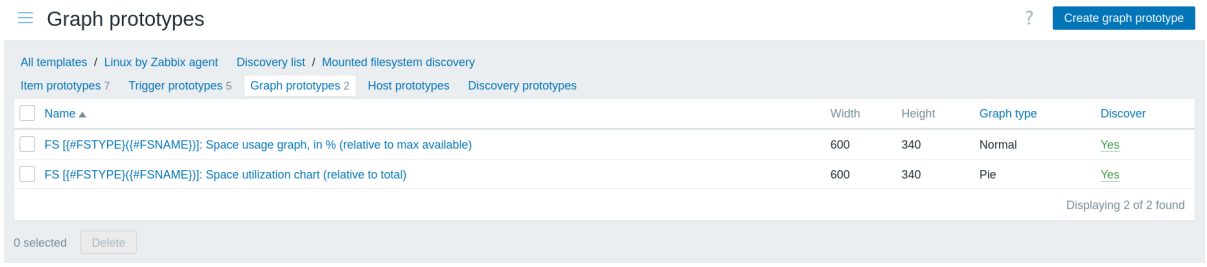
これらのオプションを使用するには、該当するトリガープロトタイプの前のチェックボックスをマークし、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、テンプレートのローレベルディスカバリールールで設定されたグラフプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、グラフプロトタイプはローレベルディスカバリー時に実際のホストグラフ作成の基礎となります。



表示されるデータ:

列	説明
名前	グラフプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、グラフプロトタイプの設定フォームが開きます。 グラフプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がグラフ名の前に灰色のリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのグラフプロトタイプ一覧が開きます。
幅	グラフプロトタイプの幅が表示されます。
高さ	グラフプロトタイプの高さが表示されます。
タイプ	グラフプロトタイプのタイプが表示されます - 標準、積み上げ、円グラフ、分割円グラフ。
ディスカバリ	このプロトタイプに基づいてグラフをディスカバリするか: はい - ディスカバリする いいえ - ディスカバリしない。’ はい’ と’ いいえ’ をクリックすることで切り替えられます。

新しいグラフプロトタイプを設定するには、右上隅のグラフプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

- 削除 - これらのグラフのプロトタイプを削除します

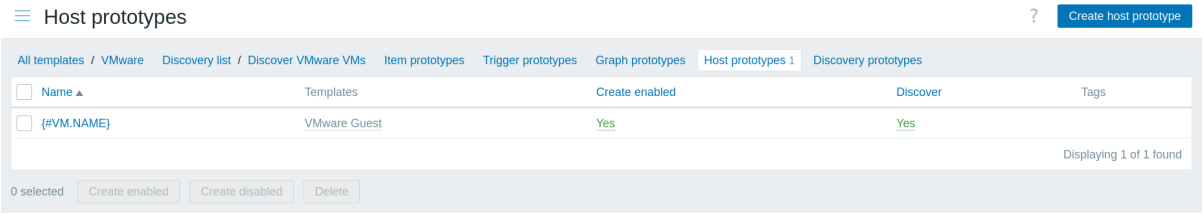
これらのオプションを使用するには、該当するグラフのプロトタイプの前のチェックボックスをオンにし、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリールールで設定されたホストプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ホストプロトタイプは低レベルディスカバリ中に実際のホストを作成するための基礎となります。



表示されるデータ:

列	説明
名前	ホストプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、ホストプロトタイプの設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がホスト名の前にグレーのリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでホストプロトタイプのリストが開きます。
テンプレート	ホストプロトタイプのテンプレートが表示されます。
有効化	このプロトタイプに基づいてホストを作成するかどうか: はい - 有効 いいえ - 無効。’はい’と’いいえ’をクリックして切り替えることができます。
ディスカバリ	このプロトタイプに基づいてホストをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。’はい’と’いいえ’をクリックして切り替えることができます。
タグ	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅のホストプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - ホストを有効として作成します
- 自動生成無効 - ホストを無効として作成します
- 削除 - プロトタイプホストを削除する

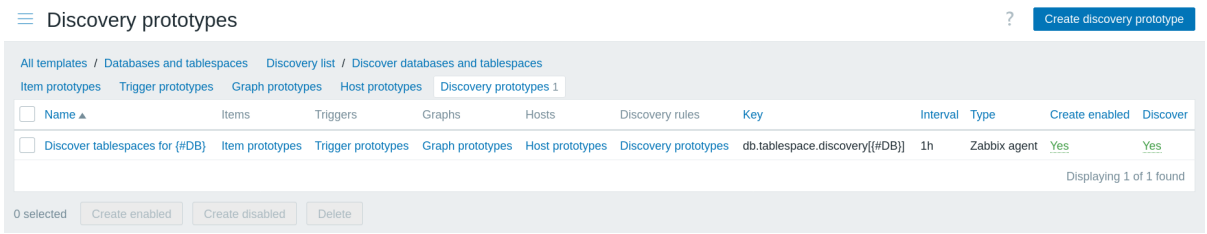
これらのオプションを使用するには、プロトタイプホストの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

5 ディスカバリのプロトタイプ

概要

このセクションでは、テンプレートのローレベルディスカバリルールで設定されたディスカバリプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ディスカバリプロトタイプは親ディスカバリルールの実際のディスカバリプロトタイプ作成の基礎となります。



表示されるデータ:

列	説明
名前	ディスカバリプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、ディスカバリプロトタイプの設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がディスカバリプロトタイプ名の前に灰色のリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのディスカバリプロトタイプ一覧が開きます。
アイテム	アイテムプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数が灰色で表示されます。
トリガー	トリガープロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数が灰色で表示されます。
グラフ	グラフプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数が灰色で表示されます。
ホスト	ホストプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数が灰色で表示されます。
ディスカバリルール	ディスカバリプロトタイプ一覧へのリンクが表示されます。 既存のディスカバリプロトタイプ数が灰色で表示されます。
キー	ディスカバリに使用されるアイテムキーが表示されます。
間隔	ディスカバリを実行する頻度が表示されます。
タイプ	ディスカバリに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
作成有効	ディスカバリプロトタイプの作成ステータスが表示されます - 作成有効（はい）または作成無効（いいえ）。ステータスをクリックすることで、はいからいいえ（またはその逆）に変更できます。
ディスカバリ	ディスカバリプロトタイプのディスカバリステータスが表示されます - ディスカバリする（はい）またはディスカバリしない（いいえ）。ステータスをクリックすることで、はいからいいえ（またはその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上のディスカバリプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで、一括編集オプションを利用できます。

- 有効で作成 - これらのディスカバリープロトタイプを有効で作成します
- 無効で作成 - これらのディスカバリープロトタイプを無効で作成します
- 削除 - これらのディスカバリープロトタイプを削除します

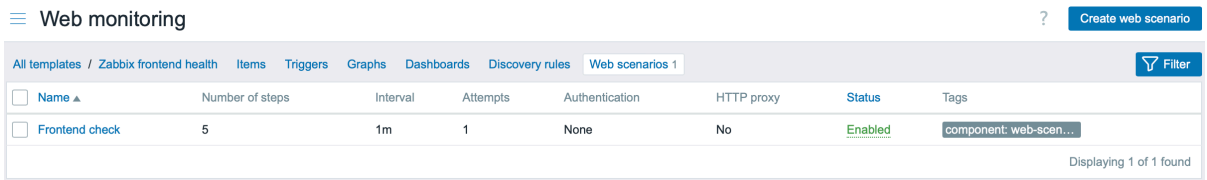
これらのオプションを使用するには、該当するディスカバリープロトタイプの前のチェックボックスをマークし、必要なボタンをクリックします。

5 Web シナリオ

概要

テンプレートのウェブシナリオ一覧は、データ収集 → テンプレートから該当テンプレートの *Web* をクリックすることでアクセスできます。

既存のウェブシナリオの一覧が表示されます。



表示されるデータ:

列	説明
名前	ウェブシナリオの名前。ウェブシナリオ名をクリックすると、ウェブシナリオの設定フォームが開きます。 ウェブシナリオが他のテンプレートから継承されている場合、テンプレート名がウェブシナリオ名の前にグレーのリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのウェブシナリオ一覧が開きます。
ステップ数	シナリオに含まれるステップの数。
更新間隔	シナリオが実行される頻度。
試行回数	ウェブシナリオのステップを実行する試行回数。
認証	認証方式が表示されます - Basic、NTLM、None。
HTTP プロキシ	HTTP プロキシが表示されます。使用しない場合は「No」と表示されます。
ステータス	ウェブシナリオのステータスが表示されます - 有効または無効。 ステータスをクリックすることで変更できます。
タグ	ウェブシナリオのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）が表示されます。タグが 3 つを超える場合は「...」リンクが表示され、マウスオーバーで全てのタグを確認できます。

新しいウェブシナリオを設定するには、右上のウェブシナリオの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

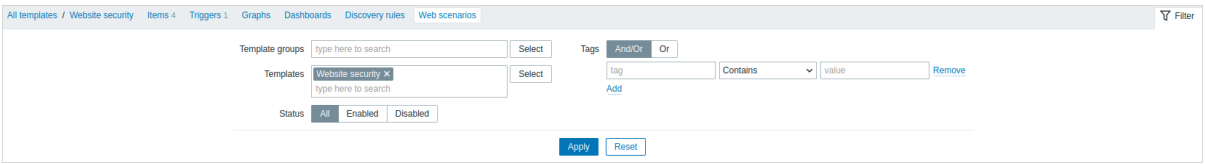
- 有効 - シナリオのステータスを有効にします
- 無効 - シナリオのステータスを無効にします
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、Web シナリオ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるシナリオのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが解決されていない状態で検索されます。

フィルターリンクは Web シナリオのリストの上にあります。クリックすると、テンプレートグループ、テンプレート、ステータス、タグでシナリオをフィルタリングできるフィルターが利用可能になります。

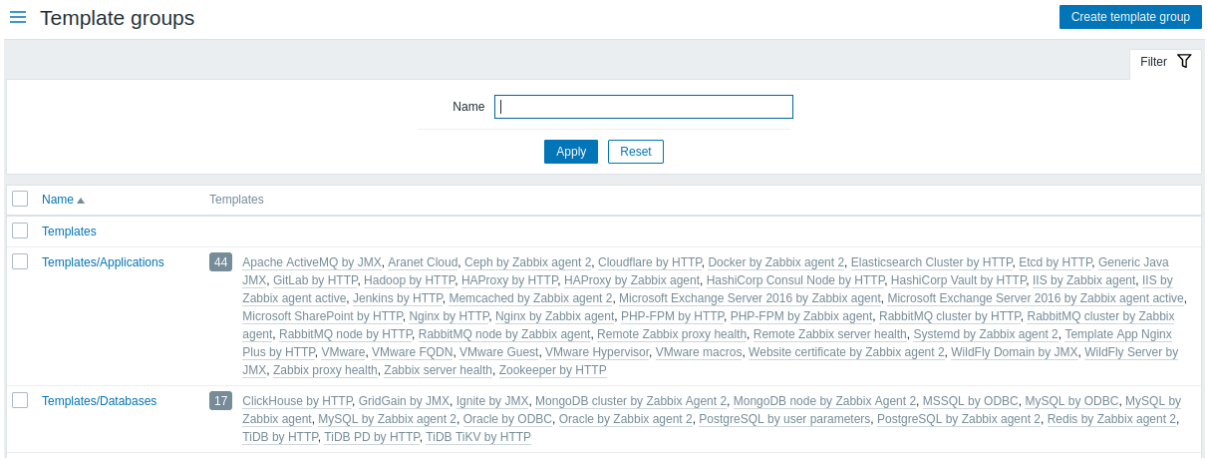


1 テンプレートグループ

概要

データ収集 → テンプレートグループセクションでは、ユーザーはテンプレートグループの設定と管理ができます。

既存のテンプレートグループの一覧とその詳細が表示されます。テンプレートグループ名で検索やフィルタリングが可能です。



表示されるデータ:

列	説明
名前	テンプレートグループの名前。グループ名をクリックすると、グループの設定フォームが開きます。
テンプレート	グループ内のテンプレート数 (グレーで表示) とグループメンバーのリスト。 テンプレート名をクリックするとテンプレートの設定フォームが開きます。 数値をクリックすると、このグループ内のテンプレート一覧が開きます。

一括編集オプション

複数のテンプレートグループを一度に削除するには、該当するグループの前のチェックボックスをマークし、リスト下の [削除] ボタンをクリックします。

フィルターの使用

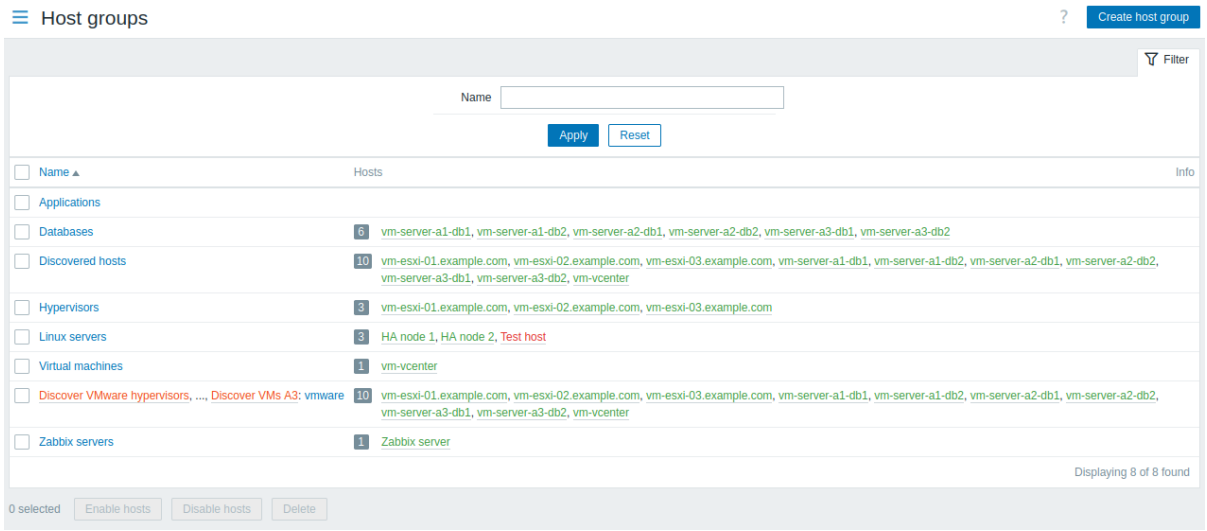
フィルターを使用して、関心のあるテンプレートグループのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを展開せずに検索されます。

2 ホストグループ

概要

データ収集 → ホストグループセクションでは、ユーザーはホストグループの設定と管理を行うことができます。

既存のホストグループの一覧とその詳細が表示されます。ホストグループ名で検索やフィルタリングが可能です。



表示されるデータ:

列	説明
名前	ホストグループの名前。グループ名をクリックすると、グループの設定フォームが開きます。 検出されたホストグループは、低レベルディスカバリールール名がプレフィックスとして表示されます。LLD ルール名をクリックすると、ホストプロトタイプの設定フォームが開きます。検出されたホストグループは、それを検出したすべての LLD ルールが削除されると削除されることに注意してください。
ホスト	グループ内のホスト数（グレーで表示）とグループメンバーのリスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。 数値をクリックすると、ホストの全リストの中から、そのグループに属するホストだけがフィルタリングされます。
情報	ホストグループに関するエラー情報（ある場合）が表示されます。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- ホストを有効 - グループに所属するホストのステータスを監視対象にします
- ホストを無効 - グループに所属するホストのステータスを監視対象外にします
- 削除 - ホストグループをを削除します

これらのオプションを使用するには、それぞれのホストグループの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

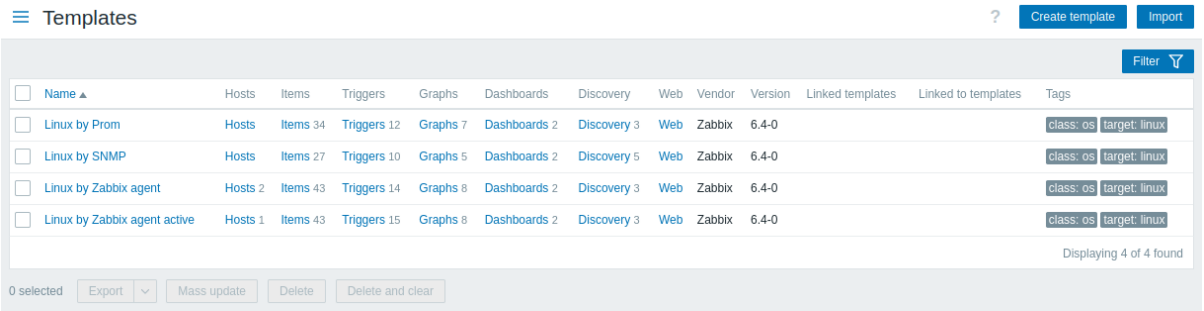
フィルタを使用して、任意のホストグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

3 テンプレート

概要

データ収集 → テンプレートセクションでは、ユーザーはテンプレートの設定と管理を行うことができます。

既存のテンプレートの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
名前	テンプレートの名前。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。
ホスト	テンプレートにリンクされている編集可能なホストの数。読み取り専用ホストは含まれません。 ホストをクリックすると、そのテンプレートにリンクされているホストのみがフィルタリングされたホストリストが開きます。
エンティティ（アイテム、トリガー、グラフ、ダッシュボード、ディスカバリ、*Web*）	テンプレート内の各エンティティの数（グレーで表示）。 エンティティ名をクリックすると、そのエンティティの全リストの中でテンプレートに属するものだけがフィルタリングされます。
リンクされたテンプレート	テンプレートにリンクされているテンプレート。
テンプレートにリンク	テンプレートがリンクされているテンプレート。

列	説明
ベンダー、バージョン	テンプレートのベンダーとバージョン。テンプレート設定にその情報が含まれている場合に表示され、標準テンプレート、インポートされたテンプレート、またはテンプレート APIを通じて変更されたテンプレートのみが対象です。 標準テンプレートの場合、バージョンは Zabbix のメジャーバージョン、区切り文字（”-”）、リビジョン番号（テンプレートの新バージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされる）として表示されます。例: 7.0-0、7.0-3、8.0-0、8.0-3。
タグ	テンプレートのタグ（マクロは未解決）。

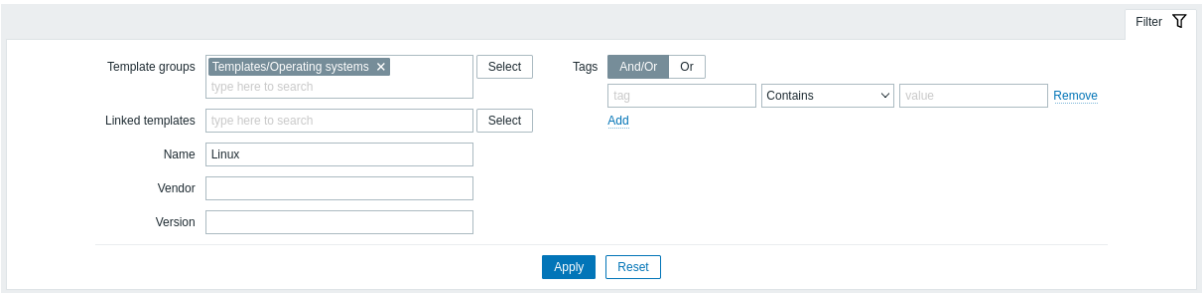
新しいテンプレートを設定するには、右上のテンプレートの作成ボタンをクリックします。

YAML、XML、または JSON ファイルからテンプレートをインポートするには、右上のインポートボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるテンプレートのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルターリンクは、テンプレートの作成およびインポートボタンの下にあります。クリックすると、テンプレートグループ、リンクされたテンプレート、名前、タグでテンプレートをフィルタリングできるフィルターが利用可能になります。



パラメータ	説明
テンプレートグループ	1 つ以上のテンプレートグループでフィルタリングします。 親テンプレートグループを指定すると、すべてのネストされたグループが暗黙的に選択されます。
リンクされたテンプレート	直接リンクされたテンプレートでフィルタリングします。
名前	テンプレート名でフィルタリングします。
ベンダー	テンプレートのベンダーでフィルタリングします。
バージョン	テンプレートのバージョンでフィルタリングします。
タグ	テンプレートタグの名前と値でフィルタリングします。 テンプレートレベルのタグ（継承されたものではない）でのみフィルタリングが可能です。特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子が利用可能です： 存在する - 指定したタグ名を含めます; 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含めます（部分一致、大文字と小文字を区別しない）; 存在しない - 指定したタグ名を除外します; 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります： **And/Or** - すべての条件を満たす必要があり、同じタグ名を持つ条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

- エクスポート - テンプレートを YAML、XML、または JSON ファイルにエクスポートします。
- 一括更新 - 複数のテンプレートに対して複数のプロパティを一度に更新します。

- 削除 - テンプレートを削除し、リンクされたエンティティ（アイテム、トリガーなど）はホストに残します。
- 削除してクリア - テンプレートとそのリンクされたエンティティをホストから削除します。

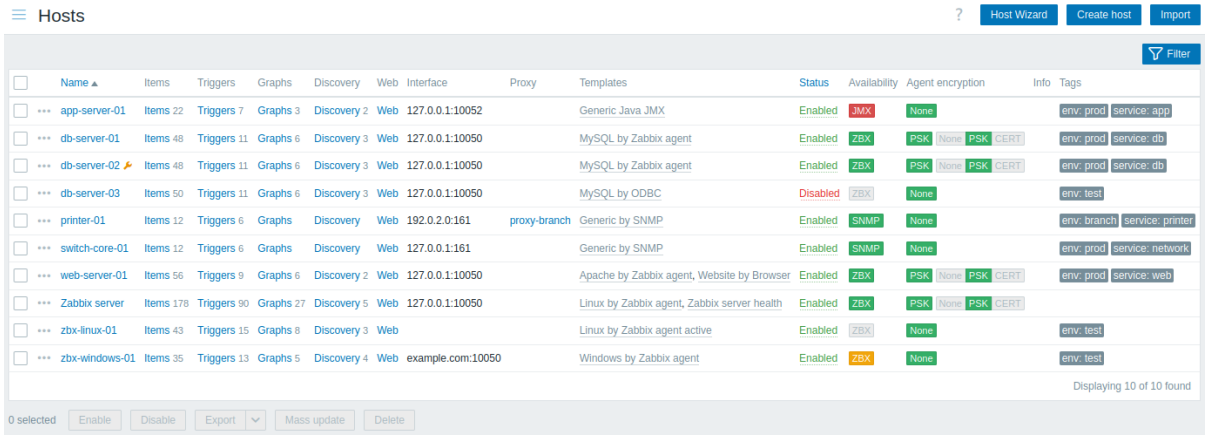
これらのオプションを使用するには、該当するテンプレートのチェックボックスをオンにしてから、必要なボタンをクリックします。

4 ホスト

概要

データ収集 → ホストセクションでは、ユーザーがホストの設定と管理を行うことができます。

既存のホストの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
ホストメニュー	三点リーダーアイコンをクリックするとホストメニューが開きます。
名前	ホストの名前。 ホスト名をクリックすると、ホストの設定フォームが開きます。 ホスト名の後にオレンジ色のレンチアイコン が表示されている場合、そのホストはメンテナンス中です。アイコンにマウスポインタを合わせるとメンテナンスの詳細が表示されます。
エンティティ *(アイテム、トリガー、グラフ、ディスカバリ、Web)*	エンティティ名をクリックすると、ホストのアイテムやトリガーなどが表示されます。 該当するエンティティの数がグレーで表示されます。
インターフェース	ホストのメインインターフェースが表示されます。
プロキシ	割り当てられたプロキシがこの列に表示されます: < プロキシ名 > - ホストはスタンドアロンプロキシによって監視されています（プロキシがプロキシグループの一部であっても）; < プロキシグループ名: プロキシ名 > - ホストはプロキシグループによって監視され、Zabbix サーバーがホストを監視するプロキシを割り当てています; < プロキシグループ名 > - ホストはプロキシのないプロキシグループによって監視されているか、Zabbix サーバーがホストを監視するプロキシを割り当てていません; 何も表示されない - ホストはプロキシまたはプロキシグループのいずれによっても監視されていません。 この列は、監視方法フィルターオプションが「すべて」、「プロキシ」、または「プロキシグループ」に設定されている場合のみ表示されます。
テンプレート	ホストにリンクされているテンプレートが表示されます。 リンクされたテンプレートに他のテンプレートが含まれている場合、それらはカッコ内にカンマ区切りで表示されます。 テンプレート名をクリックすると、その設定フォームが開きます。
ステータス	ホストのステータスが表示されます - 有効または 無効。 ステータスをクリックすると手動で変更できます。 検出されたホストで失われたものには情報アイコンが付けられます。ツールチップテキストでそのステータスの詳細が提供されます。

列	説明
可用性	設定されたインターフェースごとにホストの可用性が表示されます。 可用性アイコンは、Zabbix サーバー上でのホストインターフェースの現在のステータスを表します。そのため、フロントエンドでホストを無効にした場合、Zabbix サーバーが設定変更を同期した後に可用性が更新されます。同様に、ホストを有効にした場合も、Zabbix サーバーが設定変更を同期し、ホストをポーリングした後に可用性が更新されます。 可用性アイコンは、設定されているインターフェースタイプ（エージェント、SNMP、IPMI、JMX）のみを表します。 アイコンにマウスを合わせると、同じタイプのすべてのインターフェースの詳細、ステータス、エラーのリストがポップアップで表示されます。エージェントインターフェースの場合、ポップアップにはインターフェース（パッシブ）とアクティブチェックが表示されます。ホストがアクティブチェックのみの場合、エージェントインターフェースが設定されていなくてもエージェントインターフェースアイコンが表示されます。 インターフェースがないホストの場合、この列は空になります。 単一のホストインターフェースのステータスは、そのインターフェースを使用する有効なアイテムとホスト間の接続によって決まります。ステータスは次のとおりです: 利用可能 - ホストへの接続に成功した; 利用不可 - ホストへの接続に失敗した（タイムアウト、ファイアウォールの問題など）; 不明 - ホストへの接続が試行されていないか、結果が不明。 Zabbix サーバーがインターフェースのステータスをどのように判断するかの詳細は、不明なインターフェースステータスおよび到達不能/利用不可のホストインターフェース設定を参照してください。 単一タイプ（エージェント、SNMP、IPMI、JMX）のすべてのホストインターフェースのステータスは、少なくとも 1 つの有効なアイテムで使用されているインターフェースによって決まります。ステータスはアイコンの色で示されます: 緑 - すべてのインターフェースが利用可能; 黄 - 少なくとも 1 つのインターフェースが利用不可で、少なくとも 1 つが利用可能または不明; 赤 - すべてのインターフェースが利用不可; グレー - 少なくとも 1 つのインターフェースが不明だが、利用不可のものはない。 アクティブチェックの可用性。ホストで少なくとも 1 つのアクティブチェックが有効になっている場合、アクティブチェックも上記のようにエージェントインターフェースの合計可用性に影響します。アクティブチェックの可用性を判断するために、エージェントのアクティブチェックスレッドでハートビートメッセージが送信されます。ハートビートメッセージの頻度は、Zabbixエージェントまたはエージェント 2の設定ファイルの `HeartbeatFrequency` パラメータで制御されます (デフォルト 60 秒、範囲 0-3600)。アクティブチェックのハートビートが 2 x HeartbeatFrequency 秒より古い場合、アクティブチェックは利用不可と見なされます。 注意**:** バージョン 6.2.x より前の Zabbix エージェントはアクティブチェックのハートビートを送信しないため、それらのホストの可用性は不明のままです。
エージェント暗号化	ホストへの接続およびホストからの接続の暗号化ステータスが表示されます: なし - 暗号化なし; **PSK** - 事前共有鍵を使用; 証明書 - 証明書を使用。
情報	ホストに関するエラー情報（ある場合）が表示されます。
タグ	マクロが未解決のホストのタグ。

新しいホストを設定するには、右上のホストウィザードまたはホストの作成ボタンをクリックします。YAML、XML、JSON ファイルからホストをインポートするには、右上のインポートボタンをクリックします。

一括編集オプション

リストの下のボタンには、一括編集オプションがあります。

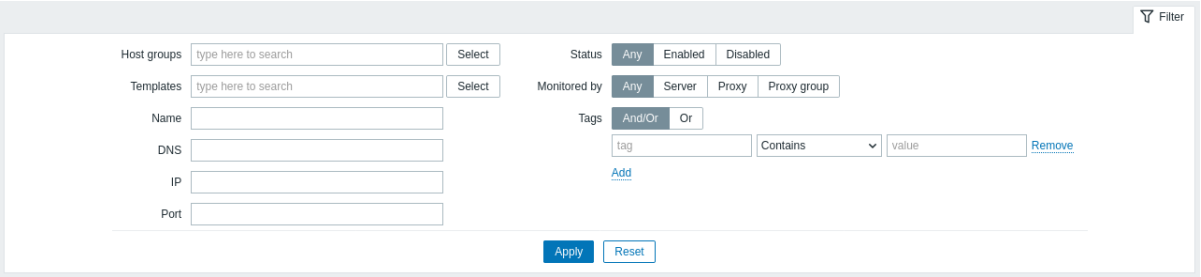
- 有効 - ホストのステータスを監視中に変更します。
- 無効 - ホストのステータスを監視対象外に変更します。
- エクスポート - ホストを YAML、XML、または JSON ファイルにエクスポートします。
- 一括更新 - 複数のホストに対して複数のプロパティを一度に更新します。
- 削除 - ホストを削除します。

これらのオプションを使用するには、該当するホストの前のチェックボックスをオンにし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるホストのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターアイコンは右上隅にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。



パラメータ	説明
ホストグループ	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。
テンプレート	リンクされたテンプレートでフィルターします。
名前	表示されるホスト名でフィルターします。
DNS	DNS 名でフィルターします。
IP	IP アドレスでフィルターします。
ポート	ポート番号でフィルターします。
ステータス	ホストのステータスでフィルターします。
監視元	Zabbix サーバー、プロキシ、またはプロキシグループによって監視されているホストでフィルターします。
プロキシ	ここで指定したプロキシによって監視されているホストでフィルターします。このフィールドは、監視元フィールドで「プロキシ」が選択されている場合にのみ利用できます。
プロキシグループ	ここで指定したプロキシグループによって監視されているホストでフィルターします。このフィールドは、監視元フィールドで「プロキシグループ」が選択されている場合にのみ利用できます。
タグ	ホストタグ名と値でフィルターします。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含めます; 等しい - 指定したタグ名と値を含めます（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含めます（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外します; 等しくない - 指定したタグ名と値を除外します（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外します（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

不明なインターフェースステータス

Zabbix サーバーは、以下の場合にホストインターフェース（エージェント、SNMP、IPMI、JMX）のステータスを「不明」と表示します。

- ホストが無効になっている。
- ホストがプロキシによって監視されるように設定されている、または以前プロキシによって監視されていた場合に異なるプロキシまたはサーバーによって監視されるように設定されている。
- ホストがオフラインと思われるプロキシによって監視されている（最大ハートビート間隔（1 時間）内にプロキシからの更新が受信されていない）。
- そのインターフェースタイプのすべてのホストアイテムが無効になっている。
- そのインターフェースタイプのポーラーが設定されていない（たとえば、`StartAgentPollers`や`StartSNMPPollers`サーバー設定パラメータが 0 に設定されている）。

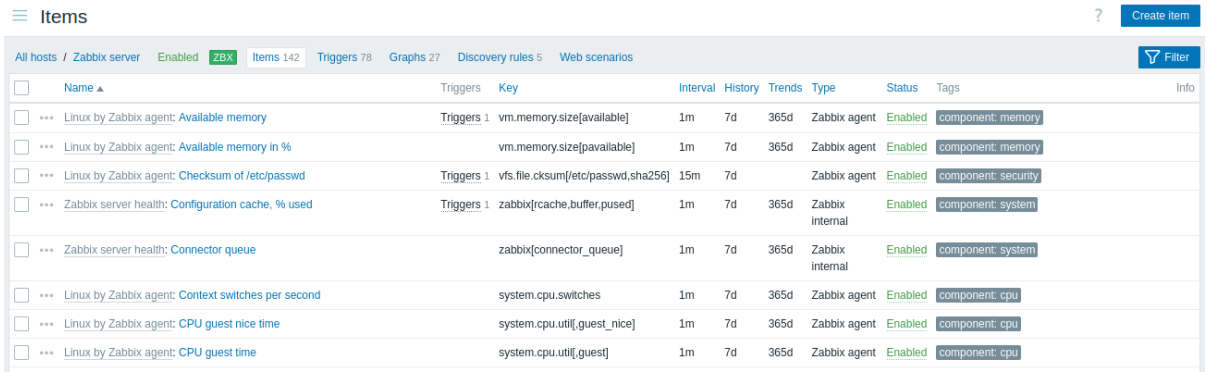
インターフェースの可用性は、Zabbix サーバーの設定キャッシュ同期後に「不明」に設定されます。

プロキシによって監視されているホストのインターフェースの可用性（利用可能/利用不可）は、プロキシの設定キャッシュ同期後に復元されます。

概要

ホストのアイテムリストには、データ収集 → ホストから該当ホストの アイテムをクリックすることでアクセスできます。

既存のアイテムのリストが表示されます。



表示されるデータ:

列	説明
アイテムメニュー	三点アイコンをクリックすると、アイテムメニューが開きます。
ホスト	アイテムのホスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルタで複数のホストまたはホストが選択されていない場合にのみ表示されます。
名前	アイテムの名前が青いリンクでアイテム詳細として表示されます。 アイテム名のリンクをクリックすると、アイテムの設定フォームが開きます。 ホストアイテムがテンプレートに属している場合、テンプレート名がアイテム名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルのアイテムリストが開きます。 アイテムがアイテムプロトタイプから作成されている場合、その名前の前に低レベルディスカバリールール名がオレンジ色で表示されます。ディスカバリールール名をクリックすると、アイテムプロトタイプリストが開きます。
トリガー	トリガーの上にマウスを移動すると、アイテムに関連付けられたトリガーを表示するインフォボックスが表示されます。 トリガーの数がグレーで表示されます。
キー	アイテムキーが表示されます。
間隔	チェックの頻度が表示されます。 注意受動アイテムは、今すぐ実行ボタンを押すことで即時にチェックすることもできます。
ヒストリ	アイテムデータのヒストリが何日間保持されるかが表示されます。
トレンド	アイテムのトレンドヒストリが何日間保持されるかが表示されます。
タイプ	アイテムタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
ステータス	アイテムのステータスが表示されます - 有効、無効、またはサポート対象外。ステータスをクリックすることで手動で変更できます - 有効から無効（およびその逆）、サポート対象外から無効（およびその逆）。 検出されたアイテムが失われた場合は、情報アイコンが表示されます。ツールチップテキストでステータスの詳細が提供されます。
タグ	アイテムタグが表示されます。 最大 3 つのタグ（name:value ペア）が表示されます。タグが 3 つを超える場合は、「...」リンクが表示され、マウスオーバーで全てのタグを確認できます。
情報	アイテムが正常に動作している場合、この列にはアイコンは表示されません。エラーが発生している場合は、四角い「i」アイコンが表示されます。アイコンにカーソルを合わせると、エラー内容のツールチップが表示されます。

新しいアイテムを設定するには、右上の アイテムの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

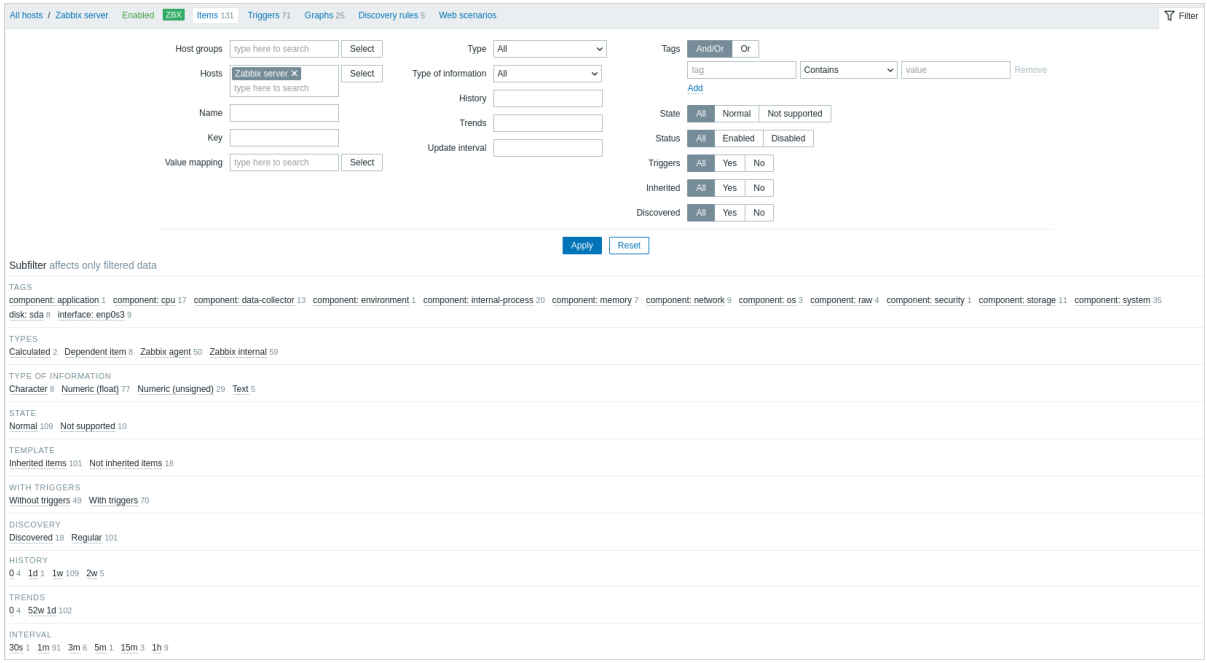
- 有効 - アイテムのステータスを有効に変更します
- 無効 - アイテムのステータスを無効に変更します
- 今すぐ実行 - 新しいアイテム値のチェックを即時に実行します。パッシブチェックのみサポートされています（詳細はこちら）。値を即時にチェックする場合、設定キャッシュは更新されないため、アイテム設定のごく最近の変更は反映されません。
- 履歴とトレンドのクリア - アイテムの履歴およびトレンドデータを削除します。
- コピー - アイテムを他のホストやテンプレートにコピーします。
- 一括更新 - 複数のアイテムのいくつかのプロパティを一度に更新します。
- 削除 - アイテムを削除します。

これらのオプションを使用するには、該当するアイテムの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるアイテムのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターアイコンは右上隅にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。



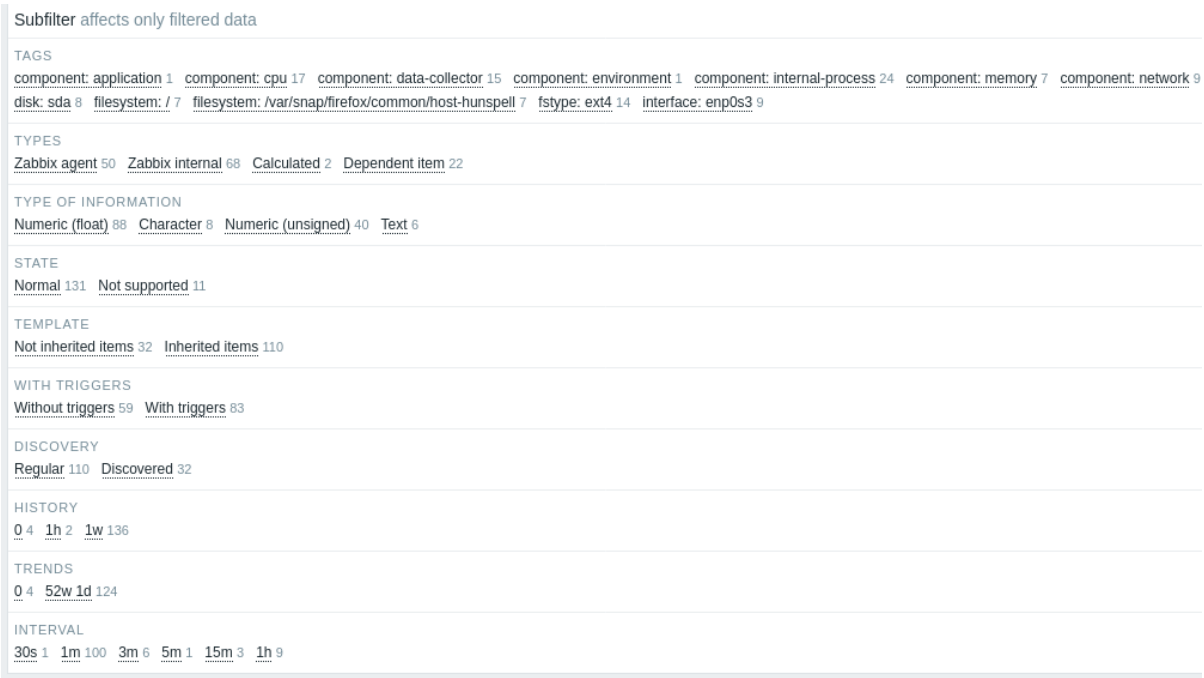
パラメータ	説明
ホストグループ	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
ホスト	1 つ以上のホストでフィルターします。
名前	アイテム名でフィルターします。
キー	アイテムキーでフィルターします。
値のマッピング	使用されている値のマップでフィルターします。 このパラメータは、ホストオプションが空の場合は表示されません。
タイプ	アイテムタイプ（Zabbix エージェント、SNMP エージェントなど）でフィルターします。
情報のタイプ	情報のタイプ（数値（符号なし）、浮動小数点など）でフィルターします。
履歴	アイテムの履歴が保持される期間でフィルターします。
トレンド	アイテムのトレンドが保持される期間でフィルターします。
更新間隔	アイテムの更新間隔でフィルターします。

パラメータ	説明
タグ	表示されるアイテムの数を制限するタグを指定します。特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字・小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
状態	アイテムの状態（通常または サポートされていない）でフィルターします。
ステータス	アイテムのステータス（有効または 無効）でフィルターします。
トリガー	トリガーがある（またはない）アイテムでフィルターします。
継承	テンプレートから継承された（または継承されていない）アイテムでフィルターします。
ディスカバリ	ローレベルディスカバリで検出された（または検出されていない）アイテムでフィルターします。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターからのフィルタリングをさらに変更できます。

サブフィルターには、関連するアイテムへのクイックアクセス用のクリック可能なリンクが含まれています。アイテムは、タグ、アイテムタイプ、アイテムの状態、アイテムの更新間隔などの共通エンティティによって関連付けられています。エンティティをクリックすると、そのエンティティがグレーの背景でハイライトされ、アイテムが即座にフィルタリングされます（メインフィルターで適用をクリックする必要はありません）。別のエンティティをクリックすると、フィルタリングされた結果に追加されます。もう一度エンティティをクリックすると、フィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータに基づいて生成されます。このデータは最大 1000 件に制限されています。サブフィルターでさらに多くのレコードを表示したい場合は、検索およびフィルター結果の制限パラメータ（管理 -> 一般 -> *GUI*）の値を増やす必要があります。

メインフィルターとは異なり、サブフィルターはテーブルのリフレッシュ要求ごとに更新され、利用可能なフィルタリングオプションとそのカウンター番号の最新情報を常に保持します。

表示されるエンティティの数は、横方向に最大 100 までに制限されています。それ以上の場合は、末尾に三点リーダーアイコンが表示されますが、クリックはできません。

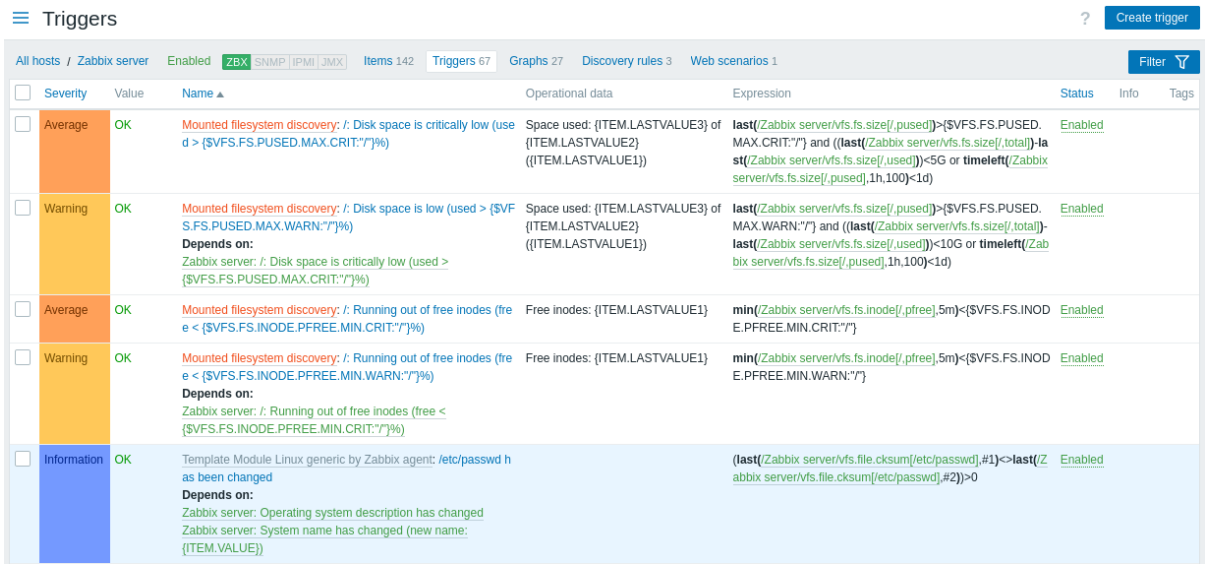
各クリック可能なエンティティの横に表示される数字は、そのエンティティにグループ化されたアイテムの数を示します（メインフィルターの結果に基づく）。エンティティをクリックすると、他の利用可能なエンティティの数字がプラス記号付きで表示され、現在

の選択に追加できるアイテム数を示します。アイテムがないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

2 トリガー

概要

ホストのトリガーリストには、データ収集 → ホストから該当ホストの トリガーをクリックしてアクセスできます。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が名前とセルの背景色で表示されます。
値	トリガーの値が表示されます: **OK** - トリガーが OK 状態 **PROBLEM** - トリガーが問題状態
ホスト	トリガーのホスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルターで複数のホストまたはホストが選択されていない場合のみ表示されます。
名前	トリガーの名前が青いリンクでトリガー詳細として表示されます。 トリガー名のリンクをクリックすると、トリガーの設定フォームが開きます。 ホストトリガーがテンプレートに属している場合、テンプレート名がトリガー名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルのトリガーリストが開きます。 トリガープロトタイプから作成されたトリガーの場合、名前の前にローレベルディスカバリールール名がオレンジ色で表示されます。ディスカバリールール名をクリックすると、トリガープロトタイプリストが開きます。
運用データ	トリガーの運用データ定義。任意の文字列やマクロを含み、監視 → 問題で動的に解決されます。
式	トリガーの式が表示されます。式のホスト-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動します。
ステータス	トリガーのステータスが表示されます - 有効、無効、不明。ステータスをクリックすると手動で変更できます - 有効から無効（およびその逆）、不明から無効（およびその逆）。 無効なトリガーの問題はフロントエンドに表示されなくなりますが、削除はされません。 検出されたトリガーが失われた場合、情報アイコンでマークされます。ツールチップテキストでステータスの詳細が提供されます。
情報	すべてが正常に動作している場合、この列にはアイコンは表示されません。エラーがある場合は、四角い「i」アイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明がツールチップで表示されます。
タグ	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上の トリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

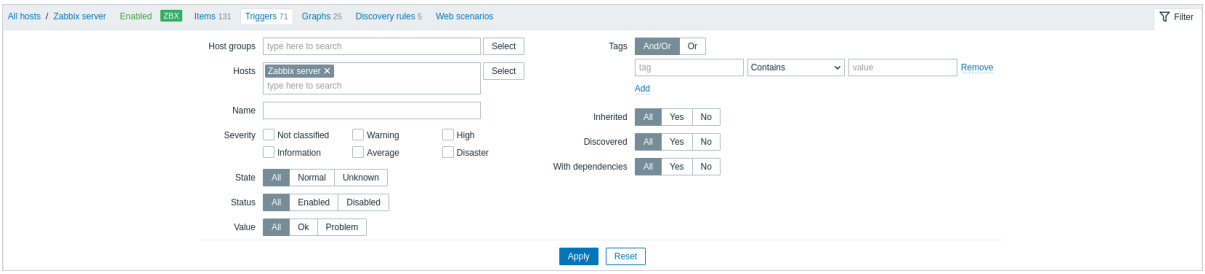
- 有効 - トリガーのステータスを有効に
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるトリガーのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターアイコンは右上隅にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。



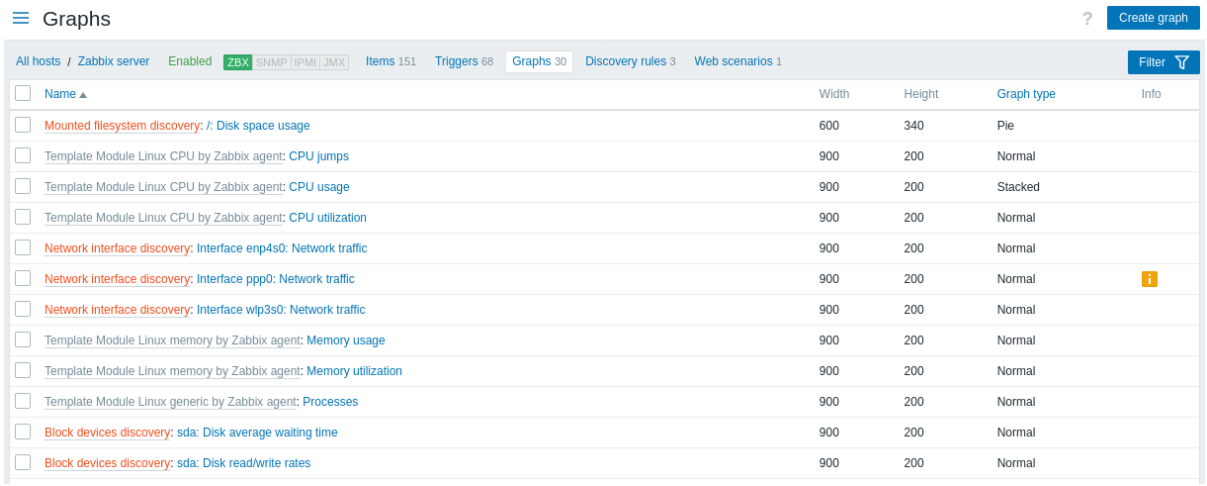
パラメータ	説明
ホストグループ	1 つまたは複数のホストグループでフィルターします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
ホスト	1 つまたは複数のホストでフィルターします。 上記でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに限定されます。
名前	トリガー名でフィルターします。
深刻度	1 つまたは複数のトリガーの深刻度でフィルターします。
状態	トリガーの状態でフィルターします。
ステータス	トリガーのステータスでフィルターします。
値	トリガーの値でフィルターします。
タグ	トリガーのタグ名と値でフィルターします。特定のタグやタグ値を含めたり除外したりできます。 複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める 等しい - 指定したタグ名と値を含める（大文字・小文字を区別） 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない） 存在しない - 指定したタグ名を除外する 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別） 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名およびタグ値フィールドの両方でマクロおよびマクロ関数がサポートされています。
継承	テンプレートから継承された（または継承されていない）トリガーでフィルターします。
ディスカバリー	ローレベルディスカバリーでディスカバリーされた（またはされていない）トリガーでフィルターします。
依存関係あり	依存関係のある（またはない）トリガーでフィルターします。

3 グラフ

概要

ホストのカスタムグラフリストには、データ収集 → ホストから該当ホストの グラフをクリックしてアクセスできます。

既存のグラフのリストが表示されます。



表示されるデータ:

列	説明
ホスト	グラフのホスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルタで複数のホストまたはホストが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前。グラフ詳細への青いリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフの設定フォームが開きます。 ホストグラフがテンプレートに属している場合、テンプレート名がグラフ名の前に灰色のリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルのグラフリストが開きます。 グラフがグラフプロトタイプから作成されている場合、その名前の前にローレベルディスカバリールール名がオレンジ色で表示されます。ディスカバリールール名をクリックすると、グラフプロトタイプリストが開きます。
幅	グラフの幅が表示されます。
高さ	グラフの高さが表示されます。
グラフタイプ	グラフタイプが表示されます - 標準、積算、円、分割円。
情報	グラフが正常に動作している場合、この列にはアイコンは表示されません。エラーが発生している場合は、”i” の文字が入った四角いアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明がツールチップで表示されます。

新しいグラフを設定するには、右上隅の グラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

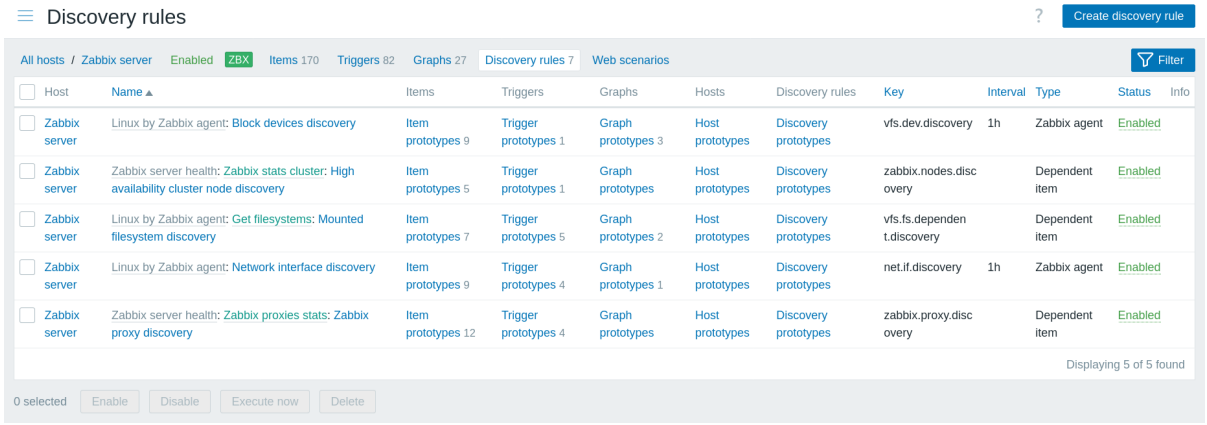
グラフはホストグループとホストでフィルタリングできます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

4 ディスカバリルール

概要

ホストのローレベルディスカバリールールのリストには、データ収集 → ホストから該当ホストの ディスカバリをクリックしてアクセスできます。

既存のローレベルディスカバリールールのリストが表示されます。フィルタ設定を変更することで、ホストに関係なくすべてのディスカバリールールや、特定のホストグループのすべてのディスカバリールールを表示することも可能です。



表示されるデータ:

列	説明
ホスト	表示用ホスト名が表示されます。 ホスト名をクリックすると、ホストの設定フォームが開きます。 表示用ホスト名がない場合は、技術的なホスト名が表示されます。
名前	ルール名が青いリンクで表示されます。 ルール名をクリックすると、ローレベルディスカバリールールの設定フォームが開きます。 ディスカバリールールがテンプレートに属している場合は、テンプレート名がルール名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルのルールリストが開きます。
アイテム	アイテムプロトタイプのリストへのリンクが表示されます。 既存のアイテムプロトタイプの数がグレーで表示されます。
トリガー	トリガープロトタイプのリストへのリンクが表示されます。 既存のトリガープロトタイプの数がグレーで表示されます。
グラフ	グラフプロトタイプのリストへのリンクが表示されます。 既存のグラフプロトタイプの数がグレーで表示されます。
ホスト	ホストプロトタイプのリストへのリンクが表示されます。 既存のホストプロトタイプの数がグレーで表示されます。
キー	ディスカバリに使用されるアイテムキーが表示されます。
間隔	ディスカバリを実行する頻度が表示されます。 ※ ディスカバリは、リスト下部の 今すぐ実行ボタンを押すことで即時に実行することもできます。
タイプ	ディスカバリに使用されるアイテムタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
ステータス	ディスカバリールールのステータスが表示されます - 有効、無効、サポート外。ステータスをクリックすると変更できます - 有効から無効へ（またはその逆）、サポート外から無効へ（またはその逆）。
情報	問題がなければ、この列にはアイコンは表示されません。エラーがある場合は、”i” の文字が入った四角いアイコンが表示されます。アイコンにマウスオーバーすると、エラー内容のツールチップが表示されます。

新しいローレベルディスカバリールールを設定するには、右上の ディスカバリールールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

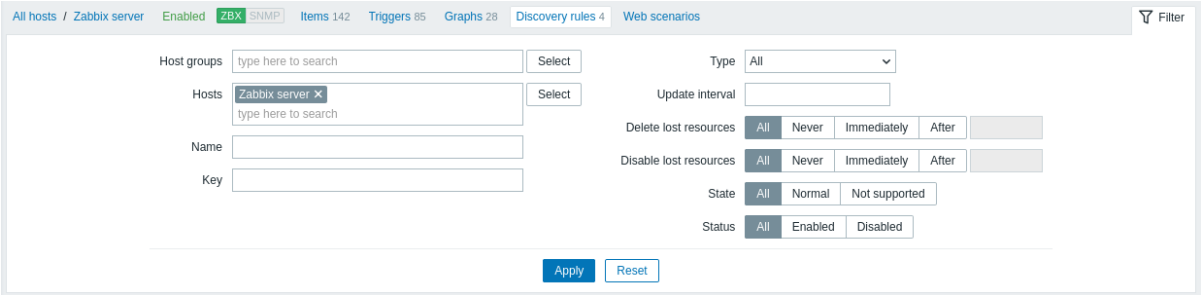
- 有効 - ローレベルディスカバリルールのステータスを有効に変更します。
- 無効 - ローレベルディスカバリルールのステータスを無効に変更します。
- 今すぐ実行 - ディスカバリルールに基づいて即座にディスカバリを実行します。詳細はこちらをご覧ください。即時にディスカバリを実行する場合、設定キャッシュは更新されないため、ディスカバリルール設定のごく最近の変更は反映されません。
- 削除 - ローレベルディスカバリルールを削除します。

これらのオプションを使用するには、該当するディスカバリルールの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるディスカバリールールのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターリンクはディスカバリールールのリストの上にあります。これをクリックすると、ホストグループ、ホスト、名前、アイテムキー、アイテムタイプ、その他のパラメータでディスカバリールールをフィルタリングできるフィルターが利用可能になります。

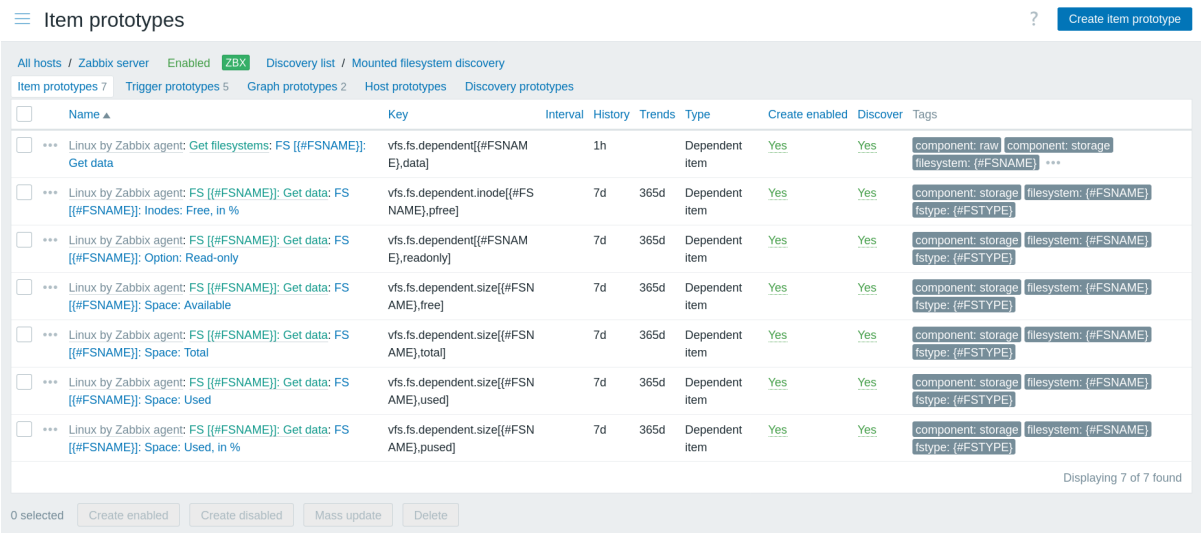


パラメータ	説明
ホストグループ	1 つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。
ホスト	1 つ以上のホストでフィルタリングします。
名前	ディスカバリールール名でフィルタリングします。
キー	ディスカバリーアイテムキーでフィルタリングします。
タイプ	ディスカバリーアイテムタイプでフィルタリングします。
更新間隔	更新間隔でフィルタリングします。 Zabbix トラッパーおよび従属アイテムでは利用できません。
失われたリソースの削除	失われたリソースの削除期間でフィルタリングします。
失われたリソースの無効化	失われたリソースの無効化期間でフィルタリングします。
SNMP OID	SNMP OID でフィルタリングします。 タイプとして *SNMP* エージェントが選択されている場合のみ利用可能です。
状態	ディスカバリールールの状態（すべて/通常/サポートされていない）でフィルタリングします。
ステータス	ディスカバリールールのステータス（すべて/有効/無効）でフィルタリングします。

1 アイテムのプロトタイプ

概要

このセクションでは、ホストのローレベルディスカバリールールのアイテムプロトタイプが表示されます。アイテムプロトタイプは、ローレベルディスカバリー中に作成される実際のホストアイテムの基礎となります。



表示されるデータ:

列	説明
名前	アイテムプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、アイテムプロトタイプの設定フォームが開きます。 アイテムプロトタイプがテンプレートに属している場合、テンプレート名がルール名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルのアイテムプロトタイプ一覧が開きます。 アイテムプロトタイプが「従属アイテム」タイプの場合、親アイテム名がアイテムプロトタイプ名の前に緑色で表示されます。
キー	アイテムプロトタイプのキーが表示されます。
間隔	チェックの頻度が表示されます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムトレンドのヒストリを保持する日数が表示されます。
タイプ	アイテムプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
有効で作成	このプロトタイプに基づいてアイテムを作成する際の状態: はい - 有効 いいえ - 無効。「はい」と「いいえ」をクリックすることで切り替えられます。
ディスカバリ	このプロトタイプに基づいてアイテムをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。「はい」と「いいえ」をクリックすることで切り替えられます。
タグ	アイテムプロトタイプのタグが表示されます。

新しいアイテムプロトタイプを設定するには、右上隅のアイテムプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

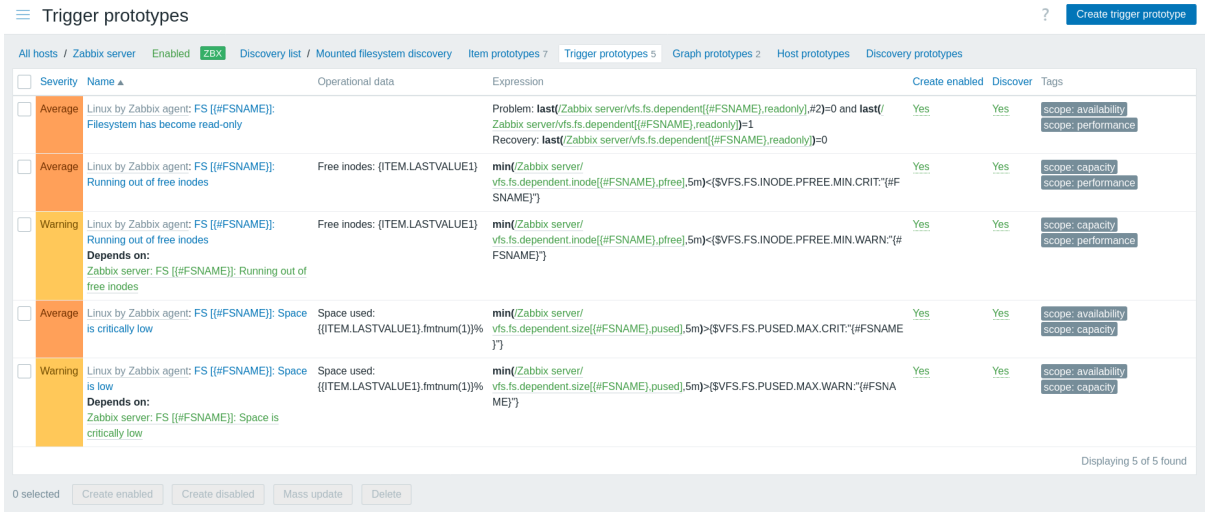
- 有効で作成 - これらのアイテムを有効で作成します
- 無効で作成 - これらのアイテムを無効で作成します
- 一括更新 - これらのアイテムプロトタイプを一括更新します
- 削除 - これらのアイテムプロトタイプを削除します

これらのオプションを使用するには、該当するアイテムプロトタイプの前のチェックボックスをマークし、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、ホストのローレベルディスカバリールールのトリガープロトタイプが表示されます。トリガープロトタイプは、ローレベルディスカバリー中に作成される実際のホストトリガーの基礎となります。



表示されるデータ:

列	説明
名前	トリガープロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、トリガープロトタイプの設定フォームが開きます。 トリガープロトタイプがリンクされたテンプレートに属している場合、テンプレート名がトリガー名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでトリガープロトタイプの一覧が開きます。
運用データ	トリガーの運用データのフォーマットが表示されます。任意の文字列やマクロを含み、監視 → 障害で動的に解決されます。
有効作成	このプロトタイプに基づいてトリガーを作成するかどうか: はい - 有効 いいえ - 無効。’はい’と’いいえ’をクリックして切り替えることができます。
ディスカバリ	このプロトタイプに基づいてトリガーをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。’はい’と’いいえ’をクリックして切り替えることができます。
タグ	トリガープロトタイプのタグが表示されます。

新しいトリガープロトタイプを設定するには、右上隅のトリガープロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

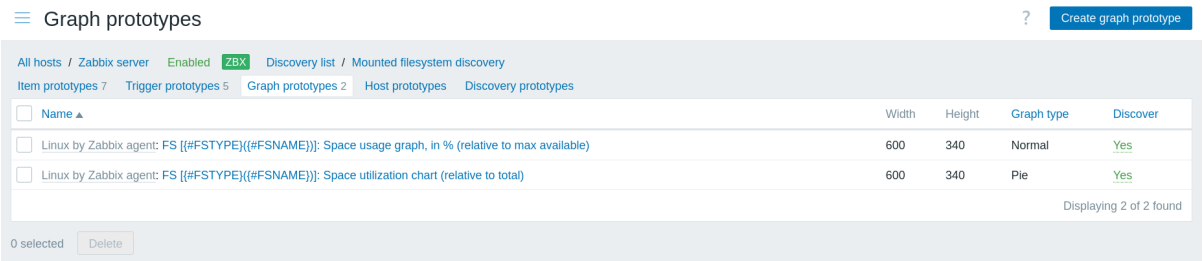
- 有効で作成 - これらのトリガーを有効で作成します
- 無効で作成 - これらのトリガーを無効で作成します
- 一括更新 - これらのトリガープロトタイプを一括更新します
- 削除 - これらのトリガープロトタイプを削除します

これらのオプションを使用するには、該当するトリガープロトタイプの前のチェックボックスをマークし、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、ホストのローレベルディスカバリールールのグラフプロトタイプが表示されます。グラフプロトタイプは、ローレベルディスカバリー中に作成される実際のホストグラフの基礎となります。



表示されるデータ:

列	説明
名前	グラフプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、グラフプロトタイプの設定フォームが開きます。 グラフプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がグラフ名の前に灰色のリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでグラフプロトタイプのリストが開きます。
幅	グラフプロトタイプの幅が表示されます。
高さ	グラフプロトタイプの高さが表示されます。
タイプ	グラフプロトタイプのタイプが表示されます - 標準、積み上げ、円グラフ、分解円グラフ。
ディスカバリ	このプロトタイプに基づいてグラフをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。’はい’と’いいえ’をクリックすることで切り替えることができます。

新しいグラフプロトタイプを設定するには、右上隅のグラフプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

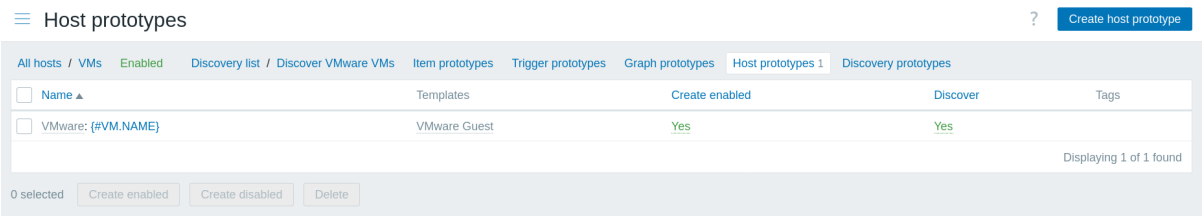
- 削除 - これらのグラフプロトタイプを削除

これらのオプションを使用するには、該当するグラフプロトタイプの前のチェックボックスをオンにし、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、ホストのローレベルディスカバリールールのホストプロトタイプが表示されます。ホストプロトタイプは、ローレベルディスカバリー中に作成される実際のホストの基礎となります。



表示されるデータ:

列	説明
名前	ホストプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、ホストプロトタイプの設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がホスト名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのホストプロトタイプリストが開きます。
テンプレート	ホストプロトタイプのテンプレートが表示されます。
有効化	このプロトタイプに基づいてホストを作成するかどうか: はい - 有効 いいえ - 無効。'はい' と 'いいえ' をクリックして切り替えることができます。
ディスカバリ	このプロトタイプに基づいてホストをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。'はい' と 'いいえ' をクリックして切り替えることができます。
タグ	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅のホストプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

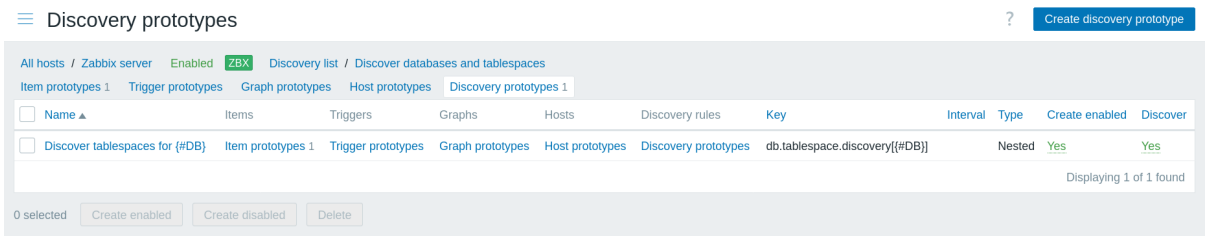
- 有効で作成 - これらのホストを有効として作成します
- 無効で作成 - これらのホストを無効として作成します
- 削除 - これらのホストプロトタイプを削除します

これらのオプションを使用するには、該当するホストプロトタイプの前のチェックボックスをオンにし、必要なボタンをクリックします。

5 ディスカバリのプロトタイプ

概要

このセクションでは、ホストのローレベルディスカバリルールのディスカバリプロトタイプが表示されます。ディスカバリプロトタイプは、親ディスカバリルールのディスカバリルールのネストされたものです。



表示されるデータ:

列	説明
名前	ディスカバリプロトタイプの名前が青いリンクで表示されます。 名前をクリックすると、ディスカバリプロトタイプの設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がホスト名の前にグレーのリンクで表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでディスカバリプロトタイプのリストが開きます。
アイテム	アイテムプロトタイプのリストへのリンクが表示されます。 既存のアイテムプロトタイプの数がグレーで表示されます。
トリガー	トリガープロトタイプのリストへのリンクが表示されます。 既存のトリガープロトタイプの数がグレーで表示されます。
グラフ	グラフプロトタイプのリストへのリンクが表示されます。 既存のグラフプロトタイプの数がグレーで表示されます。
ホスト	ホストプロトタイプのリストへのリンクが表示されます。 既存のホストプロトタイプの数がグレーで表示されます。
ディスカバリルール	ディスカバリプロトタイプのリストへのリンクが表示されます。 既存のディスカバリプロトタイプの数がグレーで表示されます。
キー	ディスカバリに使用されるアイテムキーが表示されます。
間隔	ディスカバリの実行頻度が表示されます。 注意ディスカバリは、リスト下部の今すぐ実行ボタンを押すことで即時に実行することもできます。
タイプ	ディスカバリに使用されるアイテムタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
有効作成	ディスカバリプロトタイプの作成ステータスが表示されます - 有効 (はい) または無効 (いいえ)。ステータスをクリックすることで、はいからいいえ（またはその逆）に変更できます。
ディスカバリ	ディスカバリプロトタイプのディスカバリステータスが表示されます - ディスカバリ (はい) またはディスカバリしない (いいえ)。ステータスをクリックすることで、はいからいいえ（またはその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上隅のディスカバリプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

- 有効で作成 - これらのディスカバリプロトタイプを有効で作成します
- 無効で作成 - これらのディスカバリプロトタイプを無効で作成します
- 削除 - これらのディスカバリプロトタイプを削除します

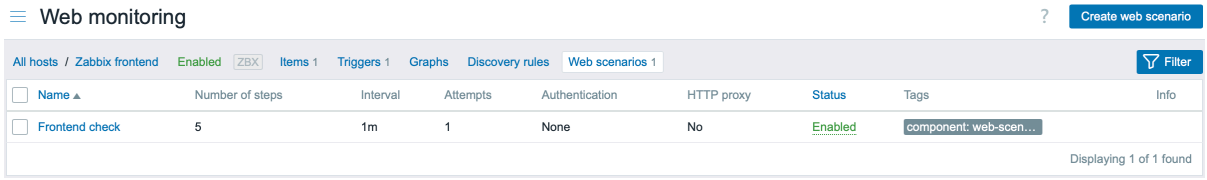
これらのオプションを使用するには、該当するディスカバリプロトタイプの前のチェックボックスをマークし、必要なボタンをクリックします。

5 Web シナリオ

概要

Web シナリオのリストは、データ収集 → ホストから、該当するホストの *Web* をクリックすることでアクセスできます。

既存の Web シナリオのリストが表示されます。



表示されるデータ:

列	説明
名前	Web シナリオの名前。Web シナリオ名をクリックすると、Web シナリオの設定フォームが開きます。 ホストの Web シナリオがテンプレートに属している場合、テンプレート名が Web シナリオ名の前にグレーのリンクとして表示されます。テンプレートリンクをクリックすると、テンプレートレベルで Web シナリオのリストが開きます。
ステップ数	シナリオに含まれるステップの数。
更新間隔	シナリオが実行される頻度。
試行回数	Web シナリオのステップを実行する試行回数。
認証	認証方式が表示されます - Basic、NTLM、または None。
HTTP プロキシ	HTTP プロキシが表示されます。使用しない場合は「No」と表示されます。
ステータス	Web シナリオのステータスが表示されます - 有効または 無効。 ステータスをクリックすることで変更できます。
タグ	Web シナリオのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）が表示されます。タグが 3 つを超える場合は「...」リンクが表示され、マウスオーバーで全てのタグを確認できます。
情報	すべてが正常に動作している場合、この列にはアイコンは表示されません。エラーが発生した場合は、”i” の文字が入った四角いアイコンが表示されます。アイコンにマウスオーバーすると、エラーの説明がツールチップで表示されます。

新しい Web シナリオを設定するには、右上隅の *Web* シナリオの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

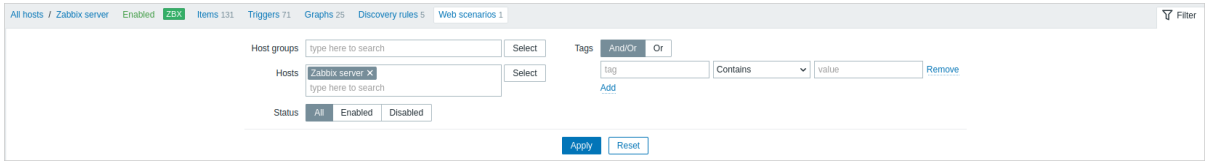
- 有効 - シナリオのステータスを有効に変更します
- 無効 - シナリオのステータスを無効に変更します
- 履歴とトレンドのクリア - シナリオの履歴とトレンドデータをクリアします
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、該当する Web シナリオの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるシナリオのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルターリンクは Web シナリオのリストの上にあります。クリックすると、希望するフィルタリング条件を指定できるフィルターが開きます。



パラメータ	説明
ホストグループ	1 つまたは複数のホストグループでフィルターします。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。
ホスト	1 つまたは複数のホストでフィルターします。 上でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに限定されます。
ステータス	Web シナリオのステータスでフィルターします。

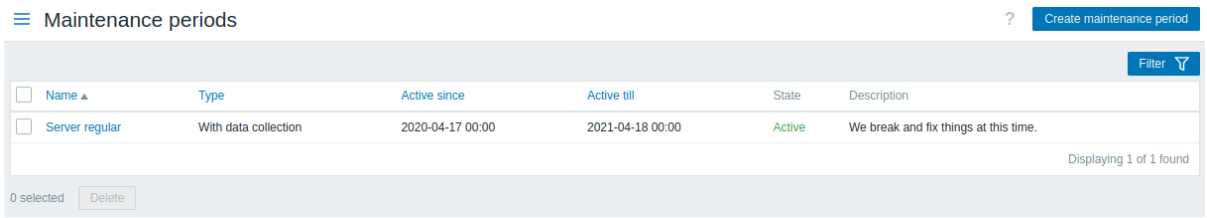
パラメータ	説明
タグ	Web シナリオのタグ名と値でフィルターします。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

5 メンテナンス

概要

データ収集 → メンテナンスセクションでは、ユーザーはホストのメンテナンス期間を設定および管理できます。

既存のメンテナンス期間とその詳細の一覧が表示されます。



表示されるデータ:

列	説明
名前	メンテナンス期間の名前。メンテナンス期間名をクリックすると、メンテナンス期間の設定フォームが開きます。
タイプ	メンテナンスタイプが表示されます: データ収集ありまたは データ収集なし
有効開始	メンテナンス期間の実行が有効になる日時。 注: この時刻はメンテナンス期間を有効にするものではありません。メンテナンス期間は別途設定する必要があります。
有効終了	メンテナンス期間の実行が有効でなくなる日時。
状態	メンテナンス期間の状態: 間近 - まもなく有効になる 有効 - 有効 期限切れ - すでに有効でない
説明	メンテナンス期間の説明が表示されます。

新しいメンテナンス期間を設定するには、右上隅の メンテナンス期間の作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを 1 つ利用できます。

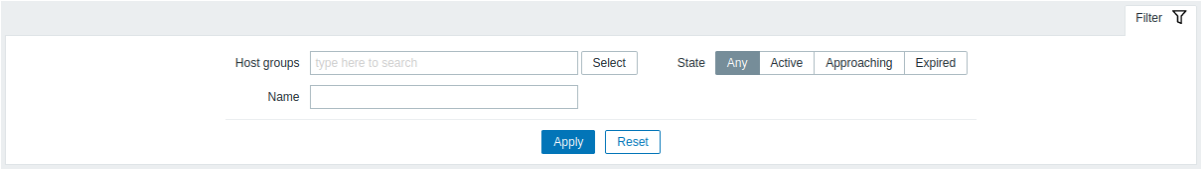
- 削除 - メンテナンス期間を削除します

このオプションを使用するには、該当するメンテナンス期間の前のチェックボックスをオンにして、削除をクリックします。

フィルターの使用

フィルターを使用して、関心のあるメンテナンス期間のみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決のまま検索されます。

フィルターリンクは、メンテナンス期間のリストの上にあります。これをクリックすると、ホストグループ、名前、状態でメンテナンス期間をフィルタリングできるフィルターが利用可能になります。



メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

6 イベント相関関係

概要

データ収集 → イベント相関セクションでは、Zabbix イベントのグローバル相関ルールを設定および管理できます。

表示されるデータ:

列	説明
名前	相関ルールの名前。相関ルール名をクリックすると、ルールの設定フォームが開きます。
条件	相関ルールの条件が表示されます。
オペレーション	相関ルールのオペレーションが表示されます。
ステータス	相関ルールのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。

新しい相関ルールを設定するには、右上隅の イベント相関の作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

- 有効 - 相関ルールのステータスを有効に変更します
- 無効 - 相関ルールのステータスを無効に変更します
- 削除 - 相関ルールを削除します

これらのオプションを使用するには、該当する相関ルールの前のチェックボックスをオンにし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のある相関ルールのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを展開せずに検索されます。

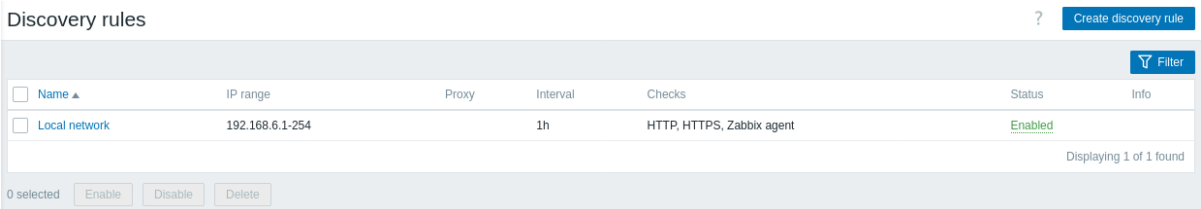
フィルターリンクは相関ルールのリストの上にあります。これをクリックすると、名前やステータスで相関ルールをフィルタリングできるフィルターが利用可能になります。

7 ディスカバリ

概要

データ収集 → ディスカバリセクションでは、ユーザーはディスカバリルールの設定と管理を行うことができます。

既存のディスカバリルールの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
名前	ディスカバリルールの名前。ディスカバリルール名をクリックすると、ディスカバリルールの設定フォームが開きます。
IP 範囲	ネットワークスキャンに使用する IP アドレスの範囲が表示されます。
プロキシ	ディスカバリがプロキシによって実行される場合、プロキシ名が表示されます。
間隔	ディスカバリの実行頻度が表示されます。
チェック	ディスカバリに使用されるチェックの種類が表示されます。
ステータス	ディスカバリルールのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。
情報	すべてが正常に動作している場合、この列には何も表示されません。エラーが発生した場合は、赤い「i」マークの情報アイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明がツールチップで表示されます。

新しいディスカバリルールを設定するには、右上隅の ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

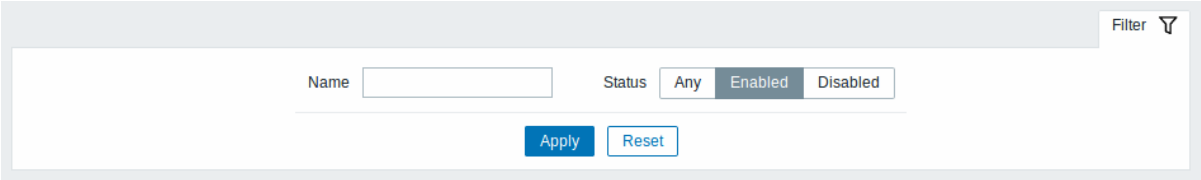
- 有効 - ディスカバリールールのステータスを有効に変更
- 無効 - ディスカバリールールのステータスを無効に変更
- 削除 - ディスカバリールールを削除

これらのオプションを使用するには、該当するディスカバリールールの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターリンクはディスカバリルールのリストの上にあります。これをクリックすると、名前やステータスでディスカバリルールをフィルタリングできるフィルターが利用可能になります。



7 アラート

概要

このメニューには、Zabbix でのアラートの設定に関連するセクションがあります。

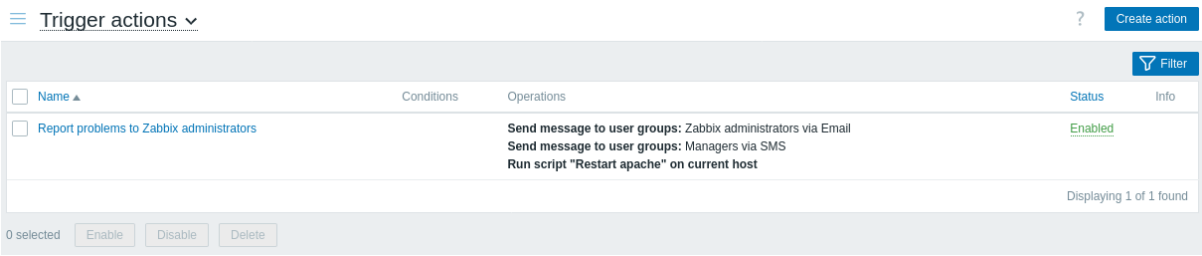
1 アクション

概要

アラート → アクションセクションでは、ユーザーはアクションの設定と管理ができます。

表示されるアクションは、選択したイベントソース (トリガー、サービス、ディスカバリ、自動登録、内部アクション) に割り当てられたアクションです。

別のイベントソースに切り替えるには、アラートメニューセクションの アクションをクリックします。左上のタイトルドロップダウンを使用してソース間を切り替えることもできます。



表示されるデータ:

列	説明
名前	アクションの名前。アクション名をクリックすると、アクションの設定フォームが開きます。
条件	アクションの条件が表示されます。
オペレーション	アクションのオペレーションが表示されます。 オペレーションリストには、通知に使用されるメディアタイプ (メール、SMS、スクリプト) や、通知の受信者の名前と姓 (ユーザー名の後ろに括弧で表示) も表示されます。 アクションのオペレーションは、選択したオペレーションの種類に応じて、通知またはリモートコマンドのいずれかになります。
ステータス	アクションのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。 エスカレーション中にアクションが無効になった場合に何が起こるかについては、エスカレーションセクションを参照してください。
情報	すべてが正常に動作している場合、この列にはアイコンは表示されません。アップグレードやローレベルディスカバリ後にアクションのオペレーションや条件が不足しているなどのエラーが発生した場合は、警告アイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明がツールチップで表示されます。

新しいアクションを設定するには、右上の アクションの作成ボタンをクリックします。

スーパ管理者権限のないユーザーには、権限設定に従ってアクションが表示されます。つまり、特定の権限制限により、スーパ管理者権限のないユーザーがアクションリスト全体を表示できない場合があります。スーパ管理者権限のないユーザーにアクションが表示されるのは、次の条件が満たされている場合です:

- ユーザーがアクション条件のホストグループ、ホスト、テンプレート、トリガーに対して読み書きアクセス権を持っている
- ユーザーがアクションのオペレーション、リカバリオペレーション、更新オペレーションのホストグループ、ホスト、テンプレートに対して読み書きアクセス権を持っている
- ユーザーがアクションのオペレーション、リカバリオペレーション、更新オペレーションのユーザーグループおよびユーザーに対して読み取りアクセス権を持っている

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

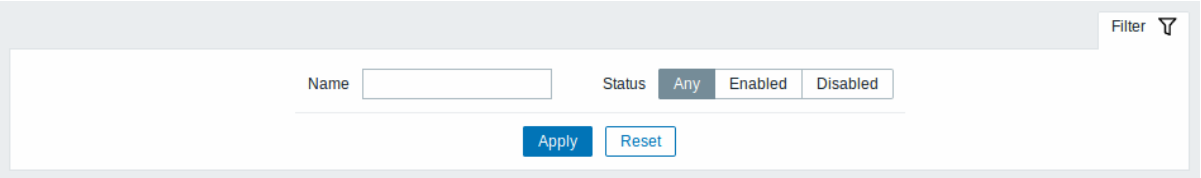
- 有効 - アクションステータスを有効に変更します
- 無効 - アクションステータスを無効に変更します
- 削除 - アクションを削除します

これらのオプションを使用するには、それぞれのアクションの前あるにチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のアクションのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクはアクションリストの上にあります。クリックすると名前とステータスでアクションをフィルタリングできるフィルターが使用可能になります。



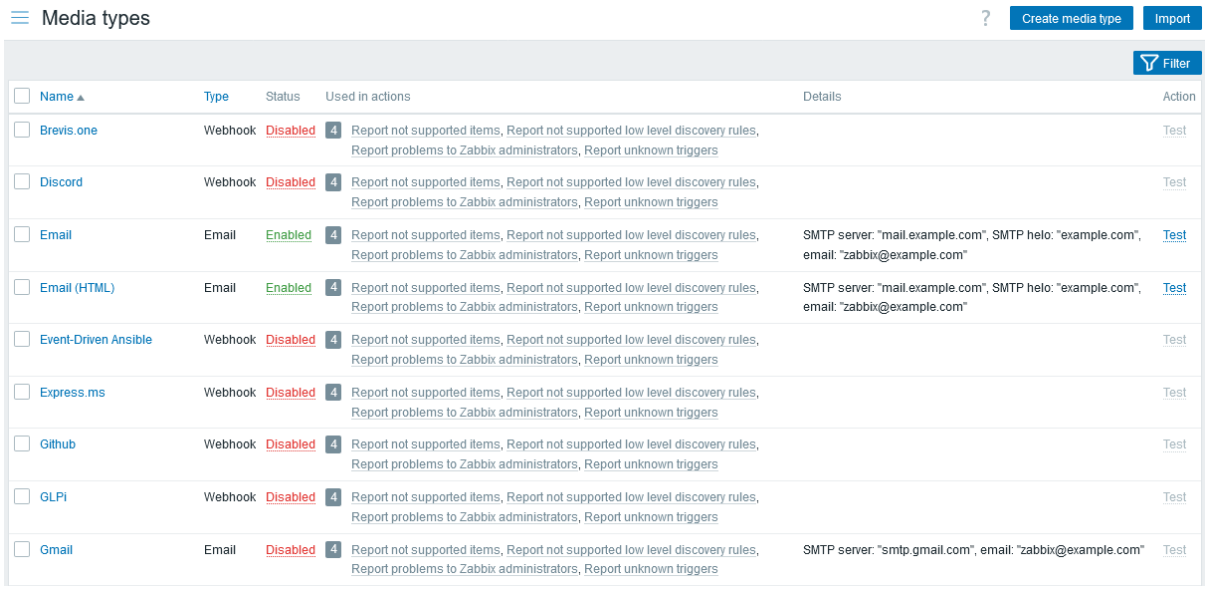
2 メディアタイプ

概要

アラート → メディアタイプセクションでは、ユーザーがメディアタイプ情報を設定および管理できます。

メディアタイプ情報には、通知の配信チャネルとしてメディアを使用するための一般的な手順が含まれています。通知を送信する個々のメールアドレスなどの詳細は、個々のユーザーごとに保持されます。

既存のメディアタイプの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
名前	メディアタイプの名前。名前をクリックすると、メディアタイプの設定フォームが開きます。
タイプ	メディアのタイプ（メール、SMS など）が表示されます。
ステータス	メディアタイプのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。
アクションでの使用	メディアタイプが使用されているアクションが、これらのアクションの合計数を先頭に付けて表示されます。 アクション名をクリックすると、アクションの設定フォームが開きます。ユーザーにアクションへの権限がない場合、名前はクリックできません。
詳細	メディアタイプの詳細情報が表示されます。
アクション	次のアクションが利用可能です: テスト - クリックするとテストフォームが開き、メディアタイプのパラメータ（例: テスト用の件名と本文を含む受信者アドレス）を入力して、設定したメディアタイプが機能するかどうかを確認するためのテストメッセージを送信できます。詳細はメディアタイプのテストも参照してください。

新しいメディアタイプを設定するには、右上隅の メディアタイプの作成ボタンをクリックします。

メディアタイプをインポートするには、右上隅の インポートボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

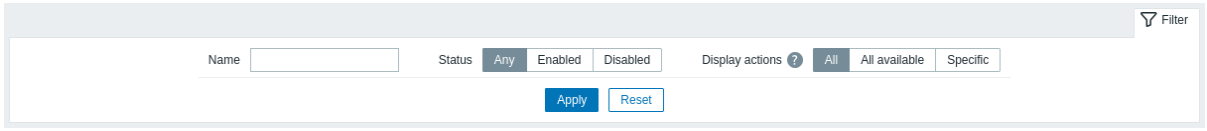
- 有効 - メディアタイプのステータスを *Enabled* に変更します
- 無効 - メディアタイプのステータスを *Disabled* に変更します
- エクスポート - メディアタイプを YAML, XML または JSON ファイルとしてエクスポートします
- 削除 - メディアタイプを削除します

これらのオプションを使用するには、それぞれのメディアタイプの前のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるメディアタイプのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターリンクは、メディアタイプのリストの上にあります。これをクリックすると、名前とステータスでメディアタイプをフィルターできるフィルターが利用可能になります。さらに、フィルターを使用して、メディアタイプの使用範囲（アクション操作の詳細のメディアタイプに送信パラメータで定義）に基づいて、アクションで使用列にアクションを表示できます。



3 スクリプト

概要

アラート → スクリプトセクションでは、ユーザー定義のグローバルスクリプトを設定および管理できます。

グローバルスクリプトは、設定されたスコープおよびユーザー権限に応じて、以下の場所から実行できます。

- ホストメニューから、さまざまなフロントエンドの場所（ダッシュボード、障害、最新データ、マップなど）
- イベントメニューから
- アクションのオペレーションとして実行

スクリプトは、Zabbix エージェント、Zabbix サーバー（プロキシ）、または Zabbix サーバーのみで実行されます。コマンド実行も参照してください。

Zabbix エージェントおよび Zabbix プロキシの両方で、リモートスクリプトはデフォルトで無効になっています。有効にするには、以下の設定が必要です。

- Zabbix エージェントで実行されるリモートコマンドの場合:
  - エージェントの設定ファイルに、許可する各コマンドごとに AllowKey=system.run[<command>,*] パラメータを追加します。* は wait モードと nowait モードの両方を意味します。
- Zabbix プロキシで実行されるリモートコマンドの場合:
  - 注意**: Zabbix** プロキシで監視されている **Zabbix** エージェントでリモートコマンドを実行する場合、**Zabbix** プロキシでリモートコマンドを有効にする必要はありません。ただし、Zabbix プロキシでリモートコマンドを実行する必要がある場合は、プロキシの設定ファイルで *EnableRemoteCommands* パラメータを'1' に設定してください。

Zabbix サーバーでのグローバルスクリプトの実行は、サーバー設定で EnableGlobalScripts=0 を設定することで無効にできます。新規インストールの場合、Zabbix 7.0 以降では、Zabbix サーバーでのグローバルスクリプトの実行はデフォルトで無効になっています。

既存のスクリプトの一覧とその詳細が表示されます。

表示されるデータ:

列	説明
名前	スクリプトの名前。スクリプト名をクリックすると、スクリプトの設定フォームが開きます。
スコープ	スクリプトのスコープ - アクションオペレーション、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトが利用可能な場所が決まります。
アクションでの使用	スクリプトが使用されているすべてのアクションが、これらのアクションの合計数を先頭に付けて表示されます。 アクション名をクリックすると、アクションの設定フォームが開きます。ユーザーにアクションへの権限がない場合、名前はクリックできません。
タイプ	スクリプトのタイプが表示されます - *URL*、*Webhook*、スクリプト、*SSH*、*Telnet*、または *IPMI* コマンド。
実行先	スクリプトが Zabbix エージェント、Zabbix プロキシまたはサーバー、または Zabbix サーバーのみで実行されるかどうかが表示されます。
コマンド	スクリプト内で実行されるすべてのコマンドが表示されます。 Webhook の場合は何も表示されません。
ユーザーグループ	スクリプトが利用可能なユーザーグループが表示されます（すべてのユーザーグループの場合は すべて）。
ホストグループ	スクリプトが利用可能なホストグループが表示されます（すべてのホストグループの場合は すべて）。
ホストアクセス	ホストグループの権限レベルが表示されます - 読み取りまたは 書き込み。必要な権限レベルを持つユーザーのみがスクリプトの実行にアクセスできます。

新しいスクリプトを設定するには、右上隅の スクリプトの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを 1 つ利用できます。

- 削除 - スクリプトの削除

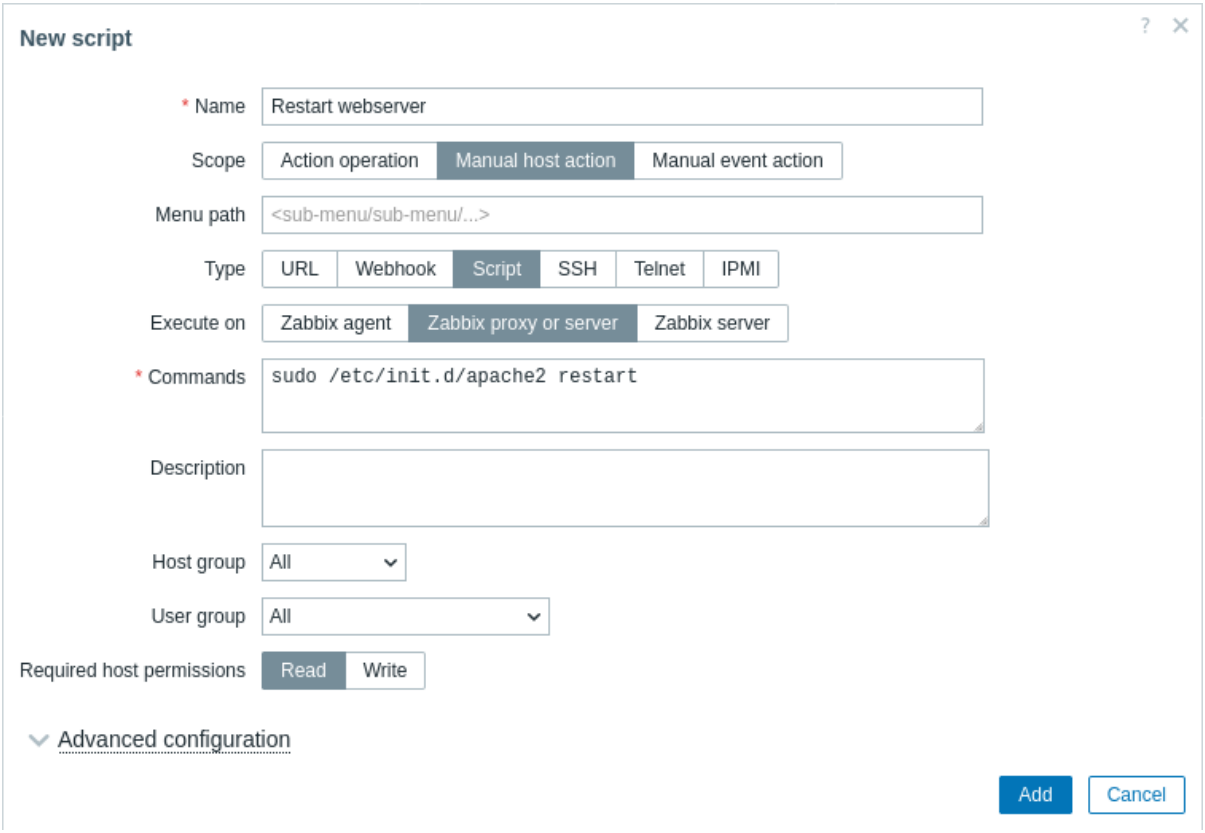
このオプションを使用するには、該当するスクリプトの前のチェックボックスをオンにして、削除をクリックします。

フィルターの使用

フィルターを使用して、関心のあるスクリプトのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

フィルターリンクはスクリプトのリストの上にあります。これをクリックすると、名前やスコープでスクリプトをフィルタリングできるフィルターが利用可能になります。

グローバルスクリプトの設定



スクリプトの属性:

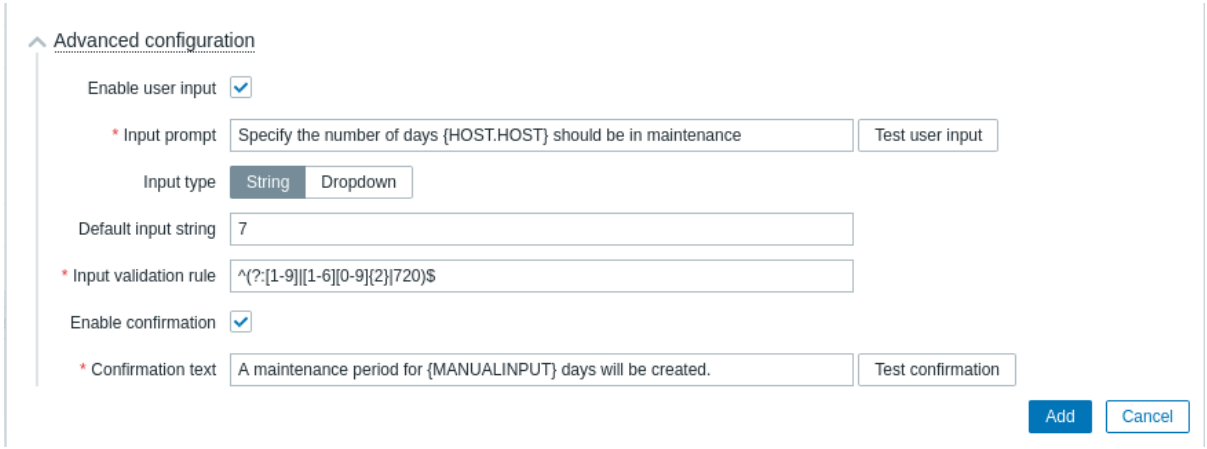
パラメータ	説明
名前	スクリプトの一意の名前。 例: `Clear /tmp filesystem`
スコープ	スクリープトのスコープ - アクション操作、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトが使用できる場所が決まります。アクション操作のリモートコマンド、ホストメニュー、またはイベントメニューからそれぞれ実行できます。 スコープを「アクション操作」に設定すると、アラート → アクションへのアクセス権を持つすべてのユーザーがスクリプトを利用できるようになります。 スクリプトが実際にアクションで使用されている場合、スコープを「アクション操作」以外に変更することはできません。 マクロのサポート スコープは利用可能なマクロの範囲に影響します。たとえば、スクリプトを起動したユーザーに関する情報を渡すために、ユーザー関連のマクロ ({USER.*}) がサポートされています。ただし、アクション操作のスクリプトでは、アクション操作が自動的に実行されるため、サポートされません。 {MANUALINPUT} マクロを使用すると、スクリプト実行時に手動入力を指定できます。これは手動ホストアクションおよび手動イベントアクションのスクリプトでサポートされています。 他にどのマクロがサポートされているかを確認するには、サポートされているマクロテーブルで「トリガーベースの通知とコマンド/トリガーベースのコマンド」、「手動ホストアクションスクリプト」、「手動イベントアクションスクリプト」を検索してください。マクロがスペースを含む値（例: ホスト名）に解決される可能性がある場合は、必要に応じて引用符で囲むことを忘れないでください。

パラメータ		説明
メニューパス		スクリプトの希望するメニューパス。たとえば、`Default` または `Default/` とすると、該当するディレクトリにスクリプトが表示されます。メニューはネスト可能で、例: `Main menu/Sub menu1/Sub menu2`。監視セクションでホスト/イベントメニューからスクリプトにアクセスする場合、指定されたディレクトリに従って整理されます。 このフィールドは、スコープに「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ表示されます。
タイプ		該当するボタンをクリックしてスクリプトタイプを選択します: **URL**、**Webhook**、スクリプト、**SSH**、**Telnet**、または**IPMI**コマンド。 **URL** タイプは、スコープに「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ利用できます。
	スクリプトタイプ: **URL**	
	URL	ホストメニューまたはイベントメニューからのクイックアクセス用の URL を指定します。 マクロおよびカスタムユーザーマクロがサポートされています。マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。 このフィールドで {MANUALINPUT} マクロを使用すると、スクリプト実行時に手動入力を指定できます。例: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` マクロ値は URL エンコードしないでください。
	新しいウィンドウで開く	URL を新しいブラウザタブまたは同じタブで開くかどうかを決定します。
	スクリプトタイプ: **Webhook**	
	パラメータ	Webhook 変数を属性-値のペアで指定します。 参照: Webhookメディアの設定。 パラメータ値ではマクロおよびカスタムユーザーマクロがサポートされています。マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。
	スクリプト	パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。 参照: Webhookメディアの設定、追加の Javascript オブジェクト。
	タイムアウト	JavaScript の実行タイムアウト（1-60 秒、デフォルト 30 秒）。 30s、1m などの時間サフィックスがサポートされています。
	スクリプトタイプ: スクリプト	
	実行先	該当するボタンをクリックしてシェルスクリプトの実行先を選択します: **Zabbix** エージェント - Zabbix エージェント（system.run アイテムが許可されている場合）によってホスト上でスクリプトが実行されます **Zabbix** プロキシまたはサーバー - ホストがプロキシまたはサーバーのどちらで監視されているかに応じて、Zabbix プロキシまたはサーバーによってスクリプトが実行されます。 EnableRemoteCommandsで有効になっている場合、プロキシで実行されます。 EnableGlobalScriptsサーバーパラメータでグローバルスクリプトが有効になっている場合、サーバーで実行されます。 **Zabbix** サーバー - スクリプトは Zabbix サーバーでのみ実行されます。 このオプションは、EnableGlobalScriptsサーバーパラメータでグローバルスクリプトが無効になっている場合は利用できません。
	コマンド	スクリプト内で実行するコマンドのフルパスを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。カスタムユーザーマクロがサポートされています。
	スクリプトタイプ: **SSH**	
	認証方式	認証方式を選択します - パスワードまたは公開鍵。
	ユーザー名	ユーザー名を入力します。
	パスワード	パスワードを入力します。 このフィールドは認証方式に「パスワード」を選択した場合に利用できます。
	公開鍵ファイル	公開鍵ファイルのパスを入力します。 このフィールドは認証方式に「公開鍵」を選択した場合に利用できます。

パラメータ		説明
	秘密鍵ファイル	秘密鍵ファイルのパスを入力します。 このフィールドは認証方式に「公開鍵」を選択した場合に利用できます。
	パスフレーズ	パスフレーズを入力します。 このフィールドは認証方式に「公開鍵」を選択した場合に利用できます。
	ポート	ポートを入力します。
	コマンド	コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。カスタムユーザーマクロがサポートされています。
	スクリプトタイプ: **Telnet**	
	ユーザー名	ユーザー名を入力します。
	パスワード	パスワードを入力します。
	ポート	ポートを入力します。
	コマンド	コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。カスタムユーザーマクロがサポートされています。
	スクリプトタイプ: **IPMI**	
	コマンド	IPMI コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記スコープを参照）。カスタムユーザーマクロがサポートされています。
説明		スクリプトの説明を入力します。
ホストグループ		スクリプトが利用可能となるホストグループを選択します（またはすべてのホストグループの場合はすべて）。
ユーザーグループ		スクリプトが利用可能となるユーザーグループを選択します（またはすべてのユーザーグループの場合はすべて）。 このフィールドは、スコープに「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ表示されます。
必要なホスト権限		ホストグループの権限レベルを選択します - 読み取りまたは書き込み。必要な権限レベルを持つユーザーのみがスクリプトの実行にアクセスできます。 このフィールドは、スコープに「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ表示されます。
詳細設定		詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。 このフィールドは、スコープに「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ表示されます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用できます。



パラメータ	説明
ユーザー入力を有効にする	スクリプトの実行前に手動でユーザー入力を有効にするには、チェックボックスをオンにします。 手動ユーザー入力は、スクリプト内の {MANUALINPUT} マクロ値を置き換えます。 詳細は、手動ユーザー入力を参照してください。
入力プロンプト	カスタムユーザー入力を促すカスタムテキストを入力します。このテキストは、手動入力ポップアップの入力フィールドの上に表示されます。 手動入力ポップアップのプレビューを表示するには、ユーザー入力のテストをクリックします。プレビューでは、入力文字列が入力検証ルール（下記パラメータ参照）に準拠しているかどうかもテストできます。 マクロおよびユーザーマクロのサポートは、スクリプトのスコープによって異なります（一般スクリプト設定パラメータのスコープを参照）。
入力タイプ	手動入力タイプを選択します: 文字列 - 単一の文字列; ドロップダウン - 複数のドロップダウンオプションから値を選択。
ドロップダウンオプション	ユーザー入力ドロップダウンの一意の値をカンマ区切りリストで入力します。 ドロップダウンに空のオプションを含めるには、リストの先頭、中間、または末尾にカンマを追加します。 このフィールドは、入力タイプで「ドロップダウン」が選択されている場合のみ表示されます。
デフォルト入力文字列	ユーザー入力のデフォルト文字列（またはなし）を入力します。 このフィールドは、入力検証ルールフィールドで指定された正規表現に対して検証されます。 ここで入力した値は、手動入力ポップアップでデフォルトで表示されます。 このフィールドは、入力タイプで「文字列」が選択されている場合のみ表示されます。
入力検証ルール	ユーザー入力文字列を検証するための正規表現を入力します。 グローバル正規表現がサポートされています。 このフィールドは、入力タイプで「文字列」が選択されている場合のみ表示されます。
確認を有効にする	スクリプトの実行前に確認メッセージを表示するには、チェックボックスをオンにします。この機能は、（リブートスクリプトのような）危険な操作や長時間かかる可能性のある操作で特に便利です。
確認テキスト	上記のチェックボックスで有効にした確認ポップアップ用のカスタム確認テキストを入力します（例：リモートシステムが再起動されます。本当に実行しますか？）。テキストの表示例を確認するには、フィールド横の確認のテストをクリックします。 マクロおよびカスタムユーザーマクロがサポートされています。 注: 確認メッセージのテスト時にはマクロは展開されません。

手動ユーザー入力と確認メッセージの両方が設定されている場合、連続したポップアップウィンドウで表示されます。

手動ユーザー入力

手動ユーザー入力を使用すると、スクリプトの実行ごとにカスタムパラメータを指定できます。これにより、パラメータが 1 つだけ異なる複数の類似したユーザースクリプトを作成する必要がなくなります。

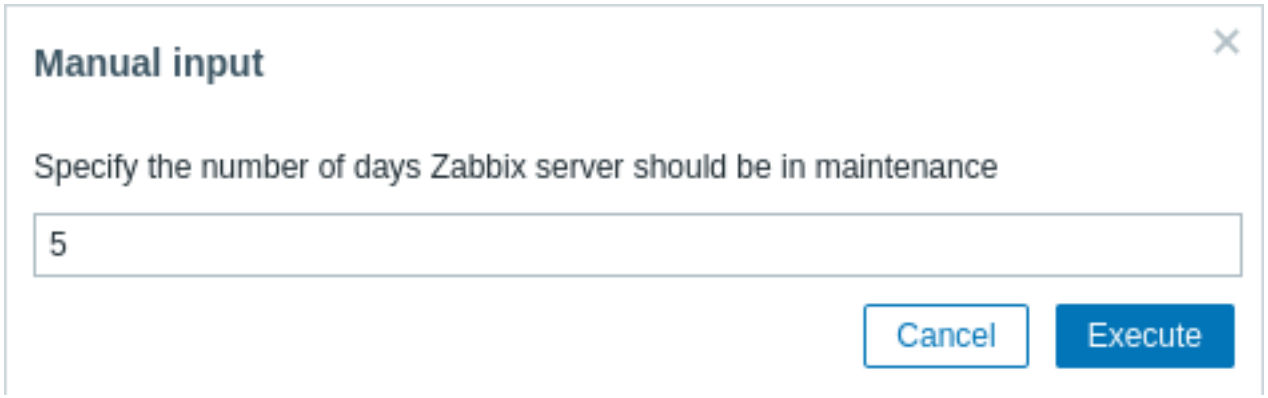
たとえば、スクリプトの実行時に異なる整数や異なる URL アドレスを指定したい場合があります。

手動ユーザー入力を有効にするには、次の手順を実行します。

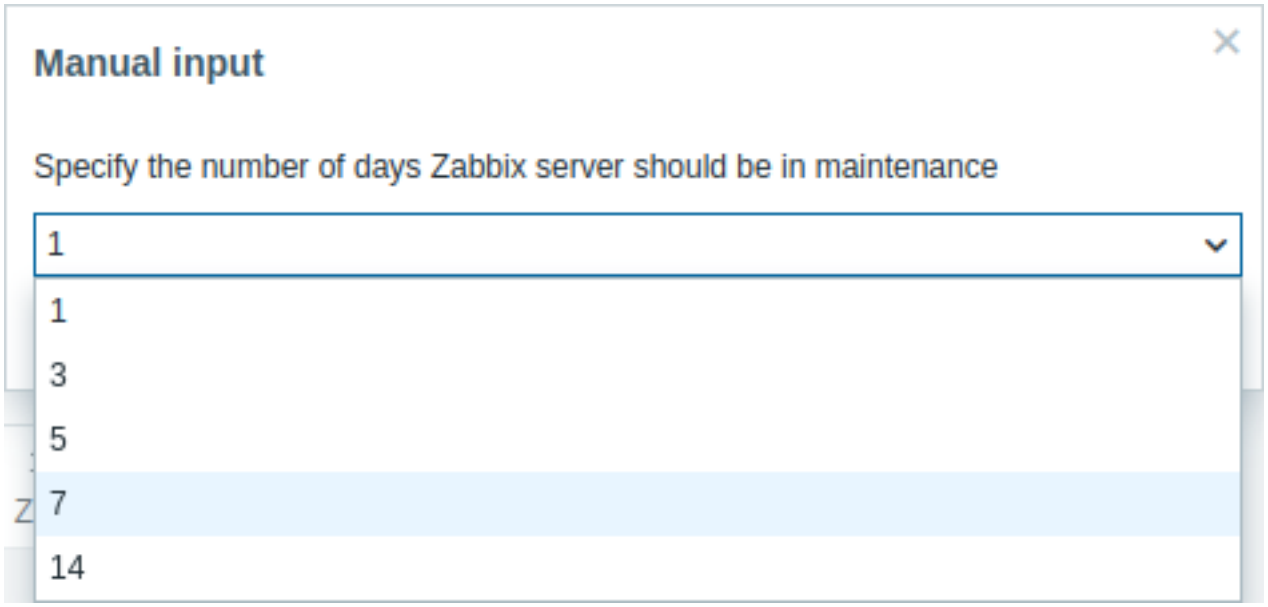
- 必要な場所でスクリプト（コマンド、スクリプト、スクリプトパラメータ）内、または URL スクリプトの URL フィールド内で {MANUALINPUT} マクロを使用します。
- 詳細スクリプト設定で手動ユーザー入力を有効にし、入力オプションを設定します。

ユーザー入力が有効になっている場合、スクリプトの実行前に手動入力のポップアップがユーザーに表示され、カスタム値の入力を求められます。指定された値はスクリプト内の {MANUALINPUT} と置き換えられます。

設定によっては、ユーザーは文字列値の入力を求められます。



または、あらかじめ決められたオプションのドロップダウンから値を選択します。



手動ユーザー入力は、スコープが「手動ホストアクション」または「手動イベントアクション」のスクリプトでのみ利用できます。

スクリプトの実行と結果

Zabbix サーバーによって実行されるスクリプトは、コマンド実行ページで説明されている順序で実行されます。

スクリプトの結果は、スクリプト実行後に表示されるポップアップウィンドウに表示されます。スクリプトの戻り値は標準出力です:

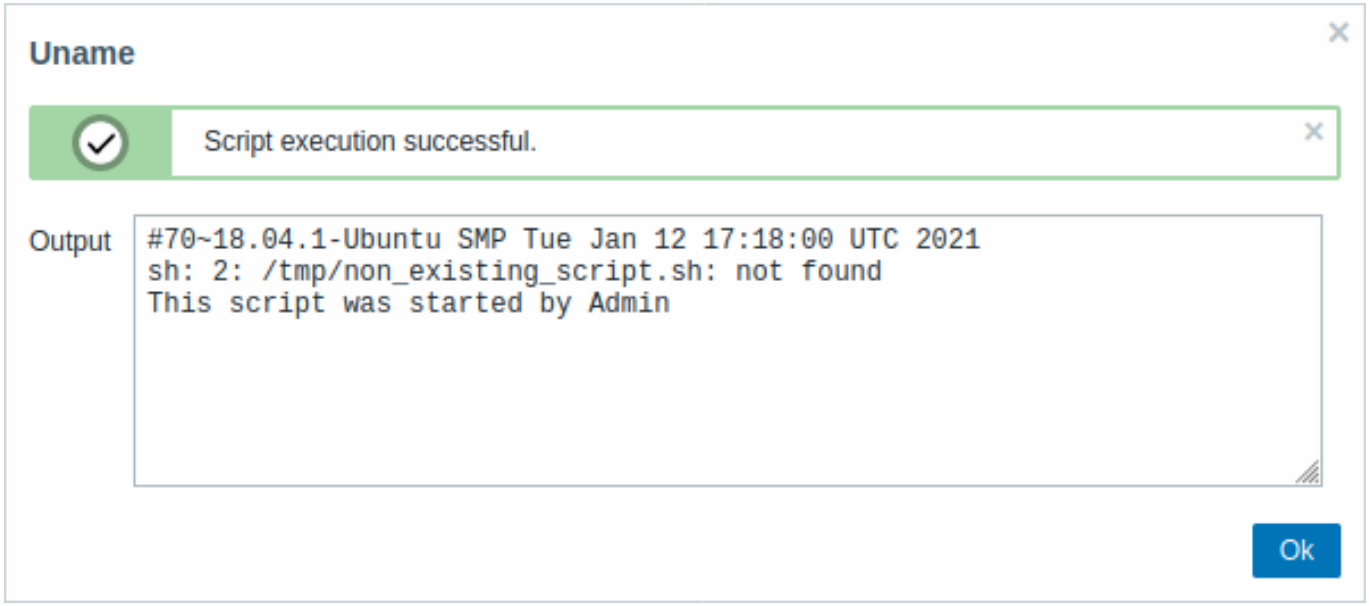
- スクリプトが正常に終了した場合（終了コードが 0）、戻り値は 16MB に制限されます（末尾の空白も含み、切り捨てられます）。
- スクリプトがエラーで終了した場合（非ゼロの終了コード）、戻り値は標準エラー出力で 2KB に制限されます。

Zabbix はデフォルトで拡張スクリプト出力を保存しません。出力の詳細を保持するには、スクリプト内でロギングを実装することができます（例: 出力をローカルログファイルにリダイレクトするなど）。

Zabbix サーバーまたは Zabbix プロキシで実行されるスクリプトについては、データベースの制限も適用されることに注意してください。

以下はスクリプトと結果ウィンドウの例です:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



スクリプトの結果にはスクリプト自体は表示されません。

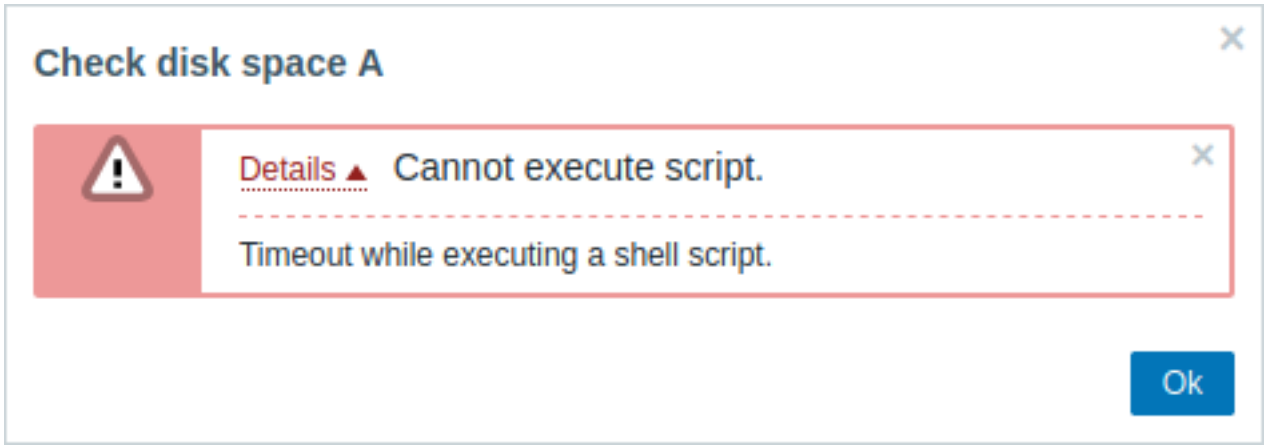
スクリプトのタイムアウト

Zabbix エージェント

スクリプトの実行中にタイムアウトが発生する場合があります。

Zabbix エージェントでスクリプトを実行した例と、その結果ウィンドウを以下に示します。

```
sleep 5
df -h
```



この場合のエラーメッセージは次のとおりです。

```
Timeout while executing a shell script.
```

このような状況を回避するには、Zabbix エージェントの設定やZabbix サーバーの設定の `Timeout` パラメータを調整するのではなく、スクリプト自体（上記の例では「5」）を最適化することを推奨します。ただし、Zabbix エージェントのアクティブモードの場合は、Zabbix エージェントの設定の `RefreshActiveChecks` パラメータよりも、Zabbix サーバーの設定の `Timeout` パラメータを数秒以上長く設定する必要があります。これにより、サーバーがエージェントからアクティブチェックの結果を受信するのに十分な時間が確保されます。アクティブエージェントでのスクリプト実行は、Zabbix エージェント 7.0 以降でサポートされています。

Zabbix エージェントの設定で `Timeout` パラメータを変更した場合、次のエラーメッセージが表示されます。

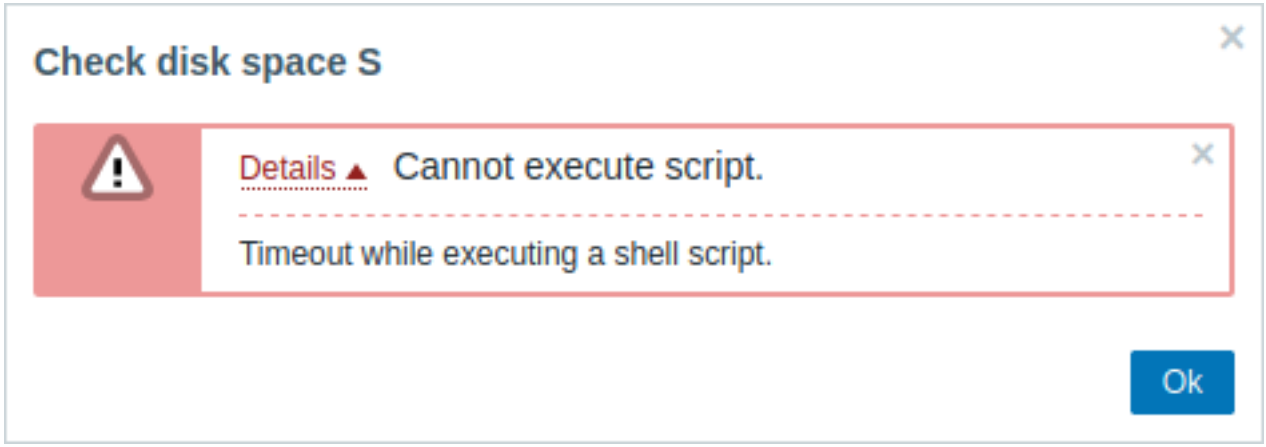
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

これは、Zabbix エージェントの設定で変更が行われたことを意味しますが、Zabbix サーバーの設定の `Timeout` パラメータも変更する必要があります。

Zabbix サーバー/プロキシ

Zabbix サーバーでスクリプトを実行した例と、下の結果ウィンドウを参照してください:

```
sleep 11
df -h
```



また、Zabbix サーバーの設定を変更して `TrapperTimeout` パラメータを対応する値 (この場合は `11` より大きい値) に調整するのではなく、スクリプト自体を最適化することを推奨します。

8 ユーザー

概要

このメニューには、Zabbix でのユーザーの設定に関連するセクションがあります。このメニューは、SuperAdminタイプのユーザーのみが利用できます。

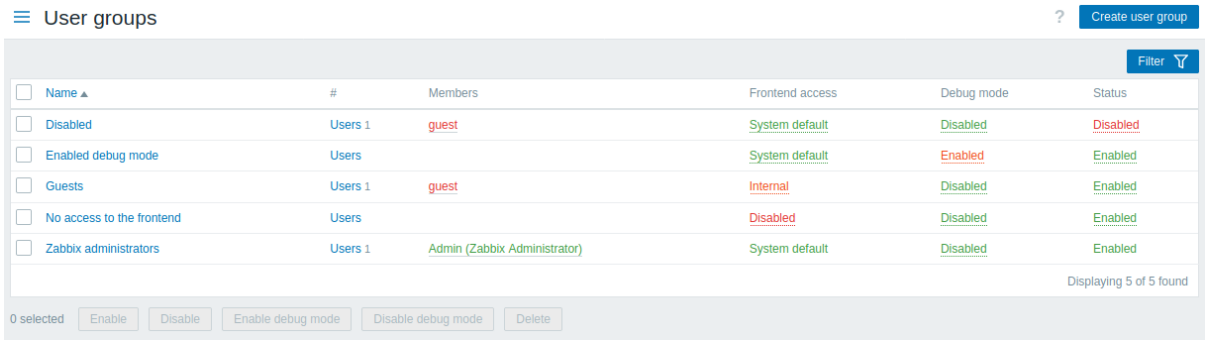
1 ユーザーグループ

概要

ユーザー → ユーザーグループセクションでは、システムのユーザーグループをメンテナンスできます。

ユーザーグループ

既存のユーザーグループの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
名前	ユーザーグループの名前。ユーザーグループ名をクリックすると、ユーザーグループの設定フォームが開きます。
#	グループ内のユーザー数。ユーザーをクリックすると、該当するユーザーがユーザーリストでフィルタリングされて表示されます。
メンバー	ユーザーグループ内の個々のユーザーのユーザー名（括弧内に名前と姓）。ユーザー名をクリックすると、ユーザーの設定フォームが開きます。無効なグループのユーザーは赤色で表示されます。
フロントエンドアクセス	フロントエンドのアクセスレベルが表示されます: システムデフォルト - ユーザーは Zabbix、LDAP、または HTTP（グローバルに設定された認証方法による）で認証されます; 内部 - ユーザーは Zabbix で認証されます; HTTP 認証がグローバルデフォルトの場合は無視されます; **LDAP** - ユーザーは LDAP で認証されます; HTTP 認証がグローバルデフォルトの場合は無視されます; 無効 - このグループの Zabbix フロントエンドへのアクセスは禁止されます。 現在のレベルをクリックすると変更できます。

列	説明
デバッグモード	デバッグモードのステータスが表示されます - 有効または無効。ステータスをクリックすると変更できます。
ステータス	ユーザーグループのステータスが表示されます - 有効または無効。ステータスをクリックすると変更できます。

新しいユーザーグループを設定するには、右上のユーザーグループの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

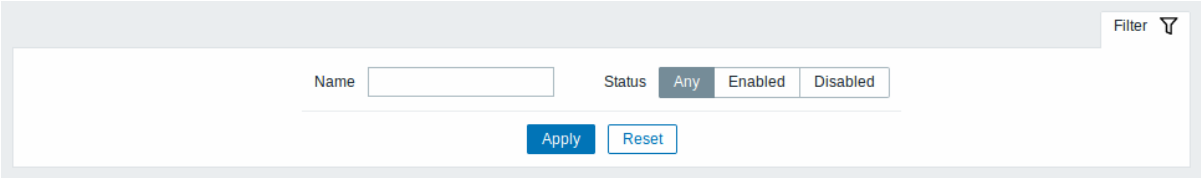
- 有効 - ユーザーグループのステータスを有効に
- 無効 - ユーザーグループのステータスを無効に
- デバッグモードを有効 - ユーザーグループのデバッグモードを有効に
- デバッグモードを無効 - ユーザーグループのデバッグモードを無効に
- 削除 - ユーザーグループを削除

これらのオプションを使用するには、それぞれのユーザーグループの前にチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、関心のあるユーザーグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ユーザーグループのリストの上にあります。それをクリックすると、名前とステータスでユーザーグループをフィルタリングできます。



2 ユーザーの役割

概要

ユーザー > ユーザーロールセクションでは、ユーザーロールを作成できます。

ユーザーロールを使用すると、最初に選択したユーザータイプ（ユーザー、管理者、スーパ管理者）に基づいて、きめ細かな権限を作成できます。

ユーザータイプを選択すると、このユーザータイプに利用可能なすべての権限が付与されます（デフォルトでチェックされています）。

権限は、ユーザータイプで利用可能なサブセットからのみ取り消すことができ、ユーザータイプで利用可能な範囲を超えて拡張することはできません。

利用できない権限のチェックボックスはグレー表示されます。この要素への直接 URL をブラウザに入力しても、ユーザーはその要素にアクセスできません。

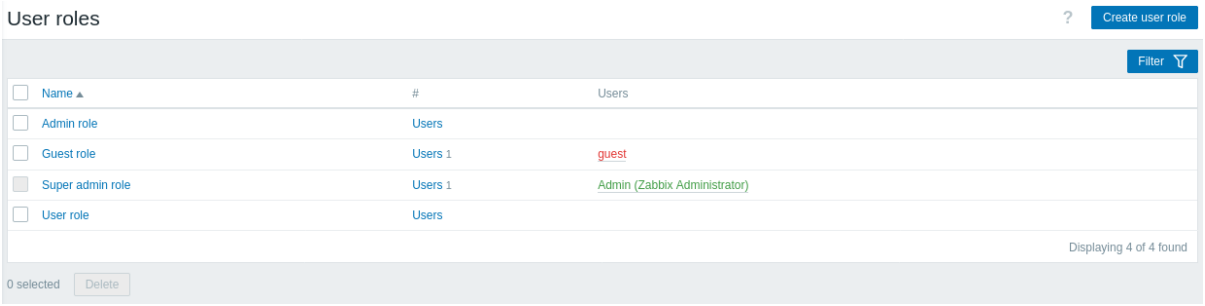
Note:
一部の UI 要素へのアクセスを制限しても、そのページを開くことだけが防止され、インターフェースの他の部分で基礎となるデータにアクセスする機能が削除されるわけではありません。

ユーザーロールはシステムユーザーに割り当てることができます。各ユーザーには 1 つのロールしか割り当てることができません。

デフォルトのユーザーロール

デフォルトでは、Zabbix には 4 つのユーザーロールが設定されており、あらかじめ定義された権限セットが割り当てられています。

- ゲストロール
- ユーザーロール
- 管理者ロール
- スーパ管理者ロール

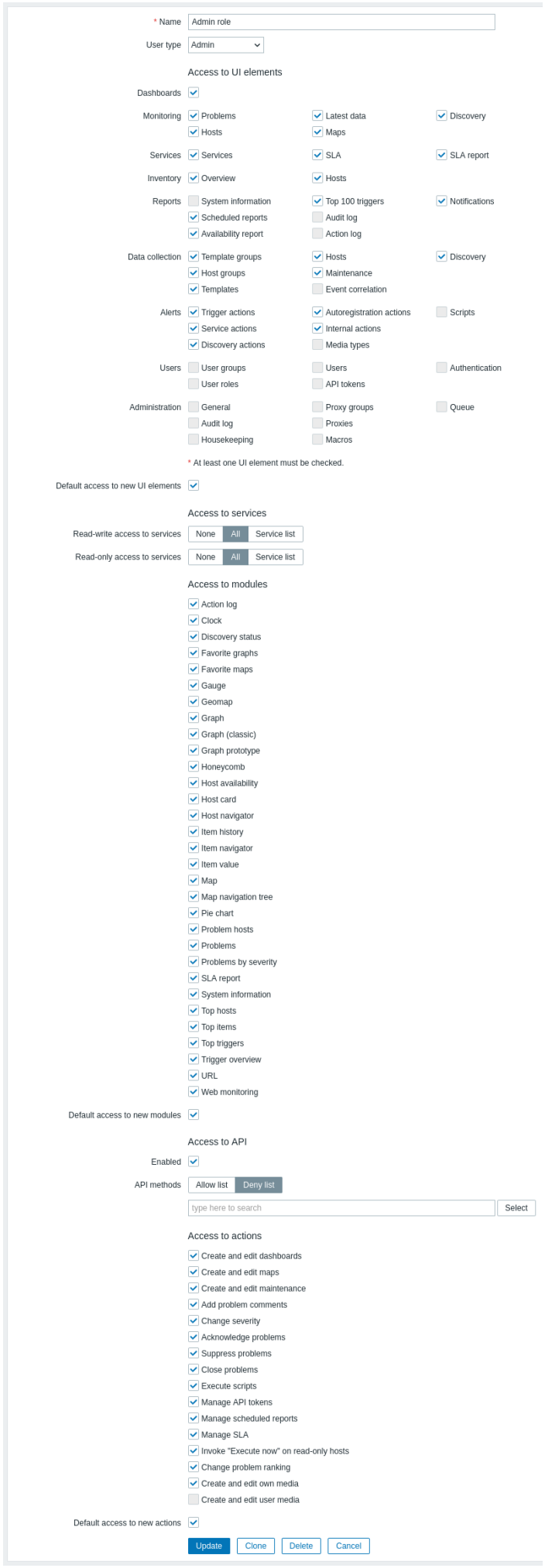


これらは Zabbix の主なユーザータイプに基づいています。それぞれのロールに割り当てられているすべてのユーザーのリストが表示されます。無効なグループに含まれるユーザーは赤色で表示されます。ゲストロールは、いくつかのフロントエンドセクションを閲覧する権限のみを持つユーザータイプのロールです。

Note:
デフォルトのスーパ管理者ロールは、Zabbix に無制限の権限を持つスーパ管理者ユーザーが少なくとも 1 人存在する必要があるため、変更や削除はできません。スーパ管理者タイプのユーザーは、自分のロールの設定を変更できますが、ユーザータイプは変更できません。

設定

新しいロールを作成するには、右上隅のユーザーロールの作成ボタンをクリックします。既存のロールを更新するには、ロール名をクリックして設定フォームを開きます。


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Default access to new actions
Update
Clone
Delete
Cancel

利用可能な権限が表示されます。特定の権限を取り消すには、そのチェックボックスのチェックを外します。

Zabbix で事前設定されている各ユーザーロールのデフォルトとともに、利用可能な権限について以下で説明します。

デフォルトの権限

UI 要素へのアクセス

メニューセクションへのデフォルトのアクセスは、ユーザータイプによって異なります。詳細は権限ページを参照してください。

その他のオプションへのアクセス

パラメータ		説明	デフォルトのユーザーロール スーパ管理者ロール	管理者ロール	ユーザーロール	ゲストロール
新しい *UI* 要素へのデフォルトアクセス		このオプションは、Zabbix のアップグレード後に新しいメニューセクションへのアクセス方法を指定します。 モジュールの既存のメニューセクションには影響しません。	はい	はい	はい	はい
サービスへの読み書きアクセス		サービスへの読み書きアクセスを選択します: なし - 全くアクセスできません すべて - すべてのサービスへのアクセスが読み書き可能 サービスリスト - 読み書きアクセスするサービスを選択 読み書きアクセスが付与されている場合、読み取り専用アクセスの設定よりも優先され、子サービスにも動的に継承されます。	すべて	すべて	なし	なし
	タグ付きサービスへの読み書きアクセス	タグ名と、必要に応じて値を指定し、タグに一致するサービスへの読み書きアクセスを追加で付与します。 このオプションは、サービスへの読み書きアクセスパラメータで「サービスリスト」が選択されている場合に利用できます。 読み書きアクセスが付与されている場合、読み取り専用アクセスの設定よりも優先され、子サービスにも動的に継承されます。				

	サービスへの読み取り専用アクセス	サービスへの読み取り専用アクセスを選択します: なし - 全くアクセスできません すべて - すべてのサービスへのアクセスが読み取り専用 サービスリスト - 読み取り専用アクセスするサービスを選択 読み取り専用アクセスは、読み書きアクセスよりも優先されず、子サービスにも動的に継承されます。			すべて	すべて
	タグ付きサービスへの読み取り専用アクセス	タグ名と、必要に応じて値を指定し、タグに一致するサービスへの読み取り専用アクセスを追加で付与します。 このオプションは、サービスへの読み取り専用アクセスパラメータで「サービスリスト」が選択されている場合に利用できます。 読み取り専用アクセスは、読み書きアクセスよりも優先されず、子サービスにも動的に継承されます。				
モジュールへのアクセス						
	< モジュール名 >	特定のモジュールへのアクセスを許可/拒否します。このセクションには有効なモジュールのみが表示されます。現在無効になっているモジュールへのアクセスを許可または制限することはできません。	はい	はい	はい	はい
	新しいモジュールへのデフォルトアクセス	このオプションは、Zabbix のアップグレード後に新しいモジュールやウィジェットへのアクセス方法を指定します。また、管理 > 一般 > モジュールセクションで追加されたモジュールやウィジェットにも適用されます。				
API へのアクセス						
	有効	API へのアクセスを有効/無効にします。	はい	はい	はい	いいえ

	API メソッド	検索フィールドに指定した API メソッドを許可する場合は許可リスト、拒否する場合は拒否リストを選択します。なお、一部の API メソッドを許可し、他を拒否することはできません。 検索フィールドにメソッド名を入力し始めると、オートコンプリートリストからメソッドを選択できます。 また、[選択] ボタンを押して、このユーザータイプで利用可能な全メソッドのリストから選択することもできます。アクセス権限ブロックの特定のアクションがチェック解除されている場合、ユーザーはそのアクションに関連する API メソッドを使用できません。 ワイルドカードがサポートされています。例: `dashboard.*`（'dashboard.' API サービスのすべてのメソッド）、`*`（任意のメソッド）、`*.export`（すべての API サービスの'.export' 名のメソッド）。 メソッドが指定されていない場合、許可/拒否リストルールは無視されます。				
アクションへのアクセス	ダッシュボードの作成と編集	このチェックボックスをクリアすると、対応する要素の`.create`、`.update`、`.delete`API メソッドの使用権も取り消されます。	はい	はい	はい	いいえ
	マップの作成と編集					
	メンテナンスの作成と編集				いいえ	
	問題コメントの追加	このチェックボックスをクリアすると、`event.acknowledge`API メソッドによる対応するアクションの実行権も取り消されます。			はい	
	深刻度の変更					
	問題の認識					
	問題の抑制					
	問題のクローズ					
	スクリプトの実行	このチェックボックスをクリアすると、`script.execute`API メソッドの使用権も取り消されます。				
	API トークンの管理	このチェックボックスをクリアすると、すべての`token.`API メソッドの使用権も取り消されます。				
	スケジュールレポートの管理	このチェックボックスをクリアすると、すべての`report.`API メソッドの使用権も取り消されます。			いいえ	
	SLA の管理	SLAの管理権限を有効/無効にします。				
	読み取り専用ホストで「今すぐ実行」を呼び出す	読み取り専用ホストのアイテムに対して最新データで「今すぐ実行」オプションを使用できるようにします。			はい	
	問題ランクの変更	原因から症状、またはその逆に問題ランクを変更できるようにします。				
	自身のメディアの作成と編集	自身のメディアを作成/編集できるようにします。				

ユーザーメディアの作成と編集	ユーザーのメディアを作成/編集できるようにします。 このオプションはスーパ管理者ユーザーのみ利用可能です。	いいえ	いいえ
新しいアクションへのデフォルトアクセス	このオプションは、Zabbix のアップグレード後に新しいアクションへのアクセス方法を指定します。	はい	はい

関連情報:

- ユーザーの設定

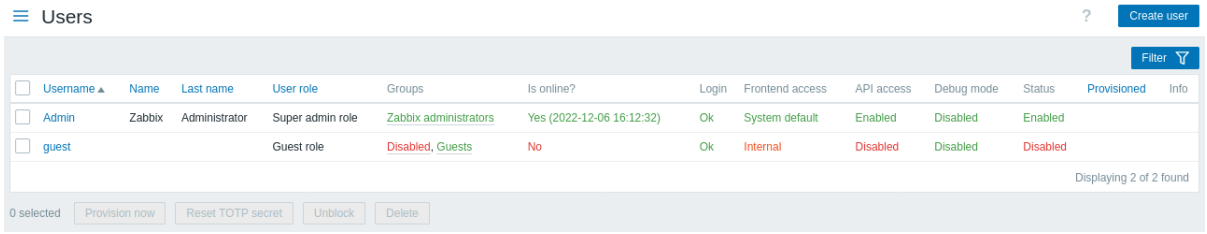
3 ユーザー

概要

ユーザー → ユーザーセクションでシステムのユーザーを管理できます。

ユーザー

既存のユーザーのリストとその詳細が表示されます。



表示されるデータ:

列	説明
ユーザー名	Zabbix へのログインに使用するユーザー名。ユーザー名をクリックすると、ユーザーの設定フォームが開きます。
名	ユーザーの名。
姓	ユーザーの姓。
ユーザーロール	ユーザーロールが表示されます。
グループ	ユーザーが所属するグループが一覧表示されます。ユーザーグループ名をクリックすると、ユーザーグループの設定フォームが開きます。無効なグループは赤色で表示されます。
オンラインか？	ユーザーのオンライン状態が表示されます - はいまたは いいえ。最後のユーザーアクティビティの時刻が括弧内に表示されます。
ログイン	ユーザーのログイン状態が表示されます - *Ok* または ブロック。管理 → 一般 → その他 セクションで設定された失敗したログイン試行回数（デフォルトは 5 回）を超えると、ユーザーは一時的にブロックされることがあります。ブロックをクリックすると、ユーザーのブロックを解除できます。
フロントエンドアクセス	フロントエンドのアクセスレベルが表示されます - システムデフォルト、内部、*LDAP*、または無効。ユーザーグループ全体に設定されたものに応じて表示されます。
API アクセス	API アクセスの状態が表示されます - 有効または 無効。ユーザーロールに設定されたものに応じて表示されます。
デバッグモード	デバッグモードの状態が表示されます - 有効または 無効。ユーザーグループ全体に設定されたものに応じて表示されます。
ステータス	ユーザーのステータスが表示されます - 有効または 無効。ユーザーグループ全体に設定されたものに応じて表示されます。
プロビジョニング済み	ユーザーが最後にプロビジョニングされた日付が表示されます。 LDAP/SAML から JIT プロビジョニングで作成されたユーザーに使用されます。
情報	エラーに関する情報が表示されます。 ユーザーグループがないユーザーには黄色の警告が表示されます。 ロールがないユーザー、およびロールとユーザーグループがないユーザーには赤色の警告が表示されます。

新しいユーザーを設定するには、右上のユーザーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで、一括編集オプションを利用できます:

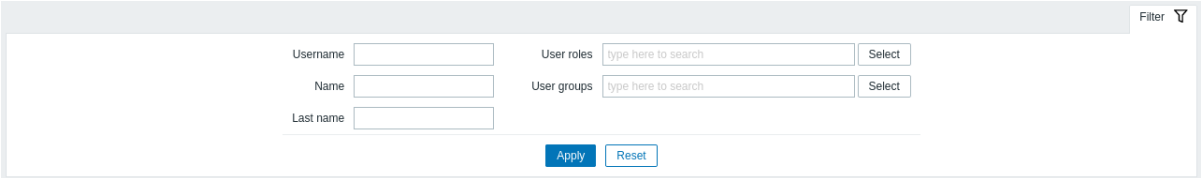
- 今すぐプロビジョニング - LDAP からユーザー情報を更新します（このオプションはLDAPユーザーが選択されている場合のみ有効です）
- *TOTP* シークレットのリセット - すべての TOTP 方式のユーザー TOTP シークレットをリセットし、ユーザーセッションを削除します（このオプションはMFAが有効な場合のみ有効です。TOTP シークレットを持たないユーザーのセッションは削除されません）
- ブロック解除 - ブロックされたユーザーのシステムアクセスを再度有効にします
- 削除 - ユーザーを削除します

これらのオプションを使用するには、該当するユーザーの前のチェックボックスをマークし、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるユーザーのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決の状態で検索されます。

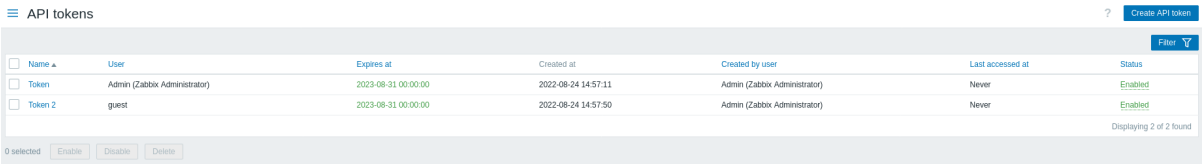
フィルターリンクはユーザーのリストの上にあります。クリックすると、ユーザー名、名、姓、ユーザーロール、ユーザーグループでユーザーをフィルタリングできるフィルターが利用可能になります。



4 API トークン

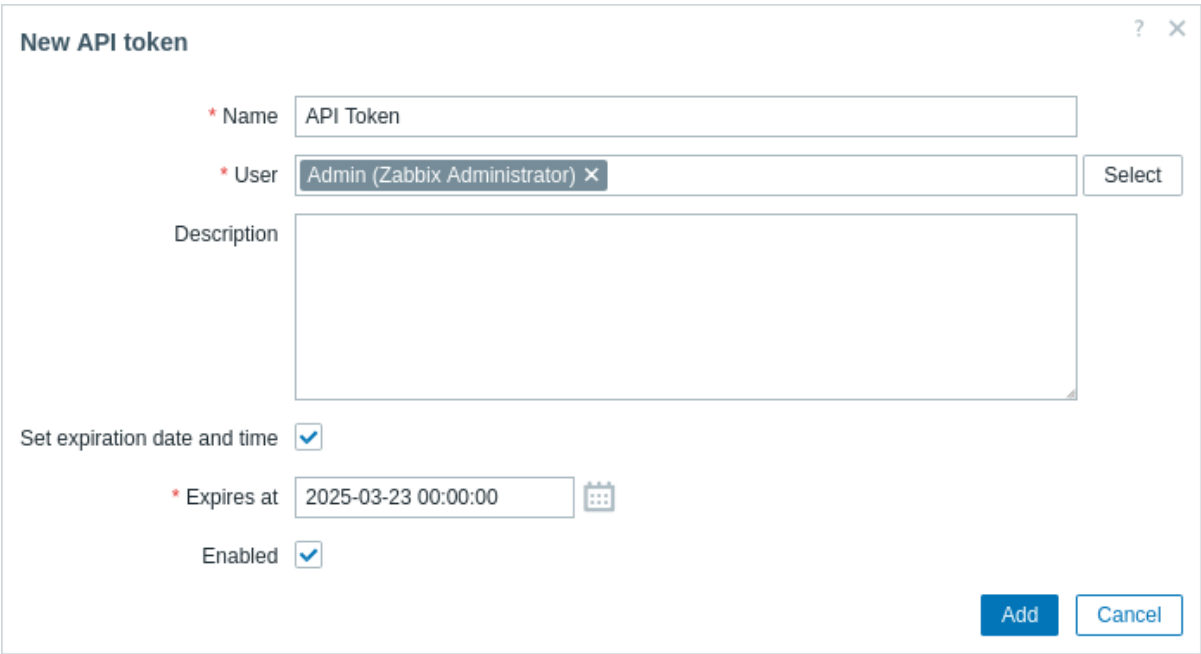
概要

このセクションでは、API トークンの作成と管理ができます。



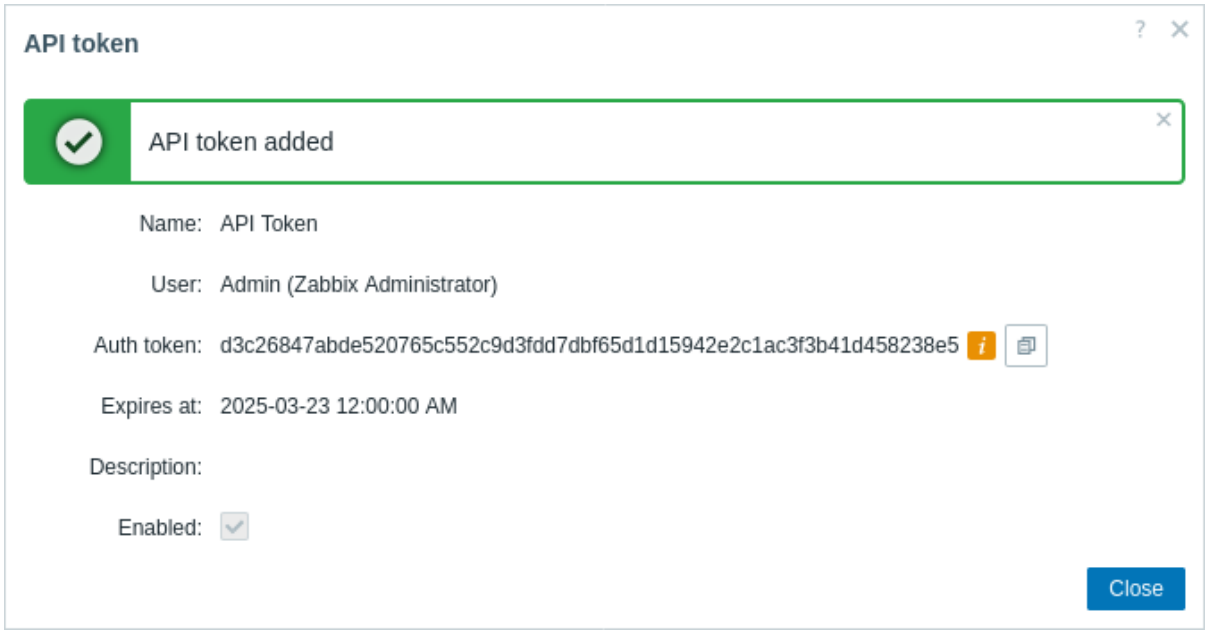
API トークンは、名前、トークンが割り当てられているユーザー、有効期限、トークンを作成したユーザー、またはステータス（有効/無効）でフィルタリングできます。リスト内のトークンステータスをクリックすると、トークンをすばやく有効/無効にできます。また、リストでトークンを選択し、リスト下部の有効/無効ボタンをクリックすることで、複数のトークンをまとめて有効/無効にすることもできます。

新しいトークンを作成するには、右上の *API* トークンの作成ボタンを押し、トークン設定画面で必要な項目を入力します。



パラメータ	説明
名前	トークンの表示名。
ユーザー	トークンを割り当てるユーザー。ユーザー名、姓または名を入力し始めると、オートコンプリートリストから必要なユーザーを選択できます。または、選択ボタンを押して、全ユーザーリストからユーザーを選択することもできます。トークンは 1 人のユーザーにのみ割り当て可能です。
説明	トークンの説明（任意）。
有効期限を設定	トークンに有効期限を設定しない場合は、このチェックボックスを外します。
有効期限	カレンダーアイコンをクリックしてトークンの有効期限を選択するか、YYYY-MM-DD hh:mm:ss 形式で手動入力します。
有効	無効状態でトークンを作成する場合は、このチェックボックスを外します。

追加を押してトークンを作成します。



認証トークンの値をコピーしてページを閉じる前に安全な場所に保存し、[閉じる] を押します。トークンがリストに表示されます。

Warning:
認証トークンの値は後から再表示できません。トークン作成直後のみ表示されます。保存したトークンを紛失した場合は再生成する必要があり、その際は新しい認証文字列が作成されます。

トークン名をクリックすると、名前、説明、有効期限設定、トークンステータスを編集できます。なお、トークンの割り当て先ユーザーは変更できません。[更新] ボタンを押して変更を保存します。トークンを紛失したり漏洩した場合は、[再生成] ボタンを押して新しいトークン値を生成できます。確認ダイアログが表示され、操作を続行すると以前のトークンは無効になります。

管理メニューセクションへのアクセス権がないユーザーでも、[ユーザープロファイル → API トークン] セクションで、自分に割り当てられたトークンの詳細を表示・変更できます。ただし、ユーザーロール権限で *API* トークンの管理が許可されている場合のみです。

5 認証

概要

ユーザー → 認証セクションでは、Zabbix のユーザー認証方法と内部パスワードの要件を指定できます。

利用可能な認証方法は、内部認証、HTTP 認証、LDAP 認証、SAML 認証、MFA 認証です。

デフォルト認証

デフォルトでは、Zabbix はすべてのユーザーに対して内部 Zabbix 認証を使用します。

デフォルトの認証方法をLDAPにシステム全体で変更することも可能です。そのためには、*LDAP* タブに移動して LDAP パラメータを設定し、認証タブに戻ってデフォルト認証セレクタを LDAP に切り替えます。

認証方法はユーザーグループ単位で細かく調整することもできます。たとえ LDAP 認証がグローバルに設定されていても、一部のユーザーグループは Zabbix によって認証されることがあります。これらのグループはフロントエンドアクセスを内部に設定する必要があります。

また、グローバルで内部認証を使用している場合でも、特定のユーザーグループに対してのみ LDAP 認証を有効にすることも可能です。この場合、LDAP 認証の詳細を指定し、フロントエンドアクセスを LDAP に設定する必要があります。ユーザーが LDAP 認証のユーザーグループに 1 つでも含まれている場合、そのユーザーは内部認証方法を使用できなくなります。

HTTP、SAML 2.0、および MFA 認証方法は、デフォルトの認証方法に加えて使用できます。

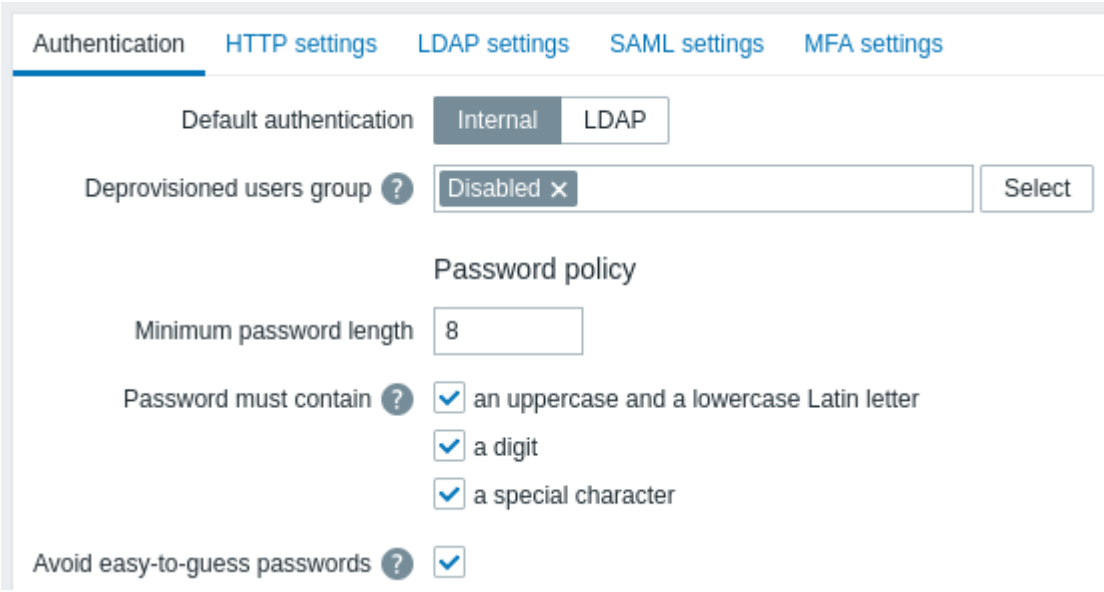
Zabbix はジャストインタイム（JIT）プロビジョニングをサポートしており、外部ユーザーが初めて認証した際に Zabbix にユーザーアカウントを作成し、これらのユーザーアカウントをプロビジョニングできます。JIT プロビジョニングは LDAP および SAML でサポートされています。

関連項目:

- HTTP 認証
- LDAP 認証
- SAML 認証
- MFA 認証

設定

認証タブでは、デフォルトの認証方法の設定、プロビジョニング解除されたユーザーのグループの指定、Zabbix ユーザーのパスワード複雑性要件の設定ができます。



設定パラメータ:

パラメータ	説明
デフォルト認証	Zabbix のデフォルト認証方法を内部または *LDAP* から選択します。
プロビジョニング解除されたユーザーグループ	プロビジョニング解除されたユーザーのユーザーグループを指定します。この設定は、JIT プロビジョニングで、LDAP または SAML システムから Zabbix に作成されたが、もはやプロビジョニングが不要なユーザーにのみ必要です。 無効化されたユーザーグループを指定する必要があります。
最小パスワード長	デフォルトでは、最小パスワード長は 8 に設定されています。サポートされる範囲: 1-70。72 文字を超えるパスワードは切り捨てられることに注意してください。
パスワードに含める必要があるもの	パスワードに指定した文字を使用することを要求するには、1 つまたは複数のチェックボックスをマークします: - 大文字および小文字のラテン文字 - 数字 - 特殊文字 各オプションの文字リストのヒントを見るには、クエスチョンマークにカーソルを合わせてください。

パラメータ	説明
推測しやすいパスワードを避ける	チェックを入れると、パスワードは以下の要件に対してチェックされます: - ユーザーの名前、姓、またはユーザー名を含まないこと - 一般的またはコンテキスト固有のパスワードのいずれでもないこと 一般的およびコンテキスト固有のパスワードのリストは、NCSC「Top 100k passwords」リスト、SecLists「Top 1M passwords」リスト、および Zabbix コンテキスト固有パスワードリストから自動的に生成されます。内部ユーザーは、このリストに含まれるパスワードを設定できません。これらのパスワードは一般的に使用されているため、弱いと見なされます。

パスワード複雑性要件の変更は既存のユーザーパスワードには影響しませんが、既存ユーザーがパスワードを変更する場合は、新しいパスワードが現在の要件を満たす必要があります。要件のリストのヒントは、ユーザープロファイルおよびユーザー → ユーザーメニューからアクセスできるユーザー設定フォームのパスワードフィールドの横に表示されます。

1 HTTP

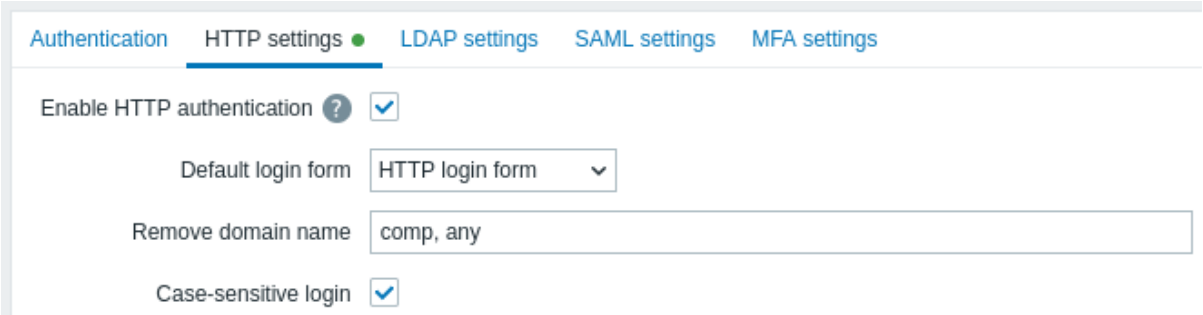
概要

HTTP または Web サーバーベースの認証（例：BasicAuthentication、NTLM/Kerberos）を使用してユーザー名とパスワードを確認できます。ユーザーは Zabbix にも存在する必要がありますが、その Zabbix パスワードは使用されません。

Attention:
注意してください！Web サーバー認証が正しく設定され、正常に動作していることを確認してから有効にしてください。

HTTP 認証は、ユーザー > 認証セクションの *HTTP* 設定タブで対応するオプションを設定することで、フロントエンドで無効にできます。HTTP 認証が無効になっている場合、HTTP 認証オプションのタブはフロントエンドに表示されません。フロントエンドを再インストール（setup.php の実行）すると、HTTP 認証の設定を含む認証設定がリセットされることに注意してください。

設定



設定パラメータ:

パラメータ	説明
HTTP 認証を有効にする	チェックボックスをマークすると、HTTP 認証が有効になります。にマウスを重ねると、ヒントボックスが表示され、Web サーバー認証の場合、すべてのユーザー（フロントエンドアクセスが LDAP/内部に設定されている場合でも）は Zabbix ではなく Web サーバーによって認証されることが警告されます。
デフォルトのログインフォーム	認証されていないユーザーをどこに誘導するかを指定します: **Zabbix** ログインフォーム - 標準の Zabbix ログインページ。 **HTTP** ログインフォーム - HTTP ログインページ。 `index_http.php` ページのみに Web サーバーベースの認証を有効にすることを推奨します。デフォルトのログインフォームが「HTTP ログインページ」に設定されている場合、Web サーバー認証モジュールが `$_SERVER` 変数に有効なユーザーログインを設定すると、ユーザーは自動的にログインされます。 サポートされている `$_SERVER` キーは `PHP_AUTH_USER`、`REMOTE_USER`、`AUTH_USER` です。
ドメイン名を削除	ユーザー名から削除するドメイン名のカンマ区切りリスト。 例: `comp,any` - ユーザー名が'Admin@any'、'comp\Admin' の場合、'Admin' としてログインします。ユーザー名が'notacompany\Admin' の場合、ログインは拒否されます。

パラメータ	説明
大文字小文字を区別したログイン	ユーザー名の大文字小文字を区別したログインを無効にするには、チェックボックスのマークを外します（デフォルトで有効）。 大文字小文字を区別しないログインを無効にすると、たとえば Zabbix ユーザーが”Admin”や”ADMIN” であっても、”admin” としてログインできます。 大文字小文字を区別しないログインが無効で、似たユーザー名（例: Admin と admin）を持つ複数の Zabbix ユーザーがいる場合、これらのユーザーのログインは常に次のエラーメッセージで拒否されることに注意してください: ”Authentication failed: supplied credentials are not unique.”

Note:
HTTP 認証情報（デフォルトで HTTP ログインフォームが設定されている場合）でログインできず 401 エラーとなる内部ユーザーについては、基本認証ディレクティブに `ErrorDocument 401 /index.php?form=default` 行を追加することで、通常の Zabbix ログインフォームにリダイレクトできます。

2 LDAP

概要

外部 LDAP 認証 を使用して、ユーザー名とパスワードを確認できます。

Zabbix の LDAP 認証は、少なくとも Microsoft Active Directory および OpenLDAP で動作します。

LDAP サインインのみが設定されている場合、ユーザーは Zabbix にも存在する必要がありますが、その Zabbix パスワードは使用されません。認証が成功すると、Zabbix は LDAP から返されたユーザー名属性とローカルのユーザー名を照合します。

ユーザーのプロビジョニング

LDAP ユーザーに対して JIT（ジャストインタイム）ユーザーのプロビジョニングを設定することができます。この場合、Zabbix にユーザーがすでに存在している必要はありません。ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

LDAP ユーザーが LDAP のログイン名とパスワードを入力すると、Zabbix はデフォルトの LDAP サーバーでこのユーザーが存在するかどうかを確認します。ユーザーが存在し、まだ Zabbix にアカウントがない場合は、Zabbix に新しいユーザーが作成され、ユーザーはログインできるようになります。

Attention:
JIT プロビジョニングが有効な場合は、認証タブでプロビジョニング解除されたユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングでは、LDAP の変更に基づいてプロビジョニングされたユーザーアカウントを更新することもできます。たとえば、ユーザーがある LDAP グループから別のグループに移動された場合、Zabbix でもユーザーはあるグループから別のグループに移動されます。ユーザーが LDAP グループから削除された場合、Zabbix でもそのグループから削除され、他のグループに属していない場合は、プロビジョニング解除されたユーザー用のユーザーグループに追加されます。プロビジョニングされたユーザーアカウントは、設定されたプロビジョニング期間またはユーザーが Zabbix にログインしたときに更新されます。

バックグラウンドプロビジョニングは、ユーザーが Zabbix フロントエンドとやり取りしている場合、または少なくともブラウザでフロントエンドページを開いている場合に、Zabbix フロントエンドによって実行されることに注意してください。ユーザーをプロビジョニングするための専用のバックグラウンドプロセスはありません。

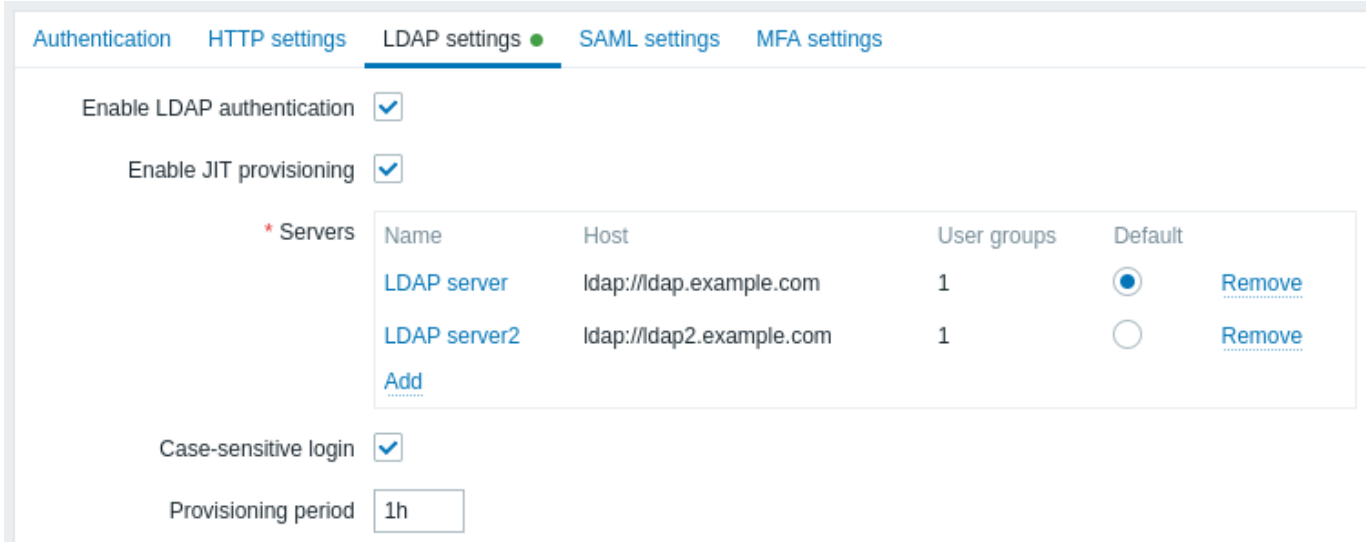
LDAP JIT プロビジョニングは、LDAP がバインドに「匿名」または「特別なユーザー」を使用するように設定されている場合にのみ利用できます。直接ユーザーバインディングの場合、そのタイプのバインディングにはログインユーザーのパスワードが使用されるため、プロビジョニングはユーザーログインアクションに対してのみ行われます。

複数のサーバー

必要に応じて、複数の LDAP サーバーを定義できます。たとえば、異なるユーザーグループを認証するために異なるサーバーを使用できます。LDAP サーバーが設定されると、ユーザーグループの設定で、該当するユーザーグループに必要な LDAP サーバーを選択できるようになります。

ユーザーが複数のユーザーグループおよび複数の LDAP サーバーに所属している場合、名前で昇順にソートされた LDAP サーバーのリストの最初のサーバーが認証に使用されます。

設定



設定パラメータ:

パラメータ	説明
LDAP 認証を有効にする	チェックボックスをオンにすると、LDAP 認証が有効になります。
JIT プロビジョニングを有効にする	チェックボックスをオンにすると、JIT プロビジョニングが有効になります。
サーバー	追加をクリックして LDAP サーバーを設定します（下記のLDAP サーバーの設定を参照）。
大文字と小文字を区別したログイン	チェックボックスをオフにすると、ユーザー名の大文字と小文字を区別したログインが無効になります（デフォルトでは有効）。 大文字と小文字を区別したログインを無効にすると、たとえば Zabbix ユーザーが「Admin」や「ADMIN」であっても「admin」としてログインできます。 大文字と小文字を区別したログインを無効にし、同じようなユーザー名（例：Admin と admin）を持つ複数の Zabbix ユーザーがいる場合、これらのユーザーのログインは常に次のエラーメッセージで拒否されることに注意してください：「認証に失敗しました：入力された認証情報が一意ではありません。」
プロビジョニング期間	プロビジョニング期間を設定します。つまり、フロントエンドで作業している間、ログインしたユーザーがどのくらいの頻度でプロビジョニングされるかを指定します。

LDAP サーバーの設定

LDAP サーバーの設定パラメータ:

パラメータ	説明
名前	Zabbix の設定での LDAP サーバー名。
ホスト	LDAP サーバーのホスト名、IP または URI。例: ldap.example.com, 127.0.0.1, ldap://ldap.example.com セキュアな LDAP サーバーの場合は、*ldaps* プロトコルとホスト名を使用します。例: ldaps://ldap.example.com OpenLDAP 2.x.x 以降では、ldap://hostname:port または ldaps://hostname:port の形式の完全な LDAP URI を使用できます。
ポート	LDAP サーバーのポート。デフォルトは 389 です。 セキュアな LDAP 接続の場合、通常は 636 です。 完全な LDAP URI を使用する場合は使用されません。
Base DN	LDAP サーバー上のユーザーアカウントへのベースパス: ou=Users,ou=system (OpenLDAP の場合), DC=company,DC=com (Microsoft Active Directory の場合) uid=%{user},dc=example,dc=com (直接ユーザーバインディングの場合、下記の注意参照)
検索属性	検索に使用する LDAP アカウント属性: uid (OpenLDAP の場合), sAMAccountName (Microsoft Active Directory の場合)

パラメータ	説明
Bind DN	LDAP サーバー上でバインドおよび検索を行うための LDAP アカウント。例: uid=ldap_search,ou=system (OpenLDAP の場合), CN=ldap_search,OU=user_group,DC=company,DC=com (Microsoft Active Directory の場合) 匿名バインディングもサポートされています。匿名バインディングは、ドメイン設定を不正なユーザーに公開する可能性があることに注意してください（ユーザー、コンピューター、サーバー、グループ、サービスなどの情報）。セキュリティ上の理由から、LDAP ホストで匿名バインドを無効にし、認証付きアクセスを使用してください。
Bind パスワード	LDAP サーバー上でバインドおよび検索を行うアカウントの LDAP パスワード。
説明	LDAP サーバーの説明。
JIT プロビジョニングの設定	このチェックボックスをオンにすると、JIT プロビジョニングに関連するオプションが表示されます。
グループ設定	グループ設定方法を選択: **memberOf** - ユーザーとそのグループメンバーシップ属性で検索 **groupOfNames** - メンバー属性でグループを検索 memberOf の方が高速なため推奨されます。LDAP サーバーが `memberOf` をサポートしていない場合やグループフィルタリングが必要な場合は groupOfNames を使用してください。
グループ名属性	`memberOf` 属性内のすべてのオブジェクトからグループ名を取得する属性を指定します（ユーザーグループメンバーシップ属性フィールド参照） グループ名はユーザーグループマッピングに必要です。
ユーザーグループメンバーシップ属性	ユーザーが所属するグループ情報を含む属性を指定します（例: `memberOf`）。 例えば、memberOf 属性は次のような情報を持つ場合があります: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` このフィールドは memberOf メソッドでのみ利用可能です。
ユーザー名属性	ユーザーの名を含む属性を指定します。
ユーザー姓属性	ユーザーの姓を含む属性を指定します。
ユーザーグループマッピング	LDAP ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 プロビジョニングされたユーザーが Zabbix でどのユーザーグループ/ロールを取得するかを決定するために必要です。 追加をクリックしてマッピングを追加します。 *LDAP* グループパターンフィールドはワイルドカードをサポートします。グループ名は既存のグループと一致する必要があります。 LDAP ユーザーが複数の Zabbix ユーザーグループに一致する場合、ユーザーはすべてのグループのメンバーになります。 ユーザーが複数の Zabbix ユーザーロールに一致する場合、ユーザーはその中で最も高い権限レベルのロールを取得します。
メディアタイプマッピング	ユーザーの LDAP メディア属性（例: メール）を Zabbix ユーザーメディアにマッピングし、通知を送信します。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます（下記参照）。
StartTLS	LDAP サーバーに接続する際に StartTLS 操作を使用するには、このチェックボックスをオンにします。サーバーが StartTLS をサポートしていない場合、接続は失敗します。 StartTLS は *ldaps* プロトコルを使用するサーバーでは使用できません。
検索フィルター	LDAP でユーザーを認証する際にカスタム文字列を定義します。以下のプレースホルダーがサポートされています: `%{attr}` - 検索属性名（uid, sAMAccountName） `%{user}` - 認証するユーザー名の値 大文字小文字を区別しない LDAP や Microsoft Active Directory 環境で大文字小文字を区別した検索を行う場合、次のように定義できます: `(%{attr}:caseExactMatch:=%{user})`。 フィルターをカスタマイズしない場合、LDAP はデフォルトで `(%{attr}=%{user})` を使用します。

Note:
直接ユーザーバインディング用に LDAP サーバーを設定するには、*Base DN* パラメータに uid=%{user} 属性を追加し（例: *uid=%{user},dc=example,dc=com*）、*BindDN* および *Bind* パスワードパラメータを空にします。認証時には、%{user} プレースホルダーがログイン時に入力されたユーザー名に置き換えられます。

グループ設定方法として”groupOfNames” を選択した場合、以下のフィールドが特有となります:

パラメータ	説明
グループ *Base DN*	LDAP サーバー上のグループへのベースパス。
グループ名属性	グループへのベースパスでグループ名を取得する属性を指定します。 グループ名はユーザーグループマッピングに必要です。
グループメンバー属性	LDAP 内のグループメンバー情報を含む属性を指定します（例: `member`）。
参照属性	グループフィルター用の参照属性を指定します（グループフィルターフィールド参照）。 ここで指定した属性の値を取得するためにグループフィルターで`%{ref}` を使用します。
グループフィルター	ユーザーがメンバーであるグループを取得するためのフィルターを指定します。 例えば、(`member=uid=%{ref},ou=Users,dc=example,dc=com`) は、グループの member 属性が `uid=User1,ou=Users,dc=example,dc=com` の場合に”User1” に一致し、そのユーザーがメンバーであるグループを返します。

Warning:
証明書に問題がある場合、セキュアな LDAP 接続（ldaps）を機能させるには /etc/openldap/ldap.conf 設定ファイルに `TLS_REQCERT allow` 行を追加する必要がある場合があります。この設定は LDAP カタログへの接続のセキュリティを低下させる可能性があります。

Note:
LDAP サーバー上でバインドおよび検索を行うために、実際のユーザーアカウント（Zabbix フロントエンドへのログインに使用される）ではなく、LDAP 内で最小限の権限を持つ別の LDAP アカウント（*Bind DN*）を作成することを推奨します。
この方法はセキュリティを高め、ユーザーが LDAP サーバーで自身のパスワードを変更した場合でも *Bind* パスワードを変更する必要がありません。
上記の表では *ldap_search* アカウント名です。

アクセステスト

テストボタンでユーザーアクセスをテストできます:

パラメータ	説明
ログイン	テストする LDAP ユーザー名 (Zabbix フロントエンドの現在のユーザー名で自動入力されます)。このユーザー名は LDAP サーバーに存在している必要があります。 Zabbix は、テストユーザーの認証に失敗した場合、LDAP 認証を有効にしません。
ユーザーパスワード	テストする LDAP ユーザーパスワード。

3 SAML

概要

SAML 2.0 認証を使用して Zabbix にサインインできます。

SAML サインインのみが設定されている場合、ユーザーは Zabbix にも存在している必要がありますが、その Zabbix パスワードは使用されません。認証が成功すると、Zabbix は SAML から返されたユーザー名属性とローカルのユーザー名を照合します。

ユーザーのプロビジョニング

SAML ユーザーに対して JIT（ジャストインタイム）ユーザープロビジョニングを設定することができます。この場合、Zabbix にユーザーがすでに存在している必要はありません。ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

Attention:
JIT プロビジョニングが有効な場合、認証タブでプロビジョニング解除されたユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングに加えて、SCIM（System for Cross-domain Identity Management）プロビジョニング、つまりユーザープロビジョニングによって作成されたユーザーの継続的なユーザーアカウント管理を有効化および設定することも可能です。SCIM プロビジョニングには、Zabbix への認証のために Zabbix のAPI トークン（スーパ管理者権限付き）が必要です。

たとえば、ユーザーがある SAML グループから別のグループに移動された場合、Zabbix でもユーザーは一方のグループからもう一方のグループに移動されます。ユーザーが SAML グループから削除された場合、Zabbix でもそのグループから削除され、他のグループに属していない場合はプロビジョニング解除されたユーザー用のユーザーグループに追加されます。

SCIM が有効かつ設定されている場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニングされ、その後 SAML の変更に基づいて継続的に更新されます。すでに存在する SAML ユーザーはプロビジョニングされず、プロビジョニングされたユーザーのみが更新されます。ユーザーがプロビジョニングまたは更新される際、有効なメディアのみがユーザーに追加されることに注意してください。

SCIM が有効でない場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニング（および後で更新）されます。

Note:
SAML 認証が有効な場合、ユーザーはローカルログインまたは SAML シングルサインオンのいずれかを選択できます。JIT プロビジョニングを使用する場合、シングルサインオンのみが可能です。

アイデンティティプロバイダーの設定

Zabbix で動作させるためには、SAML アイデンティティプロバイダー (onelogin.com、auth0.com、okta.comなど) を以下のように設定する必要があります。

- *Assertion Consumer URL* には `<path_to_zabbix_ui>/index_sso.php?acs` を設定します
- *Single Logout URL* には `<path_to_zabbix_ui>/index_sso.php?sls` を設定します

`<path_to_zabbix_ui>` の例: `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_public_`

Zabbix のセットアップ

Attention:
フロントエンドで SAML 認証を使用する場合は、php-openssl のインストールが必要です。

SAML 認証を使用するには、Zabbix を以下のように設定する必要があります。

1. SP の秘密鍵と証明書を用意します。SAML をフロントエンドで有効にするには、IdP 証明書を指定する必要があります（貼り付けまたはファイル選択）。署名または暗号化オプションを選択した場合は、SP 証明書と SP 秘密鍵を指定する必要があります。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'database'`が設定されている場合、SAML 設定時にフロントエンドで証明書テキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルは不要です。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'file'`が設定されている場合、証明書はファイルシステム上（デフォルトでは ui/conf/certs または zabbix.conf.php で設定したパス）に配置する必要があり、フロントエンドは証明書をデータベースに保存しません。`$SSO['CERT_STORAGE']` が設定されていない、またはコメントアウトされている場合は、ファイル保存が前提となります。

デフォルトでは、Zabbix は以下の場所を参照します。

- ui/conf/certs/sp.key - SP 秘密鍵ファイル
- ui/conf/certs/sp.crt - SP 証明書ファイル
- ui/conf/certs/idp.crt - IDP 証明書ファイル

Attention:
証明書情報は、フロントエンドでテキストを貼り付けるかファイルをアップロードしてインポートできます。一部の環境では利便性のために暗号化されていないインポートも許可されていますが、暗号化された通信の使用を強く推奨します（例：Zabbix フロントエンドで HTTPS/TLS を有効化）。SAML 証明書/鍵をデータベースに保存する場合は、アプリケーションとデータベース間の通信を保護するためにデータベース接続で TLS/SSL を有効化してください。
 SAML 証明書や秘密鍵を含む可能性のあるデータベースバックアップは暗号化し、バックアップファイルや SAML 情報を保存するデータベーステーブルへのアクセスを制限してください。データベースユーザーの権限は必要最小限に制限してください。

2. 主要な設定はすべて Zabbix フロントエンドで設定できます。ただし、設定ファイルで追加設定を指定することも可能です。

Zabbix フロントエンドで設定可能なパラメータ：

パラメータ	説明
SAML 認証を有効	チェックボックスをオンにすると SAML 認証が有効になります。
JIT プロビジョニングを有効	チェックボックスをオンにすると JIT ユーザープロビジョニングが有効になります。
IDP エンティティ *ID*	SAML アイデンティティプロバイダー内で一意のエンティティ識別子。
SSO サービス *URL*	ログイン時にリダイレクトされる URL。
SLO サービス *URL*	ログアウト時にリダイレクトされる URL。空欄の場合、SLO サービスは使用されません。

パラメータ	説明
ユーザー名属性	Zabbix ログイン時にユーザー名として使用する SAML 属性。 サポートされる値のリストはアイデンティティプロバイダーによって決まります。 例： uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
SP エンティティ *ID*	サービスプロバイダーの一意の識別子（不一致の場合は操作が拒否されます）。 URL または任意の文字列を指定できます。
SP name ID フォーマット	レスポンスで特定の name ID フォーマットを要求します。 例： urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified urn:oasis:names:tc:SAML:2.0:nameid-format:transient
IdP 証明書	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）証明書の内容。
SP 秘密鍵	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）秘密鍵の内容。アイデンティティプロバイダー（IdP）との安全な認証とデータ交換を提供します。
SP 証明書	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）証明書の内容。
署名	SAML 署名を有効にするエンティティを選択するためにチェックボックスをオンにします： メッセージ アサーション *AuthN* リクエスト ログアウトリクエスト ログアウトレスポンス
暗号化	SAML 暗号化を有効にするエンティティを選択するためにチェックボックスをオンにします： *Name ID* アサーション
大文字小文字を区別したログイン	ユーザー名の大文字小文字を区別したログインを無効にするにはチェックを外します（デフォルトは有効）。 大文字小文字を区別しない場合、たとえば Zabbix ユーザーが「Admin」や「ADMIN」でも「admin」でログインできます。 ただし、大文字小文字を区別しない設定で同じようなユーザー名（例：Admin と admin）が複数存在する場合、これらのユーザーのログインは常に「認証に失敗しました：指定された認証情報が一意ではありません。」というエラーで拒否されます。
JIT プロビジョニングを設定	このチェックボックスをオンにすると、JIT ユーザープロビジョニングに関連するオプションが表示されます。
グループ名属性	JIT ユーザープロビジョニング用のグループ名属性を指定します。
ユーザー名属性	JIT ユーザープロビジョニング用のユーザー名属性を指定します。
ユーザー姓属性	JIT ユーザープロビジョニング用のユーザー姓属性を指定します。
ユーザーグループマッピング	SAML ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 プロビジョニングされたユーザーが Zabbix でどのユーザーグループ/ロールを取得するかを決定するために必要です。 追加をクリックしてマッピングを追加します。 *SAML* グループパターンフィールドはワイルドカードをサポートします。グループ名は既存のグループと一致する必要があります。 SAML ユーザーが複数の Zabbix ユーザーグループに一致する場合、ユーザーはすべてのグループのメンバーになります。 複数の Zabbix ユーザーロールに一致する場合、ユーザーはその中で最も高い権限レベルを取得します。
メディアタイプマッピング	ユーザーの SAML メディア属性（例：メール）を Zabbix ユーザーメディアにマッピングし、通知送信に使用します。
SCIM プロビジョニングを有効	このチェックボックスをオンにすると SCIM 2.0 プロビジョニングが有効になります。

Zabbix へのサインインやユーザープロビジョニング用の SAML アイデンティティプロバイダー設定例：

- Microsoft Azure AD
- Okta
- Onelogin

SCIM プロビジョニングに関する注意事項

SCIM プロビジョニングの場合、Zabbix フロントエンドへのパスを指定し、api_scim.php を追加して、アイデンティティプロバイダー側で指定します。例:

`https://<path-to-zabbix-ui>/api_scim.php`

Zabbix で使用されるユーザー属性（ユーザー名、名、姓、メディア属性）はカスタム属性として追加する必要があり、必要に応じて外部名前空間はユーザースキーマと同じである必要があります: `urn:ietf:params:scim:schemas:core:2.0:User`。

詳細設定

追加の SAML パラメータは、Zabbix フロントエンドの設定ファイル（*zabbix.conf.php*）で設定できます。

- $SSO['SP_KEY'] = '<SP 秘密鍵ファイルへのパス >';
- $SSO['SP_CERT'] = '<SP 証明書ファイルへのパス >';
- $SSO['IDP_CERT'] = '<IDP 証明書ファイルへのパス >';
- $SSO['SETTINGS']

Note:
Zabbix はOneLogin の SAML PHP Toolkitライブラリ（バージョン 3.4.1）を使用しています。$SSO['SETTINGS'] セクションの構造は、ライブラリで使用されている構造と同様である必要があります。設定オプションの説明については、公式ライブラリのドキュメントを参照してください。

$SSO['SETTINGS'] の一部として設定できるオプションは以下の通りです。

- *strict*
- *baseurl*
- *compress*
- *contactPerson*
- *organization*
- *sp*（このリストで指定されたオプションのみ）
  - *attributeConsumingService*
  - *x509certNew*
- *idp*（このリストで指定されたオプションのみ）
  - *singleLogoutService*（1 つのオプションのみ）
    - *responseUrl*
  - *certFingerprint*
  - *certFingerprintAlgorithm*
  - *x509certMulti*
- *security*（このリストで指定されたオプションのみ）
  - *signMetadata*
  - *wantNameId*
  - *requestedAuthnContext*
  - *requestedAuthnContextComparison*
  - *wantXMLValidation*
  - *relaxDestinationValidation*
  - *destinationStrictlyMatches*
  - *rejectUnsolicitedResponsesWithInResponseTo*
  - *signatureAlgorithm*
  - *digestAlgorithm*
  - *lowercaseUrlencoding*

その他のすべてのオプションはデータベースから取得され、上書きできません。*debug* オプションは無視されます。

さらに、Zabbix UI がプロキシやロードバランサの背後にある場合は、カスタムの *use_proxy_headers* オプションを使用できます。

- *false*（デフォルト）- オプションを無視する
- *true* - base URL の構築に X-Forwarded-* HTTP ヘッダーを使用する

ロードバランサを使用して Zabbix インスタンスに接続し、ロードバランサが TLS/SSL を使用し Zabbix が使用しない場合は、'baseurl'、'strict'、'use_proxy_headers' パラメータを次のように指定する必要があります。

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

設定例:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Kerberos/ADFS によるフロントエンドの設定

Zabbix フロントエンドの設定ファイル（*zabbix.conf.php*）を使用して、Kerberos 認証および ADFS による SSO を設定できます。

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

この場合、SAML 設定の *SP name ID* フィールドには以下を設定します。

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

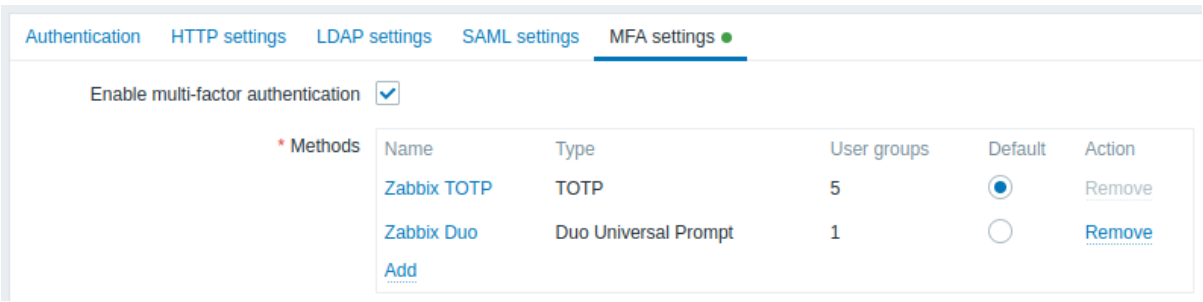
概要

Zabbix へのサインインには、多要素認証（MFA）を使用でき、ユーザー名とパスワードだけでなく、追加のセキュリティ層を提供します。

MFA を使用する場合、ユーザーは Zabbix に存在し、ログイン時に Zabbix の認証情報を提供し、さらに通常はユーザーの携帯電話の認証アプリによって生成されたコードなど、他の手段で本人確認を行う必要があります。

複数の MFA 方式が利用可能であり、ユーザーは自分のセキュリティ要件や好みに最適なオプションを選択できます。これらの方式には、時刻ベースのワンタイムパスワード（TOTP）と Duo Universal Prompt があります。

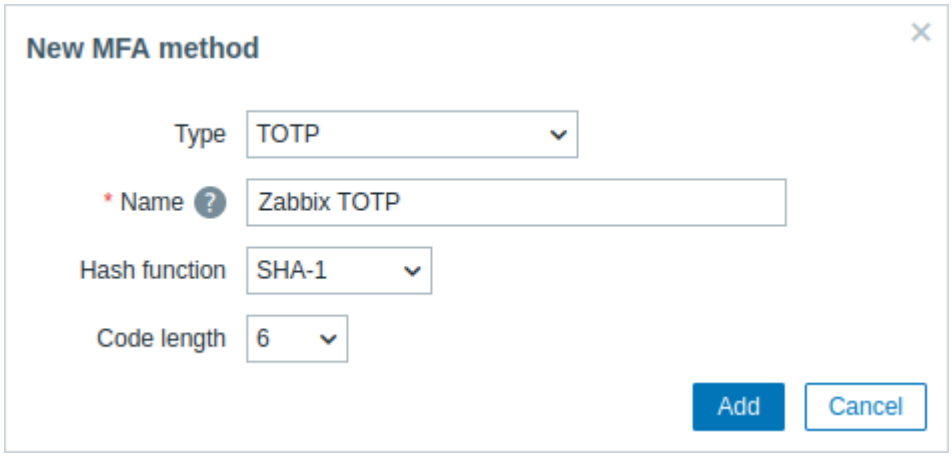
設定



設定パラメータ:

パラメータ	説明
多要素認証を有効にする	チェックボックスをオンにすると多要素認証が有効になります。
メソッド	追加をクリックして MFA メソッドを設定します（下記のメソッド設定を参照）。

メソッドの設定



メソッド設定パラメータ:

パラメータ	説明
タイプ	MFA メソッドのタイプを選択: **TOTP** - 認証アプリを使用して時限ワンタイムパスワードを生成します。 **Duo Universal Prompt** - Duo認証サービスを使用して多要素認証を提供します。
名前	認証アプリで全ての MFA ユーザーにアカウント名として表示される名前を入力します（例：”Zabbix”）。
ハッシュ関数	TOTP コードを生成するためのハッシュ関数（SHA-1、SHA-256、または SHA-512）を選択します。 このパラメータは、MFA メソッドタイプが「TOTP」に設定されている場合に利用できます。 SHA-256 または SHA-512 を選択すると、多くのアプリケーションがこれらの関数をサポートしていないため、互換性が大幅に制限される可能性があることに注意してください。
コード長	認証コードの長さ（6 または 8）を選択します。 このパラメータは、MFA メソッドタイプが「TOTP」に設定されている場合に利用できます。
API ホスト名	Duo 認証サービスから提供された API ホスト名を入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。
クライアント *ID*	Duo 認証サービスから提供されたクライアント ID を入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。
クライアントシークレット	Duo 認証サービスから提供されたクライアントシークレットを入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。

設定例

このセクションでは、Time-Based One-Time Password (TOTP) およびDuo Universal Prompt を使用した MFA の設定例を示します。

TOTP

TOTP の場合、ユーザーは認証アプリ（例：Google Authenticatorアプリ）を使用して本人確認を行う必要があります。

1. Zabbix の *MFA* 設定（ユーザー → 認証）に移動し、多要素認証を有効にします。

2. 次の設定で新しい MFAメソッドを追加します。

- タイプ: TOTP
- 名前: Zabbix TOTP
- ハッシュ関数: SHA-1
- コード長: 6

3. 追加ボタンをクリックし、続いて更新ボタンをクリックします。

4. ユーザー → ユーザーグループに移動し、次の設定で新しいユーザーグループを作成します。

- グループ名: TOTP group
- ユーザー: Admin
- 多要素認証: デフォルト（またはデフォルトに設定されていない場合は「Zabbix TOTP」）

5. Zabbix からログアウトし、再度認証情報を使用してログインします。ログインに成功すると、MFA への登録を求められ、QR コードとシークレットキーが表示されます。

6. QR コードをスキャンするか、シークレットキーを Google Authenticator アプリに入力します。アプリが認証コードを生成するので、それを入力してログインプロセスを完了します。

7. 次回以降のログイン時には、Google Authenticator アプリから認証コードを取得し、ログイン時に入力します。

Duo Universal Prompt

Duo Universal Prompt では、ユーザーはDuo Mobile認証アプリを使用して本人確認を行う必要があります。

> **Attention:**
> Duo Universal Prompt の MFA 方式を利用するには、php-curl拡張モジュールのインストール、HTTPS 経由での Zabbix へのアクセス、および Duo サーバーへの外部接続の許可が必要です。さらに、Web サーバーで Content Security Policy (CSP) を有効にしている場合は、仮想ホストの設定ファイルの CSP ディレクティブに”duo.com” を追加してください。

1. Duo Signupで無料の Duo 管理者アカウントにサインアップします。

2. Duo 管理パネルを開き、*Applications* → *Protect an Application*に移動し、*Web SDK* アプリケーションを検索して *Protect* をクリックします。

3. Zabbix で MFA 方式を設定するために必要な認証情報（Client ID、Client secret、API hostname）をメモします。

4. Zabbix のユーザー → 認証の MFA 設定に移動し、多要素認証を有効にします。

5. 以下の設定で新しい MFA方式を追加します。

- 種類: Duo Universal Prompt
- 名前: Zabbix Duo
- API ホスト名: (Duo の API ホスト名を使用)
- クライアント ID: (Duo の Client ID を使用)
- クライアントシークレット: (Duo の Client secret を使用)

6. 追加ボタンをクリックし、続けて更新ボタンをクリックします。

7. ユーザー → ユーザーグループに移動し、以下の設定で新しいユーザーグループを作成します。

- グループ名: Duo group
- ユーザー: Admin
- 多要素認証: デフォルト（またはデフォルトに設定されていない場合は”Zabbix Duo”）

8. Zabbix からログアウトし、認証情報を使用して再度ログインします。ログインに成功すると、MFA への登録を求められ、Duo にリダイレクトされます。Duo のセットアップを完了し、電話の Duo アプリでユーザーを認証してログインします。

9. 以降のログインでは、Duo アプリで提供される適切な MFA 方式（認証コードの取得、プッシュ通知への応答、ハードキーの使用など）を使用し、ログイン時に必要な情報を入力します。

9 管理

概要

管理メニューは、Zabbix の管理機能のためのものです。このメニューは、SuperAdminユーザータイプのユーザーのみが利用できます。

1 一般設定

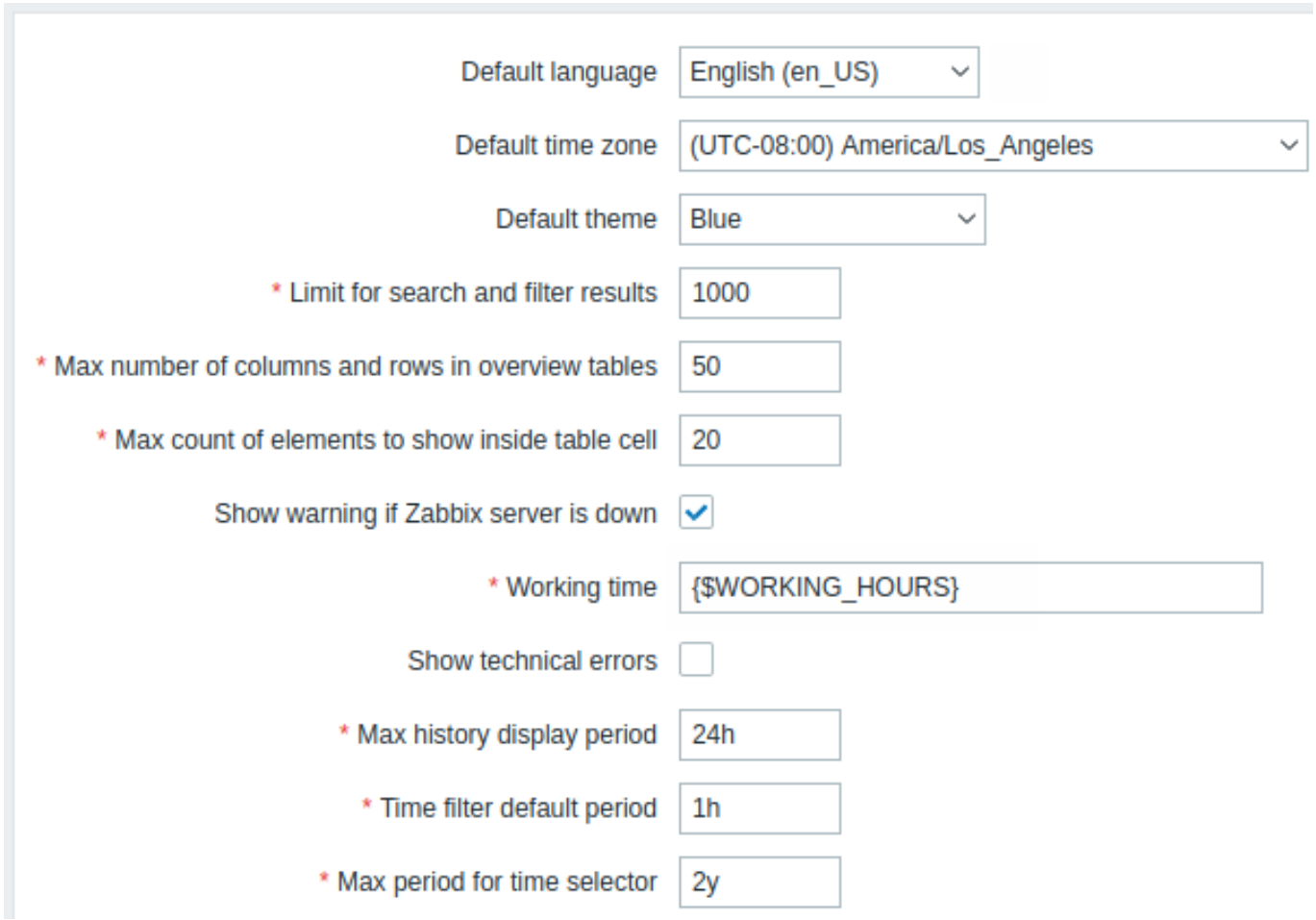
概要

管理 → 一般セクションには、フロントエンド関連のデフォルト設定や Zabbix のカスタマイズのためのサブセクションがいくつか含まれています。

利用可能なサブセクションのリストは、管理メニューセクションの 一般を押すと表示されます。また、左上隅のタイトルドロップダウンを使用してサブセクション間を切り替えることもできます。

GUI

このセクションでは、フロントエンドに関連するいくつかのデフォルト値のカスタマイズを行います。



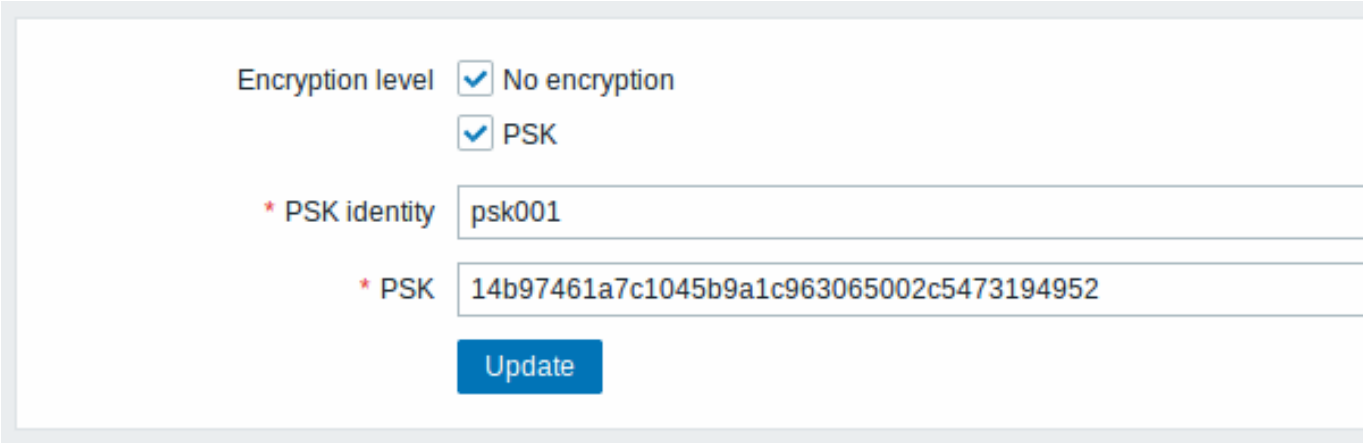
設定パラメータ:

パラメータ	説明
デフォルト言語	ユーザープロファイルで言語を指定していないユーザーおよびゲストユーザーのデフォルト言語。 詳細は追加フロントエンド言語のインストールを参照してください。
デフォルトタイムゾーン	ユーザープロファイルでタイムゾーンを指定していないユーザーおよびゲストユーザーのデフォルトタイムゾーン。
デフォルトテーマ	ユーザープロファイルでテーマを指定していないユーザーおよびゲストユーザーのデフォルトテーマ。
検索およびフィルター結果の上限	Web インターフェースのリスト（例: データ収集 > ホスト）で表示される要素（行）の最大数。 注意: 例えば「50」に設定した場合、影響を受けるすべてのフロントエンドリストで最初の 50 要素のみが表示されます。リストに 50 を超える要素が含まれている場合、「**50+** 件中 1 から 50 件を表示」のように「+」記号で示されます。また、フィルタリングを使用しても 50 件を超える一致がある場合、最初の 50 件のみが表示されます。 このパラメータの値を増やすと、Web サーバー側のパフォーマンス低下やメモリ消費の増加につながる場合があります。
概要テーブルの最大列数および行数	トリガー概要ダッシュボードウィジェットで表示する最大の列数および行数。同じ制限が列と行の両方に適用されます。表示されているより多くの行や列が存在する場合、テーブルの下部に「すべての結果が表示されていません。より具体的な検索条件を指定してください。」という警告が表示されます。
テーブルセル内に表示する要素の最大数	1 つのテーブルセルに表示されるエントリについて、ここで設定した数を超えて表示されません。
Zabbix サーバーがダウンしている場合に警告を表示	Zabbix サーバーに接続できない（ダウンしている可能性がある）場合、ブラウザウィンドウに警告メッセージを表示するパラメータです。メッセージは、ユーザーがページをスクロールしても表示され続けます。マウスオーバーすると、下の内容を表示するために一時的にメッセージが非表示になります。
稼働時間	このシステム全体のパラメータは、稼働時間を定義します。グラフでは、稼働時間が白い背景、非稼働時間がグレーで表示されます。 時間のフォーマットについては時間期間の指定ページを参照してください。 ユーザーマクロがサポートされています。

パラメータ	説明
技術的なエラーを表示	チェックを入れると、すべての登録ユーザーが技術的なエラー（PHP/SQL）を確認できます。チェックを外すと、情報はZabbix スーパ管理者およびデバッグモードが有効なユーザーグループに属するユーザーのみが利用できます。
履歴表示期間の最大値	監視データ > 最新データ、監視データ > ホストのホストWeb シナリオ情報、およびトップアイテムダッシュボードウィジェットで履歴データを表示する最大期間。 許可範囲: 24 時間（デフォルト）~1 週間。時間のサフィックス（例: 1w（1 週間）、36h（36 時間））がサポートされています。
時間フィルターのデフォルト期間	グラフやダッシュボードでデフォルトで使用される時間期間。許可範囲: 1 分~10 年（デフォルト: 1 時間）。 時間のサフィックス（例: 10m（10 分）、5w（5 週間））がサポートされています。 注意: ユーザーがグラフを表示中に期間を変更した場合、この期間はユーザーの設定として保存され、グローバルデフォルトや以前のユーザー選択を置き換えます。
時間セレクターの最大期間	グラフやダッシュボードで利用可能な最大期間。ユーザーはこれより古いデータを表示できません。 許可範囲: 1 年~10 年（デフォルト: 2 年）。 時間のサフィックス（例: 1y（1 年）、365w（365 週間））がサポートされています。

自動登録

このセクションでは、アクティブエージェントの自動登録の暗号化レベルを設定できます。



アスタリスクが付いているパラメータは必須です。

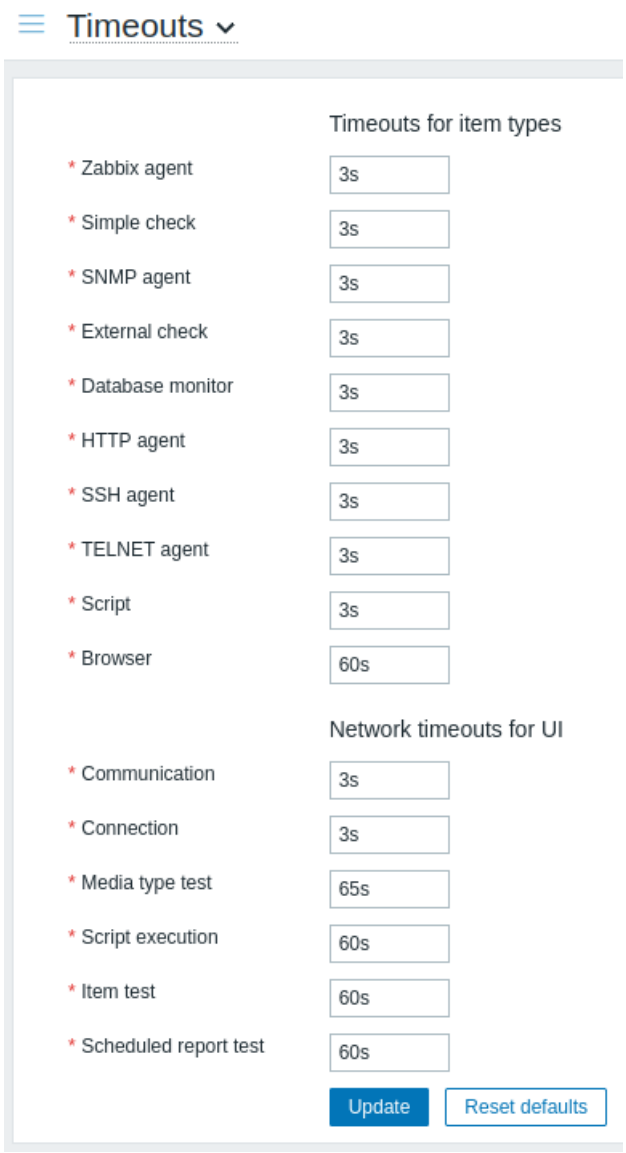
設定パラメータ:

パラメータ	説明
暗号化レベル	暗号化レベルのオプションを 1 つまたは両方選択します: 暗号化なし - 暗号化されていない接続を許可する **PSK** - 事前共有鍵による TLS 暗号化接続を許可する
PSK 識別子	事前共有鍵の識別子文字列を入力します。 このフィールドは、暗号化レベルで「PSK」を選択した場合のみ利用できます。 PSK 識別子には機密情報を入力しないでください。ネットワーク上で暗号化されずに送信され、受信者にどの PSK を使用するかを通知します。
PSK	事前共有鍵（16 進数の偶数桁）を入力します。 Zabbix が GnuTLS または OpenSSL ライブラリを使用している場合の最大長: 512 桁の 16 進数（256 バイトの PSK）、mbed TLS（PolarSSL）ライブラリを使用している場合の最大長: 64 桁の 16 進数（32 バイトの PSK）。 例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 このフィールドは、暗号化レベルで「PSK」を選択した場合のみ利用できます。

関連項目: 安全な自動登録

タイムアウト

このセクションでは、グローバルなアイテムタイプのタイムアウトとネットワークタイムアウトを設定できます。このフォームのすべてのフィールドは必須です。

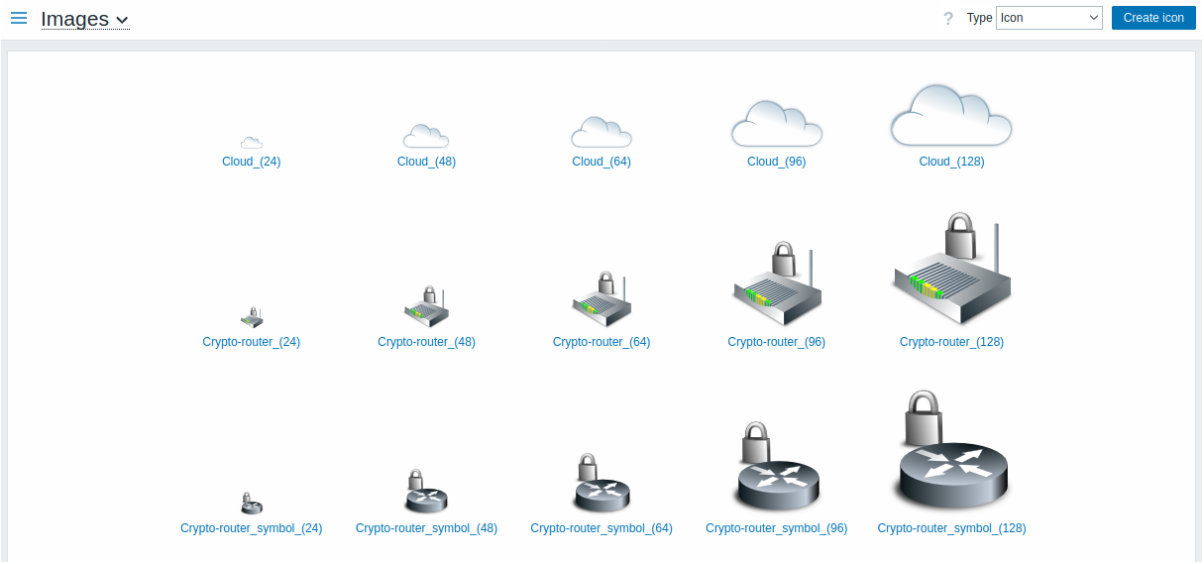


パラメータ		説明
アイテムタイプのタイムアウト		Zabbix がチェックを中止するまでに待機する秒数（タイプに基づく）。 追加の詳細と制限については、アイテムのタイムアウトを参照してください。 許可される範囲: 1 - 600 秒（デフォルト: 3 秒; ブラウザアイテムのデフォルト: 60 秒）。 時間のサフィックス（例: 30s, 1m）やユーザーマクロがサポートされています。 プロキシレベルでタイムアウトが設定されている場合は、これらのグローバル設定よりも優先されることに注意してください。さらに、特定のアイテムにタイムアウトが設定されている場合は、グローバルおよびプロキシの両方の設定よりも優先されます。
UI のネットワークタイムアウト		
	通信	アイドル状態のソケットを閉じるまでに待機する秒数（Zabbix サーバーへの接続が以前に確立されているが、フロントエンドがこの時間内にデータの読み取り/送信操作を完了できない場合、接続は切断されます）。許可される範囲: 1 - 300 秒（デフォルト: 3 秒）。
	接続	Zabbix サーバーへの接続試行を停止するまでに待機する秒数。許可される範囲: 1 - 30 秒（デフォルト: 3 秒）。
	メディアタイプテスト	メディアタイプをテストする際に応答を待機する秒数。許可される範囲: 1 - 300 秒（デフォルト: 65 秒）。

パラメータ		説明
	スクリプト実行	スクリプトを実行する際に応答を待機する秒数。許可される範囲: 1 - 300 秒（デフォルト: 60 秒）。 このタイムアウトは、さまざまな長さのスクリプト実行チェーン全体に適用されます。たとえば、スクリプトがエージェントで実行される場合、サーバー（場合によってはプロキシも含む）からエージェントへの往復となります。 エージェントで実行されるスクリプトには、サーバーまたはプロキシのタイムアウトが適用されます。 アクティブエージェントのみで実行されるスクリプトの場合、デフォルトのサーバー/プロキシのタイムアウトを引き上げる必要がある場合があります。サーバー/プロキシのタイムアウトは、アクティブチェックの更新頻度よりも高く設定する必要があります。そうしないと、アクティブエージェントがスクリプトを受信する前にサーバー/プロキシのタイムアウトが超過する可能性があります。 関連項目: スクリプトのタイムアウト
	アイテムテスト	アイテムをテストする際に返されたデータを待機する秒数。許可される範囲: 1 - 600 秒（デフォルト: 60 秒）。
	スケジュールレポートテスト	スケジュールレポートをテストする際に返されたデータを待機する秒数。許可される範囲: 1 - 300 秒（デフォルト: 60 秒）。

画像

画像セクションには、Zabbix で利用可能なすべての画像が表示されます。画像はデータベースに保存されます。

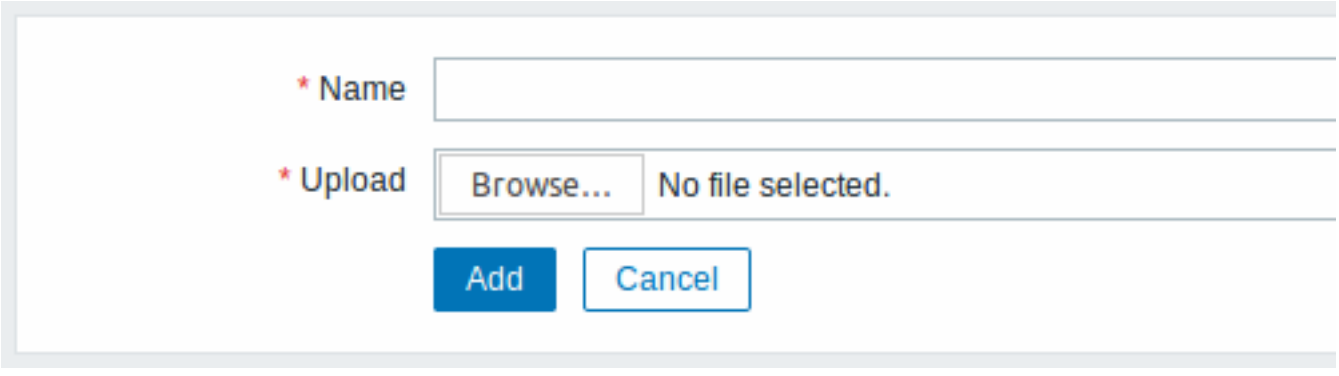


タイプドロップダウンで、アイコン画像と背景画像を切り替えることができます。

- アイコンはネットワークマップの要素の表示に使用されます
- 背景はネットワークマップの背景画像として使用されます

画像の追加

右上の アイコンの作成または 背景の作成ボタンをクリックして、独自の画像を追加できます。



画像の属性:

パラメータ	説明
名前	画像の一意の名前。
アップロード	Zabbix にアップロードするローカルシステムからファイル（PNG、JPEG、GIF）を選択します。 注：アップロード時に PNG に変換される他の形式もアップロードできる場合があります。画像処理には GD ライブラリが使用されるため、サポートされる形式は使用されるライブラリのバージョンに依存します（Zabbix では 2.0.28 以上が必要です）。

Note:
アップロードファイルの最大サイズは ZBX_MAX_IMAGE_SIZE の値で制限されており、1024x1024 バイトまたは 1MB です。

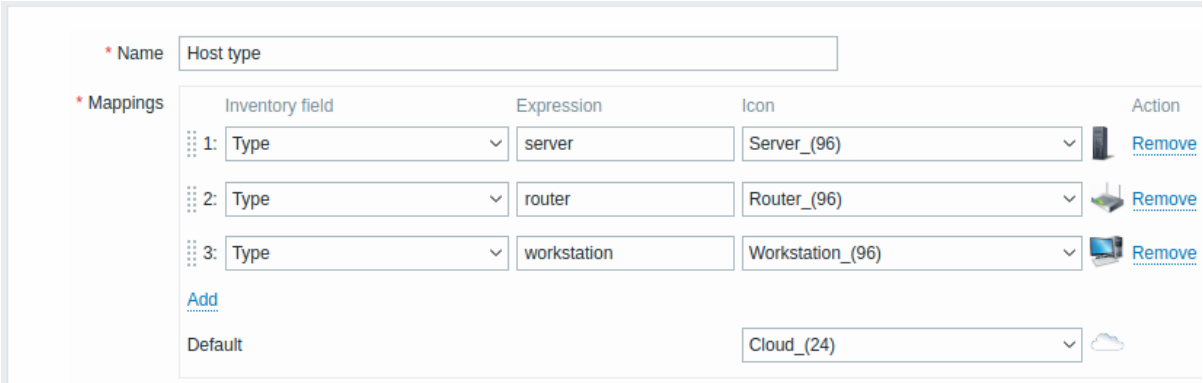
画像サイズが 1MB に近く、MySQL の `max_allowed_packet` 設定パラメータがデフォルトの 1MB の場合、画像のアップロードに失敗することがあります。この場合は、max_allowed_packet パラメータを増やしてください。

アイコンマッピング

このセクションでは、特定のホストと特定のアイコンのマッピングを作成できます。ホストインベントリフィールドの情報を使用してマッピングを作成します。

作成したマッピングは、ネットワークマップの設定で使用でき、一致するホストに自動的に適切なアイコンを割り当てることができます。

新しいアイコンマップを作成するには、右上隅のアイコンマップの作成をクリックします。



設定パラメータ:

パラメータ	説明
名前	アイコンマップの一意の名前。
マッピング	マッピングのリスト。マッピングの順序は、どれが優先されるかを決定します。ドラッグ＆ドロップでマッピングの
インベントリフィールド	一致を探すために参照するホストインベントリフィールド。
式	一致を記述する正規表現。
アイコン	式に一致した場合に使用するアイコン。
デフォルト	デフォルトで使用するアイコン。

正規表現

このセクションでは、フロントエンドのさまざまな場所で使用できるカスタム正規表現を作成できます。詳細は正規表現セクションを参照してください。

トリガー表示オプション

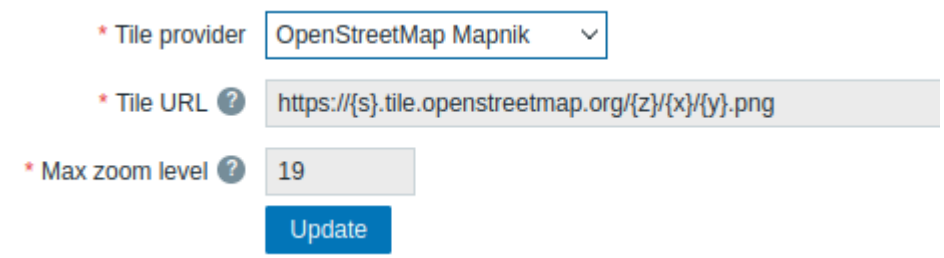
このセクションでは、フロントエンドでのトリガーステータスの表示方法やトリガーの深刻度の名前や色をカスタマイズできます。



パラメータ	説明
カスタムイベントステータス色を使用	このパラメータをチェックすると、未確認/確認済みの問題の色のカスタマイズが有効になります。
未確認の問題イベント、 確認済みの問題イベント、 未確認の復旧イベント、 確認済みの復旧イベント	新しいカラーコードを入力するか、色をクリックしてパレットから新しい色を選択します。 点滅チェックボックスをマークすると、ステータス変更時にトリガーが一定時間点滅してより目立つようになります。
OK トリガーの表示期間	OK トリガーを表示する期間。許容範囲: 0～24 時間。時間のサフィックス（例: 5m、2h、1d）がサポートされています。
ステータス変更時のトリガー点滅時間	トリガーの点滅時間。許容範囲: 0～24 時間。時間のサフィックス（例: 5m、2h、1d）がサポートされています。
未分類、 情報、 警告、 軽度障害、 重度障害、 致命的障害	システムデフォルトの代わりに表示するカスタムの深刻度名や色。 新しいカラーコードを入力するか、色をクリックしてパレットから新しい色を選択します。 ここで入力したカスタムの深刻度名は、すべてのロケールで使用されることに注意してください。特定のユーザー向けに他の言語に翻訳する必要がある場合は、トリガーの深刻度のカスタマイズページを参照してください。

地理マップ

このセクションでは、地理マップタイルサービスプロバイダーの選択と、Geomap ダッシュボードウィジェット用のサービスプロバイダー設定の構成ができます。地理マップによる可視化を提供するために、Zabbix はオープンソースの JavaScript インタラクティブマップライブラリ Leaflet を使用しています。Zabbix は、定義済みのタイルプロバイダーを含むサードパーティのタイルプロバイダーによって提供される画像の品質を管理していないことに注意してください。



パラメータ	説明
タイルプロバイダー	利用可能なタイルサービスプロバイダーのいずれかを選択するか、その他を選択して別のタイルプロバイダーまたはセルフホストのタイルを追加します（カスタムタイルサービスプロバイダーの使用を参照）。
タイル URL	地理マップ上でタイルレイヤーを読み込み表示するための URL テンプレート（最大 2048 文字）。 このフィールドは、タイルプロバイダーがその他に設定されている場合のみ編集可能です。 以下のプレースホルダーがサポートされています： *{s}* は利用可能なサブドメインの 1 つを表します； *{z}* は URL 内のズームレベルパラメータを表します； *{x}* および *{y}* はタイル座標を表します； *{r}* は URL に”@2x” を追加して Retina タイルを読み込むために使用できます。 例: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
帰属テキスト	マップ上の小さなテキストボックスに表示されるタイルプロバイダーの帰属テキスト。このフィールドはタイルプロバイダーがその他に設定されている場合のみ表示されます。
最大ズームレベル	マップの最大ズームレベル。このフィールドはタイルプロバイダーがその他に設定されている場合のみ編集可能です。

カスタムタイルサービスプロバイダーの使用

Geomap ウィジェットは、カスタムのセルフホストまたはサードパーティのタイルプロバイダーサービスからラスタータイル画像を読み込むことができます。カスタムサードパーティタイルプロバイダーサービスまたはセルフホストタイルフォルダーまたはサーバーを使用するにはタイルプロバイダーフィールドでその他を選択し、適切なプレースホルダーを使用してタイル *URL* フィールドでカスタム URL を指定します。

モジュール

このセクションでは、カスタムおよび組み込みのフロントエンドモジュールを管理できます。

ディレクトリのスキャンをクリックすると、カスタムモジュールの登録/登録解除が行えます。登録されたモジュールはリストに表示され、登録解除されたモジュールはリストから削除されます。

リスト内のモジュールのステータスをクリックすると、モジュールの有効化/無効化ができます。また、リスト内でモジュールを選択し、リスト下部の有効化/無効化ボタンをクリックすることで、複数のモジュールを一括で有効化/無効化できます。

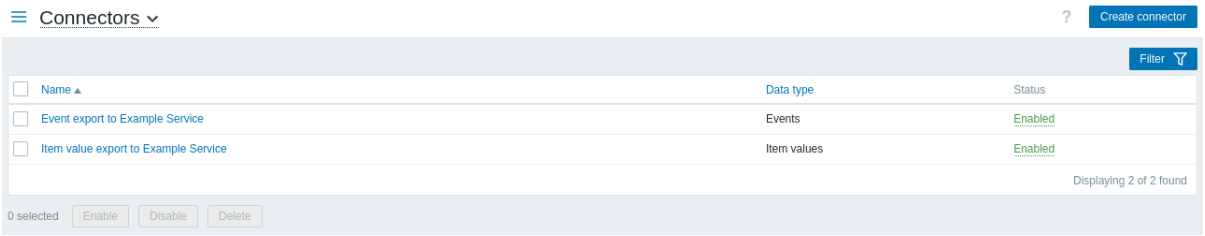
リスト内のモジュール名をクリックすると、その詳細をポップアップウィンドウで表示できます。

モジュールの詳細ポップアップウィンドウでも、モジュールのステータスを更新できます。これを行うには、有効チェックボックスをオン/オフにし、更新をクリックします。

モジュール名またはステータス（有効/無効）でフィルタリングできます。

コネクタ

このセクションでは、Zabbix データを HTTP 経由で外部システムにストリーミングするためのコネクタを設定できます。



コネクタの作成をクリックして、新しいコネクタを設定します。

コネクタは名前またはステータス（有効/無効）でフィルタリングできます。リスト内のコネクタのステータスをクリックすると、コネクタを有効/無効にできます。また、リストでコネクタを選択し、リスト下部の有効/無効ボタンをクリックすることで、コネクタを一括で有効/無効にすることもできます。

その他

このセクションでは、その他のフロントエンドパラメータを設定できます。



パラメータ	説明
フロントエンド *URL*	Zabbix Web インターフェースへの URL（最大 2048 文字）。 このパラメータは、Zabbix Web サービスがフロントエンドと通信するために使用され、スケジュールレポートを有効にするには指定する必要があります。
検出されたホストのグループ	ネットワーク検出およびエージェントの自動登録で検出されたホストは、自動的にここで選択したホストグループに配置されます。
デフォルトのホストインベントリモード	ホストインベントリのデフォルトモード。新しいホストまたはホストプロトタイプがサーバーまたはフロントエンドによって作成されるたびに、ホストインベントリモードの設定操作でホスト検出/自動登録時に上書きされない限り、このモードが適用されます。

パラメータ	説明
データベースダウンメッセージのユーザーグループ	アラームメッセージを送信するユーザーグループ、または「なし」。 Zabbix サーバーはバックエンドデータベースの可用性に依存しています。データベースがなければ動作できません。データベースがダウンした場合、選択したユーザーに Zabbix から通知できます。通知は、ここで設定したユーザーグループの有効なユーザーメディアエントリを使用して送信されます。通知は、以下のメディアタイプ（有効な場合）で送信されます：メール、SMS、カスタムアラートスクリプト。Webhook メディアエントリが設定され有効になっていても、通知はトリガーされません。 Zabbix サーバーは停止せず、データベースが復旧するまで待機し、処理を継続します。 通知の内容は以下の通りです： `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <DBMS（データベース）の種類に応じたエラーメッセージ >` <DB Host> が空の場合はメッセージに追加されず、<DB Port> がデフォルト値（”0”）の場合も追加されません。アラートマネージャ（Zabbix サーバーの特別なプロセス）は、10 秒ごとにデータベースへの新しい接続を確立しようとします。データベースがまだダウンしている場合、アラートマネージャはアラートの送信を繰り返しますが、15 分に 1 回より頻繁には送信しません。
一致しない *SNMP* トラップを記録	対応する SNMP インターフェースが見つからない場合、SNMP トラップを記録します。

認証

パラメータ	説明
ログイン試行回数	ログインがブロックされるまでのログイン失敗回数。
ログインブロック間隔	ログイン試行回数の制限を超えた場合にログインが禁止される時間。許容範囲: 0 - 3600 秒。時間のサフィックス（例: 90s, 5m, 1h）がサポートされています。

シークレットの保存

パラメータ	説明
Vault プロバイダー	ユーザーマクロの値を保存するためのシークレット管理ソフトウェアを選択します - *HashiCorp Vault*（デフォルト）または *CyberArk Vault*。
シークレット *Vault* マクロの解決方法	シークレット Vault マクロの解決方法： **Zabbix** サーバー - シークレットは Zabbix サーバーによって取得され、必要に応じてプロキシに転送されます（デフォルト）； **Zabbix** サーバーとプロキシ - シークレットは Zabbix サーバーとプロキシの両方によって取得され、マクロを独立して解決できるようになります。

関連項目：シークレットの保存。

セキュリティ

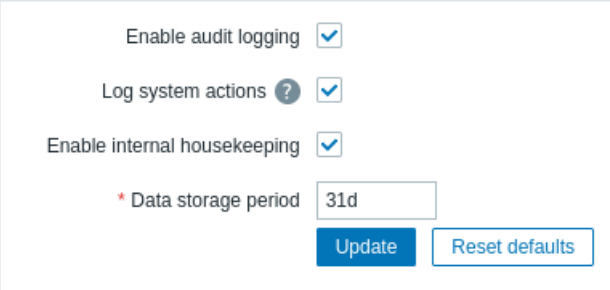
パラメータ	説明
URI スキームの検証	このチェックボックスを外すと、URI スキームの検証が無効になります (デフォルトは有効)。 チェックを入れると、許可する URI スキームのカンマ区切りリストを指定できます (デフォルト: http,https,ftp,file,mailto,tel,ssh)。URI が使用されるフロントエンドのすべてのフィールド (例: マップ要素の URL) に適用されます。
X-Frame-Options HTTP ヘッダーの使用	このチェックボックスを外すと、HTTP X-Frame-options ヘッダーが無効になります (非推奨)。 チェックを入れると、HTTP X-Frame-options ヘッダーの値を指定できます。サポートされる値: **SAMEORIGIN**(デフォルト) または**’self’**(シングルクォート必須) - ページ自身と同じオリジンのフレームでのみページを表示できます; **DENY** または**’none’**(シングルクォート必須) - ページをフレームで表示できません。どのサイトが表示しようとしても無効です; スペース区切りのホスト名の文字列; リストに**’self’**(シングルクォート必須) を追加すると、ページ自身と同じオリジンのフレームでページを表示できます。 **’self’** や**’none’** をシングルクォートなしで使用すると、ホスト名として扱われることに注意してください。

パラメータ	説明
iframe サンドボックスの使用	このチェックボックスを外すと、取得した URL コンテンツをサンドボックスに入れることが無効になります (非推奨)。 チェックを入れると、iframe サンドボックスの例外を指定できます。指定されていない制限は引き続き適用されます。このフィールドが空の場合、すべてのサンドボックス属性の制限が適用されます。 詳細は、sandbox属性の説明を参照してください。

2 監査ログ

概要

このセクションでは、監査ログの設定を行うことができます。



以下のパラメータが利用可能です。

パラメータ	説明
監査ログを有効にする	監査ログを有効（デフォルト）または無効にします。
システムアクションのログ	サーバー（システムユーザー）によって実行される低レベルディスカバリ、ネットワークディスカバリ、自動登録アクティビティの監査ログを有効（デフォルト）または無効にします。
内部ハウスキーピングを有効にする	監査ログレコードの内部ハウスキーピングを有効（デフォルト）または無効にします。
データ保存期間	ハウスキーパーによって削除されるまで監査ログレコードを保持する日数。 ハウスキーピングが有効な場合は必須です。 デフォルト：31 日。

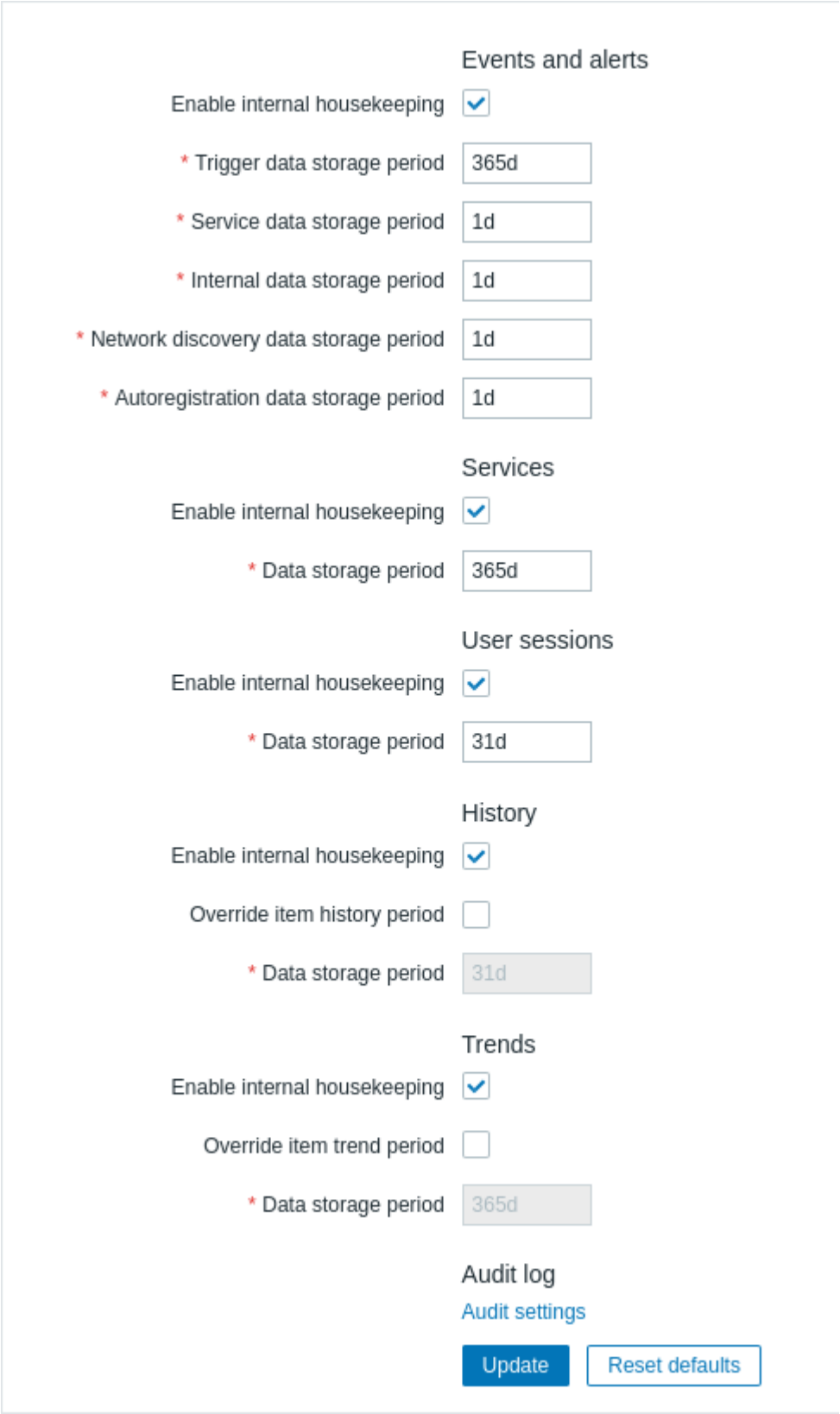
監査ログ圧縮のパラメータは、TimescaleDBを使用している場合に表示されるヒストリ、トレンドおよび監査ログ圧縮ブロック内の管理 > ハウスキーピングセクションで利用できます。

3 ハウスキーピング

概要

管理 > ハウスキーピングセクションでは、以下のデータのうち、古くなったものやユーザーが削除したものをいつ削除するかを設定できます。

- イベントとアラート
- サービス
- ユーザーセッション
- ヒストリ
- トレンド



監査のハウスキーピングは別途設定します。

設定

以下の表は、ハウスキーピングの設定パラメータについて説明しています。

パラメータ	説明
内部ハウスキーピングを有効にする	Zabbix サーバーのhousekeeper プロセスによって実行される内部ハウスキーピングを有効（デフォルト）または無効にします。

パラメータ	説明
データ保存期間	ハウスキーパーによって削除されるまでレコードを保持する期間を指定します。 内部ハウスキーピングが有効な場合は必須です。 範囲：1 日（履歴は 1 時間）～25 年、または「0」。時間のサフィックス（例：1d、1w）がサポートされています。 イベントとアラートについては、トリガー、サービス、内部データ、ネットワークディスカバリ、自動登録ごとにデータ保存期間を個別に設定します。 ハウスキーパーは、問題に関連付けられていないイベントのみを削除します。たとえば、古い問題/復旧イベントが未解決の問題に関連付けられている場合は削除されません。ハウスキーパーが古いエンティティを削除する際は、最初に問題を削除し、その後イベントを削除します。 トリガーが削除されると、その問題と関連するイベントも削除されます。 履歴およびトレンドについては、内部ハウスキーピングが無効でも、データ保存期間によって監視データ > 最新データにデータが表示される期間が決まります。
アイテムの履歴保存期間を上書き	有効にすると、アイテムの設定で指定された履歴/トレンドの保存期間がデータ保存期間の設定で上書きされます（保存しないオプションが有効なアイテムを除く）。 このオプションは、内部ハウスキーピングが無効で外部ハウスキーパーを使用している場合でも利用できます。
アイテムのトレンド保存期間を上書き	

TimescaleDB を使用している場合は、履歴、トレンド、監査ログの圧縮セクションが利用可能になります。

Note:
TimescaleDBを使用する場合は、アイテムの履歴保存期間を上書き、アイテムのトレンド保存期間を上書き、および内部ハウスキーピングを有効にするを有効にして、履歴とトレンドの自動パーティショニングの恩恵を最大限に受けてください。これらのオプションが無効な場合でも、履歴およびトレンドテーブルのデータはパーティション分割されますが、ハウスキーパーは古いパーティションを削除せず、設定警告が表示されます。古いパーティションの削除が有効な場合、Zabbix サーバーおよびフロントエンドは削除されたアイテムを追跡しなくなり、古いパーティションが削除されるとそれらのアイテムの履歴もクリアされます。

デフォルトにリセットボタンを使用すると、行った変更を元に戻すことができます。

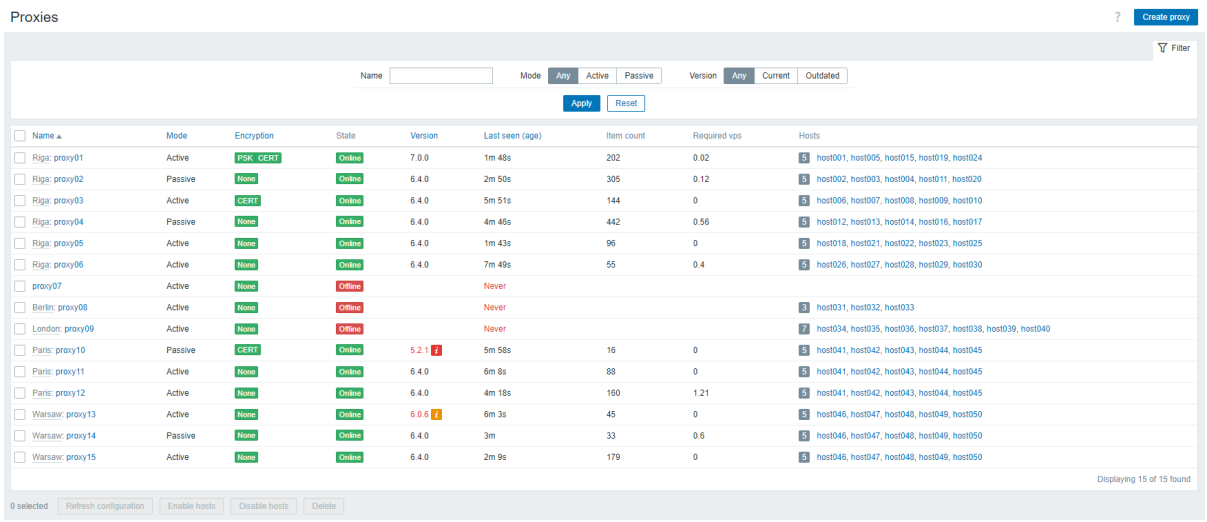
4 プロキシ

概要

管理 → プロキシセクションでは、分散監視用のプロキシを Zabbix フロントエンドで設定できます。

プロキシ

既存のプロキシのリストとその詳細が表示されます。



表示されるデータ:

列	説明
名前	プロキシの名前。 プロキシ名をクリックすると、プロキシの設定フォームが開きます。 プロキシがプロキシグループに属している場合、グループ名がプロキシ名の前にグレーのリンクとして表示されます。グループ名をクリックすると、プロキシグループの設定フォームが開きます。
モード	プロキシモード - アクティブまたは パッシブ。
暗号化	プロキシからの接続の暗号化ステータス: なし - 暗号化なし; **PSK** - 事前共有鍵を使用; **Cert** - 証明書を使用。
状態	プロキシの状態: 不明 - Zabbix サーバーがダウンしている間にプロキシが作成されたか、サーバーがまだ状態を更新していない; オンライン - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信した（パッシブプロキシがサーバーのリクエストに応答した、またはアクティブプロキシがリクエストを送信した）; オフライン - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信していない。
バージョン	プロキシのバージョン（3 桁のバージョン番号）。プロキシが古い、またはサポートされていない場合、バージョン番号が強調表示（赤）され、情報ステータスアイコン（黄色または赤）が表示されます。アイコンにカーソルを合わせると詳細が表示されます。
最終確認（経過時間）	サーバーが最後にプロキシを確認した時刻。
アイテム数	プロキシに割り当てられた有効なホスト上の有効なアイテムの数。
必要な *vps*	必要なプロキシのパフォーマンス（1 秒あたりに収集する必要がある値の数）。
ホスト	プロキシに割り当てられた有効なホストの数と、プロキシが監視しているホストのリスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。

新しいプロキシを設定するには、右上隅のプロキシの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

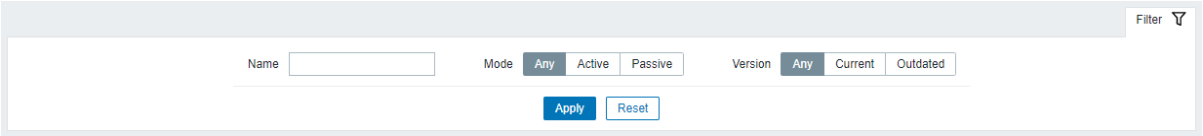
- 設定の更新 - プロキシの設定を更新します。
- ホストを有効 - プロキシで監視されているホストのステータスを監視中に変更します。
- ホストを無効 - プロキシで監視されているホストのステータスを監視対象外に変更します。
- 削除 - プロキシを削除します。

これらのオプションを使用するには、該当するプロキシのチェックボックスをオンにして、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるプロキシのみを表示できます。検索パフォーマンスを向上させるために、データはマクロが未解決のまま検索されます。

フィルターリンクはプロキシのリストの上にあります。これをクリックすると、名前、モード、バージョンでプロキシをフィルタリングできるフィルターが利用可能になります。フィルターオプションの 古いは、古い（部分的にサポートされている）プロキシとサポートされていないプロキシの両方を表示することに注意してください。



5 プロキシグループ

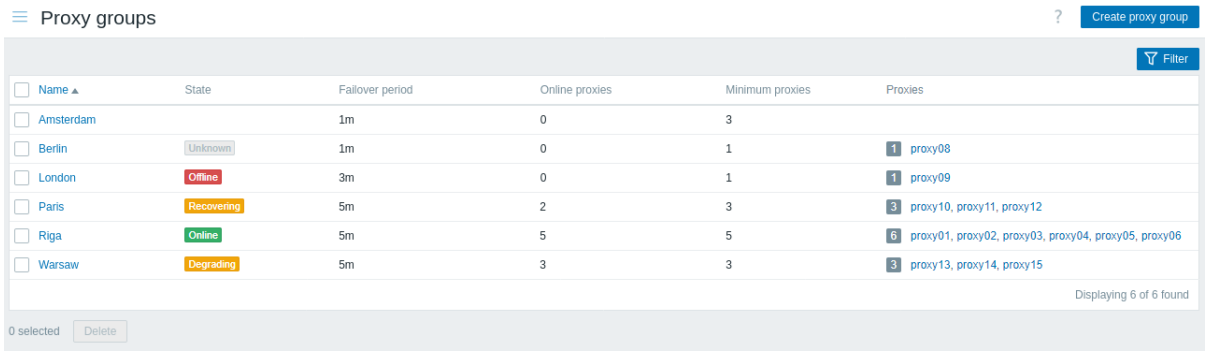
概要

管理 → プロキシグループでは、プロキシグループを設定できます。

プロキシグループは、プロキシ間の自動ホスト分散と高可用性で使用されます。

プロキシグループ

既存のプロキシグループの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
名前	プロキシグループの名前。プロキシグループ名をクリックすると、プロキシグループの設定フォームが開きます。
状態	プロキシグループの状態（グループに少なくとも 1 つのプロキシが含まれている場合に表示）: 不明 - Zabbix サーバーがダウンしている間にグループが作成された、またはサーバーがまだ状態を更新していない; オンライン - フェイルオーバー期間内に最小数のプロキシが Zabbix サーバーと通信した; 低下中 - オンラインプロキシが最小プロキシ数のしきい値を下回ったため、グループがまもなくオフラインになる; オフライン - フェイルオーバー期間内に最小数のプロキシが Zabbix サーバーと通信しなかった; 回復中 - オンラインプロキシが最小プロキシ数のしきい値に達したため、グループがまもなくオンラインに戻る。 プロキシグループの状態は、zabbix[proxy group,<name>,state]およびzabbix[proxy group,discovery]アイテムで監視できます。
フェイルオーバー期間	プロキシグループ内のプロキシがオンラインと見なされるために Zabbix サーバーと通信しなければならない期間。
オンラインプロキシ	オンラインプロキシの数（グループの最小値を下回る場合は赤で表示）。
最小プロキシ数	プロキシグループをオンラインに保つために必要なオンラインプロキシの最小数。
プロキシ	グループ内のプロキシの数と、グループ内のプロキシのリスト（プロキシ設定フォームへのリンク付き）。 リストされるプロキシの最大数は、テーブルセル内に表示する要素の最大数の値によって制限されます。

新しいプロキシグループを設定するには、右上隅のプロキシグループの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます。

- 削除 - プロキシグループを削除します。

これらのオプションを使用するには、該当するプロキシグループの前のチェックボックスをオンにし、必要なボタンをクリックします。

フィルターの使用

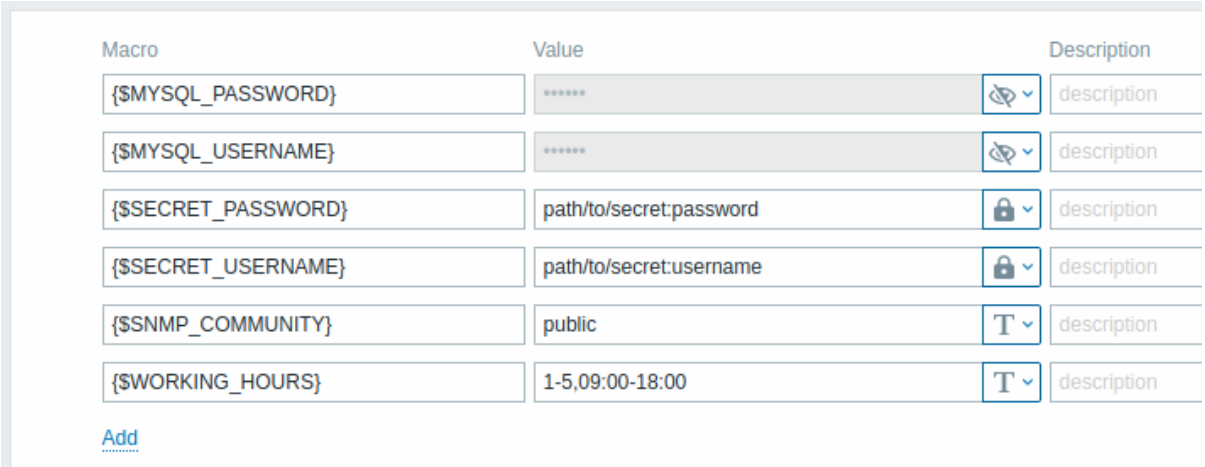
フィルターを使用して、関心のあるプロキシグループのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

フィルターリンクはプロキシグループのリストの上にあります。これをクリックすると、名前やステータスでプロキシグループをフィルタリングできるフィルターが利用可能になります。

6 マクロ

概要

このセクションでは、システム全体で有効なユーザーマクロを名前と値のペアで定義できます。マクロの値はプレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できることに注意してください。説明の追加もサポートされています。



7 キュー

概要

管理 → キューセクションには、更新待ちのアイテムが表示されます。

理想的には、このセクションを開いたときにすべてが「緑色」であるべきです。これは、キューにアイテムがないことを意味します。すべてのアイテムが遅延なく更新されていれば、待機中のアイテムはありません。しかし、サーバーのパフォーマンスが不足している場合、一部のアイテムが遅延し、その情報がこのセクションに表示されます。詳細については、キュー セクションを参照してください。

Note:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

Note:
キューは Zabbix サーバーが稼働している場合のみ利用可能です。接続の問題やエージェントの不具合によりアイテムインターフェースが利用できなくなった場合、アイテムはキューにカウントされません。

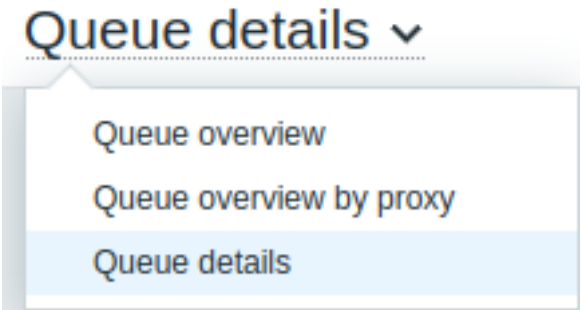
管理 → キューセクションには、以下のページがあります。

- キューの概要 — アイテムタイプごとのキューを表示します。
- プロキシごとのキューの概要 — プロキシごとのキューを表示します。
- キューの詳細 — 遅延しているアイテムのリストを表示します。

利用可能なページのリストは、管理メニューセクションの キューを押すと表示されます。また、左上隅のタイトルドロップダウンを使用してページを切り替えることもできます。



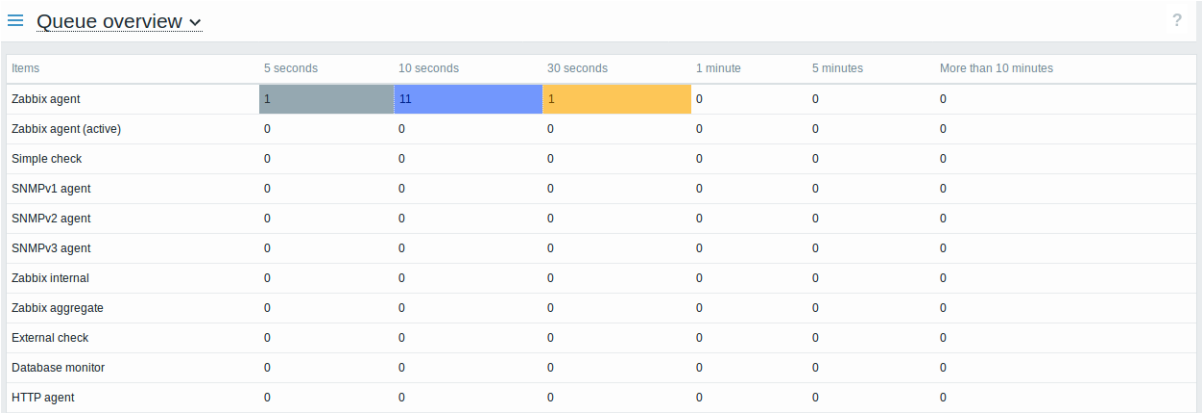
第 3 レベルのメニュー。



タイトルドロップダウン。

キュー概要

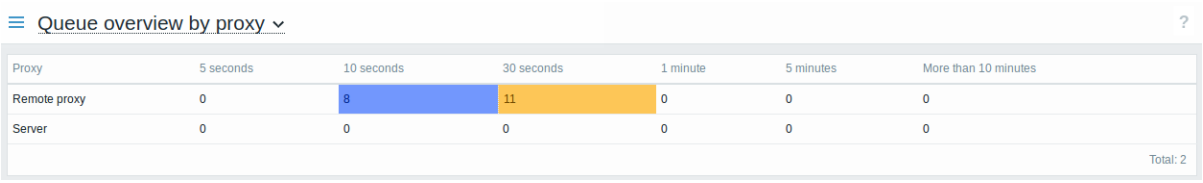
この画面では、問題が 1 つまたは複数のアイテムタイプに関連しているかどうかを簡単に特定できます。



各行にはアイテムタイプが含まれています。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

プロキシごとのキュー概要

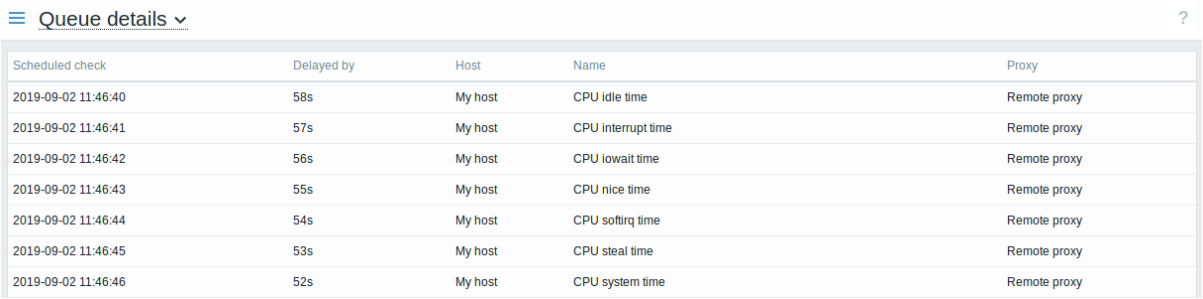
この画面では、問題がプロキシの 1 つに関連しているか、サーバーに関連しているかを簡単に特定できます。



各行にはプロキシが含まれ、サーバーはリストの最後にあります。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

待機中アイテムのリスト

この画面では、各待機中アイテムがリスト表示されます。



表示されるデータ:

列	説明
スケジュールされたチェック	チェックが実行される予定だった時刻が表示されます。
遅延時間	遅延の長さが表示されます。
ホスト	アイテムのホストが表示されます。
名前	待機中アイテムの名前が表示されます。
プロキシ	ホストがプロキシによって監視されている場合、プロキシ名が表示されます。

考えられるエラーメッセージ

データが表示されず、次のエラーメッセージが表示される場合があります。



この場合のエラーメッセージは次のとおりです。

```
Cannot display item queue. Permission denied
```

これは、*zabbix.conf.php* ファイルの PHP 設定パラメータ - $ZBX_SERVER または $ZBX_SERVER と $ZBX_SERVER_PORT の両方 - が、異なるデータベースを使用している既存の Zabbix サーバーを指している場合に発生します。

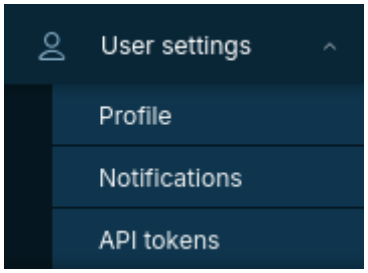
3 ユーザー設定

概要

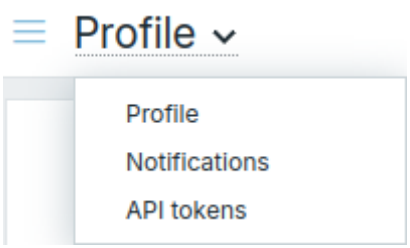
ユーザーロールの権限に応じて、ユーザー設定セクションには次のページが含まれる場合があります。

- プロファイルまたは ユーザープロファイル - Zabbix フロントエンドの特定の機能をカスタマイズするため
- 通知 - 現在のユーザーの通知をカスタマイズするため
- *API* トークン - 現在のユーザーに割り当てられた API トークンを管理するため

利用可能なページのリストは、Zabbix メニューの下部近くにある ユーザーアイコンをクリックすると表示されます（*guest* ユーザーには利用できません）。また、左上隅のタイトルドロップダウンを使用してページを切り替えることもできます。



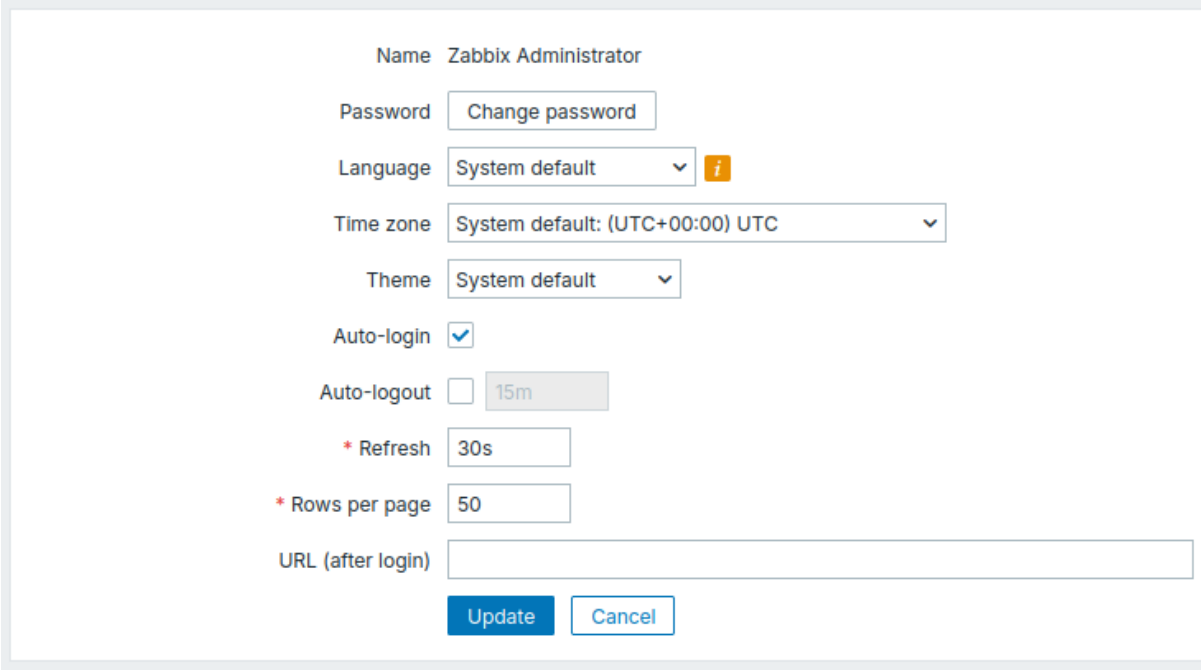
第 3 レベルのメニュー。



タイトルのドロップダウン。

ユーザープロファイル

ユーザープロファイルセクションでは、カスタムインターフェース言語、カラーテーマ、リストに表示される行数などを設定できます。ここで行った変更は現在のユーザーのみに適用されます。



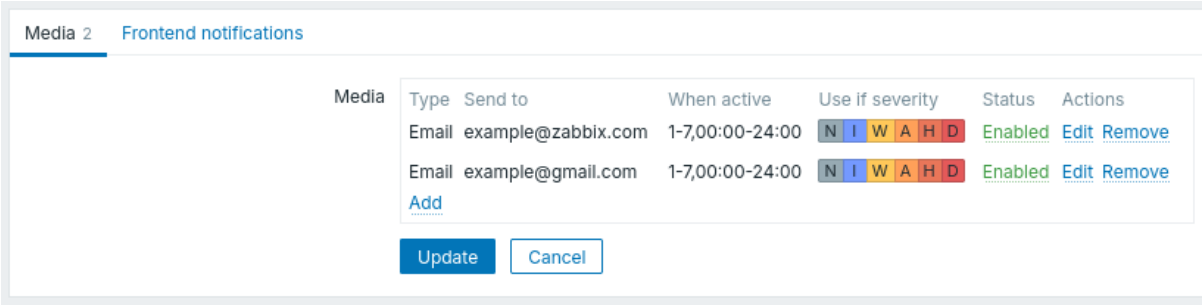
パラメータ	説明
名前	ユーザー名と姓が表示されます。 名前と姓がない場合は、ユーザー名が表示されます。
パスワード	パスワードの変更ボタンをクリックすると、現在のパスワード、新しいパスワード、新しいパスワード (再入力) の 3 つのフィールドが表示されます。 パスワードの変更に成功すると、すべてのアクティブなセッションからログアウトされます。 パスワードは Zabbix の内部認証を使用しているユーザーのみ変更できます。

パラメータ	説明
言語	希望するインターフェース言語を選択するか、システムデフォルトを選択してシステムのデフォルト設定を使用します。 *English (en_US)* を選択すると、フロントエンドで米国の時刻/日付形式も有効になります。 詳細は追加フロントエンド言語のインストールを参照してください。
タイムゾーン	ユーザーレベルでグローバルなタイムゾーンを上書きするタイムゾーンを選択するか、システムデフォルトを選択してグローバルなタイムゾーン設定を使用します。
テーマ	プロファイル専用のカラーテーマを選択します: システムデフォルト - システムのデフォルト設定を使用; ブルー - 標準のブルーテーマ; ブルー (クラシック) - レガシーフォントを使用した標準のブルーテーマ; ダーク - 代替のダークテーマ; ダーク (クラシック) - レガシーフォントを使用した代替のダークテーマ; ハイコントラスト (ライト) - 高コントラストのライトテーマ; ハイコントラスト (ダーク) - 高コントラストのダークテーマ。
自動ログイン	このチェックボックスをオンにすると、Zabbix があなたを記憶し、30 日間自動的にログインします。 *30* 日間記憶するを有効にしてログインした場合: - 自動ログアウトタイマーがリセットされます (セッションは 30 日間持続します)。 - 30 日以内の次回アクセス時に自動ログインが有効になります。 *30* 日間記憶するを使用せずにログインした場合: - 自動ログインパラメータがクリアされます。 - 自動ログアウトは通常通り設定されたタイムアウトに従います。 この機能にはブラウザの Cookie が使用されます。
自動ログアウト	このチェックボックスをオンにすると、設定した秒数後 (最小 90 秒、最大 1 日) に自動的にログアウトされます。 時間のサフィックスがサポートされています (例: 90s, 5m, 2h, 1d)。 このオプションは以下の場合には機能しません: * 監視メニューページがバックグラウンドで情報を更新している場合。一定間隔でデータを更新するページ (ダッシュボード、グラフ、最新データなど) を開いたままにすると、セッションの有効期間が延長され、自動ログアウト機能が無効になります。 * *30* 日間記憶するオプションをオンにしてログインした場合。この場合、自動ログアウトタイマーは上書きされ、セッションは全期間アクティブのままです。 * SSO アイデンティティプロバイダー (IdP) 経由で認証が行われている場合、自動ログアウトは Zabbix セッションのみを終了します。IdP との SSO セッションはアクティブなままなので、ユーザーはパスワードを入力せずに再度ログインできる場合があります。 自動ログアウトには”0” も指定でき、プロファイル設定の更新後に自動ログアウト機能が無効になります。
更新間隔	監視メニューページの情報をどのくらいの間隔で更新するかを設定します (最小 0 秒、最大 1 時間)。 時間のサフィックスがサポートされています (例: 30s, 90s, 1m, 1h)。
1 ページあたりの行数	リストで 1 ページに表示する行数を設定します。行数 (および表示するレコード数) が少ないほど、読み込み時間が短くなります。
ログイン後の *URL*	ログイン後に表示する特定の URL を設定します。デフォルトのダッシュボードの代わりに、例えば監視 > トリガーの URL などを指定できます。

通知

通知セクションでは、現在のユーザーの通知をカスタマイズするためのオプションを提供します。

メディアタブでは、ユーザーのメディアの詳細（メディアタイプや通知を配信するために使用するアドレス、使用するタイミングなど）を指定できます。



メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に「無効」と表示されます。

Note:
ユーザー自身がメディアの詳細を変更する権限は、ユーザーロール（自身のメディアの作成と編集オプションを参照）に基づいて付与/取り消しできます。スーパ管理者が他のユーザーのメディアの詳細を変更する権限も、ユーザーロール（ユーザーメディアの作成と編集オプションを参照）に基づいて付与/取り消しできます。

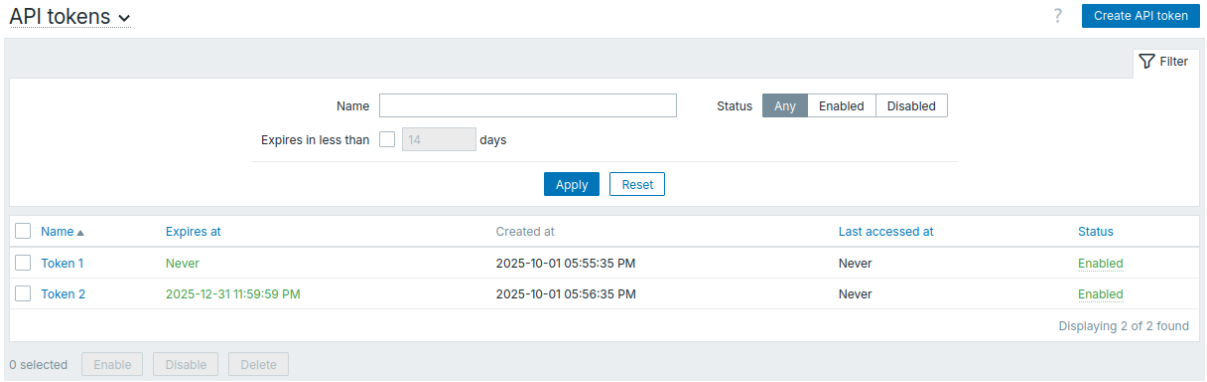
プロビジョニングされたユーザーの場合は、次の点に注意してください:

- プロビジョニングされたユーザーメディアは削除できません。
- プロビジョニングされたユーザーメディアは無効化/有効化できます。
- 有効期間、深刻度で使用、有効などのプロビジョニングされたユーザーメディアのフィールドは手動で編集できます。
- プロビジョニングされたユーザーには、追加のユーザーメディア（たとえば追加のメールアドレス）を手動で追加できます。
- 手動で追加したユーザーメディアは削除できます。

フロントエンド通知タブでは、グローバル通知を設定できます。

API トークン

API トークンセクションでは、ユーザーに割り当てられたトークンの表示、トークンの詳細の編集、新しいトークンの作成ができます。このセクションは、ユーザーロールの設定で *API* トークンの管理アクションが許可されている場合にのみ、ユーザーが利用できます。



API トークンは、名前、有効期限、またはステータス (有効/無効) でフィルタリングできます。リスト内のトークンステータスをクリックすると、トークンをすばやく有効/無効にできます。また、リスト内で複数のトークンを選択し、リスト下部の有効/無効ボタンをクリックすることで、一度に複数のトークンを有効/無効にすることもできます。

Attention:
ユーザーは、Zabbix で自分に割り当てられたトークンの *Auth token* 値を表示できません。*Auth token* 値は、トークン作成直後の一度だけ表示されます。トークンを紛失した場合は、再生成する必要があります。

1 グローバル通知

概要

グローバル通知は、Zabbix フロントエンド内の現在の画面にリアルタイムの障害を直接表示する方法を提供します。

グローバル通知がない場合、障害やダッシュボードセクション以外で作業していると、現在の障害に関する情報を受け取ることができません。グローバル通知は、Zabbix フロントエンド内の現在の場所に関係なく、この情報が表示されることを保証します。

グローバル通知には、メッセージの表示とサウンドの再生の両方が含まれます。

Attention:
最近のブラウザバージョンでは、サウンドの自動再生が（デフォルトで）無効になっている場合があります。その場合は、この設定を手動で有効にする必要があります。

設定

グローバル通知は、通知セクションのフロントエンド通知タブでユーザーごとに有効にできます。

profile_c.png

パラメータ	説明
フロントエンド通知	チェックボックスをオンにすると、グローバル通知が有効になります。
メッセージタイムアウト	メッセージが表示される時間を設定します。デフォルトでは、メッセージは 60 秒間画面に表示されます。 時間のサフィックスがサポートされています。例: 30s, 5m, 2h, 1d。
サウンドを再生	サウンドが再生される時間を設定します。 一度だけ - サウンドが一度だけ完全に再生されます; **10** 秒 - サウンドが 10 秒間繰り返されます; メッセージタイムアウト - メッセージが表示されている間、サウンドが繰り返されます。
トリガーの深刻度	グローバル通知とサウンドを有効にするトリガーの深刻度を設定します。さまざまな深刻度に応じたサウンドも選択できます。 深刻度が 1 つも選択されていない場合、メッセージは表示されません。 また、復旧メッセージは選択された深刻度に対してのみ表示されます。たとえば、復旧と致命的が選択されている場合、致命的のトリガーの問題と復旧に対してグローバル通知が表示されます。
抑制された問題を表示	チェックボックスをオンにすると、ホストのメンテナンスによって抑制（非表示）されるはずの問題についても通知が表示されます。

グローバルメッセージの表示

メッセージが到着すると、画面右側のフローティングセクションに表示されます。このセクションは、セクションヘッダーをドラッグすることで自由に位置を変更できます。

global_messages.png

このセクションには、いくつかのコントロールがあります:

- スヌーズボタンは、現在アクティブなアラーム音を消音します。
- ミュート/ミュート解除ボタンは、アラーム音の再生/非再生を切り替えます。

2 ブラウザのサウンド

概要

サウンドはグローバル通知で使用されます。

Zabbix フロントエンドでサウンドを再生するには、ユーザープロファイルのフロントエンド通知タブでフロントエンド通知を有効にし、すべてのトリガーの深刻度にチェックを入れる必要があります。さらに、グローバル通知のポップアップウィンドウでサウンドを有効にする必要があります。

何らかの理由でデバイスでオーディオを再生できない場合、グローバル通知のポップアップウィンドウの  ボタンは常に「ミュート」状態のままとなり、  ボタンにカーソルを合わせると「このデバイスでは通知オーディオをサポートできません」というメッセージが表示されます。

デフォルトのオーディオクリップを含むサウンドは、MP3 形式のみサポートされています。

Zabbix フロントエンドのサウンドは、Linux 上の最新の Firefox および Opera ブラウザ、Windows 上の Chrome、Firefox、Microsoft Edge、Opera ブラウザで正常にテストされています。

Attention:
最近のブラウザバージョンでは、サウンドの自動再生が（デフォルトで）無効になっている場合があります。その場合は、この設定を手動で有効にする必要があります。

4 グローバル検索

Zabbix フロントエンドで、ホスト、ホストグループ、テンプレート、テンプレートグループを検索することができます。

検索入力ボックスは、メニューの Zabbix ロゴの下にあります。*Enter* を押すか、  検索アイコンをクリックすると、検索が開始されます。



入力した文字列が名前のいずれかの部分に含まれているホストがある場合、ドロップダウンが表示され、そのようなすべてのホストがリストされます（該当部分はオレンジ色でハイライトされます）。また、ホストの表示名が検索文字列として入力された技術名と一致する場合も、そのホストがドロップダウンにリストされますが、ハイライトはされません。

検索可能な属性

ホストは以下のプロパティで検索できます:

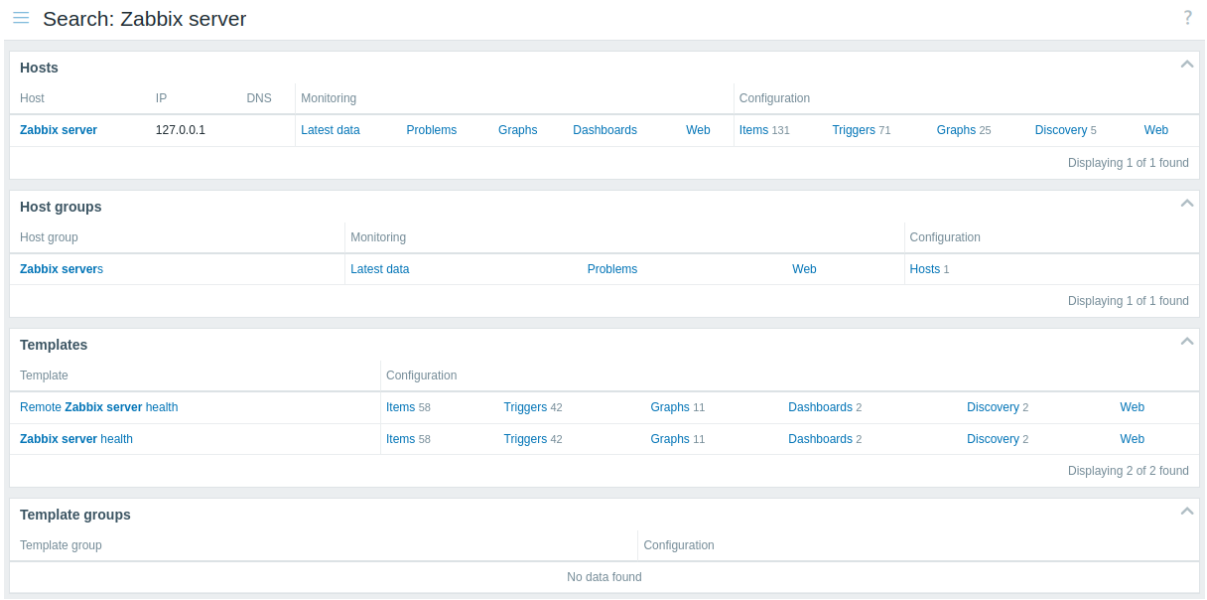
- ホスト名
- 表示名
- IP アドレス
- DNS 名

テンプレートは名前または表示名で検索できます。表示名と異なる名前（テンプレート/ホストの）で検索した場合、検索結果では表示名の下に括弧付きで表示されます。

ホストグループおよびテンプレートグループは名前で検索できます。親グループを指定すると、すべてのネストされたグループが暗黙的に選択されます。

検索結果

検索結果は、ホスト、ホストグループ、テンプレート、テンプレートグループの 4 つの個別のブロックで構成されています。



各ブロックは個別に折りたたみ/展開が可能です。各ブロックの下部には、エントリ数が表示されます。例: *13* 件中 *13* 件を表示。エントリがない場合は、エントリ数は表示されません。1 つのブロック内で表示されるエントリの合計は 100 件までに制限されています。

各エントリには、監視および設定データへのリンクが用意されています。リンクの全リストを参照してください。

すべての設定データ（アイテム、トリガー、グラフなど）については、エンティティ名の横にグレーで見つかったエンティティの数が表示されます。注意: エンティティが 0 の場合は、数値は表示されません。

有効なホストは青色、無効なホストは赤色で表示されます。

利用可能なリンク

各エントリについて、次のリンクを利用できます:

- Hosts
  - Monitoring
    - Latest data
    - Problems
    - Graphs
    - Host dashboards
    - Web scenarios
  - Configuration
    - Items
    - Triggers
    - Graphs
    - Discovery rules
    - Web scenarios
- Host groups
  - Monitoring
    - Latest data
    - Problems
    - Web scenarios
  - Configuration
    - Hosts
- Templates
  - Configuration
    - Items
    - Triggers
    - Graphs
    - Template dashboards
    - Discovery rules
    - Web scenarios
- Template groups
  - Configuration
    - Templates

5 フロントエンドメンテナンスモード

概要

Zabbix フロントエンドへのアクセスを制限するために、一時的に Zabbix フロントエンドを無効にすることができます。これは、ユーザーによる変更から Zabbix データベースを保護し、その整合性を維持するのに役立ちます。

Zabbix フロントエンドがメンテナンスモードの場合、データベースを安全に停止し、メンテナンス作業を実行できます。

定義された IP アドレスからのユーザーは、メンテナンスモード中も通常通りフロントエンドとやり取りできます。

設定

メンテナンスモードを有効にするには、`maintenance.inc.php` ファイル（Web サーバーの Zabbix HTML ドキュメントディレクトリの`/conf` にあります）を開き、次の行のコメントを解除します。

```
// メンテナンスモード
define('ZBX_DENY_GUI_ACCESS', 1);

// フロントエンドへの接続を許可するIPアドレスの配列（オプション）
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// 警告画面に表示されるメッセージ（オプション）
$ZBX_GUI_ACCESS_MESSAGE = 'MySQLデータベースを15:00までアップグレードしています。しばらくお待ちください..
```

Note:
ほとんどの場合、maintenance.inc.php ファイルは Web サーバーの Zabbix HTML ドキュメントディレクトリの /conf にあります。ただし、一部のオペレーティングシステムや Web サーバーでは異なる場所を使用する場合があります。
例えば、以下の場所です:

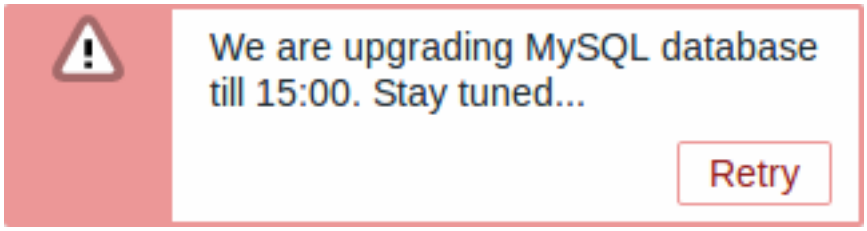
- SUSE および RedHat では /etc/zabbix/web/maintenance.inc.php
- Debian 系システムでは /usr/share/zabbix/conf/

PHP ファイルのコピー も参照してください。

パラメータ	詳細
ZBX_DENY_GUI_ACCESS	何らかの値で定義されている場合、メンテナンスモードが有効になります。 メンテナンスモードを無効にするには、コメントアウトまたは削除してください。
ZBX_GUI_ACCESS_IP_RANGE	フロントエンドへの接続を許可する IP アドレスの配列（オプション）。 例: array('192.168.1.1', '192.168.1.2')
ZBX_GUI_ACCESS_MESSAGE	メンテナンスについてユーザーに通知するメッセージ（オプション）。 未定義の場合、デフォルトメッセージ *'Zabbix is under maintenance'* が使用されます。

表示

メンテナンスモード中に Zabbix フロントエンドにアクセスしようとすると、ユーザーには以下の画面が表示されます。メンテナンスが終了した際にユーザーの操作なしで通常の状態に戻るように、画面は 30 秒ごとに更新されます。



ZBX_GUI_ACCESS_IP_RANGE で定義された IP アドレスは、通常通りフロントエンドにアクセスできます。

6 ページパラメーター

概要

ほとんどの Zabbix Web インターフェースのページは、表示内容を制御するさまざまな HTTP GET パラメータをサポートしています。これらは、URL の後にパラメータ = 値のペアを指定し、URL とは疑問符（?）で区切り、パラメータ同士はアンパサンド（&）で区切ることで渡すことができます。

監視 > 障害

以下のパラメータがサポートされています:

パラメータ	説明	例
show	フィルターオプション 表示。 可能な値: 1 - 最近の障害; 2 - すべて; 3 - 障害状態。	show=1
name	フィルターオプション 障害: フリーフォーム文字列。	name=Zabbix agent

パラメータ	説明	例
severities	フィルターオプション 深刻度: `severities[*]=*` 形式の選択された深刻度の配列 (* を深刻度レベルに置き換えます)。 可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。	severities[3]=3
inventory	フィルターオプション ホストインベントリ: インベントリフィールド `[field],[value]` の配列	inventory[0][field]=type&invento
evaltype	フィルターオプション タグ: タグ評価方法。 可能な値: 0 - and/or; 2 - or。	evaltype=0
tags	フィルターオプション タグ: 定義済みタグ `[tag], [operator], [value]` の配列 operator の可能な値: 0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。	tags[0][tag]=target&tags[0][ope
show_tags	フィルターオプション タグの表示。 可能な値: 0 - なし; 1 - 1 つ; 2 - 2 つ; 3 - 3 つ。	show_tags=3
tag_name_format	フィルターオプション タグ名。 可能な値: 0 - フルネーム; 1 - 短縮名; 2 - なし。	tag_name_format=1
tag_priority	フィルターオプション タグ表示優先度: カンマ区切りのタグ表示優先度の文字列	tag_priority=customer, target
show_suppressed	フィルターオプション 抑制された障害を表示。 可能な値: 1 - 表示; 2 - 表示しない。	show_suppressed=1
acknowledgement_status	フィルターオプション 障害対応状況。 可能な値: 0 - すべて; 1 - 未対応; 2 - 対応済み。	acknowledgement_status=0
acknowledged_by_me	フィルターオプション 自分が対応; `acknowledgement_status=2` の場合のみサポート。 可能な値: 0 - 無効; 1 - 有効。	acknowledged_by_me=1

パラメータ	説明	例
compact_view	フィルターオプション コンパクト表示。 可能な値: 0 - 無効; 1 - 有効。	compact_view=1
highlight_row	フィルターオプション 行全体を強調表示。 可能な値: 0 - 無効; 1 - 有効。	highlight_row=1
filter_name	フィルターのプロパティオプション 名前: フリーフォーム文字列	filter_name=Databases
filter_show_counter	フィルターのプロパティオプション レコード数を表示。 可能な値: 0 - 無効; 1 - 有効。	filter_show_counter=1
filter_custom_time	フィルターのプロパティオプション カスタム期間を設定。 可能な値: 0 - 無効; 1 - 有効。	filter_custom_time=1
sort	ソートする列。 可能な値: clock - 時刻列でソート; host - ホスト列でソート; severity - 深刻度列でソート; name - 障害列でソート。	sort=clock
sortorder	結果のソート順。 可能な値: DESC - 降順; ASC - 昇順。	sortorder=DESC
age_state	フィルターオプション 経過日数が次より小さい; `show=3` の場合のみサポート。 可能な値: 0 - age パラメータを無効化; 1 - age パラメータを有効化。	age_state=1
age	フィルターオプション 経過日数が次より小さい: 整数、日数; `age_state=1` かつ `show=3` の場合のみサポート。	age=7
groupids	フィルターオプション ホストグループ: ホストグループ ID の配列。	groupids[]=4
hostids	フィルターオプション ホスト: ホスト ID の配列	hostids[]=10084
triggerids	フィルターオプション トリガー: トリガー ID の配列	triggerids[]=22382
show_timeline	フィルターオプション タイムラインを表示; `compact_view=1` ではサポートされません。 可能な値: 0 - 表示しない; 1 - 表示。	show_timeline=1
details	フィルターオプション 詳細を表示。 可能な値: 0 - 表示しない; 1 - 表示。	details=1
from	日付範囲の開始、相対指定可 (例: now-1m); `filter_custom_time=1` の場合のみサポート	from=now-2h
to	日付範囲の終了、相対指定可 (例: now-1m); `filter_custom_time=1` の場合のみサポート	to=now

参照: 障害ページのフィルターオプション。

キオスクモード

サポートされているフロントエンドページでは、URL パラメータを使用してキオスクモードを有効にできます。例えば、ダッシュボードでは:

- `/zabbix.php?action=dashboard.view&kiosk=1` - キオスクモードを有効にする
- `/zabbix.php?action=dashboard.view&kiosk=0` - 通常モードを有効にする

スライドショー

ダッシュボードでスライドショーをアクティブにするには次の通りです。

- `/zabbix.php?action=dashboard.view&slideshow=1` - スライドショーをアクティブに

7 定義

概要

フロントエンドの多くの項目はフロントエンド自体で設定できますが、現在一部のカスタマイズは定義ファイルを編集することでのみ可能です。

このファイルは、Zabbix HTML ドキュメントディレクトリの/include にある `defines.inc.php` です。

パラメータ

このファイルの中でユーザーが関心を持つ可能性のあるパラメータ:

- ZBX_MIN_PERIOD

グラフの最小期間（秒単位）。デフォルトは 1 分。

- GRAPH_YAXIS_SIDE_DEFAULT

シンプルグラフの Y 軸のデフォルト位置、およびカスタムグラフにアイテムを追加する際のドロップダウンボックスのデフォルト値。可能な値: 0 - 左, 1 - 右。

デフォルト: 0

- ZBX_SESSION_NAME

Zabbix フロントエンドのセッション Cookie の名前として使用される文字列。

デフォルト: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Vault レスポンスのデータキャッシュを無効にするために使用される TTL タイムアウト（秒単位）。0 を設定すると Vault レスポンスのキャッシュを無効にします。

デフォルト: 60

- SUBFILTER_VALUES_PER_GROUP

グループごとのサブフィルター値の数（例: 最新データサブフィルター）。

デフォルト: 1000

- ZBX_MAX_WIDGET_LINES

表示するウィジェット行の最大数。

デフォルト: 1000

8 独自テーマ作成

概要

デフォルトで Zabbix はいくつか事前定義されたテーマを提供します。独自のテーマを作成するには、ここに記載されている手順に従ってください。何かいいものを作成した場合は、作業の結果を Zabbix コミュニティと自由に共有してください。

ステップ 1

独自のテーマを定義するには、CSS ファイルを作成し、それを `assets/styles/`フォルダー（たとえば、*custom-theme.css*）に保存します。別のテーマからファイルをコピーして、それを元にテーマを作成するか、最初から始めることができます。

ステップ 2

APP::getThemes() メソッドによって返されるテーマのリストにテーマを追加します。このためには APP クラスの ZBase::getThemes() メソッドをオーバーライドします。これは *include/classes/core/APP.php* の閉じ中括弧の前に次のコードを追加することで実行できます。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:
引用符の最初のペアで指定する名前は、拡張子なしのテーマファイルの名前と一致する必要があることに注意してください。

複数のテーマを追加するには、最初のテーマの下にそれらをリストします。次に例を示します。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

最後のテーマを除くすべてのテーマには、末尾にカンマが必要であることに注意してください。

Note:
グラフの色を変更するには、エントリを *graph_theme* データベーステーブルに追加する必要があります。

ステップ 3

新しいテーマをアクティブにします。

Zabbix フロントエンドで、新しいテーマをデフォルトのテーマに設定したり、ユーザープロファイルで各ユーザーのテーマを変更することができます。

新しいルックアンドフィールをお楽しみください！

9 デバッグモード

概要

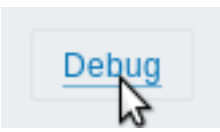
デバッグモードは、フロントエンドページの問題を診断するために使用できます。

設定

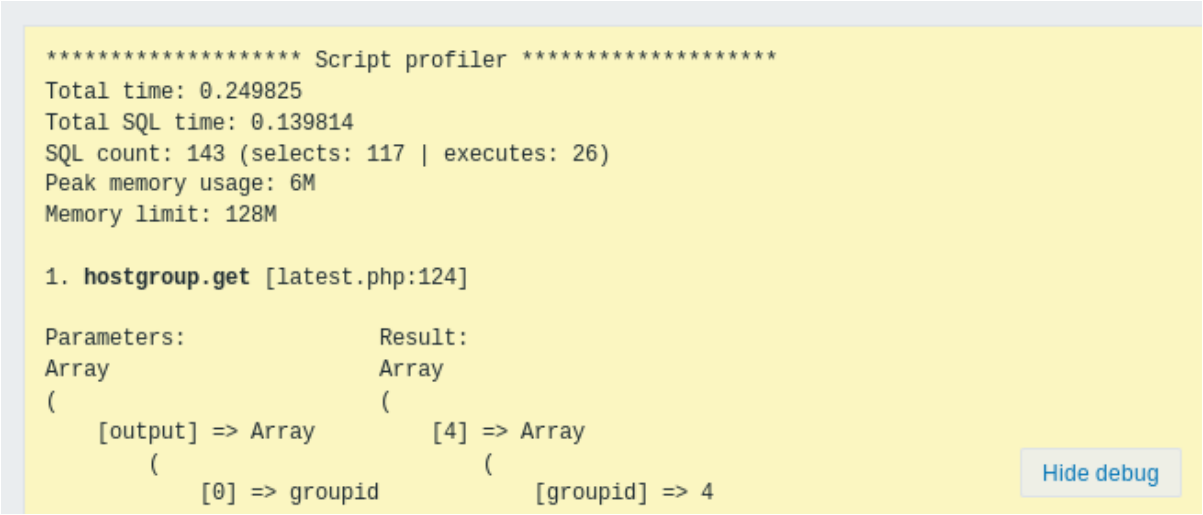
デバッグモードは、ユーザーグループに属する個々のユーザーに対して有効化できます。

- ユーザーグループの設定時
- 設定済みのユーザーグループの表示時

デバッグモードがユーザーグループに対して有効化されている場合、そのグループのユーザーはブラウザウィンドウの右下に *Debug* ボタンが表示されます。



Debug ボタンをクリックすると、ページ内容の下に新しいウィンドウが開き、そのページの SQL 統計情報、API コールのリスト、個々の SQL 文が表示されます。



ページのパフォーマンスに問題がある場合、このウィンドウを使用して問題の根本原因を調査できます。

Warning:
デバッグモードを有効にすると、フロントエンドのパフォーマンスが低下します。

10 Zabbix 用 Cookie

概要

このページでは、Zabbix で使用されるクッキーの一覧を提供します。

名前	説明	値	有効期限/最大年齢	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Zabbix フロントエンドのセッションデータ。base64 でエンコードされた JSON として保存される		セッション（ブラウジングセッションが終了すると期限切れ）	+	+（Web サーバーで HTTPS が有効な場合のみ）
tab	アクティブなタブ番号。このクッキーは複数のタブを持つページ（例：ホスト、トリガー、アクションの設定ページ）でのみ使用され、ユーザーがプライマリタブから他のタブ（タグや依存関係タブなど）に移動したときに作成されます。 0 はプライマリタブを表します。	例：1	セッション（ブラウジングセッションが終了すると期限切れ）	-	-
browserwarning_ignore	古いブラウザを使用していることに関する警告を無視するかどうか。	yes	セッション（ブラウジングセッションが終了すると期限切れ）	-	-
system-message-ok	ページがリロードされた直後に表示するメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされた直後	+	-
system-message-error	ページがリロードされた直後に表示するエラーメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされた直後	+	-

Note:
Web サーバーのディレクティブによる Zabbix クッキーへの'HttpOnly' フラグの強制はサポートされていません。

脚注

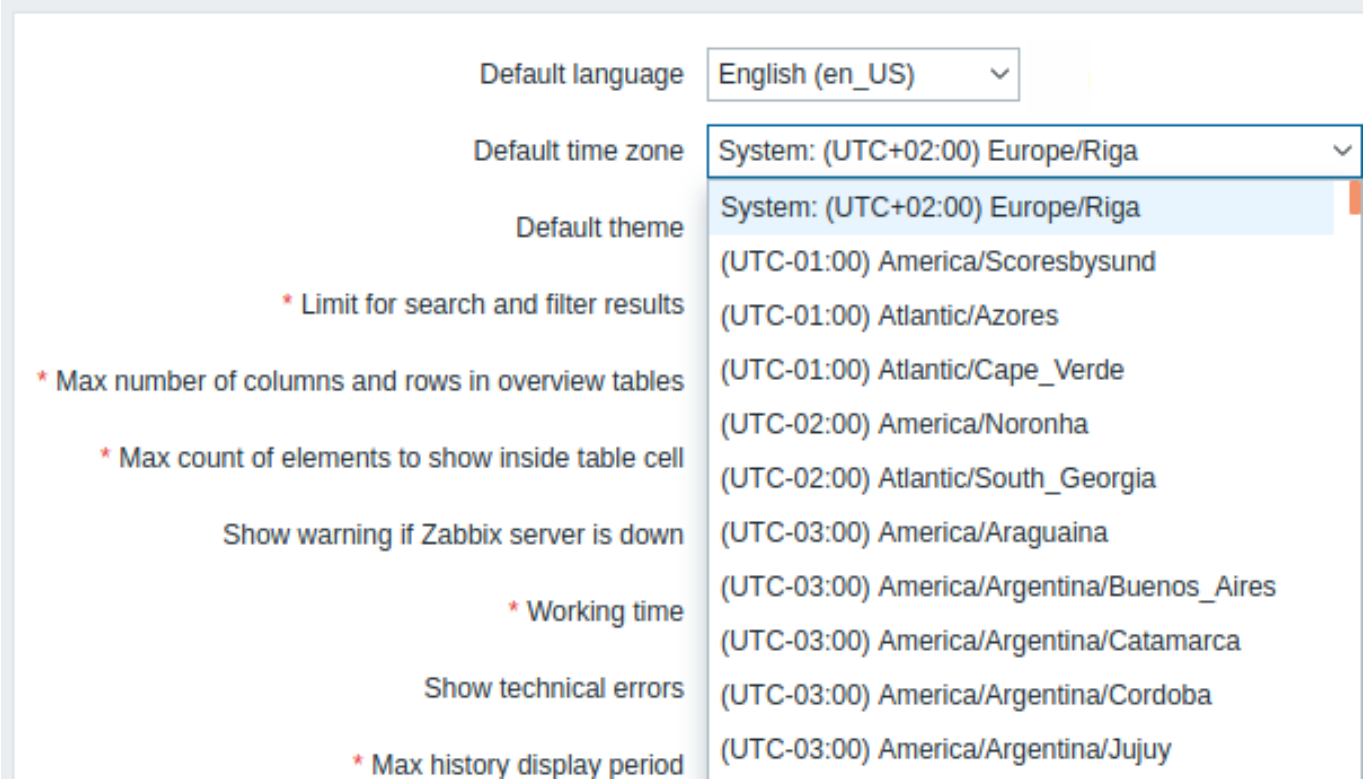
[1] `HttpOnly` が'true' の場合、クッキーは HTTP プロトコルを通じてのみアクセス可能になります。つまり、JavaScript などのスクリプト言語からはクッキーにアクセスできません。この設定は、XSS 攻撃による ID 盗難のリスクを効果的に減らすのに役立ちます（ただし、すべてのブラウザでサポートされているわけではありません）。

[2] `Secure` は、クッキーがクライアントから安全な HTTPS 接続を介してのみ送信されるべきであることを示します。'true' に設定すると、安全な接続が存在する場合にのみクッキーが設定されます。

11 タイムゾーン

概要

フロントエンドのタイムゾーンは、フロントエンドでグローバルに設定でき、個々のユーザーごとに調整することもできます。



System を選択した場合、フロントエンドには Web サーバーのタイムゾーン（php.ini の'date.timezone' の値が設定されていればそれも含む）が使用され、Zabbix サーバーは実行されているマシンのタイムゾーンを使用します。

Note:
Zabbix サーバーは、通知内のマクロ展開時（例：{EVENT.TIME} がユーザーごとに異なるタイムゾーンで展開される場合）および通知送信時の時間制限（ユーザーメディア設定の「有効な時間」設定を参照）にのみ、指定されたグローバル/ユーザーのタイムゾーンを使用します。

タイムゾーンの選択は、フロントエンドの日時フォーマットには影響しません。代わりに、インストール時またはユーザー設定でインターフェース言語を調整できます。*English (en_US)* を選択すると、フロントエンドで米国の日時フォーマットも有効になります。

設定

グローバルタイムゾーン:

- フロントエンドのインストール時に手動で設定可能
- 管理 → 一般 → *GUI* で変更可能

ユーザーレベルのタイムゾーン:

- ユーザーを設定/更新する際に設定可能
- 各ユーザーがユーザープロファイルで設定可能

関連情報**:** スケジューリング間隔使用時のタイムゾーンの調整についても参照してください。

12 パスワードのリセット

概要　このセクションでは、Zabbix のユーザーパスワードをリセットする手順について説明します。

ステップ　Zabbix パスワードを忘れてログインできない場合は、Zabbix 管理者に問い合わせてください。

スーパー管理者ユーザーは、ユーザー設定フォーム 内のすべてのユーザーのパスワードを変更できます。

スーパー管理者がパスワードを忘れてログインできない場合は、次の SQL クエリを実行して、デフォルトパスワードをスーパー管理者ユーザーに適用する必要があります (「Admin」をスーパー管理者ユーザー名に置き換えます)

UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username = 'Admin';

このクエリを実行すると、ユーザー名'Admin' のパスワードは *zabbix* に設定されます。最初のログイン時に必ずパスワードをデフォルトから変更してください。

13 時間帯とホストセレクター

概要

このページでは、ダッシュボードで利用可能な期間およびホストセレクターについて説明します。これらのセレクターを使用すると、個々のウィジェットの設定を編集することなく、すべての互換性のあるダッシュボードウィジェットで表示されるデータを動的に調整できます。

期間セレクター

期間セレクターを使用すると、よく使う期間をワンクリックで選択できます。ダッシュボードの右上隅にある期間セレクターボタンをクリックすることで、展開または折りたたむことができます。

今日、今週などのオプションは、将来の時間/日を含む期間全体を表示します。今日まで、今週までなどのオプションは、経過した時間のみを表示します。

期間を選択した後、  の矢印ボタンをクリックすることで、期間を前後に移動できます。ズームアウトボタンを使用すると、期間を各方向に 50% ずつ拡大できます。

Note:
グラフでは、グラフ内で左マウスボタンを使って範囲をハイライトすることで、表示する期間を選択することもできます。左マウスボタンを離すと、グラフはハイライトした範囲にズームインします。グラフ内をダブルクリックすることでズームアウトも可能です。

From/To フィールドには、選択した期間が絶対時間構文（`Y-m-d H:i:s` 形式）または相対時間構文で表示されます。相対期間には、1 つまたは複数の数学演算（`-`または `+`）を含めることができ、例えば `now-1d` や `now-1d-2h+5m` のように指定できます。

サポートされている相対時間の略語は以下の通りです。

- `now`
- `s`（秒）
- `m`（分）
- `h`（時間）
- `d`（日）
- `w`（週）
- `M`（月）
- `y`（年）

期間セレクターでは精度指定（例えば、`now-1d/M` の`/M`）もサポートされています。精度指定の詳細は以下の通りです。

精度	From	To
`m`	Y-m-d H:m:00	Y-m-d H:m:59
`h`	Y-m-d H:00:00	Y-m-d H:59:59
`d`	Y-m-d 00:00:00	Y-m-d 23:59:59
`w`	その週の月曜日 00:00:00	その週の日曜日 23:59:59
`M`	その月の 1 日 00:00:00	その月の最終日 23:59:59

精度	From	To
y	その年の 1 月 1 日 00:00:00	その年の 12 月 31 日 23:59:59

日付ピッカーを使用して期間を選択することも可能です。開くには、*From/To* フィールドの横にあるカレンダーアイコンをクリックします。

Note:
日付ピッカー内では、Tab、Shift+Tab、キーボードの矢印ボタンで年/月/日を移動できます。Enter キーを押すと選択が確定します。

例:

From	To	選択した期間
now/d	now/d	本日の 00:00 - 23:59
now/d	now/d+1d	本日の 00:00 - 明日の 23:59
now/w	now/w	今週の月曜日 00:00:00 - 日曜日 23:59:59
now-1y/w	now-1y/w	1 年前の週の月曜日 00:00:00 - 日曜日 23:59:59

Attention:
To パラメータに”now/M+1M” を使用すると、31 日が加算され、月の日数によっては日付が 1～3 日ずれる場合があります。たとえば、1 月に使用すると、期待される 2 月 28 日ではなく 3 月 2 日になることがあります。この問題を回避するには、”now/M-3d+1M/M” を使用してください。これにより、月の日数に正確に調整されます。同様に、*From* を過去に設定する場合は、”now/M+3d-1M/M” を使用してください。

ホストセレクター

ホストセレクターを使用すると、互換性のあるダッシュボードウィジェットにデータを表示するホストを選択できます。

ダッシュボードを編集して保存した後、少なくとも 1 つのウィジェットがホストの上書きパラメータでダッシュボードをデータソースとして使用するように設定されている場合に利用可能になります。



17 ベストプラクティス

概要　このセクションでは、Zabbix を設定するためのベストプラクティスについて説明します。

これらのプラクティスは Zabbix の動作に必須ではありませんが、最適で安全な使用エクスペリエンスを実現するために実装することを強くお勧めします。

1 セキュリティのベストプラクティス

概要

このセクションでは、Zabbix を安全にセットアップするためのベストプラクティスを説明しています。

Zabbix の機能には必須ではありませんが、システムのセキュリティを向上させるために推奨されます。

UTF-8 エンコーディング

Zabbix がサポートするエンコーディングは UTF-8 のみです。セキュリティ上の欠陥なく動作することが知られています。ユーザーは他のエンコーディングを使用する場合、既知のセキュリティ上の問題があることを留意してください。

Windows インストーラーのパス　Windows インストーラーを使用する場合は、インストーラーによって提供されるデフォルトのパスを使用することをお勧めします。適切なアクセス許可なしでカスタムパスを使用すると、インストールのセキュリティが損なわれる可能性があります。

Zabbix セキュリティアドバイザリと CVE データベース

*Zabbix セキュリティアドバイザリと CVE データベース*を参照してください。

1 アクセス制御

概要

このセクションでは、アクセス制御を安全に設定するためのベストプラクティスについて説明します。

最小権限の原則

ユーザーアカウントは常に、できるだけ少ない権限で実行する必要があります。これは、Zabbix Web インターフェースのユーザーアカウント、データベースユーザー、Zabbix サーバー/プロキシ/エージェントのプロセスユーザーには、目的の機能を実行するために必要な権限のみが与えられる必要があることを意味します。

Attention:
'zabbix' ユーザーに追加の権限を与えると、設定ファイルにアクセスし、インフラストラクチャのセキュリティを侵害する可能性のある操作を実行できるようになります。

ユーザーアカウントの権限を設定するときは、Zabbix Web インターフェースのユーザータイプを考慮する必要があります。*Admin* ユーザータイプは、*Super Admin* ユーザータイプより権限が少ないにもかかわらず、設定を管理し、カスタムスクリプトを実行できることに注意してください。

Note:
一部の情報は、権限を持たないユーザーでも利用できます。たとえば、アラート → スクリプトは *Super Admin* ユーザーのみが利用できますが、スクリプトは Zabbix API を通じて取得することもできます。この場合、スクリプトのアクセス許可を制限し、スクリプトから機密情報 (アクセス資格情報など) を除外すると、グローバルスクリプトで利用可能な機密情報の公開を避けることができます。

Zabbix エージェントのセキュアユーザー

デフォルトでは、Zabbix サーバー、プロキシ、およびエージェント（またはエージェント 2）のプロセスは 1 つの `zabbix` ユーザーを共有します。Zabbix エージェント/エージェント 2（サーバー/プロキシと同じマシンで実行されている）がサーバー/プロキシの設定内の機密情報（例：データベース認証情報）にアクセスするのを防ぐために、エージェントは別のユーザーで実行する必要があります。

Zabbix エージェントの場合：

1. セキュアなグループとユーザー（例：`zabbix-agent`）を作成します。
2. このユーザーをエージェントの設定ファイルのUserパラメータに設定します。
3. エージェントを再起動して、新しいユーザーに権限を落とします。

Zabbix エージェント 2 の場合、エージェント 2 の設定ファイルは `User` パラメータをサポートしていないため、サービスレベルで設定を適用する必要があります。例については、ZBX-26442を参照してください。

SSL 設定への書き込みアクセス権を取り消す (Windows)

Zabbix エージェントを Windows 上でコンパイルし、OpenSSL が保護されていないディレクトリ (例: `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static`, `C:\dev\openssl`) にある場合は、このディレクトリへの書き込みアクセス権を管理者以外のユーザーから必ず取り消してください。そうしないと、エージェントが特権のないユーザーによって変更可能なパスから SSL 設定を読み込むことになり、潜在的なセキュリティ脆弱性が発生します。

Zabbix コンポーネントのセキュリティの強化

Zabbix コンポーネントのセキュリティを強化するために、一部の機能をオフにすることができます。

- Zabbix サーバーでのグローバルスクリプトの実行は、サーバー設定で EnableGlobalScripts=0 を設定することで無効にできます。
- Zabbix プロキシでのグローバルスクリプトの実行は、デフォルトで無効になっています (プロキシ設定で EnableRemoteCommands=1 を設定することで有効にできます)。
- Zabbix エージェントでのグローバルスクリプトの実行は、デフォルトで無効になっています (エージェント設定で許可されている各コマンドに対して、AllowKey=system.run[<command>,*] パラメータを追加することで有効にできます)。

- ユーザーの HTTP 認証は、フロントエンド設定ファイル (zabbix.conf.php) で `$ALLOW_HTTP_AUTH=false` を設定することで無効にできます。フロントエンドを再インストールすると (setup.php を実行すると)、このパラメータが削除されることに注意してください。

1 MySQL/MariaDB のセキュリティ保護

概要

このセクションでは、MySQL/MariaDB データベースを保護するためのベストプラクティスについて説明します。

> **Note:**
> 基本的なセットアップについては、デフォルトのMySQL/MariaDB データベースの作成手順を参照してください。この手順には、Zabbix データベースに対する完全な権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーは、Zabbix をアップグレードするときにデータベース構造を変更するために必要な権限も持つデータベース所有者です。

セキュリティを強化するには、最小限の権限を持つ追加のデータベースロールとユーザーを作成することをお勧めします。これらのロールとユーザーは、最小権限の原則に基づいて、つまり、意図した機能を実行するために必要な権限のみを設定する必要があります。

ユーザーロールの作成

対応する権限を持つ次のロールを作成します。

- **zbx_srv** - Zabbix サーバーとプロキシを実行するためのロール

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - Zabbix フロントエンドと API を実行するためのロール。

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - テーブルバックアップのロール

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

> **Note:**
> テーブルの復元とアップグレードはデータベース所有者が実行する必要があります。

- **zbx_part** - データベースのパーティション化に対する権限セットが制限されたロール。このロールは特定のデータベーステーブルに対する権限を付与するため、データベースの作成後にのみ作成できることに注意してください。

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

ロールを作成した後、ユーザーに割り当てることができます。

ユーザーロールの割り当て

To assign the created user roles, create users and assign the relevant roles to them. 作成したユーザーロールを割り当てるには、ユーザーを作成し、関連するロールをユーザーに割り当てます。<user>、<host>、<role>、<password> は必要に応じて置き換えてください。

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- For MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

たとえば、Zabbix サーバーとプロキシを実行するためのロールを作成して割り当てるには、次のようにします。

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 PostgreSQL/TimescaleDB のセキュリティ保護

概要

このセクションでは、PostgreSQL データベースを保護するためのベストプラクティスについて説明します。

Note:
基本的なセットアップについては、デフォルトのPostgreSQL データベースの作成手順を参照してください。この手順には、Zabbix データベースに対する完全な権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーは、Zabbix をアップグレードするときにデータベース構造を変更するために必要な権限も持つデータベース所有者です。

セキュリティを強化するには、安全なスキーマ使用パターンを構成し、最小限の権限を持つ追加のデータベースロールとユーザーを作成することをお勧めします。これらのロールとユーザーは、最小権限の原則((/manual/best_practices/security/access_control#principle-of-least-privilege) に基づいて、つまり、意図した機能を実行するために必要な権限のみを設定する必要があります。

データベースのセットアップ

データベース所有者となるユーザーを作成し、Zabbix データベースを作成します。データベースの所有者は、データベースの作成で指定されているユーザーです。

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

Note:
データベースのクリーンなインストールまたはアップグレードは、データベース所有者が実行する必要があります。これは、データベースオブジェクトをドロップする、あるいはその定義を変更する権限がデータベース所有者に固有の特権であり、許可または取り消されることができないためです。

Attention:
このページの次のコマンドは、PostgreSQL への接続が zabbix データベースに作成されている間に実行する必要があります。

zabbix スキーマを作成し、データベースの所有者 (usr_owner) をこのスキーマの所有者に設定します。

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

安全なスキーマ使用パターンを設定します。

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

データベースを設定した後、ユーザーロールの作成に進みます。

ユーザーロールの作成

対応する権限を持つ次の役割を作成します。

- **zbx_srv** - Zabbix サーバーとプロキシの実行の役割:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - Zabbix フロントエンドと API を実行するための役割:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - テーブルバックアップの役割:

```
CREATE ROLE zbx_bckp;
GRANT CONNECT ON DATABASE zabbix TO zbx_bckp;
GRANT USAGE ON SCHEMA zabbix TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT ON TABLES TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

> **Note:**
> テーブルの復元は、データベースの所有者によってのみ可能です。

役割を作成した後、ユーザーに割り当てることができます。

ユーザーロールの割り当て

作成されたユーザーロールを割り当てるには、ユーザーを作成し関連するロールを割り当てます。<user> と <role>、<password> は適切に置き換えてください。

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

たとえば、Zabbix サーバーとプロキシを実行するロールを作成して割り当てるには、

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

TimescaleDB を使用したデータベースパーティショニング

データベースパーティショニングは、TimescaleDB によって促進されます。TimescaleDB を利用するには、Zabbix サーバーにはデータベース所有者の権限が必要です。

PostgreSQL zabbix スキーマが既に zabbix データベースを作成していた場合、次のコマンドで TimescaleDB を有効にすることができます。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

2 暗号化

概要

このセクションでは、暗号化を安全に設定するためのベストプラクティスについて説明します。

Zabbix の Web インターフェースの SSL のセットアップ

RHEL ベースのシステムでは、mod_ssl パッケージをインストールします:

```
dnf install mod_ssl
```

SSL 鍵用のディレクトリを作成します:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

SSL 証明書を作成します:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

プロンプトに適切に入力します。最も重要な行は、Common Name を指定する行です。サーバーに関連付けるドメイン名を入力する必要があります。ドメイン名がない場合は、代わりにパブリック IP アドレスを入力することができます。

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Apache SSL 設定ファイル (/etc/httpd/conf.d/ssl.conf) を編集します:

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

変更を反映させるため Apache のサービスを再起動します:

```
systemctl restart httpd.service
```

3 Web サーバー

概要

このセクションでは、Web サーバーを安全に設定するためのベストプラクティスについて説明します。

URL のルートディレクトリで Zabbix を有効にする

RHEL ベースのシステムでは、仮想ホストを Apache 構成 (/etc/httpd/conf/httpd.conf) に追加し、ドキュメントルートの永続的なリダイレクトを Zabbix SSL URL に設定します。example.com は実際のサーバー名に置き換えてください。

```
#### Add lines:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

変更を適用するため、Apache サービスを再起動します。

```
systemctl restart httpd.service
```

Web サーバーで HTTP Strict Transport Security (HSTS) を有効にする

Zabbix Web コンソールをプロトコルダウングレード攻撃から保護するには、Web サーバーでHSTSポリシーを有効にすることをお勧めします。

Apache 構成で Zabbix フロントエンドの HSTS ポリシーを有効にするには、次の手順に従います。

1. 仮想ホストの構成ファイルを見つけます。

- RHEL ベースのシステムの場合、/etc/httpd/conf/httpd.conf
- Debian/Ubuntu の場合、/etc/apache2/sites-available/000-default.conf

2. 次のディレクティブを仮想ホストの設定ファイルに追加します。

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. 変更を適用させるため、Apache サービスを再起動します。

```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Zabbix でセキュアなセッションクッキーと SameSite セッションクッキーを強制する

Zabbix を構成する場合、セキュリティを強化し、クロスサイトリクエストフォージェリ (CSRF) 攻撃を防ぐために、セッションクッキーにセキュアな属性と SameSite 属性を強制することが不可欠です。ただし、SameSite=Strict を強制すると、次のような特定のシナリオで問題が発生する可能性があります。

- 同じドメインの iframe を埋め込むと、ダッシュボード URL ウィジェットに" ユーザーはログインしていません" と表示される。
- HTTPS ではなく HTTP 経由でダッシュボードにアクセスするユーザーは、ログインの問題に直面する可能性がある。
- 特定の Zabbix メニューセクションまたはホストへの URL を共有できない。

これらの問題を軽減するには、ユーザーが SameSite ポリシーを調整する方法が必要です。

1. セキュアなクッキー

secure フラグを設定すると、クッキーが HTTPS 経由でのみ送信されるようになり、暗号化されていない接続での露出が防止されます。

Zabbix でセキュアクッキーを有効にするには、Web サーバー構成で次の設定を追加または変更します:

Apache の場合:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Nginx の場合:

```
proxy_cookie_path / "/; Secure";
```

Zabbix フロントエンドが HTTPS 経由でアクセスされていることを確認します。そうでない場合、Secure フラグ付きのクッキーは送信されません。

2. SameSite 属性の構成

Web サーバー設定で SameSite 属性を適用することもできます:

Apache の場合:

```
<IfModule mod_headers.c>
Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Nginx(バージョン 1.19.3+) の場合:

```
proxy_cookie_flags ~ samesite=Strict; # 特異性のために~を'zbx_session'に置き換えます
```

Web サーバーでコンテンツセキュリティポリシー (CSP) を有効にする

Zabbix Web インターフェースをクロスサイトスクリプティング (XSS)、データインジェクション、および同様の攻撃から保護するには、Web サーバーでコンテンツセキュリティポリシーを有効にすることをお勧めします。これを行うには、HTTP ヘッダーを返すように Web サーバーを設定します。

Attention:
次の CSP ヘッダー設定は、デフォルトの Zabbix Web インターフェースインストールおよびすべてのコンテンツがサイトのドメイン (サブドメインを除く) から発信されている場合にのみ適用されます。たとえば、サイトのサブドメインまたは外部ドメインのコンテンツを表示するように*URL* ウィジェットを構成している場合、別の CSP ヘッダー構成が必要になる場合があります。*OpenStreetMap* を別のマップエンジンに追加するか、外部 CSS またはウィジェットを追加します。Duo Universal Prompt 多要素認証方式を使用している場合は、仮想ホストの構成ファイルの CSP ディレクティブに"duo.com" を必ず追加してください。

Apache 構成で Zabbix Web インターフェースの CSP を有効にするには、次の手順に従います。

1. 仮想ホストの設定ファイルを見つけます。

- RHEL ベースのシステムの場合、/etc/httpd/conf/httpd.conf
- Debian/Ubuntu の場合、/etc/apache2/sites-available/000-default.conf

2. 次のディレクティブを仮想ホストの設定ファイルに追加します。

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. 変更を適用させるため、Apache サービスを再起動します。

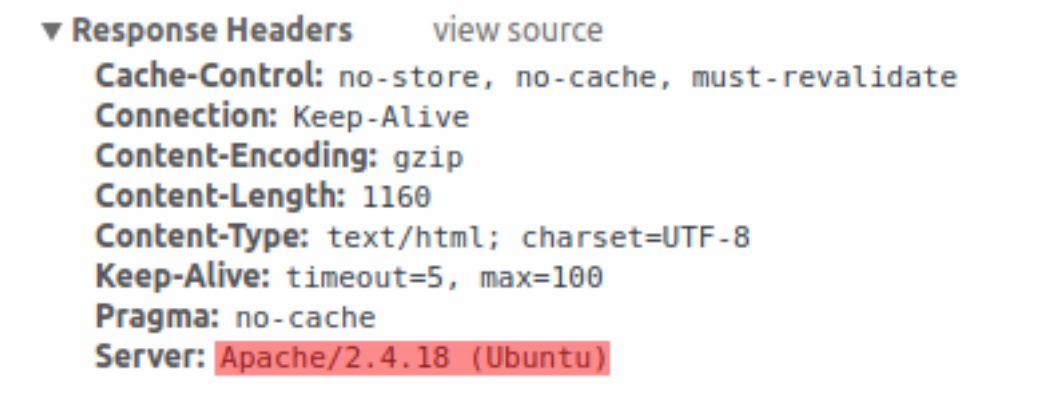
```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Web サーバー情報の公開を無効にする

セキュリティを向上させるために、すべての Web サーバーの署名を無効にすることをお勧めします。

デフォルトでは、Web サーバーはソフトウェア署名を公開します。



署名を無効にするには、Apache 構成ファイルに次のパラメータを追加します。

```
ServerSignature Off
ServerTokens Prod
```

PHP 署名 (X-Powered-By HTTP ヘッダー) は、`php.ini` 設定ファイルを変更することで無効にできます (デフォルトでは、署名は無効になっています)。

```
expose_php = Off
```

構成ファイルの変更を適用するには、Web サーバーの再起動が必要です。

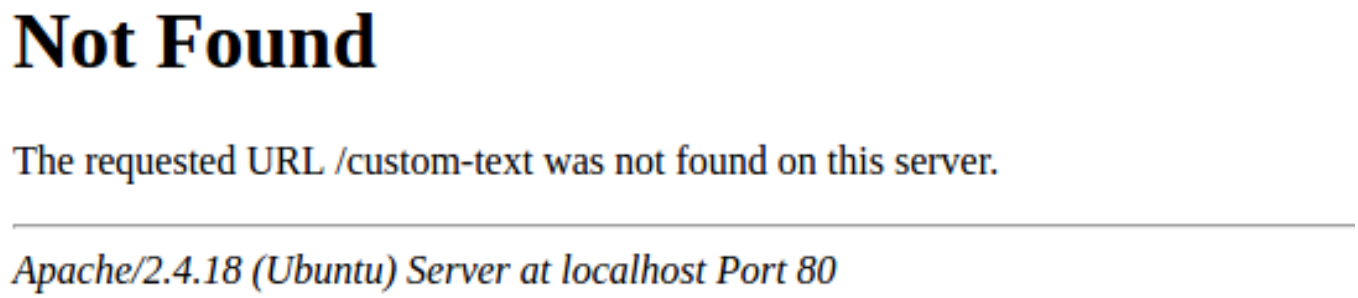
セキュリティを強化するには、Apache で *mod_security* ツール (パッケージ *libapache2-mod-security2*) を使用できます。このツールを使用すると、サーバー署名からバージョンのみを削除するのではなくサーバー署名を削除できます。*mod_security* をインストールした後、"SecServerSignature" を任意の値に設定することで、サーバー署名を任意の値に変更できます。

ソフトウェア署名を削除/変更する方法については、Web サーバーのドキュメントを参照してください。

Web サーバーのデフォルトエラーページを無効にする

情報漏洩を避けるため、デフォルトのエラーページを無効にすることをお勧めします。

デフォルトでは、Web サーバーは組み込みのエラーページを使用します。



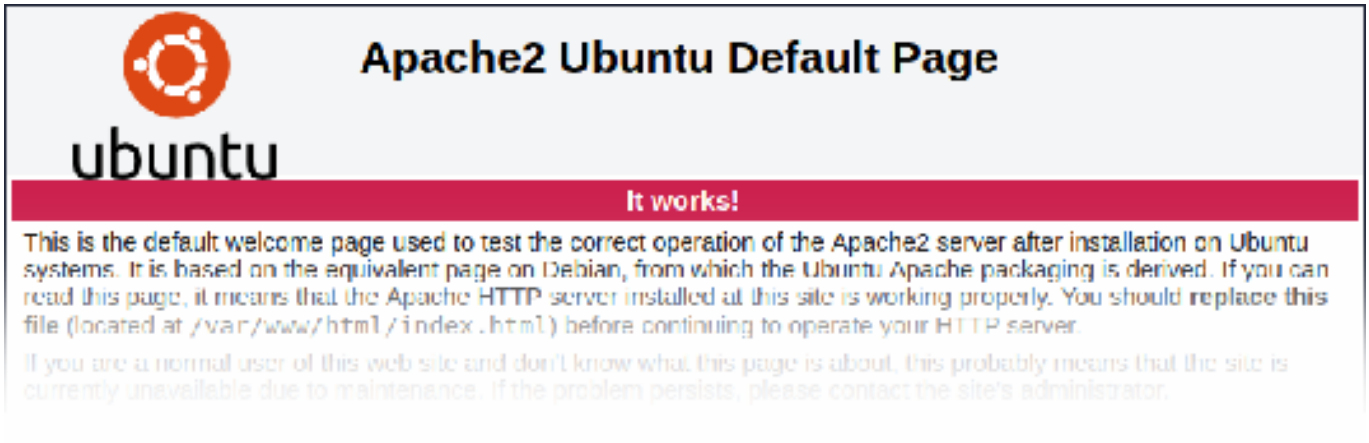
これらのデフォルトのエラーページは置き換えるか削除する必要があります。たとえば、"ErrorDocument" ディレクティブを使用して、Apache Web サーバーのカスタムエラーページ/テキストを定義できます。

デフォルトのエラーページを置き換えたり削除したりする方法については、Web サーバーのドキュメントを参照してください。

Web サーバーのテストページを削除する

情報漏洩を避けるため、Web サーバのテストページを削除することをお勧めします。

By default, the Apache web server webroot contains the `index.html` test page: デフォルトでは、Apache Web サーバーの Webroot には `index.html` テストページが含まれています。



デフォルトのテストページを削除する方法については、Web サーバーのドキュメントを参照してください。

X-Frame-Options HTTP 応答ヘッダーを設定する

デフォルトでは、Zabbix は *Use X-Frame-Options HTTP header* パラメーターが `SAMEORIGIN` に設定されて構成されています。つまり、ページ自体と同じオリジンを持つフレームにのみコンテンツをロードできます。

外部 URL からコンテンツをプルする Zabbix フロントエンド要素 (つまり、URLダッシュボードウィジェット) は、すべてのサンドボックス制限が有効になった状態で、取得したコンテンツをサンドボックスに表示します。

これらの設定により、Zabbix フロントエンドのセキュリティが強化され、XSS およびクリックジャッキング攻撃に対する保護が提供されます。*Super admin* ユーザーは、必要に応じて *Use iframe sandboxing* および *Use X-Frame-Options HTTP header* パラメータを変更できます。デフォルト設定を変更する前に、リスクとメリットを慎重に比較検討してください。iframe サンドボックスまたは X-Frame-Options HTTP ヘッダーを完全にオフにすることはお勧めできません。

一般的なパスワードリストのファイルを非表示にする

パスワード総当たり攻撃の複雑さを増すために、`ui/data/top_passwords.txt` ファイルへのアクセスを制限することをお勧めします。このファイルには、最も一般的でコンテキスト固有のパスワードのリストが含まれており、ユーザーがそのようなパスワードを設定できないようにします (パスワードポリシーで 推測しやすいパスワードを避けるパラメーターが有効になっている場合)。

`top_passwords.txt` ファイルへのアクセスを制限するには、Web サーバーの構成を変更します。

Apache では、`.htaccess` ファイルを使用してファイルアクセスを制限できます。

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

NGINX では、`location` ディレクティブを使用してファイルアクセスを制限できます。

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 設定のベストプラクティス

概要

このセクションでは、Zabbix を最適なパフォーマンスと使いやすさで構成するためのベストプラクティスを説明します。これらの推奨事項は、Zabbix 開発者のアドバイスと、Zabbix トレーナーやサポートエンジニアの実践的な経験に基づいています。

すべての Zabbix インストールはユニークであり、これらのガイドラインの一部が特定の構成に適さない場合があります。ただし、一般的な潜在的な問題を回避するために、できるだけこれらのガイドラインに従うことをお勧めします。

Note:
このページが改善できると思われる場合は、ぜひご意見をお聞かせください！該当するテキストをハイライトし、**ctrl+Enter** を押して間違いを報告したり、フィードバックを共有したりしてください。

ホストとアイテム　ホストの定義

Zabbix におけるホストは、物理的なマシンやデバイスではなく、論理的なエンティティです。監視の目的で、データベースや仮想マシンなど、個別のホストを作成することができます。あるいは、*John* のノートパソコンのような汎用ホストを作成し、そのホストの下ですべてのメトリックを監視することもできます。

ベストプラクティスは、仮想マシン、データベース、コンテナ、ネットワークスイッチなど、独立したインスタンスごとに個別のホストを作成することです。このアプローチを利用することで、以下のメリットがあります。

1. 各ホストごとに個別のアイテム、トリガー、アラート通知を持つことで、監視データの混乱を避けることができます。
2. ユーザーアクセスレベルを細かく調整できます。ユーザーロールを設定して、特定のホストの閲覧や設定のみを許可することができます。最小権限の原則も参照してください。

重複したアイテムを持つホスト

Network switch 1 や *Network switch 2* のように、いくつかの類似したホストがある場合、Zabbix はホストを迅速に再作成するための複数の方法を提供します。Clone ボタンを押すだけで、すべてのメトリクスを持つホストを単純にクローンできますが、この場合、後でアイテムを更新するには、各ホストで手動で行う必要があります。

ベストプラクティスは、必要なすべてのメトリクスを持つテンプレート、たとえば *Network switch template* を作成することです。次に、類似したホストをホストグループにまとめます。上記の例では *Network switches* となります。これで、データ収集 -> ホストセクションでホストグループごとにすべてのホストをフィルタリングし、一括更新ボタンを使用してテンプレートをすべてのネットワークスイッチにリンクできます。

依存アイテム

ターゲットエンティティへのリクエスト数を最小限に抑えるために、Zabbix ではマスターアイテムと依存アイテムを作成することができます。この場合、マスターアイテムは 1 回のリクエストで大量の情報を収集します。その後、依存アイテムを設定して、前処理を通じてそのコレクションから特定のデータを抽出し、個別のメトリクスとして保存することができます。

例えば、マスターアイテムが複数のメトリクスを含む JSON や XML レスポンスを収集したり、複数のカラム（例：オープン接続数、中断された接続数、同時接続の最大許容数、起動後の累積接続数など）を返すデータベースクエリを実行したりし、依存アイテムがそれぞれ必要な値を個別に解析して保存します。

この構成のベストプラクティスは、収集直後にマスターアイテムの履歴を破棄し、依存アイテムのデータのみを保持することです。

サーバーとプロキシ

すべてのホストが Zabbix サーバーと同じローカルネットワーク内にあり、スケーラビリティやパフォーマンスの問題がなければ、プロキシは必要ないかもしれません。より大規模または複雑な環境では、Zabbix サーバーでホストを直接監視するだけでは不十分な場合があります。プロキシを追加し、一部のホストをそのプロキシに割り当てることで、より均等な負荷分散が可能になります。

ベストプラクティスとして、以下の場合に Zabbix プロキシを追加します:

1. ファイアウォールの背後で複数のメトリック収集方法を使用して複数のホストを監視している場合。プロキシはホストからデータを収集し、それを Zabbix サーバーに転送するため、複数のファイアウォールポートを開く必要がなくなります。
2. リモート拠点、支店、および/またはネットワークを監視している場合。Zabbix サーバーとリモート拠点間のネットワークが中断された場合でも、リモート拠点に配備された Zabbix プロキシはデータ収集を継続し、ネットワーク接続が復旧した際に収集したデータを Zabbix サーバーに送信します。
3. 大規模な導入を行っており、Zabbix サーバーの負荷を軽減し、パフォーマンスを向上させたい場合。大規模な導入の定義は非常に幅広く、ホスト数だけでなく、1 秒あたりに収集される値の数にも依存します。

シークレットマクロ

シークレットユーザーマクロを、シークレットテキストまたはシークレットボールトマクロとして使用することができます。

シークレットボールトマクロを使用する場合、セキュリティを強化するために、Zabbix サーバーと Zabbix プロキシがそれぞれ独立してマクロ値を取得するように設定することを推奨します。デフォルトでは、シークレットマクロの値は Zabbix サーバーによって取得され、Zabbix プロキシに伝播されます。

18 API

概要　Zabbix API を使用すると、Zabbix の設定をプログラムで取得および変更でき、ヒストリデータへのアクセスが可能になります。次の目的で広く使用されています。

- Zabbix と連携する新しいアプリケーションを作成する
- Zabbix をサードパーティソフトウェアに統合する
- ルーチンタスクを自動化する

Zabbix API は HTTP ベースの API であり、Web フロントエンドの一部として発送されます。JSON-RPC 2.0 プロトコルを使用し、次の 2 つのことを意味します。

- API は一連の個別メソッドで構成されている
- クライアントと API 間の要求と応答は、JSON 形式を使用してエンコードされる

プロトコルと JSON の詳細については、JSON-RPC 2.0 仕様 と JSON 形式のホームページ を参照してください。

Zabbix 機能を Python アプリケーションに統合する方法の詳細については、Zabbix API の Python ライブラリzabbix_utilsを参照してください。

構造　API は、名目上、別々の API にグループ化された多数のメソッドから構成されています。それぞれのメソッドはある特定のタスクを実行します。例えば、`host.create` メソッドは *host* API に属し、新しいホストを作成に使用されます。歴史的には、API は" クラス" と呼ばれることもあります。

> **Note:**
> ほとんどの API は、少なくとも `get`, `create`, `update`, `delete` という 4 つのメソッドを持ち、それぞれデータの取得、作成、更新、削除を行います。しかし、一部の API では、全く別のメソッドを提供することもあります。

リクエストの実行　フロントエンドをセットアップしたら、リモート HTTP リクエストを使用して API を呼び出すことができます。そのためには、フロントエンドディレクトリにある `api_jsonrpc.php` ファイルに HTTP POST リクエストを送信する必要があります。たとえば、Zabbix フロントエンドが `https://example.com/zabbix` にインストールされている場合、`apiinfo.version` メソッドを呼び出す HTTP リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

リクエストには、`Content-Type` ヘッダーを次のいずれかの値に設定する必要があります: `application/json-rpc`、`application/json`、または `application/jsonrequest`。

リクエストオブジェクトには、次のプロパティを含める必要があります。

- `jsonrpc` - API で使用される JSON-RPC プロトコルのバージョン（Zabbix API は JSON-RPC バージョン 2.0 を実装しています）;
- `method` - 呼び出される API メソッド;
- `params` - API メソッドに渡されるパラメータ;
- `id` - リクエストの任意の識別子（省略した場合、API はリクエストを通知として扱います）。

リクエストが正しい場合、API から返されるレスポンスは次のようになります。

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

レスポンスオブジェクトには、次のプロパティが含まれます。

- `jsonrpc` - JSON-RPC プロトコルのバージョン;
- `result` - メソッドによって返されるデータ;
- `id` - 対応するリクエストの識別子。

ワークフローの例　次のセクションでは、いくつかの使用例を詳しく説明します。

認証　Zabbix のデータにアクセスするには、次のどちらかを行う必要があります。

- 既存のAPI トークンを使用 (Zabbix フロントエンドまたはトークン APIを使用して作成)
- user.loginメソッドで取得した認証トークンを使用。

たとえば、標準の *Admin* ユーザーとしてログインして新しい認証トークンを取得したい場合、JSON リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

資格情報を正しく指定した場合、API から返されるレスポンスにはユーザー認証トークンが含まれているはずです。

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証方法　”Authorization” ヘッダーによる認証

すべての API リクエストには認証または API トークンが必要です。リクエストの Authorization ヘッダーを使用して認証情報を指定できます:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
認証に問題がある場合は、Authorization ヘッダーの転送を参照してください。

Zabbix API はヘッダーを大文字・小文字を区別せずに受け付けます（例: `authorization`、`Authorization`、`AUTHORIZATION` は同じものとして扱われます）。

Authorization ヘッダーはクロスオリジンリクエスト (CORS) でもサポートされています。

Zabbix クッキー

”zbx_session” クッキーは、(モジュールまたはカスタムウィジェットから) JavaScript を使用して実行される Zabbix UI からの API リクエストを承認するために使用されます。

ホストの取得　これで、Zabbix のデータにアクセスするために使用できる有効なユーザー認証トークンが完成しました。たとえば、host.getメソッドを使用して、構成されているすべてのhostsの ID、ホスト名、およびインターフェイスを取得できます。

リクエスト:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data @data.json
```

Note:
`data.json` は、JSON クエリを含むファイルです。ファイルの代わりにクエリを`--data` 引数で渡すことができます。

data.json

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
```

```
    "id": 2
}
```

レスポンスオブジェクトには、要求されたホストに関するデータが含まれます。

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
パフォーマンス上の理由から、取得するオブジェクトプロパティを常にリストアップし、すべてのオブジェクトプロパティを取得しないようにすることをお勧めします。

新しいアイテムの作成　先ほどの `host.get` リクエストで取得したデータを使って、"Zabbix サーバー" に新しいアイテムを作成してみましょう。これはitem.createメソッドを使用して実行できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

作成が成功した場合のレスポンスには、新しく作成されたアイテムの ID が含まれます。これは、以降のリクエストでアイテムを参照するために使用できます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
なお `item.create` メソッドや他の *create* メソッドも、オブジェクトの配列を受け入れて、1 回の API 呼び出しで複数のアイテムを作成することもできます。

Creating multiple triggers　したがって、*create* メソッドが配列を受け入れる場合、次のように複数のトリガーを追加することも可能です。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

作成が成功した場合のレスポンスには、新しく作成したトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

アイテムの更新　アイテムのステータスを"0" に設定することで、アイテムを有効にします。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

成功したレスポンスには、更新されたアイテムの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
item.update メソッドやその他の *update* メソッドは、オブジェクトの配列を受け入れ、1 回の API 呼び出しで複数のアイテムを更新することができます。

複数のトリガーの更新　ステータスを"0" に設定して、複数のトリガーを有効にします。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

成功したレスポンスには、更新されたトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Note:
これは推奨される更新方法です。host.massupdate などの一部の API メソッドを使用すると、より単純なコードを作成できます。ただし、これらのメソッドは将来のリリースで削除されるため、使用することはお勧めできません。

エラーハンドリング　今まで試したメソッドはすべて成功しました。しかし、API に対して間違った呼び出しを行おうとするとどうなるでしょうか？host.create を呼び出して別のホストを作成する際に、必須の groups パラメータを省略してみましょう。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

レスポンスにはエラーメッセージが含まれます。

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

エラーが発生した場合、レスポンスオブジェクトには `result` プロパティの代わりに、次のデータを持つ `error` プロパティが含まれます。

- code - エラーコード
- message - エラーの概要
- data - 詳細なエラーメッセージ

エラーは、不正な入力値の使用、セッションタイムアウト、存在しないオブジェクトへのアクセスの試行など、さまざまなケースで発生する可能性があります。アプリケーションは、この種のエラーを適切に処理できる必要があります。

API バージョン　API のバージョン管理を簡素化するために、Zabbix 2.0.4 以降、API のバージョンは Zabbix 自体のバージョンと一致していますが、apiinfo.versionメソッドを使用して、使用している API のバージョンを確認することもできます。これは、バージョン固有の機能を使用するようにアプリケーションを調整する場合に役立ちます。

Zabbix は、メジャーバージョン内では機能の下位互換性を保証します。メジャーリリース間で下位互換性のない変更を行う場合、通常は古い機能を次のリリースで非推奨として残し、その後のリリースで削除します。場合によっては、下位互換性を提供せずにメジャーリリース間で機能を削除することもあります。非推奨の機能には決して依存せず、できるだけ早く新しい代替機能に移行することが重要です。

> **Note:**
> API 変更ログ で、API に加えられたすべての変更を追うことができます。

参考文献　これで Zabbix API を使い始めるのに十分な知識が得られましたが、これが全てではありません。更に詳しく知りたい場合は、利用可能な API のリストを参照することをお勧めします。

メソッドリファレンス

このセクションでは、Zabbix API が提供する機能の概要を説明し、利用可能なクラスとメソッドの使い方を説明します。

監視　Zabbix API では、監視中に収集した履歴やその他のデータにアクセスすることができます。

ダッシュボード

ダッシュボードを管理し、それに基づいてスケジュールされたレポートを作成します。

Dashboard API | Template dashboard API | Report API

高可用性 (HA) クラスタ

サーバノードとそのステータスの一覧を取得します。

High availability cluster API

ヒストリ

Zabbix 監視プロセスによって収集された過去の値を取得し、表示やさらなる処理を行うことができます。

History API

トレンド

Zabbix サーバーで計算されたトレンドの値を取得し、表示やさらなる処理を行うことができます。

Trend API

イベント

トリガー、ネットワークディスカバリ、その他の Zabbix システムから生成されたイベントを取得し、より柔軟な状況管理やサードパーティツールの統合を行うことができます。

Event API

障害

指定されたパラメータに従って、障害を取得します。

Problem API

マップ

マップを設定して、IT インフラストラクチャの詳細な動的表示を作成します。

Map API

タスク

Zabbix サーバーのタスクマネージャと連動し、タスクの作成とレスポンスの取得を行います。

Task API

サービス　Zabbix API を使用すると、サービスの監視中に収集されたデータにアクセスできます。

サービスレベルアグリーメント (SLA)

サービスレベル目標（SLO）を定義し、サービスのパフォーマンスに関する詳細なサービスレベル指標（SLI）情報を取得します。

SLA API

サービス

サービスレベルのモニタリングのためにサービスを管理し、あらゆるサービスの詳細な SLA 情報を取得することができます。

Service API

データ収集　Zabbix API では、監視システムの設定を管理することができます。

ホストとホストグループ

ホストグループとホスト、およびホストインタフェース、ホストマクロ、メンテナンス期間など、ホストに関連するすべてのものを管理します。

Host API | Host group API | Host interface API | User macro API | Value map API | Maintenance API

アイテム

監視するアイテムを定義します

Item API

トリガー

システムの問題を通知するトリガーを構成します。トリガーの依存関係を管理します。

Trigger API

グラフ

収集したデータを見やすくするために、グラフを編集したり、グラフの item を分けたりすることができます。

Graph API | Graph item API

テンプレートとテンプレートグループ

テンプレートを管理し、ホストまたは他のテンプレートにリンクします。

Template API | Template group API | Value map API

ローレベルディスカバリ

動的なエンティティを監視するために、ローレベルディスカバリルール、item、trigger、グラフのプロトタイプを設定します。

LLD rule API | Item prototype API | Trigger prototype API | Graph prototype API | Host prototype API

イベント相関

カスタムイベントの相関ルールを作成します。

Correlation API

ネットワークディスカバリ

ネットワークレベルの検出ルールを管理し、新しいホストを自動的に検出して監視します。検出されたサービスとホストに関する情報に完全にアクセスできます。

Discovery rule API | Discovery check API | Discovered host API | Discovered service API

エクスポートとインポート

設定のバックアップ、マイグレーション、大規模な設定更新のために Zabbix の設定データをエクスポート、インポートすることができます。

Configuration API

Web 監視

Web アプリケーションやサービスを監視するための Web シナリオを設定します。

Web scenario API

アラート　Zabbix API を使用すると、監視システムのアクションとアラートを管理できます。

アクションとアラート

特定のイベントをユーザーに通知したり、リモートコマンドを自動的に実行するためのアクションと操作を定義します。生成されたアラートとその受信者に関する情報にアクセスできます。

Action API | Alert API

メディアタイプ

メディアの種類と、ユーザーがアラートを受け取る複数の方法を構成します。

Media type API

スクリプト

日々の作業を支援するスクリプトを設定・実行します。

Script API

ユーザー　Zabbix API を使用すると、監視システムのユーザーを管理できます。

ユーザーとユーザーグループ

Zabbix にアクセスするユーザを追加し、ユーザグループに割り当て、権限を付与します。ユーザーの権限を細かく管理するための役割を作成します。

User API | User group API | User directory API | User role API

API トークン

認可トークンを管理します。

Token API

認証

認証の設定オプションを変更します。

Authentication API

管理画面　Zabbix API を使用すると、監視システムの管理設定を変更することができます。

一般

特定のグローバル設定オプションを変更します。

Autoregistration API | Icon map API | Image API | Settings API | Regular expression API | Module API | Connector API

監査ログ

各ユーザーが行った設定の変更を追跡します。

Audit log API

ハウスキーピング

ハウスキーピングの設定をします。

Housekeeping API

プロキシとプロキシグループ

分散監視設定で使用される Zabbix プロキシを管理します。

Proxy API | Proxy group API

マクロ

マクロを管理します。

User macro API

API 情報　Zabbix API のバージョンを取得し、アプリケーションでバージョン固有の機能を使用できるようにします。

API info API

Action

このクラスは、アクションを操作するためのものです。

オブジェクトリファレンス:

- Action
- Action operation
- Action operation message
- Action operation condition
- Action recovery operation
- Action update operation
- Action filter
- Action filter condition

利用可能なメソッド:

- action.create - 新しいアクションを作成する
- action.delete - アクションを削除する
- action.get - アクションを取得する
- action.update - アクションを更新する

Action オブジェクト

次のオブジェクトは、`action`API に直接関連しています。

Action

action オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
actionid	string	アクションの ID。 プロパティの動作: - 読取専用 - 更新操作の場合、必須
esc_period	string	デフォルトの操作ステップ期間。60 秒以上でなければなりません。秒、サフィックス付きのタイムユニット、およびユーザーマクロを受け入れます。 プロパティの動作: - `eventsource` が” トリガーによって作成されたイベント”、” 内部イベント” または” サービスステータス更新時に作成されたイベント” に設定されている場合、サポート

プロパティ	タイプ	説明
eventsource	integer	アクションが処理するイベントの種類。 サポートされているイベントタイプのリストについては、event の”source” プロパティを参照してください。 プロパティの動作: - 定数 - 作成操作の場合必須
name	string	アクション名。 プロパティの動作: - 作成操作の場合必須
status	integer	アクションが有効か無効か。 有効な値: 0 - *(デフォルト)* 有効 1 - 無効
pause_symptoms	integer	イベントが副次的なイベントの場合にエスカレーションを一時停止するか。 有効な値: 0 - 一時停止しない 1 - *(デフォルト)* 一時停止する プロパティの動作: - `eventsource` が”トリガーによって作成されたイベント” に設定されている場合サポート
pause_suppressed	integer	メンテナンス期間中にエスカレーションを一時停止するか。 有効な値: 0 - 一時停止しない 1 - *(デフォルト)* 一時停止する プロパティの動作: - `eventsource` が”トリガーによって作成されたイベント” に設定されている場合サポート
notify_if_canceled	integer	エスカレーションがキャンセルされたときに通知するか。 有効な値: 0 - 通知しない 1 - *(デフォルト)* 通知する プロパティの動作: - `eventsource` が”トリガーによって作成されたイベント” に設定されている場合サポート

アクションオペレーション

アクションオペレーションオブジェクトは、アクションが実行されたときに実行されるオペレーションを定義します。以下のプロパティを持ちます。

プロパティ	型	説明
operationtype	integer	オペレーションのタイプ。 可能な値： 0 - メッセージ送信; 1 - グローバルスクリプト; 2 - ホスト追加; 3 - ホスト削除; 4 - ホストグループに追加; 5 - ホストグループから削除; 6 - テンプレートリンク; 7 - テンプレートリンク解除; 8 - ホスト有効化; 9 - ホスト無効化; 10 - ホストインベントリモード設定; 13 - ホストタグ追加; 14 - ホストタグ削除。 アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の可能な値： 0 - 「メッセージ送信」; 1 - 「グローバルスクリプト」。 アクションオブジェクトの `eventsource` が「内部イベント」に設定されている場合の可能な値： 0 - 「メッセージ送信」。 プロパティの動作： - 必須
esc_period	string	エスカレーションステップの継続時間（秒）。60 秒より大きい必要があります。秒数、サフィックス付きの時間単位、またはユーザーマクロを受け付けます。0 または 0s に設定すると、デフォルトのアクションエスカレーション期間が使用されます。 デフォルト：0s。 プロパティの動作： - アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合にサポート
esc_step_from	integer	エスカレーションを開始するステップ。 デフォルト：1。 プロパティの動作： - アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合にサポート
esc_step_to	integer	エスカレーションを終了するステップ。 デフォルト：1。 プロパティの動作： - アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合にサポート
evaltype	integer	オペレーション条件の評価方法。 可能な値： 0 - *(デフォルト)* And/Or; 1 - And; 2 - Or。

プロパティ	型	説明
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトには `scriptid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「グローバルスクリプト」に設定されている場合は必須
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループには `groupid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「グローバルスクリプト」に設定され、`opcommand_hst` が設定されていない場合は必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストには `hostid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「グローバルスクリプト」に設定され、`opcommand_grp` が設定されていない場合は必須
opconditions	array	トリガーアクションに使用されるオペレーション条件。 オペレーション条件オブジェクトは以下で詳細に説明されています。
opgroup	array	ホストを追加するホストグループ。 ホストグループには `groupid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「ホストグループに追加」または「ホストグループから削除」に設定されている場合は必須
opmessage	object	オペレーションによって送信されるメッセージに関するデータを含むオブジェクト。 オペレーションメッセージオブジェクトは以下で詳細に説明されています。 プロパティの動作： - `operationtype` が「メッセージ送信」に設定されている場合は必須
opmessage_grp	array	メッセージを送信するユーザーグループ。 ユーザーグループには `usrgrpid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「メッセージ送信」に設定され、`opmessage_usr` が設定されていない場合は必須
opmessage_usr	array	メッセージを送信するユーザー。 ユーザーには `userid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「メッセージ送信」に設定され、`opmessage_grp` が設定されていない場合は必須

プロパティ	型	説明
optemplate	array	ホストにリンクするテンプレート。 テンプレートには `templateid` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「テンプレートリンク」または「テンプレートリンク解除」に設定されている場合は必須
opinventory	object	ホストに設定するインベントリモード。 インベントリには `inventory_mode` プロパティが定義されている必要があります。 プロパティの動作： - `operationtype` が「ホストインベントリモード設定」に設定されている場合は必須
optag	array	追加または削除するホストタグ。 タグには `tag` プロパティが定義されている必要があります。 `value` プロパティはオプションです。 プロパティの動作： - `operationtype` が「ホストタグ追加」または「ホストタグ削除」に設定されている場合にサポート。

Action operation message

operation message オブジェクトには、オペレーションによって送信されるメッセージに関するデータが含まれています。以下のような性質を持っています。

プロパティ	タイプ	説明
default_msg	integer	デフォルトのアクションメッセージテキストと件名を使用するかどうか。 有効な値: 0 - オペレーションからのデータを使用 1 - *(デフォルト)* メディアタイプからのデータを使用
mediatypeid	ID	メッセージの送信に使用されるメディアタイプの ID。 プロパティの動作: - Action operation オブジェクト、Action recovery operation オブジェクト、またはAction update operation オブジェクトの `operationtype` が”send message” に設定されている場合、あるいはAction update operation オブジェクトの `operationtype` が”notify all involved” に設定されている場合、サポート
message	string	オペレーションメッセージ本文。 プロパティの動作: - `default_msg` が”use the data from the operation” に設定されている場合、サポート
subject	string	オペレーションメッセージ件名。 プロパティの動作: - `default_msg` が”use the data from the operation” に設定されている場合、サポート

Action operation condition

Action operation condition オブジェクトは、現在の操作を実行するために満たさなければならない条件を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
conditiontype	integer	条件のタイプ。 有効な値: 14 - イベントが確認された プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - 必須
operator	integer	条件演算子。 有効な値: 0 - *(デフォルト)* =

次の演算子と値は、各操作条件タイプでサポートされています。

条件	条件名	対応演算子	期待値
14	イベント承認	=	イベントが承認されたか。 可能な値: 0 - 承認されていない 1 - 承認されている

Action recovery operation

Action recovery operation オブジェクトは、障害が復旧されたときに実行される操作を定義します。リカバリオペレーションはトリガー、内部、およびサービスアクションのみに可能です。次のプロパティがあります。

プロパティ	タイプ	説明
operationtype	integer	オペレーションタイプ。 Action オブジェクトの `eventsource` が”event created by a trigger” または”event created on service status update” に設定されている場合、 有効な値: 0 - メッセージを送信 1 - グローバルスクリプト 11 - 関係者全員に通知 Action オブジェクトの `eventsource` が”internal event” に設定されている場合、有効な値: 0 - メッセージを送信 11 - 関係者全員に通知 プロパティの動作: - 必須
opcommand	object	実行されるグローバルスクリプト。 グローバルスクリプトは `scriptid` プロパティが定義されている必要があります。 プロパティの動作: - `operationtype` に”global script” が設定されている場合、必須

プロパティ	タイプ	説明
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループは `groupid` プロパティが定義されている必要があります。 プロパティの動作: - Action オブジェクトの `eventsource` が”event created by a trigger”、`operationtype` が”global script” に設定されており、`opcommand_hst` には何も設定されていない場合、必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストは `hostid` プロパティが定義されている必要があります。 プロパティの動作: - Action オブジェクトの `eventsource` が”event created by a trigger”、`operationtype` が”global script” に設定されており、`opcommand_hst` には何も設定されていない場合、必須
opmessage	object	リカバリオペレーションによって送信されたメッセージに関するデータを含むオブジェクト。 操作メッセージオブジェクトは上記で詳しく説明しています。 プロパティの動作: - `operationtype` に”send message” が設定されている場合、必須
opmessage_grp	array	メッセージを送信するユーザーグループ。 ユーザーグループは `usrgrpid` プロパティが定義されている必要があります プロパティの動作: - `operationtype` に”send message” が設定されており、`opmessage_usr` には何も設定されていない場合、必須
opmessage_usr	array	メッセージを送信するユーザー。 ユーザーは `userid` プロパティが定義されている必要があります。 プロパティの動作: - `operationtype` に”send message” が設定されており、`opmessage_grp` には何も設定されていない場合、必須

Action update operation

アクション更新操作オブジェクトは、問題が更新された際に実行される操作（コメントの追加、確認、重大度の変更、または手動でクローズ）を定義します。更新操作は、トリガーアクションとサービスアクションに対してのみ可能です。以下のプロパティがあります。

プロパティ	タイプ	説明
操作タイプ	整数	操作のタイプ 有効な値: 0 - メッセージを送信する。 1 - グローバルスクリプトを実行する。 12 - 関係者全員に通知する。 プロパティの動作: - 必須

プロパティ	タイプ	説明
opcommand	オブジェクト	実行するグローバルスクリプト グローバルスクリプトには、scriptid プロパティが定義されている必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定されている場合は必須です。
opcommand_grp	配列	グローバルスクリプトを実行するホストグループ ホストグループには、groupid プロパティが定義されている必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、opcommand_hst が設定されていない場合は 必須です。
opcommand_hst	array	グローバルスクリプトを実行するホスト ホストには hostid プロパティが定義されている必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、opcommand_grp が設定されていない場合は 必須です。
opmessage	object	更新操作によって送信されたメッセージに関するデータを含むオブジェクト 操作メッセージオブジェクトについては、上記で詳しく説明しています。
opmessage_grp	array	メッセージを送信するユーザーグループ ユーザーグループには usrgrpid プロパティが定義されている必要があります。 プロパティの動作: - operationtype が”send message” に設定され、opmessage_usr が設定されていない場合は 必須です。
opmessage_usr	array	メッセージを送信するユーザー ユーザーには userid プロパティが定義されている必要があります。 プロパティの動作: - operationtype が”send message” に設定され、opmessage_grp が設定されていない場合は 必須です。

アクションフィルター

アクションフィルターオブジェクトは、設定されたアクション操作を実行するために満たす必要がある条件のセットを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
conditions	array	結果をフィルタリングするために使用するフィルター条件のセット。条件は、式内での配置順に並べ替えられます。 プロパティの動作: - 必須

プロパティ	タイプ	説明
evaltype	integer	フィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	フィルター条件を評価するために使用される生成された式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用
formula	string	カスタム式のフィルターの条件を評価するために使用されるユーザー定義の式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、フィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

アクションフィルター条件

アクションフィルター条件オブジェクトは、アクション操作を実行する前にチェックする必要がある特定の条件を定義します。

プロパティ	タイプ	説明
conditiontype	integer	条件のタイプ。 アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」に設定されている場合の値: 0 - ホストグループ; 1 - ホスト; 2 - トリガー; 3 - イベント名; 4 - トリガーの深刻度; 6 - 時間帯; 13 - ホストテンプレート; 16 - 問題が抑制されている; 25 - イベントタグ; 26 - イベントタグ値。 アクションオブジェクトの `eventsource` が「ディスカバリールールによって作成されたイベント」に設定されている場合の値: 7 - ホスト IP; 8 - 検出されたサービスのタイプ; 9 - 検出されたサービスのポート; 10 - ディスカバリーステータス; 11 - アップタイムまたはダウンタイムの期間; 12 - 受信値; 18 - ディスカバリールール; 19 - ディスカバリーチェック; 20 - プロキシ; 21 - ディスカバリーオブジェクト。 アクションオブジェクトの `eventsource` が「アクティブエージェントの自動登録によって作成されたイベント」に設定されている場合の値: 20 - プロキシ; 22 - ホスト名; 24 - ホストメタデータ。 アクションオブジェクトの `eventsource` が「内部イベント」に設定されている場合の値: 0 - ホストグループ; 1 - ホスト; 13 - ホストテンプレート; 23 - イベントタイプ; 25 - イベントタグ; 26 - イベントタグ値。 アクションオブジェクトの `eventsource` が「サービスステータスの更新時に作成されたイベント」に設定されている場合の値: 25 - イベントタグ; 26 - イベントタグ値; 27 - サービス; 28 - サービス名。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - 必須

プロパティ	タイプ	説明
value2	string	比較する 2 番目の値。 プロパティの動作: - アクションオブジェクトの `eventsource` が「トリガーによって作成されたイベント」に設定されており、`conditiontype` がトリガーアクションのいずれかの値に設定されていて、条件タイプ（下記参照）が「26」の場合は必須 - アクションオブジェクトの `eventsource` が「内部イベント」に設定されており、`conditiontype` が内部アクションのいずれかの値に設定されていて、条件タイプ（下記参照）が「26」の場合は必須 - アクションオブジェクトの `eventsource` が「サービスステータスの更新時に作成されたイベント」に設定されており、`conditiontype` がサービスアクションのいずれかの値に設定されていて、条件タイプ（下記参照）が「26」の場合は必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみ使用可能。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後リクエスト時には新たに生成されます。
operator	integer	条件演算子。 可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 4 - 範囲内; 5 - 以上; 6 - 以下; 7 - 範囲外; 8 - 正規表現一致; 9 - 正規表現不一致; 10 - はい; 11 - いいえ。

Note:
さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには、action.getおよびaction.createメソッドページの例を参照してください。

各条件タイプでサポートされる演算子と値は以下の通りです。

条件	条件名	サポートされる演算子	期待される値
0	ホストグループ	等しい, 等しくない	ホストグループ ID。
1	ホスト	等しい, 等しくない	ホスト ID。
2	トリガー	等しい, 等しくない	トリガー ID。
3	イベント名	含む, 含まない	イベント名。
4	トリガーの深刻度	等しい, 等しくない, 以上, 以下	トリガーの深刻度。サポートされているトリガーの深刻度についてはトリガー `severity` プロパティを参照してください。
5	トリガー値	等しい	トリガー値。サポートされているトリガー値についてはトリガー `value` プロパティを参照してください。
6	時間帯	範囲内, 範囲外	イベントがトリガーされた時刻を時間帯で指定。
7	ホスト IP	等しい, 等しくない	カンマ区切りで指定した 1 つまたは複数の IP 範囲。サポートされている IP 範囲の形式についてはネットワークディスカバリ設定セクションを参照してください。

条件	条件名	サポートされる演算子	期待される値
8	検出されたサービスのタイプ	等しい, 等しくない	検出されたサービスのタイプ。サービスのタイプはサービス検出に使用されたディスカバリーチェックのタイプと一致します。 サポートされているタイプについてはディスカバリーチェック type プロパティを参照してください。
9	検出されたサービスのポート	等しい, 等しくない	カンマ区切りで指定した 1 つまたは複数のポート範囲。
10	ディスカバリーステータス	等しい	検出されたオブジェクトのステータス。 可能な値: 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが検出された; 3 - ホストまたはサービスが消失。
11	アップタイムまたはダウンタイムの期間	以上, 以下	検出されたオブジェクトが現在のステータスにある秒数。
12	受信値	等しい, 等しくない, 以上, 以下, 含む, 含まない	Zabbix エージェント、SNMPv1、SNMPv2、SNMPv3 ディスカバリーチェック実行時に返された値。
13	ホストテンプレート	等しい, 等しくない	リンクされたテンプレート ID。
16	問題が抑制されている	はい, いいえ	値は不要: 「はい」演算子は問題が抑制されていること、「いいえ」は抑制されていないことを意味します。
18	ディスカバリールール	等しい, 等しくない	ディスカバリールールの ID。
19	ディスカバリーチェック	等しい, 等しくない	ディスカバリーチェックの ID。
20	プロキシ	等しい, 等しくない	プロキシの ID。
21	ディスカバリーオブジェクト	等しい	ディスカバリーイベントをトリガーしたオブジェクトのタイプ。 可能な値: 1 - 検出されたホスト; 2 - 検出されたサービス。
22	ホスト名	含む, 含まない, 正規表現一致, 正規表現不一致	ホスト名。自動登録条件で正規表現一致および正規表現不一致演算子を使用する場合は正規表現がサポートされます。
23	イベントタイプ	等しい	特定の内部イベント。 可能な値: 0 - アイテムが「サポートされていない」状態; 1 - アイテムが「通常」状態; 2 - LLD ルールが「サポートされていない」状態; 3 - LLD ルールが「通常」状態; 4 - トリガーが「不明」状態; 5 - トリガーが「通常」状態。
24	ホストメタデータ	含む, 含まない, 正規表現一致, 正規表現不一致	自動登録されたホストのメタデータ。正規表現一致および正規表現不一致演算子を使用する場合は正規表現がサポートされます。
25	タグ	等しい, 等しくない, 含む, 含まない	イベントタグ。
26	タグ値	等しい, 等しくない, 含む, 含まない	イベントタグ値。

条件	条件名	サポートされる演算子	期待される値
27	サービス	等しい, 等しくない	サービス ID。
28	サービス名	等しい, 等しくない	サービス名。

action.create

説明

`object action.create(object/array actions)`

このメソッドは、新しいアクションを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 作成するアクション。

action の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
filter	object	アクション用のアクションフィルターオブジェクト。
operations	array	アクション用に作成するアクション操作 。
recovery_operations	array	アクション用に作成するアクション回復操作。
update_operations	array	アクション用に作成するアクション更新操作。

戻り値

(object) actionids プロパティの下に作成されたアクションの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアクションの順序と一致します。

例

トリガーアクションの作成

ホスト"10084" からのトリガー (名前に"memory" が含まれる) が障害状態になると実行されるトリガーアクションを作成する。アクションには 4 つの操作が構成されます。最初の即時操作では、メディアタイプ"1" を介してユーザーグループ"7" 内のすべてのユーザーにメッセージが送信されます。イベントが 30 分以内に解決されない場合、2 番目の操作ではグループ"2" のすべてのホストでスクリプト"5"(スコープ" アクション操作" のスクリプト) が実行されます。イベントが解決された場合は、回復操作により、問題に関するメッセージを受け取ったすべてのユーザーに通知されます。イベントが更新されると、確認/更新操作により、問題に関するメッセージを受信したすべてのユーザーに (カスタムの件名とメッセージを使用して) 通知します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "30m",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084"
                },
```

```
            {
                "conditiontype": 3,
                "operator": 2,
                "value": "memory"
            }
        ]
    },
    "operations": [
        {
            "operationtype": 0,
            "esc_step_from": 1,
            "esc_step_to": 1,
            "opmessage_grp": [
                {
                    "usrgrpid": "7"
                }
            ],
            "opmessage": {
                "default_msg": 1,
                "mediatypeid": "1"
            }
        },
        {
            "operationtype": 1,
            "esc_step_from": 2,
            "esc_step_to": 2,
            "opconditions": [
                {
                    "conditiontype": 14,
                    "operator": 0,
                    "value": "0"
                }
            ],
            "opcommand_grp": [
                {
                    "groupid": "2"
                }
            ],
            "opcommand": {
                "scriptid": "5"
            }
        }
    ],
    "recovery_operations": [
        {
            "operationtype": "11",
            "opmessage": {
                "default_msg": 1
            }
        }
    ],
    "update_operations": [
        {
            "operationtype": "12",
            "opmessage": {
                "default_msg": 0,
                "message": "Custom update operation message body",
                "subject": "Custom update operation message subject"
            }
        }
    ]
},
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

ディスカバリアクションの作成

検出されたホストにテンプレート”10001” をリンクするディスカバリアクションを作成します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Discovery action",
        "eventsource": 1,
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 21,
                    "operator": 0,
                    "value": "1"
                },
                {
                    "conditiontype": 10,
                    "operator": 0,
                    "value": "2"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 6,
                "optemplate": [
                    {
                        "templateid": "10001"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
```

```
}
```

カスタム式フィルターの使用

アクション条件を評価するためのカスタム式"A and (B or C)" を使用するトリガーアクションを作成します。ホスト"10084" またはホスト"10106" からの" 警告" 以上の深刻度を持つトリガーが障害状態になると、このアクションは、メディアタイプ"1" を介してユーザーグループ"7" のすべてのユーザーにメッセージを送信します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "15m",
        "filter": {
            "evaltype": 3,
            "formula": "A and (B or C)",
            "conditions": [
                {
                    "conditiontype": 4,
                    "operator": 5,
                    "value": "2",
                    "formulaid": "A"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084",
                    "formulaid": "B"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10106",
                    "formulaid": "C"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

エージェントの自動登録ルール作成

ホスト名に”SRV ” が含まれているか、メタデータに”AlmaLinux” が含まれている場合、ホストをホストグループ”2” に追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

ホストタグとエージェント自動登録ルールの作成

ホストをホストグループ”2” に追加し、2 つのホストタグを追加する自動登録アクションを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers with tags",
        "eventsource": "2",
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            },
            {
                "operationtype": "13",
                "optag": [
                    {
                        "tag": "location",
                        "value": "office"
                    },
                    {
                        "tag": "city",
                        "value": "Riga"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

参照

- Action filter
- Action operation
- Script

ソース

CAction::create() in *ui/include/classes/api/services/CAction.php*.

action.delete

説明

`object action.delete(array actionIds)`

このメソッドは、アクションを削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアクションの ID。

戻り値

(object) actionids プロパティの下にある削除されたアクションの ID を含むオブジェクトを返します。

例

複数のアクションの削除

2 つのアクションを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

ソース

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

action.get

説明

`integer/array action.get(object parameters)`

このメソッドは、指定したパラメーターに応じたアクションを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 希望する出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートしています。

パラメーター	タイプ	説明
actionids	ID/array	指定された ID のアクションのみを返します。
groupids	ID/array	アクション条件で指定されたホストグループを使用するアクションのみを返します。

パラメーター	タイプ	説明
hostids	ID/array	アクション条件で指定されたホストを使用するアクションのみを返します。
triggerids	ID/array	アクション条件で指定されたトリガーを使用するアクションのみを返します。
mediatypeids	ID/array	指定されたメディアタイプを使用してメッセージを送信するアクションのみを返します。
usrgrpids	ID/array	指定されたユーザーグループにメッセージを送信するように構成されたアクションのみを返します。
userids	ID/array	指定されたユーザーにメッセージを送信するように構成されたアクションのみを返します。
scriptids	ID/array	指定されたスクリプトを実行するように構成されたアクションのみを返します。
selectFilter	query	アクション条件フィルターを含む`filter`プロパティを返します。
selectOperations	query	アクション操作を伴う`operation`プロパティを返します。
selectRecoveryOperations	query	アクション回復操作を含む`recovery\_operations`プロパティを返します。
selectUpdateOperations	query	アクション更新操作を含む`update\_operations`プロパティを返します。
sortfield	string/array	指定されたプロパティによって結果を並べ替えます。 利用可能な値: `actionid`、`name`、`status`
countOutput	boolean	これらのパラメーターについては、リファレンスの解説で解説されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のどちらかを返します。

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合は、取得したオブジェクトの数

例

トリガーアクションの取得

設定されたすべてのトリガーアクションを、アクションの実行条件および実行内容と共に取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Report problems to Zabbix administrators",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nTrigger status: {TRIGGER.STATUS}\r\nTrigger
                        "mediatypeid": "0"
                    }
                }
            ],
            "update_operations": [
                {
                    "operationid": "31",
                    "operationtype": "12",
                    "evaltype": "0",
```

```
                    "opmessage": {
                        "default_msg": "1",
                        "subject": "",
                        "message": "",
                        "mediatypeid": "0"
                    }
                },
                {
                    "operationid": "32",
                    "operationtype": "0",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "Updated: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                        "mediatypeid": "1"
                    },
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ],
                    "opmessage_usr": []
                },
                {
                    "operationid": "33",
                    "operationtype": "1",
                    "evaltype": "0",
                    "opcommand": {
                        "scriptid": "3"
                    },
                    "opcommand_hst": [
                        {
                            "hostid": "10084"
                        }
                    ],
                    "opcommand_grp": []
                }
            ]
        }
    ],
    "id": 1
}
```


ディスカバリアクションの取得

設定されたすべてのディスカバリアクションを、アクションの条件と操作とともに取得します。このフィルターは"and" 評価タイプを使用するため、formula プロパティは空で、eval_formula が自動的に生成されます。

リクエスト：


```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "2",
            "name": "Auto discovery. Linux servers.",
            "eventsource": "1",
            "status": "1",
            "esc_period": "0s",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "conditiontype": "10",
                        "operator": "0",
                        "value": "0",
                        "value2": "",
                        "formulaid": "B"
                    },
                    {
                        "conditiontype": "8",
                        "operator": "0",
                        "value": "9",
                        "value2": "",
                        "formulaid": "C"
                    },
                    {
                        "conditiontype": "12",
                        "operator": "2",
                        "value": "Linux",
                        "value2": "",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A and B and C"
            },
            "operations": [
                {
                    "operationid": "1",
                    "actionid": "2",
                    "operationtype": "6",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "optemplate": [
                        {
                            "templateid": "10001"
                        }
                    ]
                },
                {
                    "operationid": "2",
                    "actionid": "2",
                    "operationtype": "4",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
```

```
                    "evaltype": "0",
                    "opconditions": [],
                    "opgroup": [
                        {
                            "groupid": "2"
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```

参照

- Action filter
- Action operation

ソース

CAction::get() in *ui/include/classes/api/services/CAction.php*.

action.update

説明

`object action.update(object/array actions)`

このメソッドは、既存のアクションを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 更新するアクションのプロパティ。

`actionid` プロパティはアクションごとに定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

action の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
filter	object	現在のフィルターを置き換えるアクションフィルターオブジェクト。
operations	array	既存の操作を置き換えるアクション操作。
recovery_operations	array	既存の回復操作を置き換えるアクション回復操作。 パラメーターの動作: - Action オブジェクトの `eventsource` が”event created by a trigger”、”internal event” または”event created on service status update” に設定されている場合、サポート
update_operations	array	既存の更新操作を置き換えるアクション更新操作。 パラメーターの動作: - Action オブジェクトの `eventsource` が”event created by a trigger” または”event created on service status update” に設定されている場合、サポート

戻り値

(`object`) `actionids` プロパティの下で更新されたアクション ID を含むオブジェクトを返します。

例

アクションの無効化

ステータスを”1” に設定してアクションを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

参照

- Action filter
- Action operation

ソース

CAction::update() in *ui/include/classes/api/services/CAction.php*.

Alert

このクラスは、アラートを操作するためのものです。

オブジェクトリファレンス:

- Alert

利用可能なメソッド:

- alert.get - アラートを取得する

Alert オブジェクト

次のオブジェクトは、`alert`API に直接関連しています。

Alert

Note:
Alert は Zabbix サーバーによって作成され、API 経由で変更することはできません。

Alert オブジェクトは、特定のアクション操作が正常に実行されたかどうかに関する情報を含んでいます。以下のプロパティを持ちます。

プロパティ	タイプ	説明
alertid	ID	アラートの ID。
actionid	ID	アラートを生成したアクションの ID。

プロパティ	タイプ	説明
alerttype	integer	アラートタイプ。 有効な値: 0 - メッセージ 1 - リモートコマンド
clock	timestamp	アラートが生成された時刻。
error	string	メッセージの送信またはコマンドの実行に問題がある場合のエラーテキスト。
esc_step	integer	アラートが生成されたアクションエスカレーションステップ。
eventid	ID	アクションをトリガーしたイベントの ID。
mediatypeid	ID	メッセージの送信に使用されたメディアタイプの ID。
message	text	メッセージテキスト。 プロパティの動作: - `alerttype` が”message” に設定されている場合、サポート
retries	integer	Zabbix がメッセージの送信を試行した回数。
sendto	string	受信者のアドレス、ユーザー名、またはその他の識別子。 プロパティの動作: - `alerttype` が”message” に設定されている場合、サポート
status	integer	アクション操作が正常に実行されたかどうかを示すステータス。 `alerttype` が”message” の場合の有効な値: 0 - メッセージは送信されませんでした。 1 - メッセージが送信されました。 2 - 複数回試行しましたが失敗しました。 3 - 新しいアラートはアラートマネージャーによってまだ処理されていません。 `alerttype` が”remote command” の場合の有効な値: 0 - コマンドは実行されていません。 1 - コマンドが実行されました。 2 - Zabbix エージェントでコマンドを実行しようとしましたが、利用できませんでした。
subject	string	メッセージの件名。 プロパティの動作: - `alerttype` が”message” に設定されている場合、サポート
userid	ID	メッセージの送信先のユーザー ID。
p_eventid	ID	アラートを生成した障害のイベント ID。
acknowledgeid	ID	アラートを生成した確認の ID。

alert.get

説明

`integer/array alert.get(object parameters)`

このメソッドは、指定されたパラメーターに従ってアラートを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
alertids	ID/array	指定した ID のアラートのみを返します。
actionids	ID/array	指定したアクションによって生成されたアラートのみを返します。
eventids	ID/array	指定したイベントによって生成されたアラートのみを返します。
groupids	ID/array	指定したホストグループのオブジェクトによって生成されたアラートのみを返します。
hostids	ID/array	指定したホストのオブジェクトによって生成されたアラートのみを返します。
mediatypeids	ID/array	指定したメディアタイプを使用したメッセージアラートのみを返します。
objectids	ID/array	指定したオブジェクトによって生成されたアラートのみを返します
userids	ID/array	指定したユーザーに送信されたメッセージアラートのみを返します。
eventobject	integer	指定したタイプのオブジェクトに関連するイベントによって生成されたアラートのみを返します。 サポートされているオブジェクトタイプのリストについては、イベントobject を参照してください。 デフォルト: 0 - トリガー。
eventsource	integer	指定したタイプのイベントによって生成されたアラートのみを返します。 サポートされているイベントタイプのリストについては、イベントsource を参照してください。 デフォルト: 0 - トリガーイベント。
time_from	timestamp	指定した時刻以降に生成されたアラートのみを返します。
time_till	timestamp	指定した時刻以前に生成されたアラートのみを返します。
selectHosts	query	アクション操作をトリガーしたホストのデータを持つhosts プロパティを返します。
selectMediatypes	query	メッセージアラートで使用されたメディアタイプの配列を持つmediatypes プロパティを返します。 ユーザータイプによる制限については、mediatype.get を参照してください。
selectUsers	query	メッセージが送信されたユーザーの配列を持つusers プロパティを返します。 ユーザータイプによる制限については、user.get を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: alertid, clock, eventid, mediatypeid, sendto, status.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

アクション ID でアラートを取得する

アクション"3" によって生成されたすべてのアラートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

参照

- Host
- Media type
- User

ソース

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

API info

このクラスは、API に関するメタ情報を取得するためのものです。

利用可能なメソッド:

- apiinfo.version - Zabbix API のバージョンの取得

apiinfo.version

説明

`string apiinfo.version(array)`

このメソッドは、Zabbix API のバージョンを取得することができます。

> **Attention:**
> このメソッドは認証されていないユーザーのみが使用できます。

パラメーター

(array) このメソッドは空の配列を受け入れます。

戻り値

(string) Zabbix API のバージョンを返します。

> **Note:**
> Zabbix 2.0.4 以降、API のバージョンは Zabbix のバージョンと一致します。

例

API のバージョンの取得

Zabbix API のバージョンを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

ソース

CAPIInfo::version() in *ui/include/classes/api/services/CAPIInfo.php*.

Audit log

このクラスは、監査ログを操作するためのものです。

オブジェクトリファレンス:

- Audit log

利用可能なメソッド:

- auditlog.get - 監査ログを取得する

Audit log オブジェクト

次のオブジェクトは、auditlogAPI に直接関連しています。

監査ログ

監査ログオブジェクトには、ユーザーの操作に関する情報が含まれています。以下のプロパティがあります。

プロパティ	タイプ	説明
auditid	ID	監査ログエントリの ID。CUID アルゴリズムを使用して生成されます。
userid	ID	監査ログエントリの作成者のユーザー ID。
username	string	監査ログエントリの作成者のユーザー名。
clock	timestamp	監査ログエントリの作成タイムスタンプ。
ip	string	監査ログエントリの作成者の IP アドレス。

プロパティ	タイプ	説明
action	integer	監査ログエントリのアクション。 指定可能な値： 0 - 追加; 1 - 更新; 2 - 削除; 4 - ログアウト; 7 - 実行; 8 - ログイン; 9 - ログイン失敗; 10 - 履歴クリア; 11 - 設定リフレッシュ; 12 - プッシュ。

プロパティ	タイプ	説明
resourcetype	integer	監査ログエントリのリソースタイプ。 指定可能な値： 0 - ユーザー; 3 - メディアタイプ; 4 - ホスト; 5 - アクション; 6 - グラフ; 11 - ユーザーグループ; 13 - トリガー; 14 - ホストグループ; 15 - アイテム; 16 - イメージ; 17 - 値のマッピング; 18 - サービス; 19 - マップ; 22 - Web シナリオ; 23 - ディスカバリールール; 25 - スクリプト; 26 - プロキシ; 27 - メンテナンス; 28 - 正規表現; 29 - マクロ; 30 - テンプレート; 31 - トリガープロトタイプ; 32 - アイコンマッピング; 33 - ダッシュボード; 34 - イベント相関; 35 - グラフプロトタイプ; 36 - アイテムプロトタイプ; 37 - ホストプロトタイプ; 38 - 自動登録; 39 - モジュール; 40 - 設定; 41 - ハウスキーピング; 42 - 認証; 43 - テンプレートダッシュボード; 44 - ユーザーロール; 45 - API トークン; 46 - スケジュールレポート; 47 - 高可用性ノード; 48 - SLA; 49 - ユーザーディレクトリ; 50 - テンプレートグループ; 51 - コネクタ; 52 - LLD ルール; 53 - 履歴; 54 - 多要素認証; 55 - プロキシグループ; 56 - LLD ルールプロトタイプ。
resourceid	ID	監査ログエントリのリソース識別子。
resourcename	string	監査ログエントリのリソースの人間が読める名前。
recordsetid	ID	監査ログエントリのレコードセット ID。同じ操作中に作成された監査ログレコードは同じレコードセット ID を持ちます。CUID アルゴリズムを使用して生成されます。

プロパティ	タイプ	説明
details	text	監査ログエントリの詳細。詳細は JSON オブジェクトとして保存され、各プロパティ名は変更が発生したプロパティまたはネストされたオブジェクトへのパスであり、各値にはこのプロパティまたはネストされたオブジェクトでの変更に関するデータ（配列形式）が含まれます。 値の形式例： ["add"] - ネストされたオブジェクトが追加された; ["add", "<value>"] - 追加されたオブジェクトのプロパティが <value> である; ["update"] - ネストされたオブジェクトが更新された; ["update", "<new value>", "<old value>"] - 更新されたオブジェクトのプロパティが <old value> から <new value> に変更された; ["delete"] - ネストされたオブジェクトが削除された。

auditlog.get

説明

`integer/array auditlog.get(object parameters)`

このメソッドは、与えられたパラメーターに従って監査ログレコードを取得することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object`) 目的の出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートしています。

パラメーター	タイプ	説明
auditids	ID/array	指定された ID の監査ログのみを返します。
userids	ID/array	指定されたユーザーによって作成された監査ログのみを返します。
time_from	timestamp	指定された時刻以降に作成された監査ログのみを返します。
time_till	timestamp	指定された時刻以前に作成された監査ログのみを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。 利用可能な値: `auditid`、`userid`、`clock`
countOutput	boolean	これらのパラメーターについては、リファレンスの解説で説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

監査ログの取得

最新の監査ログレコードを 2 件取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

参照

- Audit log object

ソース

CAuditLog::get() in *ui/include/classes/api/services/CAuditLog.php*.

Authentication

このクラスは、認証設定を操作するためのものです。

オブジェクトリファレンス:

- Authentication

利用可能なメソッド:

- authentication.get - 認証を取得する
- authentication.update - 認証を更新する

Authentication オブジェクト

次のオブジェクトは、`authentication`API に直接関連しています。

認証

認証オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
authentication_type	integer	デフォルト認証。 可能な値: 0 - *(デフォルト)* 内部; 1 - LDAP。
http_auth_enabled	integer	HTTP 認証。 可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 プロパティの動作: - フロントエンド設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合にサポートされます。
http_login_form	integer	デフォルトのログインフォーム。 可能な値: 0 - *(デフォルト)* Zabbix ログインフォーム; 1 - HTTP ログインフォーム。 プロパティの動作: - フロントエンド設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合にサポートされます。
http_strip_domains	string	削除するドメイン名。 プロパティの動作: - フロントエンド設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合にサポートされます。
http_case_sensitive	integer	HTTP 大文字・小文字を区別したログイン。 可能な値: 0 - オフ; 1 - *(デフォルト)* オン。 プロパティの動作: - フロントエンド設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合にサポートされます。
ldap_auth_enabled	integer	LDAP 認証。 可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
ldap_case_sensitive	integer	LDAP 大文字・小文字を区別したログイン。 可能な値: 0 - オフ; 1 - *(デフォルト)* オン。

プロパティ	タイプ	説明
ldap_userdirectoryid	ID	LDAP 認証のデフォルトユーザーディレクトリの ID。 LDAP またはシステムデフォルトに `gui_access` が設定されたユーザーグループに使用されます。 プロパティの動作: - `ldap_auth_enabled` が「有効」に設定されている場合は必須
saml_auth_enabled	integer	SAML 認証。 可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
saml_case_sensitive	integer	SAML 大文字・小文字を区別したログイン。 可能な値: 0 - オフ; 1 - *(デフォルト)* オン。
passwd_min_length	integer	パスワードの最小長要件。 可能な値の範囲は 1 から 70 まで。 デフォルト: 8。
passwd_check_rules	integer	パスワードチェックルール。 可能なビットマップ値: 0 - パスワードの長さをチェック; 1 - パスワードが大文字と小文字のラテン文字を使用しているかチェック; 2 - パスワードが数字を使用しているかチェック; 4 - パスワードが特殊文字を使用しているかチェック; 8 - *(デフォルト)* パスワードが一般的に使用されるパスワードのリストに含まれていないこと、および「Zabbix」やユーザーの名前、姓、ユーザー名の派生語を含まないことをチェック。 これはビットマスクフィールドであり、可能なビットマップ値の任意の合計が許容されます（例: すべてのルールをチェックする場合は 15）。
ldap_jit_status	integer	LDAP プロビジョニングのステータス。 可能な値: 0 - 設定された LDAP IdP で無効; 1 - 設定された LDAP IdP で有効。
saml_jit_status	integer	SAML プロビジョニングのステータス。 可能な値: 0 - 設定された SAML IdP で無効; 1 - 設定された SAML IdP で有効。
jit_provision_interval	string	ログインユーザーの JIT プロビジョニングリクエスト間の時間間隔。 秒数または単位付きの時間（3600s,60m,1h,1d,1M,1y）を受け付けます。 最小値: 1h。 デフォルト: 1h。
disabled_usrgrpid	ID	LDAP プロビジョニングでのみ利用可能。 プロビジョニング解除されたユーザーを割り当てるユーザーグループの ID。 ユーザーグループは無効である必要があり、設定時に有効化や削除はできません。 プロパティの動作: - `ldap_jit_status` が「設定された LDAP IdP で有効」または `saml_jit_status` が「設定された SAML IdP で有効」に設定されている場合は必須

プロパティ	タイプ	説明
mfa_status	integer	多要素認証。 可能な値: 0 - （すべての設定済み MFA 方式で）無効; 1 - （すべての設定済み MFA 方式で）有効。
mfaid	ID	MFA 有効なユーザーグループのデフォルト MFA 方式。 プロパティの動作: - `mfa_status` が「有効」に設定されている場合は必須

authentication.get

説明

`object authentication.get(object parameters)`

このメソッドは、指定されたパラメーターに従って認証オブジェクトを取得することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object`) 目的の出力を定義するパラメーター。

このメソッドはパラメーターを 1 つだけサポートします。

パラメーター	タイプ	説明
output	query	これらのパラメーターについては、リファレンスの解説で解説されています。

戻り値

(`object`) 認証オブジェクトを返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "authentication_type": "0",
        "http_auth_enabled": "0",
        "http_login_form": "0",
        "http_strip_domains": "",
        "http_case_sensitive": "1",
        "ldap_auth_enabled": "0",
        "ldap_case_sensitive": "1",
        "ldap_userdirectoryid": "0",
        "saml_auth_enabled": "0",
        "saml_case_sensitive": "0",
```

```
        "passwd_min_length": "8",
        "passwd_check_rules": "15",
        "jit_provision_interval": "1h",
        "saml_jit_status": "0",
        "ldap_jit_status": "0",
        "disabled_usrgrpid": "9",
        "mfa_status": "0",
        "mfaid": "0"
    },
    "id": 1
}
```


ソース

CAuthentication::get() in *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

説明

`object authentication.update(object authentication)`

このメソッドは、既存の認証設定を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 更新する認証のプロパティ。

戻り値

(array) 更新したパラメーターの名前を配列で返します。

例

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```


ソース

CAuthentication::update() in *ui/include/classes/api/services/CAuthentication.php*.

Autoregistration

このクラスは、自動登録を操作するためのものです。

オブジェクトリファレンス:

- Autoregistration

利用可能なメソッド:

- autoregistration.get - 自動登録を取得する
- autoregistration.update - 自動登録を更新する

Autoregistration オブジェクト

次のオブジェクトは、`autoregistrationAPI` に直接関連しています。

自動登録

自動登録オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tls_accept	integer	自動登録のために許可される受信接続のタイプ。 可能な値: 1 - 暗号化されていない接続を許可; 2 - PSK による TLS を許可; 3 - 暗号化されていない接続と PSK による TLS の両方を許可。
tls_psk_identity	string	PSK アイデンティティ; 1 つの PSK とだけペアにする必要があります (autoregistration、hosts、proxies全体で)。 PSK アイデンティティには、機密情報を含めないでください。どの PSK を使用するかを受信側に通知するためにネットワーク上で暗号化されずに送信されます。 プロパティの動作: - 書き込み専用
tls_psk	string	事前共有鍵 (PSK); 少なくとも 32 桁の 16 進数である必要があります。 プロパティの動作: - 書き込み専用

autoregistration.get

説明

`object autoregistration.get(object parameters)`

このメソッドは、指定されたパラメーターに従って自動登録オブジェクトを取得することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object)` 目的の出力を定義するパラメーター。

このメソッドはパラメーターを 1 つだけサポートします。

パラメーター	タイプ	説明
output	query	これらのパラメーターについては、リファレンスの解説で解説されています。

戻り値

(object) autoregistration オブジェクトを返します。

例

リクエスト:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tls_accept": "3"
    },
    "id": 1
}
```

ソース

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

説明

`object autoregistration.update(object autoregistration)`

このメソッドは、既存の自動登録を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 更新する自動登録のプロパティ。

戻り値

(boolean) 更新が成功すると、ブール値 true を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
```

```
    "id": 1
}
```

ソース

CAutoregistration::update() in *ui/include/classes/api/services/CAutoregistration.php*.

Configuration

このクラスは、Zabbix の設定データのエクスポートとインポートを行うためのものです。

利用可能なメソッド:

- configuration.export - 設定のエクスポート
- configuration.import - 設定のインポート
- configuration.importcompare - インポートファイルと現在のシステムの要素の比較

configuration.export

説明

`string configuration.export(object parameters)`

このメソッドは、構成データをシリアル化された文字列としてエクスポートできます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) エクスポートするオブジェクトと使用するフォーマットを定義するパラメータ。

パラメータ	型	説明
format	string	データをエクスポートするフォーマット。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - 未加工の PHP 配列。 パラメータの動作: - 必須
prettyprint	boolean	インデントを追加して出力をより人間が読みやすくする。 指定可能な値: `true` - インデントを追加; `false` - *(デフォルト)* インデントを追加しない。

パラメータ	型	説明
options	object	エクスポートするオブジェクト。 options オブジェクトには以下のパラメータがある: host_groups - (array) エクスポートするホストグループの ID; hosts - (array) エクスポートするホストの ID; images - (array) エクスポートするイメージの ID; maps - (array) エクスポートするマップの ID; mediaTypes - (array) エクスポートするメディアタイプの ID; template_groups - (array) エクスポートするテンプレートグループの ID; templates - (array) エクスポートするテンプレートの ID。 *Admin* および *User* タイプのユーザーは、読み取り専用または読み書きの権限を持つオブジェクトおよびイメージのみエクスポートでき、メディアタイプはエクスポートできません。 パラメータの動作: - 必須

戻り値

(string) 要求された構成データを含むシリアル化された文字列を返します。

例

テンプレートのエクスポート

テンプレート"10571" の設定を XML 文字列としてエクスポートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```

ソース

CConfiguration::export() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

説明

`boolean configuration.import(object parameters)`

このメソッドは、シリアル化された文字列から設定データをインポートすることができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) インポートするデータとデータの処理方法に関するルールを含むパラメータ。

パラメータ	型	説明
format	string	シリアライズされた文字列のフォーマット。 指定可能な値: yaml - YAML; xml - XML; json - JSON。 パラメータの動作: - 必須
source	string	設定データを含むシリアライズされた文字列。 パラメータの動作: - 必須
rules	object	新規および既存オブジェクトのインポート方法に関するルール。 *Admin* タイプのユーザーは、自分が読み書き権限を持つオブジェクトとマップのみをインポートできます。たとえば、ホストおよびそのエンティティ（アイテム、トリガー、グラフなど）は、インポートされるホストが所属するホストグループに対してユーザーのユーザーグループが権限を持っている場合にのみインポートできます。*Admin* タイプのユーザーはイメージやメディアタイプをインポートできません。 rules パラメータの詳細は下記の表を参照してください。 パラメータの動作: - 必須

Attention:
ルールが指定されていない場合、設定は更新されません。

rules オブジェクトは以下のパラメータをサポートします。

パラメータ	型	説明
discoveryRules	object	LLD ルールのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい LLD ルールが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の LLD ルールが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: false。

パラメータ	型	説明
graphs	object	グラフのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいグラフが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のグラフが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないグラフがデータベースから削除されます。デフォルト: false。
host_groups	object	ホストグループのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいホストグループが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のホストグループが更新されます。デフォルト: false。
template_groups	object	テンプレートグループのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートグループが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートグループが更新されます。デフォルト: false。
hosts	object	ホストのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいホストが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のホストが更新されます。デフォルト: false。
httptests	object	Web シナリオのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false。
images	object	イメージのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいイメージが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のイメージが更新されます。デフォルト: false。
items	object	アイテムのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないアイテムがデータベースから削除されます。デフォルト: false。

パラメータ	型	説明
maps	object	マップのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のマップが更新されます。デフォルト: false。
mediaTypes	object	メディアタイプのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のメディアタイプが更新されます。デフォルト: false。
templateLinkage	object	テンプレートリンクのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、インポートされるホストまたはテンプレートにリンクされていないが、インポートデータに存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートされるホストまたはテンプレートにリンクされているが、インポートデータに存在しないテンプレートが、エンティティ（アイテム、トリガーなど）を削除せずにリンク解除されます。デフォルト: false。
templates	object	テンプレートのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートが更新されます。デフォルト: false。
templateDashboards	object	テンプレートダッシュボードのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false。
triggers	object	トリガーのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないトリガーがデータベースから削除されます。デフォルト: false。
valueMaps	object	ホストまたはテンプレートの値のマップのインポート方法に関するルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい値のマップが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の値のマップが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない値のマップがデータベースから削除されます。デフォルト: false。

戻り値

(boolean) インポートに成功した場合は true を返します。

例

テンプレートのインポート

XML 文字列に含まれるテンプレート設定をインポートします。XML 文字列にアイテムやトリガーが不足している場合、それらはデータベースから削除され、それ以外は変更されません。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.import",
    "params": {
        "format": "xml",
        "rules": {
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": false
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

ソース

CConfiguration::import() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

説明

`array configuration.importcompare(object parameters)`

このメソッドは、インポートファイルを現在のシステム要素と比較し、このインポートファイルがインポートされる場合に何が変更されるかを表示することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) インポート可能なデータと、そのデータをどのように処理するかのルールを含むパラメータ。

パラメータ	型	説明
format	string	シリアライズされた文字列のフォーマット。 指定可能な値: yaml - YAML; xml - XML; json - JSON。 パラメータの動作: - 必須
source	string	設定データを含むシリアライズされた文字列。 パラメータの動作: - 必須
rules	object	新規および既存オブジェクトをどのように比較するかのルール。 *Admin* および *User* タイプのユーザーは、読み取り専用または 読み書きの権限を持つ既存のオブジェクトとのみ新規オブジェクトを比較できます。 rules パラメータの詳細は下記の表を参照してください。 パラメータの動作: - 必須

Attention:
ルールが指定されていない場合、更新するものがなく、結果は空になります。

Note:
比較はホストグループとテンプレートのみで行われます。トリガーとグラフはインポートされたテンプレートのみ比較され、それ以外は「新規」とみなされます。

rules オブジェクトは以下のパラメータをサポートします。

パラメータ	型	説明
discoveryRules	object	LLD ルールのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい LLD ルールが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の LLD ルールが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: false。
graphs	object	グラフのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいグラフが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のグラフが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないグラフがデータベースから削除されます。デフォルト: false。

パラメータ	型	説明
host_groups	object	ホストグループのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいホストグループが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のホストグループが更新されます。デフォルト: false。
template_groups	object	テンプレートグループのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートグループが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートグループが更新されます。デフォルト: false。
hosts	object	ホストのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいホストが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のホストが更新されます。デフォルト: false。 このパラメータは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
httptests	object	Web シナリオのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false。
images	object	イメージのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいイメージが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のイメージが更新されます。デフォルト: false。 このパラメータは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
items	object	アイテムのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないアイテムがデータベースから削除されます。デフォルト: false。

パラメータ	型	説明
maps	object	マップのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のマップが更新されます。デフォルト: false。 このパラメータは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
mediaTypes	object	メディアタイプのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のメディアタイプが更新されます。デフォルト: false。 このパラメータは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
templateLinkage	object	テンプレートリンクのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、インポート対象のホストまたはテンプレートにリンクされていないが、インポートデータに存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true の場合、インポート対象のホストまたはテンプレートにリンクされているが、インポートデータに存在しないテンプレートが、エンティティ（アイテム、トリガーなど）を削除せずにリンク解除されます。デフォルト: false。
templates	object	テンプレートのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートが更新されます。デフォルト: false。
templateDashboards	object	テンプレートダッシュボードのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false。
triggers	object	トリガーのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しないトリガーがデータベースから削除されます。デフォルト: false。

パラメータ	型	説明
valueMaps	object	ホストまたはテンプレートの値マップのインポート方法のルール。 サポートされるパラメータ: createMissing - (boolean) true の場合、新しい値マップが作成されます。デフォルト: false; updateExisting - (boolean) true の場合、既存の値マップが更新されます。デフォルト: false; deleteMissing - (boolean) true の場合、インポートデータに存在しない値マップがデータベースから削除されます。デフォルト: false。

戻り値

(array) 行われる設定の変更を含む配列を返します。

例

テンプレートインポートの比較

XML 文字列に含まれるテンプレートを現在のシステム要素と比較し、このテンプレートをインポートした場合に何が変更されるかを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.importcompare",
    "params": {
        "format": "xml",
        "rules": {
            "discoveryRules": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "graphs": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "host_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "template_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "httptests": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "templateLinkage": {
                "createMissing": true,
                "deleteMissing": true
            },
            "templates": {
                "createMissing": true,
                "updateExisting": true
```

```
            },
            "templateDashboards": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templates": {
            "updated": [
                {
                    "before": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "after": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "items": {
                        "added": [
                            {
                                "after": {
                                    "uuid": "648006da5971424ead0c47ddbbf1ea2e",
                                    "name": "CPU utilization",
                                    "key": "system.cpu.util",
                                    "value_type": "FLOAT",
                                    "units": "%"
                                },
                                "triggers": {
                                    "added": [
                                        {
```

```
                                        "after": {
                                            "uuid": "736225012c534ec480c2a66a91322ce0",
                                            "expression": "avg(/New template/system.cpu.util,3m)>70",
                                            "name": "CPU utilization too high on 'New host' for 3 minu
                                            "priority": "WARNING"
                                        }
                                    }
                                ]
                            }
                        }
                    ],
                    "removed": [
                        {
                            "before": {
                                "uuid": "6805d4c39a624a8bab2cc8ab63df1ab3",
                                "name": "CPU load",
                                "key": "system.cpu.load",
                                "value_type": "FLOAT"
                            },
                            "triggers": {
                                "removed": [
                                    {
                                        "before": {
                                            "uuid": "ab4c2526c2bc42e48a633082255ebcb3",
                                            "expression": "avg(/New template/system.cpu.load,3m)>2",
                                            "name": "CPU load too high on 'New host' for 3 minutes",
                                            "priority": "WARNING"
                                        }
                                    }
                                ]
                            }
                        }
                    ],
                    "updated": [
                        {
                            "before": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping"
                            },
                            "after": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping",
                                "delay": "3m"
                            }
                        }
                    ]
                }
            }
        ]
    }
},
"id": 1
}
```


ソース

CConfiguration::importcompare() in *ui/include/classes/api/services/CConfiguration.php*.

Connector

このクラスは、コネクタオブジェクトを操作するためのものです。

オブジェクトリファレンス:

- Connector
- Tag filter

利用可能なメソッド:

- connector.create - 新しいコネクタを作成する
- connector.delete - コネクタを削除する
- connector.get - コネクタを取得する
- connector.update - コネクタを更新する

Connector オブジェクト

次のオブジェクトは、connectorAPI に直接関連しています。

コネクタ

コネクタオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
connectorid	ID	コネクタの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	コネクタの名前。 プロパティの動作: - 作成操作では必須
url	string	エンドポイント URL、つまり受信側の URL。 ユーザーマクロがサポートされています。 プロパティの動作: - 作成操作では必須
protocol	integer	通信プロトコル。 可能な値: 0 - *(デフォルト)* Zabbix Streaming Protocol v1.0.
data_type	integer	データタイプ。 可能な値: 0 - *(デフォルト)* アイテム値; 1 - イベント。
item_value_type	integer	送信するアイテム値タイプの合計。 可能な値: 1 - 数値 (浮動小数点); 2 - 文字列; 4 - ログ; 8 - 数値 (整数); 10 - テキスト; 20 - バイナリ。 デフォルト: 31 - すべてのアイテムタイプ (バイナリを除く)。 プロパティの動作: - data_type が「アイテム値」に設定されている場合にサポートされます。
max_records	integer	1 つのメッセージ内で送信できるイベントまたはアイテムの最大数。 可能な値: 0-2147483647 (32 ビット符号付き整数の最大値)。 デフォルト: 0 - 無制限。

プロパティ	タイプ	説明
max_senders	integer	このコネクタのために実行する送信プロセスの数。 可能な値: 1-100。 デフォルト: 1。
max_attempts	integer	試行回数。 可能な値: 1-5。 デフォルト: 1。
attempt_interval	string	再試行間隔。 秒単位で指定します。 可能な値: 0s-10s。 デフォルト: 5s。 プロパティの動作: - `max_attempts` が 1 より大きい場合にサポートされます。
timeout	string	タイムアウト。 時間のサフィックスがサポートされています (例: 30s, 1m)。 ユーザーマクロがサポートされています。 可能な値: 1s-60s。 デフォルト: 5s。
http_proxy	string	HTTP(S) プロキシ接続文字列。 *[protocol]://[username[:password]@]proxy.example.com[:port]* の形式。 ユーザーマクロがサポートされています。
authtype	integer	HTTP 認証方式。 可能な値: 0 - *(デフォルト)* なし; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer。
username	string	ユーザー名。 ユーザーマクロがサポートされています。 プロパティの動作: - `authtype` が「Basic」「NTLM」「Kerberos」「Digest」に設定されている場合にサポートされます
password	string	パスワード。 ユーザーマクロがサポートされています。 プロパティの動作: - `authtype` が「Basic」「NTLM」「Kerberos」「Digest」に設定されている場合にサポートされます
token	string	Bearer トークン。 ユーザーマクロがサポートされています。 プロパティの動作: - `authtype` が「Bearer」に設定されている場合に必須
verify_peer	integer	ホストの証明書が正当かどうかを検証するかどうか。 可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。

プロパティ	タイプ	説明
verify_host	integer	接続先のホスト名が証明書のホスト名と一致するかどうかを検証するかどうか。 可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
ssl_cert_file	string	公開 SSL キーのファイルパス。 ユーザーマクロがサポートされています。
ssl_key_file	string	秘密 SSL キーのファイルパス。 ユーザーマクロがサポートされています。
ssl_key_password	string	SSL キーのパスワード。 ユーザーマクロがサポートされています。
description	text	コネクタの説明。
status	integer	コネクタが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
tags_evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。

タグフィルター

タグフィルターを使用すると、一致するアイテム値またはイベントのみをエクスポートできます。設定しない場合は、すべてがエクスポートされます。タグフィルターオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	タグ名。 プロパティの動作: - 必須
operator	integer	条件演算子。 可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 12 - 存在する; 13 - 存在しない。
value	string	タグ値。

connector.create

説明

`object connector.create(object/array connectors)`

このメソッドは、新しいコネクタオブジェクトを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object/array)` 作成するコネクタオブジェクト。

標準コネクタプロパティに加えて、メソッドは次のパラメータを受け入れます。

パラメーター	タイプ	説明
tags	array	コネクタタグフィルター。

戻り値

(object) connectorids プロパティの下に作成されたコネクタの ID を含むオブジェクトを返します。返される ID の順序は、渡されたコネクタの順序と一致します。

例

コネクタの作成

タグフィルターを使用してトリガーイベントをエクスポートするためのコネクタを作成します。HTTP 認証は Bearer トークンを使用して実行されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.create",
    "params": [
        {
            "name": "Export of events",
            "data_type": 1,
            "url": "{$DATA_EXPORT_URL}",
            "authtype": 5,
            "token": "{$DATA_EXPORT_BEARER_TOKEN}",
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

ソース

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

説明

`object connector.delete(array connectorids)`

このメソッドは、コネクタのエントリを削除することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するコネクタの ID。

戻り値

(object) connectorids プロパティの下にある、削除されたコネクタの ID を含むオブジェクトを返します。

例

複数のコネクタの削除

2 つのコネクタエントリを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CConnector::delete() in *ui/include/classes/api/services/CConnector.php*.

connector.get

説明

`integer/array connector.get(object parameters)`

このメソッドは、与えられたパラメーターに従ってコネクタオブジェクトを取得することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメーター	タイプ	説明
connectorids	ID/array	指定された ID を持つコネクタのみを返します。

パラメーター	タイプ	説明
selectTags	query	コネクタタグフィルターを使用して tags プロパティを返します。 count をサポートします。
sortfield	string/array	指定されたプロパティによって結果を並べ替えます。 利用可能な値: connectorid、name、data_type、status。
countOutput	boolean	これらのパラメーターについては、リファレンスの解説で解説されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのコネクタを取得する

すべてのコネクタとそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
```

```
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
            "protocol": "0",
            "data_type": "1",
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```


ソース

CConnector:get() in *ui/include/classes/api/services/CConnector.php*.

connector.update

説明

`object connector.update(object/array connectors)`

このメソッドは、既存のコネクタを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 更新されるコネクタのプロパティ。

connectorid プロパティはコネクタごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のコネクタプロパティに加えて、このメソッドは次のパラメーターをサポートします。

パラメーター	タイプ	説明
tags	array	現在のタグフィルタを置き換えるコネクタタグフィルタ

戻り値

(object) connectorids プロパティの下にある、更新されたコネクタの ID を含むオブジェクトを返します。

例

HTTP 認証タイプの変更

ID "3" のコネクタの HTTP 認証タイプをベアラーに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

タグフィルターの更新

ID "5" のコネクタのタグフィルターを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": [
        {
            "connectorid": 5,
            "tags_evaltype": 2,
            "tags": [
```

```
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                },
                {
                    "tag": "error",
                    "operator": 12,
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Correlation

このクラスは、相関関係を操作するためのものです。

オブジェクトリファレンス:

- Correlation
- Correlation operation
- Correlation filter
- Correlation filter condition

利用可能なメソッド:

- correlation.create - 新しい相関関係の作成
- correlation.delete - 相関関係の削除
- correlation.get - 相関関係の取得
- correlation.update - 相関関係の更新

Correlation オブジェクト

次のオブジェクトは、`correlation`API に直接関連しています。

コリレーション

コリレーションオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
correlationid	ID	コリレーションの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	コリレーションの名前。 プロパティの動作: - 作成操作時に必須
description	string	コリレーションの説明。
status	integer	コリレーションが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。

相関操作

相関操作オブジェクトは、相関が実行されたときに実行される操作を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
type	integer	操作のタイプ。 可能な値: 0 - 古いイベントをクローズ; 1 - 新しいイベントをクローズ。 プロパティの動作: - 必須

コリレーションフィルター

コリレーションフィルターオブジェクトは、設定されたコリレーション操作を実行するために満たす必要がある条件のセットを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
conditions	array	結果をフィルタリングするために使用するフィルター条件のセット。条件は、式内での配置順にソートされます。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	フィルター条件の評価に使用される生成された式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
formula	string	カスタム式のフィルター条件の評価に使用するユーザー定義の式。式には、`formulaid` で特定のフィルター条件を参照する ID を含める必要があります。式で使用される ID は、フィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

相関フィルター条件

相関フィルター条件オブジェクトは、相関操作を実行する前にチェックする必要がある特定の条件を定義します。

プロパティ	タイプ	説明
type	integer	条件のタイプ。 可能な値： 0 - 旧イベントタグ; 1 - 新イベントタグ; 2 - 新イベントホストグループ; 3 - イベントタグペア; 4 - 旧イベントタグ値; 5 - 新イベントタグ値。 プロパティの動作: - 必須
tag	string	イベントタグ（旧または新）。 プロパティの動作: - `type` が「旧イベントタグ」、「新イベントタグ」、「旧イベントタグ値」、「新イベントタグ値」に設定されている場合は必須
groupid	ID	ホストグループの ID。 プロパティの動作: - `type` が「新イベントホストグループ」に設定されている場合は必須
oldtag	string	旧イベントタグ。 プロパティの動作: - `type` が「イベントタグペア」に設定されている場合は必須
newtag	string	旧イベントタグ。 プロパティの動作: - `type` が「イベントタグペア」に設定されている場合は必須
value	string	イベントタグ（旧または新）の値。 プロパティの動作: - `type` が「旧イベントタグ値」または「新イベントタグ値」に設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。大文字のみ使用可能。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後リクエスト時には新たに生成されます。
operator	integer	条件演算子。 プロパティの動作: - `type` が「新イベントホストグループ」、「旧イベントタグ値」、「新イベントタグ値」に設定されている場合は必須

Note:
さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには、correlation.getおよびcorrelation.createメソッドページの例を参照してください。

各条件タイプでサポートされる演算子と値は以下の通りです。

条件	条件名	サポートされる演算子	期待される値
2	ホストグループ	=, <>	ホストグループ ID
4	旧イベントタグ値	=, <>, like, not like	string
5	新イベントタグ値	=, <>, like, not like	string

correlation.create

説明

`object correlation.create(object/array correlations)`

このメソッドは、新しい相関関係を作成することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 作成する相関関係。

標準相関プロパティに加えて、メソッドは次のパラメーターをサポートしています。

パラメーター	タイプ	説明
operations	array	相関用に作成する相関操作。 パラメーターの動作: - 必須
filter	object	相関用の相関フィルターオブジェクト。 パラメーターの動作: - 必須

戻り値

(`object`) `correlationids` プロパティの下に作成された相関関係の ID を含むオブジェクトを返します。返す ID の順序は、渡された相関関係の順序と一致します。

例

新しいイベントタグ相関の作成

1 つの条件と 1 つの演算による評価方法 `AND/OR` を使用して相関関係を作成します。デフォルトでは相関関係は有効になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new event tag correlation",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "type": 1,
                    "tag": "ok"
                }
            ]
        },
```

```
        "operations": [
            {
                "type": 0
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

カスタムフィルター条件を使用する相関関係を作成します。式 ID、”A” または”B” は任意に選択されています。条件タイプは、演算子”<>” を使用した” ホストグループ” になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new host group correlation",
        "description": "a custom description",
        "status": 0,
        "filter": {
            "evaltype": 3,
            "formula": "A or B",
            "conditions": [
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "A"
                },
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "B"
                }
            ]
        },
        "operations": [
            {
                "type": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
```

```
            "2"
        ]
    },
    "id": 1
}
```

参照

- Correlation filter
- Correlation operation

ソース

CCorrelation::create() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

説明

`object correlation.delete(array correlationids)`

このメソッドは、相関関係を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する相関関係の ID。

戻り値

(object) correlationids プロパティの下にある削除された相関関係の ID を含むオブジェクトを返します。

例

複数の相関関係の削除

2 つの相関関係を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CCorrelation::delete() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

説明

`integer/array correlation.get(object parameters)`

このメソッドは、指定したパラメーターに応じた相関関係を取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 希望する出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートしています。

パラメーター	Type	説明
correlationids	ID/array	指定された ID の相関のみを返します。
selectFilter	query	相関条件の`filter`プロパティを返します。
selectOperations	query	相関条件の`operations`プロパティを返します。
sortfield	string/array	与えられたプロパティで結果を並べ替えます。 可能な値: `correlationid`、`name`、`status`。
countOutput	boolean	これらのパラメーターについては、リファレンスの解説で解説されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

相関関係の取得

設定されたすべての相関を、相関条件および操作とともに取得します。フィルターは"and/or" 評価タイプを使用するため、`formula` プロパティは空で、`eval_formula` が自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "correlationid": "1",
            "name": "Correlation 1",
            "description": "",
            "status": "0",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "type": "3",
                        "oldtag": "error",
                        "newtag": "ok",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "type": "0"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Correlation filter
- Correlation operation

ソース

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

説明

`object correlation.update(object/array correlations)`

このメソッドは、既存の相関関係を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 更新する相関関係プロパティ。

`correlationid` プロパティは相関ごとに定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準相関プロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメーター	タイプ	説明
filter	object	現在のフィルターを置き換える相関関係フィルターオブジェクト。
operations	array	既存の操作を置き換える相関関係操作。

戻り値

(object) correlationids'プロパティの下で更新された相関関係 ID を含むオブジェクトを返します。

例

相関関係の無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

評価方法を維持したままで条件の置き換え

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

参照

- Correlation filter

- Correlation operation

ソース

CCorrelation::update() in *ui/include/classes/api/services/CCorrelation.php*.

Dashboard

このクラスは、ダッシュボードを操作するためのものです。

オブジェクトリファレンス:

- Dashboard
- Dashboard page
- Dashboard widget - Dashboard widget field
- Dashboard user group
- Dashboard user

利用可能なメソッド:

- dashboard.create - 新しいダッシュボードの作成
- dashboard.delete - ダッシュボードの削除
- dashboard.get - ダッシュボードの取得
- dashboard.update - ダッシュボードの更新

Dashboard オブジェクト

次のオブジェクトは、dashboardAPI に直接関連しています。

ダッシュボード

ダッシュボードオブジェクトは以下のプロパティを持ちます。

プロパティ	型	説明
dashboardid	ID	ダッシュボードの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ダッシュボードの名前。 プロパティの動作: - 作成操作では必須
userid	ID	ダッシュボードの所有者であるユーザーの ID。
private	integer	ダッシュボードの共有タイプ。 指定可能な値: 0 - パブリックダッシュボード; 1 - *(デフォルト)* プライベートダッシュボード。
display_period	integer	デフォルトのページ表示期間（秒単位）。 指定可能な値: 10, 30, 60, 120, 600, 1800, 3600。 デフォルト: 30。
auto_start	integer	スライドショーの自動開始。 指定可能な値: 0 - スライドショーを自動開始しない; 1 - *(デフォルト)* スライドショーを自動開始する。

ダッシュボードページ

ダッシュボードページオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
dashboard_pageid	ID	ダッシュボードページの ID。 プロパティの動作: - 読み取り専用
name	string	ダッシュボードページ名。 デフォルト：空文字列。
display_period	integer	ダッシュボードページの表示期間（秒単位）。 指定可能な値：0, 10, 30, 60, 120, 600, 1800, 3600。 デフォルト：0（デフォルトのページ表示期間を使用）。
widgets	array	ダッシュボードウィジェットオブジェクトの配列。

ダッシュボードウィジェット

ダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
type	string	ダッシュボードウィジェットのタイプ。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリーステータス; favgraphs - お気に入りグラフ; favmaps - お気に入りマップ; gauge - ゲージ; geomap - ジオマップ; graph - グラフ（クラシック）; graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホスト可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemhistory - アイテム履歴; itemnavigator - アイテムナビゲーター; item - アイテム値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; problemhosts - 問題のあるホスト; problems - 問題; problemsbysv - 深刻度別の問題; scatterplot - 散布図; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - トップホスト; topitems - トップアイテム; toptriggers - トップトリガー; trigover - トリガー概要; url - URL; web - Web 監視。 プロパティの動作: - 必須
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左端からの水平位置。 指定可能な値は 0 から 71 まで。
y	integer	ダッシュボードの上端からの垂直位置。 指定可能な値は 0 から 63 まで。
width	integer	ウィジェットの幅。 指定可能な値は 1 から 72 まで。
height	integer	ウィジェットの高さ。 指定可能な値は 1 から 64 まで。
view_mode	integer	ウィジェットの表示モード。 指定可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダー非表示;
fields	array	ダッシュボードウィジェットフィールドオブジェクトの配列。 プロパティの動作: - ダッシュボードウィジェットフィールドの各ウィジェットを参照

ダッシュボードウィジェットフィールド

ダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	ウィジェットフィールドのタイプ。 可能な値: 0 - 整数; 1 - 文字列; 2 - ホストグループ; 3 - ホスト; 4 - アイテム; 5 - アイテムプロトタイプ; 6 - グラフ; 7 - グラフプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 プロパティの動作: - 必須
name	string	ウィジェットフィールド名。 可能な値: ダッシュボードウィジェットフィールドを参照。 プロパティの動作: - 必須
value	mixed	タイプに応じたウィジェットフィールドの値。 可能な値: ダッシュボードウィジェットフィールドを参照。 プロパティの動作: - 必須

ダッシュボードユーザーグループ

ユーザーグループに基づくダッシュボード権限のリスト。以下のプロパティがあります。

プロパティ	タイプ	説明
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 必須
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

ダッシュボードユーザー

ユーザーに基づくダッシュボード権限のリスト。次のプロパティがあります。

プロパティ	タイプ	説明
userid	ID	ユーザーの ID。 プロパティの動作: - 必須
permission	integer	権限レベルの種類。 可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

dashboard.create

説明

`object dashboard.create(object/array dashboards)`

このメソッドは、新しいダッシュボードを作成することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 作成するダッシュボード。

dashboard の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
pages	array	ダッシュボード用に作成されるダッシュボードページ。ダッシュボードページは、指定された順序と同じ順序で並べられます。 パラメーターの動作: - 必須
users	array	ダッシュボードユーザー共有はダッシュボード上に作成されます。
userGroups	array	ダッシュボードユーザーグループ共有はダッシュボード上に作成されます。

戻り値

(`object`) `dashboardids` プロパティの下に作成されたダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたダッシュボードの順序と一致します。

例

ダッシュボードの作成

単一のダッシュボードページで、タグ付きの障害ウィジェットと 2 種類の共有 (ユーザーグループとユーザー) を持つ、"My dashboard" という名前のダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
```

```
            {
                "widgets": [
                    {
                        "type": "problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "service"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "zabbix_server"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

参照

- Dashboard page
- Dashboard widget
- Dashboard widget field
- Dashboard user

- Dashboard user group

ソース

CDashboard::create() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

説明

`object dashboard.delete(array dashboardids)`

このメソッドは、ダッシュボードを削除することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたダッシュボードの ID を含むオブジェクトを返します。

例

複数のダッシュボードの削除

2 つのダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

ソース

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

説明

`integer/array dashboard.get(object parameters)`

このメソッドは、指定したパラメーターに応じたダッシュボードを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
dashboardids	ID/array	指定した ID のダッシュボードのみを返します。
selectPages	query	ダッシュボードページを正しい順序で持つpagesプロパティを返します。
selectUsers	query	ダッシュボードが共有されているユーザーを持つusersプロパティを返します。
selectUserGroups	query	ダッシュボードが共有されているユーザーグループを持つuserGroupsプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: dashboardid。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ID によるダッシュボードの取得

ダッシュボード"1" と"2" に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "dashboardid": "1",
            "name": "Dashboard",
            "userid": "1",
            "private": "0",
            "display_period": "30",
            "auto_start": "1",
            "users": [],
            "userGroups": [],
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "9",
                            "type": "systeminfo",
                            "name": "",
                            "x": "12",
                            "y": "8",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "8",
                            "type": "problemsbysv",
                            "name": "",
                            "x": "12",
                            "y": "4",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "7",
                            "type": "problemhosts",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "6",
                            "type": "discovery",
                            "name": "",
                            "x": "6",
                            "y": "9",
                            "width": "18",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "5",
                            "type": "web",
```

```
                        "name": "",
                        "x": "0",
                        "y": "9",
                        "width": "18",
                        "height": "4",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "4",
                        "type": "problems",
                        "name": "",
                        "x": "0",
                        "y": "3",
                        "width": "12",
                        "height": "6",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "3",
                        "type": "favmaps",
                        "name": "",
                        "x": "8",
                        "y": "0",
                        "width": "12",
                        "height": "3",
                        "view_mode": "0",
                        "fields": []
                    },
                    {
                        "widgetid": "1",
                        "type": "favgraphs",
                        "name": "",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "3",
                        "view_mode": "0",
                        "fields": []
                    }
                ]
            },
            {
                "dashboard_pageid": "2",
                "name": "",
                "display_period": "0",
                "widgets": []
            },
            {
                "dashboard_pageid": "3",
                "name": "Custom page name",
                "display_period": "60",
                "widgets": []
            }
        ]
    },
    {
        "dashboardid": "2",
        "name": "My dashboard",
        "userid": "1",
        "private": "1",
```

```
            "display_period": "60",
            "auto_start": "1",
            "users": [
                {
                    "userid": "4",
                    "permission": "3"
                }
            ],
            "userGroups": [
                {
                    "usrgrpid": "7",
                    "permission": "2"
                }
            ],
            "pages": [
                {
                    "dashboard_pageid": "4",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "10",
                            "type": "problems",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "2",
                                    "name": "groupids",
                                    "value": "4"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Dashboard page
- Dashboard widget
- Dashboard widget field
- Dashboard user
- Dashboard user group

ソース

CDashboard::get() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

説明

`object dashboard.update(object/array dashboards)`

このメソッドは、既存のダッシュボードを更新することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 更新されるダッシュボードのプロパティ。

dashboardid プロパティはダッシュボードごとに定義する必要があり、他のプロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティはすべて変更されません。

dashboard の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
pages	array	既存のダッシュボードページを置き換えるダッシュボードページ。 ダッシュボードページは、dashboard_pageid プロパティによって更新されます。dashboard_pageid プロパティのないオブジェクトに対して新しいダッシュボードページが作成され、再利用されない場合は既存のダッシュボードページが削除されます。ダッシュボードページは、指定された順序と同じ順序で並べられます。ダッシュボードページの指定されたプロパティのみが更新されます。
users	array	ダッシュボードユーザー共有を既存の要素と置き換えます。
userGroups	array	ダッシュボードユーザーグループ共有を既存の要素と置き換えます。

戻り値

(object) dashboardids プロパティの下にある更新されたダッシュボードの ID を含むオブジェクトを返します。

例

ダッシュボードの名前変更

ダッシュボードの名前を”SQL server status” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換えて、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードの所有者の変更

Admin と super admin のみが使用できます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
```

```
    },
    "id": 1
}
```

参照

- Dashboard page
- Dashboard widget
- Dashboard widget field
- Dashboard user
- Dashboard user group

ソース

CDashboard::update() in *ui/include/classes/api/services/CDashboard.php*.

Dashboard widget fields

このページには、ダッシュボードウィジェットのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値へのナビゲーションリンクが含まれています。

各ウィジェットのパラメータとプロパティ値を確認するには、サイドバーを使用して該当するウィジェットのページに移動してください。

1 Action log

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドでアクションログウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アクションログウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

アクションログウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	値
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
受信者	11	userids.0	ユーザーID。 注: 複数のユーザーを設定するには、プロパティ名の番号を増やして各ユーザーごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。

パラメーター		type	name	値
アクションメディアタイプステータス		12	actionids.0	アクションID。 注: 複数のアクションを設定するには、プロパティ名の番号を増やして各アクションごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
		13	mediatypeids.0	メディアタイプID。 注: 複数のメディアタイプを設定するには、プロパティ名の番号を増やして各メディアタイプごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
		0	statuses.0	0 - 進行中; 1 - 送信/実行済み; 2 - 失敗。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
検索文字列		1	message	任意の文字列値。
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換ウィジェット（参照パラメーターを"ABCDE"に設定）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターで期間のみを設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
エントリの並び替え		0	sort_triggers	3 - 時刻（昇順）; 4 - *(デフォルト)* 時刻（降順）; 5 - タイプ（昇順）; 6 - タイプ（降順）; 7 - ステータス（昇順）; 8 - ステータス（降順）; 11 - 受信者（昇順）; 12 - 受信者（降順）。
表示行数		0	show_lines	1-100 の範囲で指定可能。 デフォルト: 25。

例

次の例は、アクションログウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アクションログウィジェットの設定

時間でソート (昇順) したアクション操作の詳細の 10 のエントリを表示するアクションログウィジェットを設定します。さらに、ユーザー"1" に電子メールを送信しようとしたが失敗したアクション操作のみの詳細を表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "actionlog",
                        "name": "Action log",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "sort_triggers",
                                "value": 3
                            },
                            {
                                "type": 11,
                                "name": "userids.0",
                                "value": 1
                            },
                            {
                                "type": 13,
                                "name": "mediatypeids.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "statuses.0",
                                "value": 2
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
```

```
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

2 Clock

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで時計ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。時計ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

時計ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分
時間の形式	0	time_type	0 - *(デフォルト)* ローカル時間 1 - サーバー時間 2 - ホスト時間
時計のタイプ	0	clock_type	0 - *(デフォルト)* アナログ 1 - デジタル

時間の形式が” ホスト時間” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
アイテム	4	itemid.0	アイテムID。 パラメーターの動作: - 必須

時計のタイプが” デジタル” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
表示	0	show.0	1 - 日付 2 - *(デフォルト)* 時刻 3 - タイムゾーン 注意: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。

詳細設定

時計のタイプが” デジタル” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
背景色	1	bg_color	16 進数の色コード (例: FF0000)。 デフォルト: "" (空)。

日付

時計のタイプが” デジタル” に設定され、表示が” 日付” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	date_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	date_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。

時間

時計のタイプが” デジタル” に設定され、表示が” 時刻” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	time_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	time_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。
秒	0	time_sec	0 - 無効 1 - *(デフォルト)* 有効
形式	0	time_format	0 - *(デフォルト)* 24 時間 1 - 12 時間

タイムゾーン

時計のタイプが” デジタル” に設定され、表示が” タイムゾーン” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	tzone_bold	0 - *(デフォルト)* 無効 1 - 有効

パラメーター	type	name	value
色	1	tzone_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。
タイムゾーン	1	tzone_timezone	有効なタイムゾーン文字列 (例: Europe/Riga、system、UTC など)。サポートされているタイムゾーンの完全なリストについては、PHP ドキュメントを参照してください。 デフォルト: local。
フォーマット	0	tzone_format	パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート 0 - (デフォルト) 短縮形式 1 - 完全形式 パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート

例

次の例は、時計ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

時計ウィジェットの設定

カスタマイズされたデジタル時計にローカルの日付、時間、およびタイムゾーンを表示する時計ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "clock",
                        "name": "Clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "clock_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
```

```
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "date_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 0,
                                "name": "time_bold",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tzone_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "tzone_timezone",
                                "value": "Europe/Riga"
                            },
                            {
                                "type": 0,
                                "name": "tzone_format",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

3 Discovery status

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでディスカバリのステータスウィジェットを設定できます。

パラメーター

ディスカバリのステータスウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

例

次の例は、ディスカバリのステータスウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ディスカバリのステータスウィジェットの設定

ディスカバリのステータスウィジェットを更新間隔を 15 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "discovery",
                        "name": "Discovery status",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 900
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

4 Favorite graphs

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでお気に入りのグラフウィジェットを設定できます。

パラメーター

お気に入りのグラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのグラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのグラフウィジェットの設定

お気に入りのグラフウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favgraphs",
                        "name": "Favorite graphs",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

5 Favorite maps

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでお気に入りのマップウィジェットを設定できます。

パラメーター

お気に入りのマップウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのマップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのマップウィジェットの設定

お気に入りのマップウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favmaps",
                        "name": "Favorite maps",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

6 Gauge

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでゲージウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ゲージウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

ゲージウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
アイテム	4	itemid.0	アイテムの ID。 パラメーターの動作: - アイテム (ウィジェット) が設定されていない場合は必須

パラメーター		type	name	value
	アイテム (ウィジェット)	1	itemid._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。 パラメーターの動作: - アイテムが設定されていない場合は必須
最小値		1	min	任意の数値。サフィックス（例："1d"、"2w"、"4K"、"8G"）がサポートされています。 デフォルト: "0"。
最大値		1	max	任意の数値。サフィックス（例："1d"、"2w"、"4K"、"8G"）がサポートされています。 デフォルト: "100"。
値アーク		1	value_arc_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
アーク背景		1	empty_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
背景		1	bg_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
表示		0	show.0	1 - 説明; 2 - 値; 3 - 針; 4 - 目盛り; 5 - 値アーク。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 デフォルト: 1, 2, 4, 5。 以下の両方の場合、「針」と「目盛り」の値はサポートされません: - 値「値アーク」の表示用のダッシュボードウィジェットフィールドオブジェクトが設定されていない; - アーク表示詳細設定パラメーターが「無効」に設定されている。 表示オプションの詳細設定パラメーターは、該当する値のダッシュボードウィジェットフィールドオブジェクトが設定されていない場合はサポートされません。
ホストの上書き		1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターを"ABCDE" に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

Advanced configuration

Gauge ウィジェットでは、以下の詳細設定パラメータがサポートされています。

Note:
Thresholds プロパティ名（例：thresholds.0.color）の数字は、リスト内のしきい値の位置を示し、昇順でソートされます。ただし、しきい値が異なる順序で設定されている場合は、Zabbix フロントエンドでウィジェット設定を更新すると、値は昇順でソートされます。(例："thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5")

パラメータ	type	名前	値
角度	0	角度	指定可能な値: 180 *(デフォルト)* または 270
説明			
説明	1	説明	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {ITEM.NAME}
サイズ	0	desc_size	指定可能な値の範囲は 1-100 です。 デフォルト: 15
垂直位置	0	desc_v_pos	0 - 上; 1 - *(デフォルト)* 下
太字	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	desc_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)
値			
小数点	0	decimal_places	指定可能な値の範囲は 1-10 です。 デフォルト: 2
サイズ	0	value_size	指定可能な値の範囲は 1-100 です。 デフォルト: 25
太字	0	value_bold	0 - *(デフォルト)* 無効; 1 - 有効
Color	1	value_color	16 進数のカラーコード（例：FF0000） デフォルト："" （空）
単位			
単位（チェックボックス）	0	units_show	0 - 無効 1 - （デフォルト）有効
単位（値）	1	units	任意の文字列値 パラメータの動作: - 単位（チェックボックス）が” 有効” に設定されている場合にサポートされます
サイズ	0	units_size	有効な値の範囲は 1-100 です。 デフォルト: 25 パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合にサポートされます
太字	0	units_bold	0 - *(デフォルト)* 無効 1 - 有効 パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合にサポートされます

パラメータ	type	名前	値
位置	0	units_pos	0 - 値の前 1 - 値の上 2 - *(デフォルト)* 値の後 3 - 値の下 パラメータの動作: - 単位 (チェックボックス) が" 有効" に設定されている場合に サポートされます。 このパラメータは、次の時間関連の単位 のいずれかに設定されている場合、無視されます。: unixtime、uptime、s
色	1	units_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)
値の円弧			
円弧サイズ	0	value_arc_size	有効な値の範囲は 1-100 です。 デフォルト: 20
ニードル			
色	1	needle_color	16 進数のカラーコード（例: FF0000） デフォルト: ""（空） パラメータの動作: - ダッシュボードウィジェットのフィールドオブジェクトで、表示に" 値円弧" が設定されている場合、または表示円弧が" 有効" に設定されている場合にサポートされます。
スケール			
単位の表示	0	scale_show_units	0 - 無効; 1 - *(デフォルト)* 有効 パラメータの動作: - 単位 (チェックボックス) が" 有効" に設定され、ダッシュボードウィジェットの 表示のフィールドオブジェクトで値が" 値の円弧" に設定されているか、表示円弧が" 有効" に設定されている場合に サポートされます。
サイズ	0	scale_size	指定可能な値の範囲は 1-100 です。 デフォルト: 15 パラメータの動作: - 表示のダッシュボードウィジェットのフィールドオブジェクトで値が" 値の円弧" に設定されているか、表示円弧が" 有効" に設定されている場合に サポートされます。
小数点以下の桁数	0	scale_decimal_places	有効な値の範囲は 1-10 です。 デフォルト: 0 パラメータの動作: - ダッシュボードウィジェットのフィールドオブジェクトで、表示に"Value arc" が設定されている場合、または表示 *arc* が" 有効" に設定されている場合にサポートされます。
しきい値			
色	1	thresholds.0.color	16 進数のカラーコード (例: FF0000)
しきい値	1	thresholds.0.threshold	任意の数値サフィックス (例: "1d"、"2w"、"4K"、"8G") がサポートされています。
ラベルを表示	0	th_show_labels	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - しきい値が設定されており、ダッシュボードウィジェットの 表示フィールドオブジェクトに値"Value arc" が設定されているか、表示アークが" 有効" に設定されている場合は サポートされます。

パラメータ	type	名前	値
円弧 'を表示	0	th_show_arc	0 - (デフォルト) 無効; 1 - 有効
円弧のサイズ	0	th_arc_size	パラメータの動作: - しきい値が設定されている場合は サポートされます。 可能な値の範囲は 1-100 です。 デフォルト: 5 パラメータの動作: - 円弧の表示が" 有効" に設定されている場合は サポートされます。

例

次の例は、ゲージウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ゲージウィジェットの設定

アイテム"44474" (インターフェース enp0s3: 送信ビット数) の値を表示するゲージウィジェットを設定します。さらに、しきい値などの複数の詳細オプションを使用して、ウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "gauge",
                        "name": "Gauge",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 44474
                            },
                            {
                                "type": 1,
                                "name": "min",
                                "value": "100000"
                            },
                            {
                                "type": 1,
                                "name": "max",
                                "value": "1000000"
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
```

```
                {
                    "type": 0,
                    "name": "show.1",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "show.2",
                    "value": 3
                },
                {
                    "type": 0,
                    "name": "show.4",
                    "value": 4
                },
                {
                    "type": 0,
                    "name": "show.5",
                    "value": 5
                },
                {
                    "type": 0,
                    "name": "angle",
                    "value": 270
                },
                {
                    "type": 0,
                    "name": "desc_size",
                    "value": 10
                },
                {
                    "type": 0,
                    "name": "desc_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "decimal_places",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "value_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "units_size",
                    "value": 15
                },
                {
                    "type": 0,
                    "name": "units_pos",
                    "value": 3
                },
                {
                    "type": 1,
                    "name": "needle_color",
                    "value": "3C3C3C"
                },
                {
                    "type": 1,
```

```
                            "name": "thresholds.0.color",
                            "value": "FF465C"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.threshold",
                            "value": "700000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "FFD54F"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.threshold",
                            "value": "500000"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.color",
                            "value": "0EC9AC"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.threshold",
                            "value": "100000"
                        },
                        {
                            "type": 0,
                            "name": "th_show_labels",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "th_show_arc",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "th_arc_size",
                            "value": 15
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"id": 1
```

```
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

7 Geomap

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで地理マップウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。地理マップウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

Geomap ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	type	name	値
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	値
ホスト	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメータは未設定、または指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクタをホストのデータソースとして設定します; ABCDE._hostids - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
タグ	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 (*Tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 (*Tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	値
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 (*Tags* を設定する場合)
初期表示クラスタリング		1	default_view	このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。 カンマ区切りの緯度、経度、ズームレベル（省略可能、0-30 の範囲）。 例: `40.6892494,-74.0466891,10`。
	クラスタリングモード	0	clustering_mode	0 - *(デフォルト)* 自動; 1 - ズームレベル。
	クラスタリングズームレベル	0	clustering_zoom_level	0-30 の範囲。 デフォルト: 0。 パラメータの動作: - サポート (*Clustering mode* が”Zoom level” の場合)
Reference		1	reference	5 文字からなる任意の文字列値（例: `ABCDE` や `JBPNL`）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメータの動作: - 必須

例

次の例は、地理マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

地理マップウィジェットの設定

以下のタグ設定に基づいて、ホストグループ”2” と”22” のホストを表示する地理マップウィジェットを設定します。”component” という名前のタグに”node” という値が含まれる、または”location” という名前のタグに”New York” という値と等しいもの。さらに、マップの初期表示を座標”40.6892494”(緯度)、”-74.0466891”(経度)、ズームレベル”10” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
```

```
                    "type": "geomap",
                    "name": "Geomap",
                    "x": 0,
                    "y": 0,
                    "width": 36,
                    "height": 5,
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 2,
                            "name": "groupids.0",
                            "value": 22
                        },
                        {
                            "type": 2,
                            "name": "groupids.1",
                            "value": 2
                        },
                        {
                            "type": 1,
                            "name": "default_view",
                            "value": "40.6892494,-74.0466891,10"
                        },
                        {
                            "type": 0,
                            "name": "evaltype",
                            "value": 2
                        },
                        {
                            "type": 1,
                            "name": "tags.0.tag",
                            "value": "component"
                        },
                        {
                            "type": 0,
                            "name": "tags.0.operator",
                            "value": 0
                        },
                        {
                            "type": 1,
                            "name": "tags.0.value",
                            "value": "node"
                        },
                        {
                            "type": 1,
                            "name": "tags.1.tag",
                            "value": "location"
                        },
                        {
                            "type": 0,
                            "name": "tags.1.operator",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "tags.1.value",
                            "value": "New York"
                        }
                    ]
                }
            ]
        }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

8 Graph

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでグラフウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。グラフウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

パラメーター	type	name	value
参照	1	reference	5 文字で構成される文字列値 (例: ABCDE または JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

データセット

データセットの設定には、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の数字（例: ds.0.hosts.0, ds.0.items.0）は特定のデータセットを表し、2 つ目の数字（存在する場合）は設定されたホストまたはアイテムを表します。

パラメータ		type	name	値
データセットタイプ		0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。
	アイテム	4	ds.0.itemids.0	アイテムの ID。 ウィジェットをテンプレートダッシュボードで設定する場合、テンプレートで設定されたアイテムのみを指定する必要があります。 注: 複数のアイテムを設定するには、プロパティ名の番号を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテム（ウィジェット）が設定されていない場合は必須
	アイテム（ウィジェット）	1	ds.0.itemids.0._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注: 複数のウィジェットを設定するには、プロパティ名の番号を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテムが設定されていない場合は必須
	色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合は必須

パラメータ		type	name	値
	ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例: ”Zabbix*”）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
	アイテムパターン	1	ds.0.items.0	このパラメータは、ウィジェットをテンプレートダッシュボードで設定する場合はサポートされません。 アイテム名またはパターン（例: ”*: Number of processed *values per second”）。 ウィジェットをテンプレートダッシュボードで設定する場合、テンプレートで設定されたアイテムのパターンのみを指定する必要があります。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
	色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合はサポート
	カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 0-11 の値が指定可能。 デフォルト: 0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合はサポート
描画		0	ds.0.type	0 - *(デフォルト)* 折れ線; 1 - 点; 2 - ステアケース; 3 - 棒グラフ。
積み上げ		0	ds.0.stacked	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 描画が「折れ線」「ステアケース」「棒グラフ」に設定されている場合はサポート
幅		0	ds.0.width	1-10 の値が指定可能。 デフォルト: 1。 パラメータの動作: - 描画が「折れ線」または「ステアケース」に設定されている場合はサポート
点のサイズ		0	ds.0.pointsize	1-10 の値が指定可能。 デフォルト: 3。 パラメータの動作: - 描画が「点」に設定されている場合はサポート
透明度		0	ds.0.transparency	1-10 の値が指定可能。 デフォルト: 5。

パラメータ	type	name	値
塗りつぶし	0	ds.0.fill	1-10 の値が指定可能。 デフォルト: 3。 パラメータの動作: - 描画が「折れ線」または「ステアケース」に設定されている場合はサポート
欠損データ	0	ds.0.missingdatafunc	0 - *(デフォルト)* なし; 1 - 接続; 2 - 0 として扱う; 3 - 最後の値。 パラメータの動作: - 描画が「折れ線」または「ステアケース」に設定されている場合はサポート
ホストの上書き	1	ds.0.override_hostid	ABCDE - 互換性のあるウィジェット（”参照”パラメータを”ABCDE”に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定。 このパラメータは、ウィジェットをテンプレートダッシュボードで設定する場合はサポートされません。
Y軸	0	ds.0.axisy	0 - *(デフォルト)* 左; 1 - 右。
タイムシフト	1	ds.0.timeshift	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスが使用可能。負の値も許可されます。 デフォルト: ""（空）。
集約関数	0	ds.0.aggregate_function	0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last。
集約間隔	1	ds.0.aggregate_interval	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスが使用可能。 デフォルト: 1h。
集約	0	ds.0.aggregate_grouping	0 - *(デフォルト)* 各アイテム; 1 - データセット。 パラメータの動作: - 集約関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合はサポート
近似	0	ds.0.approximation	1 - min; 2 - *(デフォルト)* avg; 4 - max; 7 - all。

パラメータ	type	name	値
データセットラベル	1	ds.0.data_set_label	任意の文字列値。 デフォルト: ""（空）。

Display options

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド
シンプルトリガー稼働時間パーセンタイル線(左)	0	simple_triggers	0 - *(デフォルト)* 無効; 1 - 有効
ステータス	0	working_time	0 - *(デフォルト)* 無効; 1 - 有効
	0	percentile_left	0 - *(デフォルト)* 無効; 1 - 有効 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 左" に設定されている場合に サポートされます。

パラメータ		type	name	value
パーセンタイル線 (右)	値	0	percentile_left_value	指定可能な値の範囲は 1-100 です。 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 左" に設定されている場合に サポートされます。
	ステータス	0	percentile_right	0 - *(デフォルト)* 無効; 1 - 有効 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 右" に設定されている場合に サポートされます。
	値	0	percentile_right_value	可能な値の範囲は 1-100 です。 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 右" に設定されている場合に サポートされます。

期間

期間の設定には、以下のパラメーターがサポートされています。

パラメーター		type	name	値
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターで期間のみを設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート

Axes

軸の設定には、以下のパラメーターがサポートされています。

パラメーター	type	name	value
左 *Y* 軸	0	lefty	0 - 無効 1 - *(*デフォルト*)* 有効 パラメーターの動作: - *Y* 軸 (データセット 設定内) が" 左" に設定されている場合に サポートされます。
右 *Y*	0	righty	0 - *(*デフォルト*)* 無効; 1 - 有効 パラメータの動作: - *Y* 軸 (データセット 設定内) が" 右" に設定されている場合に サポートされます。
スケール	0	lefty_scale righty_scale	0 - *(*デフォルト*)* 線形; 1 - 対数
最小値	1	lefty_min righty_min	任意の数値 デフォルト: "" (空)
最大値	1	lefty_max righty_max	任意の数値 デフォルト: "" (空)
単位 (タイプ)	0	lefty_units righty_units	0 - *(*デフォルト*)* 自動; 1 - 静的
単位 (値)	1	lefty_static_units righty_static_units	任意の文字列値。 デフォルト: "" (空)
X 軸	0	xaxis	0 - 無効; 1 - *(*デフォルト*)* 有効

凡例

凡例の設定に対応するパラメーターは以下の通りです。

パラメーター	型	名前	値
凡例を表示	0	legend	0 - 無効; 1 - *(*デフォルト*)* 有効
最小/平均/最大を表示	0	legend_statistic	0 - *(*デフォルト*)* 無効; 1 - 有効。 パラメーターの動作: - 凡例を表示が「有効」の場合に対応
集約関数を表示	0	legend_aggregation	0 - *(*デフォルト*)* 無効; 1 - 有効。 パラメーターの動作: - 凡例を表示が「有効」の場合に対応
行数	0	legend_lines_mode	0 - *(*デフォルト*)* 固定; 1 - 可変。 パラメーターの動作: - 凡例を表示が「有効」の場合に対応

パラメーター	型	名前	値
行数/ 最大行数	0	legend_lines	1-10 の範囲で指定可能。 デフォルト: 1。 パラメーターの動作: - 凡例を表示が「有効」の場合に対応
列数	0	legend_columns	1-4 の範囲で指定可能。 デフォルト: 4。 パラメーターの動作: - 凡例を表示が「有効」かつ最小/平均/最大を表示が「無効」の場合に対応

Problems

問題の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
問題を表示	0	show_problems	0 - (デフォルト) 無効; 1 - (デフォルト) 有効
選択された項目のみ	0	graph_item_problems	0 - (デフォルト) 有効
問題のあるホスト	1	problemhosts.0	ホスト 名 注: プロパティ名の数字は、設定されたホストを参照します。 複数のホストを設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。
重大度	0	severities.0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトを値ごとに作成します。
問題	1	problem_name	問題イベント名 (大文字と小文字は区別されません。完全な名前またはその一部)
問題タグ			

パラメータ	type	name	value
評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or
タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 問題タグを構成する場合は 必須
演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須
タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須

Overrides

オーバーライドの設定には、以下のパラメーターがサポートされています。

Note:
プロパティ名の最初の数字（例: or.0.hosts.0、or.0.items.0）は特定のデータセットを表し、2 番目の数字は（存在する場合）設定されたホストまたはアイテムを表します。

パラメーター	type	名前	値
ホストパターン	1	または.0.hosts.0	ホスト の名前またはパターン (例: Zabbix*) このパラメーターは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。 パラメーターの動作: - オーバーライドを設定する場合は 必須
アイテムパターン	1	または.0.items.0	アイテム の名前またはパターン (例: *: 1 秒あたりに処理される * 値の数) テンプレートダッシュボード でウィジェットを設定する場合は、テンプレートで設定されているアイテムのパターンのみを設定する必要があります。 パラメーターの動作: - オーバーライドを設定する場合は 必須
基本色	1	または.0.color	16 進数カラーコード (例: FF0000)
幅	0	または.0.width	指定可能な値の範囲は 1-10 です。
描画	0	or.0.type	0 - 線; 1 - ポイント; 2 - 階段; 3 - 棒グラフ

パラメーター	type	名前	値
透明度	0	or.0.transparency	有効な値の範囲は 1-10 です。
塗りつぶし	0	or.0.fill	有効な値の範囲は 1-10 です。
ポイントサイズ	0	or.0.pointsize	有効な値の範囲は 1-10 です。
欠損データ	0	or.0.missingdatafunc	0 - なし; 1 - 接続済み; 2 - 0 として処理; 3 - 既知の最後
Y 軸	0	or.0.axisy	0 - 左; 1 - 右
時間シフト	1	または.0.timeshift	有効な時間文字列（例：3600、1h など） 時間接尾辞を使用できます。負の値も使用できます。

例

次の例は、グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフウィジェットの設定

グラフウィジェットを次のように設定します。

- 1 つのホストに合計 9 つのアイテムの 2 つのデータセットがあります。
- 最初のデータセットはタイプが" アイテムリスト " で、3 つのアイテムで構成され、異なる色で幅、透明度および塗りつぶしは同じ線で表されます。
- 2 番目のデータセットはタイプが" アイテムパターン " で、6 つのアイテムで構成され、集計を設定されており、カスタムの色、幅、透明度および塗りつぶしの線で表されます。
- 2 番目のデータセットには、カスタムデータセットラベルもあります。
- グラフのデータは、過去 3 時間のデータが表示されます。
- グラフ内の障害は、設定されたアイテムについてのみ表示されます。
- グラフには 2 つの Y 軸があり、右側の Y 軸には 2 番目のデータセットの値のみが表示されます。
- グラフの凡例には、設定されたアイテムが 4 行で表示され、データセットの最小値、最大値、平均値を表します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23264
                            },
                            {
```

```
            "type": 1,
            "name": "ds.0.color.1",
            "value": "FF0000"
        },
        {
            "type": 4,
            "name": "ds.0.itemids.2",
            "value": 23269
        },
        {
            "type": 1,
            "name": "ds.0.color.2",
            "value": "BF00FF"
        },
        {
            "type": 4,
            "name": "ds.0.itemids.3",
            "value": 23257
        },
        {
            "type": 1,
            "name": "ds.0.color.3",
            "value": "0040FF"
        },
        {
            "type": 0,
            "name": "ds.0.width",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.0.transparency",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.0.fill",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "Zabbix server"
        },
        {
            "type": 1,
            "name": "ds.1.items.0",
            "value": "*: Number of processed *values per second"
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "000000"
        },
        {
            "type": 0,
            "name": "ds.1.transparency",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.1.fill",
```

```
                            "value": 0
                        },
                        {
                            "type": 0,
                            "name": "ds.1.axisy",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "ds.1.aggregate_function",
                            "value": 3
                        },
                        {
                            "type": 1,
                            "name": "ds.1.aggregate_interval",
                            "value": "1m"
                        },
                        {
                            "type": 0,
                            "name": "ds.1.aggregate_grouping",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "ds.1.data_set_label",
                            "value": "Number of processed values per second"
                        },
                        {
                            "type": 0,
                            "name": "graph_time",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "time_period.from",
                            "value": "now-3h"
                        },
                        {
                            "type": 0,
                            "name": "legend_statistic",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "legend_lines",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "show_problems",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "YZABC"
                        }
                    ]
                }
            ]
        }
    ],
```

```
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

9 Graph (classic)

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドでグラフ（クラシック）ウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。グラフ（クラシック）ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

グラフ（クラシック）ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	値
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ソースグラフ		0	source_type	0 - *(デフォルト)* グラフ; 1 - シンプルグラフ。
		6	graphid.0	グラフの ID。 パラメーターの動作: - ソースが「グラフ」に設定されている場合は必須
	グラフ（ウィジェット）	1	graphid._reference	グラフの ID の代わりに: ABCDE._graphid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をグラフのデータソースとして設定します。 パラメーターの動作: - ソースが「シンプルグラフ」に設定され、グラフが設定されていない場合は必須
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - ソースが「シンプルグラフ」に設定され、アイテム（ウィジェット）が設定されていない場合は必須
	アイテム（ウィジェット）	1	itemid._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をアイテムのデータソースとして設定します。 パラメーターの動作: - ソースが「シンプルグラフ」に設定され、アイテムが設定されていない場合は必須
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターで期間のみを設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート

パラメーター	type	name	値
凡例を表示	0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされていません。
参照	1	reference	任意の 5 文字からなる文字列値（例：ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、グラフ（クラシック）ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフ（クラシック）ウィジェットの設定

アイテム"42269" のシンプルなグラフを表示するグラフ（クラシック）ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "name": "Graph (classic)",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 1
                            },
                            {
                                "type": 4,
                                "name": "itemid.0",
```

```
                                "value": 42269
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "RSTUV"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

10 Graph prototype

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドでグラフのプロトタイプウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。グラフのプロトタイプウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

グラフプロトタイプウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
リフレッシュ間隔		0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ソース		0	source_type	2 - *(デフォルト)* グラフプロトタイプ; 3 - シンプルグラフプロトタイプ。
グラフプロトタイプ		7	graphid.0	グラフプロトタイプの ID。 パラメーターの動作: - 必須 - ソースが「グラフプロトタイプ」に設定されている場合
アイテムプロトタイプ		5	itemid.0	アイテムプロトタイプの ID。 パラメーターの動作: - 必須 - ソースが「シンプルグラフプロトタイプ」に設定されている場合
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターのみで期間を設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - サポート - 期間が設定されていない場合
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - サポート - 期間が設定されていない場合
凡例を表示		0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。

パラメーター	type	name	value
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
列数	0	columns	1～24 の値が設定可能。 デフォルト: 2。
行数	0	rows	1～16 の値が設定可能。 デフォルト: 1。
参照	1	reference	5 文字からなる任意の文字列（例：ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、グラフのプロトタイプウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフのプロトタイプウィジェットの設定

ローレベルディスカバリによってアイテムのプロトタイプ (ID: "42316") から作成された 3 つのグラフ (3 列、1 行) のグリッドを表示するグラフのプロトタイプウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graphprototype",
                        "name": "Graph prototype",
                        "x": 0,
                        "y": 0,
                        "width": 48,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 3
                            },
                            {
                                "type": 5,
```

```
                                "name": "itemid.0",
                                "value": 42316
                            },
                            {
                                "type": 0,
                                "name": "columns",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "OPQWX"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

11 Honeycomb

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでハニカムウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ハニカムウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

Honeycomb ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
リフレッシュ間隔		0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ				

パラメーター		type	name	value
	評価タイプ	0	evaltype_host	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
アイテムパターンアイテムタグ		1	items.0	アイテム名またはパターン。 注: 複数のアイテムパターンを設定するには、プロパティ名の番号を増やして各アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメーターの動作: - 必須
	評価タイプ	0	evaltype_item	0 - *(デフォルト)* And/Or; 2 - Or。

パラメーター		type	name	value
	タグ名	1	item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
	タグ値	1	item_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。
表示		0	show.0	1 - プライマリラベル; 2 - セカンダリラベル。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 デフォルト: 1, 2。
Reference		1	reference	5 文字からなる任意の文字列値（例: ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

Advanced configuration

Honeycomb ウィジェットでは、以下の詳細設定パラメータがサポートされています。

Note:
Thresholds プロパティ名（例：thresholds.0.color）の数字は、リスト内のしきい値の位置を示し、昇順でソートされます。ただし、しきい値が異なる順序で設定されている場合は、Zabbix フロントエンドでウィジェット設定を更新すると、値は昇順でソートされます。(例："thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5")

パラメータ	type	名前	値
プライマリラベルタイプ	0	primary_label_type	0 - *(デフォルト)* テキスト; 1 - 値
テキスト	1	primary_label	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {HOST.NAME> パラメータの動作: - タイプが" テキスト " に設定されている場合に サポートされます。
小数点以下の桁数	0	primary_label_decimal_places	有効な値の範囲は 0-6 です。 デフォルト: 2 パラメータの動作: - *Type* が"Value" に設定されている場合にサポートされます。
Size (type)	0	primary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム
Size	0	primary_label_size	有効な値の範囲は 1-100 です。 デフォルト: 20 パラメータの動作: - *Size* (type) が"Custom" に設定されている場合にサポートされます。
太字	0	primary_label_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	primary_label_color	16 進数カラーコード (例: FF0000) デフォルト: 設定オブジェクトとユーザーオブジェクトの theme に基づきます。 1F2C33 (blue-theme) または"hc-light" EEEEEE (dark-theme) または"hc-dark"
単位 (チェックボックス)	0	primary_label_units_show	0 無効; 1 - *(デフォルト)* 有効 パラメータの動作: - タイプが" 値" に設定されている場合に サポートされます。
単位 (値)	1	primary_label_units	任意の文字列値 "" (空) パラメータの動作: - タイプが" 値" に設定され、単位 (チェックボックス) が" 有効" に設定されている場合に サポートされます。
位置	0	primary_label_units_pos	0 - 前の値 1 - *(デフォルト)* 後の値 パラメータの動作: - タイプが" 値" に設定され、単位 (チェックボックス) が" 有効" に設定されている場合にサポートされます。 このパラメータは、次の時間関連の単位のいずれかに設定されている場合、無視されます。: unixtime、uptime、s

パラメータ	type	名前	値
セカンダリラベルタイプ	0	secondary_label_type	0 - テキスト 1 - *(デフォルト)* 値
テキスト	1	secondary_label	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {{ITEM.LASTVALUE}.fmtnum(2)> パラメータの動作: - *Type* が”Text” に設定されている場合に サポートされます。
小数点以下の桁数	0	secondary_label_decimal_places	可能な値の範囲は 0-6 です。 デフォルト: 2 パラメータの動作: - *Type* が”Value” に設定されている場合に サポートされます。
サイズ (type)	0	secondary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム
サイズ	0	secondary_label_size	有効な値の範囲は 1-100 です。 デフォルト: 30 パラメータの動作: - サイズ (type) が” カスタム” に設定されている場合に サポートされます。
太字	0	secondary_label_bold	0 - 無効; 1 - *(デフォルト)* 有効
色	1	secondary_label_color	16 進数カラーコード（例：FF0000） デフォルト：設定オブジェクトおよびユーザーオブジェクトの `theme` に基づきます。 ”blue-theme” または”hc-light” の場合は `1F2C33`、 ”dark-theme” または”hc-dark” の場合は `EEEEEE`
単位（チェックボックス）	0	secondary_label_units_show	0 - 無効 1 - （デフォルト）有効 パラメータの動作: - *Type* が”Value” に設定されている場合にサポートされます。
単位 (値)	1	secondary_label_units	任意の文字列値 "" (空) パラメータの動作: - タイプが” 値” に設定され、単位 (チェックボックス) が” 有効” に設定されている場合に サポートされます。
位置	0	secondary_label_position	0 - 前の値 1 - *(デフォルト)* 後の値 パラメータの動作: - タイプが” 値” に設定され、単位 (チェックボックス) が” 有効” に設定されている場合に サポートされます。 このパラメータは、次の時間関連の単位 のいずれかに設定されている場合、無視されます。: unixtime、uptime、s
背景色			

パラメータ	type	名前	値
背景色	1	bg_color	16 進数カラーコード（例：FF0000） デフォルト：設定オブジェクトおよびユーザーオブジェクトの theme に基づきます。 ”blue-theme” の場合は D9E7ED、 ”dark-theme” の場合は 3D5059、 ”hc-light” の場合は AAD7E9、 ”hc-dark” の場合は 335463
しきい値			
色補間	0	補間	0 - 無効、 1 - *(デフォルト)* 有効
Color	1	thresholds.0.color	16 進数のカラーコード（例：FF0000）
Threshold	1	thresholds.0.threshold	任意の数値サフィックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています。

例

次の例は、ハニカムウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ハニカムウィジェットの設定

Zabbix サーバープロセスの使用率を表示するハニカムウィジェットを設定します。さらに、ハニカムセルのプライマリラベルを変更し、しきい値を設定してウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "honeycomb",
                        "name": "Honeycomb",
                        "x": "0",
                        "y": "0",
                        "width": "24",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "items.0",
                                "value": "Zabbix server: Utilization*"
                            },
                            {
                                "type": 1,
```

```
                                "name": "primary_label",
                                "value": "{ITEM.NAME}"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "0EC9AC"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "FFD54F"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "70"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.color",
                                "value": "FF465C"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.threshold",
                                "value": "90"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "KSTMQ"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

12 Host availability

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホスト稼働状況ウィジェットを設定できます。

Parameters

ホスト可用性ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。

パラメーター	type	name	value
インターフェースタイプ	0	interface_type.0	0 - なし; 1 - Zabbix エージェント (パッシブチェック); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix エージェント (アクティブチェック) デフォルト: 1、2、3、4、5 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。
レイアウト	0	layout	0 - *(デフォルト)* 水平; 1 - 垂直
メンテナンス中のホストを表示	0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効
合計のみを表示	0	only_totals	0 - *(デフォルト)* 無効; 1 - 有効

例

次の例は、ホスト稼働状況ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ホスト稼働状況ウィジェットの設定

”Zabbix エージェント” および”SNMP” インターフェースが設定されているホストグループ”4” のホストの可用性情報を (縦型レイアウトで) 表示するホスト稼働状況ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
```

```
            {
                "widgets": [
                    {
                        "type": "hostavail",
                        "name": "Host availability",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

13 Host card

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホストカードウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ホストカードウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

ホストカードウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホスト	3	hostid.0	ホストの ID。 パラメータの動作: - 必須（ホスト（ウィジェット/ダッシュボード）が設定されていない場合） このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト（ウィジェット/ダッシュボード）	1	hostid._reference	ホストの ID の代わりに： DASHBOARD.hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定; ABCDE._hostid - 互換性のあるウィジェット（参照パラメータが"ABCDE"に設定されているもの）をホストのデータソースとして設定。 パラメータの動作: - 必須（ホストが設定されていない場合） このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ	type	name	value
抑制された障害を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
表示	0	sections.0	0 - ホストグループ; 1 - 説明; 2 - 監視; 3 - 可用性; 4 - 監視サーバー; 5 - テンプレート; 6 - インベントリ; 7 - タグ。 注：プロパティ名の数字はセクションリスト内のセクション順序を参照します。複数のセクションを設定するには、各セクションごとにプロパティ名の数字を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。

表示が「インベントリ」に設定されている場合、以下のパラメータがサポートされます。

パラメータ	type	name	value
インベントリフィールド	0	inventory.0	インベントリの ID。 注：複数のインベントリフィールドを設定するには、各インベントリフィールドごとにプロパティ名の数字を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。

例

次の例は、*Host card* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

Configuring a *Host card* widget

”監視”、”可用性”、”監視元”、”インベントリ”、”タグ” の各セクションを表示する ホストカードウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
```

```
                    "y": 0,
                    "width": 14,
                    "height": 7,
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 3,
                            "name": "hostid.0",
                            "value": 10084
                        },
                        {
                            "type": 0,
                            "name": "show_suppressed",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "sections.0",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "sections.1",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "sections.2",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "sections.3",
                            "value": 6
                        },
                        {
                            "type": 0,
                            "name": "sections.4",
                            "value": 7
                        },
                        {
                            "type": 0,
                            "name": "inventory.0",
                            "value": 25
                        },
                        {
                            "type": 0,
                            "name": "inventory.1",
                            "value": 26
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
```

```
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

14 Host navigator

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホストナビゲーターウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ホストナビゲーターウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

ホストナビゲーターウィジェットでは、以下のパラメータがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分

パラメータ		type	name	value
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	ホストグループ (ウィジェット)	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (その *Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホストパターン		1	hosts.0	ホスト の名前またはパターン 注: 複数のホストパターンを構成するには、プロパティ名に増分番号を含む、ホストパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストパターンの場合、パラメータ ホストグループはまったく設定しないか、設定されたホストパターンに一致するホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
ホストステータス		0	ステータス	-1 - *(デフォルト)* 任意; 0 - 有効; 1 - 無効 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
ホストタグ	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。

パラメータ		type	name	value
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを構成する場合は 必須 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
重大度		0	severities.0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効) 注: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。
問題を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制なし; 2 - なし。
グループ化				

パラメータ		type	name	value
	属性	0	group_by.0.attribute	0 - ホストグループ; 1 - タグ値; 2 - 重大度 注: プロパティ名の数字は、グループ化属性リスト内の属性の順序を示します。 パラメータの動作: - グループ化を構成する場合は 必須
	値	1	group_by.0.tag_name	任意の文字列値 注: プロパティ名に含まれる数字は、*Attribute* パラメータで設定されたグループ化属性を参照します。 パラメータの動作: - *Group by* を設定し、*Attribute* が” タグ値” に設定されている場合は 必須
ホスト制限		0	show_lines	有効な値の範囲は 1-9999 です。 デフォルト: 100 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
参照		1	reference	5 文字の任意の文字列値 (例: ABCDE または JBPNL) この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、ホストナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ホストナビゲーターウィジェットの設定

ホストグループ別にグループ化されたホストと、”city” タグの値別にグループ化されたホストを表示するホストナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostnavigator",
                        "name": "Host navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
```

```
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "city"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "SWKLB"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

15 アイテムカード

説明

これらのパラメータと、該当するダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値により、dashboard.create および dashboard.update メソッドでアイテムカードウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アイテムカードウィジェットの作成または更新を確実に行うには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

アイテムカードウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - *Item (Widget)* が設定されていない場合は必須
	Item (Widget)	1	itemid._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。 パラメーターの動作: - *Item* が設定されていない場合は必須
表示		0	sections.0	0 - 説明; 1 - エラーテキスト; 2 - メトリクス; 3 - 最新データ; 4 - 情報の種類; 5 - トリガー; 6 - ホストインターフェース; 7 - タイプ; 8 - ホストインベントリ; 9 - タグ。 注: プロパティ名の数字はセクションリスト内のセクション順序を参照します。複数のセクションを設定するには、プロパティ名の数字を増やして各セクションごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。

パラメーター	type	name	value
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 デフォルト: "" (空) このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

スパークライン

表示が「最新データ」に設定されている場合、以下のパラメータがサポートされます。

パラメータ	type	name	値
幅	0	sparkline.width	0-10 の範囲の値が可能です。 デフォルト: 1.
塗りつぶし色	0	sparkline.fill	0-10 の範囲の値が可能です。 デフォルト: 3.
	1	sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5.
期間	1	sparkline.time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクタをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE のもの）をデータソースとして設定。 デフォルト: ""（空） または、*From* および *To* パラメータでのみ期間を設定することもできます。
From	1	sparkline.time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 デフォルト: now-1h. パラメータの動作: - 期間が設定されていない場合にサポート
To	1	sparkline.time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 デフォルト: now. パラメータの動作: - 期間が設定されていない場合にサポート

パラメータ	type	name	値
履歴データ	0	sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。

例

以下の例は、アイテムカードウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定のみを説明することを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムカードウィジェットの設定

これらのセクションを表示するアイテムカードウィジェットを設定します: ”説明”、”最新データ”、”トリガー”、”タグ”。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
```

```
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

こちらもご参照ください

- ダッシュボードウィジェットフィールド
- dashboard.create
- dashboard.update

16 アイテムの履歴

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでアイテムヒストリウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテムヒストリウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

アイテム履歴ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	タイプ	名前	値
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
レイアウト	0	layout	0 - *(デフォルト)* 横方向; 1 - 縦方向。
カラム (下記参照)			
表示行数	0	show_lines	1～100 の範囲で指定可能。 デフォルト: 25。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターを"ABCDE" に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
詳細設定 (下記参照)			
参照	1	reference	5 文字からなる任意の文字列（例: ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

列

列には、アイテムパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメータはすべての列でサポートされています。

パラメータ	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメータの動作: - 必須
項目	4	columns.0.itemid	項目 ID テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートで設定されている項目のみを設定する必要があります。 パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

設定された *Item* が数値型の項目の場合、以下の列パラメータがサポートされます。

Parameter	type	name	value
Display	0	columns.0.display	1 - *(default)* そのまま; 2 - Bar; 3 - Indicators.
Min	1	columns.0.min	任意の数値 Parameter behavior: - *Display* が”Bar” または”Indicators” に設定されている場合に サポートされます。
最大	1	columns.0.max	任意の数値 パラメータの動作: - 表示が” バー” または” インジケーター” に設定されている場合にサポートされます。
しきい値			
色	1	columns.0.threshold.color	16進数のカラーコード（例：FF0000）
しきい値	1	columns.0.threshold.d	任意の数値サフィックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - 傾向

設定された項目が文字、テキスト、またはログタイプの項目の場合、以下の列パラメータがサポートされます。

パラメータ	type	name	value
ハイライト			
ハイライト	1	columns.0.highlights.color	16進数のカラーコード（例：FF0000）
しきい値	1	columns.0.highlights.pattern	任意の正規表現
表示	0	columns.0.display	1 - *(デフォルト)* そのまま; 4 - HTML; 5 - 単一行
単一行	0	columns.0.max_length	有効な値の範囲は 1～500 です。 デフォルト: 100 パラメータの動作: - *Display* が” 単一行” に設定されている場合に サポートされます
等幅フォントを使用する	0	columns.0.monospace_font	0 - *(デフォルト)* デフォルトフォントを使用する。 1 - 等幅フォントを使用する。

パラメータ	type	name	value
ローカル時刻を表示	0	columns.0.local_time	0 - *(デフォルト)* タイムスタンプを表示します。 1 - ローカル時刻を表示します。 パラメーターの動作: - *Item* がログタイプ項目に設定され、*Show timestamp* が”Enabled” に設定されている場合に サポートされます。

設定されたアイテムがバイナリタイプのアイテムの場合、以下の列パラメータがサポートされます。

パラメータ	type	名前	値
サムネイルを表示	1	columns.0.show_thumbnail	0 - *(デフォルト)* 無効; 1 - 有効

詳細な設定

アイテム履歴ウィジェットでは、以下の詳細な設定パラメータがサポートされています。

パラメータ	タイプ	名前	値
新しい値タイムスタンプを表示	0	sortorder	0 - *(デフォルト)* 上部; 1 - 下部。
	0	show_timestamp	0 - *(デフォルト)* 無効; 1 - 有効。
列ヘッダーを表示	0	show_column_header	0 - オフ; 1 - 横方向; 2 - *(デフォルト)* 縦方向。

パラメータ		タイプ	名前	値
期間		1	time_period._referenced	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが"ABCDE" に設定されているもの）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート

例

次の例は、アイテムヒストリウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムヒストリウィジェットの設定

2 つの数値アイテム"42269" と"42270" の最新データを表示するアイテムヒストリウィジェットを設定します。さらに、アイテム列を縦に、列名を横に表示するように設定します。表示データは 15 行に制限し、タイムスタンプ列を別に用意します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemhistory",
                        "name": "Item history",
                        "x": "0",
                        "y": "0",
                        "width": "18",
                        "height": "6",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "0",
                                "name": "layout",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "columns.0.name",
```

```
                                "value": "CPU utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.0.itemid",
                                "value": "42269"
                            },
                            {
                                "type": "1",
                                "name": "columns.1.name",
                                "value": "Memory utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.1.itemid",
                                "value": "42270"
                            },
                            {
                                "type": "0",
                                "name": "show_lines",
                                "value": "15"
                            },
                            {
                                "type": "0",
                                "name": "show_timestamp",
                                "value": "1"
                            },
                            {
                                "type": "0",
                                "name": "show_column_header",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "reference",
                                "value": "KIVKD"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

17 アイテムナビゲーター

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでアイテムナビゲーターウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテムナビゲーターウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

アイテムナビゲーターウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定であるか、または指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト (ウィジェット/ダッシュボード)	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostid` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostid` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ				
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Host tags* を設定する場合) このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Host tags* を設定する場合) このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Host tags* を設定する場合)
アイテムパターンアイテムタグ		1	items.0	このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。 アイテム名またはパターン。 注: 複数のアイテムパターンを設定するには、プロパティ名の番号を増やして各アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。
	評価タイプ	0	item_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Item tags* を設定する場合)
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Item tags* を設定する場合)
	タグ値	1	item_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (*Item tags* を設定する場合)
状態		0	state	-1 - *(デフォルト)* すべて; 0 - 正常; 1 - サポート外。

パラメーター		type	name	value
障害を表示グループ化		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されていない; 2 - なし。
	属性	0	group_by.0.attribute	0 - ホストグループ; 1 - ホスト名; 2 - ホストタグ値; 3 - アイテムタグ値。 注: プロパティ名の番号はグループ化属性リスト内の属性の順序を参照します。 パラメーターの動作: - 必須 (*Group by* を設定する場合)
	値	1	group_by.0.tag_name	任意の文字列値。 注: プロパティ名の番号は属性パラメーターで設定したグループ化属性を参照します。 パラメーターの動作: - 必須 (*Group by* を設定し、属性が「ホストタグ値」または「アイテムタグ値」に設定されている場合)
アイテム数上限		0	show_lines	1-9999 の範囲で指定可能。 デフォルト: 100。
Reference		1	reference	5 文字からなる任意の文字列値（例: ABCDE や JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、アイテムナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムナビゲーターウィジェットの設定

最大 1000 個のアイテムをホスト別、さらに” コンポーネント” アイテムタグ値別にグループ化して表示するアイテムナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
```

```
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemnavigator",
                        "name": "Item navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 1000
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "DFNLK"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

18 アイテム値

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでアイテムの値ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテムの値ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

アイテム値ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
アイテム	4	itemid.0	アイテムの ID。 パラメーターの動作: - *Item (Widget)* が設定されていない場合は必須
Item (Widget)	1	itemid._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。 パラメーターの動作: - *Item* が設定されていない場合は必須

パラメーター	type	name	value
表示	0	show.0	1 - 説明; 2 - 値; 3 - 時刻; 4 - 変化インジケーター; 5 - スパークライン。 デフォルト: 1, 2, 3, 4。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

高度な設定

アイテム値ウィジェットでは、以下の高度な設定パラメータがサポートされています。

Note:
しきい値プロパティ名の数字（例：thresholds.0.color）は、昇順でソートされたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix フロントエンドでウィジェット設定を更新した後、値は昇順にソートされます（例："thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5"）。

パラメータ	type	name	値
背景色	1	bg_color	16 進数のカラーコード（例：FF0000）。 デフォルト："" （空）。
しきい値			
色	1	thresholds.0.color	16 進数のカラーコード（例：FF0000）。
しきい値	1	thresholds.0.threshold	任意の文字列値。
集約関数	0	aggregate_function	0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.

パラメータ	type	name	値
期間	1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（その reference パラメータが ABCDE であるもの）をデータソースとして設定。 デフォルト：DASHBOARD._timeperiod または、*From* および *To* パラメータのみで期間を設定することもできます。 パラメータの動作： - 集約関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作： - 期間が設定されておらず、集約関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作： - 期間が設定されておらず、集約関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
履歴データ	0	history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

Description

表示が” 説明” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
説明	1	説明	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {ITEM.NAME}
水平位置	0	desc_h_pos	0 - 左 1 - *(デフォルト)* 中央 2 - 右 2 つ以上の要素 (説明、値、時間) で同じ 水平位置と 垂直位置を共有することはできません。
垂直位置	0	desc_v_pos	0 - 上; 1 - 中央; 2 - *(デフォルト)* 下 2 つ以上の要素 (説明、値、時刻) が同じ 水平位置と 垂直位置を共有することはできません。

パラメータ	type	名前	値
サイズ	0	desc_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15
太字	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	desc_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

Value

表示が” 値” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ		type	名前	値
小数点以下の桁数				
	小数点以下の桁数	0	decimal_places	有効な値の範囲は 1～10 です。 デフォルト: 2
	サイズ	0	decimal_size	有効な値の範囲は 1～100 です。 デフォルト: 35
位置				
	水平位置	0	value_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右 2 つ以上の要素 (説明、値、時間) が同じ 水平位置と 垂直位置を共有することはできません。
	垂直位置	0	value_v_pos	0 - 上; 1 - *(デフォルト)* 中央; 2 - 下 2 つ以上の要素 (説明、値、時間) が同じ 水平位置と 垂直位置を共有することはできません。
	サイズ	0	value_size	有効な値の範囲は 1～100 です。 デフォルト: 45
	太字	0	value_bold	0 - 無効; 1 - *(デフォルト)* 有効
	色	1	値 _ 色	16 進数のカラーコード（例：FF0000） デフォルト："" （空）
単位				
	単位（チェックボックス）	0	単位表示	0 - 無効 1 - （デフォルト）有効
	単位（値）	1	単位	任意の文字列値

パラメータ	type	名前	値
位置	0	単位 _ 位置	0 - 前の値 1 - 上の値 2 - （デフォルト）後の値 3 - 下の値
サイズ	0	単位 _ サイズ	可能な値の範囲は 1～100 です。 デフォルト：35
太字	0	units_bold	0 - 無効; 1 - *(デフォルト)* 有効
色	1	units_color	16 進数の色コード (例: FF0000) デフォルト: "" (空)

時間

表示が” 時間” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
水平位置	0	time_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右 2 つ以上の要素（説明、値、時間）が同じ水平位置と垂直位置を共有することはできません。
垂直位置	0	time_v_pos	0 - *(デフォルト)* 上; 1 - 中央; 2 - 下 2 つ以上の要素（説明、値、時間）が同じ水平位置と垂直位置を共有することはできません。
サイズ	0	time_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15
太字	0	time_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	time_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

変更インジケーター

表示が” 変更インジケーター” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
変更インジケーター ↑ の色	1	up_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）
変更インジケーター ↓ の色	1	down_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）
インジケーターの色変更	1	updown_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）

スパークライン

表示が「スパークライン」に設定されている場合、以下の詳細な設定パラメータがサポートされます。

パラメータ	type	name	value
幅	0	sparkline.width	0～10 の値が設定可能です。 デフォルト: 1
塗りつぶし	0	sparkline.fill	0～10 の値が設定可能です。 デフォルト: 3
色	1	sparkline.color	16 進数のカラーコード（例：FF0000）。 デフォルト: 42A5F5
期間	1	sparkline.time_period.from sparkline.time_period.to	DASHBOARD._timeperiod - ダッシュボードの期間セレクタをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE のもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。
From	1	sparkline.time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。
To	1	sparkline.time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。
履歴データ	0	sparkline.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

例

次の例は、アイテムの値ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムの値ウィジェットの設定

アイテム”42266”(Zabbix エージェントの可用性) のアイテム値を表示するアイテムの値ウィジェットを設定します。さらにウィジェットを複数の詳細オプションを使用して、Zabbix エージェントの可用性ステータスに基づいて変更される動的な背景色といった視覚的な調整をします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "item",
                        "name": "Item value",
                        "x": 0,
                        "y": 0,
                        "width": 12,
```

```
                "height": 3,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 4,
                        "name": "itemid.0",
                        "value": 42266
                    },
                    {
                        "type": 0,
                        "name": "show.0",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "show.1",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "show.2",
                        "value": 3
                    },
                    {
                        "type": 1,
                        "name": "description",
                        "value": "Agent status"
                    },
                    {
                        "type": 0,
                        "name": "desc_h_pos",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "desc_v_pos",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "desc_bold",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "desc_color",
                        "value": "F06291"
                    },
                    {
                        "type": 0,
                        "name": "value_h_pos",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "value_size",
                        "value": 25
                    },
                    {
                        "type": 1,
                        "name": "value_color",
                        "value": "FFFF00"
```

```
                        },
                        {
                            "type": 0,
                            "name": "units_show",
                            "value": 0
                        },
                        {
                            "type": 0,
                            "name": "time_h_pos",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "time_v_pos",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "time_size",
                            "value": 10
                        },
                        {
                            "type": 0,
                            "name": "time_bold",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "time_color",
                            "value": "9FA8DA"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.color",
                            "value": "E1E1E1"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.threshold",
                            "value": "0"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "D1C4E9"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.threshold",
                            "value": "1"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
```

```
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

19 マップ

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでマップウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。マップウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

マップウィジェットでは、以下のパラメーターがサポートされています。

パラメータ	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - (デフォルト) 15 分
マップ	8	sysmapid.0	マップID。 パラメーターの動作: - マップ (ウィジェット) が設定されていない場合、必須
マップ (ウィジェット)	1	sysmapid._reference	ABCDE._mapid - マップのデータソースとして、マップナビゲーションウィジェット (参照パラメーターが"ABCDE" に設定されている) を設定します。 パラメーターの動作: - マップが設定されていない場合、必須

パラメータ	type	name	value
参照	1	reference	5 文字からなる任意の文字列値 (例: ABCDE や JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

マップウィジェットの設定

マップ"1" を表示するマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 0,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 8,
                                "name": "sysmapid.0",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

リンクされたマップウィジェットの設定

マップナビゲーションウィジェットにリンクされたマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 5,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "sysmapid._reference",
                                "value": "ABCDE._mapid"
                            }
                        ]
                    },
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
```

```
            "value": "Element C"
        },
        {
            "type": 1,
            "name": "navtree.4.name",
            "value": "Element A1"
        },
        {
            "type": 1,
            "name": "navtree.5.name",
            "value": "Element A2"
        },
        {
            "type": 1,
            "name": "navtree.6.name",
            "value": "Element B1"
        },
        {
            "type": 1,
            "name": "navtree.7.name",
            "value": "Element B2"
        },
        {
            "type": 0,
            "name": "navtree.4.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.6.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.7.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.1.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.2.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.3.order",
            "value": 3
        },
        {
            "type": 0,
            "name": "navtree.4.order",
            "value": 1
        },
```

```
                            {
                                "type": 0,
                                "name": "navtree.5.order",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "navtree.6.order",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.7.order",
                                "value": 2
                            },
                            {
                                "type": 8,
                                "name": "navtree.6.sysmapid",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "ABCDE"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map navigation tree

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでマップナビゲーションウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。マップナビゲーションウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

マップナビゲーションツリーウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分
利用できないマップを表示	1	show_unavailable	0 - *(デフォルト)* 無効; 1 - 有効
参照	1	reference	5 文字の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

マップナビゲーションツリー要素の設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
名前	1	navtree.1.name	任意の文字列値 注: プロパティ名に含まれる数字は要素番号を設定します。
リンクされたマップ	8	navtree.1.sysmapid	マップ ID 注: プロパティ名に含まれる数字は、マップがリンクされている要素を参照します。
要素階層を作成するためのパラメーター	0	navtree.1.parent	親要素番号 注: プロパティ名に含まれる数字は、子要素を参照します。プロパティ値は親要素を参照します。
	0	navtree.1.order	マップナビゲーションツリー内の要素の位置 注: プロパティ名に含まれる数字は、要素番号を参照します。プロパティ値は、マップナビゲーションツリー内の要素の位置を参照します。親要素の位置は、マップナビゲーションツリー全体内で決定されます。子要素の位置は、親要素内で決定されます。

例

次の例は、マップナビゲーションウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

マップナビゲーションウィジェットの設定

以下のマップナビゲーションツリーを表示するマップナビゲーションウィジェットを設定します:

- 要素 A
  - 要素 A1
  - 要素 A2
- 要素 B
  - 要素 B1 (リンクされたマップウィジェットに表示できるリンクされたマップ"1" を含む)
  - 要素 B2
- 要素 C

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
                                "value": "Element C"
                            },
                            {
                                "type": 1,
                                "name": "navtree.4.name",
                                "value": "Element A1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.5.name",
                                "value": "Element A2"
                            },
                            {
                                "type": 1,
                                "name": "navtree.6.name",
                                "value": "Element B1"
                            },
                            {
                                "type": 1,
```

```
            "name": "navtree.7.name",
            "value": "Element B2"
        },
        {
            "type": 0,
            "name": "navtree.4.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.6.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.7.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.1.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.2.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.3.order",
            "value": 3
        },
        {
            "type": 0,
            "name": "navtree.4.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.6.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.7.order",
            "value": 2
        },
        {
            "type": 8,
            "name": "navtree.6.sysmapid",
            "value": 1
```

```
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "HJQXF"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map

21 パイチャート

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで円グラフウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。円グラフウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

円グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

データセット

データセットの設定には、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の数字（例：ds.0.hosts.0、ds.0.items.0）は特定のデータセットを表し、2 つ目の数字がある場合は設定されたホストまたはアイテムを表します。

パラメータ		type	name	value
データセットタイプ		0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。
	アイテム	4	ds.0.itemids.0	アイテムの ID。 ウィジェットをテンプレートダッシュボードで設定する場合、テンプレートで設定されたアイテムのみを指定してください。 注：複数のアイテムを設定するには、プロパティ名の番号を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテム（ウィジェット）が設定されていない場合は必須
	アイテム（ウィジェット）	1	ds.0.itemids.0._reference	アイテムの ID の代わりに： `ABCDE._itemid` - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注：複数のウィジェットを設定するには、プロパティ名の番号を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテムが設定されていない場合は必須
	色	1	ds.0.color.0	16 進数のカラーコード（例：FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合にサポート

パラメータ		type	name	value
	アイテムタイプ	0	ds.0.type.0	0 - *(デフォルト)* 通常; 1 - 合計。 「合計」は、チャート全体で 1 つのアイテムにのみ設定できます。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合にサポート
	ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例：”Zabbix*”）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	アイテムパターン	1	ds.0.items.0	アイテム名またはパターン（例：”*: Number of processed* values per second”）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを指定してください。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
	色	1	ds.0.color	16 進数のカラーコード（例：FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合にサポート
	カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 0-11 の値が指定可能。 デフォルト：0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合にサポート
集計関数		0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(デフォルト)* last。
データセット集計		0	ds.0.dataset_aggregation	0 - *(デフォルト)* なし; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum。 パラメータの動作: - アイテムタイプが「合計」に設定されている場合にサポート

パラメータ	type	name	value
データセットラベル	1	ds.0.data_set_label	任意の文字列値。 デフォルト："（空）。

Displaying options

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
履歴データの選択	0	source	0 - *(default)* 自動; 1 - 履歴; 2 - トレンド
描画	0	draw_type	0 - *(default)* 円グラフ; 1 - ドーナツグラフ
幅	0	width	20 - 半径の 20%。 30 - 半径の 30%。 40 - 半径の 40%。 50 - *(デフォルト)* 半径の 50% パラメータの動作: - *Draw* が"Doughnut" に設定されている場合に サポートされます
ストローク幅	0	stroke	設定可能な値の範囲は 0～10 です。 デフォルト: 0 パラメータの動作: - *Draw* が"Doughnut" に設定されている場合に サポートされます
合計値を表示	0	total_show	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - *Draw* が" ドーナツ" に設定されている場合に サポートされます
Size	0	value_size_type	0 - *(デフォルト)* 自動; 1 - カスタム パラメータの動作: - 合計値を表示が" 有効" に設定されている場合に サポートされます

パラメータ		type	name	value
	サイズ (カスタムサイズの値)	0	value_size	有効な値の範囲は 1～100 です。 デフォルト: 20
	小数点以下の桁数	0	decimal_places	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます。 有効な値の範囲は 0～6 です。 デフォルト: 2
	単位 (チェックボックス)	0	units_show	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます。 0 - *(デフォルト)* 無効; 1 - 有効
	単位 (値)	1	units	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます 任意の文字列値
	太字	0	value_bold	パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合に サポートされます 0 - *(デフォルト)* 無効; 1 - 有効。
	色	1	value_color	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます 16 進数のカラーコード (例: FF0000)。
セクター間のスペース		0	space	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます 指定可能な値の範囲は 0～10 です。 デフォルト: 1。

パラメータ	type	name	value
N% 未満のセクターをマージ (チェックボックス)	0	merge	0 - *(デフォルト)* 無効; 1 - 有効。
N% 未満のセクターをマージ (値)	0	merge_percent	指定可能な値の範囲は 1～10 です。 デフォルト: 1。 パラメータの動作: - *N*% 未満のセクターをマージ (チェックボックス) が" 有効" に設定されている場合に サポートされます。
N% 未満のセクターをマージ (色)	1	merge_color	16 進数のカラーコード (例: FF0000)。 パラメーターの動作: - *N*% 未満のセクターをマージ (チェックボックス) が" 有効" に設定されている場合に サポートされます

期間

期間の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
期間	1	time_period._referenceDASHBOARD._timeperiod - ダッシュボードの期間セレクタをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート

Legend

凡例の設定では、以下のパラメータがサポートされています。

パラメータ	type	name	value
凡例の表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効
値の表示	0	legend_value	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - 凡例の表示が"有効"に設定されている場合にサポートされます。
集計関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - 凡例の表示が"有効"に設定されている場合にサポートされます
行	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変 パラメータの動作: - 凡例の表示が"有効"に設定されている場合にサポートされます 有効な値の範囲は 1～10 です。
行数/ 最大行数	0	legend_lines	デフォルト: 1. パラメータの動作: - 凡例の表示が"有効"に設定されている場合に サポートされます。 有効な値の範囲は 1～4 です。
列数	0	legend_columns	デフォルト: 4. パラメータの動作: - 凡例の表示が"有効"に設定され、値の表示が"無効"に設定されている場合に サポートされます。

例

次の例は、円グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

円グラフウィジェットの設定

円グラフウィジェットは、以下の手順で設定します:

- 1 つのホストに、合計 9 つのアイテムのデータセットを 2 つ作成します。
- 最初のデータセットはタイプが" アイテムリスト " で、3 つのアイテムで構成されています。これらのアイテムはすべて " 通常 " タイプで、それぞれ異なる色で表示されます。
- 2 つめのデータセットはタイプが" アイテムパターン " で、6 つのアイテムで構成されています。これらのアイテムは集計が設定され、カスタム色で表示されます。
- 2 つめのデータセットには、カスタムデータセットラベルも設定します。
- 円グラフのデータは、幅がカスタムで、中央に単位が付いた合計値を持つドーナツグラフとして表示されます。
- 円グラフのデータは、過去 3 時間のカスタム期間について表示および集計されます。
- 円グラフの凡例には、設定されたアイテムが 4 行で表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
                        "y": 0,
                        "width": 24,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "FF0000"
                            },
                            {
                                "type": 0,
                                "name": "ds.0.type.1",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23269
                            },
                            {
                                "type": 1,
```

```
            "name": "ds.0.color.2",
            "value": "BF00FF"
        },
        {
          "type": 0,
          "name": "ds.0.type.2",
          "value": 0
        },
        {
            "type": 4,
            "name": "ds.0.itemids.3",
            "value": 23257
        },
        {
            "type": 1,
            "name": "ds.0.color.3",
            "value": "0040FF"
        },
        {
          "type": 0,
          "name": "ds.0.type.3",
          "value": 0
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "Zabbix server"
        },
        {
            "type": 1,
            "name": "ds.1.items.0",
            "value": "*: Number of processed *values per second"
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "000000"
        },
        {
            "type": 0,
            "name": "ds.1.aggregate_function",
            "value": 3
        },
        {
            "type": 1,
            "name": "ds.1.data_set_label",
            "value": "Number of processed values per second"
        },
        {
            "type": 0,
            "name": "draw_type",
            "value": 1
        },
        {
            "type": 0,
            "name": "width",
            "value": 30
        },
        {
            "type": 0,
            "name": "total_show",
            "value": 1
```

```
                            },
                            {
                                "type": 0,
                                "name": "units_show",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "graph_time",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

22 問題のあるホスト

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドで障害中のホストウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。障害中のホストウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

障害ホストウィジェットでは、以下のパラメーターがサポートされています。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
除外するホストグループ		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定であるか、指定したホストが属するホストグループが少なくとも 1 つ設定されている必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
	ホスト (ウィジェット/ダッシュボード)	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害		1	problem	障害イベント名（大文字小文字を区別しない、完全一致または部分一致）。
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）

パラメーター	type	name	value
抑止中の障害を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
障害のないグループを非表示	0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害の表示	0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離。
Reference	1	reference	5 文字からなる任意の文字列値 (例: ABCDE または JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、障害中のホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

障害中のホストウィジェットの設定

ホストグループ”2” と”4” に属する、名前に”CPU” という文字列が含まれ、深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” の障害が発生しているホストを表示する障害中のホストウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemhosts",
                        "name": "Problem hosts",
```

```
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "problem",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

23 問題

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値は、dashboard.create および dashboard.update メソッドで障害ウィジェットを設定することを可能にします。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。障害ウィジェットの作成または更新を確実に行うには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

障害ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
表示	0	show	1 - *(デフォルト)* 最近の障害; 2 - 履歴; 3 - 障害。
ホストグループ	2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定する場合は、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
除外するホストグループ		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外する場合は、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定する場合は、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクタをホストのデータソースとして設定します; ABCDE._hostids - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害		1	problem	障害イベント名（大文字小文字を区別しない、完全一致または部分一致）。
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 致命的障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定する場合は、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。

パラメーター		type	name	value
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
タグの表示		0	show_tags	0 - *(デフォルト)* なし; 1 - 1; 2 - 2; 3 - 3。
タグ名*(フォーマット)*		0	tag_name_format	0 - *(デフォルト)* フル; 1 - 省略; 2 - なし。 パラメーターの動作: - タグの表示が”1”、”2”、”3” の場合にサポート
タグ表示優先度		1	tag_priority	カンマ区切りのタグリスト。 パラメーターの動作: - タグの表示が”1”、”2”、”3” の場合にサポート
運用データの表示		0	show_opdata	0 - *(デフォルト)* なし; 1 - 別表示; 2 - 障害名と一緒に表示。

パラメーター	type	name	value
抑制された障害の表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
障害対応状況	0	acknowledgement_status	0 - *(デフォルト)* すべて; 1 - 未対応; 2 - 対応済み。
自分が対応エントリ	0	acknowledged_by_me	0 - *(デフォルト)* 無効; 1 - 有効。
の並び順	0	sort_triggers	1 - 深刻度 (降順); 2 - ホスト (昇順); 3 - 時刻 (昇順); 4 - *(デフォルト)* 時刻 (降順); 13 - 深刻度 (昇順); 14 - ホスト (降順); 15 - 障害 (昇順); 16 - 障害 (降順)。 ”時刻 (降順)” および”時刻 (昇順)” 以外のすべての値については、タイムラインの表示パラメーターを”無効” に設定する必要があります。 ”ホスト (昇順)” および”ホスト (降順)” の値は、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
タイムラインの表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効。 パラメーターの動作: - エントリの並び順が”時刻 (降順)” または”時刻 (昇順)” の場合にサポート
行全体のハイライト表示	0	highlight_row	0 - *(デフォルト)* 無効; 1 - 有効。 このパラメーターはハイコントラストテーマではサポートされません。
行数	0	show_lines	1-100 の範囲で指定可能。 デフォルト: 25。

パラメーター	type	name	value
参照名	1	reference	5 文字からなる任意の文字列値（例: ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、障害ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

障害ウィジェットの設定

ホストグループ”4” で、以下の条件を満たす障害を表示する障害ウィジェットを設定します:

- ”scope” という名前のタグが付いており、そのタグの値が”performance”、”availability” または”capacity” である障害。
- 深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” である障害。

さらに、タグと運用データを表示するようにウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "name": "Problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
```

```
            "value": "performance"
        },
        {
            "type": 1,
            "name": "tags.1.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.1.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.1.value",
            "value": "availability"
        },
        {
            "type": 1,
            "name": "tags.2.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.2.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.2.value",
            "value": "capacity"
        },
        {
            "type": 0,
            "name": "severities.0",
            "value": 2
        },
        {
            "type": 0,
            "name": "severities.1",
            "value": 3
        },
        {
            "type": 0,
            "name": "severities.2",
            "value": 4
        },
        {
            "type": 0,
            "name": "severities.3",
            "value": 5
        },
        {
            "type": 0,
            "name": "show_tags",
            "value": 1
        },
        {
            "type": 0,
            "name": "show_opdata",
            "value": 1
        }
```

```
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

24 問題の深刻度別

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで深刻度ごとの障害数ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。深刻度ごとの障害数ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

重大度別障害ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - 参照パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
除外するホストグループ		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - 参照パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害		1	problem	障害のイベント名（大文字小文字を区別しない、完全一致または部分一致）。

パラメーター		type	name	value
重大度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
表示		0	show_type	0 - *(デフォルト)* ホストグループ; 1 - 合計。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされず、デフォルトで「合計」に設定されます。
レイアウト		0	layout	0 - *(デフォルト)* 横並び; 1 - 縦並び。 パラメーターの動作: - 表示が「合計」の場合のみサポート

パラメーター	type	name	value
運用データの表示	0	show_opdata	0 - *(デフォルト)* なし; 1 - 個別に; 2 - 障害名と一緒に。
抑制された障害の表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
障害のないグループを非表示	0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効。 パラメーターの動作: - 表示が「ホストグループ」の場合のみサポート このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害の表示	0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離。
タイムラインの表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効。
参照	1	reference	5 文字からなる任意の文字列値（例: ABCDE または JBPNL）。この値はウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、深刻度ごとの障害数ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

深刻度ごとの障害数ウィジェットの設定

すべてのホストグループの障害の合計数を表示する深刻度ごとの障害数ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemsbysv",
                        "name": "Problems by severity",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_type",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値により、`dashboard.create` および `dashboard.update` メソッドで散布図ウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。散布図ウィジェットの作成または更新を確実に行うには、以下の表に示すパラメータの動作を参照してください。

パラメーター

散布図ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	値
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
参照	1	reference	5 文字からなる任意の文字列値（例：`ABCDE` または `JBPNL`）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

データセット

データセットの設定には、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の数字（例: ds.0.hosts.0, ds.0.items.0）は特定のデータセットを表し、2 つ目の数字がある場合は設定されたホストまたはアイテムを表します。

パラメータ	type	name	value
データセットタイプ	0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。

パラメータ	type	name	value
データセット: アイテムリスト			
色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合は必須
X 軸	4	ds.0.x_axis_itemids.0	アイテムID。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみを設定する必要があります。 注: 複数のアイテムを設定するには、プロパティ名の数字を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*X* 軸（ウィジェット）が設定されていない場合は必須
X 軸（ウィジェット）	1	ds.0.x_axis_itemids.0.reference	アイテムIDの代わりに: `ABCDE._itemid` - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注: 複数のウィジェットを設定するには、プロパティ名の数字を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*X* 軸が設定されていない場合は必須
Y 軸	4	ds.0.y_axis_itemids.0	アイテムID。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみを設定する必要があります。 注: 複数のアイテムを設定するには、プロパティ名の数字を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*Y* 軸（ウィジェット）が設定されていない場合は必須

パラメータ	type	name	value
Y 軸（ウィジェット）	1	ds.0.y_axis_itemids.0	アイテムIDの代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注: 複数のウィジェットを設定するには、プロパティ名の数字を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*Y* 軸が設定されていない場合は必須
ホストの上書きデータセット:	1	ds.0.override_hostid.0	ABCDE._hostid - 互換性のあるウィジェット（" 参照" パラメータを"ABCDE"に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
アイテムパターン			
色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合はサポート
カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 有効な値の範囲: 0-11。 デフォルト: 0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合はサポート
ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例: "Zabbix*"）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ	type	name	value
X 軸アイテムパターン	1	ds.0.x_axis_items.0	アイテム名またはパターン（例: "*: Number of processed *values per second"）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを設定する必要があります。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
Y 軸アイテムパターン	1	ds.0.y_axis_items.0	アイテム名またはパターン（例: "*: Number of processed *values per second"）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを設定する必要があります。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
ホストグループ	2	groupids.0	ホストグループID。 注: 複数のホストグループを設定するには、プロパティ名の数字を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストグループ (ウィジェット)	1	groupids._reference	ホストグループID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（参照パラメータを"ABCDE" に設定）をホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストの上書きホストタグ	1	ds.0.override_hostid._reference	`ABCDE._hostid` - 互換性のあるウィジェット（" 参照" パラメータを"ABCDE" に設定）をホストのデータソースとして設定; `DASHBOARD._hostid` - ダッシュボードのホストセレクタをホストのデータソースとして設定。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		type	name	value
	評価タイプ	0	evaltype_host	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
マーカー		0	ds.0.marker	0 - *(デフォルト)* 楕円; 1 - 四角; 2 - 三角; 3 - ひし形; 4 - 星; 5 - 十字。
マーカーサイズ		0	ds.0.marker_size	0 - *(デフォルト)* 小（6px）; 1 - 中（9px）; 2 - 大（15px）。

パラメータ	type	name	value
タイムシフト	1	ds.0.timeshift	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスを使用できます。負の値も許可されます。 デフォルト: ""（空）。
集約間隔	1	ds.0.aggregate_interval	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスを使用できます。 デフォルト: 15m。
集約関数	0	ds.0.aggregate_function	0 - 使用しない; 1 - min; 2 - max; 3 - *(デフォルト)* avg; 4 - count; 5 - sum; 6 - first; 7 - last。

表示オプション

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

期間

期間の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
期間	1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクタをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが"ABCDE"に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート

パラメータ	type	name	値
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート

軸

軸の設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
X 軸	0	x_axis	0 - 無効; 1 - *(デフォルト)* 有効。
Y 軸	0	y_axis	0 - *(デフォルト)* 無効; 1 - 有効。
最小	1	x_axis_min y_axis_min	任意の数値。 デフォルト: "" (空)。
最大	1	x_axis_max y_axis_max	任意の数値。 デフォルト: "" (空)。
単位 (タイプ)	0	x_axis_units y_axis_units	0 - *(デフォルト)* 自動; 1 - 静的。
単位 (値)	1	x_axis_static_units y_axis_static_units	任意の文字列。 デフォルト: "" (空)。

凡例

凡例の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
凡例を表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効。
集約関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
行数	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
行数/ 最大行数	0	legend_lines	可能な値の範囲: 1-10。 デフォルト: 1。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート

パラメータ	type	name	値
列数	0	legend_columns	可能な値の範囲: 1-4。 デフォルト: 4。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート

閾値

閾値の設定には、以下のパラメータがサポートされています。

Note:
閾値プロパティ名の数字 (例: thresholds.0.color) は、昇順にソートされたリスト内の閾値の位置を参照します。ただし、閾値が異なる順序で設定されている場合、Zabbix フロントエンドでウィジェット設定を更新した後に値が昇順にソートされます (例: `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`)。

パラメータ	型	名前	値
色の補間	0	interpolation	0 - *(デフォルト)* 無効; 1 - 有効。
色	1	thresholds.0.color	16 進数のカラーコード（例: FF0000）。
X 軸	1	thresholds.0.x_axis_threshold	任意の数値。サフィックス（例: "1d", "2w", "4K", "8G"）がサポートされています。
Y 軸	1	thresholds.0.y_axis_threshold	任意の数値。サフィックス（例: "1d", "2w", "4K", "8G"）がサポートされています。

例

次の例は、散布図ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定のみを説明することを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

散布図ウィジェットの設定

散布図ウィジェットは、以下のように設定します。

- データセットタイプ「アイテムパターン」のデータセットを 5 つ、それぞれデフォルトの集計 (平均、15 分) と異なるマーカーで設定
- 凡例は、設定したアイテムを 1 行 2 列で表示
- しきい値を設定し、いずれかの軸で値が 80 に達した場合にマーカーを赤色で表示

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "scatterplot",
                        "name": "Scatter plot",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
```

```
            "type": 1,
            "name": "ds.0.color",
            "value": "0C5A87"
        },
        {
            "type": 1,
            "name": "ds.0.hosts.0",
            "value": "de-frankfurt*"
        },
        {
            "type": 1,
            "name": "ds.0.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.0.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.0.marker",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.0.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "66B0D9"
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "fr-paris*"
        },
        {
            "type": 1,
            "name": "ds.1.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.1.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.1.marker",
            "value": 1
        },
        {
            "type": 0,
            "name": "ds.1.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.2.color",
```

```
                "value": "0A466A"
            },
            {
                "type": 1,
                "name": "ds.2.hosts.0",
                "value": "lv-riga*"
            },
            {
                "type": 1,
                "name": "ds.2.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.2.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.2.marker",
                "value": 2
            },
            {
                "type": 0,
                "name": "ds.2.marker_size",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.3.color",
                "value": "3394C3"
            },
            {
                "type": 1,
                "name": "ds.3.hosts.0",
                "value": "pl-warsaw*"
            },
            {
                "type": 1,
                "name": "ds.3.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.3.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.3.marker",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.3.marker_size",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.4.color",
                "value": "1492C8"
            },
```

```
                {
                    "type": 1,
                    "name": "ds.4.hosts.0",
                    "value": "se-stockholm*"
                },
                {
                    "type": 1,
                    "name": "ds.4.x_axis_items.0",
                    "value": "Memory utilization"
                },
                {
                    "type": 1,
                    "name": "ds.4.y_axis_items.0",
                    "value": "CPU utilization"
                },
                {
                    "type": 0,
                    "name": "ds.4.marker",
                    "value": 4
                },
                {
                    "type": 0,
                    "name": "ds.4.marker_size",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "legend_columns",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "interpolation",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "thresholds.0.color",
                    "value": "D40000"
                },
                {
                    "type": 1,
                    "name": "thresholds.0.x_axis_threshold",
                    "value": "80"
                },
                {
                    "type": 1,
                    "name": "thresholds.0.y_axis_threshold",
                    "value": ""
                },
                {
                    "type": 1,
                    "name": "thresholds.1.color",
                    "value": "D40000"
                },
                {
                    "type": 1,
                    "name": "thresholds.1.x_axis_threshold",
                    "value": ""
                },
                {
                    "type": 1,
```

```
                                "name": "thresholds.1.y_axis_threshold",
                                "value": "80"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "JQISY"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- ダッシュボードウィジェットフィールド
- dashboard.create
- dashboard.update

26 SLA レポート

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで*SLA* レポートウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*SLA* レポートウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

SLA レポートウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - *(デフォルト)* 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - 15 分
SLA	10	slaid.0	SLA ID。 パラメーターの動作: - 必須
サービス	9	serviceid.0	サービスID。
表示期間	0	show_periods	有効な値の範囲は 1～100 です。 デフォルト: 20
From	1	date_from	YYYY-MM-DD 形式の有効な日付文字列。 修飾子 d、w、M、y(例: now、now/d、now/w-1w など) を使用した相対日付がサポートされています。
To	1	date_to	YYYY-MM-DD 形式の有効な日付文字列。 修飾子 d、w、M、y(例: now、now/d、now/w-1w など) を使用した相対日付がサポートされています。

例

次の例は、*SLA* レポートウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

SLA レポートウィジェットの設定

SLA "4"、サービス"2" の過去 30 日間の SLA レポートを表示する *SLA* レポートウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "slareport",
                        "name": "SLA report",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 10,
                                "name": "slaid.0",
                                "value": 4
                            },
                            {
                                "type": 9,
                                "name": "serviceid.0",
                                "value": 2
                            },
                            {
```

```
                                "type": 1,
                                "name": "date_from",
                                "value": "now-30d"
                            },
                            {
                                "type": 1,
                                "name": "date_to",
                                "value": "now"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

27 システム情報

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでシステム情報ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。システム情報ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

システム情報ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分
表示	0	info_type	0 - *(デフォルト)* システムステータス 1 - HA ノード
ソフトウェア更新チェックの詳細を表示	0	show_software_update_check_details	0 - *(デフォルト)* 無効 1 - 有効 パラメーターの動作: - Zabbix サーバー設定のAllowSoftwareUpdateCheckが有効で、表示が” システムステータス” に設定されている場合、サポート

例

次の例は、システム情報ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

システム情報ウィジェットの設定

10 分間隔で更新し、ソフトウェアアップデートのチェックを有効にした状態でシステム統計情報を表示するシステム情報ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "systeminfo",
                        "name": "System information",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            },
                            {
                                "type": 0,
                                "name": "show_software_update_check_details",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

28 トップホスト

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドで上位ホストウィジェットを設定できます。

> **Attention:**
> ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。上位ホストウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

Top hosts ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメータを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメータは未設定、または指定したホストが属するホストグループが少なくとも 1 つ設定されている必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクタをホストのデータソースとして設定します; `ABCDE._hostids` - 互換性のあるウィジェット（*Reference* パラメータを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 ホストタグを設定する場合 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 ホストタグを設定する場合 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 必須 ホストタグを設定する場合 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
メンテナンス中のホストを表示カラム(下記参照)		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	type	name	value
並び順	0	column	設定されたカラムの数値。
順序	0	order	2 - (デフォルト) Top N; 3 - Bottom N。
ホスト数上限	0	show_lines	1-1000 の範囲で指定可能。 デフォルト: 10。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

列

列には、データパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメーターはすべての列でサポートされています。

パラメーター	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメーターの動作: - 必須
Data	0	columns.0.data	1 - 項目値 2 - ホスト名 3 - テキスト パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数カラーコード（例：FF0000） パラメータの動作: - 必須

アイテム値

Data が「アイテム値」に設定されている場合、以下のパラメータがサポートされます。

Note:
Thresholds プロパティ名の最初の数字（例: columnsthresholds.0.color.0）は、しきい値が設定されている列を参照し、2 番目の数字は昇順で並べ替えられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix フロントエンドでウィジェット構成を更新した後、値は昇順で並べ替えられます（例: `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`）。

パラメータ	type	name	value
アイテム	1	columns.0.item	有効なアイテム名。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみを設定する必要があります。
アイテム値の表示形式	0	columns.0.display_value_as	0 - *(デフォルト)* 数値; 1 - テキスト; 2 - バイナリ。
表示	0	columns.0.display	1 - *(デフォルト)* そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。 パラメータの動作: - アイテム値の表示形式が「数値」に設定されている場合にサポート
最小値	1	columns.0.min	任意の数値。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
最大値	1	columns.0.max	任意の数値。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
小数点以下の桁数	0	columns.0.decimal_places	0-10の範囲で指定可能。 デフォルト: 2。 パラメータの動作: - アイテム値の表示形式が「数値」に設定されている場合にサポート
スパークライン 幅	0	columns.0.sparkline.width	1-10の範囲で指定可能。 デフォルト: 1。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート

パラメータ		type	name	value
	塗りつぶし	0	columns.0.sparkline.fill	0-10 の範囲で指定可能。 デフォルト: 3。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	色	1	columns.0.sparkline.color	16進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	期間	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE のもの）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	From	1	columns.0.sparkline.time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	To	1	columns.0.sparkline.time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	履歴データ	0	columns.0.sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - アイテム値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
しきい値				
	色	1	columnsthresholds.0.color	16進数のカラーコード（例: FF0000）。 パラメータの動作: - アイテム値の表示形式が「数値」に設定されている場合にサポート
	しきい値	1	columnsthresholds.0.threshold	任意の文字列。 パラメータの動作: - アイテム値の表示形式が「数値」に設定されている場合にサポート

パラメータ		type	name	value
ハイライト				
	色	1	columns.0.highlights.0.color	16進数のカラーコード（例: FF0000）。 パラメータの動作: - アイテム値の表示形式が「テキスト」に設定されている場合にサポート
	パターン	1	columns.0.highlights.0.pattern	任意の文字列。 パラメータの動作: - アイテム値の表示形式が「テキスト」に設定されている場合にサポート
サムネイルの表示		0	columns.0.show_thumbnail	0 - (デフォルト) 無効; 1 - 有効。 パラメータの動作: - アイテム値の表示形式が「バイナリ」に設定されている場合にサポート
集計関数		0	columns.0.aggregate_function	0 - (デフォルト) 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last。
期間		1	columns.0.time_period.reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE のもの）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。 パラメータの動作: - 集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
	From	1	columns.0.time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
	To	1	columns.0.time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート

パラメータ	type	name	value
履歴データ	0	columns.0.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - アイテム値の表示形式が「数値」に設定されている場合にサポート
参照	1	reference	5 文字で構成される任意の文字列（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

Text

データが” テキスト ” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
テキスト	1	columns.0.text	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{INVENTORY.*} パラメーターの動作: - データがテキスト ” に設定されている場合、必須

例

次の例は、上位ホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

上位ホストウィジェットの設定

ホストグループ”4” で CPU 使用率上位のホストを表示する上位ホストウィジェットを構成します。さらに、次のカスタム列を構成します: ”Host name”, ”CPU utilization in %”, ”1m avg”, ”5m avg”, ”15m avg”, ”Processes”。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "tophosts",
                        "name": "Top hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
```

```
            "value": 4
        },
        {
            "type": 1,
            "name": "columns.0.name",
            "value": "Host"
        },
        {
            "type": 0,
            "name": "columns.0.data",
            "value": 2
        },
        {
            "type": 1,
            "name": "columns.0.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.1.name",
            "value": "CPU utilization in %"
        },
        {
            "type": 0,
            "name": "columns.1.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.1.base_color",
            "value": "4CAF50"
        },
        {
            "type": 1,
            "name": "columns.1.item",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "columns.1.display",
            "value": 3
        },
        {
            "type": 1,
            "name": "columns.1.min",
            "value": "0"
        },
        {
            "type": 1,
            "name": "columns.1.max",
            "value": "100"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.0",
            "value": "FFFF00"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.0",
            "value": "50"
        },
```

```
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.1",
                        "value": "FF8000"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.threshold.1",
                        "value": "80"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.2",
                        "value": "FF4000"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.threshold.2",
                        "value": "90"
                    },
                    {
                        "type": 1,
                        "name": "columns.2.name",
                        "value": "1m avg"
                    },
                    {
                        "type": 0,
                        "name": "columns.2.data",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "columns.2.base_color",
                        "value": "FFFFFF"
                    },
                    {
                        "type": 1,
                        "name": "columns.2.item",
                        "value": "Load average (1m avg)"
                    },
                    {
                        "type": 1,
                        "name": "columns.3.name",
                        "value": "5m avg"
                    },
                    {
                        "type": 0,
                        "name": "columns.3.data",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "columns.3.base_color",
                        "value": "FFFFFF"
                    },
                    {
                        "type": 1,
                        "name": "columns.3.item",
                        "value": "Load average (5m avg)"
                    },
                    {
                        "type": 1,
```

```
                            "name": "columns.4.name",
                            "value": "15m avg"
                        },
                        {
                            "type": 0,
                            "name": "columns.4.data",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "columns.4.base_color",
                            "value": "FFFFFF"
                        },
                        {
                            "type": 1,
                            "name": "columns.4.item",
                            "value": "Load average (15m avg)"
                        },
                        {
                            "type": 1,
                            "name": "columns.5.name",
                            "value": "Processes"
                        },
                        {
                            "type": 0,
                            "name": "columns.5.data",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "columns.5.base_color",
                            "value": "FFFFFF"
                        },
                        {
                            "type": 1,
                            "name": "columns.5.item",
                            "value": "Number of processes"
                        },
                        {
                            "type": 0,
                            "name": "columns.5.decimal_places",
                            "value": 0
                        },
                        {
                            "type": 0,
                            "name": "column",
                            "value": 1
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
```

```
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

29 トップアイテム

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで上位アイテムウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。上位アイテムウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

Top items ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
ホスト	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに: ABCDE._hostgroupids - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメータは未設定であるか、指定したホストが属するホストグループが少なくとも 1 つ設定されている必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクタをホストのデータソースとして設定します; ABCDE._hostids - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
レイアウト		0	layout	0 - *(デフォルト)* 横方向; 1 - 縦方向。
障害を表示アイテム(下記参照)		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されていない; 2 - なし。

Advanced configuration

上位アイテムウィジェットでは、以下の詳細設定パラメータがサポートされています。

パラメータ	type	name	value
ホストの順序並べ替え	0	host_ordering_order_by	0 - *(default)* ホスト名; 3 - アイテムの値

パラメータ		type	name	value
	アイテムパターン	1	host_ordering_item_0	アイテム の名前またはパターン（例：”*: 1 秒あたりに処理される * 値の数”） 注: 複数のアイテムパターンを設定するには、プロパティ名に増分番号を含む各アイテムパターンのダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボード でウィジェットを設定する場合は、テンプレートで構成されているアイテムのパターンのみを設定する必要があります。 パラメータの動作: - *Order by* が”Item value” に設定されている場合、必須
Order		0	host_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件
制限		0	host_ordering_limit	有効な値の範囲は 1～100 です。 デフォルト: 10
アイテムの順序				
順序付け		0	item_ordering_order_by	1 - ホスト 2 - アイテム名 3 - *(デフォルト)* アイテムの値
	ホストパターン	1	item_ordering_host_0	ホスト の名前またはパターン 注: 複数のホストパターンを設定するには、プロパティ名に増分番号を付けた、ホストパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメータの動作: - 順序付けが”Host” に設定されている場合は 必須
順序		0	item_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件
制限		0	item_ordering_limit	指定可能な値の範囲は 1 ～100 です。 デフォルト: 10 件

カラム

Note:
プロパティ名の最初の数字 (例: columns.0.items.0, columns.0.item_tags_evaltype) は特定のカラムを表し、2 つ目の数字がある場合は設定されたエンティティ (例: アイテムパターン、タグ) を表します。

パラメーター		type	name	value
アイテムパターン		1	columns.0.items.0	アイテム名またはパターン (例: ”*: Number of processed *values per second”)。 注: 複数のアイテムパターンを設定するには、プロパティ名の 2 つ目の数字を増やして、各アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートで設定されたアイテムのパターンのみを設定してください。 パラメーターの動作: - 必須
アイテムタグ				
	評価タイプ	0	columns.0.item_tags.0.evaltype	0 (デフォルト) And/Or; 2 - Or.
	タグ名	1	columns.0.item_tags.0.tag	任意の文字列値。 注: プロパティ名の 2 つ目の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - アイテムタグを設定する場合は必須
	演算子	0	columns.0.item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の 2 つ目の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - アイテムタグを設定する場合は必須
	タグ値	1	columns.0.item_tags.0.value	任意の文字列値。 注: プロパティ名の 2 つ目の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - アイテムタグを設定する場合は必須
ベースカラー		1	columns.0.base_color	16 進数のカラーコード (例: FF0000)。

パラメーター		type	name	value
値の表示形式		0	columns.0.display_value_as	1 - *(デフォルト)* 数値; 2 - テキスト。
表示		0	columns.0.display	1 - *(デフォルト)* そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。
最小値		1	columns.0.min	任意の数値。サフィックス (例: ”1d”, ”2w”, ”4K”, ”8G”) がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」で表示が「バー」または「インジケーター」の場合にサポート
最大値		1	columns.0.max	任意の数値。サフィックス (例: ”1d”, ”2w”, ”4K”, ”8G”) がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」で表示が「バー」または「インジケーター」の場合にサポート
スパークライン				
	幅	0	columns.0.sparkline.width	1-10の範囲の値が可能。 デフォルト: 1。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
	塗りつぶし	0	columns.0.sparkline.fill	1-10 の範囲の値が可能。 デフォルト: 3。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
	カラー	1	columns.0.sparkline.color	16進数のカラーコード (例: FF0000)。 デフォルト: 42A5F5。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート

パラメーター		type	name	value
	期間	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット (その reference パラメーターが ABCDE と等しいもの) をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターのみで期間を設定することもできます。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
	From	1	columns.0.sparkline.time_period.from	絶対 (YYYY-MM-DD hh:mm:ss) または相対時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
	To	1	columns.0.sparkline.time_period.to	絶対 (YYYY-MM-DD hh:mm:ss) または相対時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
	履歴データ	0	columns.0.sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメーターの動作: - 表示が「スパークライン」の場合にサポート
しきい値	カラー	1	columns.0.thresholds.color	16 進数のカラーコード (例: FF0000)。 パラメーターの動作: - 値の表示形式が「数値」の場合にサポート
	しきい値	1	columns.0.thresholds.threshold	任意の数値と サフィックス (例: "1d", "2w", "4K", "8G") がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」の場合にサポート
ハイライト	ハイライト	1	columns.0.highlights.color	16 進数のカラーコード (例: FF0000)。 パラメーターの動作: - 値の表示形式が「テキスト」の場合にサポート
	しきい値	1	columns.0.highlights.pattern	任意の正規表現。 パラメーターの動作: - 値の表示形式が「テキスト」の場合にサポート

パラメーター	type	name	value
小数点以下の桁数高度な設定	0	columns.0.decimal_places	0-10の範囲の値が可能。 デフォルト: 2。
集計関数	0	columns.0.aggregate_function	値の表示形式が「数値」の場合: 0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last。 値の表示形式が「テキスト」の場合: 0 - *(デフォルト)* 使用しない; 4 - count; 6 - first; 7 - last。
期間	1	columns.0.time_period.reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット (その reference パラメーターが ABCDE と等しいもの) をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターのみで期間を設定することもできます。 パラメーターの動作: - 集計関数が「min」「max」「avg」「count」「sum」「first」「last」の場合にサポート
From	1	columns.0.time_period.from	絶対 (YYYY-MM-DD hh:mm:ss) または相対時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメーターの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」の場合にサポート
To	1	columns.0.time_period.to	絶対 (YYYY-MM-DD hh:mm:ss) または相対時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメーターの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」の場合にサポート

パラメーター	type	name	value
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。 パラメーターの動作: - 値の表示形式が「数値」の場合にサポート

例

次の例は、*Top items* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

トップアイテムウィジェットの設定

ホスト”10084” のデータと、名前が”component” のタグに値が”cpu” を含むアイテムのみを表示する トップアイテムウィジェットを設定します。さらに、ホストを上位に表示し、セルを表すために色付きのゲージバーを使用します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
                        "name": "Top items",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
```

```
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

30 のトップトリガー

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで上位トリガーウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。上位トリガーウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

Top triggers ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	値
リフレッシュ間隔	0	rf_rate	0 - (デフォルト) リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト	3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは、まったく設定しないか、設定したホストが属する少なくとも 1 つのホストグループで設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害	1	problem	障害イベント名（大文字小文字を区別しない、完全名またはその一部）。
深刻度	0	severities.0	トリガーの深刻度。 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ			

パラメーター		type	name	値
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 障害タグを設定する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 障害タグを設定する場合は必須
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 障害タグを設定する場合は必須
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換ウィジェット（参照パラメーターを"ABCDE"に設定）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメーターで期間のみを設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - 期間が設定されていない場合にサポート
トリガーの上限		0	show_lines	有効な値の範囲は 1-100。 デフォルト: 10。

例

次の例は、上位トリガーウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

上位トリガーウィジェットの設定

ホストグループ”4” の各トリガーのすべての障害を数えて上位 5 トリガーを表示する上位トリガー ウィジェットを設定します。このウィジェットには、深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” のトリガーで、名前が”scope” でタグ名に”performance”、”availability”、または”capacity” のいずれかの値が含まれる障害のみが表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
```

```
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "performance"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "availability"
                            },
                            {
                                "type": 1,
                                "name": "tags.2.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.2.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.2.value",
                                "value": "capacity"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

31 トリガーの概要

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでトリガーの概要ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。トリガーの概要ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

トリガー概要ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	type	name	value
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
表示	0	show	1 - *(デフォルト)* 最近の障害; 2 - すべて; 3 - 障害。
ホストグループ	2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
ホスト	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - 参照パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを設定する場合、ホストグループパラメータは未設定であるか、設定したホストが属する少なくとも 1 つのホストグループで設定する必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクタをホストのデータソースとして設定します; ABCDE._hostids - 参照パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害タグ	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須

パラメータ	type	name	value
抑制された障害を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
レイアウト	0	layout	0 - *(デフォルト)* 水平; 1 - 垂直。

例

次の例は、トリガーの概要ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

トリガーの概要ウィジェットの設定

”scope” という名前で値に”availability” を含むタグでトリガーされ、すべてのホストグループのトリガー状態を表示するトリガー概要ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
```

```
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

32 URL

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドで*URL*ウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*URL* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

URL ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	タイプ	名前	値
リフレッシュ間隔	0	rf_rate	0 - *(デフォルト)* リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
URL	1	url	有効な URL 文字列。 パラメータの動作: - 必須
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメータを"ABCDE" に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

例

次の例は、*URL* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

URL ウィジェットの設定

Zabbix マニュアルのホームページを表示する *URL* ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "url",
                        "name": "URL",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "url",
                                "value": "https://www.zabbix.com/documentation/8.0/en"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
```

```
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

33 Web 監視

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで*Web* 監視ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*Web* 監視ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

Web 監視ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。

パラメーター		type	name	value
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストグループの除外		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクターをホストのデータソースとして設定します; ABCDE._hostids - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
シナリオタグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。

パラメーター		type	name	value
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
メンテナンス中のホストを表示		0	maintenance	0 - 無効; 1 - *(デフォルト)* 有効。
Reference		1	reference	5 文字からなる任意の文字列値（例: ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、*Web* 監視ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

Web 監視ウィジェットの設定

ホストグループ”4” のアクティブな Web 監視シナリオのステータス概要を表示する *Web* 監視ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

Discovered host

このクラスは、検出されたホストを操作するためのものです。

オブジェクトリファレンス:

- Discovered host

利用可能なメソッド:

- dhost.get - 検出されたホストを取得する

Discovered host オブジェクト

次のオブジェクトは、dhostAPI に直接関連しています。

検出されたホスト

> **Note:**
> 検出されたホストは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたホストオブジェクトは、ネットワーク検出ルールによって検出されたホストに関する情報を含みます。以下のプロパティがあります。

プロパティ	タイプ	説明
dhostid	ID	検出されたホストの ID。
druleid	ID	ホストを検出した検出ルールの ID。
lastdown	timestamp	検出されたホストが最後にダウンした時刻。
lastup	timestamp	検出されたホストが最後にアップした時刻。
status	integer	検出されたホストがアップかダウンか。少なくとも 1 つのアクティブな検出サービスがある場合、ホストはアップとみなされます。 可能な値： 0 - ホストはアップ; 1 - ホストはダウン;

dhost.get

説明

`integer/array dhost.get(object parameters)`

このメソッドは、指定したパラメーターに応じた検出されたホストを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
dhostids	ID/array	指定した ID の検出済みホストのみを返します。
druleids	ID/array	指定したディスカバリールールによって作成された検出済みホストのみを返します。
dserviceids	ID/array	指定したサービスを実行している検出済みホストのみを返します。
selectDRules	query	ホストを検出したディスカバリールールの配列を持つ`drules`プロパティを返します。

パラメーター	型	説明
selectDServices	query	ホスト上で実行されている検出済みサービスを持つdservicesプロパティを返します。
limitSelects	integer	count をサポートします。 サブセレクトによって返されるレコード数を制限します。
sortfield	string/array	以下のサブセレクトに適用されます: selectDServices - 結果は dserviceid でソートされます。 指定したプロパティで結果をソートします。
countOutput	boolean	可能な値: dhostid, druleid。 これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールによって検出されたホストの取得

ディスカバリルール"4" によって検出されたすべてのホストと実行中の発見されたサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dservices": [
                {
                    "dserviceid": "1",
                    "dhostid": "1",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
```

```
                "lastup": "1337697227",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.1",
                "dns": "station.company.lan"
            }
        ],
        "dhostid": "1",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697227",
        "lastdown": "0"
    },
    {
        "dservices": [
            {
                "dserviceid": "2",
                "dhostid": "2",
                "type": "4",
                "key_": "",
                "value": "",
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.4",
                "dns": "john.company.lan"
            }
        ],
        "dhostid": "2",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    },
    {
        "dservices": [
            {
                "dserviceid": "3",
                "dhostid": "3",
                "type": "4",
                "key_": "",
                "value": "",
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.26",
                "dns": "printer.company.lan"
            }
        ],
        "dhostid": "3",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    },
    {
        "dservices": [
            {
```

```
                    "dserviceid": "4",
                    "dhostid": "4",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.7",
                    "dns": "mail.company.lan"
                }
            ],
            "dhostid": "4",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        }
    ],
    "id": 1
}
```


参照

- Discovered service
- Discovery rule

ソース

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

Discovered service

このクラスは、検出されたサービスを操作するためのものです。

オブジェクトリファレンス:

- Discovered service

利用可能なメソッド:

- dservice.get - 検出されたサービスを取得する

Discovered service オブジェクト

次のオブジェクトは、dserviceAPI に直接関連しています。

検出されたサービス

Note:
検出されたサービスは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたサービスオブジェクトは、ホスト上のネットワークディスカバリールールによって検出されたサービスに関する情報を含みます。以下のプロパティを持ちます。

プロパティ	タイプ	説明
dserviceid	ID	検出されたサービスの ID。
dcheckid	ID	サービスの検出に使用されたディスカバリーチェックの ID。
dhostid	ID	サービスを実行している検出されたホストの ID。
dns	string	サービスを実行しているホストの DNS。
ip	string	サービスを実行しているホストの IP アドレス。
lastdown	timestamp	検出されたサービスが最後にダウンした時刻。
lastup	timestamp	検出されたサービスが最後にアップした時刻。

プロパティ	タイプ	説明
port	integer	サービスのポート番号。
status	integer	サービスのステータス。 可能な値： 0 - サービスは稼働中; 1 - サービスはダウン。
value	string	Zabbix エージェント、SNMPv1、SNMPv2 または SNMPv3 ディスカバリーチェックを実行した際にサービスから返された値。

dservice.get

説明

`integer/array dservice.get(object parameters)`

このメソッドは、指定したパラメーターに応じた検出されたサービスを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
dserviceids	ID/array	指定した ID の検出サービスのみを返します。
dhostids	ID/array	指定した検出ホストに属する検出サービスのみを返します。
dcheckids	ID/array	指定した検出チェックによって検出された検出サービスのみを返します。
druleids	ID/array	指定した検出ルールによって検出された検出サービスのみを返します。
selectDRules	query	サービスを検出した検出ルールの配列を持つ`drules`プロパティを返します。
selectDHosts	query	サービスが属する検出ホストの配列を持つ`dhosts`プロパティを返します。
selectHosts	query	サービスと同じ IP アドレスおよびプロキシを持つホストの`hosts`プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `hostid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `dserviceid`, `dhostid`, `ip`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストで検出されたサービスの取得

検出されたホスト"11" で検出されたすべてサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dserviceid": "12",
            "dhostid": "11",
            "value": "",
            "port": "80",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650607",
            "dcheckid": "5",
            "ip": "192.168.1.134",
            "dns": "john.local"
        },
        {
            "dserviceid": "13",
            "dhostid": "11",
            "value": "",
            "port": "21",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650610",
            "dcheckid": "6",
            "ip": "192.168.1.134",
            "dns": "john.local"
        }
    ],
    "id": 1
}
```

参照

- Discovered host
- Discovery check
- Host

ソース

CDService::get() in *ui/include/classes/api/services/CDService.php*.

Discovery check

このクラスは、ディスカバリチェックを操作するためのものです。

オブジェクトリファレンス:

- Discovery check

利用可能なメソッド:

- dcheck.get - ディスカバリチェックを取得する

Discovery check オブジェクト

次のオブジェクトは、dcheckAPI に直接関連するものです。

ディスカバリチェック

ディスカバリチェックオブジェクトは、ネットワークディスカバリルールによって実行される特定のチェックを定義します。以下のプロパティがあります。

プロパティ	型	説明
dcheckid	ID	ディスカバリチェックの ID。
druleid	ID	チェックが属するディスカバリルールの ID。
key_	string	アイテムキー（type が「Zabbix agent」に設定されている場合）または SNMP OID（type が「SNMPv1 agent」、「SNMPv2 agent」、「SNMPv3 agent」に設定されている場合）。 プロパティの動作: - type が「Zabbix agent」、「SNMPv1 agent」、「SNMPv2 agent」、「SNMPv3 agent」に設定されている場合は必須
ports	string	チェックする 1 つまたは複数のポート範囲。カンマで区切ります。 デフォルト: 0。 プロパティの動作: - type が「SSH」(0)、「LDAP」(1)、「SMTP」(2)、「FTP」(3)、「HTTP」(4)、「POP」(5)、「NNTP」(6)、「IMAP」(7)、「TCP」(8)、「Zabbix agent」(9)、「SNMPv1 agent」(10)、「SNMPv2 agent」(11)、「SNMPv3 agent」(13)、「HTTPS」(14)、「Telnet」(15) に設定されている場合にサポート
snmp_community	string	SNMP コミュニティ。 プロパティの動作: - type が「SNMPv1 agent」または「SNMPv2 agent」に設定されている場合は必須
snmpv3_authpassphrase	string	認証パスフレーズ。 プロパティの動作: - type が「SNMPv3 agent」に設定され、snmpv3_securitylevel が「authNoPriv」または「authPriv」に設定されている場合にサポート
snmpv3_authprotocol	integer	認証プロトコル。 指定可能な値: 0 - *(デフォルト)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512。 プロパティの動作: - type が「SNMPv3 agent」に設定され、snmpv3_securitylevel が「authNoPriv」または「authPriv」に設定されている場合にサポート

プロパティ	型	説明
snmpv3_contextname	string	SNMPv3 コンテキスト名。 プロパティの動作: - type が「SNMPv3 agent」に設定されている場合にサポート
snmpv3_privpassphrase	string	プライバシーパスフレーズ。 プロパティの動作: - type が「SNMPv3 agent」に設定され、snmpv3_securitylevel が「authPriv」に設定されている場合にサポート
snmpv3_privprotocol	integer	プライバシープロトコル。 指定可能な値: 0 - *(デフォルト)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C。 プロパティの動作: - type が「SNMPv3 agent」に設定され、snmpv3_securitylevel が「authPriv」に設定されている場合にサポート
snmpv3_securitylevel	string	セキュリティレベル。 指定可能な値: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv。 プロパティの動作: - type が「SNMPv3 agent」に設定されている場合にサポート
snmpv3_securityname	string	セキュリティ名。 プロパティの動作: - type が「SNMPv3 agent」に設定されている場合にサポート
type	integer	チェックのタイプ。 指定可能な値: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet。 プロパティの動作: - 必須

プロパティ	型	説明
uniq	integer	このチェックをデバイスの一意性判定基準として使用するかどうか。ディスカバリルールごとに一意のチェックは 1 つだけ設定できます。 指定可能な値: 0 - *(デフォルト)* このチェックを一意性判定基準として使用しない; 1 - このチェックを一意性判定基準として使用する。 プロパティの動作: - type が「Zabbix agent」、「SNMPv1 agent」、「SNMPv2 agent」、「SNMPv3 agent」に設定されている場合にサポート
host_source	integer	ホスト名の取得元。 指定可能な値: 1 - *(デフォルト)* DNS; 2 - IP; 3 - このチェックのディスカバリ値。
name_source	integer	表示名の取得元。 指定可能な値: 0 - *(デフォルト)* 指定なし; 1 - DNS; 2 - IP; 3 - このチェックのディスカバリ値。
allow_redirect	integer	ICMP ping のターゲットが異なる IP アドレスから応答した場合を許可するかどうか。 指定可能な値: 0 - *(デフォルト)* リダイレクト応答をターゲットホストがダウンしている（失敗）とみなす; 1 - リダイレクト応答をターゲットホストがアップしている（成功）とみなす。 プロパティの動作: - type が「ICMP ping」に設定されている場合にサポート

dcheck.get

説明

`integer/array dcheck.get(object parameters)`

このメソッドは、指定したパラメーターに応じたディスカバリチェックを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object`) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
dcheckids	ID/array	指定した ID のディスカバリチェックのみを返します。
druleids	ID/array	指定したディスカバリルールに属するディスカバリチェックのみを返します。
dserviceids	ID/array	指定した検出済みサービスを検出したディスカバリチェックのみを返します。
selectDRules	query	ディスカバリチェックに関連するディスカバリルールを返します。

パラメーター	タイプ	説明
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dcheckid`, `druleid`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールのディスカバリチェックを取得

ディスカバリルール"6" によって使用されるすべてのディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
        }
```

```
    ],
    "id": 1
}
```

ソース

CDCheck::get() in *ui/include/classes/api/services/CDCheck.php*.

Discovery rule

このクラスは、ネットワークディスカバリルールを操作するためのものです。

> **Note:**
> この API は、ネットワークディスカバリルールを扱うためのものです。ローレベルのディスカバリルールについては、LLD rule APIを参照してください。

オブジェクトリファレンス:

- Discovery rule

利用可能なメソッド:

- drule.create - 新しいディスカバリルールを作成する
- drule.delete - ディスカバリルールを削除する
- drule.get - ディスカバリルールを取得する
- drule.update - ディスカバリルールを更新する

Discovery rule オブジェクト

次のオブジェクトは、druleAPI に直接関連しています。

ディスカバリールール

ディスカバリールールオブジェクトは、ネットワークディスカバリールールを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
druleid	ID	ディスカバリールールの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
iprange	string	チェックする 1 つまたは複数の IP 範囲。カンマで区切ります。 サポートされている IP 範囲の形式については、ネットワークディスカバリの設定セクションを参照してください。 プロパティの動作: - 作成操作時に必須
name	string	ディスカバリールールの名前。 プロパティの動作: - 作成操作時に必須
delay	string	ディスカバリールールの実行間隔。 秒またはサフィックス付きの時間単位（例：30s、1m、2h、1d）、またはユーザーマクロを受け付けます。 デフォルト: 1h。
proxyid	ID	ディスカバリーに使用されるプロキシの ID。
status	integer	ディスカバリールールが有効かどうか。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。

プロパティ	タイプ	説明
concurrency_max	integer	ディスカバリールールごとの同時チェックの最大数。 可能な値: 0 - (デフォルト) チェック数無制限; 1 - 1 つのチェック; 2-999 - カスタムのチェック数。
error	string	ディスカバリールールの実行時に問題が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用

drule.create

説明

`object drule.create(object/array discoveryRules)`

このメソッドは、新しいディスカバリルールを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 作成するディスカバリルール。

discovery rule の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
dchecks	array	ディスカバリルール用に作成するディスカバリチェック。 パラメーターの動作: - 必須

戻り値

(object) druleids プロパティの下に作成されたディスカバリルールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたディスカバリルールの順序と一致します。

例

ディスカバリルールの作成

ローカルネットワークで Zabbix エージェントを実行しているマシンを見つけるためのディスカバリルールを作成します。ルールはポート 10050 で単一の Zabbix エージェントチェックを使用する必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.create",
    "params": {
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "concurrency_max": "10",
        "dchecks": [
            {
                "type": "9",
                "key_": "system.uname",
                "ports": "10050",
                "uniq": "0"
            }
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

参照

- Discovery check

ソース

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

説明

`object drule.delete(array discoveryRuleIds)`

このメソッドは、ディスカバリルールを削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するディスカバリルールの ID。

戻り値

(object) druleids プロパティの下にある削除されたディスカバリルールの ID を含むオブジェクトを返します。

例

複数のディスカバリルールの削除

2 つのディスカバリルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
```

```
        ]
    },
    "id": 1
}
```

ソース

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

説明

`integer/array drule.get(object parameters)`

このメソッドは、指定したパラメーターに応じたディスカバリールールを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
dhostids	ID/array	指定した検出済みホストを作成した検出ルールのみを返します。
druleids	ID/array	指定した ID の検出ルールのみを返します。
dserviceids	ID/array	指定した検出済みサービスを作成した検出ルールのみを返します。
selectDChecks	query	検出ルールで使用される検出チェックを持つ`dchecks`プロパティを返します。 `count` をサポートします。
selectDHosts	query	検出ルールで作成された検出済みホストを持つ`dhosts`プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectDChecks` - 結果は `dcheckid` でソートされます; `selectDHosts` - 結果は `dhostsid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `druleid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

`(integer/array)` 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのディスカバリルールの取得

設定されたすべてのディスカバリルールと、それらが使用するディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "druleid": "2",
            "proxyid": "0",
            "name": "Local network",
            "iprange": "192.168.3.1-255",
            "delay": "5s",
            "status": "0",
            "concurrency_max": "0",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "7",
                    "druleid": "2",
                    "type": "3",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "21",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                },
                {
                    "dcheckid": "8",
                    "druleid": "2",
                    "type": "4",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "80",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
```

```
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                }
            ]
        },
        {
            "druleid": "6",
            "proxyid": "0",
            "name": "Zabbix agent discovery",
            "iprange": "192.168.1.1-255",
            "delay": "1h",
            "status": "0",
            "concurrency_max": "10",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "10",
                    "druleid": "6",
                    "type": "9",
                    "key_": "system.uname",
                    "snmp_community": "",
                    "ports": "10050",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "2",
                    "name_source": "3",
                    "allow_redirect": "0"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Discovered host
- Discovery check

ソース

CDRule::get() in *ui/include/classes/api/services/CDRule.php*.

drule.update

説明

`object drule.update(object/array discoveryRules)`

このメソッドは、既存のディスカバリールールを更新することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 更新するディスカバリルールのプロパティ。

druleid プロパティはディスカバリルールごとに定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

discovery rule の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
dchecks	array	既存のチェックを置き換えるディスカバリチェック。

戻り値

(object) druleids プロパティの下で更新されたディスカバリルール ID を含むオブジェクトを返します。

例

ディスカバリルールの IP アドレス範囲の変更

ディスカバリルールの IP アドレス範囲を”192.168.2.1-255” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

参照

- Discovery check

ソース

CDRule::update() in *ui/include/classes/api/services/CDRule.php*.

Event

このクラスは、イベントを操作するためのものです。

オブジェクトリファレンス:

- Event
- Event tag
- Media type URL

利用可能なメソッド:

- event.get - イベントの取得
- event.acknowledge - イベントの確認

Event オブジェクト

次のオブジェクトは、eventAPI に直接関連しています。

イベント

> **Note:**
> イベントは Zabbix サーバーによって作成され、API 経由で変更することはできません。

イベントオブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
eventid	ID	イベントの ID。
source	integer	イベントのタイプ。 指定可能な値： 0 - トリガーによって作成されたイベント; 1 - ディスカバリールールによって作成されたイベント; 2 - アクティブエージェントの自動登録によって作成されたイベント; 3 - 内部イベント; 4 - サービスステータスの更新時に作成されたイベント。
object	integer	イベントに関連するオブジェクトのタイプ。 source が「トリガーによって作成されたイベント」に設定されている場合の指定可能な値： 0 - トリガー。 source が「ディスカバリールールによって作成されたイベント」に設定されている場合の指定可能な値： 1 - 検出されたホスト; 2 - 検出されたサービス。 source が「アクティブエージェントの自動登録によって作成されたイベント」に設定されている場合の指定可能な値： 3 - 自動登録されたホスト。 source が「内部イベント」に設定されている場合の指定可能な値： 0 - トリガー; 4 - アイテム; 5 - LLD ルール。 source が「サービスステータスの更新時に作成されたイベント」に設定されている場合の指定可能な値： 6 - サービス。
objectid	ID	関連オブジェクトの ID。
acknowledged	integer	イベントが認識されたかどうか。
clock	timestamp	イベントが作成された時刻。
ns	integer	イベントが作成されたナノ秒。
name	string	解決済みイベント名。

プロパティ	タイプ	説明
value	integer	関連オブジェクトの状態。 source が「トリガーによって作成されたイベント」または「サービスステータスの更新時に作成されたイベント」に設定されている場合の指定可能な値： 0 - OK; 1 - 問題。 source が「ディスカバリールールによって作成されたイベント」に設定されている場合の指定可能な値： 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが検出された; 3 - ホストまたはサービスが消失した。 source が「内部イベント」に設定されている場合の指定可能な値： 0 - 「正常」状態; 1 - 「不明」または「サポートされていない」状態。 プロパティの動作： - サポート source が「トリガーによって作成されたイベント」、「ディスカバリールールによって作成されたイベント」、「内部イベント」、または「サービスステータスの更新時に作成されたイベント」に設定されている場合
severity	integer	イベントの現在の深刻度。 指定可能な値： 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度; 4 - 高度; 5 - 障害。
r_eventid	ID	復旧イベントの ID。
c_eventid	ID	グローバル相関ルールの下で現在のイベントを上書き（クローズ）するために使用されたイベントの ID。正確な相関ルールを識別するには correlationid を参照してください。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
cause_eventid	ID	原因イベントの ID。
correlationid	ID	問題のクローズを生成した相関ルールの ID。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
userid	ID	イベントをクローズしたユーザーの ID（イベントが手動でクローズされた場合）。
suppressed	integer	イベントが抑制されているかどうか。 指定可能な値： 0 - イベントは通常状態; 1 - イベントは抑制されている。
opdata	string	展開されたマクロを含む運用データ。
urls	array	有効なメディアタイプ URL。

Event tag

Event tag オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
tag	string	Event tag name.
value	string	Event tag value.

メディアタイプ URL

メディアタイプ URL オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	メディアタイプで定義された URL 名。
url	string	メディアタイプで定義された URL 値。

結果には、イベントメニューエントリが有効なアクティブなメディアタイプのエントリのみが含まれます。プロパティで使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては、サポートされているマクロを参照してください。

event.acknowledge

説明

`object event.acknowledge(object/array parameters)`

このメソッドはイベントを更新するためのものです。以下の更新アクションを実行できます:

- イベントのクローズ。イベントがすでに解決済みの場合、このアクションはスキップされます。
- イベントのアクノレッジ。イベントがすでにアクノレッジされている場合、このアクションはスキップされます。
- イベントのアクノレッジ解除。イベントがアクノレッジされていない場合、このアクションはスキップされます。
- メッセージの追加。
- イベントの深刻度の変更。イベントがすでに同じ深刻度の場合、このアクションはスキップされます。
- イベントの抑制。イベントがすでに抑制されている場合、このアクションはスキップされます。
- イベントの抑制解除。イベントが抑制されていない場合、このアクションはスキップされます。

Attention:
トリガーイベントのみが更新できます。
問題イベントのみが更新できます。
イベントをクローズしたり、イベントの深刻度を変更するには、トリガーに対する読み書き権限が必要です。
イベントをクローズするには、トリガーで手動クローズが許可されている必要があります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。
メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) イベントの ID と実行する更新操作を含むパラメータ。

パラメータ	タイプ	説明
eventids	ID/array	承認するイベントの ID。 パラメータの動作: - 必須

パラメータ	タイプ	説明
action	integer	イベントの更新アクション。 可能なビットマップ値: 1 - 問題をクローズ; 2 - イベントを承認; 4 - メッセージを追加; 8 - 深刻度を変更; 16 - イベントの承認を解除; 32 - イベントを抑制; 64 - イベントの抑制を解除; 128 - イベントのランクを原因に変更; 256 - イベントのランクを症状に変更。 このフィールドはビットマスクです。可能なビットマップ値の任意の合計が許容されます（例: 34 はイベントの承認と抑制）。 パラメータの動作: - 必須
cause_eventid	ID	原因イベントの ID。 パラメータの動作: - `action` に「イベントのランクを症状に変更」ビットが含まれている場合は必須
message	string	メッセージのテキスト。 パラメータの動作: - `action` に「メッセージを追加」ビットが含まれている場合は必須
severity	integer	イベントの新しい深刻度。 可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度; 4 - 高度; 5 - 障害。 パラメータの動作: - `action` に「深刻度を変更」ビットが含まれている場合は必須
suppress_until	integer	イベントを抑制する Unix タイムスタンプ。 ”0” に設定すると、抑制は無期限になります。 パラメータの動作: - `action` に「イベントを抑制」ビットが含まれている場合は必須

戻り値

(`object`) `eventids` プロパティの下で更新されたイベントの ID を含むオブジェクトを返します。

例

イベントの確認

1 つのイベントを確認し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
```

```
        "message": "Problem resolved."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

イベントの深刻度の変更

複数のイベントの深刻度を変更し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

ソース

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

説明

`integer/array event.get(object parameters)`

このメソッドは、指定されたパラメータに従ってイベントを取得することができます。

> **Attention:**
> このメソッドは、ハウスキーパーによってまだ削除されていない場合、削除されたエンティティのイベントを返すことがあります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
eventids	ID/array	指定した ID のイベントのみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成されたイベントのみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成されたイベントのみを返します。
objectids	ID/array	指定したオブジェクトによって作成されたイベントのみを返します。
source	integer	指定したタイプのイベントのみを返します。 サポートされているイベントタイプの一覧は、event object pageを参照してください。 デフォルト: 0 - トリガーイベント。
object	integer	指定したタイプのオブジェクトによって作成されたイベントのみを返します。 サポートされているオブジェクトタイプの一覧は、event object pageを参照してください。 デフォルト: 0 - トリガー。
acknowledged	boolean	true に設定すると、認識済みのイベントのみを返します。
action	integer	指定したイベント更新アクションが実行されたイベントのみを返します。 複数のアクションの場合、任意のビットマップ値の合計をビットマスクとして使用します（例: 34 は認識と抑制イベント）。
action_userids	ID/array	イベント更新アクションを実行したユーザーの ID が指定されたイベントのみを返します。
suppressed	boolean	true - 抑制されたイベントのみを返します; false - 通常状態のイベントを返します。
symptom	boolean	true - 症状イベントのみを返します; false - 原因イベントのみを返します。
severities	integer/array	指定したイベントの深刻度のイベントのみを返します。object がトリガーの場合のみ適用されます。
trigger_severities	integer/array	指定したトリガーの深刻度のイベントのみを返します。object がトリガーの場合のみ適用されます。
evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグのイベントのみを返します。 フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]。 空の配列はすべてのイベントを返します。 可能なoperatorの値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
eventid_from	string	指定した ID 以上の ID を持つイベントのみを返します。
eventid_till	string	指定した ID 以下の ID を持つイベントのみを返します。

パラメーター	タイプ	説明
time_from	timestamp	指定した時刻以降に作成されたイベントのみを返します。
time_till	timestamp	指定した時刻以前に作成されたイベントのみを返します。
problem_time_from	timestamp	現在の状態に関係なく、problem_time_from から問題状態にあったイベントのみを返します。ソースがトリガーイベントで、オブジェクトがトリガーの場合のみ適用されます。problem_time_till が指定されている場合、このパラメーターは必須です。
problem_time_till	timestamp	現在の状態に関係なく、problem_time_till まで問題状態にあったイベントのみを返します。ソースがトリガーイベントで、オブジェクトがトリガーの場合のみ適用されます。problem_time_from が指定されている場合、このパラメーターは必須です。
value	integer/array	指定した値のイベントのみを返します。
selectAcknowledges	query	イベント更新を含む acknowledges プロパティを返します。イベント更新は逆時系列でソートされます。 イベント更新オブジェクトは以下のプロパティを持ちます: acknowledgeid - (ID) 認識の ID; userid - (ID) イベントを更新したユーザーの ID; clock - (timestamp) イベントが更新された時刻; message - (string) メッセージのテキスト; action - (integer) 実行された更新アクション、event.acknowledgeを参照; old_severity - (integer) この更新アクション前のイベントの深刻度; new_severity - (integer) この更新アクション後のイベントの深刻度; suppress_until - (timestamp) イベントが抑制されるまでの時刻; taskid - (ID) 現在のイベントがランク変更中の場合のタスク ID; username - (string) イベントを更新したユーザーのユーザー名; name - (string) イベントを更新したユーザーの名前; surname - (string) イベントを更新したユーザーの姓。 count をサポートします。
selectAlerts	query	イベントによって生成されたアラートを含むalertsプロパティを返します。アラートは逆時系列でソートされます。
selectHosts	query	イベントを作成したオブジェクトを含むホストのhostsプロパティを返します。トリガー、アイテム、LLD ルールによって生成されたイベントのみサポートされます。
selectRelatedObject	query	イベントを作成したオブジェクトを含む relatedObject プロパティを返します。返されるオブジェクトのタイプはイベントタイプによって異なります。
selectSuppressionData	query	アクティブなメンテナンスおよび手動抑制のリストを含む suppression_data プロパティを返します: maintenanceid - (ID) メンテナンスの ID; userid - (ID) イベントを抑制したユーザーの ID; suppress_until - (integer) イベントが抑制されるまでの時刻。
selectTags	query	イベントタグを含むtagsプロパティを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 textデータ型のプロパティはサポートしません。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: eventid, objectid, clock。 groupBy と一緒に使用する場合の可能な値: objectid。 countOutput および groupBy と一緒に使用する場合の可能な値: objectid, rowscount。

パラメーター	タイプ	説明
groupBy	string/array	指定したプロパティで結果をグループ化します。指定したプロパティは結果に返されます。 可能な値: `objectid`。
countOutput	boolean	これらのパラメーターはreference commentaryで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されているが、`groupBy` パラメータが使用されていない場合、取得したオブジェクトの数
- `groupBy` パラメーターが使用されている場合、集計結果を含むオブジェクトの配列

例

トリガーイベントの取得

トリガー"22395" から最新のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
```

```
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "suppressed": "0",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
```

```
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


期間によるイベントの取得

2012 年 10 月 17 日から 18 日までに作成されたすべてのイベントを、逆時系列で取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
```

```
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Less than 25% free in the history cache",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


指定したユーザーによって認識されたイベントの取得

ID=10 のユーザーによって認識されたイベントの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

問題イベント数によるトップトリガーの取得

指定した期間内に「警告」、「軽度の障害」、「重度の障害」、「致命的な障害」の深刻度を持つトリガーのうち、問題イベント数が多い上位 5 件を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "countOutput": true,
        "groupBy": "objectid",
        "source": 0,
```

```
        "object": 0,
        "value": 1,
        "time_from": 1672531200,
        "time_till": 1677628800,
        "trigger_severities": [2, 3, 4, 5],
        "sortfield": ["rowscount"],
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "objectid": "232124",
            "rowscount": "27"
        },
        {
            "objectid": "29055",
            "rowscount": "23"
        },
        {
            "objectid": "253731",
            "rowscount": "18"
        },
        {
            "objectid": "254062",
            "rowscount": "11"
        },
        {
            "objectid": "23216",
            "rowscount": "7"
        }
    ],
    "id": 1
}
```

参照

- Alert
- Item
- Host
- LLD rule
- Service
- Trigger

ソース

CEvent::get() in *ui/include/classes/api/services/CEvent.php*.

Graph

このクラスは、グラフを操作するためのものです。

オブジェクトリファレンス:

- Graph

利用可能なメソッド:

- graph.create - 新しいグラフの作成
- graph.delete - グラフの削除
- graph.get - グラフの取得

- graph.update - グラフの更新

Graph オブジェクト

次のオブジェクトは、graphAPI に直接関連しています。

グラフ

グラフオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
graphid	ID	グラフの ID。 プロパティの挙動: - 読み取り専用 - 更新操作時に必須
height	integer	グラフの高さ（ピクセル単位）。 プロパティの挙動: - 作成操作時に必須
name	string	グラフの名前。 プロパティの挙動: - 作成操作時に必須
width	integer	グラフの幅（ピクセル単位）。 プロパティの挙動: - 作成操作時に必須
flags	integer	グラフの起源。 可能な値： 0 - *(デフォルト)* 通常のグラフ; 4 - プロトタイプから変換されたグラフ。 プロパティの挙動: - 読み取り専用
graphtype	integer	グラフのレイアウトタイプ。 可能な値： 0 - *(デフォルト)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 爆発円グラフ。
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	円グラフおよび爆発円グラフを 3D で表示するかどうか。 可能な値： 0 - *(デフォルト)* 2D で表示; 1 - 3D で表示。
show_legend	integer	グラフに凡例を表示するかどうか。 可能な値： 0 - 非表示; 1 - *(デフォルト)* 表示。

プロパティ	タイプ	説明
show_work_period	integer	グラフに稼働時間を表示するかどうか。 可能な値： 0 - 非表示; 1 - *(デフォルト)* 表示。
show_triggers	integer	グラフにトリガーラインを表示するかどうか。 可能な値： 0 - 非表示; 1 - *(デフォルト)* 表示。
templateid	ID	親テンプレートグラフの ID。 プロパティの挙動: - 読み取り専用
yaxismax	float	Y 軸の固定最大値。 デフォルト: 100。
yaxismin	float	Y 軸の固定最小値。 デフォルト: 0。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 指定されたアイテムへのアクセス権がない場合、`ymax_type` が「計算」に設定されているかのようにグラフが描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 可能な値： 0 - *(デフォルト)* 計算; 1 - 固定値; 2 - アイテム。
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 指定されたアイテムへのアクセス権がない場合、`ymin_type` が「計算」に設定されているかのようにグラフが描画されます。
ymin_type	integer	Y 軸の最小値の計算方法。 可能な値： 0 - *(デフォルト)* 計算; 1 - 固定値; 2 - アイテム。
uuid	string	インポートされたグラフを既存のグラフにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。 プロパティの挙動: - グラフがテンプレートに属している場合にサポート

graph.create

説明

`object graph.create(object/array graphs)`

このメソッドは、新しいグラフを作成することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object/array)` 作成するグラフ。

graph の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
gitems	array	グラフ用に作成するグラフアイテム。 パラメーターの動作: - 必須

戻り値

(object) graphids プロパティの下に作成されたグラフの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグラフの順序と一致します。

例

グラフの作成

2 つのアイテムからなるグラフを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.create",
    "params": {
        "name": "MySQL bandwidth",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

参照

- Graph item

ソース

CGraph::create() in *ui/include/classes/api/services/CGraph.php*.

graph.delete

説明

`object graph.delete(array graphIds)`

このメソッドは、グラフを削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグラフの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフの ID を含むオブジェクトを返します。

例

複数のグラフの削除

2 つのグラフを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraph::delete() in *ui/include/classes/api/services/CGraph.php*.

graph.get

説明

`integer/array graph.get(object parameters)`

このメソッドは、指定したパラメーターに応じたグラフを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
graphids	ID/array	指定した ID のグラフのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのグラフのみを返します。

パラメーター	型	説明
templateids	ID/array	指定したテンプレートに属するグラフのみを返します。
hostids	ID/array	指定したホストに属するグラフのみを返します。
itemids	ID/array	指定したアイテムを含むグラフのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するグラフのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたグラフのみを返します。
expandName	flag	グラフ名のマクロを展開します。
selectDiscoveryData	query	グラフディスカバリーオブジェクトデータを持つ `discoveryData` プロパティを返します。グラフディスカバリーオブジェクトは、検出されたグラフを、そのグラフが検出されたグラフプロトタイプにリンクします。 以下のプロパティを持ちます: `parent_graphid` - (ID) グラフが作成されたグラフプロトタイプのID; `status` - (int) グラフディスカバリーステータス: 0 - *(デフォルト)* グラフが検出されている; 1 - グラフがもはや検出されていない; `ts_delete` - (timestamp) もはや検出されていないグラフが削除される時刻。
selectDiscoveryRule	query	グラフを作成した低レベルディスカバリールールを持つdiscoveryRuleプロパティを返します。
selectHostGroups	query	グラフが属するホストグループを持つhostgroupsプロパティを返します。
selectTemplateGroups	query	グラフが属するテンプレートグループを持つtemplategroupsプロパティを返します。
selectTemplates	query	グラフが属するテンプレートを持つtemplatesプロパティを返します。
selectHosts	query	グラフが属するホストを持つhostsプロパティを返します。
selectItems	query	グラフで使用されているアイテムを持つitemsプロパティを返します。
selectGraphItems	query	グラフで使用されているアイテムを持つgitemsプロパティを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 `text`データ型のプロパティはサポートされていません。 追加プロパティをサポートします: `host` - グラフが属するホストの技術名; `hostid` - グラフが属するホストの ID。
sortfield	string/array	指定したプロパティで結果をソートします。 利用可能な値: `graphid`, `name`, `graphtype`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGraphDiscovery	query	グラフディスカバリーオブジェクトを持つ `graphDiscovery` プロパティを返します。グラフディスカバリーオブジェクトは、グラフを作成元のグラフプロトタイプにリンクします。 このクエリは非推奨です。代わりに `selectDiscoveryData` を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストからグラフの取得

ホスト"10107" から全てのグラフを取得し、名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "612",
            "name": "CPU jumps",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "439",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "613",
            "name": "CPU load",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "433",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
```

```
            "ymin_type": "1",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "614",
            "name": "CPU utilization",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "387",
            "show_work_period": "1",
            "show_triggers": "0",
            "graphtype": "1",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "1",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "645",
            "name": "Disk space usage /",
            "width": "600",
            "height": "340",
            "yaxismin": "0",
            "yaxismax": "0",
            "templateid": "0",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "4"
        }
    ],
    "id": 1
}
```


参照

- Discovery rule
- Graph item
- Item
- Host
- Host group
- Template
- Template group

ソース

CGraph::get() in *ui/include/classes/api/services/CGraph.php*.

graph.update

説明

`object graph.update(object/array graphs)`

このメソッドは、既存のグラフを更新するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプのみが利用できます。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するグラフのプロパティ。

各グラフには graphid プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべてのプロパティは変更されません。

標準のグラフプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムに gitemid プロパティが定義されている場合は更新され、そうでない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新されたグラフの ID を含むオブジェクトを返す。

例

Y スケールの最大値の設定

Y スケールの最大値を 100 の固定値に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraph::update() in *ui/include/classes/api/services/CGraph.php*.

Graph item

このクラスは、グラフアイテムを操作するためのものです。

オブジェクトリファレンス:

- Graph item

利用可能なメソッド:

- graphitem.get - グラフアイテムの取得

Graph item オブジェクト

次のオブジェクトは、graphitemAPI に直接関連しています。

グラフアイテム

> **Note:**
> グラフアイテムは graph API を介してのみ変更できます。

グラフアイテムオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
gitemid	ID	グラフアイテムの ID。 プロパティの動作: - 読み取り専用
color	string	グラフアイテムの描画色（16 進数カラーコード）。 プロパティの動作: - 作成操作時に必須
itemid	ID	アイテムの ID。 プロパティの動作: - 作成操作時に必須
calc_fnc	integer	表示されるアイテムの値。 可能な値： 1 - 最小値; 2 - *(デフォルト)* 平均値; 4 - 最大値; 7 - すべての値; 9 - 最後の値（円グラフと爆発グラフのみで使用）。
drawtype	integer	グラフアイテムの描画スタイル。 可能な値： 0 - *(デフォルト)* 線; 1 - 塗りつぶし領域; 2 - 太線; 3 - 点; 4 - 破線; 5 - グラデーション線。
graphid	ID	グラフアイテムが属するグラフの ID。
sortorder	integer	グラフ内でのアイテムの位置。 デフォルト：”0” から始まり、エントリごとに 1 ずつ増加。
type	integer	グラフアイテムのタイプ。 可能な値： 0 - *(デフォルト)* シンプル; 2 - グラフ合計（円グラフと爆発グラフのみで使用）。

プロパティ	タイプ	説明
yaxisside	integer	グラフアイテムの Y スケールが描画されるグラフの側。 可能な値： 0 - *(デフォルト)* 左側; 1 - 右側。

graphitem.get

説明

`integer/array graphitem.get(object parameters)`

このメソッドは、指定したパラメーターに応じたグラフアイテムを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
graphids	ID/array	指定したグラフに属するグラフアイテムのみを返します。
itemids	ID/array	指定したアイテム ID のグラフアイテムのみを返します。
type	integer	指定したタイプのグラフアイテムのみを返します。 サポートされているグラフアイテムタイプの一覧はグラフアイテムオブジェクトページを参照してください。
selectGraphs	query	アイテムが属するグラフの配列を持つ`graphs`プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `gitemid`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

グラフからグラフアイテムの取得

グラフで使用されているすべてのグラフアイテムを、アイテムとホストに関する追加情報とともに取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "gitemid": "1242",
            "graphid": "387",
            "itemid": "22665",
            "drawtype": "1",
            "sortorder": "1",
            "color": "FF5555",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1243",
            "graphid": "387",
            "itemid": "22668",
            "drawtype": "1",
            "sortorder": "2",
            "color": "55FF55",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1244",
            "graphid": "387",
            "itemid": "22671",
            "drawtype": "1",
            "sortorder": "3",
            "color": "009999",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        }
    ],
    "id": 1
}
```

参照

- Graph

ソース

CGraphItem::get() in *ui/include/classes/api/services/CGraphItem.php*.

Graph prototype

このクラスは、グラフプロトタイプを操作するためのものです。

オブジェクトリファレンス:

- Graph prototype

利用可能なメソッド:

- graphprototype.create - 新しいグラフプロトタイプを作成する
- graphprototype.delete - グラフプロトタイプを削除する
- graphprototype.get - グラフプロトタイプを取得する
- graphprototype.update - グラフプロトタイプを更新する

Graph prototype オブジェクト

次のオブジェクトは、graphprototypeAPI に直接関連しています。

グラフプロトタイプ

グラフプロトタイプオブジェクトは以下のプロパティを持ちます。

プロパティ	型	説明
graphid	ID	グラフプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
height	integer	グラフプロトタイプの高さ（ピクセル単位）。 プロパティの動作: - 作成操作時に必須
name	string	グラフプロトタイプの名前。 プロパティの動作: - 作成操作時に必須
width	integer	グラフプロトタイプの幅（ピクセル単位）。 プロパティの動作: - 作成操作時に必須
flags	integer	グラフプロトタイプの起源。 可能な値： 2 - グラフプロトタイプ; 6 - 検出されたグラフプロトタイプ プロパティの動作: - 読み取り専用
graphtype	integer	グラフプロトタイプのレイアウトタイプ。 可能な値： 0 - *(デフォルト)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 爆発円グラフ。
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	検出された円グラフおよび爆発円グラフを 3D で表示するかどうか。 可能な値： 0 - *(デフォルト)* 2D で表示; 1 - 3D で表示。
show_legend	integer	検出されたグラフに凡例を表示するかどうか。 可能な値： 0 - 非表示; 1 - *(デフォルト)* 表示。
show_work_period	integer	検出されたグラフに作業時間を表示するかどうか。 可能な値： 0 - 非表示; 1 - *(デフォルト)* 表示。

プロパティ	型	説明
templateid	ID	親テンプレートグラフプロトタイプの ID。 プロパティの動作: - 読み取り専用
yaxismax	float	Y 軸の固定最大値。
yaxismin	float	Y 軸の固定最小値。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 指定されたアイテムへのアクセス権がない場合、ymax_type が「計算」に設定されているかのようにグラフが描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 可能な値： 0 - *(デフォルト)* 計算; 1 - 固定値; 2 - アイテム。
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 指定されたアイテムへのアクセス権がない場合、ymin_type が「計算」に設定されているかのようにグラフが描画されます。
ymin_type	integer	Y 軸の最小値の計算方法。 可能な値： 0 - *(デフォルト)* 計算; 1 - 固定値; 2 - アイテム。
discover	integer	グラフプロトタイプの検出ステータス。 可能な値： 0 - *(デフォルト)* 新しいグラフが検出される; 1 - 新しいグラフは検出されず、既存のグラフはロストとしてマークされる。
uuid	string	インポートされたグラフプロトタイプを既存のものにリンクするために使用されるユニバーサル一意識別子。指定がない場合は自動生成されます。 プロパティの動作: - グラフプロトタイプがテンプレートに属している場合にサポート

graphprototype.create

説明

`object graphprototype.create(object/array graphPrototypes)`

このメソッドは新しいグラフプロトタイプを作成するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 作成するグラフのプロトタイプ。

標準のグラフプロトタイプのプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
gitems	array	グラフプロトタイプに作成するグラフアイテム。グラフアイテムはアイテムとアイテムプロトタイプの両方を参照できますが、少なくとも 1 つのアイテムプロトタイプが必要です。 パラメーターの動作: - 必須

戻り値

(object) graphids プロパティの下に作成されたグラフプロトタイプの ID を含むオブジェクトを返します。
 返された ID の順番は、渡されたグラフプロトタイプの順番と一致する。

例

グラフプロトタイプの作成

2 つのアイテムを持つグラフプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

参照

- Graph item

ソース

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

説明

```
object graphprototype.delete(array graphPrototypeIds)
```

このメソッドは、グラフプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグラフプロトタイプの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフプロトタイプの ID を含むオブジェクトを返します。

例

複数のグラフプロトタイプの削除

2 つのグラフプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

説明

`integer/array graphprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従ってグラフプロトタイプを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
discoveryids	ID/array	指定したディスカバリールールに属するグラフのプロトタイプのみを返します。
graphids	ID/array	指定した ID のグラフのプロトタイプのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのグラフのプロトタイプのみを返します。
hostids	ID/array	指定したホストに属するグラフのプロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたグラフのプロトタイプのみを返します。
itemids	ID/array	指定したアイテムのプロトタイプを含むグラフのプロトタイプのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するグラフのプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するグラフのプロトタイプのみを返します。
selectDiscoveryData	query	グラフのプロトタイプディスカバリーオブジェクトデータを持つ `discoveryData` プロパティを返します。グラフのプロトタイプディスカバリーオブジェクトは、検出されたグラフのプロトタイプを、検出元のグラフのプロトタイプにリンクします。 以下のプロパティを持ちます: `parent_graphid` - (ID) グラフが作成されたグラフのプロトタイプの ID; `status` - (int) グラフディスカバリーステータス: 0 - *(デフォルト)* グラフのプロトタイプが検出されている; 1 - グラフのプロトタイプがこれ以上検出されていない; `ts_delete` - (timestamp) もはや検出されていないグラフのプロトタイプが削除される時刻。
selectDiscoveryRule	query	グラフのプロトタイプが属する LLD ルールを持つ`discoveryRule`プロパティを返します。
selectDiscoveryRulePrototype	query	グラフのプロトタイプが属する親 LLD ルールプロトタイプを持つ`discoveryRulePrototype`プロパティを返します。
selectGraphItems	query	グラフのプロトタイプで使用されているグラフアイテムを持つ`gitems`プロパティを返します。
selectHostGroups	query	グラフのプロトタイプが属するホストグループを持つ`hostgroups`プロパティを返します。
selectHosts	query	グラフのプロトタイプが属するホストを持つ`hosts`プロパティを返します。
selectItems	query	グラフのプロトタイプで使用されているアイテムおよびアイテムのプロトタイプを持つ `items` プロパティを返します。
selectTemplateGroups	query	グラフのプロトタイプが属するテンプレートグループを持つ`templategroups`プロパティを返します。
selectTemplates	query	グラフのプロトタイプが属するテンプレートを持つ`templates`プロパティを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一の値または値の配列です。 `text`データ型のプロパティはサポートされません。 追加プロパティをサポートします: `host` - グラフのプロトタイプが属するホストの技術名; `hostid` - グラフのプロトタイプが属するホストの ID。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `graphid`, `name`, `graphtype`, `discovered`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	

パラメーター	タイプ	説明
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

LLD ルールから graph prototypes を取得する

LLD ルールからすべての graph prototypes を取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "parent_itemid": "27426",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "discover": "0"
        }
    ],
    "id": 1
}
```

LLD ルールからグラフプロトタイプを取得する

LLD ルールからすべてのグラフプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2"
            "discover": "0"
        }
    ],
    "id": 1
}
```

参考

- ディスカバリールール
- グラフアイテム
- アイテム
- ホスト
- ホストグループ
- テンプレート
- テンプレートグループ

ソース

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

説明

`object graphprototype.update(object/array graphPrototypes)`

このメソッドは、既存のグラフプロトタイプを更新するためのものです。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプのみが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するグラフプロトタイプのプロパティ。

各グラフプロトタイプには graphid プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のグラフプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムに gitemid プロパティが定義されている場合は更新され、定義されていない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新された graph prototypes の ID を含むオブジェクトを返します。

例

グラフプロトタイプのサイズ変更

グラフプロトタイプのサイズを 1100×400 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::update() in *ui/include/classes/api/services/CGraphPrototype.php*.

High availability node

このクラスは、高可用性クラスターの一部であるサーバーノードや、スタンドアロンのサーバーインスタンスを操作するためのものです。

オブジェクトリファレンス:

- High availability node

利用可能なメソッド:

- hanode.get - ノードの取得

High availability node オブジェクト

次のオブジェクトは、Zabbix サーバーの HA (高可用性) クラスタの運用に関連しています。

高可用性ノード

> **Note:**
> ノードは Zabbix サーバーによって作成され、API 経由で変更することはできません。

高可用性ノードオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
ha_nodeid	ID	ノードの ID。
name	string	zabbix_server.conf の HANodeName 設定エントリを使用してノードに割り当てられた名前。スタンドアロンモードで実行されているサーバーの場合は空。
address	string	ノードが接続する IP または DNS 名。
port	integer	ノードが稼働しているポート。
lastaccess	integer	ハートビート時刻、つまりノードからの最終更新時刻。UTC タイムスタンプ。
status	integer	ノードの状態。 可能な値: 0 - 待機; 1 - 手動で停止; 2 - 利用不可; 3 - アクティブ;

hanode.get

説明

`integer/array hanode.get(object parameters)`

このメソッドは、指定されたパラメータに従って、HA クラスタノードのリストを取得することができます。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。詳細はユーザーロールを参照してください。

パラメーター

(`object`) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
ha_nodeids	ID/array	指定したノード ID を持つノードのみを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 サポートされるプロパティ: `name`, `address`, `status`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `name`, `lastaccess`, `status`。
countOutput	flag	これらのパラメーターはリファレンス解説で説明されています。
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ステータスで並べ替えたノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",
      "address": "192.168.1.6",
      "port": "10051",
      "lastaccess": "1635335705",
      "status": "0"
    }
  },
  "id": 1
}
```

ID で特定のノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "name": "node4",
            "address": "192.168.1.8",
            "port": "10052",
            "lastaccess": "1635334214",
            "status": "1"
        },
        {
            "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
            "name": "node6",
            "address": "192.168.1.10",
            "port": "10053",
            "lastaccess": "1635332902",
            "status": "2"
        }
    ],
    "id": 1
}
```

停止したノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
            "address": "192.168.1.12",
            "port": "10051"
        },
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "address": "192.168.1.8",
            "port": "10052"
        },
```

```
        {
            "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
            "address": "192.168.1.9",
            "port": "10053"
        }
    ],
    "id": 1
}
```

スタンバイノードの数を取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

特定の IP アドレスでノードのステータスを確認する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

ソース

CHaNode::get() in *ui/include/classes/api/services/CHaNode.php*.

History

このクラスは、ヒストリデータを操作するためのものです。

オブジェクトリファレンス:

- Float history
- Integer history
- String history
- Text history
- Log history

利用可能なメソッド:

- history.clear - ヒストリデータをクリアする
- history.get - ヒストリデータを取得する
- history.push - Zabbix サーバーへヒストリデータを送信する

History オブジェクト

次のオブジェクトは、`history`API に直接関連しています。

> **Note:**
> History オブジェクトはアイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API から変更することはできません。

浮動小数点履歴

浮動小数点履歴オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値を受信した時刻
itemid	ID	関連するアイテムの ID
ns	integer	値を受信したナノ秒
value	float	受信した値

整数履歴

整数履歴オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値が受信された時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値が受信されたナノ秒。
value	integer	受信した値。

文字列履歴

文字列履歴オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信したナノ秒。
value	string	受信した値。

テキスト履歴

テキスト履歴オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
id	ID	履歴エントリの ID。
clock	timestamp	値が受信された時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値が受信されたナノ秒。
value	text	受信した値。

ログ履歴

ログ履歴オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
id	ID	履歴エントリの ID。
clock	timestamp	値が受信された時刻。
itemid	ID	関連するアイテムの ID。
logeventid	integer	Windows イベントログエントリの ID。
ns	integer	値が受信されたナノ秒。
severity	integer	Windows イベントログエントリのレベル。
source	string	Windows イベントログエントリのソース。
timestamp	timestamp	Windows イベントログエントリの時刻。
value	text	受信した値。

history.clear

説明

`object history.clear(array itemids)`

このメソッドは、アイテムのヒストリを消去することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) ヒストリを消去するアイテムの ID。

戻り値

(object) itemids プロパティの下にあるヒストリが消去されたアイテムの ID を含むオブジェクトを返します。

例

ヒストリの消去

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
```

```
            "13205"
        ]
    },
    "id": 1
}
```

ソース

CHistory::clear() in *ui/include/classes/api/services/CHistory.php*.

history.get

説明

`integer/array history.get(object parameters)`

このメソッドは、指定されたパラメータに従って履歴データを取得するためのものです。

Attention:
このメソッドは、ハウスキーパーによってまだ削除されていない場合、削除されたエンティティの履歴データを返すことがあります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
history	integer	返す履歴オブジェクトのタイプ。 可能な値: 0 - 数値 (浮動小数点); 1 - 文字列; 2 - ログ; 3 - *(デフォルト)* 数値 (符号なし); 4 - テキスト; 5 - バイナリ。
hostids	ID/array	指定したホストからの履歴のみを返す。
itemids	ID/array	指定したアイテムからの履歴のみを返す。
time_from	timestamp	指定した時刻以降に受信した値のみを返す。
time_till	timestamp	指定した時刻以前に受信した値のみを返す。
sortfield	string/array	指定したプロパティで結果をソートする。 可能な値: `itemid`, `clock`, `ns`.
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

アイテム履歴データの取得

数値 (浮動小数点) アイテムから受信した最新の 10 個の値を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23296",
            "clock": "1351090996",
            "value": "0.085",
            "ns": "563157632"
        },
        {
            "itemid": "23296",
            "clock": "1351090936",
            "value": "0.16",
            "ns": "549216402"
        },
        {
            "itemid": "23296",
            "clock": "1351090876",
            "value": "0.18",
            "ns": "537418114"
        },
        {
            "itemid": "23296",
            "clock": "1351090816",
            "value": "0.21",
            "ns": "522659528"
        },
        {
            "itemid": "23296",
            "clock": "1351090756",
            "value": "0.215",
            "ns": "507809457"
        },
        {
            "itemid": "23296",
            "clock": "1351090696",
            "value": "0.255",
            "ns": "495509699"
```

```
        },
        {
            "itemid": "23296",
            "clock": "1351090636",
            "value": "0.36",
            "ns": "477708209"
        },
        {
            "itemid": "23296",
            "clock": "1351090576",
            "value": "0.375",
            "ns": "463251343"
        },
        {
            "itemid": "23296",
            "clock": "1351090516",
            "value": "0.315",
            "ns": "447947017"
        },
        {
            "itemid": "23296",
            "clock": "1351090456",
            "value": "0.275",
            "ns": "435307141"
        }
    ],
    "id": 1
}
```


ソース

CHistory::get() in *ui/include/classes/api/services/CHistory.php*.

history.push

説明

`object history.push(object/array itemHistoryData)`

このメソッドは、アイテムヒストリデータを Zabbix サーバーに送信できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

Parameters

(`object/array`) 送信するアイテム履歴データ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
itemid	ID	関連アイテムの ID パラメータの動作: - 必須 `host` と `key` が設定されていない場合
host	文字列	ホストの技術名 パラメータの動作: - 必須 `itemid` が設定されていない場合
key	文字列	アイテムキー パラメータの動作: - 必須 `itemid` が設定されていない場合
value	mixed	アイテム値 パラメータの動作: - 必須

パラメータ	タイプ	説明
clock	タイムスタンプ	値を受信した時刻
ns	整数	値を受信したナノ秒数

戻り値

(object) データ送信操作の結果を返します。

例

アイテムヒストリデータの送信

Zabbix サーバーへ、アイテム"10600"、"10601"、"999999" のヒストリデータを送信します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
```

```
    },
    "id": 1
}
```

参照

- Trapper items
- HTTP agent items
- Host
- Item

ソース

CHistory::push() in *ui/include/classes/api/services/CHistory.php*.

Host

このクラスは、ホストを操作するためのものです。

オブジェクトリファレンス:

- Host
- Host inventory
- Host tag

利用可能なメソッド:

- host.create - 新しいホストを作成する
- host.delete - ホストを削除する
- host.get - ホストを取得する
- host.massadd - ホストへ関連オブジェクトを追加する
- host.massremove - ホストから関連オブジェクトを削除する
- host.massupdate - ホストの関連オブジェクトを交換または削除する
- host.update - ホストを更新する

Host オブジェクト

次のオブジェクトは、hostAPI に直接関連しています。

ホスト

ホストオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
hostid	ID	ホストの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
host	string	ホストの技術名。 プロパティの動作: - 作成操作では必須
description	text	ホストの説明。
flags	integer	ホストの起源。 可能な値: 0 - 通常のホスト; 4 - プロトタイプから変換されたホスト。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
inventory_mode	integer	ホストインベントリの収集モード。 可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
ipmi_authtype	integer	IPMI 認証アルゴリズム。 可能な値: -1 - *(デフォルト)* デフォルト; 0 - なし; 1 - MD2; 2 - MD5 4 - ストレート; 5 - OEM; 6 - RMCP+。
ipmi_password	string	IPMI パスワード。
ipmi_privilege	integer	IPMI 権限レベル。 可能な値: 1 - コールバック; 2 - *(デフォルト)* ユーザー; 3 - オペレーター; 4 - 管理者; 5 - OEM。
ipmi_username	string	IPMI ユーザー名。
maintenance_from	timestamp	有効なメンテナンスの開始時刻。 プロパティの動作: - 読み取り専用
maintenance_status	integer	有効なメンテナンスステータス。 可能な値: 0 - *(デフォルト)* メンテナンスなし; 1 - メンテナンス中。 プロパティの動作: - 読み取り専用
maintenance_type	integer	有効なメンテナンスタイプ。 可能な値: 0 - *(デフォルト)* データ収集ありのメンテナンス; 1 - データ収集なしのメンテナンス。 プロパティの動作: - 読み取り専用
maintenanceid	ID	現在ホストで有効なメンテナンスの ID。 プロパティの動作: - 読み取り専用
name	string	ホストの表示名。 デフォルト: `host` プロパティの値。
monitored_by	integer	ホストの監視に使用されるソース。 可能な値: 0 - *(デフォルト)* Zabbix サーバー; 1 - プロキシ; 2 - プロキシグループ。

プロパティ	タイプ	説明
proxyid	ID	ホストの監視に使用されるプロキシの ID。 プロパティの動作: - `monitored_by` が「プロキシ」に設定されている場合は必須
proxy_groupid	ID	ホストの監視に使用されるプロキシグループの ID。 プロパティの動作: - `monitored_by` が「プロキシグループ」に設定されている場合は必須
status	integer	ホストのステータスと機能。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
tls_connect	integer	ホストへの接続。 可能な値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	ホストからの接続。 可能なビットマップ値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドであり、可能なビットマップ値の合計値が許容されます（例: PSK と証明書の場合は 6）。
tls_issuer	string	証明書発行者。
tls_subject	string	証明書サブジェクト。
tls_psk_identity	string	PSK 識別子；1 つの PSK（自動登録、ホスト、プロキシ全体で）とだけペアにする必要があります。 PSK 識別子には機密情報を含めないでください。ネットワーク上で暗号化されずに送信され、受信者にどの PSK を使用するかを通知します。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
tls_psk	string	事前共有鍵（PSK）；32 桁以上の 16 進数である必要があります。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
active_available	integer	ホストのアクティブインターフェースの可用性ステータス。 可能な値: 0 - インターフェースのステータスは不明; 1 - インターフェースは利用可能; 2 - インターフェースは利用不可。 プロパティの動作: - 読み取り専用
assigned_proxyid	ID	ホストがプロキシグループによって監視されている場合に、Zabbix サーバーによって割り当てられたプロキシの ID。 プロパティの動作: - 読み取り専用

ホストインベントリ

ホストインベントリオブジェクトには以下のプロパティがあります。

Note:
各プロパティには固有の ID 番号があり、ホストインベントリフィールドとアイテムを関連付けるために使用されます。

ID	プロパティ	型	説明	最大長
4	alias	string	エイリアス。	128 文字
11	asset_tag	string	資産タグ。	64 文字
28	chassis	string	シャーシ。	64 文字
23	contact	string	連絡先。	SQL データベースの場合 65535 文字
32	contract_number	string	契約番号。	64 文字
47	date_hw_decomm	string	ハードウェア廃棄日。	64 文字
46	date_hw_expiry	string	ハードウェア保守期限。	64 文字
45	date_hw_install	string	ハードウェア設置日。	64 文字
44	date_hw_purchase	string	ハードウェア購入日。	64 文字
34	deployment_status	string	展開状況。	64 文字
14	hardware	string	ハードウェア。	255 文字
15	hardware_full	string	詳細なハードウェア。	SQL データベースの場合 65535 文字
39	host_netmask	string	ホストサブネットマスク。	39 文字
38	host_networks	string	ホストネットワーク。	SQL データベースの場合 65535 文字
40	host_router	string	ホストルーター。	39 文字
30	hw_arch	string	ハードウェアアーキテクチャ。	32 文字
33	installer_name	string	インストーラー名。	64 文字
24	location	string	場所。	SQL データベースの場合 65535 文字
25	location_lat	string	場所の緯度。	16 文字
26	location_lon	string	場所の経度。	16 文字
12	macaddress_a	string	MAC アドレス A。	64 文字
13	macaddress_b	string	MAC アドレス B。	64 文字
29	model	string	モデル。	64 文字
3	name	string	名前。	128 文字
27	notes	string	メモ。	SQL データベースの場合 65535 文字
41	oob_ip	string	OOB IP アドレス。	39 文字
42	oob_netmask	string	OOB ホストサブネットマスク。	39 文字
43	oob_router	string	OOB ルーター。	39 文字
5	os	string	OS 名。	128 文字
6	os_full	string	詳細な OS 名。	255 文字
7	os_short	string	短縮 OS 名。	128 文字
61	poc_1_cell	string	一次 POC 携帯番号。	64 文字
58	poc_1_email	string	一次メールアドレス。	128 文字
57	poc_1_name	string	一次 POC 名。	128 文字
63	poc_1_notes	string	一次 POC メモ。	SQL データベースの場合 65535 文字
59	poc_1_phone_a	string	一次 POC 電話 A。	64 文字
60	poc_1_phone_b	string	一次 POC 電話 B。	64 文字
62	poc_1_screen	string	一次 POC スクリーン名。	64 文字
68	poc_2_cell	string	二次 POC 携帯番号。	64 文字
65	poc_2_email	string	二次 POC メールアドレス。	128 文字
64	poc_2_name	string	二次 POC 名。	128 文字
70	poc_2_notes	string	二次 POC メモ。	SQL データベースの場合 65535 文字
66	poc_2_phone_a	string	二次 POC 電話 A。	64 文字
67	poc_2_phone_b	string	二次 POC 電話 B。	64 文字
69	poc_2_screen	string	二次 POC スクリーン名。	64 文字
8	serialno_a	string	シリアル番号 A。	64 文字
9	serialno_b	string	シリアル番号 B。	64 文字
48	site_address_a	string	サイト住所 A。	128 文字
49	site_address_b	string	サイト住所 B。	128 文字
50	site_address_c	string	サイト住所 C。	128 文字
51	site_city	string	サイト都市。	128 文字
53	site_country	string	サイト国。	64 文字
56	site_notes	string	サイトメモ。	SQL データベースの場合 65535 文字
55	site_rack	string	サイトラック位置。	128 文字

ID	プロパティ	型	説明	最大長
52	site_state	string	サイト都道府県。	64 文字
54	site_zip	string	サイト郵便番号。	64 文字
16	software	string	ソフトウェア。	255 文字
18	software_app_a	string	ソフトウェアアプリケーション A。	64 文字
19	software_app_b	string	ソフトウェアアプリケーション B。	64 文字
20	software_app_c	string	ソフトウェアアプリケーション C。	64 文字
21	software_app_d	string	ソフトウェアアプリケーション D。	64 文字
22	software_app_e	string	ソフトウェアアプリケーション E。	64 文字
17	software_full	string	ソフトウェア詳細。	SQL データベースの場合 65535 文字
10	tag	string	タグ。	64 文字
1	type	string	タイプ。	64 文字
2	type_full	string	タイプ詳細。	64 文字
35	url_a	string	URL A。	2048 文字
36	url_b	string	URL B。	2048 文字
37	url_c	string	URL C。	2048 文字
31	vendor	string	ベンダー。	64 文字

ホストタグ

ホストタグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	ホストタグ名。 プロパティの動作: - 必須
value	string	ホストタグの値。
automatic	integer	ホストタグのタイプ。 指定可能な値: 0 - *(デフォルト)* 手動（ユーザーによって作成されたタグ）; 1 - 自動（ローレベルディスカバリによって作成されたタグ） プロパティの動作: - 読み取り専用

host.create

説明

`object host.create(object/array hosts)`

このメソッドは新しいホストを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するホスト。

標準のホストプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	ホストを追加するホストグループ。 ホストグループには groupid プロパティのみが定義されている必要があります。 パラメータの動作: - 必須
interfaces	object/array	ホストに作成するインターフェース。
tags	object/array	ホストタグ。
templates	object/array	ホストにリンクするテンプレート。 テンプレートには templateid プロパティのみが定義されている必要があります。
macros	object/array	ホストに作成するユーザーマクロ。
inventory	object	ホストインベントリのプロパティ。

戻り値

(object) hostids プロパティの下に生成されたホストの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストの順序と一致します。

例

ホストの作成

IP インターフェースとタグを持つ「Linux server」というホストを作成し、グループに追加し、テンプレートをリンクし、ホストインベントリに MAC アドレスを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
```

```
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

SNMP インターフェースを持つホストの作成

詳細付きの SNMPv3 インターフェースを持つ”SNMP host” というホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "SNMP host",
        "interfaces": [
            {
                "type": 2,
                "main": 1,
                "useip": 1,
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": 3,
                    "bulk": 0,
                    "securityname": "mysecurityname",
                    "contextname": "",
                    "securitylevel": 1
                }
            }
        ],
        "groups": [
            {
                "groupid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

PSK 暗号化でホストを作成する

PSK 暗号化を設定した「PSK host」というホストを作成します。ホストはPSK を使用するように事前に設定されている必要があることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

プロキシで監視するホスト情報の作成

ID ”1” のプロキシで監視するホスト情報を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
```

```
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

プロキシグループで監視するホストの作成

ID "1" のプロキシグループで監視するホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

```
{
"jsonrpc": "2.0",
"method": "host.create",
"params": {
"host": " プロキシグループで監視するホスト",
"groups": [
{
```

```
"groupid": "2"
}
],
"monitored_by": 2,
"proxy_groupid": 1
},
"id": 1
}
```

レスポンス:

```
{
"jsonrpc": "2.0",
"result": {
"hostids": [
"10592"
]
},
"id": 1
}
```

参照

- Host group
- Template
- User macro
- Host interface
- Host inventory
- Host tag
- Proxy
- Proxy group

ソース

CHost::create() in *ui/include/classes/api/services/CHost.php*.

host.delete

説明

`object host.delete(array hosts)`

このメソッドは、ホストを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストの ID を含むオブジェクトを返します。

例

複数のホストの削除

2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
```

```
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CHost::delete() in *ui/include/classes/api/services/CHost.php*.

host.get

説明

`integer/array host.get(object parameters)`

このメソッドは、与えられたパラメータにしたがってホストを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
groupids	ID/array	指定したグループに属するホストのみを返します。
dserviceids	ID/array	指定した検出サービスに関連するホストのみを返します。
graphids	ID/array	指定したグラフを持つホストのみを返します。
hostids	ID/array	指定したホスト ID を持つホストのみを返します。
httptestids	ID/array	指定した Web チェックを持つホストのみを返します。
interfaceids	ID/array	指定したインターフェースを使用するホストのみを返します。
itemids	ID/array	指定したアイテムを持つホストのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストのみを返します。
monitored_hosts	flag	監視対象のホストのみを返します。
proxyids	ID/array	指定したプロキシによって監視されているホストのみを返します。
proxy_groupids	ID/array	指定したプロキシグループによって監視されているホストのみを返します。
templated_hosts	flag	ホストとテンプレートの両方を返します。
templateids	ID/array	指定したテンプレートにリンクされているホストのみを返します。
triggerids	ID/array	指定したトリガーを持つホストのみを返します。
with_items	flag	アイテムを持つホストのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメーターを上書きします。
with_item_prototypes	flag	アイテムプロトタイプを持つホストのみを返します。 `with_simple_graph_item_prototypes` パラメーターを上書きします。

パラメーター	タイプ	説明
with_simple_graph_item_prototypes	flag	作成が有効で、数値型の情報タイプを持つアイテムプロトタイプを持つホストのみを返します。
with_graphs	flag	グラフを持つホストのみを返します。
with_graph_prototypes	flag	グラフプロトタイプを持つホストのみを返します。
with_httptests	flag	Web チェックを持つホストのみを返します。 `with_monitored_httptests` パラメーターを上書きします。
with_monitored_httptests	flag	有効な Web チェックを持つホストのみを返します。
with_monitored_items	flag	有効なアイテムを持つホストのみを返します。 `with_simple_graph_items` パラメーターを上書きします。
with_monitored_triggers	flag	有効なトリガーを持つホストのみを返します。トリガーで使用されているすべてのアイテムも有効でなければなりません。
with_simple_graph_items	flag	数値型の情報タイプを持つアイテムを持つホストのみを返します。
with_triggers	flag	トリガーを持つホストのみを返します。 `with_monitored_triggers` パラメーターを上書きします。
withProblemsSuppressed	boolean	抑制された問題を持つホストを返します。 可能な値: `null` - *(デフォルト)* すべてのホスト; `true` - 抑制された問題を持つホストのみ; `false` - 抑制されていない問題を持つホストのみ。
evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
severities	integer/array	指定した深刻度の問題のみを持つホストを返します。問題オブジェクトがトリガーの場合のみ適用されます。
tags	object/array	指定したタグを持つホストのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのホストを返します。 可能なoperator値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
inheritedTags	boolean	すべてのリンクされたテンプレートにも指定した `tags` があるホストを返します。 可能な値: `true` - リンクされたテンプレートにも指定した `tags` が必要; `false` - *(デフォルト)* リンクされたテンプレートのタグは無視されます。

パラメーター	タイプ	説明
selectDiscoveryData	query	ホスト検出オブジェクトデータを持つ discoveryData プロパティを返します。ホスト検出オブジェクトは、検出されたホストを、そのホストが検出されたホストプロトタイプにリンクします。 以下のプロパティがあります: host - (string) ホストの ID; parent_hostid - (string) ホストが作成されたホストプロトタイプの ID; status - (int) ホスト検出ステータス: 0 - *(デフォルト)* ホストは検出されている; 1 - ホストはもはや検出されていない; ts_delete - (timestamp) もはや検出されていないホストが削除される時刻; ts_disable - (timestamp) もはや検出されていないホストが無効化される時刻; disable_source - (int) ホストが LLD ルールまたは手動で無効化されたかどうかの指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	ホストを作成した低レベル検出ルール（VMware 監視のホストプロトタイプから）のdiscoveryRuleプロパティを返します。
selectDiscoveryRules	query	ホストの LLD ルールを持つ discoveryRules プロパティを返します。 count をサポートします。
selectGraphs	query	ホストグラフを持つgraphsプロパティを返します。 count をサポートします。
selectHostGroups	query	ホストが属するホストグループデータを持つhostgroupsプロパティを返します。
selectHttpTests	query	ホスト Web シナリオを持つhttpTestsプロパティを返します。 count をサポートします。
selectInterfaces	query	ホストインターフェースを持つinterfacesプロパティを返します。 count をサポートします。
selectInventory	query	ホストインベントリデータを持つinventoryプロパティを返します。
selectItems	query	ホストアイテムを持つitemsプロパティを返します。 count をサポートします。
selectMacros	query	ホストマクロを持つmacrosプロパティを返します。
selectParentTemplates	query	ホストがリンクされているテンプレートを持つ parentTemplates プロパティを返します。 テンプレートオブジェクトフィールドに加えて、link_type - (integer) テンプレートがホストにリンクされている方法を含みます。 可能な値: 0 - *(デフォルト)* 手動でリンク; 1 - LLD によって自動的にリンク。 count をサポートします。
selectDashboards	query	dashboardsプロパティを返します。 count をサポートします。
selectTags	query	ホストタグを持つtagsプロパティを返します。
selectInheritedTags	query	ホストにリンクされているすべてのテンプレートにあるタグを持つinheritedTagsプロパティを返します。
selectTriggers	query	ホストトリガーを持つtriggersプロパティを返します。 count をサポートします。
selectValueMaps	query	ホスト値マップを持つvaluemapsプロパティを返します。

パラメーター	タイプ	説明
filter	object	指定したフィルターと完全に一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる値または値の配列です。 textデータ型のプロパティはサポートされません。
limitSelects	integer	追加プロパティをサポートします: ホストインターフェースプロパティ。 サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectParentTemplates - 結果は host でソートされます; selectInterfaces; selectItems - name でソート; selectDiscoveryRules - name でソート; selectTriggers - description でソート; selectGraphs - name でソート; selectDashboards - name でソート。
search	object	指定したパターンに一致する結果を返します（大文字小文字を区別しません）。 オブジェクトを受け入れ、キーはプロパティ名、値は検索する文字列です。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 string および textデータ型のプロパティのみをサポートします。
searchInventory	object	追加プロパティをサポートします: ホストインターフェースプロパティ。 指定したパターンに一致するインベントリデータを持つホストを返します（大文字小文字を区別しません）。 オブジェクトを受け入れ、キーはプロパティ名、値は検索する文字列です。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 string および textデータ型のプロパティのみをサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: hostid, host, name, status.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	ホスト低レベル検出ルールを持つdiscoveriesプロパティを返します。 count をサポートします。
selectHostDiscovery	query	このクエリは非推奨です。代わりに selectDiscoveryRules を使用してください。 ホスト検出オブジェクトデータを持つ hostDiscovery プロパティを返します。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix server" と "Linux server" という名前の 2 つのホストに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "proxyid": "0",
            "host": "Zabbix server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Zabbix server",
            "flags": "0",
            "description": "The Zabbix monitoring server.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
```

```
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```

ホストグループの取得

ホスト”Zabbix server” が所属するホストグループ取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "hostgroups": [
                {
                    "groupid": "2",
                    "name": "Linux servers",
                    "flags": "0",
                    "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
                },
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ]
```

```
        }
    ],
    "id": 1
}
```

リンク先テンプレートの取得

ホスト"10084" にリンクしているテンプレートの ID および名前を取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートでホストの取得

テンプレート"10001"(*Linux by Zabbix agent*) がリンクされているホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

ホストインベントリのデータで検索

ホストインベントリの”OS” フィールドに”Linux” が含まれるホストを検索します

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
```

```
}
```

ホストタグで検索

タグ”host-name” が”linux-server” と等しいホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Retrieve hosts that have these tags not only on host level but also in their linked parent templates.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
```

```
        "inheritedTags": true
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

タグとテンプレートタグでホストの検索

タグと親テンプレートにリンクされているすべてのタグでホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10502",
            "name": "Desktop",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ],
            "inheritedTags": [
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
```

```
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ]
        }
    ],
    "id": 1
}
```

障害の深刻度でホストの検索

”Disaster” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

”Average” と”High” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
```

```
    ],
    "id": 1
}
```

参照

- Host group
- Template
- User macro
- Host interface
- Proxy
- Proxy group

ソース

CHost::get() in *ui/include/classes/api/services/CHost.php*.

host.massadd

説明

`object host.massadd(object parameters)`

このメソッドは、指定されたすべてのホストに複数の関連オブジェクトを同時に追加することができます。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するホストの ID と、すべてのホストに追加するオブジェクトを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
hosts	object/array	更新するホスト。 ホストには `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	指定したホストに追加するホストグループ。 ホストグループには `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	指定したホストに作成するホストインターフェース。
macros	object/array	指定したホストに作成するユーザーマクロ。
templates	object/array	指定したホストにリンクするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(object) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

マクロの追加

2 つのホストに 2 つの新しいマクロを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Test description"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

参照

- host.update
- Host group
- Template
- User macro
- Host interface

ソース

CHost::massAdd() in *ui/include/classes/api/services/CHost.php*.

host.massremove

説明

`object host.massremove(object parameters)`

このメソッドは、複数のホストから関連するオブジェクトを削除することができます。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するホストの ID と削除するオブジェクトを含むパラメータ。

パラメータ	型	説明
hostids	ID/array	更新するホストの ID。 パラメータの動作: - 必須
groupids	ID/array	指定したホストを削除するホストグループの ID。
interfaces	object/array	指定したホストから削除するホストインターフェース。 ホストインターフェースオブジェクトには、ip、dns、port プロパティのみを定義する必要があります。
macros	string/array	指定したホストから削除するユーザーマクロ。
templateids	ID/array	指定したホストからリンクを解除するテンプレートの ID。
templateids_clear	ID/array	指定したホストからリンクを解除し、クリアするテンプレートの ID。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

2 つのホストからテンプレートのリンクを解除し、テンプレート化されたすべてのエンティティを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

参照

- host.update
- User macro
- Host interface

ソース

CHost::massRemove() in *ui/include/classes/api/services/CHost.php*.

host.massupdate

説明

```
object host.massupdate(object parameters)
```

このメソッドは、複数のホストの関連オブジェクトを同時に置換または削除し、プロパティを更新することができます。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するホストの ID と更新するプロパティを含むパラメータ。

標準のホストプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
hosts	object/array	更新するホスト。 ホストには `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	ホストが属する現在のホストグループを置き換えるホストグループ。 ホストグループには `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	指定したホストの現在のホストインターフェースを置き換えるホストインターフェース。
inventory	object	ホストインベントリのプロパティ。 ホストインベントリモードは `inventory` パラメータでは更新できません。代わりに `inventory_mode` を使用してください。
macros	object/array	指定したホストの現在のユーザーマクロを置き換えるユーザーマクロ。
templates	object/array	指定したホストに現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	指定したホストからリンク解除およびクリアするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(object) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

複数ホストの有効化

2 つのホストの監視を有効にします。つまり、status を”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massupdate",
    "params": {
        "hosts": [
            {
                "hostid": "69665"
            },
            {
                "hostid": "69666"
            }
        ],
        "status": 0
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

参照

- host.update
- host.massadd
- host.massremove
- Host group
- Template
- User macro
- Host interface

ソース

CHost::massUpdate() in *ui/include/classes/api/services/CHost.php*.

host.update

説明

`object host.update(object/array hosts)`

このメソッドは、既存のホストを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するホストのプロパティ。

各ホストには `hostid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。指定されたプロパティのみが更新され、他のプロパティは変更されません。

ただし、ホストの技術名を更新すると、ホストの表示名（指定されていない場合や空の場合）もホストの技術名の値で更新されます。

標準のホストプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	現在のホストグループを置き換えるホストグループ。 リクエストに記載されていないすべてのホストグループはリンク解除されます。 ホストグループには `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	現在のホストインターフェースを置き換えるホストインターフェース。 リクエストに記載されていないすべてのインターフェースは削除されます。
tags	object/array	現在のホストタグを置き換えるホストタグ。 リクエストに記載されていないすべてのタグは削除されます。
inventory	object	ホストインベントリのプロパティ。

パラメータ	型	説明
macros	object/array	現在のユーザーマクロを置き換えるユーザーマクロ。 リクエストに記載されていないすべてのマクロは削除されます。
templates	object/array	現在リンクされているテンプレートを置き換えるテンプレート。 リクエストに記載されていないすべてのテンプレートはリンク解除のみされます。 テンプレートには templateid プロパティのみを定義する必要があります。
templates_clear	object/array	ホストからリンク解除およびクリアするテンプレート。 テンプレートには templateid プロパティのみを定義する必要があります。

Note:
Zabbix フロントエンドとは異なり、name（表示ホスト名）が host（技術ホスト名）と同じ場合、API 経由で host を更新しても name は自動的に更新されません。両方のプロパティを明示的に更新する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

ホストの有効化

ホストの監視を有効にする、つまりステータスを 0 にします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

テンプレートのリンク削除

ホストから 2 つのテンプレートのリンクを削除してクリアします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
```

```
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストマクロの更新

すべてのホストマクロを新しい 2 つのマクロに置き換えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Updated description"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストインベントリの更新

インベントリモードの変更とロケーションの追加をします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
```

```
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホストタグの更新

すべてのホストタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

検出されたホストマクロの更新

作成した検出ルールの” 自動” マクロを” 手動” に変換し、その値を”new-value” に変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
```

```
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホスト暗号化の更新

ホスト”10590” を更新して、ホストから Zabbix サーバーへの接続にのみ PSK 暗号化を使用するようにし、PSK ID と PSK キーを変更します。

ホストはPSK を使用するように事前設定する必要があることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10590",
        "tls_connect": 1,
        "tls_accept": 2,
        "tls_psk_identity": "PSK 002",
        "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

参照

- host.massadd
- host.massupdate
- host.massremove
- Host group
- Template
- User macro
- Host interface
- Host inventory
- Host tag
- Proxy
- Proxy group

ソース

CHost::update() in *ui/include/classes/api/services/CHost.php*.

Host group

このクラスは、ホストグループを操作するためのものです。

オブジェクトリファレンス:

- Host group

利用可能なメソッド:

- hostgroup.create - 新しいホストグループを作成する
- hostgroup.delete - ホストグループを削除する
- hostgroup.get - ホストグループを取得する
- hostgroup.massadd - ホストグループへ関連オブジェクトを追加する
- hostgroup.massremove - ホストグループから関連オブジェクトを削除する
- hostgroup.massupdate - ホストグループの関連するオブジェクトを交換または削除する
- hostgroup.propagate - ホストグループのサブグループへ権限とタグフィルターを伝達する
- hostgroup.update - ホストグループを更新する

Host group オブジェクト

次のオブジェクトは、hostgroupAPI に直接関連しています。

ホストグループ

ホストグループオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
groupid	ID	ホストグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ホストグループの名前。 プロパティの動作: - 作成操作では必須
flags	integer	ホストグループの起源。 可能な値: 0 - 通常のホストグループ; 4 - 検出されたホストグループ。 プロパティの動作: - 読み取り専用
uuid	string	インポートされたホストグループを既存のものにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。

hostgroup.create

説明

`object hostgroup.create(object/array hostGroups)`

このメソッドは新しいホストグループを作成します。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するホストグループ。

このメソッドは、標準のホストグループプロパティを持つホストグループを受け付けます。

戻り値

(object) groupids プロパティの下に作成されたホストグループの ID を含むオブジェクトを返します。
 返される ID の順序は、渡されたホストグループの順序と一致します。

例

ホストグループの作成

”Linux servers” というホストグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::create() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

説明

`object hostgroup.delete(array hostGroupIds)`

このメソッドは、ホストグループを削除することができます。

以下の場合、ホストグループを削除することはできません。

- このグループのみに属するホストが含まれている。
- 内部としてマークされている。
- ホストプロトタイプで使用されている。
- グローバルスクリプトで使用されている。
- 相関関係で使用されている。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストグループの ID。

戻り値

(object) groupids プロパティの下にある削除されたホストグループの ID を含むオブジェクトを返します。

例

複数のホストグループの削除

2 つのホストグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

説明

`integer/array hostgroup.get(object parameters)`

このメソッドは、指定されたパラメータに従ってホストグループを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
graphids	ID/array	指定したグラフを持つホストを含むホストグループのみを返します。
groupids	ID/array	指定したホストグループ ID を持つホストグループのみを返します。
hostids	ID/array	指定したホストを含むホストグループのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストグループのみを返します。
triggerids	ID/array	指定したトリガーを持つホストを含むホストグループのみを返します。
with_graphs	flag	グラフを持つホストを含むホストグループのみを返します。
with_graph_prototypes	flag	グラフプロトタイプを持つホストを含むホストグループのみを返します。
with_hosts	flag	ホストを含むホストグループのみを返します。
with_httptests	flag	Web チェックを持つホストを含むホストグループのみを返します。 `with_monitored_httptests` パラメーターを上書きします。
with_items	flag	アイテムを持つホストを含むホストグループのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメーターを上書きします。

パラメーター	型	説明
with_item_prototypes	flag	アイテムプロトタイプを持つホストを含むホストグループのみを返します。 `with_simple_graph_item_prototypes` パラメーターを上書きします。
with_simple_graph_item_prototypes	flag	作成が有効で数値型の情報を持つアイテムプロトタイプを持つホストを含むホストグループのみを返します。
with_monitored_httptests	flag	有効な Web チェックを持つホストを含むホストグループのみを返します。
with_monitored_hosts	flag	監視対象ホストを含むホストグループのみを返します。
with_monitored_items	flag	有効なアイテムを持つホストを含むホストグループのみを返します。 `with_simple_graph_items` パラメーターを上書きします。
with_monitored_triggers	flag	有効なトリガーを持つホストを含むホストグループのみを返します。トリガーで使用されているすべてのアイテムも有効でなければなりません。
with_simple_graph_items	flag	数値アイテムを持つホストを含むホストグループのみを返します。
with_triggers	flag	トリガーを持つホストを含むホストグループのみを返します。 `with_monitored_triggers` パラメーターを上書きします。
selectDiscoveryRules	query	ホストグループを検出した LLD ルールを持つ`discoveryRules`プロパティを返します。
selectDiscoveryData	query	ホストグループディスカバリオブジェクトを持つ `discoveryData` プロパティを返します。 各ホストグループディスカバリオブジェクトは、検出されたホストグループにリンクされたホストグループプロトタイプであり、以下のプロパティを持ちます: `parent_group_prototypeid` - (ID) ホストグループが検出されたホストグループプロトタイプの ID; `name` - (string) ホストグループプロトタイプの名前; `ts_delete` - (timestamp) もはや検出されなくなったホストグループが削除される時刻; `status` - (int) ホストグループディスカバリのステータス: 0 - *(デフォルト)* ホストグループは検出されています; 1 - ホストグループはもはや検出されていません。
selectHostPrototypes	query	このホストグループを検出したホストプロトタイプを持つ`hostPrototypes`プロパティを返します。
selectHosts	query	ホストグループに属するホストを持つ`hosts`プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `host` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `groupid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

パラメーター	型	説明
selectGroupDiscoveries	query	ホストグループディスカバリオブジェクトを持つ `groupDiscoveries` プロパティを返します。 各ホストグループディスカバリオブジェクトは、検出されたホストグループにリンクされたホストグループプロトタイプです。 このクエリは非推奨です。代わりに `selectDiscoveryData` を使用してください。

戻り値

(`integer/array`) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix servers" と"Linux servers" という 2 つのホストグループに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "internal": "0"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "internal": "0"
        }
    ],
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::get() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

説明

`object hostgroup.massadd(object parameters)`

このメソッドは、複数の関連オブジェクトをすべての指定されたホストグループに同時に追加することを可能にします。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 更新するホストグループの ID と、すべてのホストグループに追加するオブジェクトを含むパラメーター。

このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
groups	object/array	更新するホストグループ。 ホストグループには `groupid` プロパティのみが定義されている必要があります。 パラメーターの動作: - 必須
hosts	object/array	すべてのホストグループに追加するホスト。 ホストには `hostid` プロパティのみが定義されている必要があります。

戻り値

(object) `groupids` プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストをホストグループに追加する

2 つのホストを ID が 5 と 6 のホストグループに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

説明

`object hostgroup.massremove(object parameters)`

このメソッドは、複数のホストグループから関連するオブジェクトを削除するためのものです。

Note:
このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するホストグループの ID と削除するオブジェクトを含むパラメータ。

パラメータ	タイプ	説明
groupids	ID/array	更新するホストグループの ID。 パラメータの動作: - 必須
hostids	ID/array	すべてのホストグループから削除するホストの ID。

戻り値

(object) groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループからホストを削除する

指定したホストグループから 2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massupdate

説明

`object hostgroup.massupdate(object parameters)`

このメソッドは、複数のホストグループ内のホストおよびテンプレートを指定したものに置き換えることができます。

Note:
このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するホストグループの ID と、更新するオブジェクトを含むパラメータ。

パラメータ	型	説明
groups	object/array	更新するホストグループ。 ホストグループには `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
hosts	object/array	指定したホストグループの現在のホストを置き換えるホスト。 指定したホスト以外のすべてのホストはホストグループから除外されます。 検出されたホストには影響しません。 ホストには `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループ内のホストの置換

ホストグループ内のすべてのホストを指定したホストに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

関連項目

- hostgroup.update
- hostgroup.massadd
- Host

ソース

CHostGroup::massUpdate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

説明

`object hostgroup.propagate(object parameters)`

このメソッドは、ホストグループのすべてのサブグループに権限とタグフィルターを適用することができます。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール をご覧ください。

パラメータ

(object) は、必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
groups	object/array	伝播するホストグループ ホストグループには groupid プロパティが定義されている必要があります。 パラメータの動作: - 必須
permissions	boolean	権限を伝播するには、”true” に設定します。 パラメータの動作: - 必須 (tag_filters が設定されていない場合)
tag_filters	boolean	タグフィルターを伝播するには「true」に設定します。 パラメータの動作: - permissions が設定されていない場合は 必須

戻り値

(object) は、groupids プロパティの下で、伝播されたホストグループの ID を含むオブジェクトを返します。

例

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

参照

- hostgroup.update
- hostgroup.massadd
- Host

ソース

CHostGroup::propagate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

説明

`object hostgroup.update(object/array hostGroups)`

このメソッドは、既存のホストグループを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) Host group properties が更新されます。

groupid' プロパティは各ホストグループに対して定義する必要があり、他のプロパティはすべてオプションです。
 与えられたプロパティのみが更新され、その他のプロパティは変更されません。

戻り値

(`object`) groupids プロパティの下で更新された host gtoup の ID を含むオブジェクトを返します。

例

ホストグループの名前の変更

ホストグループの名前を”Linux hosts” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::update() in *ui/include/classes/api/services/CHostGroup.php*.

Host interface

このクラスは、ホストインターフェースを操作するためのものです。

オブジェクトリファレンス:

- Host interface
- Details

利用可能なメソッド:

- hostinterface.create - 新しいホストインターフェースを作成する
- hostinterface.delete - ホストインターフェースを削除する
- hostinterface.get - ホストインターフェースを取得する

- hostinterface.massadd - ホストへホストインターフェースを追加する
- hostinterface.massremove - ホストからホストインターフェースを削除する
- hostinterface.replacehostinterfaces - ホストのホストインターフェースを置換する
- hostinterface.update - ホストインターフェースを更新する

Host interface オブジェクト

次のオブジェクトは、`hostinterface`API に直接関連しています。

ホストインターフェース

ホストインターフェースオブジェクトには以下のプロパティがあります。

Attention:
`ip` と `dns` の両方のプロパティは、作成操作時に必須であることに注意してください。DNS を使用しない場合は、空文字列に設定してください。

プロパティ	型	説明
interfaceid	ID	インターフェースの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
available	integer	ホストインターフェースの可用性。 可能な値： 0 - *(デフォルト)* 不明; 1 - 利用可能; 2 - 利用不可。 プロパティの動作: - 読み取り専用
hostid	ID	インターフェースが属するホストの ID。 プロパティの動作: - 定数 - 作成操作時に必須
type	integer	インターフェースのタイプ。 可能な値： 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX。 プロパティの動作: - 作成操作時に必須
ip	string	インターフェースで使用される IP アドレス。 DNS 経由で接続する場合は空でも可。 プロパティの動作: - 作成操作時に必須
dns	string	インターフェースで使用される DNS 名。 IP 経由で接続する場合は空でも可。 プロパティの動作: - 作成操作時に必須

プロパティ	型	説明
port	string	インターフェースで使用されるポート番号。ユーザーマクロを含めることができます。 プロパティの動作: - 作成操作時に必須
useip	integer	IP 経由で接続するかどうか。 可能な値： 0 - ホストの DNS 名で接続; 1 - ホストの IP アドレスで接続。 プロパティの動作: - 作成操作時に必須
main	integer	インターフェースがホストのデフォルトとして使用されるかどうか。同じタイプのインターフェースはホストごとに 1 つだけデフォルトに設定できます。 可能な値： 0 - デフォルトでない; 1 - デフォルト。 プロパティの動作: - 作成操作時に必須
details	object	インターフェースの追加詳細オブジェクト。 プロパティの動作: - type が”SNMP” に設定されている場合は必須
disable_until	timestamp	利用不可のホストインターフェースの次回ポーリング時刻。 プロパティの動作: - 読み取り専用
error	string	ホストインターフェースが利用不可の場合のエラーテキスト。 プロパティの動作: - 読み取り専用
errors_from	timestamp	ホストインターフェースが利用不可になった時刻。 プロパティの動作: - 読み取り専用

詳細

details オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
version	integer	SNMP インターフェースのバージョン。 可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3。 プロパティの動作: - 必須
bulk	integer	バルク SNMP リクエストを使用するかどうか。 可能な値: 0 - バルクリクエストを使用しない; 1 - (デフォルト) - バルクリクエストを使用する。

プロパティ	タイプ	説明
community	string	SNMP コミュニティ。SNMPv1 および SNMPv2 インターフェースでのみ使用されます。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は必須
max_repetitions	integer	ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDUs）の最大繰り返し値。 SNMPv2 および v3 の `discovery[]` および `walk[]` アイテムでのみ使用されます。 デフォルト: 10。
securityname	string	SNMPv3 セキュリティ名。SNMPv3 インターフェースでのみ使用されます。
securitylevel	integer	SNMPv3 セキュリティレベル。SNMPv3 インターフェースでのみ使用されます。 可能な値: 0 - (デフォルト) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv。
authpassphrase	string	SNMPv3 認証パスフレーズ。SNMPv3 インターフェースでのみ使用されます。
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。SNMPv3 インターフェースでのみ使用されます。
authprotocol	integer	SNMPv3 認証プロトコル。SNMPv3 インターフェースでのみ使用されます。 可能な値: 0 - (デフォルト) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512。
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 可能な値: 0 - (デフォルト) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C。
contextname	string	SNMPv3 コンテキスト名。SNMPv3 インターフェースでのみ使用されます。

hostinterface.create

説明

`object hostinterface.create(object/array hostInterfaces)`

このメソッドは、新しいホストインターフェースを作成するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するホストインターフェース。

このメソッドは、標準のホストインターフェースプロパティを持つホストインターフェースを受け入れます。

戻り値

(object) interfaceids プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストインターフェイスの順序と一致します。

例

新しいインターフェースの作成

ホスト"30052" にセカンダリ IP エージェントインターフェースを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

SNMP の詳細でインターフェースを作成する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```


参照

- hostinterface.massadd
- host.massadd

ソース

CHostInterface::create() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

説明

`object hostinterface.delete(array hostInterfaceIds)`

このメソッドは、ホストインターフェースを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストインターフェースの ID。

戻り値

(object) interfaceids プロパティの下にある削除されたホストインターフェースの ID を含むオブジェクトを返します。

例

ホストインターフェースの削除

ID 30062 のホストインターフェースを削除します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```


参照

- hostinterface.massremove
- host.massremove

ソース

CHostInterface::delete() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

説明

`integer/array hostinterface.get(object parameters)`

このメソッドは、指定されたパラメータに従ってホストインターフェースを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
hostids	ID/array	指定したホストで使用されているホストインターフェースのみを返します。
interfaceids	ID/array	指定した ID のホストインターフェースのみを返します。
itemids	ID/array	指定したアイテムで使用されているホストインターフェースのみを返します。
triggerids	ID/array	指定したトリガーのアイテムで使用されているホストインターフェースのみを返します。
selectItems	query	インターフェースを使用するアイテムを含む`items`プロパティを返します。 `count` をサポートします。
selectHosts	query	インターフェースを使用するホストの配列を含む`hosts`プロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `interfaceid`, `dns`, `ip`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

`(integer/array)` 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストインターフェースの取得

ホスト"30057" で使用されているインターフェースに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "interfaceid": "50039",
            "hostid": "30057",
            "main": "1",
            "type": "1",
            "useip": "0",
            "ip": "",
            "dns": "localhost",
            "port": "10050",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": []
        },
        {
            "interfaceid": "55082",
            "hostid": "30057",
            "main": "1",
            "type": "2",
            "useip": "1",
            "ip": "127.0.0.1",
            "dns": "",
            "port": "161",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": {
                "version": "2",
                "bulk": "0",
                "community": "{$SNMP_COMMUNITY}",
                "max_repetitions": "10"
            }
        }
    ],
    "id": 1
}
```

参照

- Host
- Item

ソース

CHostInterface::get() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massadd

説明

`object hostinterface.massadd(object parameters)`

このメソッドは、複数のホストにホストインターフェースを同時に追加することができます。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 指定されたホストに作成するホストインターフェースを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
interfaces	object/array	指定されたホストに作成するホストインターフェース。 パラメータの動作: - 必須
hosts	object/array	更新するホスト。 ホストには `hostid` プロパティのみが定義されている必要があります。 パラメータの動作: - 必須

戻り値

(object) `interfaceids` プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。

例

インターフェースの作成

2 つのホストにインターフェースを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.create
- host.massadd
- Host

ソース

CHostInterface::massAdd() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

説明

`object hostinterface.massremove(object parameters)`

このメソッドは、指定したホストからホストインターフェースを削除するためのものです。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object`) 更新するホストの ID と削除するインターフェースを含むパラメーター。

パラメーター	タイプ	説明
interfaces	object/array	指定したホストから削除するホストインターフェース。 ホストインターフェースオブジェクトには、`ip`、`dns`、`port` プロパティのみを定義する必要があります。 パラメーターの動作: - 必須
hostids	ID/array	更新するホストの ID。 パラメーターの動作: - 必須

戻り値

(`object`) `interfaceids` プロパティの下で削除されたホストインターフェイスの ID を含むオブジェクトを返します。

例

インターフェースの削除

2 つのホストから”127.0.0.1” の SNMP インターフェースを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.delete
- host.massremove

ソース

CHostInterface::massRemove() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.replacehostinterfaces

説明

`object hostinterface.replacehostinterfaces(object parameters)`

このメソッドは、指定したホストのすべてのホストインターフェースを置き換えることができます。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドを呼び出す権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 更新するホストの ID と新しいホストインターフェースを含むパラメーター。

パラメーター	型	説明
interfaces	object/array	現在のホストインターフェースを置き換えるホストインターフェース。 パラメーターの動作: - 必須
hostid	ID	更新するホストの ID。 パラメーターの動作: - 必須

戻り値

(object) interfaceids プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェースの置換

すべてのホストインターフェースを単一のエージェントインターフェースに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.replacehostinterfaces",
    "params": {
        "hostid": "30052",
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 1,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30081"
        ]
    },
    "id": 1
}
```

参照

- host.update
- host.massupdate

ソース

CHostInterface::replaceHostInterfaces() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

説明

`object hostinterface.update(object/array hostInterfaces)`

このメソッドは、既存のホストインターフェースを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) ホストインターフェースのプロパティが更新されます。

interfaceid プロパティは、ホストインターフェイスごとに定義する必要があり、他のすべてのプロパティはオプションです。指定されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) interfaceids プロパティの下で更新されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェースポートの変更

ホストインターフェースのポートを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "30050"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

ソース

CHostInterface::update() in *ui/include/classes/api/services/CHostInterface.php*.

Host prototype

このクラスは、ホストプロトタイプを操作するためのものです。

オブジェクトリファレンス:

- Host prototype
- Group link
- Group prototype
- Host prototype tag
- Custom interface
    - Custom interface details

利用可能なメソッド:

- hostprototype.create - 新しいホストプロトタイプを作成する
- hostprototype.delete - ホストプロトタイプを削除する
- hostprototype.get - ホストプロトタイプを取得する
- hostprototype.update - ホストプロトタイプを更新する

Host prototype オブジェクト

次のオブジェクトは、hostprototypeAPI に直接関連しています。

ホストプロトタイプ

ホストプロトタイプオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
hostid	ID	ホストプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作時は必須
host	string	ホストプロトタイプの技術名。 プロパティの動作: - 作成操作時は必須 - 継承されたオブジェクトでは読み取り専用
name	string	ホストプロトタイプの表示名。 デフォルト: `host` プロパティの値。 プロパティの動作: - 継承されたオブジェクトでは読み取り専用
status	integer	ホストプロトタイプのステータス。 可能な値: 0 - *(デフォルト)* 監視対象ホスト; 1 - 監視対象外ホスト。
flags	integer	ホストプロトタイプの由来。 可能な値: 2 - ホストプロトタイプ; 6 - 検出されたホストプロトタイプ プロパティの動作: - 読み取り専用
inventory_mode	integer	ホストインベントリの設定モード。 可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
templateid	ID	親テンプレートホストプロトタイプの ID。 プロパティの動作: - 読み取り専用
discover	integer	ホストプロトタイプの検出ステータス。 可能な値: 0 - *(デフォルト)* 新しいホストが検出される; 1 - 新しいホストは検出されず、既存のホストはロストとしてマークされる。
custom_interfaces	integer	ホストプロトタイプによって作成されたホストのカスタムインターフェースのソース。 可能な値: 0 - *(デフォルト)* 親ホストからインターフェースを継承; 1 - ホストプロトタイプのカスタムインターフェースを使用。 プロパティの動作: - 継承されたオブジェクトでは読み取り専用
uuid	string	インポートされたホストプロトタイプを既存のものにリンクするために使用されるユニバーサルユニーク識別子。指定されていない場合は自動生成されます。 プロパティの動作: - ホストプロトタイプがテンプレートに属している場合にサポート

グループリンク

グループリンクオブジェクトは、ホストプロトタイプとホストグループをリンクします。以下のプロパティを持ちます。

プロパティ	タイプ	説明
groupid	ID	ホストグループの ID。 プロパティの動作: - 必須

グループプロトタイプ

グループプロトタイプオブジェクトは、検出されたホストに対して作成されるグループを定義します。以下のプロパティを持ちます。

プロパティ	タイプ	説明
group_prototypeid	ID	グループプロトタイプの ID。 プロパティの動作: - 読み取り専用
name	string	グループプロトタイプの名前。 プロパティの動作: - 作成操作時に必須

ホストプロトタイプタグ

ホストプロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	ホストプロトタイプタグ名。 プロパティの動作: - 必須
value	string	ホストプロトタイプタグの値。

カスタムインターフェース

`custom_interfaces` がホストプロトタイプオブジェクトで「ホストプロトタイプのカスタムインターフェースを使用する」に設定されている場合、カスタムインターフェースがサポートされます。カスタムインターフェースオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	インターフェースタイプ。 可能な値： 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX。 プロパティの動作： - 必須
useip	integer	IP 経由で接続するかどうか。 可能な値： 0 - ホストの DNS 名で接続; 1 - ホストの IP アドレスで接続。 プロパティの動作： - 必須

プロパティ	タイプ	説明
ip	string	インターフェースで使用される IP アドレス。 マクロを含めることができます。 プロパティの動作： - `useip` が「ホストの IP アドレスで接続」に設定されている場合は必須
dns	string	インターフェースで使用される DNS 名。 マクロを含めることができます。 プロパティの動作： - `useip` が「ホストの DNS 名で接続」に設定されている場合は必須
port	string	インターフェースで使用されるポート番号。 ユーザーマクロおよび LLD マクロを含めることができます。 プロパティの動作： - 必須
main	integer	インターフェースがホストのデフォルトとして使用されるかどうか。 あるタイプのインターフェースはホストごとに 1 つだけデフォルトに設定できます。 可能な値： 0 - デフォルトでない; 1 - デフォルト。 プロパティの動作： - 必須
details	object	カスタムインターフェースの詳細の追加オブジェクト。 プロパティの動作： - `type` が「SNMP」に設定されている場合は必須

カスタムインターフェースの詳細

詳細オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
version	integer	SNMP インターフェースのバージョン。 可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3。 プロパティの動作: - 必須
bulk	integer	バルク SNMP リクエストを使用するかどうか。 可能な値: 0 - バルクリクエストを使用しない; 1 - *(デフォルト)* - バルクリクエストを使用する。
community	string	SNMP コミュニティ。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は必須
max_repetitions	integer	ネイティブ SNMP バルクリクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 SNMPv2 および v3 の `discovery[]` および `walk[]` アイテムでのみ使用されます。 デフォルト: 10。

プロパティ	タイプ	説明
securityname	string	SNMPv3 セキュリティ名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合はサポート
securitylevel	integer	SNMPv3 セキュリティレベル。 可能な値: 0 - *(デフォルト)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合はサポート
authpassphrase	string	SNMPv3 認証パスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合はサポート
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合はサポート
authprotocol	integer	SNMPv3 認証プロトコル。 可能な値: 0 - *(デフォルト)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合はサポート
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 可能な値: 0 - *(デフォルト)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合はサポート
contextname	string	SNMPv3 コンテキスト名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合はサポート

hostprototype.create

説明

`object hostprototype.create(object/array hostPrototypes)`

このメソッドは新しいホストプロトタイプを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 作成するホストプロトタイプ。

標準のホストプロトタイププロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
groupLinks	array	ホストプロトタイプに作成するグループリンク。 パラメーターの動作: - 必須
ruleid	ID	ホストプロトタイプが属する LLD ルールの ID。 パラメーターの動作: - 必須
groupPrototypes	array	ホストプロトタイプに作成するグループプロトタイプ。
macros	object/array	ホストプロトタイプに作成するユーザーマクロ。
tags	object/array	ホストプロトタイプタグ。
interfaces	object/array	ホストプロトタイプのカスタムインターフェース。
templates	object/array	ホストプロトタイプにリンクするテンプレート。 テンプレートには templateid プロパティのみが定義されている必要があります。

戻り値

(object) hostids プロパティの下で作成されたホストプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストプロトタイプの順序と一致します。

例

ホストプロトタイプの作成

グループプロトタイプ"{#HV.NAME}"、タグペア"datacenter": "{#DATACENTER.NAME}"、カスタム SNMPv2 インターフェース 127.0.0.1:161（コミュニティ {$SNMP_COMMUNITY}）を持つ LLD ルール"23542" 上にホストプロトタイプ"{#VM.NAME}" を作成します。ホストグループ"2" にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.create",
    "params": {
        "host": "{#VM.NAME}",
        "ruleid": "23542",
        "custom_interfaces": "1",
        "groupLinks": [
            {
                "groupid": "2"
            }
        ],
        "groupPrototypes": [
            {
                "name": "{#HV.NAME}"
            }
        ],
        "tags": [
            {
                "tag": "datacenter",
```

```
                "value": "{#DATACENTER.NAME}"
            }
        ],
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

参照

- Group link
- Group prototype
- Host prototype tag
- Custom interface
- User macro

ソース

CHostPrototype::create() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

説明

`object hostprototype.delete(array hostPrototypeIds)`

このメソッドは、ホストのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストのプロトタイプの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストのプロトタイプの ID を含むオブジェクトを返します。

例

複数のホストのプロトタイプの削除

2 つのホストのプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

ソース

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

説明

`integer/array hostprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従ってホストプロトタイプを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
hostids	ID/array	指定した ID のホストプロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するホストプロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムのみを返します。

パラメーター	型	説明
selectDiscoveryData	query	ホストプロトタイプディスカバリーオブジェクトデータを持つdiscoveryDataプロパティを返します。ホストプロトタイプディスカバリーオブジェクトは、検出されたホストプロトタイプを、それが検出されたホストプロトタイプにリンクします。 以下のプロパティを持ちます: host - (string) ホストのID; parent_hostid - (string) ホストプロトタイプが作成された元のホストプロトタイプのID; status - (int) ホストプロトタイプディスカバリーステータス: 0 - *(デフォルト)* ホストプロトタイプが検出されている; 1 - ホストプロトタイプがもはや検出されていない; ts_delete - (timestamp) もはや検出されていないホストプロトタイプが削除される時刻; ts_disable - (timestamp) もはや検出されていないホストプロトタイプが無効化される時刻; disable_source - (int) ホストプロトタイプがLLDルールまたは手動で無効化されたかどうかの指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLDルールによって無効化。
selectDiscoveryRule	query	ホストプロトタイプが属するLLDルールを持つdiscoveryRuleプロパティを返します。
selectDiscoveryRulePrototype	query	ホストプロトタイプが属する親LLDルールプロトタイプを持つdiscoveryRulePrototypeプロパティを返します。
selectInterfaces	query	ホストプロトタイプのカスタムインターフェースを持つinterfacesプロパティを返します。
selectGroupLinks	query	ホストプロトタイプのグループリンクを持つgroupLinksプロパティを返します。
selectGroupPrototypes	query	ホストプロトタイプのグループプロトタイプを持つgroupPrototypesプロパティを返します。
selectMacros	query	ホストプロトタイプのマクロを持つmacrosプロパティを返します。
selectParentHost	query	ホストプロトタイプが属するホストを持つparentHostプロパティを返します。
selectTags	query	ホストプロトタイプのタグを持つtagsプロパティを返します。
selectTemplates	query	ホストプロトタイプにリンクされたテンプレートを持つtemplatesプロパティを返します。 countをサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: hostid, host, name, status, discovered。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutputパラメーターが使用されている場合、取得したオブジェクトの数

例

LLD ルールからホストプロトタイプを取得する

LLD ルールから、すべてのホストプロトタイプ、そのグループリンク、グループプロトタイプ、およびタグを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "flags": "2",
            "discover": "0",
            "custom_interfaces": "1",
            "inventory_mode": "-1",
            "groupLinks": [
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
```

```
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```

参照

- Group link
- Group prototype
- User macro

ソース

CHostPrototype::get() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

説明

`object hostprototype.update(object/array hostPrototypes)`

このメソッドは、既存のホストプロトタイプを更新します。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するホストプロトタイプのプロパティ。

各ホストプロトタイプには `hostid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のホストプロトタイププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groupLinks	array	ホストプロトタイプの現在のグループリンクを置き換えるグループリンク。
groupPrototypes	array	パラメータの動作: - 継承オブジェクトの場合は読み取り専用 ホストプロトタイプの既存のグループプロトタイプを置き換えるグループプロトタイプ。 リクエストに記載されていないすべてのグループプロトタイプは削除されます。
macros	object/array	パラメータの動作: - 継承オブジェクトの場合は読み取り専用 現在のユーザーマクロを置き換えるユーザーマクロ。 リクエストに記載されていないすべてのマクロは削除されます。

パラメータ	型	説明
tags	object/array	現在のタグを置き換えるホストプロトタイプタグ。 リクエストに記載されていないすべてのタグは削除されます。 パラメータの動作: - 継承オブジェクトの場合は読み取り専用
interfaces	object/array	現在のインターフェースを置き換えるホストプロトタイプのカスタムインターフェース。 カスタムインターフェースオブジェクトにはすべてのパラメータを含める必要があります。 リクエストに記載されていないすべてのインターフェースは削除されます。 パラメータの動作: - ホストプロトタイプオブジェクトの `custom_interfaces` が「ホストプロトタイプのカスタムインターフェースを使用する」に設定されている場合はサポート - 継承オブジェクトの場合は読み取り専用
templates	object/array	現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(object) `hostids` プロパティの下で更新されたホストプロトタイプの ID を含むオブジェクトを返します。

例

ホストプロトタイプの無効化

ホストプロトタイプを無効にします。つまり、ステータスを"1" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプタグの更新

ホストプロトタイプタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
```

```
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプのカスタムインターフェースの更新

継承されたインターフェースをホストプロトタイプのカスタムインターフェースに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "custom_interfaces": "1",
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
```

```
    },
    "id": 1
}
```

参照

- Group link
- Group prototype
- Host prototype tag
- Custom interface
- User macro

ソース

CHostPrototype::update() in *ui/include/classes/api/services/CHostPrototype.php*.

Housekeeping

このクラスは、ハウスキーピングを操作するためのものです。

オブジェクトリファレンス:

- Housekeeping

利用可能なメソッド:

- housekeeping.get - ハウスキーピングの取得
- housekeeping.update - ハウスキーピングの更新

Housekeeping オブジェクト

次のオブジェクトは、housekeepingAPI に直接関連しています。

ハウスキーピング

settings オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
hk_events_mode	integer	イベントとアラートの内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_events_trigger	string	トリガーデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
hk_events_service	string	サービスデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_internal	string	内部データの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_discovery	string	ネットワークディスカバリーデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_autoreg	string	自動登録データの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。

プロパティ	タイプ	説明
hk_services_mode	integer	サービスの内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_services	string	サービスデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
hk_audit_mode	integer	監査の内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_audit	string	監査データの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_sessions_mode	integer	セッションの内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_sessions	string	セッションデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_history_mode	integer	履歴の内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_history_global	integer	アイテムの履歴期間を上書きします。 可能な値: 0 - 上書きしない; 1 - *(デフォルト)* 上書きする。
hk_history	string	履歴データの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_trends_mode	integer	トレンドの内部ハウスキーピングを有効にします。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_trends_global	integer	アイテムのトレンド期間を上書きします。 可能な値: 0 - 上書きしない; 1 - *(デフォルト)* 上書きする。
hk_trends	string	トレンドデータの保存期間。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
db_extension	string	設定フラグ DB 拡張。このフラグが”timescaledb” に設定されている場合、サーバーはハウスキーピングとアイテム削除の動作を変更します。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
compression_availability	integer	データベース（またはその拡張機能）でデータ圧縮がサポートされているかどうか。 可能な値: 0 - 利用不可; 1 - 利用可能。 プロパティの動作: - 読み取り専用
compression_status	integer	履歴とトレンドに対して TimescaleDB 圧縮を有効にします。 可能な値: 0 - *(デフォルト)* オフ; 1 - オン。
compress_older	string	指定した期間より古い履歴およびトレンドレコードを圧縮します。秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 7d。

housekeeping.get

説明

`object housekeeping.get(object parameters)`

このメソッドは、指定されたパラメータに従ってハウスキーピングオブジェクトを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	タイプ	説明
output	query	このパラメータについてはリファレンス解説で説明されています。

戻り値

(object) housekeeping オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
```

```
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```

ソース

CHousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

説明

`object housekeeping.update(object housekeeping)`

このメソッドは、既存のハウスキーピング設定を更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するハウスキーピングプロパティ。

戻り値

(array) 更新されたパラメーター名の配列を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.update",
    "params": {
        "hk_events_mode": "1",
        "hk_events_trigger": "200d",
        "hk_events_internal": "2d",
        "hk_events_discovery": "2d"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

ソース

CHousekeeping::update() in *ui/include/classes/api/services/CHousekeeping.php*.

Icon map

このクラスは、アイコンマップを操作するためのものです。

オブジェクトリファレンス:

- Icon map
- Icon mapping

利用可能なメソッド:

- iconmap.create - 新しいアイコンマップを作成する
- iconmap.delete - アイコンマップを削除する
- iconmap.get - アイコンマップを取得する
- iconmap.update - アイコンマップを更新する

Icon map オブジェクト

次のオブジェクトは、`iconmap`API に直接関連しています。

アイコンマップ

アイコンマップオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
iconmapid	ID	アイコンマップの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
default_iconid	ID	デフォルトアイコンの ID。 プロパティの動作: - 作成操作時に必須
name	string	アイコンマップの名前。 プロパティの動作: - 作成操作時に必須

アイコンマッピング

アイコンマッピングオブジェクトは、特定のインベントリフィールド値を持つホストに使用する特定のアイコンを定義します。以下のプロパティがあります。

プロパティ	型	説明
iconid	ID	アイコンマッピングで使用されるアイコンの ID。 プロパティの動作: - 必須

プロパティ	型	説明
expression	string	インベントリフィールドと照合する式。 プロパティの動作: - 必須
inventory_link	integer	ホストインベントリフィールドの ID。 サポートされているインベントリフィールドの一覧については、ホストインベントリオブジェクトを参照してください。 プロパティの動作: - 必須
sortorder	integer	アイコンマップ内でのアイコンマッピングの位置。 プロパティの動作: - 読み取り専用

iconmap.create

説明

`object iconmap.create(object/array iconMaps)`

このメソッドは新しいアイコンマップを作成します。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するアイコンマップ。

標準のアイコンマッププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
mappings	array	アイコンマップに作成するアイコンマッピング。 パラメータの動作: - 必須

戻り値

(`object`) `iconmapids` プロパティの下で作成されたアイコンマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイコンマップの順序と一致します。

例

アイコンマップの作成

異なるタイプのホストを表示するためのアイコンマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
        "name": "Type icons",
        "default_iconid": "2",
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
```

```
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

参照

- Icon mapping

ソース

CIconMap::create() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

説明

`object iconmap.delete(array iconMapIds)`

このメソッドは、アイコンマップを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイコンマップの ID。

戻り値

(object) iconmapids プロパティの下にある削除されたアイコンマップの ID を含むオブジェクトを返します。

例

複数のアイコンマップの削除

2 つのアイコンマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CIconMap::delete() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

説明

`integer/array iconmap.get(object parameters)`

このメソッドは、指定されたパラメータに従ってアイコンマップを取得することができます。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object`) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
iconmapids	ID/array	指定した ID のアイコンマップのみを返します。
sysmapids	ID/array	指定したマップで使用されているアイコンマップのみを返します。
selectMappings	query	使用されているアイコンマッピングを持つ`mappings`プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `iconmapid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

アイコンマップの取得

アイコンマップ"3" のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
                    "inventory_link": "1",
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

参照

- Icon mapping

ソース

CIconMap::get() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

説明

`object iconmap.update(object/array iconMaps)`

このメソッドは、既存のアイコンマップを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新されるアイコンマップのプロパティ

iconmapid プロパティはアイコンマップごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のアイコンマッププロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
mappings	array	既存のアイコンマッピングを置き換えます

戻り値

(object) iconmapids プロパティの下にある更新されたアイコンマップの ID を含むオブジェクトを返します

例

アイコンマップの名前変更

アイコンマップの名前を”OS アイコン” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

参照

- Icon mapping

ソース

CIconMap::update() in *ui/include/classes/api/services/CIconMap.php*.

Image

このクラスは、画像を操作するためのものです。

オブジェクトリファレンス:

- Image

利用可能なメソッド:

- image.create - 新しい画像を作成する
- image.delete - 画像を削除する
- image.get - 画像を取得する
- image.update - 画像を更新する

Image オブジェクト

次のオブジェクトは、imageAPI に直接関連しています。

イメージ

イメージオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
imageid	ID	イメージの ID。 プロパティの動作: - 読み取り専用 - 更新操作には必須
name	string	イメージの名前。 プロパティの動作: - 作成操作には必須
imagetype	integer	イメージのタイプ。 可能な値: 1 - *(デフォルト)* アイコン; 2 - 背景画像。 プロパティの動作: - 定数 - 作成操作には必須
image	string	Base64 エンコードされたイメージ。 エンコードされたイメージの最大サイズは 1MB です。最大サイズは ZBX_MAX_IMAGE_SIZE 定数値を変更することで調整できます。 サポートされているイメージフォーマット: PNG、JPEG、GIF。 プロパティの動作: - 作成操作には必須

image.create

説明

`object image.create(object/array images)`

このメソッドは新しいイメージを作成します。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するイメージ。

このメソッドは標準イメージプロパティを持つイメージを受け付けます。

戻り値

(object) imageids プロパティの下で作成された画像の ID を含むオブジェクトを返します。返される ID の順序は、渡された画像の順序と一致します。

例

イメージの作成

クラウドアイコンを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

ソース

CImage::create() in *ui/include/classes/api/services/CImage.php*.

image.delete

説明

`object image.delete(array imageIds)`

このメソッドは、画像を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する画像の ID。

戻り値

(object) imageids プロパティの下にある削除された画像の ID を含むオブジェクトを返します。

例

複数の画像の削除

2 つの画像を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CImage::delete() in *ui/include/classes/api/services/CImage.php*.

image.get

説明

`integer/array image.get(object parameters)`

このメソッドは、指定されたパラメータに従ってイメージを取得することができます。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object`) 目的の出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
imageids	ID/array	指定した ID のイメージのみを返します。
sysmapids	ID/array	指定したマップで使用されているイメージを返します。
select_image	flag	Base64 エンコードされたイメージを持つ `image` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `imageid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

イメージの取得

ID が”2” のイメージの全データを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

ソース

CImage::get() in *ui/include/classes/api/services/CImage.php*.

image.update

説明

`object image.update(object/array images)`

このメソッドは、既存のイメージを更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するイメージのプロパティ。

各イメージに対して `imageid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準イメージプロパティを持つイメージを受け付けます。

戻り値

(object) `imageids` プロパティの下で更新された画像の ID を含むオブジェクトを返します。

例

イメージの名前変更

イメージの名前を「Cloud icon」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CImage::update() in *ui/include/classes/api/services/CImage.php*.

Item

このクラスは、アイテムを操作するためのものです。

オブジェクトリファレンス:

- Item
  - HTTP header
  - HTTP query field
- Item tag
- Item preprocessing

利用可能なメソッド:

- item.create - 新しいアイテムを作成する
- item.delete - アイテムを削除する
- item.get - アイテムを取得する
- item.update - アイテムを更新する

Item オブジェクト

次のオブジェクトは、`itemAPI` に直接関連しています。

アイテム

> **Note:**
> Web アイテムは Zabbix API を通じて直接作成、更新、削除することはできません。

アイテムオブジェクトは以下のプロパティを持ちます。

プロパティ	型	説明
itemid	ID	アイテムの ID。 プロパティの挙動: - 読み取り専用 - 更新操作時に必須

プロパティ	型	説明
delay	string	アイテムの更新間隔。 秒またはサフィックス付きの時間単位（例: 30s, 1m, 2h, 1d）、およびオプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は柔軟間隔とスケジューリング間隔の混在が可能です。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされません。柔軟間隔はスラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの挙動: - `type` が”Zabbix agent”(0)、”Simple check”(3)、”Zabbix internal”(5)、”External check”(10)、”Database monitor”(11)、”IPMI agent”(12)、”SSH agent”(13)、”TELNET agent”(14)、”Calculated”(15)、”JMX agent”(16)、”HTTP agent”(19)、”SNMP agent”(20)、”Script”(21)、”Browser”(22) の場合、または `type` が”Zabbix agent (active)”(7) で `key_` に”mqtt.get” が含まれない場合に必須
hostid	ID	アイテムが属するホストまたはテンプレートの ID。 プロパティの挙動: - 定数 - 作成操作時に必須
interfaceid	ID	アイテムのホストインターフェースの ID。 プロパティの挙動: - アイテムがホストに属し、`type` が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP trap”、”SNMP agent” の場合に必須 - アイテムがホストに属し、`type` が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合にサポート - 検出オブジェクトの場合は読み取り専用
key_	string	アイテムキー。 プロパティの挙動: - 作成操作時に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
name	string	アイテム名。 ユーザーマクロをサポート。 プロパティの挙動: - 作成操作時に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
name_resolved	string	ユーザーマクロを解決したアイテム名。 プロパティの挙動: - 読み取り専用

プロパティ	型	説明
type	integer	アイテムのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. プロパティの挙動: - 作成操作時に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポート。 プロパティの挙動: - type が”HTTP agent” の場合に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
value_type	integer	アイテムの情報タイプ。 指定可能な値: 0 - 数値（浮動小数点）; 1 - 文字列; 2 - ログ; 3 - 数値（符号なし）; 4 - テキスト; 5 - バイナリ。 プロパティの挙動: - 作成操作時に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
allow_traps	integer	trapper アイテムと同様に値の受信を許可するか。 0 - *(デフォルト)* 受信データを許可しない; 1 - 受信データを許可する。 プロパティの挙動: - type が”HTTP agent” の場合にサポート - 検出オブジェクトの場合は読み取り専用

プロパティ	型	説明
authtype	integer	認証方式。 type が”SSH agent” の場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” の場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - ベーシック; 2 - NTLM; 3 - Kerberos。 プロパティの挙動: - type が”SSH agent” または”HTTP agent” の場合にサポート - 継承オブジェクト（type が”HTTP agent” の場合）または検出オブジェクトの場合は読み取り専用
description	string	アイテムの説明。 プロパティの挙動: - 検出オブジェクトの場合は読み取り専用
error	string	アイテム値の更新時に問題が発生した場合のエラーテキスト。 プロパティの挙動: - 読み取り専用
flags	integer	アイテムの起源。 指定可能な値: 0 - 通常のアイテム; 4 - プロトタイプから変換されたアイテム。 プロパティの挙動: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトを追従するか。 指定可能な値: 0 - リダイレクトを追従しない; 1 - *(デフォルト)* リダイレクトを追従する。 プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
history	string	履歴データを保存する期間の時間単位。 ユーザーマクロも使用可能。 デフォルト: 31d。 プロパティの挙動: - 検出オブジェクトの場合は読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用

プロパティ	型	説明
inventory_link	integer	アイテムによって入力されるホストインベントリフィールドの ID。 サポートされるホストインベントリフィールドとその ID についてはホストインベントリページを参照してください。 デフォルト: 0。 プロパティの挙動: - `value_type` が”numeric float”、”character”、”numeric unsigned”、”text” の場合にサポート - 検出オブジェクトの場合は読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの挙動: - `type` が”IPMI agent” かつ `key_` が”ipmi.get” でない場合に必須 - `type` が”IPMI agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの挙動: - `type` が”JMX agent” の場合にサポート - 検出オブジェクトの場合は読み取り専用
lastclock	timestamp	アイテム値が最後に更新された時刻。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は管理 → 一般メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの挙動: - 読み取り専用
lastns	integer	アイテム値が最後に更新された時のナノ秒。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は管理 → 一般メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの挙動: - 読み取り専用
lastvalue	string	アイテムの最新値。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は管理 → 一般メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの挙動: - 読み取り専用
logtimefmt	string	ログエントリの時刻フォーマット。 プロパティの挙動: - `value_type` が”log” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの挙動: - `type` が”Dependent item” の場合に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用

プロパティ	型	説明
output_format	integer	レスポンスを JSON に変換するかどうか。 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。
params	string	プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用 アイテムのタイプに応じた追加パラメータ: - SSH agent および TELNET agent アイテムの場合は実行スクリプト; - database monitor アイテムの場合は SQL クエリ; - calculated アイテムの場合は計算式; - script および browser アイテムの場合はスクリプト。
parameters	object/array	プロパティの挙動: - type が”Database monitor”、”SSH agent”、”TELNET agent”、”Calculated”、”Script”、”Browser” の場合に必須 - 継承オブジェクト（type が”Script” または”Browser” の場合）または検出オブジェクトの場合は読み取り専用 type が”Script” または”Browser” の場合の追加パラメータ。name と value プロパティを持つオブジェクトの配列で、name は一意でなければなりません。
password	string	プロパティの挙動: - type が”Script” または”Browser” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用 認証用パスワード。
post_type	integer	プロパティの挙動: - type が”JMX agent” かつ username が設定されている場合に必須 - type が”Simple check”、”SSH agent”、”TELNET agent”、”Database monitor”、”HTTP agent” の場合にサポート - 継承オブジェクト（type が”HTTP agent” の場合）または検出オブジェクトの場合は読み取り専用 posts プロパティに格納される POST データ本体のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。
posts	string	プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用 HTTP(S) リクエストボディデータ。
prevvalue	string	プロパティの挙動: - type が”HTTP agent” かつ post_type が”JSON data” または”XML data” の場合に必須 - type が”HTTP agent” かつ post_type が”Raw data” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用 アイテムの前回値。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は管理 → 一般メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの挙動: - 読み取り専用

プロパティ	型	説明
privatekey	string	秘密鍵ファイル名。 プロパティの挙動: - `type` が”SSH agent” かつ `authtype` が”public key” の場合に必須 - 検出オブジェクトの場合は読み取り専用
publickey	string	公開鍵ファイル名。 プロパティの挙動: - `type` が”SSH agent” かつ `authtype` が”public key” の場合に必須 - 検出オブジェクトの場合は読み取り専用
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、”PUT” の場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” の場合の指定可能な値: 1 - ヘッダー。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
snmp_oid	string	SNMP OID。 プロパティの挙動: - `type` が”SNMP agent” の場合に必須 - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
ssl_key_password	string	SSL キーのパスワード。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用

プロパティ	型	説明
state	integer	アイテムの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポート外。 プロパティの挙動: - 読み取り専用
status	integer	アイテムのステータス。 指定可能な値: 0 - *(デフォルト)* 有効なアイテム; 1 - 無効なアイテム。
status_codes	string	カンマ区切りの必要な HTTP ステータスコードの範囲。 カンマ区切りリストの一部としてユーザーマクロもサポート。 例: 200,200-{$M},{$M},200-400 プロパティの挙動: - type が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
templateid	ID	親テンプレートアイテムの ID。 ヒント: アイテムが属するテンプレートを指定するには hostid プロパティを使用してください。 プロパティの挙動: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒またはサフィックス付きの時間単位（例: 30s, 1m）を指定可能。ユーザーマクロも使用可能。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの挙動: - type が”Zabbix agent”(0)、”Simple check”(3) かつ key_ が”vmware.” および”icmpping” で始まらない場合、”Zabbix agent (active)”(7)、”External check”(10)、”Database monitor”(11)、”SSH agent”(13)、”TELNET agent”(14)、”HTTP agent”(19)、”SNMP agent”(20) かつ snmp_oid が”walk[” または”get[” で始まる場合、”Script”(21)、”Browser”(22) の場合にサポート - 継承および検出オブジェクトの場合は読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの挙動: - 検出オブジェクトの場合は読み取り専用 - type が”Zabbix trapper”、または type が”HTTP agent” かつ allow_traps が” 受信データを許可する” の場合にサポート
trends	string	トレンドデータを保存する期間の時間単位。 ユーザーマクロも使用可能。 デフォルト: 365d。 プロパティの挙動: - value_type が”numeric float” または”numeric unsigned” の場合にサポート - 検出オブジェクトの場合は読み取り専用

プロパティ	型	説明
units	string	値の単位。 プロパティの挙動: - `value_type` が”numeric float” または”numeric unsigned” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
username	string	認証用ユーザー名。 プロパティの挙動: - `type` が”SSH agent”、”TELNET agent”、または `type` が”JMX agent” かつ `password` が設定されている場合に必須 - `type` が”Simple check”、”Database monitor”、”HTTP agent” の場合にサポート - 継承オブジェクト（`type` が”HTTP agent” の場合）または検出オブジェクトの場合は読み取り専用
uuid	string	インポートされたアイテムを既存のものにリンクするためのユニバーサル一意識別子。指定がない場合は自動生成。 プロパティの挙動: - アイテムがテンプレートに属する場合にサポート
valuemapid	ID	関連付けられた値のマップの ID。 プロパティの挙動: - `value_type` が”numeric float”、”character”、”numeric unsigned” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
verify_host	integer	接続先のホスト名が証明書のホスト名と一致するか検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
verify_peer	integer	ホストの証明書が正当かどうか検証するか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの挙動: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用

HTTP header

ヘッダーオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	HTTP ヘッダー名 プロパティの動作: - 必須
value	文字列	ヘッダー値 プロパティの動作: - 必須

HTTP query field

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	パラメータ名 プロパティの動作: - 必須
value	文字列	パラメータ値 プロパティの動作: - 必須

アイテムタグ

アイテムタグオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	アイテムタグ名。 プロパティの動作: - 必須
value	string	アイテムタグの値。

アイテムの事前処理

アイテムの事前処理オブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	事前処理オプションのタイプ。 指定可能な値: 1 - カスタム乗数; 2 - 右側のトリム; 3 - 左側のトリム; 4 - トリム; 5 - 正規表現; 6 - ブール値を 10 進数に変換; 7 - 8 進数を 10 進数に変換; 8 - 16 進数を 10 進数に変換; 9 - 単純な変化量; 10 - 秒間変化量; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内判定; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラー判定; 17 - XML のエラー判定; 18 - 正規表現によるエラー判定; 19 - 変化なしを破棄; 20 - ハートビート付き変化なしを破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus を JSON に変換; 24 - CSV を JSON に変換; 25 - 置換; 26 - サポート外判定; 27 - XML を JSON に変換; 28 - SNMP walk 値; 29 - SNMP walk を JSON に変換; 30 - SNMP get 値。 プロパティの動作: - 必須

プロパティ	タイプ	説明
params	string	事前処理オプションで使用される追加パラメータ。 複数のパラメータは改行 (\n) で区切ります。 type が「サポート外判定」に設定されている場合、パラメータは <scope>[\n<pattern>] の構文に従います。*pattern* は正規表現、 *scope* は以下のいずれかです: -1 - すべてのエラーに一致; 0 - エラーメッセージが *pattern* に一致するか判定; 1 - エラーメッセージが *pattern* に一致しないか判定。 プロパティの動作: - type が「カスタム乗数」(1)、「右側のトリム」(2)、「左側のトリム」(3)、「トリム」(4)、「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「範囲内判定」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラー判定」(16)、「XML のエラー判定」(17)、「正規表現によるエラー判定」(18)、「ハートビート付き変化なしを破棄」(20)、「JavaScript」(21)、「Prometheus パターン」(22)、「Prometheus を JSON に変換」(23)、「CSV を JSON に変換」(24)、「置換」(25)、サポート外判定 (26)、「SNMP walk 値」(28)、「SNMP walk を JSON に変換」(29)、または「SNMP get 値」(30) に設定されている場合は必須
error_handler	integer	事前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値: 0 - Zabbix サーバーがエラーメッセージを設定; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が「サポート外判定」に設定されている場合の指定可能な値: 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「カスタム乗数」(1)、「正規表現」(5)、「ブール値を 10 進数に変換」(6)、「8 進数を 10 進数に変換」(7)、「16 進数を 10 進数に変換」(8)、「単純な変化量」(9)、「秒間変化量」(10)、「XML XPath」(11)、「JSONPath」(12)、「範囲内判定」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラー判定」(16)、「XML のエラー判定」(17)、「正規表現によるエラー判定」(18)、「Prometheus パターン」(22)、「Prometheus を JSON に変換」(23)、「CSV を JSON に変換」(24)、「サポート外判定」(26)、「XML を JSON に変換」(27)、「SNMP walk 値」(28)、「SNMP walk を JSON に変換」(29)、または「SNMP get 値」(30) に設定されている場合は必須
error_handler_params	string	エラーハンドラのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は必須

各事前処理タイプでサポートされるパラメータとエラーハンドラは以下の通りです。

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
1	カスタム乗数	数値[1, 6]			0, 1, 2, 3
2	右側のトリム	文字列リスト[2]			

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
3	左側のトリム	文字列リスト[2]			
4	トリム	文字列リスト[2]			
5	正規表現	パターン[3]	出力[2]		0, 1, 2, 3
6	ブール値を 10 進数に変換				0, 1, 2, 3
7	8 進数を 10 進数に変換				0, 1, 2, 3
8	16 進数を 10 進数に変換				0, 1, 2, 3
9	単純な変化量				0, 1, 2, 3
10	秒間変化量				0, 1, 2, 3
11	XML XPath	パス[4]			0, 1, 2, 3
12	JSONPath	パス[4]			0, 1, 2, 3
13	範囲内判定	min[1, 6]	max[1, 6]		0, 1, 2, 3
14	正規表現に一致	パターン[3]			0, 1, 2, 3
15	正規表現に一致しない	パターン[3]			0, 1, 2, 3
16	JSON のエラー判定	パス[4]			0, 1, 2, 3
17	XML のエラー判定	パス[4]			0, 1, 2, 3
18	正規表現によるエラー判定	パターン[3]	出力[2]		0, 1, 2, 3

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
19	変化なしを破棄				
20	ハートビート付き変化なしを破棄	秒数[5, 6]			
21	JavaScript	スクリプト[2]			
22	Prometheus パターン	パターン[6, 7]	`value`, `label`, `function`	出力[8, 9]	0, 1, 2, 3
23	Prometheus を JSON に変換	パターン[6, 7]			0, 1, 2, 3
24	CSV を JSON に変換	文字[2]	文字[2]	0,1	0, 1, 2, 3
25	置換	検索文字列[2]	置換文字列[2]		
26	サポート外判定	スコープ[1]	パターン[3, 6]		1, 2, 3
27	XML を JSON に変換				0, 1, 2, 3
28	SNMP walk 値	OID[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk を JSON に変換[10]	フィールド名[2]	OID プレフィックス[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	フォーマット: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 浮動小数点数 (整数は暗黙的に float 値に変換されます)

[2] 文字列
[3] 正規表現
[4] JSONPath または XML XPath
[5] 正の整数 (時間のサフィックス指定可、例: 30s, 1m, 2h, 1d)
[6] ユーザーマクロ
[7] Prometheus パターン (構文: <metric name>{<label name>="<label value>", ...} == <value>)。各 Prometheus パターン要素 (メトリック、ラベル名、ラベル値、メトリック値) はユーザーマクロ可。
[8] Prometheus 出力 (構文: <label name>。ユーザーマクロ可)。2 番目のパラメータが label の場合。
[9] 2 番目のパラメータが function の場合、集計関数のいずれか: sum, min, max, avg, count。
[10] 複数の「フィールド名,OID プレフィックス, フォーマット」レコードを改行で区切って指定可能。

item.create

説明

`object item.create(object/array items)`

このメソッドは新しいアイテムを作成するためのものです。

> **Note:**
> Web アイテムは Zabbix API 経由では作成できません。

> **Note:**
> このメソッドは管理者および *Zabbix* 管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するアイテム。

標準のアイテムプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
preprocessing	array	アイテムの事前処理オプション。
tags	array	アイテムタグ。

戻り値

(object) 作成されたアイテムの ID を itemids プロパティに含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

アイテムの作成

ID "30074" のホストで空きディスク容量を監視するために、2 つのアイテムタグを持つ数値型 Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
```

```
            {
                "tag": "equipment",
                "value": "workstation"
            }
        ],
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

ホストインベントリアイテムの作成

ホストの「OS」インベントリフィールドを埋めるために、Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "uname",
        "key_": "system.uname",
        "hostid": "30021",
        "type": 0,
        "interfaceid": "30007",
        "value_type": 1,
        "delay": "10s",
        "inventory_link": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

前処理付きアイテムの作成

カスタム乗数を使用してアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
```

```
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

従属アイテムの作成

ID 24759 のマスターアイテムに対して従属アイテムを作成します。同じホストでのみ依存関係が許可されているため、マスターアイテムと従属アイテムは同じ hostid である必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
      "hostid": "30074",
      "name": "Dependent test item",
      "key_": "dependent.item",
      "type": 18,
      "master_itemid": "24759",
      "value_type": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの作成

JSON レスポンスの前処理を行う POST リクエストメソッドアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "url":"http://127.0.0.1/http.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value": "100"
            }
        ],
        "interfaceid": "1",
        "type": 19,
        "hostid": "10254",
        "delay": "5s",
        "key_": "json",
        "name": "HTTP agent example JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの作成

スクリプトアイテムを使用してシンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.item",
        "hostid": "12345",
```

```
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItem::create() in *ui/include/classes/api/services/CItem.php*.

item.delete

説明

`object item.delete(array itemIds)`

このメソッドは、アイテムを削除することができます。

Note:
Web アイテムは、Zabbix API を使用しての削除はできません。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイテムの ID。

戻り値

(object) itemids プロパティの下にある削除されたアイテムの ID を含むオブジェクトを返します。

例

複数のアイテムの削除

2 つのアイテムを削除します。
マスターアイテムが削除されると、依存アイテムとアイテムのプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
```

```
        "22986"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

ソース

CItem::delete() in *ui/include/classes/api/services/CItem.php*.

item.get

説明

`integer/array item.get(object parameters)`

このメソッドは、指定したパラメータに従ってアイテムを取得するためのものです。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
itemids	ID/array	指定した ID のアイテムのみを返します。
groupids	ID/array	指定したグループのホストに属するアイテムのみを返します。
templateids	ID/array	指定したテンプレートに属するアイテムのみを返します。
hostids	ID/array	指定したホストに属するアイテムのみを返します。
proxyids	ID/array	指定したプロキシによって監視されているアイテムのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用するアイテムのみを返します。
graphids	ID/array	指定したグラフで使用されているアイテムのみを返します。
triggerids	ID/array	指定したトリガーで使用されているアイテムのみを返します。
webitems	flag	結果に Web アイテムを含めます。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するアイテムのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効なアイテムのみを返します。
group	string	指定した名前のグループに属するアイテムのみを返します。
host	string	指定した名前のホストに属するアイテムのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値： 0 - *(デフォルト)* And/Or; 2 - Or。

パラメーター	型	説明
tags	array	指定したタグを持つアイテムのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのアイテムを返します。 指定可能なoperator値： 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
with_triggers	boolean	`true` に設定すると、トリガーで使用されているアイテムのみを返します。
selectHosts	query	アイテムが属するホストの配列を持つ`hosts`プロパティを返します。
selectInterfaces	query	アイテムで使用されているホストインターフェースの配列を持つ`interfaces`プロパティを返します。
selectTriggers	query	アイテムが使用されているトリガーを持つ`triggers`プロパティを返します。 `count` をサポートします。
selectGraphs	query	アイテムを含むグラフを持つ`graphs`プロパティを返します。 `count` をサポートします。
selectDiscoveryData	query	アイテムディスカバリオブジェクトデータを持つ `discoveryData` プロパティを返します。アイテムディスカバリオブジェクトは、検出されたアイテムを、そのアイテムが検出されたアイテムプロトタイプにリンクします。 以下のプロパティを持ちます： `parent_itemid` - (`string`) アイテムが作成されたアイテムプロトタイプの ID; `key_` - (`string`) アイテムプロトタイプのキー; `status` - (`int`) アイテムディスカバリのステータス： 0 - *(デフォルト)* アイテムは検出されている, 1 - アイテムはもはや検出されていない; `ts_delete` - (`timestamp`) もはや検出されていないアイテムが削除される時刻; `ts_disable` - (`timestamp`) もはや検出されていないアイテムが無効化される時刻; `disable_source` - (`int`) アイテムが LLD ルールまたは手動で無効化されたかどうかの指標： 0 - *(デフォルト)* 自動的に無効化, 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムを作成した LLD ルールを持つ`discoveryRule`プロパティを返します。
selectPreprocessing	query	アイテムの前処理オプションを持つ`preprocessing`プロパティを返します。
selectTags	query	アイテムタグを`tags`プロパティで返します。
selectValueMap	query	アイテムの値マップを持つ`valuemap`プロパティを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる値または値の配列です。 `text`データ型のプロパティはサポートしません。 追加プロパティをサポートします： `host` - アイテムが属するホストの技術名。

パラメーター	型	説明
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます： selectGraphs - 結果は name でソートされます; selectTriggers - 結果は description でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値：itemid, name, key_, delay, history, trends, type, status。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	アイテムディスカバリオブジェクトを持つ itemDiscovery プロパティを返します。アイテムディスカバリオブジェクトは、アイテムを作成元のアイテムプロトタイプにリンクします。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

キーによるアイテムの検索

特定のホスト ID のトリガーで使用されているアイテムのうち、アイテムキーに「system.cpu」という単語が含まれているものを取得し、名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
```

```
        "type": "18",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "CPU utilization",
        "key_": "system.cpu.util",
        "delay": "0",
        "history": "7d",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "%",
        "logtimefmt": "",
        "templateid": "42267",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "0",
        "description": "CPU utilization in %.",
        "inventory_link": "0",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "42264",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0",
        "name_resolved": "CPU utilization"
    },
    {
        "itemid": "42259",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
```

```
        "name": "Load average (15m avg)",
        "key_": "system.cpu.load[all,avg15]",
        "delay": "1m",
        "history": "7d",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "logtimefmt": "",
        "templateid": "42219",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "1",
        "description": "",
        "inventory_link": "0",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0",
        "name_resolved": "Load average (15m avg)"
    },
    {
        "itemid": "42249",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Load average (1m avg)",
        "key_": "system.cpu.load[all,avg1]",
        "delay": "1m",
```

```
        "history": "7d",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "logtimefmt": "",
        "templateid": "42209",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "1",
        "description": "",
        "inventory_link": "0",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0",
        "name_resolved": "Load average (1m avg)"
    },
    {
        "itemid": "42257",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Load average (5m avg)",
        "key_": "system.cpu.load[all,avg5]",
        "delay": "1m",
        "history": "7d",
        "trends": "365d",
        "status": "0",
```

```
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (5m avg)"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "key_": "system.cpu.num",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
```

```
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Number of CPUs"
        }
    ],
    "id": 1
}
```

キーで依存アイテムを検索する

ID が”10116” のホストから、キーに”apache” という単語が含まれるすべての依存アイテムを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
```

```
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25550",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Days",
            "key_": "apache.status.uptime.days",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
```

```
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Days"
        },
        {
            "itemid": "25555",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Hours",
            "key_": "apache.status.uptime.hours",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
```

```
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Hours"
        }
    ],
    "id": 1
}
```

HTTP エージェントアイテムの検索

特定のホスト ID に対して、POST ボディタイプが XML の HTTP エージェントアイテムを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28252",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10255",
            "name": "template item",
            "key_": "ti",
            "delay": "30s",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
```

```
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": "",
            "name_resolved": "template item"
        }
    ],
    "id": 1
}
```

前処理ルールを持つアイテムの取得

特定のホスト ID のすべてのアイテムとその前処理ルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
```

```
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

参照

- Discovery rule
- Graph
- Host
- Host interface
- Trigger

ソース

CItem::get() in *ui/include/classes/api/services/CItem.php*.

item.update

説明

`object item.update(object/array items)`

このメソッドは、既存のアイテムを更新するためのものです。

Note:
Web アイテムは Zabbix API 経由で更新できません。

Note:
このメソッドは管理者および *Zabbix* 管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するアイテムのプロパティ。

各アイテムには `itemid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のアイテムプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
preprocessing	array	現在の前処理オプションを置き換えるアイテム前処理オプション。 パラメータの動作: - 継承オブジェクトまたは検出オブジェクトの場合は読み取り専用
tags	array	アイテムタグ。 パラメータの動作: - 検出オブジェクトの場合は読み取り専用

戻り値

(`object`) `itemids` プロパティの下で更新されたアイテムの ID を含むオブジェクトを返します。

例

アイテムの有効化

アイテムを有効にします。つまり、ステータスを”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

従属アイテムの更新

従属アイテム名とマスターアイテム ID を更新します。同じホストでのみ依存関係が許可されているため、マスターアイテムと従属アイテムは同じ hostid である必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": " 従属アイテムの更新名",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの更新

アイテム値のトラップを有効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

前処理を使用したアイテムの更新

アイテムの前処理ルール「範囲内」でアイテムを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの更新

異なるスクリプトでスクリプトアイテムを更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItem::update() in *ui/include/classes/api/services/CItem.php*.

Item prototype

このクラスは、アイテムプロトタイプを操作するためのものです。

オブジェクトリファレンス:

- Item prototype
  - HTTP header
  - HTTP query field
- Item prototype tag
- Item prototype preprocessing

利用可能なメソッド:

- itemprototype.create - 新しいアイテムプロトタイプを作成する
- itemprototype.delete - アイテムプロトタイプを削除する
- itemprototype.get - アイテムプロトタイプを取得する
- itemprototype.update - アイテムプロトタイプを更新する

Item prototype オブジェクト

次のオブジェクトは、`itemprototype`API に直接関連しています。

アイテムプロトタイプ

アイテムプロトタイプオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	アイテムプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須

プロパティ	型	説明
delay	string	アイテムプロトタイプの更新間隔。 秒またはサフィックス付きの時間単位 (例: 30s, 1m, 2h, 1d) と、オプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は柔軟な間隔とスケジューリング間隔の混在が可能です。 ユーザーマクロおよび LLD マクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされません。柔軟な間隔はスラッシュで区切った 2 つのマクロとして記述できます (例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`)。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が”Zabbix agent”(0)、”Simple check”(3)、”Zabbix internal”(5)、”External check”(10)、”Database monitor”(11)、”IPMI agent”(12)、”SSH agent”(13)、”TELNET agent”(14)、”Calculated”(15)、”JMX agent”(16)、”HTTP agent”(19)、”SNMP agent”(20)、”Script”(21)、”Browser”(22) の場合、または `type` が”Zabbix agent (active)”(7) で `key_` に”mqtt.get” が含まれない場合に必須
hostid	ID	アイテムプロトタイプが属するホストの ID。 プロパティの動作: - 定数 - 作成操作時に必須
interfaceid	ID	アイテムプロトタイプのホストインターフェースの ID。 プロパティの動作: - アイテムプロトタイプがホストに属し、`type` が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP trap”、”SNMP agent” の場合に必須 - アイテムプロトタイプがホストに属し、`type` が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合にサポート
key_	string	アイテムプロトタイプのキー。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
name	string	アイテムプロトタイプの名前。 ユーザーマクロをサポート。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
type	integer	アイテムプロトタイプのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
url	string	URL 文字列。 LLD マクロ、ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポート。 プロパティの動作: - type が”HTTP agent” の場合に必須 - 継承オブジェクトの場合は読み取り専用
value_type	integer	アイテムプロトタイプの情報タイプ。 指定可能な値: 0 - 数値 (浮動小数点); 1 - 文字列; 2 - ログ; 3 - 数値 (整数); 4 - テキスト; 5 - バイナリ。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
allow_traps	integer	trapper アイテムと同様に値の受信を許可するか。 0 - *(デフォルト)* 受信データを許可しない; 1 - 受信データを許可する。 プロパティの動作: - type が”HTTP agent” の場合にサポート

プロパティ	型	説明
authtype	integer	認証方式。 type が”SSH agent” の場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” の場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - ベーシック; 2 - NTLM; 3 - Kerberos。 プロパティの動作: - type が”SSH agent” または”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用 (type が”HTTP agent” の場合)
description	string	アイテムプロトタイプの説明。
flags	integer	アイテムプロトタイプのオリジン。 指定可能な値: 2 - アイテムプロトタイプ; 6 - 検出されたアイテムプロトタイプ プロパティの動作: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトを追従するか。 指定可能な値: 0 - リダイレクトを追従しない; 1 - *(デフォルト)* リダイレクトを追従する。 プロパティの動作: - type が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
history	string	履歴データを保存する期間の時間単位。 ユーザーマクロおよび LLD マクロも使用可能。 デフォルト: 31d。
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が”IPMI agent” で key_ が”ipmi.get” でない場合に必須 - type が”IPMI agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が”JMX agent” の場合にサポート

プロパティ	型	説明
logtimefmt	string	ログエントリの時刻フォーマット。 プロパティの動作: - `value_type` が”log” の場合にサポート - 継承オブジェクトの場合は読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの動作: - `type` が”Dependent item” の場合に必須 - 継承オブジェクトの場合は読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
params	string	アイテムプロトタイプのタイプに応じた追加パラメータ: - SSH agent および TELNET agent アイテムプロトタイプの場合は実行スクリプト; - database monitor アイテムプロトタイプの場合は SQL クエリ; - calculated アイテムプロトタイプの場合は計算式; - script および browser アイテムプロトタイプの場合はスクリプト。 プロパティの動作: - `type` が”Database monitor”、”SSH agent”、”TELNET agent”、”Calculated”、”Script”、”Browser” の場合に必須 - 継承オブジェクトの場合は読み取り専用 (`type` が”Script” または”Browser” の場合)
parameters	object/array	`type` が”Script” または”Browser” の場合の追加パラメータ。`name` と `value` プロパティを持つオブジェクトの配列で、`name` は一意でなければなりません。 プロパティの動作: - `type` が”Script” または”Browser” の場合にサポート - 継承オブジェクトの場合は読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が”JMX agent” で `username` が設定されている場合に必須 - `type` が”Simple check”、”SSH agent”、”TELNET agent”、”Database monitor”、”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用 (`type` が”HTTP agent” の場合)
post_type	integer	`posts` プロパティに格納される POST データボディのタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ。 2 - JSON データ。 3 - XML データ。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
posts	string	HTTP(S) リクエストボディデータ。 プロパティの動作: - `type` が”HTTP agent” で `post_type` が”JSON data” または”XML data” の場合に必須 - `type` が”HTTP agent” で `post_type` が”Raw data” の場合にサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” で `authtype` が”public key” の場合に必須
publickey	string	公開鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” で `authtype` が”public key” の場合に必須
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、”PUT” の場合の指定可能な値: 0 - *(デフォルト)* ボディ; 1 - ヘッダー; 2 - ボディとヘッダーの両方を保存。 `request_method` が”HEAD” の場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が”SNMP agent” の場合に必須 - 継承オブジェクトの場合は読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_password	string	SSL キーのパスワード。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
status	integer	アイテムプロトタイプのステータス。 指定可能な値: 0 - *(デフォルト)* 有効なアイテムプロトタイプ; 1 - 無効なアイテムプロトタイプ; 3 - サポートされていないアイテムプロトタイプ。
status_codes	string	カンマ区切りの必要な HTTP ステータスコードの範囲。カンマ区切りリストの一部としてユーザーマクロまたは LLD マクロもサポート。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
templateid	ID	親テンプレートアイテムプロトタイプの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒またはサフィックス付きの時間単位 (例: 30s, 1m) を指定可能。ユーザーマクロおよび LLD マクロも使用可能。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - `type` が”Zabbix agent”(0)、”Simple check”(3) で `key_` が”`vmware.`” または”`icmpping`” で始まらない場合、”Zabbix agent (active)”(7)、”External check”(10)、”Database monitor”(11)、”SSH agent”(13)、”TELNET agent”(14)、”HTTP agent”(19)、”SNMP agent”(20) で `snmp_oid` が”`walk[`” または”`get[`” で始まる場合、”Script”(21)、”Browser”(22) の場合にサポート - 継承オブジェクトの場合は読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - `type` が”Zabbix trapper” の場合、または `type` が”HTTP agent” で `allow_traps` が”Allow to accept incoming data” の場合にサポート
trends	string	トレンドデータを保存する期間の時間単位。 ユーザーマクロおよび LLD マクロも使用可能。 デフォルト: 365d。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” の場合にサポート
units	string	値の単位。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” の場合にサポート - 継承オブジェクトの場合は読み取り専用
username	string	認証用ユーザー名。 プロパティの動作: - `type` が”SSH agent” または”TELNET agent” の場合、または `type` が”JMX agent” で `password` が設定されている場合に必須 - `type` が”Simple check”、”Database monitor”、”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用 (`type` が”HTTP agent” の場合)

プロパティ	型	説明
uuid	string	インポートされたアイテムプロトタイプを既存のものにリンクするためのユニバーサル一意識別子。指定しない場合は自動生成。 プロパティの動作: - アイテムプロトタイプがテンプレートに属する場合にサポート
valuemapid	ID	関連付けられた値のマップの ID。 プロパティの動作: - `value_type` が”numeric float”、”character”、”numeric unsigned” の場合にサポート - 継承オブジェクトの場合は読み取り専用
verify_host	integer	接続先のホスト名が証明書のホスト名と一致するか検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
verify_peer	integer	ホストの証明書が正当かどうか検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” の場合にサポート - 継承オブジェクトの場合は読み取り専用
discover	integer	アイテムプロトタイプのディスカバリーステータス。 指定可能な値: 0 - *(デフォルト)* 新しいアイテムをディスカバリーする; 1 - 新しいアイテムをディスカバリーせず、既存のアイテムをロストとしてマークする。

HTTP header

ヘッダーオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	HTTP ヘッダー名 プロパティの動作: - 必須
value	文字列	ヘッダー値 プロパティの動作: - 必須

HTTP query field

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	パラメータ名 プロパティの動作: - 必須

プロパティ	タイプ	説明
value	文字列	パラメータ値 プロパティの動作: - 必須

アイテムプロトタイプタグ

アイテムプロトタイプタグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	アイテムプロトタイプタグ名。 プロパティの動作: - 必須
value	string	アイテムプロトタイプタグの値。

アイテムプロトタイプの事前処理

アイテムプロトタイプの事前処理オブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	事前処理オプションのタイプ。 指定可能な値： 1 - カスタム乗数; 2 - 右側トリム; 3 - 左側トリム; 4 - トリム; 5 - 正規表現; 6 - ブール値を 10 進数に変換; 7 - 8 進数を 10 進数に変換; 8 - 16 進数を 10 進数に変換; 9 - 単純な変化量; 10 - 秒あたりの変化量; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内判定; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラー判定; 17 - XML のエラー判定; 18 - 正規表現によるエラー判定; 19 - 変化なしを破棄; 20 - ハートビート付き変化なしを破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus を JSON に変換; 24 - CSV を JSON に変換; 25 - 置換; 26 - サポート外判定; 27 - XML を JSON に変換; 28 - SNMP walk 値; 29 - SNMP walk を JSON に変換; 30 - SNMP get 値。 プロパティの動作: - 必須

プロパティ	タイプ	説明
params	string	事前処理オプションで使用される追加パラメータ。 複数のパラメータは改行（\n）で区切ります。 type が「サポート外判定」に設定されている場合、パラメータは <scope>[\n<pattern>] の構文に従います。*pattern* は正規表現、*scope* は以下のいずれかです： -1 - すべてのエラーに一致; 0 - エラーメッセージが *pattern* に一致するか判定; 1 - エラーメッセージが *pattern* に一致しないか判定。 プロパティの動作: - type が「カスタム乗数」(1)、「右側トリム」(2)、「左側トリム」(3)、「トリム」(4)、「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「範囲内判定」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラー判定」(16)、「XML のエラー判定」(17)、「正規表現によるエラー判定」(18)、「ハートビート付き変化なしを破棄」(20)、「JavaScript」(21)、「Prometheus パターン」(22)、「Prometheus を JSON に変換」(23)、「CSV を JSON に変換」(24)、「置換」(25)、「サポート外判定」(26)、「SNMP walk 値」(28)、「SNMP walk を JSON に変換」(29)、「SNMP get 値」(30) の場合は必須
error_handler	integer	事前処理ステップが失敗した場合に使用されるアクションタイプ。 指定可能な値： 0 - Zabbix サーバーがエラーメッセージを設定; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が「サポート外判定」に設定されている場合の指定可能な値： 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「カスタム乗数」(1)、「正規表現」(5)、「ブール値を 10 進数に変換」(6)、「8 進数を 10 進数に変換」(7)、「16 進数を 10 進数に変換」(8)、「単純な変化量」(9)、「秒あたりの変化量」(10)、「XML XPath」(11)、「JSONPath」(12)、「範囲内判定」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラー判定」(16)、「XML のエラー判定」(17)、「正規表現によるエラー判定」(18)、「Prometheus パターン」(22)、「Prometheus を JSON に変換」(23)、「CSV を JSON に変換」(24)、「サポート外判定」(26)、「XML を JSON に変換」(27)、「SNMP walk 値」(28)、「SNMP walk を JSON に変換」(29)、「SNMP get 値」(30) の場合は必須
error_handler_params	string	エラーハンドラのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」の場合は必須

各事前処理タイプでサポートされるパラメータとエラーハンドラは以下の通りです。

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
1	カスタム乗数	数値[1, 6]			0, 1, 2, 3
2	右側トリム	文字列リスト[2]			

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
3	左側トリム	文字列リスト[2]			
4	トリム	文字列リスト[2]			
5	正規表現	パターン[3]	出力[2]		0, 1, 2, 3
6	ブール値を 10 進数に変換				0, 1, 2, 3
7	8 進数を 10 進数に変換				0, 1, 2, 3
8	16 進数を 10 進数に変換				0, 1, 2, 3
9	単純な変化量				0, 1, 2, 3
10	秒あたりの変化量				0, 1, 2, 3
11	XML XPath	パス[4]			0, 1, 2, 3
12	JSONPath	パス[4]			0, 1, 2, 3
13	範囲内判定	min[1, 6]	max[1, 6]		0, 1, 2, 3
14	正規表現に一致	パターン[3]			0, 1, 2, 3
15	正規表現に一致しない	パターン[3]			0, 1, 2, 3
16	JSON のエラー判定	パス[4]			0, 1, 2, 3
17	XML のエラー判定	パス[4]			0, 1, 2, 3
18	正規表現によるエラー判定	パターン[3]	出力[2]		0, 1, 2, 3

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
19	変化なしを破棄				
20	ハートビート付き変化なしを破棄	秒数[5, 6]			
21	JavaScript	スクリプト[2]			
22	Prometheus パターン	パターン[6, 7]	`value`, `label`, `function`	出力[8, 9]	0, 1, 2, 3
23	Prometheus を JSON に変換	パターン[6, 7]			0, 1, 2, 3
24	CSV を JSON に変換	文字[2]	文字[2]	0,1	0, 1, 2, 3
25	置換	検索文字列[2]	置換文字列[2]		
26	サポート外判定	スコープ[1]	パターン[3, 6]		1, 2, 3
27	XML を JSON に変換				0, 1, 2, 3
28	SNMP walk 値	OID[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk を JSON に変換[10]	フィールド名[2]	OID プレフィックス[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	フォーマット: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 整数または浮動小数点数

[2] 文字列
[3] 正規表現
[4] JSONPath または XML XPath
[5] 正の整数（時間のサフィックス指定可。例：30s, 1m, 2h, 1d）
[6] ユーザーマクロ、LLD マクロ
[7] Prometheus パターン。構文は `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターン要素（メトリック、ラベル名、ラベル値、メトリック値）はユーザーマクロまたは LLD マクロにできます。
[8] Prometheus 出力。構文は `<label name>`（ユーザーマクロまたは LLD マクロ可）。2 番目のパラメータが `label` の場合。
[9] 集約関数のいずれか：`sum`, `min`, `max`, `avg`, `count`。2 番目のパラメータが `function` の場合。
[10] 複数の「フィールド名,OID プレフィックス, フォーマット」レコードを改行で区切って指定可能。

itemprototype.create

説明

`object itemprototype.create(object/array itemPrototypes)`

このメソッドは新しいアイテムプロトタイプを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するアイテムプロトタイプ。

標準のアイテムプロトタイププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
ruleid	ID	アイテムが属するLLD ルールの ID。 パラメータの動作: - 必須
preprocessing	array	アイテムプロトタイプの事前処理オプション。
tags	array	アイテムプロトタイプのタグ。

戻り値

`(object)` `itemids` プロパティの下で作成されたアイテムプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムプロトタイプの順序と一致します。

例

アイテムプロトタイプの作成

検出されたファイルシステムの空きディスク容量を監視するアイテムプロトタイプを作成します。検出されたアイテムは、30 秒ごとに更新される数値型の Zabbix エージェントアイテムである必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Free disk space on {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

前処理付きアイテムプロトタイプの作成

1 秒あたりの変化量とカスタム乗数を 2 番目のステップとして使用してアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Incoming network traffic on {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

依存アイテムプロトタイプの作成

ID 44211 のマスターアイテムプロトタイプに対して依存アイテムプロトタイプを作成します。同じホスト（テンプレート/ディスカバリールール）でのみ依存関係が許可されているため、マスターと依存アイテムは同じ hostid および ruleid を持つ必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

HTTP エージェントのアイテムプロトタイプの作成

ユーザーマクロ、クエリフィールド、カスタムヘッダーを使用した URL でアイテムプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "api item prototype example",
        "key_": "api_http_item",
        "value_type": 3,
        "url": "{$URL_PROTOTYPE}",
        "query_fields": [
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value" "100"
            }
        ],
        "headers": [
            {
                "name": "X-Source",
                "value": "api"
            }
        ],
        "delay": "35"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```


スクリプトアイテムプロトタイプの作成

スクリプトアイテムプロトタイプを使用してシンプルなデータ収集を作成します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.itemprototype",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```


ソース

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

説明

`object itemprototype.delete(array itemPrototypeIds)`

このメソッドは、アイテムのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイテムのプロトタイプの ID。

戻り値

(object) prototypeids プロパティの下にある削除されたアイテムのプロトタイプの ID を含むオブジェクトを返します。

例

複数のアイテムのプロトタイプの削除

2 つのアイテムのプロトタイプを削除します。
マスターアイテムまたはアイテムのプロトタイプが削除されると、依存アイテムのプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

説明

`integer/array itemprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従ってアイテムプロトタイプを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
discoveryids	ID/array	指定した LLD ルールに属するアイテムプロトタイプのみを返します。
graphids	ID/array	指定したグラフプロトタイプで使用されているアイテムプロトタイプのみを返します。
hostids	ID/array	指定したホストに属するアイテムプロトタイプのみを返します。
inherited	boolean	true に設定すると、テンプレートから継承されたアイテムプロトタイプのみを返します。

パラメータ	型	説明
itemids	ID/array	指定した ID のアイテムプロトタイプのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効なアイテムプロトタイプのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するアイテムプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するアイテムプロトタイプのみを返します。
triggerids	ID/array	指定したトリガープロトタイプで使用されているアイテムプロトタイプのみを返します。
selectDiscoveryData	query	アイテムプロトタイプのディスカバリオブジェクトデータを持つ `discoveryData` プロパティを返します。アイテムプロトタイプのディスカバリオブジェクトは、検出されたアイテムプロトタイプと、それが検出されたアイテムプロトタイプをリンクします。 以下のプロパティを持ちます: `parent_itemid` - (`string`) アイテムプロトタイプが作成された元のアイテムプロトタイプの ID; `key_` - (`string`) アイテムプロトタイプのキー; `status` - (`int`) アイテムプロトタイプのディスカバリステータス: 0 - *(デフォルト)* アイテムプロトタイプが検出されている; 1 - アイテムプロトタイプが検出されなくなった; `ts_delete` - (`timestamp`) もはや検出されなくなったアイテムプロトタイプが削除される時刻; `ts_disable` - (`timestamp`) もはや検出されなくなったアイテムプロトタイプが無効化される時刻; `disable_source` - (`int`) アイテムプロトタイプが LLD ルールによって無効化されたか手動で無効化されたかの指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムプロトタイプが属する低レベルディスカバリルールを持つ`discoveryRule`プロパティを返します。
selectDiscoveryRulePrototype	query	アイテムプロトタイプが属する親 LLD ルールプロトタイプを持つ`discoveryRulePrototype`プロパティを返します。
selectGraphs	query	アイテムプロトタイプが使用されているグラフプロトタイプを持つ`graphs`プロパティを返します。 `count` をサポートします。
selectHosts	query	アイテムプロトタイプが属するホストの配列を持つ`hosts`プロパティを返します。
selectTags	query	アイテムプロトタイプのタグを`tags`プロパティで返します。
selectTriggers	query	アイテムプロトタイプが使用されているトリガープロトタイプを持つ`triggers`プロパティを返します。 `count` をサポートします。
selectPreprocessing	query	アイテムプロトタイプの前処理オプションを持つ`preprocessing`プロパティを返します。
selectValueMap	query	アイテムプロトタイプの値のマップを持つ`valuemap`プロパティを返します。
filter	object	指定したフィルタに完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる値または値の配列です。 `text`データ型のプロパティはサポートしません。 追加プロパティをサポートします: `host` - アイテムプロトタイプが属するホストの技術名。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectGraphs` - 結果は `name` でソートされます; `selectTriggers` - 結果は `description` でソートされます。

パラメータ	型	説明
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからアイテムプロトタイプを取得する

特定の LLD ルール ID のすべてのアイテムプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
```

```
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "2",
        "interfaceid": "0",
        "description": "",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0",
        "uuid": "",
        "parameters": []
    },
    {
        "itemid": "10010",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10001",
        "name": "Processor load (1 min average per core)",
        "key_": "system.cpu.load[percpu,avg1]",
        "delay": "1m",
        "history": "1w",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "2",
        "interfaceid": "0",
        "description": "The processor load is calculated as system CPU load divided by number of CPU c
        "evaltype": "0",
        "jmx_endpoint": "",
```

```
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


従属アイテムの検索

特定のアイテム ID に対して 1 つの従属アイテムを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25547",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Seconds",
            "key_": "apache.status.uptime.seconds",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
```

```
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "evaltype": "0",
            "master_itemid": "25545",
            "jmx_endpoint": "",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


HTTP エージェントアイテムプロトタイプの検索

特定のホスト ID に対して、リクエストメソッド HEAD の HTTP エージェントアイテムプロトタイプを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28257",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10254",
            "name": "discovered",
            "key_": "item[{#INAME}]",
            "delay": "{#IUPDATE}",
            "history": "90d",
            "trends": "30d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
            "status_codes": "",
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


参照

- Host
- Graph prototype

- Trigger prototype

ソース

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

説明

`object itemprototype.update(object/array itemPrototypes)`

このメソッドは、既存のアイテムプロトタイプを更新するためのものです。

> **Note:**
> このメソッドは *Admin* および *Super admin* ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するアイテムプロトタイプのプロパティ。

各アイテムプロトタイプには `itemid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のアイテムプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
preprocessing	array	現在の前処理オプションを置き換えるアイテムプロトタイプの前処理オプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用
tags	array	アイテムプロトタイプのタグ。

戻り値

(object) `itemids` プロパティの下で更新されたアイテムプロトタイプの ID を含むオブジェクトを返します。

例

アイテムプロトタイプのインターフェースの変更

検出されたアイテムで使用するホストインターフェースを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

従属アイテムプロトタイプの更新

新しいマスターアイテムプロトタイプ ID で従属アイテムプロトタイプを更新します。同じホスト (テンプレート/ディスカバリールール) でのみ依存関係が許可されるため、マスターと従属アイテムは同じ hostid および ruleid を持つ必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムプロトタイプの更新

クエリフィールドを変更し、すべてのカスタムヘッダーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

アイテムの事前処理オプションの更新

アイテムプロトタイプをアイテム事前処理ルール「カスタム乗数」で更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

スクリプトアイテムプロトタイプの更新

異なるスクリプトでスクリプトアイテムプロトタイプを更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

LLD rule

このクラスは、ローレベルディスカバリルールを操作するためのものです。

オブジェクトリファレンス:

- LLD rule
  - HTTP header
  - HTTP query field
- LLD rule filter
  - LLD rule filter condition
- LLD macro path
- LLD rule preprocessing
- LLD rule overrides
  - LLD rule override filter
    - LLD rule override filter condition
  - LLD rule override operation
    - LLD rule override operation status
    - LLD rule override operation discover
    - LLD rule override operation period
    - LLD rule override operation history
    - LLD rule override operation trends
    - LLD rule override operation severity
    - LLD rule override operation tag
    - LLD rule override operation template
    - LLD rule override operation inventory

利用可能なメソッド:

- discoveryrule.create - 新しい LLD ルールを作成する
- discoveryrule.delete - LLD ルールを削除する
- discoveryrule.get - LLD ルールを取得する
- discoveryrule.update - LLD ルールを更新する

LLD rule オブジェクト

次のオブジェクトは、`discoveryruleAPI` に直接関連しています。

LLD ルール

低レベルディスカバリールールオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	LLD ルールの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
delay	string	LLD ルールの更新間隔。 秒またはサフィックス付きの時間単位（例：30s、1m、2h、1d）、およびオプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされません。柔軟な間隔はスラッシュで区切った 2 つのマクロとして記述できます（例：`{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例： `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が”Zabbix agent”(0)、”Simple check”(3)、”Zabbix internal”(5)、”External check”(10)、”Database monitor”(11)、”IPMI agent”(12)、”SSH agent”(13)、”TELNET agent”(14)、”JMX agent”(16)、”HTTP agent”(19)、”SNMP agent”(20)、”Script”(21)、”Browser”(22) の場合、または `type` が”Zabbix agent (active)”(7) で `key_` に”mqtt.get” が含まれていない場合は必須

プロパティ	型	説明
hostid	ID	LLD ルールが属するホストの ID。 プロパティの動作: - 定数 - 作成操作時に必須
flags	integer	ディスカバリールールの由来。 可能な値： 1 - 低レベルディスカバリールール; 5 - プロトタイプから変換された低レベルディスカバリールール。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールのホストインターフェースの ID。 プロパティの動作: - LLD ルールがホストに属し、type が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP agent” の場合は必須 - LLD ルールがホストに属し、type が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合はサポート
key_	string	LLD ルールのキー。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
name	string	LLD ルールの名前。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
type	integer	LLD ルールのタイプ。 可能な値： 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested。このタイプは、ルールの親がテンプレート（ディスカバリでホストプロトタイプにリンクされていると想定）またはディスカバリされたホスト（ルートに LLD ルールがある）である場合のみ許可されます。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポート。 プロパティの動作: - type が”HTTP agent” の場合は必須 - 継承オブジェクトの場合は読み取り専用
allow_traps	integer	trapper アイテムと同様に値の受信を許可するかどうか。 可能な値： 0 - *(デフォルト)* 受信データの受け入れを許可しない; 1 - 受信データの受け入れを許可する。 プロパティの動作: - type が”HTTP agent” の場合はサポート
authtype	integer	認証方式。 type が”SSH agent” の場合の可能な値： 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” の場合の可能な値： 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM。 プロパティの動作: - type が”SSH agent” または”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（type が”HTTP agent” の場合）
description	string	LLD ルールの説明。
error	string	LLD ルール値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトを追従するかどうか。 可能な値： 0 - リダイレクトを追従しない; 1 - *(デフォルト)* リダイレクトを追従する。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が”IPMI agent” で key_ が”ipmi.get” でない場合は必須 - type が”IPMI agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト： service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が”JMX agent” の場合はサポート
lifetime	string	ディスカバリされなくなったアイテムが削除されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを指定できます。 デフォルト：7d。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 可能な値： 0 - *(デフォルト)* 有効期間のしきい値に達したら削除; 1 - 削除しない; 2 - 即時削除。
enabled_lifetime	string	ディスカバリされなくなったアイテムが無効化されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを指定できます。 デフォルト：0。
enabled_lifetime_type	integer	失われた LLD リソースを無効化するシナリオ。 可能な値： 0 - 有効期間のしきい値に達したら無効化; 1 - 無効化しない; 2 - *(デフォルト)* 即時無効化。
master_itemid	ID	マスターアイテムの ID。 ディスカバリールールは他のディスカバリールールのマスターアイテムにはなれません。 プロパティの動作: - type が”Dependent item” の場合は必須 - 継承オブジェクトの場合は読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 可能な値： 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
params	string	LLD ルールのタイプに応じた追加パラメータ： - SSH および Telnet LLD ルールの場合は実行スクリプト; - データベースモニタ LLD ルールの場合は SQL クエリ; - 計算 LLD ルールの場合は式; - スクリプトおよびブラウザ LLD ルールの場合はスクリプト。 プロパティの動作: - type が”Database monitor”、”SSH agent”、”TELNET agent”、”Script”、”Browser” の場合は必須 - 継承オブジェクトの場合は読み取り専用（type が”Script” または”Browser” の場合）
parameters	object/array	type が”Script” または”Browser” の場合の追加パラメータ。 name と value プロパティを持つオブジェクトの配列で、name は一意でなければなりません。 プロパティの動作: - type が”Script” または”Browser” の場合はサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
password	string	認証用パスワード。 プロパティの動作: - type が”JMX agent” で username が設定されている場合は必須 - type が”Simple check”、”Database monitor”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（type が”HTTP agent” の場合）
post_type	integer	posts プロパティに格納されるポストデータボディのタイプ。 可能な値： 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
posts	string	HTTP(S) リクエストボディデータ。 プロパティの動作: - type が”HTTP agent” で post_type が”JSON データ” または”XML データ” の場合は必須 - type が”HTTP agent” で post_type が” 生データ” の場合はサポート - 継承オブジェクトの場合は読み取り専用
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - type が”SSH agent” で authtype が”public key” の場合は必須
publickey	string	公開鍵ファイル名。 プロパティの動作: - type が”SSH agent” で authtype が”public key” の場合は必須
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
request_method	integer	リクエストメソッドのタイプ。 可能な値： 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 request_method が”GET”、”POST”、”PUT” の場合の可能な値： 0 - *(デフォルト)* ボディ; 1 - ヘッダー; 2 - ボディとヘッダーの両方を保存。 request_method が”HEAD” の場合の可能な値： 1 - ヘッダー。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
snmp_oid	string	SNMP OID。 プロパティの動作: - type が”SNMP agent” の場合は必須 - 継承オブジェクトの場合は読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_password	string	SSL キーのパスワード。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
state	integer	LLD ルールの状態。 可能な値： 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールのステータス。 可能な値： 0 - *(デフォルト)* 有効な LLD ルール; 1 - 無効な LLD ルール。
status_codes	string	カンマ区切りの必要な HTTP ステータスコードの範囲。カンマ区切りリストの一部としてユーザーマクロもサポート。 例：200,200-{$M},{$M},200-400 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
templateid	ID	親テンプレート LLD ルールの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒またはサフィックス付きの時間単位（例：30s、1m）を指定できます。 ユーザーマクロも使用可能。 可能な値の範囲：1-600s。 デフォルト：”” - プロキシ/グローバル設定を使用。 プロパティの動作: - type が”Zabbix agent”(0)、”Simple check”(3) で key_ が”vmware.” および”icmpping” で始まらない場合、”Zabbix agent (active)”(7)、”External check”(10)、”Database monitor”(11)、”SSH agent”(13)、”TELNET agent”(14)、”HTTP agent”(19)、”SNMP agent”(20) で snmp_oid が”walk[” または”get[” で始まる場合、”Script”(21)、”Browser”(22) の場合はサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が”Zabbix trapper”、または type が”HTTP agent” で allow_traps が” 受信データの受け入れを許可する” の場合はサポート
username	string	認証用ユーザー名。 プロパティの動作: - type が”SSH agent”、”TELNET agent” の場合、または type が”JMX agent” で password が設定されている場合は必須 - type が”Simple check”、”Database monitor”、”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（type が”HTTP agent” の場合）
uuid	string	インポートされた LLD ルールを既存のものにリンクするためのユニバーサル一意識別子。指定しない場合は自動生成。 プロパティの動作: - LLD ルールがテンプレートに属する場合はサポート
verify_host	integer	接続先のホスト名が証明書のホスト名と一致するか検証するかどうか。 可能な値： 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
verify_peer	integer	ホストの証明書が正当かどうか検証するかどうか。 可能な値： 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用

HTTP header

ヘッダーオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	HTTP ヘッダー名 プロパティの動作: - 必須
value	文字列	ヘッダー値 プロパティの動作: - 必須

HTTP query field

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	パラメータ名 プロパティの動作: - 必須

プロパティ	タイプ	説明
value	文字列	パラメータ値 プロパティの動作: - 必須

LLD ルールフィルター

LLD ルールフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる条件のセットを定義します。以下のプロパティがあります:

プロパティ	タイプ	説明
conditions	object/array	結果をフィルタリングするために使用するフィルター条件のセット。条件は式内での配置順にソートされます。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	フィルター条件を評価するために使用される生成された式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用
formula	string	カスタム式のフィルター条件を評価するために使用されるユーザー定義の式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、フィルター条件で定義されたものと完全に一致している必要があります: 未使用または省略された条件はありません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

LLD ルールフィルター条件

LLD ルールフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります:

プロパティ	タイプ	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は必須

プロパティ	タイプ	説明
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。大文字のみ使用可能。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後条件を取得する際には新たに生成されます。 プロパティの動作: - LLD ルールフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには、discoveryrule.getおよびdiscoveryrule.createメソッドページの例を参照してください。

LLD マクロパス

LLD マクロパスには以下のプロパティがあります。

プロパティ	タイプ	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクタ。 プロパティの動作: - 必須

LLD ルールのプリプロセス

LLD ルールのプリプロセスオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	プリプロセスオプションのタイプ。 設定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーをチェック; 17 - XML のエラーをチェック; 20 - ハートビート付きで変更なしを破棄; 21 - JavaScript; 23 - Prometheus から JSON へ; 24 - CSV から JSON へ; 25 - 置換; 27 - XML から JSON へ; 28 - SNMP walk 値; 29 - SNMP walk から JSON へ; 30 - SNMP get 値。 プロパティの動作: - 必須

プロパティ	型	説明
params	string	プリプロセスオプションで使用される追加パラメータ。複数のパラメータは改行 (\n) で区切ります。 プロパティの動作: - `type` が” 正規表現”(5)、”XML XPath”(11)、”JSONPath”(12)、” 正規表現に一致”(14)、” 正規表現に一致しない”(15)、”JSON のエラーをチェック”(16)、”XML のエラーをチェック”(17)、” ハートビート付きで変更なしを破棄”(20)、”JavaScript”(21)、”Prometheus から JSON へ”(23)、”CSV から JSON へ”(24)、” 置換”(25)、”SNMP walk 値”(28)、”SNMP walk から JSON へ”(29)、または”SNMP get 値”(30) に設定されている場合は必須
error_handler	integer	プリプロセスステップが失敗した場合に使用されるアクションタイプ。 設定可能な値: 0 - Zabbix サーバーによるエラーメッセージの設定; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。
error_handler_params	string	プロパティの動作: - `type` が” 正規表現”(5)、”XML XPath”(11)、”JSONPath”(12)、” 正規表現に一致”(14)、” 正規表現に一致しない”(15)、”JSON のエラーをチェック”(16)、”XML のエラーをチェック”(17)、”Prometheus から JSON へ”(23)、”CSV から JSON へ”(24)、”XML から JSON へ”(27)、”SNMP walk 値”(28)、”SNMP walk から JSON へ”(29)、または”SNMP get 値”(30) に設定されている場合は必須 エラーハンドラのパラメータ。 プロパティの動作: - `error_handler` が” カスタム値を設定” または” カスタムエラーメッセージを設定” に設定されている場合は必須

各プリプロセスタイプでサポートされるパラメータとエラーハンドラは以下の通りです。

プリプロセスタイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
5	正規表現	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	正規表現に一致	pattern[1]			0, 1, 2, 3
15	正規表現に一致しない	pattern[1]			0, 1, 2, 3
16	JSON のエラーをチェック	path[3]			0, 1, 2, 3
17	XML のエラーをチェック	path[3]			0, 1, 2, 3

プリプロセスタイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
20	ハートビート付きで変更なしを破棄	seconds[4, 5]			
21	JavaScript	script[2]			
23	Prometheus から JSON へ	pattern[5, 6]			0, 1, 2, 3
24	CSV から JSON へ	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON へ				0, 1, 2, 3
28	SNMP walk 値	OID[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON へ[7]	フィールド名[2]	OID プレフィックス[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	フォーマット: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 正規表現
[2] 文字列
[3] JSONPath または XML XPath
[4] 正の整数 (時間のサフィックスをサポート、例: 30s, 1m, 2h, 1d)
[5] ユーザーマクロ
[6] Prometheus パターンは、`<metric name>{<label name>="<label value>", ...} == <value>` の構文に従います。各 Prometheus パターン要素 (メトリック、ラベル名、ラベル値、メトリック値) はユーザーマクロにできます。
[7] 複数の” フィールド名,OID プレフィックス, フォーマット” レコードを改行で区切って指定できます。

LLD ルールのオーバーライド

LLD ルールのオーバーライドオブジェクトは、さまざまなプロトタイプオブジェクトのプロパティをオーバーライドするために使用されるルール（フィルター、条件、操作）のセットを定義します。以下のプロパティがあります:

プロパティ	タイプ	説明
name	string	一意のオーバーライド名。 プロパティの動作: - 必須
step	integer	オーバーライドの一意の順序番号。 プロパティの動作: - 必須
stop	integer	一致した場合、次のオーバーライドの処理を停止するかどうか。 可能な値: 0 - *(デフォルト)* オーバーライドの処理を停止しない; 1 - フィルターが一致した場合、オーバーライドの処理を停止する。
filter	object	オーバーライドフィルター。
operations	object/array	オーバーライド操作。

LLD ルールのオーバーライドフィルター

LLD ルールのオーバーライドフィルターオブジェクトは、検出されたオブジェクトに一致した場合にオーバーライドが適用される条件のセットを定義します。以下のプロパティがあります:

プロパティ	タイプ	説明
conditions	object/array	検出されたオブジェクトに一致させるために使用するオーバーライドフィルター条件のセット。条件は式内での配置順に並べ替えられます。 プロパティの動作: - 必須
evaltype	integer	オーバーライドフィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成された式。式には、`formulaid` で特定のオーバーライドフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用
formula	string	カスタム式のオーバーライドフィルター条件の評価に使用するユーザー定義の式。式には、`formulaid` で特定のオーバーライドフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、オーバーライドフィルター条件で定義されたものと完全に一致している必要があります: いかなる条件も未使用または省略されてはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

LLD ルールのオーバーライドフィルター条件

LLD ルールのオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して個別のチェックを実行するためのものです。以下のプロパティがあります:

プロパティ	タイプ	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみ使用可能。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後リクエストする際には新たに生成されます。 プロパティの動作: - LLD ルールのオーバーライドフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須
operator	integer	条件演算子。 可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールのオーバーライド操作

LLD ルールのオーバーライド操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。以下のプロパティを持ちます:

プロパティ	タイプ	説明
operationobject	integer	アクションを実行する検出オブジェクトのタイプ。 可能な値: 0 - アイテムプロトタイプ; 1 - トリガープロトタイプ; 2 - グラフプロトタイプ; 3 - ホストプロトタイプ; 4 - ディスカバリプロトタイプ。 プロパティの動作: - 必須
operator	integer	オーバーライド条件の演算子。 可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - マッチする; 9 - マッチしない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストプロトタイプ名にマッチするパターン。
opstatus	object	アイテム、トリガー、ホストプロトタイプオブジェクトのオーバーライド操作ステータスオブジェクト。
opdiscover	object	オーバーライド操作の検出ステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムプロトタイプオブジェクトのオーバーライド操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムプロトタイプオブジェクトのオーバーライド操作履歴オブジェクト。

プロパティ	タイプ	説明
optrends	object	アイテムプロトタイプオブジェクトのオーバーライド操作トレンドオブジェクト。
opseverity	object	トリガープロトタイプオブジェクトのオーバーライド操作重大度オブジェクト。
optag	object/array	トリガーおよびホストプロトタイプオブジェクトのオーバーライド操作タグオブジェクト。
optemplate	object/array	ホストプロトタイプオブジェクトのオーバーライド操作テンプレートオブジェクト。
opinventory	object	ホストプロトタイプオブジェクトのオーバーライド操作インベントリオブジェクト。

LLD ルールのオーバーライド操作ステータス

LLD ルールのオーバーライド操作ステータスは、検出されたオブジェクトに設定されます。以下のプロパティがあります:

プロパティ	タイプ	説明
status	integer	選択したオブジェクトのステータスをオーバーライドします。 可能な値: 0 - 有効で作成; 1 - 無効で作成。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作のディスカバー

LLD ルールのオーバーライド操作で、検出されたオブジェクトに設定されるディスカバーステータスです。以下のプロパティがあります。

プロパティ	タイプ	説明
discover	integer	選択したオブジェクトのディスカバーステータスをオーバーライドします。 可能な値： 0 - はい、オブジェクトのディスカバリーを継続する; 1 - いいえ、新しいオブジェクトはディスカバリーされず、既存のオブジェクトはロストとしてマークされる。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作期間

LLD ルールのオーバーライド操作期間は、検出されたアイテムに設定される更新間隔値です。以下のプロパティがあります。

プロパティ	タイプ	説明
delay	string	アイテムプロトタイプの更新間隔を上書きします。 秒またはサフィックス付きの時間単位（例：30s、1m、2h、1d）、およびオプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロまたは LLD マクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされていません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例：`{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例： `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - 必須

LLD ルールのオーバーライド操作履歴

検出されたアイテムに設定される LLD ルールのオーバーライド操作履歴値です。以下のプロパティがあります:

プロパティ	タイプ	説明
history	string	アイテムプロトタイプの履歴をオーバーライドします。これは履歴データを保存する期間の時間単位です。ユーザーマクロや LLD マクロも受け付けます。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作のトレンド

LLD ルールのオーバーライド操作のトレンド値は、ディスカバリされたアイテムに設定されます。以下のプロパティがあります。

プロパティ	タイプ	説明
trends	string	トレンドデータを保存する期間の単位であるアイテムプロトタイプのトレンドをオーバーライドします。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作の深刻度

LLD ルールのオーバーライド操作の深刻度の値は、検出されたトリガーに設定されます。以下のプロパティがあります:

プロパティ	タイプ	説明
severity	integer	トリガープロトタイプの深刻度をオーバーライドします。 可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 重度障害; 5 - 致命的障害。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作タグ

LLD ルールのオーバーライド操作タグオブジェクトには、検出されたオブジェクトに設定されるタグ名と値が含まれます。以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	新しいタグ名。 プロパティの動作: - 必須
value	string	新しいタグ値。

LLD ルールのオーバーライド操作テンプレート

検出されたホストにリンクされている LLD ルールのオーバーライド操作テンプレートオブジェクトです。以下のプロパティがあります。

プロパティ	タイプ	説明
templateid	ID	ホストプロトタイプにリンクされたテンプレートのテンプレートをオーバーライドします。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作インベントリ

検出されたホストに設定される LLD ルールのオーバーライド操作インベントリモード値。以下のプロパティがあります:

プロパティ	タイプ	説明
inventory_mode	integer	ホストプロトタイプのインベントリモードをオーバーライドします。 可能な値: -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 プロパティの動作: - 必須

discoveryrule.create

説明

`object discoveryrule.create(object/array lldRules)`

このメソッドは、新しい LLD ルールを作成することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 作成する LLD ルール。

標準 LLD ルールプロパティに加えて、メソッドは次のパラメーターをサポートします。

パラメーター	タイプ	説明
filter	object	LLD ルールのLLD ルールフィルターオブジェクト。
preprocessing	object/array	LLD ルール事前処理オプション。
lld_macro_paths	object/array	LLD ルールlld_macro_pathオプション。
overrides	object/array	LLD ルールオーバーライドオプション。

戻り値

(object) itemids プロパティの下で作成された LLD ルールの ID を含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールの順序と一致します。

例

LLD ルールの作成

マウントされたファイルシステムを検出するための Zabbix エージェント LLD ルールを作成します。検出されたアイテムは 30 秒ごとに更新されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Mounted filesystem discovery",
        "key_": "vfs.fs.discovery",
        "hostid": "10197",
        "type": 0,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

フィルターの使用

条件のセットを持つ LLD ルールを作成し、結果をフィルタリングします。条件は論理演算子「and」でまとめられます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
```

```
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

LLD マクロパスを持つ LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD rule with LLD macro paths",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
            {
```

```
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

条件を評価するためにカスタム式を使用するフィルターを持つ LLD ルールを作成します。LLD ルールは、”{#MACRO1}” マクロ値が正規表現”regex1” および”regex2” の両方に一致し、かつ”{#MACRO2}” の値が”regex3” または”regex4” のいずれかに一致するオブジェクトのみを検出する必要があります。式 ID”A”、”B”、”C”、”D” は任意に選択されています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
        }
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタムクエリフィールドとヘッダーの使用

カスタムクエリフィールドとヘッダーを持つ LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
            {
                "name": "X-Type",
                "value": "api"
            },
            {
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
```

```
    },
    "id": 1
}
```

前処理付き LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discovery rule with preprocessing",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

オーバーライド付き LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discover database host",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Discover MySQL host",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
                    "conditions": [
                        {
```

```
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mysqld\\.service$"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ],
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ]
                }
            ]
        },
        {
            "name": "Discover PostgreSQL host",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10263"
                        }
                    ],
                    "optag": [
                        {
```

```
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

スクリプト LLD ルールの作成

スクリプト LLD ルールを使用してシンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.lldrule",
        "hostid": "12345",
        "type": 21,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [{
            "name": "host",
            "value": "{HOST.CONN}"
        }],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

無効化期間を指定し、削除を行わない LLD ルールを作成します。

エンティティが検出されなくなった後に無効化する期間をカスタマイズし、削除を行わないように設定した LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "lld disable after 1h",
        "key_": "lld.disable",
        "hostid": "10001",
        "type": 2,
        "lifetime_type": 1,
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "1h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

参照

- LLD rule filter
- LLD macro paths
- LLD rule preprocessing

ソース

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

説明

`object discoveryrule.delete(array lldRuleIds)`

このメソッドは、LLD ルールを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除する LLD ルールの ID。

戻り値

(`object`) `ruleids` プロパティの下にある削除された LLD ルールの ID を含むオブジェクトを返します。

例

複数の LLD ルールの削除

2 つの LLD ルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
```

```
        "27665",
        "27668"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

説明

`integer/array discoveryrule.get(object parameters)`

このメソッドは、指定されたパラメータに従って LLD ルールを取得することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
itemids	ID/array	指定した ID の LLD ルールのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承された LLD ルールのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効な LLD ルールのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属する LLD ルールのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールのみを返します。
selectDiscoveryData	query	LLD ルールディスカバリーオブジェクトデータを持つ `discoveryData` プロパティを返します。LLD ルールディスカバリーオブジェクトは、検出された LLD ルールを、そのルールが検出された LLD ルールプロトタイプにリンクします。
selectDiscoveryRulePrototypes	query	LLD ルールに属する LLD ルールプロトタイプを持つ`discoveryRulePrototypes`プロパティを返します。 `count` をサポートします。
selectFilter	query	LLD ルールで使用されるフィルターのデータを持つ`filter`プロパティを返します。

パラメーター	型	説明
selectGraphs	query	LLD ルールに属するグラフプロトタイプを持つgraphsプロパティを返します。 count をサポートします。
selectHostPrototypes	query	LLD ルールに属するホストプロトタイプを持つhostPrototypesプロパティを返します。 count をサポートします。
selectHosts	query	LLD ルールが属するホストの配列を持つhostsプロパティを返します。
selectItems	query	LLD ルールに属するアイテムプロトタイプを持つitemsプロパティを返します。 count をサポートします。
selectTriggers	query	LLD ルールに属するトリガープロトタイプを持つtriggersプロパティを返します。 count をサポートします。
selectLLDMacroPaths	query	各対応するマクロに割り当てられた値へのパスと LLD マクロのリストを持つlld_macro_pathsプロパティを返します。
selectPreprocessing	query	LLD ルールの前処理オプションを持つpreprocessingプロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行されるオーバーライドフィルター、条件、操作のリストを持つlld_rule_overridesプロパティを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 textデータ型のプロパティはサポートしません。 追加プロパティをサポートします: host - LLD ルールが属するホストの技術名。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectItems, selectGraphs, selectTriggers。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: itemid, name, key_, delay, type, status。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ホストからのディスカバリルールの取得

特定のホスト ID のすべてのディスカバリルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Network interface discovery",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of network interfaces as defined in global regular expression \"Netw
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
```

```
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        },
        {
            "itemid": "27426",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Mounted filesystem discovery",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of file systems of different types as defined in global regular expr
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


フィルター条件の取得

LLD ルール”24681” の名前とそのフィルター条件を取得します。フィルターは”and” 評価タイプを使用しているため、formula プロ

パティは空で、eval_formula は自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "24681",
            "name": "Filtered LLD rule",
            "filter": {
                "evaltype": "1",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#MACRO1}",
                        "value": "@regex1",
                        "operator": "8",
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#MACRO2}",
                        "value": "@regex2",
                        "operator": "9",
                        "formulaid": "B"
                    },
                    {
                        "macro": "{#MACRO3}",
                        "value": "",
                        "operator": "12",
                        "formulaid": "C"
                    },
                    {
                        "macro": "{#MACRO4}",
                        "value": "",
                        "operator": "13",
                        "formulaid": "D"
                    }
                ],
                "eval_formula": "A and B and C and D"
            }
        }
    ],
    "id": 1
}
```

URL による LLD ルールの取得

ルールの URL フィールド値でホストの LLD ルールを取得します。LLD ルールに定義された URL 文字列の完全一致のみがサポートされます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28336",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10257",
            "name": "API HTTP agent",
            "key_": "api_discovery_rule",
            "delay": "5s",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
```

```
                    "value": "api"
                },
                {
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```

オーバーライド付きの LLD ルールを取得

さまざまなオーバーライド設定を持つ 1 つの LLD ルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
```

```
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$",
                        "formulaid": "B"
                    }
                ],
                "eval_formula": "A or B"
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ],
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ]
                }
            ]
        },
        {
            "name": "Discover PostgreSQL host",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optag": [
                        {
                            "tag": "database",
                            "value": "postgresql"
```

```
                            }
                        ],
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ]
                    }
                ]
            }
        ]
    }
],
"id": 1
}
```

参照

- Graph prototype
- Host
- Item prototype
- LLD rule filter
- Trigger prototype

ソース

ui/include/classes/api/services/CDiscoveryRule.php の CDiscoveryRule::get()。

discoveryrule.update

説明

`object discoveryrule.update(object/array lldRules)`

このメソッドは、既存の LLD ルールを更新します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新する LLD ルールのプロパティ。

各 LLD ルールには `itemid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準の LLD ルールプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルターを置き換えるLLD ルールフィルター。
preprocessing	object/array	既存の前処理オプションを置き換えるLLD ルール前処理オプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換える LLD ルールのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用
overrides	object/array	既存のオーバーライドオプションを置き換えるLLD ルールのオーバーライドオプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用

戻り値

(object) itemids プロパティの下で更新された LLD ルールの ID を含むオブジェクトを返します。

例

LLD ルールへのフィルターの追加

{#FSTYPE} マクロの内容が *@File systems for discovery* 正規表現に一致するようにフィルターを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

LLD マクロパスの追加

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
```

```
    },
    "id": 1
}
```

トラッピングの無効化

ディスカバリールールの LLD トラッピングを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

LLD ルールの事前処理オプションの更新

事前処理ルール「JSONPath」で LLD ルールを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.path.to.json",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

LLD ルールスクリプトの更新

LLD ルールスクリプトを別のスクリプトで更新し、以前のスクリプトで使用されていた不要なパラメーターを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

LLD ルールの有効期間の更新

LLD ルールを更新し、検出されなくなったエンティティを 12 時間後に無効化し、7 日後に削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

LLD ルールプロトタイプ

このクラスは、低レベルディスカバリールールプロトタイプで動作するように設計されています。

オブジェクト参照:

- LLD ルールプロトタイプ

    - HTTP ヘッダー
    - HTTP クエリフィールド
- LLD ルールプロトタイプフィルター
    - LLD ルールプロトタイプフィルター条件
- LLD プロトタイプマクロパス
- LLD ルールプロトタイプ前処理
- LLD ルールプロトタイプオーバーライド
    - LLD ルールプロトタイプオーバーライドフィルター
        * LLD ルールプロトタイプオーバーライドフィルター条件
    - LLD ルールプロトタイプオーバーライド操作
        * LLD ルールプロトタイプオーバーライド操作ステータス
        * LLD ルールプロトタイプオーバーライド操作ディスカバー
        * LLD ルールプロトタイプオーバーライド操作期間
        * LLD ルールプロトタイプオーバーライド操作履歴
        * LLD ルールプロトタイプオーバーライド操作トレンド
        * LLD ルールプロトタイプオーバーライド操作深刻度
        * LLD ルールプロトタイプオーバーライド操作タグ
        * LLD ルールプロトタイプオーバーライド操作テンプレート
        * LLD ルールプロトタイプオーバーライド操作インベントリ

利用可能なメソッド:

- discoveryruleprototype.create - 新しい LLD ルールプロトタイプの作成
- discoveryruleprototype.delete - LLD ルールプロトタイプの削除
- discoveryruleprototype.get - LLD ルールプロトタイプの取得
- discoveryruleprototype.update - LLD ルールプロトタイプの更新

LLD ルールプロトタイプオブジェクト

以下のオブジェクトは `discoveryruleprototype` API に直接関連しています。

LLD ルールプロトタイプ

低レベルディスカバリールールプロトタイプオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
ruleid	ID	親 LLD ルール/LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 作成操作時に必須

プロパティ	型	説明
delay	string	LLD ルールプロトタイプの更新間隔。 秒またはサフィックス付きの時間単位（例: 30s, 1m, 2h, 1d）、およびオプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は、柔軟な間隔とスケジューリング間隔の混在が可能です。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が”Zabbix agent”(0)、”Simple check”(3)、”Zabbix internal”(5)、”External check”(10)、”Database monitor”(11)、”IPMI agent”(12)、”SSH agent”(13)、”TELNET agent”(14)、”JMX agent”(16)、”HTTP agent”(19)、”SNMP agent”(20)、”Script”(21)、”Browser”(22) の場合、または `type` が”Zabbix agent (active)”(7) で `key_` に”mqtt.get” が含まれない場合は必須
hostid	ID	LLD ルールプロトタイプが属するホストの ID。 プロパティの動作: - 定数 - 作成操作時に必須
flags	integer	LLD ルールプロトタイプの由来。 可能な値: 3 - 低レベルディスカバリールールプロトタイプ; 7 - 検出された低レベルディスカバリールールプロトタイプ。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールプロトタイプのホストインターフェースの ID。 プロパティの動作: - LLD ルールプロトタイプがホストに属し、`type` が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP agent” の場合は必須 - LLD ルールプロトタイプがホストに属し、`type` が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合はサポート
key_	string	LLD ルールプロトタイプのキー。少なくとも 1 つの LLD マクロが必要です。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
name	string	LLD ルールプロトタイプの名前。 プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
type	integer	LLD ルールプロトタイプのタイプ。 可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. プロパティの動作: - 作成操作時に必須 - 継承オブジェクトの場合は読み取り専用
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が”HTTP agent” の場合は必須 - 継承オブジェクトの場合は読み取り専用
allow_traps	integer	trapper アイテムと同様に値の投入を許可するかどうか。 可能な値: 0 - *(デフォルト)* 受信データの受け入れを許可しない; 1 - 受信データの受け入れを許可する。 プロパティの動作: - type が”HTTP agent” の場合はサポート
authtype	integer	認証方式。 type が”SSH agent” の場合の可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” の場合の可能な値: 0 - *(デフォルト)* なし; 1 - ベーシック; 2 - NTLM。 プロパティの動作: - type が”SSH agent” または”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（type が”HTTP agent” の場合）
description	string	LLD ルールプロトタイプの説明。
error	string	LLD ルールプロトタイプの値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
follow_redirects	integer	データ取得時にレスポンスのリダイレクトを追従するかどうか。 可能な値: 0 - リダイレクトを追従しない; 1 - *(デフォルト)* リダイレクトを追従する。 プロパティの動作: - type が"HTTP agent" の場合はサポート - 継承オブジェクトの場合は読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が"HTTP agent" の場合はサポート - 継承オブジェクトの場合は読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が"HTTP agent" の場合はサポート - 継承オブジェクトの場合は読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が"IPMI agent" で key_ が"ipmi.get" でない場合は必須 - type が"IPMI agent" の場合はサポート - 継承オブジェクトの場合は読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が"JMX agent" の場合はサポート
lifetime	string	検出されなくなったアイテムが削除されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを指定できます。 デフォルト: 7d。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 可能な値: 0 - *(デフォルト)* 有効期限到達後に削除; 1 - 削除しない; 2 - 即時削除。
enabled_lifetime	string	検出されなくなったアイテムが無効化されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを指定できます。 デフォルト: 0。
enabled_lifetime_type	integer	失われた LLD リソースを無効化するシナリオ。 可能な値: 0 - 有効期限到達後に無効化; 1 - 無効化しない; 2 - *(デフォルト)* 即時無効化。
master_itemid	ID	マスターアイテムの ID。 ディスカバリールールは他のディスカバリールールのマスターアイテムにはなれません。 プロパティの動作: - type が"Dependent item" の場合は必須 - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
output_format	integer	レスポンスを JSON に変換するかどうか。 可能な値: 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
params	string	LLD ルールプロトタイプのタイプに応じた追加パラメータ: - SSH および Telnet LLD ルールの場合は実行スクリプト; - データベースモニタ LLD ルールの場合は SQL クエリ; - 計算 LLD ルールの場合は式; - スクリプトおよびブラウザ LLD ルールの場合はスクリプト。 プロパティの動作: - `type` が”Database monitor”、”SSH agent”、”TELNET agent”、”Script”、”Browser” の場合は必須 - 継承オブジェクトの場合は読み取り専用（`type` が”Script” または”Browser” の場合）
parameters	object/array	`type` が”Script” または”Browser” の場合の追加パラメータ。 `name` と `value` プロパティを持つオブジェクトの配列で、`name` は一意でなければなりません。 プロパティの動作: - `type` が”Script” または”Browser” の場合はサポート - 継承オブジェクトの場合は読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が”JMX agent” で `username` が設定されている場合は必須 - `type` が”Simple check”、”Database monitor”、”SSH agent”、”TELNET agent”、”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（`type` が”HTTP agent” の場合）
post_type	integer	`posts` プロパティに格納される POST データ本体のタイプ。 可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
posts	string	HTTP(S) リクエストボディデータ。 プロパティの動作: - `type` が”HTTP agent” で `post_type` が”JSON data” または”XML data” の場合は必須 - `type` が”HTTP agent” で `post_type` が”Raw data” の場合はサポート - 継承オブジェクトの場合は読み取り専用
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” で `authtype` が”public key” の場合は必須
publickey	string	公開鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” で `authtype` が”public key” の場合は必須

プロパティ	型	説明
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
request_method	integer	リクエストメソッドのタイプ。 可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、”PUT” の場合の可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” の場合の可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が”SNMP agent” の場合は必須 - 継承オブジェクトの場合は読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
ssl_key_password	string	SSL キーのパスワード。 プロパティの動作: - `type` が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
state	integer	LLD ルールプロトタイプの状態。 可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールプロトタイプのステータス。 可能な値: 0 - *(デフォルト)* 有効な LLD ルールプロトタイプ; 1 - 無効な LLD ルールプロトタイプ。

プロパティ	型	説明
status_codes	string	カンマ区切りの必要な HTTP ステータスコードの範囲。カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用
templateid	ID	親テンプレート LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒またはサフィックス付きの時間単位（例: 30s, 1m）を指定できます。 ユーザーマクロも使用可能です。 可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - type が”Zabbix agent”(0)、”Simple check”(3) で key_ が”vmware.” または”icmpping” で始まらない場合、”Zabbix agent (active)”(7)、”External check”(10)、”Database monitor”(11)、”SSH agent”(13)、”TELNET agent”(14)、”HTTP agent”(19)、”SNMP agent”(20) で snmp_oid が”walk[” または”get[” で始まる場合、”Script”(21)、”Browser”(22) の場合はサポート - 継承オブジェクトの場合は読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が”Zabbix trapper” の場合、または type が”HTTP agent” で allow_traps が” 受信データの受け入れを許可する” の場合はサポート
username	string	認証用ユーザー名。 プロパティの動作: - type が”SSH agent”、”TELNET agent” の場合、または type が”JMX agent” で password が設定されている場合は必須 - type が”Simple check”、”Database monitor”、”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用（type が”HTTP agent” の場合）
uuid	string	インポートされた LLD ルールプロトタイプを既存のものにリンクするためのユニバーサル一意識別子。指定しない場合は自動生成されます。 プロパティの動作: - LLD ルールプロトタイプがテンプレートに属する場合はサポート
verify_host	integer	接続先のホスト名が証明書のホスト名と一致するかどうかを検証するか。 可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が”HTTP agent” の場合はサポート - 継承オブジェクトの場合は読み取り専用

プロパティ	型	説明
verify_peer	integer	ホストの証明書が正当かどうかを検証するか。 可能な値: 0 - (デフォルト) 検証しない; 1 - 検証する。 プロパティの動作: - type が"HTTP agent" の場合はサポート - 継承オブジェクトの場合は読み取り専用

HTTP ヘッダー

header オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダーの値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

LLD ルールプロトタイプフィルター

LLD ルールプロトタイプフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる条件のセットを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
conditions	object/array	結果をフィルタリングするために使用するフィルター条件のセット。条件は式内での配置順にソートされます。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須

プロパティ	タイプ	説明
eval_formula	string	フィルター条件を評価するために使用される生成された式。式には、`formulaid` で特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用
formula	string	カスタム式のフィルター条件を評価するために使用するユーザー定義の式。式には、`formulaid` で特定のフィルター条件を参照する ID を含める必要があります。式で使用される ID は、フィルター条件で定義されたものと完全に一致する必要があります。未使用または省略された条件はありません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

LLD ルールプロトタイプのフィルター条件

LLD ルールプロトタイプのフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります。

プロパティ	型	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみを含めることができます。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後にリクエストする際には新たに生成されます。 プロパティの動作: - LLD ルールプロトタイプのフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須
operator	integer	条件演算子。 可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、discoveryruleprototype.getおよびdiscoveryruleprototype.createメソッドページの例を参照してください。

LLD マクロパス

LLD マクロパスには以下のプロパティがあります。

プロパティ	タイプ	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクタ。 プロパティの動作: - 必須

LLD ルールプロトタイプの事前処理

LLD ルールプロトタイプの事前処理オブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	事前処理オプションのタイプ。 指定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーチェック; 17 - XML のエラーチェック; 20 - ハートビート付きで変更なしを破棄; 21 - JavaScript; 23 - Prometheus から JSON へ; 24 - CSV から JSON へ; 25 - 置換; 27 - XML から JSON へ; 28 - SNMP walk 値; 29 - SNMP walk から JSON へ; 30 - SNMP get 値。 プロパティの動作: - 必須
params	string	事前処理オプションで使用される追加パラメータ。複数のパラメータは改行 (\n) で区切ります。 プロパティの動作: - type が” 正規表現”(5)、”XML XPath”(11)、”JSONPath”(12)、” 正規表現に一致”(14)、” 正規表現に一致しない”(15)、”JSON のエラーチェック”(16)、”XML のエラーチェック”(17)、” ハートビート付きで変更なしを破棄”(20)、”JavaScript”(21)、”Prometheus から JSON へ”(23)、”CSV から JSON へ”(24)、” 置換”(25)、”SNMP walk 値”(28)、”SNMP walk から JSON へ”(29)、または”SNMP get 値”(30) に設定されている場合は必須
error_handler	integer	事前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値: 0 - Zabbix サーバーによるエラーメッセージの設定; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が” 正規表現”(5)、”XML XPath”(11)、”JSONPath”(12)、” 正規表現に一致”(14)、” 正規表現に一致しない”(15)、”JSON のエラーチェック”(16)、”XML のエラーチェック”(17)、”Prometheus から JSON へ”(23)、”CSV から JSON へ”(24)、”XML から JSON へ”(27)、”SNMP walk 値”(28)、”SNMP walk から JSON へ”(29)、または”SNMP get 値”(30) に設定されている場合は必須

プロパティ	タイプ	説明
error_handler_params	string	エラーハンドラのパラメータ。 プロパティの動作: - `error_handler` が”カスタム値を設定”または”カスタムエラーメッセージを設定”に設定されている場合は必須

各事前処理タイプでサポートされているパラメータとエラーハンドラは以下の通りです。

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
5	正規表現	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	正規表現に一致	pattern[1]			0, 1, 2, 3
15	正規表現に一致しない	pattern[1]			0, 1, 2, 3
16	JSON のエラーチェック	path[3]			0, 1, 2, 3
17	XML のエラーチェック	path[3]			0, 1, 2, 3
20	ハートビート付きで変更なしを破棄	seconds[4, 5]			
21	JavaScript	script[2]			
23	Prometheus から JSON へ	pattern[5, 6]			0, 1, 2, 3
24	CSV から JSON へ	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON へ				0, 1, 2, 3

事前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラ
28	SNMP walk 値	OID[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON へ[7]	フィールド名[2]	OID プレフィックス[2]	フォーマット: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	フォーマット: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 正規表現
[2] 文字列
[3] JSONPath または XML XPath
[4] 正の整数 (時間のサフィックスをサポート、例: 30s, 1m, 2h, 1d)
[5] ユーザーマクロ
[6] Prometheus パターン。構文は `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターン要素 (メトリック、ラベル名、ラベル値、メトリック値) はユーザーマクロにできます。
[7] 複数の” フィールド名,OID プレフィックス, フォーマット” レコードを改行で区切って指定可能です。

LLD ルールプロトタイプのオーバーライド

LLD ルールプロトタイプのオーバーライドオブジェクトは、さまざまなプロトタイプオブジェクトのプロパティをオーバーライドするために使用されるルール（フィルター、条件、操作）のセットを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
name	string	一意のオーバーライド名。 プロパティの動作: - 必須
step	integer	オーバーライドの一意の順序番号。 プロパティの動作: - 必須
stop	integer	一致した場合、次のオーバーライドの処理を停止します。 可能な値: 0 - *(デフォルト)* オーバーライドの処理を停止しない; 1 - フィルターが一致した場合、オーバーライドの処理を停止する。
filter	object	オーバーライドフィルター。
operations	object/array	オーバーライド操作。

LLD ルールプロトタイプのオーバーライドフィルター

LLD ルールプロトタイプのオーバーライドフィルターオブジェクトは、検出されたオブジェクトに一致した場合にオーバーライドが適用される条件のセットを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
conditions	object/array	検出されたオブジェクトに一致させるために使用するオーバーライドフィルター条件のセット。条件は式内での配置順に並べ替えられます。 プロパティの動作: - 必須
eval_type	integer	オーバーライドフィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成された式。式には、`formulaid` で特定のオーバーライドフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式のフィルターの `formula` の値と等しくなります。 プロパティの動作: - 読み取り専用
formula	string	カスタム式のオーバーライドフィルターの条件評価に使用するユーザー定義の式。式には、`formulaid` で特定のオーバーライドフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、オーバーライドフィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

LLD ルールプロトタイプのオーバーライドフィルター条件

LLD ルールプロトタイプのオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して個別のチェックを実行するためのものです。以下のプロパティがあります。

プロパティ	型	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみ使用可能。フィルター条件を変更する際はユーザーが ID を定義する必要がありますが、その後にリクエストする際には新たに生成されます。 プロパティの動作: - LLD ルールプロトタイプのオーバーライドフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須

プロパティ	型	説明
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールプロトタイプのオーバーライド操作

LLD ルールプロトタイプのオーバーライド操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。
以下のプロパティがあります:

プロパティ	タイプ	説明
operationobject	integer	アクションを実行する検出オブジェクトのタイプ。 可能な値: 0 - アイテムプロトタイプ; 1 - トリガープロトタイプ; 2 - グラフプロトタイプ; 3 - ホストプロトタイプ。 プロパティの動作: - 必須
operator	integer	オーバーライド条件の演算子。 可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - マッチする; 9 - マッチしない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフまたはホストプロトタイプ名にマッチするパターン。
opstatus	object	アイテム、トリガー、ホストプロトタイプオブジェクトのオーバーライド操作ステータスオブジェクト。
opdiscover	object	オーバーライド操作検出ステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムプロトタイプオブジェクトのオーバーライド操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムプロトタイプオブジェクトのオーバーライド操作履歴オブジェクト。
optrends	object	アイテムプロトタイプオブジェクトのオーバーライド操作トレンドオブジェクト。
opseverity	object	トリガープロトタイプオブジェクトのオーバーライド操作重大度オブジェクト。
optag	object/array	トリガーおよびホストプロトタイプオブジェクトのオーバーライド操作タグオブジェクト。
optemplate	object/array	ホストプロトタイプオブジェクトのオーバーライド操作テンプレートオブジェクト。
opinventory	object	ホストプロトタイプオブジェクトのオーバーライド操作インベントリオブジェクト。

LLD ルールプロトタイプのオーバーライド操作ステータス

検出されたオブジェクトに設定される LLD ルールプロトタイプのオーバーライド操作ステータス。次のプロパティがあります。

プロパティ	タイプ	説明
status	integer	選択したオブジェクトのステータスをオーバーライドします。 可能な値： 0 - 有効として作成; 1 - 無効として作成。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作 discover

検出されたオブジェクトに設定される LLD ルールプロトタイプのオーバーライド操作 discover ステータス。以下のプロパティがあります。

プロパティ	タイプ	説明
discover	integer	選択したオブジェクトの discover ステータスをオーバーライドします。 可能な値: 0 - はい、オブジェクトの検出を継続します; 1 - いいえ、新しいオブジェクトは検出されず、既存のオブジェクトはロストとしてマークされます。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作期間

LLD ルールプロトタイプのオーバーライド操作期間は、検出されたアイテムに設定される更新間隔値です。以下のプロパティがあります:

プロパティ	タイプ	説明
delay	string	アイテムプロトタイプの更新間隔を上書きします。 秒またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）、およびオプションで 1 つ以上のカスタム間隔をセミコロンで区切って指定できます。カスタム間隔は、柔軟間隔とスケジューリング間隔の混在が可能です。 ユーザーマクロまたは LLD マクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロやテキストと混在したマクロはサポートされません。柔軟間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - 必須

LLD ルールプロトタイプの履歴の上書き

検出されたアイテムに設定される LLD ルールプロトタイプの履歴の上書き値。以下のプロパティがあります。

プロパティ	タイプ	説明
history	string	アイテムプロトタイプの履歴を上書きします。履歴データを保存する期間の時間単位です。ユーザーマクロや LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールプロトタイプのトレンドの上書き操作

LLD ルールプロトタイプのトレンドの上書き値は、検出されたアイテムに設定されます。次のプロパティがあります。

プロパティ	タイプ	説明
trends	string	トレンドデータを保存する期間の単位であるアイテムプロトタイプのトレンドを上書きします。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作の深刻度

検出されたトリガーに設定される LLD ルールプロトタイプのオーバーライド操作の深刻度値。以下のプロパティがあります。

プロパティ	タイプ	説明
severity	integer	トリガープロトタイプの深刻度を上書きします。 可能な値： 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作タグ

LLD ルールプロトタイプのオーバーライド操作タグオブジェクトには、検出されたオブジェクトに設定されるタグ名と値が含まれます。以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	新しいタグ名。 プロパティの動作: - 必須
value	string	新しいタグ値。

LLD ルールプロトタイプのオーバーライド操作テンプレート

LLD ルールプロトタイプのオーバーライド操作テンプレートオブジェクトは、検出されたホストにリンクされています。以下のプロパティがあります。

プロパティ	タイプ	説明
templateid	ID	ホストプロトタイプにリンクされたテンプレートのテンプレートをオーバーライドします。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作インベントリ

検出されたホストに設定される LLD ルールプロトタイプのオーバーライド操作インベントリモード値。以下のプロパティがあります。

プロパティ	タイプ	説明
inventory_mode	integer	ホストプロトタイプのインベントリモードを上書きします。 指定可能な値： -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 プロパティの動作: - 必須

discoveryruleprototype.create

説明

`object discoveryruleprototype.create(object/array lldRules)`

このメソッドは新しい LLD ルールプロトタイプを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object/array`) 作成する LLD ルールプロトタイプ。

標準の LLD ルールプロトタイププロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
filter	object	LLD ルールのLLD ルールプロトタイプフィルター。
preprocessing	object/array	LLD ルールプロトタイプの前処理オプション。
lld_macro_paths	object/array	LLD ルールプロトタイプのlld_macro_pathオプション。
overrides	object/array	LLD ルールプロトタイプのオーバーライドオプション。

戻り値

(`object`) 作成された LLD ルールプロトタイプの ID を `itemids` プロパティに含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールプロトタイプの順序と一致します。

例

LLD ルールプロトタイプの作成

データベースインスタンス内のテーブルスペースを検出するための LLD ルールプロトタイプ (Type: Nested) を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
```

```
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

参照

- LLD ルールプロトタイプフィルター
- LLD マクロパス
- LLD ルールプロトタイプの事前処理

ソース

CDiscoveryRulePrototype::create() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

discoveryruleprototype.delete

説明

`object discoveryruleprototype.delete(array lldRuleIds)`

このメソッドは、LLD ルールプロトタイプを削除するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(array) 削除する LLD ルールプロトタイプの ID。

戻り値

(object) 削除された LLD ルールプロトタイプの ID を `ruleids` プロパティで含むオブジェクトを返します。

例

複数の LLD ルールプロトタイプの削除

2 つの LLD ルールプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::delete() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

discoveryruleprototype.get

説明

`integer/array discoveryruleprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従って LLD ルールプロトタイプを取得することができます。

Note:
このメソッドは、すべての種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
discoveryids	ID/array	指定した LLD ルールまたは LLD ルールプロトタイプに属する LLD ルールプロトタイプのみを返します。
itemids	ID/array	指定した ID の LLD ルールプロトタイプのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールプロトタイプのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールプロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承された LLD ルールプロトタイプのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールプロトタイプのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効な LLD ルールプロトタイプのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属する LLD ルールプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールプロトタイプのみを返します。
selectDiscoveryRule	query	LLD ルールプロトタイプの親 LLD ルールを持つ`discoveryRule`プロパティを返します。 親がディスカバリールールプロトタイプの場合は空のレスポンスを返します。
selectDiscoveryRulePrototype	query	LLD ルールプロトタイプの親 LLD ルールプロトタイプを持つ`discoveryRulePrototype`プロパティを返します。 親がディスカバリールール（プロトタイプでない）の場合は空のレスポンスを返します。
selectDiscoveryRulePrototypes	query	LLD ルールプロトタイプに属する子 LLD ルールプロトタイプを持つ`discoveryRulePrototypes`プロパティを返します。 `count` をサポートします。

パラメーター	タイプ	説明
selectFilter	query	LLD ルールプロトタイプで使用されるフィルターのデータを持つfilterプロパティを返します。
selectGraphs	query	LLD ルールプロトタイプに属するグラフプロトタイプを持つgraphsプロパティを返します。 count をサポートします。
selectHostPrototypes	query	LLD ルールプロトタイプに属するホストプロトタイプを持つhostPrototypesプロパティを返します。 count をサポートします。
selectHosts	query	LLD ルールプロトタイプが属するホストの配列を持つhostsプロパティを返します。
selectItems	query	LLD ルールプロトタイプに属するアイテムプロトタイプを持つitemsプロパティを返します。 count をサポートします。
selectTriggers	query	LLD ルールプロトタイプに属するトリガープロトタイプを持つtriggersプロパティを返します。 count をサポートします。
selectLLDMacroPaths	query	各対応するマクロに割り当てられた値へのパスと LLD マクロのリストを持つlld_macro_pathsプロパティを返します。
selectPreprocessing	query	LLD ルールプロトタイプの前処理オプションを持つpreprocessingプロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行されるオーバーライドフィルター、条件、操作のリストを持つlld_rule_overridesプロパティを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 textデータ型のプロパティはサポートされません。 追加プロパティをサポートします： host - LLD ルールプロトタイプが属するホストの技術名。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます：selectItems、selectGraphs、selectTriggers。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値：itemid、name、key_、delay、type、status。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

（integer/array）は以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用された場合は、取得したオブジェクトの数

例

ホストからのディスカバリールールプロトタイプの取得

特定のホスト ID のすべてのディスカバリールールプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Discover tablespaces for {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
```

```
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


参照

- グラフプロトタイプ
- ホスト
- アイテムプロトタイプ
- LLD ルールプロトタイプフィルター
- トリガープロトタイプ

ソース

ui/include/classes/api/services/CDiscoveryRulePrototype.php の CDiscoveryRulePrototype::get()。

discoveryruleprototype.update

説明

`object discoveryruleprototype.update(object/array lldRules)`

このメソッドは、既存の LLD ルールプロトタイプを更新するためのものです。

すでに検出されたプロトタイプの更新には制限があることに注意してください。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 更新する LLD ルールプロトタイプのプロパティ。

各 LLD ルールプロトタイプには `itemid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準の LLD ルールプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルタを置き換えるLLD ルールプロトタイプフィルタ。
preprocessing	object/array	既存の前処理オプションを置き換えるLLD ルールプロトタイプ前処理オプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換える LLD ルールプロトタイプのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用

パラメータ	型	説明
overrides	object/array	既存のオーバーライドオプションを置き換えるLLD ルールプロトタイプ オーバーライドオプション。 パラメータの動作: - 継承されたオブジェクトの場合は読み取り専用

戻り値

(object) itemids プロパティの下に、更新された LLD ルールプロトタイプの ID を含むオブジェクトを返します。

例

LLD ルールプロトタイプの事前処理オプションの更新

LLD ルールプロトタイプに JSONPath の事前処理ルールを追加します。リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::update() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

Maintenance

このクラスは、メンテナンスを操作するためのものです。

オブジェクトリファレンス:

- Maintenance
- Time period
- Problem tag

利用可能なメソッド:

- maintenance.create - 新しいメンテナンスを作成する
- maintenance.delete - メンテナンスを削除する
- maintenance.get - メンテナンスを取得する
- maintenance.update - メンテナンスを更新する

Maintenance オブジェクト

次のオブジェクトは、`maintenance`API に直接関連しています。

メンテナンス

メンテナンスオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
maintenanceid	ID	メンテナンスの ID。 プロパティの動作: - 読み取り専用 - 更新操作には必須
name	string	メンテナンスの名前。 プロパティの動作: - 作成操作には必須
active_since	timestamp	メンテナンスが有効になる時刻（含む）。 指定された値は分単位に切り捨てられます。 プロパティの動作: - 作成操作には必須
active_till	timestamp	メンテナンスが無効になる時刻（含まない）。 指定された値は分単位に切り捨てられます。 プロパティの動作: - 作成操作には必須
description	string	メンテナンスの説明。
maintenance_type	integer	メンテナンスタイプ。 指定可能な値: 0 - *(デフォルト)* データ収集あり; 1 - データ収集なし。
tags_evaltype	integer	障害タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.

時間期間

時間期間オブジェクトは、メンテナンスが有効になる期間を定義するために使用されます。以下のプロパティがあります。

プロパティ	タイプ	説明
period	integer	メンテナンス期間の秒数。 指定された値は分単位に切り捨てられます。 デフォルト: 3600。
timeperiod_type	integer	時間期間のタイプ。 指定可能な値: 0 - *(デフォルト)* 1 回のみ; 2 - 毎日; 3 - 毎週; 4 - 毎月。

プロパティ	タイプ	説明
start_date	timestamp	メンテナンス期間が有効になる日付。 指定された値は分単位に切り捨てられます。 デフォルト: 現在の日付。 プロパティの動作: - `timeperiod_type` が「1 回のみ」に設定されている場合にサポート
start_time	integer	メンテナンスが開始される時刻（秒単位）。 指定された値は分単位に切り捨てられます。 デフォルト: 0。 プロパティの動作: - `timeperiod_type` が「毎日」、「毎週」または「毎月」に設定されている場合にサポート
every	integer	毎日および毎週の期間の場合、`every` はメンテナンスが有効になる日または週の間隔を定義します。 `timeperiod_type` が「毎日」または「毎週」に設定されている場合のデフォルト値: 1。 毎月の期間で `day` が設定されている場合、`every` プロパティはメンテナンスが有効になる月の日を定義します。 `timeperiod_type` が「毎月」に設定され、`day` が設定されている場合のデフォルト値: 1。 毎月の期間で `dayofweek` が設定されている場合、`every` プロパティはメンテナンスが有効になる月の週を定義します。 `timeperiod_type` が「毎月」に設定され、`dayofweek` が設定されている場合の指定可能な値: 1 - *(デフォルト)* 最初の週; 2 - 2 週目; 3 - 3 週目; 4 - 4 週目; 5 - 最終週。 プロパティの動作: - `timeperiod_type` が「毎日」、「毎週」または「毎月」に設定されている場合にサポート
dayofweek	integer	メンテナンスが有効になる曜日。 指定可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 これはビットマスクフィールドであり、指定可能なビットマップ値の任意の合計が許容されます（例: 月・水・金の場合は 21）。 プロパティの動作: - `timeperiod_type` が「毎週」に設定されている場合、または `timeperiod_type` が「毎月」に設定されていて `day` が設定されていない場合に必須
day	integer	メンテナンスが有効になる月の日。 プロパティの動作: - `timeperiod_type` が「毎月」に設定されていて `dayofweek` が設定されていない場合に必須

プロパティ	タイプ	説明
month	integer	メンテナンスが有効になる月。 指定可能なビットマップ値: 1 - 1 月; 2 - 2 月; 4 - 3 月; 8 - 4 月; 16 - 5 月; 32 - 6 月; 64 - 7 月; 128 - 8 月; 256 - 9 月; 512 - 10 月; 1024 - 11 月; 2048 - 12 月。 これはビットマスクフィールドであり、指定可能なビットマップ値の任意の合計が許容されます（例: 1 月・4 月・7 月・10 月の場合は 585）。 プロパティの動作: - `timeperiod_type` が「毎月」に設定されている場合に必須

問題タグ

問題タグオブジェクトは、メンテナンスが有効になったときに抑制する問題を定義するために使用されます。タグは、Maintenance オブジェクトの `maintenance_type` が「データ収集あり」に設定されている場合にのみ指定できます。以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	問題タグ名。 プロパティの動作: - 必須
operator	integer	条件演算子。 指定可能な値: 0 - 等しい; 2 - *(デフォルト)* 含む。
value	string	問題タグ値。

maintenance.create

説明

`object maintenance.create(object/array maintenances)`

このメソッドは、新しいメンテナンスを作成するためのものです。

Note:
このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するメンテナンス。

標準のメンテナンスプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	メンテナンスを実施するホストグループ。 ホストグループには groupid プロパティのみを定義する必要があります。 パラメータの動作: - hosts が設定されていない場合は必須
hosts	object/array	メンテナンスを実施するホスト。 ホストには hostid プロパティのみを定義する必要があります。 パラメータの動作: - groups が設定されていない場合は必須
timeperiods	object/array	メンテナンスの期間。 パラメータの動作: - 必須
tags	object/array	問題タグ。 抑制する問題を定義します。 タグが指定されていない場合、すべてのアクティブなメンテナンスホストの問題が抑制されます。 パラメータの動作: - Maintenance objectの maintenance_type が「データ収集あり」に設定されている場合にサポート

戻り値

(object) maintenanceids プロパティの下で作成されたメンテナンスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメンテナンスの順序と一致します。

例

メンテナンスの作成

ID "2" のホストグループに対して、**service:mysqld** および **error** の問題タグを持つデータ収集付きのメンテナンスを作成します。2013 年 1 月 22 日から 2014 年 1 月 22 日まで有効で、毎週日曜日の 18:00 に発効し、1 時間継続します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1358844540,
        "active_till": 1390466940,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "2"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
```

```
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Time period

ソース

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

説明

`object maintenance.delete(array maintenanceIds)`

このメソッドは、メンテナンス期間を削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除するメンテナンス期間の ID。

戻り値

(`object`) `maintenanceids` プロパティの下にある削除されたメンテナンス期間の ID を含むオブジェクトを返します。

例

複数のメンテナンス期間の削除

2 つのメンテナンス期間を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

説明

`integer/array maintenance.get(object parameters)`

このメソッドは、指定されたパラメータに従ってメンテナンスを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	タイプ	説明
groupids	ID/array	指定したホストグループに割り当てられているメンテナンスのみを返します。
hostids	ID/array	指定したホストに割り当てられているメンテナンスのみを返します。
maintenanceids	ID/array	指定した ID のメンテナンスのみを返します。
selectHostGroups	query	メンテナンスに割り当てられているホストグループを持つ`hostgroups`プロパティを返します。
selectHosts	query	メンテナンスに割り当てられているホストを持つ`hosts`プロパティを返します。
selectTags	query	メンテナンスの問題タグを持つ`tags`プロパティを返します。
selectTimeperiods	query	メンテナンスの時間帯を持つ`timeperiods`プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

パラメータ	タイプ	説明
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

メンテナンスの取得

すべての設定済みメンテナンス、および割り当てられたホストグループ、定義された時間帯、問題タグに関するデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
                {
                    "tag": "service",
```

```
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

参照

- Host
- Host group
- Time period

ソース

CMaintenance::get() in *ui/include/classes/api/services/CMaintenance.php*。

maintenance.update

説明

`object maintenance.update(object/array maintenances)`

このメソッドは、既存のメンテナンスを更新するためのものです。

Note:
このメソッドは、管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するメンテナンスのプロパティ。

各メンテナンスには maintenanceid プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他は変更されません。

標準のメンテナンスプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	現在のグループを置き換えるホストグループ。 ホストグループには groupid プロパティのみを定義する必要があります。 パラメータの動作: - hosts が設定されていない場合は必須
hosts	object/array	現在のホストを置き換えるホスト。 ホストには hostid プロパティのみを定義する必要があります。 パラメータの動作: - groups が設定されていない場合は必須
timeperiods	object/array	現在の期間を置き換えるメンテナンスの期間。
tags	object/array	現在のタグを置き換える問題タグ。 パラメータの動作: - メンテナンスオブジェクトの maintenance_type が「データ収集あり」に設定されている場合にサポート

戻り値

(object) maintenanceids プロパティの下で更新されたメンテナンスの ID を含むオブジェクトを返します。

例

異なるホストの割り当て

現在メンテナンスに割り当てられているホストを 2 つの異なるホストに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Time period

ソース

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

Map

このクラスは、マップを操作するためのものです。

オブジェクトリファレンス:

- Map
- Map element
  - Map element Host
  - Map element Host group
  - Map element Map
  - Map element Trigger
  - Map element Tag
  - Map element URL
- Map link
  - Map link trigger
  - Map link indicators
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

利用可能なメソッド:

- map.create - 新しいマップを作成する
- map.delete - マップを削除する
- map.get - マップを取得する
- map.update - マップを更新する

Map オブジェクト

次のオブジェクトは、mapAPI に直接関連しています。

マップ

マップオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
sysmapid	ID	マップの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
height	integer	マップの高さ (ピクセル単位)。 プロパティの動作: - 作成操作時に必須
name	string	マップの名前。 プロパティの動作: - 作成操作時に必須
width	integer	マップの幅 (ピクセル単位)。 プロパティの動作: - 作成操作時に必須
backgroundid	ID	マップの背景として使用される画像の ID。
background_scale	integer	背景画像の比率スケーリングを有効にするかどうか。 設定可能な値: 0 - スケーリングを無効化; 1 - *(デフォルト)* 画像のスケーリングを有効化。
expand_macros	integer	マップの設定時にラベル内のマクロを展開するかどうか。 設定可能な値: 0 - *(デフォルト)* マクロを展開しない; 1 - マクロを展開する。
expandproblem	integer	単一の問題を持つ要素に対して問題トリガーを表示するかどうか。 設定可能な値: 0 - 常に問題数を表示; 1 - *(デフォルト)* 問題が 1 つだけの場合は問題トリガーを表示。
grid_align	integer	グリッド揃えを有効にするかどうか。 設定可能な値: 0 - グリッド揃えを無効化; 1 - *(デフォルト)* グリッド揃えを有効化。
grid_show	integer	マップ上にグリッドを表示するかどうか。 設定可能な値: 0 - グリッドを表示しない; 1 - *(デフォルト)* グリッドを表示する。
grid_size	integer	マップグリッドのサイズ (ピクセル単位)。 サポートされる値: 20, 40, 50, 75, 100。 デフォルト: 50。

プロパティ	タイプ	説明
highlight	integer	アイコンのハイライト表示を有効にするかどうか。 設定可能な値: 0 - ハイライト表示を無効化; 1 - *(デフォルト)* ハイライト表示を有効化。
iconmapid	ID	マップで使用されるアイコンマップの ID。
label_format	integer	高度なラベルを有効にするかどうか。 設定可能な値: 0 - *(デフォルト)* 高度なラベルを無効化; 1 - 高度なラベルを有効化。
label_location	integer	マップ要素ラベルの位置。 設定可能な値: 0 - *(デフォルト)* 下; 1 - 左; 2 - 右; 3 - 上。
label_string_host	string	ホスト要素のカスタムラベル。 プロパティの動作: - `label_type_host` が”custom” の場合は必須
label_string_hostgroup	string	ホストグループ要素のカスタムラベル。 プロパティの動作: - `label_type_hostgroup` が”custom” の場合は必須
label_string_image	string	イメージ要素のカスタムラベル。 プロパティの動作: - `label_type_image` が”custom” の場合は必須
label_string_map	string	マップ要素のカスタムラベル。 プロパティの動作: - `label_type_map` が”custom” の場合は必須
label_string_trigger	string	トリガー要素のカスタムラベル。 プロパティの動作: - `label_type_trigger` が”custom” の場合は必須
label_type	integer	マップ要素のラベルタイプ。 設定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - 何も表示しない。
label_type_host	integer	ホスト要素のラベルタイプ。 設定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - 何も表示しない; 5 - カスタム。

プロパティ	タイプ	説明
label_type_hostgroup	integer	ホストグループ要素のラベルタイプ。 設定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - 何も表示しない; 5 - カスタム。
label_type_image	integer	ホストグループ要素のラベルタイプ。 設定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 4 - 何も表示しない; 5 - カスタム。
label_type_map	integer	マップ要素のラベルタイプ。 設定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - 何も表示しない; 5 - カスタム。
label_type_trigger	integer	トリガー要素のラベルタイプ。 設定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - 何も表示しない; 5 - カスタム。
markelements	integer	最近ステータスが変更されたマップ要素をハイライト表示するかどうか。 設定可能な値: 0 - *(デフォルト)* 要素をハイライトしない; 1 - 要素をハイライトする。
severity_min	integer	マップ上に表示されるトリガーの最小深刻度。 サポートされるトリガーの深刻度については`トリガーの severity プロパティ`を参照してください。
show_element_label	integer	デフォルトで要素ラベルをどのように表示するか。 設定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動的に非表示。
show_link_label	integer	デフォルトでリンクラベルをどのように表示するか。 設定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動的に非表示。
show_unack	integer	問題の表示方法。 設定可能な値: 0 - *(デフォルト)* すべての問題数を表示; 1 - 未確認の問題数のみを表示; 2 - 確認済みと未確認の問題数を個別に表示。
userid	ID	マップの所有者であるユーザーの ID。
private	integer	マップの共有タイプ。 設定可能な値: 0 - パブリックマップ; 1 - *(デフォルト)* プライベートマップ。

プロパティ	タイプ	説明
show_suppressed	integer	抑制された問題を表示するかどうか。 設定可能な値: 0 - *(デフォルト)* 抑制された問題を非表示; 1 - 抑制された問題を表示。

マップ要素

マップ要素オブジェクトは、マップ上に表示されるオブジェクトを定義します。以下のプロパティがあります。

プロパティ	型	説明
selementid	ID	マップ要素の ID。 プロパティの動作: - 読み取り専用
elements	array	要素データオブジェクト。 プロパティの動作: - `elementtype` が「ホスト」、「マップ」、「トリガー」または「ホストグループ」に設定されている場合は必須
elementtype	integer	マップ要素のタイプ。 可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。 プロパティの動作: - 必須
iconid_off	ID	デフォルト状態で要素を表示するために使用されるイメージの ID。 プロパティの動作: - 必須
areatype	integer	ホストグループのホストをどのように表示するか。 可能な値: 0 - *(デフォルト)* ホストグループ要素がマップ全体を占有する; 1 - ホストグループ要素が固定サイズになる。
elementsubtype	integer	ホストグループ要素をマップ上でどのように表示するか。 可能な値: 0 - *(デフォルト)* ホストグループを単一の要素として表示; 1 - グループ内の各ホストを個別に表示。
evaltype	integer	マップ要素タグフィルタ条件の評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
height	integer	固定サイズのホストグループ要素の高さ（ピクセル単位）。 デフォルト: 200。
iconid_disabled	ID	無効なマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が「ホスト」、「マップ」、「トリガー」または「ホストグループ」に設定されている場合にサポート

プロパティ	型	説明
iconid_maintenance	ID	メンテナンス中のマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が「ホスト」、「マップ」、「トリガー」または「ホストグループ」に設定されている場合にサポート
iconid_on	ID	問題のあるマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が「ホスト」、「マップ」、「トリガー」または「ホストグループ」に設定されている場合にサポート
label	string	要素のラベル。
label_location	integer	マップ要素ラベルの位置。 可能な値: -1 - *(デフォルト)* デフォルト位置; 0 - 下; 1 - 左; 2 - 右; 3 - 上。
permission	integer	権限レベルのタイプ。 可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き。
show_label	integer	要素ラベルの表示方法。 可能な値: -1 - *(デフォルト)* マップのデフォルト; 0 - 常に表示; 1 - 自動非表示。
sysmapid	ID	要素が属するマップの ID。 プロパティの動作: - 読み取り専用
urls	array	マップ要素の URL。 マップ要素 URL オブジェクトは以下で詳細に説明されています。
use_iconmap	integer	ホスト要素にアイコンマッピングを使用するかどうか。 可能な値: 0 - アイコンマッピングを使用しない; 1 - *(デフォルト)* アイコンマッピングを使用する。
viewtype	integer	ホストグループ要素の配置アルゴリズム。 可能な値: 0 - *(デフォルト)* グリッド。
width	integer	固定サイズのホストグループ要素の幅（ピクセル単位）。 デフォルト: 200。
x	integer	要素の X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	要素の Y 座標（ピクセル単位）。 デフォルト: 0。
zindex	integer	マップ要素の順序付けに使用される値（z-index）。 デフォルト: 0。

マップ要素ホスト

マップ要素ホストオブジェクトは、1 つのホスト要素を定義します。

プロパティ	タイプ	説明
hostid	ID	ホストの ID。

マップ要素ホストグループ

マップ要素ホストグループオブジェクトは、1 つのホストグループ要素を定義します。

プロパティ	タイプ	説明
groupid	ID	ホストグループの ID。

マップ要素マップ

マップ要素マップオブジェクトは、1 つのマップ要素を定義します。

プロパティ	タイプ	説明
sysmapid	ID	マップの ID。

マップ要素トリガー

マップ要素トリガーオブジェクトは、1 つ以上のトリガー要素を定義します。

プロパティ	タイプ	説明
triggerid	ID	トリガーの ID。

マップ要素タグ

マップ要素タグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	マップ要素タグ名。 プロパティの動作: - 必須
operator	integer	マップ要素タグ条件の演算子。 指定可能な値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
value	string	マップ要素タグ値。

マップ要素 URL

マップ要素 URL オブジェクトは、特定のマップ要素に対して利用可能なクリック可能なリンクを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
sysmapelementurlid	ID	マップ要素 URL の ID。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
name	string	リンクのキャプション。 プロパティの動作: - 必須
url	string	リンクの URL。 プロパティの動作: - 必須
selementid	ID	URL が属するマップ要素の ID。

マップリンク

マップリンクオブジェクトは、2 つのマップ要素間のリンクを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
linkid	ID	マップリンクの ID。 プロパティの動作: - 読み取り専用
sysmapid	ID	リンクが属するマップの ID。
selementid1	ID	一方の端でリンクされている最初のマップ要素の ID。 プロパティの動作: - 必須
selementid2	ID	もう一方の端でリンクされている最初のマップ要素の ID。 プロパティの動作: - 必須
drawtype	integer	リンク線の描画スタイル。 可能な値: 0 - *(デフォルト)* 線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	16 進数カラーコードで指定した線の色。 デフォルト: 000000。
label	string	リンクラベル。
show_label	integer	リンクラベルの表示方法。 可能な値: -1 - *(デフォルト)* マップのデフォルト; 0 - 常に表示; 1 - 自動的に非表示。
indicator_type	integer	リンクインジケーターのタイプを選択。 可能な値: 0 - *(デフォルト)* 静的リンク; 1 - トリガー; 2 - アイテム値。
linktriggers	array	リンク状態インジケーターとして使用するマップリンクトリガー。 マップリンクトリガーオブジェクトについては下記で詳細に説明しています。 プロパティの動作: - `indicator_type` が”trigger”(1) に設定されている場合は必須
itemid	ID	アイテムの ID。 プロパティの動作: - `indicator_type` が”item value”(2) に設定されている場合は必須

プロパティ	タイプ	説明
highlights	array	リンク状態インジケーターとして使用するマップリンクハイライト。 マップリンクインジケーターオブジェクトについては下記で詳細に説明しています。 プロパティの動作: - `indicator_type` が”item value”(2) に設定されている場合は必須
thresholds	array	リンク状態インジケーターとして使用するマップリンクしきい値。 マップリンクインジケーターオブジェクトについては下記で詳細に説明しています。 プロパティの動作: - `indicator_type` が”item value”(2) に設定されている場合は必須
permission	integer	権限レベルのタイプ。 可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き可;

マップリンクトリガー

マップリンクトリガーオブジェクトは、トリガーの状態に基づいてマップリンクのステータスインジケーターを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
triggerid	ID	リンクインジケーターとして使用されるトリガーの ID。 プロパティの動作: - 必須
color	string	16 進数のカラーコードによるインジケーターの色。 デフォルト: DD0000。
drawtype	integer	インジケーターの描画スタイル。 指定可能な値: 0 - *(デフォルト)* 線; 2 - 太線; 3 - 点線; 4 - 破線。

マップリンクインジケーター

マップリンクインジケーターオブジェクトは、アイテム値に基づいてマップリンクのステータスインジケーターを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
pattern	string	比較する正規表現。ハイライトでのみ利用可能。
sortorder	integer	ハイライトのソート順を定義するために使用。ハイライトでのみ利用可能。
threshold	string	比較する数値。しきい値でのみ利用可能。
drawtype	integer	インジケーターの描画スタイル。 利用可能な値: 0 - *(デフォルト)* 線; 2 - 太線; 3 - 点線; 4 - 破線。

プロパティ	タイプ	説明
color	string	16 進数カラーコードで指定するインジケーターの色。 デフォルト: DD0000。 プロパティの動作: - 必須

マップ URL

マップ URL オブジェクトは、マップ上の特定のタイプのすべての要素で利用可能なクリック可能なリンクを定義します。以下のプロパティがあります:

プロパティ	タイプ	説明
sysmapurlid	ID	マップ URL の ID。 プロパティの動作: - 読み取り専用
name	string	リンクのキャプション。 プロパティの動作: - 必須
url	string	リンクの URL。 プロパティの動作: - 必須
elementtype	integer	URL が利用可能となるマップ要素のタイプ。 サポートされているタイプの一覧については、マップ要素の `type` プロパティを参照してください。 デフォルト: 0。
sysmapid	ID	URL が属するマップの ID。

マップユーザー

ユーザーに基づくマップ権限の一覧。以下のプロパティがあります。

プロパティ	タイプ	説明
sysmapuserid	ID	マップユーザーの ID。 プロパティの動作: - 読み取り専用
userid	ID	ユーザーの ID。 プロパティの動作: - 必須
permission	integer	権限レベルのタイプ。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

マップユーザーグループ

ユーザーグループに基づくマップ権限のリスト。以下のプロパティがあります。

プロパティ	タイプ	説明
sysmapusrgrpid	ID	マップユーザーグループの ID。 プロパティの動作: - 読み取り専用
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 必須
permission	integer	権限レベルのタイプ。 可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

マップシェイプ

マップシェイプオブジェクトは、マップ上に表示される幾何学的なシェイプ（テキスト付きまたはテキストなし）を定義します。以下のプロパティがあります。

プロパティ	型	説明
sysmap_shapeid	ID	マップシェイプ要素の ID。 プロパティの動作: - 読み取り専用
type	integer	マップシェイプ要素のタイプ。 指定可能な値: 0 - 長方形; 1 - 楕円。 新しいシェイプを作成する場合は必須のプロパティです。 プロパティの動作: - 必須
x	integer	シェイプの X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	シェイプの Y 座標（ピクセル単位）。 デフォルト: 0。
width	integer	シェイプの幅（ピクセル単位）。 デフォルト: 200。
height	integer	シェイプの高さ（ピクセル単位）。 デフォルト: 200。
text	string	シェイプのテキスト。

プロパティ	型	説明
font	integer	シェイプ内テキストのフォント。 指定可能な値: 0 - Georgia, serif 1 - “Palatino Linotype”, “Book Antiqua”, Palatino, serif 2 - “Times New Roman”, Times, serif 3 - Arial, Helvetica, sans-serif 4 - “Arial Black”, Gadget, sans-serif 5 - “Comic Sans MS”, cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - “Lucida Sans Unicode”, “Lucida Grande”, sans-serif 8 - Tahoma, Geneva, sans-serif 9 - “Trebuchet MS”, Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - “Courier New”, Courier, monospace 12 - “Lucida Console”, Monaco, monospace デフォルト: 9。
font_size	integer	フォントサイズ（ピクセル単位）。 デフォルト: 11。
font_color	string	フォントカラー。 デフォルト: 000000。
text_halign	integer	テキストの水平位置。 指定可能な値: 0 - 中央; 1 - 左; 2 - 右。 デフォルト: 0。
text_valign	integer	テキストの垂直位置。 指定可能な値: 0 - 中央; 1 - 上; 2 - 下。 デフォルト: 0。
border_type	integer	枠線のタイプ。 指定可能な値: 0 - なし; 1 - ──────; 2 - ┅; 3 - – – –。 デフォルト: 0。
border_width	integer	枠線の幅（ピクセル単位）。 デフォルト: 0。
border_color	string	枠線の色。 デフォルト: 000000。
background_color	string	背景色（塗りつぶし色）。 デフォルト:（空）。
zindex	integer	すべてのシェイプと線の順序付けに使用される値（z-index）。 デフォルト: 0。

マップライン

マップラインオブジェクトは、マップ上に表示される線を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
sysmap_shapeid	ID	マップシェイプ要素の ID。 プロパティの動作: - 読み取り専用
x1	integer	ポイント 1 の X 座標（ピクセル単位）。 デフォルト: 0。
y1	integer	ポイント 1 の Y 座標（ピクセル単位）。 デフォルト: 0。
x2	integer	ポイント 2 の X 座標（ピクセル単位）。 デフォルト: 200。
y2	integer	ポイント 2 の Y 座標（ピクセル単位）。 デフォルト: 200。
line_type	integer	線の種類。 可能な値: 0 - なし; 1 - ————; 2 - ┅; 3 - – – –。 デフォルト: 0。
line_width	integer	線の幅（ピクセル単位）。 デフォルト: 0。
line_color	string	線の色。 デフォルト: 000000。
zindex	integer	すべてのシェイプとラインの順序付けに使用される値（z-index）。 デフォルト: 0。

map.create

説明

`object map.create(object/array maps)`

このメソッドは新しいマップを作成するためのものです。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するマップ。

標準のマッププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
links	array	マップ上に作成するマップリンク。
selements	array	マップ上に作成するマップ要素。
urls	array	マップ上に作成するマップ URL。
users	array	マップ上に作成するマップユーザーの共有。

パラメータ	タイプ	説明
userGroups	array	マップ上に作成するマップユーザーグループの共有。
shapes	array	マップ上に作成するマップシェイプ。
lines	array	マップ上に作成するマップライン。

Note:
マップリンクを作成するには、マップ要素の `selementid` に任意の値を設定し、この値をリンクの `selementid1` または `selementid2` プロパティで参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。例を参照してください。

戻り値

(object) `sysmapids` プロパティの下で作成されたマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたマップの順序と一致します。

例

空のマップを作成する

要素のないマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

ホストマップの作成

2 つのホスト要素とそれらの間のリンクを持つマップを作成します。マップリンクオブジェクトでマップ要素を参照するために、一時的な”selementid1” および”selementid2” の値を使用していることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "selementid": "1",
                "elements": [
                    {"hostid": "1033"}
                ],
```

```
                "elementtype": 0,
                "iconid_off": "2"
            },

            {
                "selementid": "2",
                "elements": [
                    {"hostid": "1037"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            }
        ],
        "links": [
            {
                "selementid1": "1",
                "selementid2": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

トリガーマップの作成

2 つのトリガーを含むトリガー要素でマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Trigger map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "elements": [
                    {"triggerid": "12345"},
                    {"triggerid": "67890"}
                ],
                "elementtype": 2,
                "iconid_off": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

マップの共有

2 種類の共有 (ユーザーとユーザーグループ) でマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map sharing",
        "width": 600,
        "height": 600,
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

マップシェイプ

マップ名のタイトルを持つマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "shapes": [
            {
                "type": 0,
                "x": 0,
                "y": 0,
```

```
                "width": 600,
                "height": 11,
                "text": "{MAP.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

マップライン

マップラインを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map API lines",
        "width": 500,
        "height": 500,
        "lines": [
            {
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

参照

- Map element
- Map link
- Map URL
- Map user

- Map user group
- Map shape
- Map line

ソース

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

説明

`object map.delete(array mapIds)`

このメソッドは、マップを削除することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するマップの ID。

戻り値

(object) sysmapids プロパティの下にある削除されたマップの ID を含むオブジェクトを返します。

例

複数のマップの削除

2 つのマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

ソース

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

説明

`integer/array map.get(object parameters)`

このメソッドは、指定されたパラメータに従ってマップを取得することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
sysmapids	ID/array	指定した ID のマップのみを返します。
userids	ID/array	指定したユーザー ID に属するマップのみを返します。
expandUrls	flag	グローバルマップ URL を対応するマップ要素に追加し、すべてのマップ要素 URL のマクロを展開します。
selectIconMap	query	マップで使用されているアイコンマップを持つ**iconmap**プロパティを返します。
selectLinks	query	要素間のマップリンクを持つ**links**プロパティを返します。
selectSelements	query	マップ要素を持つ**selements**プロパティを返します。
selectUrls	query	マップ URL を持つ**urls**プロパティを返します。
selectUsers	query	マップが共有されているユーザーを持つ**users**プロパティを返します。
selectUserGroups	query	マップが共有されているユーザーグループを持つ**userGroups**プロパティを返します。
selectShapes	query	マップシェイプを持つ**shapes**プロパティを返します。
selectLines	query	マップラインを持つ**lines**プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: **name**, **width**, **height**.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- **countOutput** パラメータが使用されている場合、取得されたオブジェクトの数

例

マップの取得

マップ"3" のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.get",
    "params": {
        "output": "extend",
        "selectSelements": "extend",
        "selectLinks": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
```

```
        "selectShapes": "extend",
        "selectLines": "extend",
        "sysmapids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
```

```
                "use_iconmap": "1",
                "show_label": "0",
                "zindex":"0",
                "elements": [],
                "urls": [],
                "tags": []
            },
            {
                "selementid": "12",
                "sysmapid": "3",
                "elementtype": "0",
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
```

```
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
        "grid_show": "1",
        "grid_align": "1",
        "label_format": "0",
        "label_type_host": "2",
        "label_type_hostgroup": "2",
        "label_type_trigger": "2",
        "label_type_map": "2",
```

```
            "label_type_image": "2",
            "label_string_host": "",
            "label_string_hostgroup": "",
            "label_string_trigger": "",
            "label_string_map": "",
            "label_string_image": "",
            "iconmapid": "0",
            "expand_macros": "0",
            "severity_min": "0",
            "userid": "1",
            "private": "1",
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```


参照

- Icon map
- Map element
- Map link
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

ソース

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

説明

`object map.update(object/array maps)`

このメソッドは、既存のマップを更新するためのものです。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するマップのプロパティ。

各マップには `mapid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のマッププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
links	array	既存のリンクを置き換えるマップリンク。
selements	array	既存の要素を置き換えるマップ要素。
urls	array	既存の URL を置き換えるマップ URL。
users	array	既存の要素を置き換えるマップユーザーの共有。
userGroups	array	既存の要素を置き換えるマップユーザーグループの共有。
shapes	array	既存の図形を置き換えるマップ図形。
lines	array	既存の線を置き換えるマップ線。

Note:
新しいマップ要素間でマップリンクを作成するには、要素の selementid に任意の値を設定し、この値をリンクの selementid1 または selementid2 プロパティでこの要素を参照するために使用する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。map.create の例を参照してください。

戻り値

(object) sysmapids プロパティの下で更新されたマップの ID を含むオブジェクトを返します。

例

マップのサイズ変更

マップのサイズを 1200x1200 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

マップの所有者を変更

管理者とスーパ管理者のみ利用可能です。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

参照

- Map element
- Map link
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

ソース

CMap::update() in *ui/include/classes/api/services/CMap.php*.

Media type

このクラスは、メディアタイプを操作するためのものです。

オブジェクトリファレンス:

- Media type
  - Webhook parameters
  - Script parameters
- Message template

利用可能なメソッド:

- mediatype.create - 新しいメディアタイプを作成する
- mediatype.delete - メディアタイプを削除する
- mediatype.get - メディアタイプを取得する
- mediatype.update - メディアタイプを更新する

Media type オブジェクト

次のオブジェクトは、`mediatype`API に直接関連しています。

メディアタイプ

メディアタイプオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
mediatypeid	ID	メディアタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作時は必須
name	string	メディアタイプの名前。 プロパティの動作: - 作成操作時は必須
type	integer	メディアタイプで使用されるトランスポート。 可能な値: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook。
exec_path	string	プロパティの動作: - 作成操作時は必須 AlertScriptsPathサーバー設定パラメータで指定されたディレクトリに配置されたスクリプトファイル名（例: notification.sh）。 プロパティの動作: - `type` が”Script” に設定されている場合は必須

プロパティ	型	説明
gsm_modem	string	GSM モデムのシリアルデバイス名。 プロパティの動作: - `type` が”SMS” に設定されている場合は必須
passwd	string	認証パスワード。 プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合はサポート
provider	integer	メールプロバイダー。 可能な値: 0 - *(デフォルト)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay。
smtp_email	string	通知を送信するメールアドレス。 プロパティの動作: - `type` が”Email” に設定されている場合は必須
smtp_helo	string	SMTP HELO。 プロパティの動作: - `type` が”Email” に設定されている場合はサポート
smtp_server	string	SMTP サーバー。 プロパティの動作: - `type` が”Email” に設定されている場合は必須
smtp_port	integer	接続する SMTP サーバーポート。 デフォルト: 25。 プロパティの動作: - `type` が”Email” に設定されている場合はサポート
smtp_security	integer	使用する SMTP 接続のセキュリティレベル。 可能な値: 0 - *(デフォルト)* なし; 1 - STARTTLS; 2 - SSL/TLS。 プロパティの動作: - `type` が”Email” に設定されている場合はサポート
smtp_verify_host	integer	SMTP の SSL verify host。 可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。 プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合はサポート
smtp_verify_peer	integer	SMTP の SSL verify peer。 可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。 プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合はサポート

プロパティ	型	説明
smtp_authentication	integer	使用する SMTP 認証方式。 可能な値: 0 - *(デフォルト)* なし; 1 - Normal password; 2 - OAuth token。 OAuth 認証は *Office365 relay* メールプロバイダーでは許可されていません。 プロパティの動作: - `type` が”Email” に設定されている場合はサポート
redirection_url	string	OAuth 認証後にリダイレクトする Zabbix フロントエンドの URL。 デフォルト: API 設定プロパティ `url` の値に `zabbix.php?action=oauth.authorize` を付加したもの プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
client_id	string	OAuth 認証サーバーに登録されたクライアント識別子。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
client_secret	string	OAuth 認証サーバーに登録されたクライアントシークレット。Super Admin タイプのユーザーのみアクセス可能。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
authorization_url	string	アクセストークンおよびリフレッシュトークンを取得するためのパラメータ付き OAuth URL。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
token_url	string	認証トークンをアクセストークンおよびリフレッシュトークンに交換するための OAuth URL。 この URL は、サーバーが無効なアクセストークンをリフレッシュする際にも使用されます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
tokens_status	integer	トークンの状態を示すビットマスク。 可能な値: 0 - (デフォルト) 両方のトークンが無効な値を含む 1 - アクセストークンが有効な値を含む 2 - リフレッシュトークンが有効な値を含む 3 - 両方のトークンが有効な値を含む。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合はサポート
access_token	string	OAuth アクセストークンの値。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須

プロパティ	型	説明
access_token_updated	timestamp	`access_token` がサーバーによって `refresh_token` でリフレッシュされた、または API でトークンが変更された際の最終更新タイムスタンプ。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合はサポート
access_expires_in	integer	`access_token` が期限切れとなり、`refresh_url` へのリクエストが必要になるまでの秒数。 `access_token` のリフレッシュ時や API でトークンが変更された際に Zabbix サーバーによって設定されます。 タイムスタンプは `access_token_updated` の値を加算して計算されます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合はサポート
refresh_token	string	OAuth リフレッシュトークンの値。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
status	integer	メディアタイプが有効かどうか。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
username	string	ユーザー名。 プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合はサポート
maxsessions	integer	並列で処理できるアラートの最大数。 `type` が”SMS” に設定されている場合の可能な値: 1。 `type` が”Email”、”Script”、”Webhook” に設定されている場合の可能な値: 0-100。 デフォルト: 1。
maxattempts	integer	アラート送信の最大試行回数。 可能な値: 1-100。 デフォルト: 3。
attempt_interval	string	再試行間隔。 秒数またはサフィックス付きの時間単位を受け付けます。 可能な値: 0-1h。 デフォルト: 10s。
message_format	integer	メッセージフォーマット。 可能な値: 0 - プレーンテキスト; 1 - *(デフォルト)* HTML。 プロパティの動作: - `type` が”Email” に設定されている場合はサポート

プロパティ	型	説明
script	text	Webhook スクリプト本体（JavaScript）。
timeout	string	プロパティの動作: - type が”Webhook” に設定されている場合は必須 Webhook スクリプトのタイムアウト。 秒数またはサフィックス付きの時間単位を受け付けます。 可能な値: 1-60s。 デフォルト: 30s。
process_tags	integer	プロパティの動作: - type が”Webhook” に設定されている場合はサポート Webhook スクリプトのレスポンスの JSON プロパティ値をタグとして処理します。これらのタグは既存の問題タグに追加されます。 可能な値: 0 - *(デフォルト)* webhook スクリプトのレスポンスを無視; 1 - webhook スクリプトのレスポンスをタグとして処理。
show_event_menu	integer	プロパティの動作: - type が”Webhook” に設定されている場合はサポート カスタム URL へのリンクを含むイベントメニューのエントリを含めます。また、problem.getおよびevent.getの出力に `urls` プロパティを追加します。 可能な値: 0 - *(デフォルト)* イベントメニューエントリや `urls` プロパティを含めない; 1 - イベントメニューエントリおよび `urls` プロパティを含める。
event_menu_url	string	プロパティの動作: - type が”Webhook” に設定されている場合はサポート イベントメニューエントリおよびproblem.get、event.getで返される `urls` プロパティで使用される URL。
event_menu_name	string	プロパティの動作: - `show_event_menu` が” イベントメニューエントリおよび `urls` プロパティを含める” に設定されている場合はサポート イベントメニューエントリおよびproblem.get、event.getで返される `urls` プロパティで使用される名前。
parameters	array	プロパティの動作: - `show_event_menu` が” イベントメニューエントリおよび `urls` プロパティを含める” に設定されている場合はサポート Webhookまたはスクリプトのパラメータ。
description	text	プロパティの動作: - type が”Webhook” または”Script” に設定されている場合はサポート メディアタイプの説明。

Webhook パラメータ

Webhook パラメータには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	パラメータ名。 プロパティの動作: - 必須

プロパティ	タイプ	説明
value	string	パラメータ値。マクロをサポート。 サポートされているマクロはサポートされているマクロページに記載されています。

スクリプトパラメータ

スクリプトパラメータには以下のプロパティがあります。

プロパティ	タイプ	説明
sortorder	整数	パラメータ値がコマンドライン引数としてスクリプトに渡される順序。 先頭は 0 です。 プロパティの動作: - 必須
値	文字列	パラメータ値。マクロをサポートします。 サポートされているマクロについては、サポートされているマクロページで説明されています。

メッセージテンプレート

メッセージテンプレートオブジェクトは、通知を送信するためのアクション操作のデフォルトメッセージとして使用されるテンプレートを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
eventsource	integer	イベントソース。 可能な値: 0 - トリガー; 1 - ディスカバリ; 2 - 自動登録; 3 - 内部; 4 - サービス。 プロパティの動作: - 必須
recovery	integer	操作モード。 可能な値: 0 - 操作; 1 - リカバリ操作; 2 - 更新操作。 プロパティの動作: - 必須
subject	string	メッセージの件名。
message	string	メッセージ本文。

mediatype.create

説明

`object mediatype.create(object/array mediaTypes)`

このメソッドは新しいメディアタイプを作成します。

Note:
このメソッドはスーパ管理者ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 作成するメディアタイプ。

標準のメディアタイププロパティに加えて、メソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
message_templates	array	メディアタイプに作成するメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下に作成されたメディアタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメディアタイプの順序と一致します。

例

メールメディアタイプの作成

カスタム SMTP ポートとメッセージテンプレートを使用して新しいメールメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.R
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

スクリプトメディアタイプの作成

試行回数と試行間隔にカスタム値を設定した新しいスクリプトメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Webhook メディアタイプの作成

新しい Webhook メディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "4",
        "name": "Webhook",
        "script": "var Webhook = {\r\n    token: null,\r\n    to: null,\r\n    subject: null,\r\n    messa
        "parameters": [
            {
                "name": "Message",
                "value": "{ALERT.MESSAGE}"
            },
            {
                "name": "Subject",
                "value": "{ALERT.SUBJECT}"
            },
```

```
            {
                "name": "To",
                "value": "{ALERT.SENDTO}"
            },
            {
                "name": "Token",
                "value": "<Token>"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

説明

`object mediatype.delete(array mediaTypeIds)`

このメソッドは、メディアタイプを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するメディアタイプの ID。

戻り値

(object) mediatypeids プロパティの下で削除されたメディアタイプの ID を含むオブジェクトを返します。

例

複数のメディアタイプの削除

2 つのメディアタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

説明

`integer/array mediatype.get(object parameters)`

このメソッドは、指定されたパラメータに従ってメディアタイプを取得することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

Note:
メディアタイプのユーザー関連情報をリクエストする場合、*Admin* および *User* タイプのユーザーは自分自身のユーザーに関するデータのみ取得できます。例については、*Admin* としてメディアタイプを取得するを参照してください。

パラメータ	型	説明
mediatypeids	ID/array	指定した ID のメディアタイプのみを返します。
mediaids	ID/array	指定したメディアで使用されているメディアタイプのみを返します。
userids	ID/array	指定したユーザーで使用されているメディアタイプのみを返します。
selectActions	query	メディアタイプを使用する`actions`プロパティを返します。
selectMessageTemplates	query	メディアタイプメッセージの配列を持つ`message_templates`プロパティを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
selectUsers	query	メディアタイプを使用する`users`プロパティを返します。 ユーザータイプに基づく制限については、`user.get`を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `mediatypeid`。

パラメータ	型	説明
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 textデータ型のプロパティはサポートされません。 *Admin* および *User* タイプのユーザーが指定可能なメディアタイプオブジェクトのプロパティ: mediatypeid, name, type, status, maxattempts。
output	query	返されるメディアタイプオブジェクトのプロパティ。 *Admin* および *User* タイプのユーザーは、次のプロパティのみ取得できます: mediatypeid, name, type, status, maxattempts, description。例については、*Admin* としてメディアタイプを取得するを参照してください。 デフォルト: extend。
search	object	指定したパターンに一致する結果を返します（大文字小文字を区別しません）。 オブジェクトを受け入れ、キーはプロパティ名、値は検索する文字列です。追加オプションがない場合、LIKE "%…%" 検索を実行します。 string および textデータ型のプロパティのみサポートします。 *Admin* および *User* タイプのユーザーが指定可能なメディアタイプオブジェクトのプロパティ: name, description。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

メディアタイプの取得

すべての設定済みメディアタイプを取得します。次の例では、2 つのメディアタイプを返します:

- メールのメディアタイプ
- SMS のメディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "smtp_server": "mail.example.com",
            "smtp_helo": "example.com",
            "smtp_email": "zabbix@example.com",
            "exec_path": "",
            "gsm_modem": "",
            "username": "",
            "passwd": "",
            "status": "0",
            "smtp_port": "25",
            "smtp_security": "0",
            "smtp_verify_peer": "0",
            "smtp_verify_host": "0",
            "smtp_authentication": "0",
            "maxsessions": "1",
            "maxattempts": "3",
            "attempt_interval": "10s",
            "message_format": "0",
            "script": "",
            "timeout": "30s",
            "process_tags": "0",
            "show_event_menu": "1",
            "event_menu_url": "",
            "event_menu_name": "",
            "description": "",
            "provider": "0",
            "message_templates": [
                {
                    "eventsource": "0",
                    "recovery": "0",
                    "subject": "Problem: {EVENT.NAME}",
                    "message": "Problem started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "Resolved: {EVENT.NAME}",
                    "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE
                },
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "Updated problem: {EVENT.NAME}",
                    "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                    "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "Autoregistration: {HOST.HOST}",
```

```
                "message": "Host name: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.PORT}"
            }
        ],
        "parameters": []
    },
    {
        "mediatypeid": "3",
        "type": "2",
        "name": "SMS",
        "smtp_server": "",
        "smtp_helo": "",
        "smtp_email": "",
        "exec_path": "",
        "gsm_modem": "/dev/ttyS0",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "1",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "",
                "message": "{EVENT.SEVERITY}: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "",
                "message": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "",
                "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
            },
            {
                "eventsource": "2",
                "recovery": "0",
```

```
                    "subject": "",
                    "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
                }
            ],
            "parameters": []
        }
    ],
    "id": 1
}
```

Admin としてメディアタイプを取得

Admin タイプのユーザーとして、有効になっているすべてのメディアタイプと、それらのメディアタイプを使用するユーザーを取得します。次の例では、2 つのメディアタイプが返されます。

- ユーザーが 1 人いるメールメディアタイプ（*Admin* タイプのユーザー自身のユーザーのみ）
- ユーザーがいない SMS メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
                    "surname": "Doe",
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
```

```
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```

スクリプトと Webhook のメディアタイプ取得

次の例では、3 つのメディアタイプが返されます。

- パラメータ付きのスクリプトメディアタイプ
- パラメータなしのスクリプトメディアタイプ
- パラメータ付きの Webhook メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
```

```
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- User

ソース

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

説明

`object mediatype.update(object/array mediaTypes)`

このメソッドは、既存のメディアタイプを更新します。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 更新するメディアタイプのプロパティ。

各メディアタイプには `mediatypeid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のメディアタイププロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
message_templates	array	現在のメッセージテンプレートを置き換えるメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下で更新されたメディアタイプの ID を含むオブジェクトを返します。

例

メディアタイプの有効化

メディアタイプを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

MFA

このクラスは、MFA(多要素認証) メソッドを操作するためのものです。

オブジェクトリファレンス:

- MFA

利用可能なメソッド:

- mfa.create - 新しい MFA メソッドを作成する
- mfa.delete - MFA メソッドを削除する
- mfa.get - MFA メソッドを取得する
- mfa.update - MFA メソッドを更新する

MFA オブジェクト

次のオブジェクトは、mfaAPI に直接関連しています。

MFA

MFA（多要素認証）オブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
mfaid	ID	MFA 方式の ID プロパティの動作: - 読み取り専用 - 更新操作に必須
type	整数	MFA 方式のタイプ 可能な値: 1 - TOTP (時間ベースのワンタイムパスワード); 2 - Duo ユニバーサルプロンプト
name	文字列	MFA メソッドの一意の名前 プロパティの動作: - 作成操作の場合は 必須
hash_function	整数	TOTP コードを生成するためのハッシュ関数のタイプ 可能な値: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512 プロパティの動作: - type が”TOTP” に設定されている場合は 必須
code_length	整数	検証コードの長さ 有効な値: 6 - 6 桁; 8 - 8 桁 プロパティの動作: - 必須 type が”TOTP” に設定されている場合
api_hostname	文字列	Duo 認証サービスによって提供される API ホスト名 プロパティの動作: - 必須 type が”Duo Universal Prompt” に設定されている場合
clientid	文字列	Duo 認証サービスによって提供されるクライアント ID プロパティの動作: - 必須 type が”Duo Universal Prompt” に設定されている場合
client_secret	文字列	Duo 認証サービスによって提供されるクライアントシークレット プロパティの動作: - 書き込み専用 - type が”Duo Universal Prompt” に設定されている場合、必須

mfa.create

説明

`object mfa.create(object/array MFA methods)`

このメソッドは、新しい MFA メソッドを作成します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロールをご覧ください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受けつけます。

戻り値

(object) mfaids プロパティの下で作成された MFA メソッドの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

MFA メソッドの作成

時間ベースのワンタイムパスワード (TOTP) を利用した”Zabbix TOTP” MFA メソッドを作成します。TOTP コード生成用のハッシュ関数は SHA-1 に設定し、検証コードの長さは 6 桁に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```


```
{
"jsonrpc": "2.0",
"method": "mfa.create",
"params": {
"type": 1,
"name": "Zabbix TOTP",
"hash_function": 1,
"code_length": 6
},
"id": 1
}
```


レスポンス:


```
{
"jsonrpc": "2.0",
"result": {
"mfaids": [
"1"
]
},
"id": 1
}
```


ソース

CMfa::create() in *ui/include/classes/api/services/CMfa.php*.

mfa.delete

説明

`object mfa.delete(array mfaids)`

このメソッドは、MFA メソッドを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する MFA メソッドの ID。

戻り値

(object) mfaids プロパティの下にある削除された MFA メソッドの ID を含むオブジェクトを返します。

例

MFA メソッドの削除

MFA メソッドを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CMfa::delete() in *ui/include/classes/api/services/CMfa.php*.

mfa.get

説明

`integer/array mfa.get(object parameters)`

このメソッドは、与えられたパラメータに基づいて MFA メソッドを取得します。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

パラメータ

(object) は、必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
mfaids	ID/配列	指定された ID の MFA メソッドのみを返します。
selectUsrgrps	クエリ	MFA メソッドに関連付けられたユーザーグループを含む usrgrps プロパティを返します。 count をサポートしています。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 オブジェクトを受け入れます。キーはプロパティ名、値は照合する単一の値または値の配列です。 サポートされるプロパティ: mfaid - MFA メソッドの ID type - MFA メソッドのタイプ
sortfield	文字列/配列	指定されたプロパティで結果を並べ替えます。 可能な値: name
search	object	指定されたパターンに一致する結果を返します（大文字と小文字は区別されません） オブジェクトを受け入れます。キーはプロパティ名、値は検索する文字列です。追加のオプションが指定されていない場合は、LIKE "%…%" 検索が実行されます。 サポートされるプロパティ: name
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明されています。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

名前による MFA メソッドの検索

名前に”Zabbix” が含まれるすべての MFA メソッドを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```

ソース

CMfa::get() in *ui/include/classes/api/services/CMfa.php*.

mfa.update

説明

`object mfa.update(object/array MFA methods)`

このメソッドは、既存の MFA メソッドを更新します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割を参照してください。

パラメータ

(object/array) 更新する MFA メソッドのプロパティ

`mfaid` プロパティは各項目に対して定義する必要があります。その他のプロパティはオプションです。渡されたプロパティのみが更新され、その他のプロパティは変更されません。

このメソッドは、標準 MFA メソッドのプロパティ を持つ MFA メソッドを受けつけます。

戻り値

(object) `mfaids` プロパティの下にある更新された MFA メソッドの ID を含むオブジェクトを返します。

例

Updating method properties

メソッドプロパティの更新

時間ベースのワンタイムパスワード（TOTP）を利用した”Zabbix TOTP” MFA メソッドの TOTP コード生成用ハッシュ関数と検証コードの長さを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
```

```
        "code_length": 8
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMfa::update() in *ui/include/classes/api/services/CMfa.php*.

Module

このクラスは、Web コンソールのモジュールを操作するためのものです。

オブジェクトリファレンス:

- Module

利用可能なメソッド:

- module.create - 新しいモジュールをインストールする
- module.delete - モジュールをアンインストールする
- module.get - モジュールを取得する
- module.update - モジュールを更新する

Module オブジェクト

次のオブジェクトは、`module`API に直接関連しています。

モジュール

モジュールオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
moduleid	ID	データベースに保存されているモジュールの ID プロパティの動作: - 読み取り専用 - 更新操作に必須
id	文字列	モジュールのmanifest.jsonファイルで開発者が定義した一意のモジュール ID 組み込みモジュールの可能な値: ダッシュボードウィジェットのプロパティ”type” の説明を参照してください。 プロパティの動作: - 作成操作では必須

プロパティ	タイプ	説明
relative_path	文字列	Zabbix フロントエンドのディレクトリを基準としたモジュールのディレクトリへの相対パス 使用可能な値: `widgets/*` - 組み込みウィジェットモジュールの場合 `modules/*` - サードパーティ製モジュールの場合 プロパティの動作: - 作成操作の場合は 必須
status	整数	モジュールが有効か無効か 使用可能な値: 0 - *(デフォルト)* 無効 1 - 有効
config	オブジェクト	モジュール設定

module.create

Description

`object module.create(object/array modules)`

このメソッドは、新しいフロントエンドモジュールをインストールします。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

Attention:
モジュールファイルは、モジュールの `relative_path` プロパティに一致する正しいサブディレクトリに手動で展開する必要があります。

パラメータ

`(object/array)` 作成するモジュール。

このメソッドは、標準モジュールプロパティを持つモジュールを受け入れます。

戻り値

`(object)` は、`moduleids` プロパティにインストールされているモジュールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたモジュールの順序と一致します。

例

モジュールのインストール

モジュールをステータス" 有効" でインストールします。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::create() in *ui/include/classes/api/services/CModule.php*.

module.delete

説明

`object module.delete(array moduleids)`

このメソッドは、モジュールをアンインストールすることができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

> **Attention:**
> モジュールファイルは手動で削除する必要があります。

パラメーター

(array) アンインストールするモジュールの ID。

戻り値

(object) moduleids プロパティの下にあるアンインストールされたモジュールの ID を含むオブジェクトを返します。

例

複数のモジュールのアンインストール

モジュール”27” と”28” をアンインストールします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
```

```
        ]
    },
    "id": 1
}
```

ソース

CModule::delete() in *ui/include/classes/api/services/CModule.php*.

module.get

説明

`integer/array module.get(object parameters)`

このメソッドは、指定されたパラメータに従ってモジュールを取得することができます。

Note:
このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

Parameters

`(object)` 出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
moduleids	ID/配列	指定された ID を持つモジュールのみを返します。
sortfield	文字列/配列	指定されたプロパティで結果を並べ替えます。 使用可能な値: `moduleid`、`relative_path`
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	ブール値	
search	オブジェクト	
searchByAny	ブール値	
searchWildcardsEnabled	ブール値	
sortorder	文字列/配列	
startSearch	ブール値	

戻り値

`(integer/array)` 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ID でモジュールを取得する

モジュール”1”、”2”、”25” に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
```

```
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

参照

- Module
- Dashboard widget
- Frontend modules

ソース

CModule::get() in *ui/include/classes/api/services/CModule.php*.

module.update

説明

`object proxy.update(object/array proxies)`

このメソッドは、既存のプロキシを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新されるトークンプロパティ

`tokenid` プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) roleids プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

モジュールの無効化

モジュール"25" を無効化します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::update() in *ui/include/classes/api/services/CModule.php*.

Problem

このクラスは、障害を操作するためのものです。

オブジェクトリファレンス:

- Problem
  - Media type URL
- Problem tag

利用可能なメソッド:

- problem.get - 障害を取得する

Problem オブジェクト

次のオブジェクトは、problem API に直接関連しています。

問題

> **Note:**
> 問題は Zabbix サーバーによって作成され、API 経由で変更することはできません。

問題オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
eventid	ID	問題イベントの ID
source	integer	問題イベントのタイプ 有効な値: 0 - トリガーによって作成されたイベント 3 - 内部イベント 4 - サービスステータスの更新時に作成されたイベント
object	integer	問題イベントに関連するオブジェクトのタイプ `source` が「トリガーによって作成されたイベント」に設定されている場合の可能な値: 0 - トリガー `source` が「内部イベント」に設定されている場合の可能な値: 0 - トリガー 4 - アイテム 5 - LLD ルール `source` が「サービスステータス更新時に作成されたイベント」に設定されている場合の可能な値: 6 - サービス
objectid	ID	関連オブジェクトの ID
clock	timestamp	問題イベントが作成された時刻
ns	integer	問題イベントが作成されたナノ秒
r_eventid	ID	リカバリイベントの ID
r_clock	timestamp	リカバリイベントが作成された時刻
r_ns	整数	リカバリイベントが作成されたナノ秒数
cause_eventid	ID	原因イベントの ID
correlationid	ID	このイベントがグローバル相関ルールによってリカバリされた場合の相関ルールの ID
userid	ID	問題をクローズしたユーザーの ID（問題が手動でクローズされた場合）
name	文字列	解決された問題の名前
acknowledged	整数	問題の確認状態 可能な値： 0 - 未確認 1 - 確認済み
severity	整数	問題の現在の重大度 可能な値： 0 - 未分類 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害
suppressed	整数	問題が抑制されているかどうか 可能な値: 0 - 問題は正常状態です。 1 - 問題は抑制されています。
opdata	文字列	拡張マクロを含む運用データ
urls	配列	アクティブなメディアタイプ URL

メディアタイプ URL

メディアタイプ URL オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	メディアタイプで定義された URL 名。

プロパティ	タイプ	説明
url	string	メディアタイプで定義された URL 値。

結果には、イベントメニューエントリが有効なアクティブなメディアタイプのエントリのみが含まれます。プロパティで使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては、サポートされているマクロを参照してください。

Problem tag

障害タグオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag	string	障害タグ名
value	string	障害タグ値

problem.get

説明

`integer/array problem.get(object parameters)`

このメソッドは、指定されたパラメータに従って問題を取得するためのものです。

このメソッドは、未解決の問題を取得するためのものです。また、指定した場合は、最近解決された問題も追加で取得することができます。「最近」がどれくらい前までかは、管理 → 一般 で定義されています。その期間より前に解決された問題は、問題テーブルには保持されません。さらに過去に解決された問題を取得するには、`event.get` メソッドを使用してください。

Attention:
このメソッドは、ハウスキーパーによってまだ削除されていない場合、削除されたエンティティの問題を返すことがあります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
eventids	ID/array	指定した ID の問題のみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成された問題のみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成された問題のみを返します。
objectids	ID/array	指定したオブジェクトによって作成された問題のみを返します。
source	integer	指定したタイプの問題のみを返します。 サポートされているイベントタイプの一覧は問題イベントオブジェクトページを参照してください。 デフォルト: 0 - トリガーによって作成された問題。
object	integer	指定したタイプのオブジェクトによって作成された問題のみを返します。 サポートされているオブジェクトタイプの一覧は問題イベントオブジェクトページを参照してください。 デフォルト: 0 - トリガー。

パラメーター	型	説明
acknowledged	boolean	`true` - 確認済みの問題のみを返します; `false` - 未確認のみ。
action	integer	指定したイベント更新アクションが実行された問題のみを返します。複数のアクションの場合、任意のビットマップ値の合計をビットマスクとして使用します（例: 34 はイベントの確認と抑制）。
action_userids	ID/array	問題イベントの更新アクションを実行したユーザーの ID が指定された問題のみを返します。
suppressed	boolean	`true` - 抑制された問題のみを返します; `false` - 通常状態の問題を返します。
symptom	boolean	`true` - 症状の問題イベントのみを返します; `false` - 原因の問題イベントのみを返します。
severities	integer/array	指定したイベントの深刻度の問題のみを返します。object がトリガーの場合のみ適用されます。
evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグの問題のみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべての問題を返します。 可能なoperator値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
recent	boolean	`true` - PROBLEM と最近 RESOLVED した問題を返します（N 秒間 OK トリガーを表示に依存） デフォルト: `false` - UNRESOLVED の問題のみ
eventid_from	string	指定した ID 以上の ID を持つ問題のみを返します。
eventid_till	string	指定した ID 以下の ID を持つ問題のみを返します。
time_from	timestamp	指定した時刻以降に作成された問題のみを返します。
time_till	timestamp	指定した時刻以前に作成された問題のみを返します。
selectAcknowledges	query	問題の更新情報を含む `acknowledges` プロパティを返します。問題の更新は逆時系列でソートされます。 問題の更新オブジェクトは以下のプロパティを持ちます: `acknowledgeid` - (ID) 更新の ID; `userid` - (ID) イベントを更新したユーザーの ID; `eventid` - (ID) 更新されたイベントの ID; `clock` - (timestamp) イベントが更新された時刻; `message` - (string) メッセージのテキスト; `action` - (integer) 更新アクションのタイプ (`event.acknowledge`参照); `old_severity` - (integer) この更新アクション前のイベントの深刻度; `new_severity` - (integer) この更新アクション後のイベントの深刻度; `suppress_until` - (timestamp) イベントが抑制されるまでの時刻; `taskid` - (ID) 現在のイベントがランク変更中の場合のタスク ID。 `count` をサポートします。
selectTags	query	問題のタグを含むtagsプロパティを返します。出力フォーマット: `[{"tag": "<tag>", "value": "<value>"}, ...]`。

パラメーター	型	説明
selectSuppressionData	query	アクティブなメンテナンスと手動抑制のリストを含む suppression_data プロパティを返します: maintenanceid - (ID) メンテナンスの ID; userid - (ID) 問題を抑制したユーザーの ID; suppress_until - (integer) 問題が抑制されるまでの時刻。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 text データ型のプロパティはサポートされません。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: eventid。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トリガーの問題イベントの取得

トリガー"15112" の最近のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectTags": "extend",
        "selectSuppressionData": "extend",
        "objectids": "15112",
        "recent": true,
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
```

```
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```


指定されたユーザーが確認済みの問題を取得しています

ID=10 のユーザーが確認済みの問題を取得しています

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "id": 1
}
```

参照

- Alert
- Item
- Host
- LLD rule
- Trigger

ソース

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

Proxy

このクラスは、プロキシを操作するためのものです。

オブジェクトリファレンス:

- Proxy

利用可能なメソッド:

- proxy.create - 新しいプロキシを作成する
- proxy.delete - プロキシを削除する
- proxy.get - プロキシを取得する
- proxy.update - プロキシを更新する

Proxy オブジェクト

次のオブジェクトは、proxyAPI に直接関連しています。

プロキシ

プロキシオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
proxyid	ID	プロキシの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	プロキシの名前。 プロパティの動作: - 作成操作時に必須
proxy_groupid	ID	プロキシグループの ID。 0 - プロキシはプロキシグループに割り当てられていません。
local_address	string	アクティブエージェント用のアドレス。接続先の IP アドレスまたは DNS 名。 プロパティの動作: - proxy_groupid が 0 でない場合に必須
local_port	string	接続先のローカルプロキシポート番号。 ユーザーマクロがサポートされています。 デフォルト: 10051。 プロパティの動作: - proxy_groupid が 0 でない場合にサポート
operating_mode	integer	プロキシのタイプ。 可能な値: 0 - アクティブプロキシ; 1 - パッシブプロキシ。 プロパティの動作: - 作成操作時に必須
description	text	プロキシの説明。
lastaccess	timestamp	プロキシが最後にサーバーに接続した時刻。 プロパティの動作: - 読み取り専用
address	string	接続先の IP アドレスまたは DNS 名。 ユーザーマクロがサポートされています。 プロパティの動作: - Zabbix プロキシの動作モードがパッシブの場合に必須
port	string	接続先のポート番号。 ユーザーマクロがサポートされています。 デフォルト: 10051。 プロパティの動作: - Zabbix プロキシの動作モードがパッシブの場合にサポート
allowed_addresses	string	アクティブ Zabbix プロキシのカンマ区切りの IP アドレスまたは DNS 名。

プロパティ	型	説明
tls_connect	integer	ホストへの接続。 可能な値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	ホストからの接続。 可能なビットマップ値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドです。可能なビットマップ値の任意の合計が許容されます（例: PSK と証明書の場合は 6）。
tls_issuer	string	証明書発行者。
tls_subject	string	証明書サブジェクト。
tls_psk_identity	string	PSK 識別子；1 つの PSK（自動登録、ホスト、プロキシ間で）とだけペアにする必要があります。 PSK 識別子には機密情報を含めないでください。これはネットワーク上で暗号化されずに送信され、受信者にどの PSK を使用するかを通知します。 プロパティの動作: - 書き込み専用 - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合に必須
tls_psk	string	事前共有鍵（PSK）；少なくとも 32 桁の 16 進数である必要があります。 プロパティの動作: - 書き込み専用 - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合に必須
custom_timeouts	integer	プロキシレベルでグローバルなアイテムタイムアウトを上書きするかどうか。 可能な値: 0 - *(デフォルト)* グローバル設定を使用; 1 - タイムアウトを上書き。
timeout_zabbix_agent timeout_simple_check timeout_snmp_agent timeout_external_check timeout_db_monitor timeout_http_agent timeout_ssh_agent timeout_telnet_agent timeout_script timeout_browser	string	処理に `timeout_*` 秒以上かけない。 秒またはサフィックス付きの時間単位（例: 30s, 1m）を受け付けます。 ユーザーマクロも使用可能です。 許容値範囲: 1-600s。 デフォルト: ””。 プロパティの動作: - `custom_timeouts` が 1 に設定されている場合に必須。

プロパティ	型	説明
version	integer	プロキシのバージョン。 3 つの部分からなる Zabbix のバージョン番号で、各部分に 2 桁の 10 進数が使用されます。例: バージョン 6.4.1 の場合は 60401、バージョン 7.0.2 の場合は 70002 など。 0 - 不明なプロキシバージョン。 プロパティの動作: - 読み取り専用
compatibility	integer	Zabbix サーバーバージョンと比較したプロキシのバージョン。 可能な値: 0 - 未定義; 1 - 現行バージョン（プロキシとサーバーのメジャーバージョンが同じ）; 2 - 古いバージョン（プロキシバージョンがサーバーバージョンより古いが、一部サポートされている）; 3 - サポートされていないバージョン（プロキシバージョンがサーバーの前回の LTS リリースバージョンより古い、またはサーバーのメジャーバージョンがプロキシのメジャーバージョンより古い）。 プロパティの動作: - 読み取り専用
state	integer	プロキシの状態。 可能な値: 0 - 不明; 1 - オフライン; 2 - オンライン。 プロパティの動作: - 読み取り専用

proxy.create

説明

`object proxy.create(object/array proxies)`

このメソッドは、新しいプロキシを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 作成するプロキシ。

proxy の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合は、現在のプロキシに再割り当てされます。 ホストには `hostid` プロパティのみが定義されている必要があります。

戻り値

(`object`) `proxyids` プロパティの下に作成されたプロキシの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシの順序と一致します。

例

アクティブプロキシの作成

アクションプロキシ”Active proxy” を作成し、それによって監視されるホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

パッシブプロキシの作成

パッシブプロキシ”Passive proxy” を作成し、それによって監視される 2 つのホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Passive proxy",
        "operating_mode": "1",
        "address": "127.0.0.1",
        "port": "10051",
        "hosts": [
            {
                "hostid": "10192"
            },
            {
                "hostid": "10139"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
```

```
        ]
    },
    "id": 1
}
```

プロキシを作成して、プロキシグループへ追加

アクティブプロキシ”Active proxy” を作成し、ID”1” のプロキシグループに追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

参照

- Host
- Proxy group

ソース

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

説明

`object proxy.delete(array proxies)`

このメソッドは、プロキシを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するプロキシの ID。

戻り値

(object) 削除されたプロキシの ID を `proxyids` プロパティに含むオブジェクトを返します。

例

複数のプロキシの削除

2 つのプロキシを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

ソース

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

説明

`integer/array proxy.get(object parameters)`

このメソッドは、指定されたパラメータに従ってプロキシを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
proxyids	ID/array	指定した ID のプロキシのみを返します。
proxy_groupids	ID/array	指定したプロキシグループに属するプロキシのみを返します。
selectAssignedHosts	query	プロキシに割り当てられているホストを持つassignedHostsプロパティを返します。 count をサポートします。
selectHosts	query	プロキシによって監視されているホストを持つhostsプロパティを返します。 count をサポートします。
selectProxyGroup	query	プロキシグループオブジェクトを持つproxyGroupプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値：proxyid、name、operating_mode。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	

パラメータ	型	説明
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

すべてのプロキシを取得する

設定されているすべてのプロキシとそのインターフェースを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "11",
            "name": "Active proxy",
            "proxy_groupid": "0",
            "local_address": "",
            "local_port": "10051",
            "operating_mode": "0",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "0",
            "timeout_zabbix_agent": "",
            "timeout_simple_check": "",
            "timeout_snmp_agent": "",
            "timeout_external_check": "",
            "timeout_db_monitor": "",
            "timeout_http_agent": "",
            "timeout_ssh_agent": "",
            "timeout_telnet_agent": "",
            "timeout_script": "",
            "last_access": "1693391880",
            "version": "70000",
```

```
            "compatibility": "1",
            "state": "1"
        },
        {
            "proxyid": "12",
            "name": "Passive proxy",
            "proxy_groupid": "1",
            "local_address": "127.0.0.1",
            "local_port": "10051",
            "operating_mode": "1",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "1",
            "timeout_zabbix_agent": "5s",
            "timeout_simple_check": "5s",
            "timeout_snmp_agent": "5s",
            "timeout_external_check": "5s",
            "timeout_db_monitor": "5s",
            "timeout_http_agent": "5s",
            "timeout_ssh_agent": "5s",
            "timeout_telnet_agent": "5s",
            "timeout_script": "5s",
            "lastaccess": "1693391875",
            "version": "60400",
            "compatibility": "2",
            "state": "2"
        }
    ],
    "id": 1
}
```


こちらもご参照ください

- ホスト
- プロキシグループ

ソース

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

説明

`object proxy.update(object/array proxies)`

このメソッドは、既存のプロキシを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object/array`) 更新されるプロキシプロパティ。

`proxyid` プロパティはプロキシごとに定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

proxy の標準プロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合は、現在のプロキシに再割り当てされます。 ホストには `hostid` プロパティのみが定義されている必要があります。

戻り値

(object) `proxyids` プロパティに更新されたプロキシの ID を含むオブジェクトを返します。

例

プロキシが監視するホストの変更

指定された 2 つのホストを監視するようにプロキシを更新します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            },
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシステータスの変更

プロキシをアクティブプロキシに変更し、名前を”Active proxy” に変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシをプロキシグループへ追加

ID”5” のプロキシを更新し、ID””1 のプロキシループに追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

参照

- Host
- Proxy group

ソース

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Proxy group

このクラスは、プロキシグループを操作するためのものです。

オブジェクトリファレンス:

- Proxy group

利用可能なメソッド:

- proxygroup.create - 新しいプロキシグループを作成する
- proxygroup.delete - プロキシグループを削除する
- proxygroup.get - プロキシグループを取得する
- proxygroup.update - プロキシグループを更新する

Proxy group オブジェクト

次のオブジェクトは、proxygroupAPI に直接関連しています。

プロキシグループ

プロキシグループオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
proxy_groupid	ID	プロキシグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	プロキシグループの名前。 プロパティの動作: - 作成操作では必須
description	text	プロキシグループの説明。
failover_delay	string	プロキシグループ内のプロキシがオンラインと見なされるために Zabbix サーバーと通信しなければならない期間。 時間のサフィックスがサポートされています（例: 30s, 1m）。 ユーザーマクロがサポートされています。 可能な値: 10s-15m。 デフォルト: 1m。
min_online	string	プロキシグループをオンラインに保つために必要なオンラインプロキシの最小数。 ユーザーマクロがサポートされています。 可能な値の範囲: 1-1000。 デフォルト: 1。
state	integer	プロキシグループの状態。 可能な値: 0 - 不明; 1 - オフライン; 2 - 復旧中; 3 - オンライン; 4 - 劣化中。 プロパティの動作: - 読み取り専用

proxygroup.create

説明

`object proxygroup.create(object/array proxyGroups)`

このメソッドは、新しいプロキシグループを作成します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(`object/array`) 作成するプロキシグループ。

このメソッドは、標準のプロキシグループプロパティを持つプロキシグループを受け付けます。

戻り値

(`object`) は、`proxy_groupids` プロパティに作成されたプロキシグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシグループの順序と一致します。

例

プロキシグループを作成する

カスタム設定でプロキシグループを作成します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::create() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

説明

`object proxygroup.delete(array proxyGroupIds)`

このメソッドではプロキシグループを削除できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(array) 消去するプロキシグループの ID

戻り値

(object) proxy_groupids プロパティの下にある、削除されたプロキシグループの ID を含むオブジェクトを返す。

例

複数のプロキシグループを削除する

2 つのプロキシグループを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::delete() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

説明

`integer/array proxygroup.get(object parameters)`

このメソッドは、指定したパラメーターに応じたプロキシグループを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

(`object`) 出力を定義するパラメータ

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
proxy_groupids	ID/配列	指定された ID を持つプロキシグループのみを返します。
proxyids	ID/配列	指定されたプロキシを含むプロキシグループのみを返します。
selectProxies	クエリ	プロキシグループに属するプロキシを含む`proxies` プロパティを返します。 `count` をサポートしています。
sortfield	文字列/配列	指定されたプロパティで結果を並べ替えます。 可能な値: `proxy_groupid`、`name`
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

全てのプロキシグループの取得

すべての構成済みのプロキシとプロキシグループを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxy_groupid": "1",
            "name": "Proxy group 1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "Proxy group 2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
    ],
```

```
    "id": 1
}
```

参照

- Proxy

ソース

CProxyGroup::get() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

説明

`object proxygroup.update(object/array proxies)`

このメソッドは、既存のプロキシグループを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 更新するプロキシグループプロパティ。

proxy_groupid プロパティはプロキシグループごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準プロキシグループプロパティを持つプロキシグループを受け入れます。

戻り値

(object) proxy_groupids プロパティの下にある、更新されたプロキシグループの ID を含むオブジェクトを返す。

例

オンラインプロキシの最小数を変更する

グループがオンラインになるために必要なオンラインプロキシの最小数を変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::update() in *ui/include/classes/api/services/CProxyGroup.php*.

Regular expression

このクラスは、グローバル正規表現を操作するためのものです。

オブジェクトリファレンス:

- Regular expression
- Expressions

利用可能なメソッド:

- regexp.create - 新しい正規表現を作成する
- regexp.delete - 正規表現を削除する
- regexp.get - 正規表現を取得する
- regexp.update - 正規表現を更新する

Regular expression オブジェクト

次のオブジェクトは、`regexp`API に直接関連しています。

正規表現

グローバル正規表現オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
regexpid	ID	正規表現の ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	正規表現の名前。 プロパティの動作: - 作成操作時に必須
test_string	string	テスト文字列。

式

式オブジェクトには以下のプロパティがあります。

プロパティ	型	説明
expression	string	正規表現。 プロパティの動作: - 必須
expression_type	integer	正規表現のタイプ。 可能な値: 0 - 文字列が含まれる; 1 - いずれかの文字列が含まれる; 2 - 文字列が含まれない; 3 - 結果が TRUE; 4 - 結果が FALSE。 プロパティの動作: - 必須
exp_delimiter	string	式の区切り文字。 デフォルト値: `","`。 可能な値: `","` または `"."`、または `"/"`。 プロパティの動作: - `expression_type` が「いずれかの文字列が含まれる」に設定されている場合にサポート

プロパティ	型	説明
case_sensitive	integer	大文字小文字の区別。 デフォルト値: 0。 可能な値: 0 - 大文字小文字を区別しない; 1 - 大文字小文字を区別する。

regexp.create

説明

`object regexp.create(object/array regularExpressions)`

このメソッドは、新しいグローバル正規表現を作成するためのものです。

Note:
このメソッドはスーパ管理者ユーザータイプのみが利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成する正規表現。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
expressions	array	Expressions オプション。 パラメータの動作: - 必須

戻り値

(object) regexpids プロパティの下で作成された正規表現の ID を含むオブジェクトを返します。

例

新しいグローバル正規表現の作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
      "name": "Storage devices for SNMP discovery",
      "test_string": "/boot",
      "expressions": [
        {
          "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$",
          "expression_type": "4",
          "case_sensitive": "1"
        }
      ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
```

```
            "16"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

説明

`object regexp.delete(array regexpids)`

このメソッドは、グローバル正規表現を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する正規表現の ID。

戻り値

(object) regexpids プロパティの下にある削除された正規表現の ID を含むオブジェクトを返します。

例

複数のグローバル正規表現の削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

説明

`integer/array regexp.get(object parameters)`

このメソッドは、指定されたパラメータに従ってグローバルな正規表現を取得することができます。

> **Note:**
> このメソッドはスーパ管理者のみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。
> 詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
regexpids	ID/array	指定した ID の正規表現のみを返します。
selectExpressions	query	expressionsプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: regexpid, name。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

グローバル正規表現の取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid", "name"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
```

```
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "expressions": [
        {
          "expression": "^Software Loopback Interface",
          "expression_type": "4"
        },
        {
          "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
          "expression_type": "4"
        },
        {
          "expression": "^NULL[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ll]o[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ss]ystem$",
          "expression_type": "4"
        },
        {
          "expression": "^Nu[0-9.]*$",
          "expression_type": "4"
        }
      ]
    }
  },
  "id": 1
}
```


ソース

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

説明

`object regexp.update(object/array regularExpressions)`

このメソッドは、既存のグローバル正規表現を更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新する正規表現のプロパティ。

各オブジェクトには regexpid プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
expressions	array	Expressions オプション。

戻り値

(object) regexpids プロパティの下にある更新された正規表現の ID を含むオブジェクトを返します。

例

ファイルシステムのディスカバリ用グローバル正規表現の更新

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

Report

このクラスは、定期レポートを操作するためのものです。

オブジェクトリファレンス:

- Report
- Users
- User groups

利用可能なメソッド:

- report.create - 新しい定期レポートを作成する

- report.delete - 定期レポートを削除する
- report.get - 定期レポートを取得する
- report.update - 定期レポートを更新する

Report オブジェクト

次のオブジェクトは、`report`API に直接関連しています。

レポート

レポートオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
reportid	ID	レポートの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
userid	ID	レポートを作成したユーザーの ID。 プロパティの動作: - 作成操作時に必須
name	string	レポートの一意の名前。 プロパティの動作: - 作成操作時に必須
dashboardid	ID	レポートの元となるダッシュボードの ID。 プロパティの動作: - 作成操作時に必須
period	integer	レポートが作成される期間。 可能な値: 0 - *(デフォルト)* 前日; 1 - 前週; 2 - 前月; 3 - 前年。
cycle	integer	レポートの繰り返しスケジュール。 可能な値: 0 - *(デフォルト)* 毎日; 1 - 毎週; 2 - 毎月; 3 - 毎年。
start_time	integer	レポートの送信準備を開始する時刻（秒単位）。 デフォルト: 0。

プロパティ	タイプ	説明
weekdays	integer	レポートを送信する曜日。 可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 このフィールドはビットマスクです。可能なビットマップ値の任意の合計が許容されます（例: 21 は月・水・金）。 デフォルト: 0。 プロパティの動作: - `cycle` が「毎週」に設定されている場合は必須
active_since	string	開始日。 可能な値: 空文字列 - *(デフォルト)* 指定なし（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の始まり（00:00:00）のタイムスタンプとして保存）。
active_till	string	終了日。 可能な値: 空文字列 - *(デフォルト)* 指定なし（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の終わり（23:59:59）のタイムスタンプとして保存）。
subject	string	レポートメッセージの件名。
message	string	レポートメッセージの本文。
status	integer	レポートが有効か無効か。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
description	text	レポートの説明。
state	integer	レポートの状態。 可能な値: 0 - *(デフォルト)* まだ処理されていないレポート; 1 - レポートが生成され、すべての受信者に正常に送信された; 2 - レポートの生成に失敗; "info" にエラー情報が含まれる; 3 - レポートは生成されたが、一部（またはすべて）の受信者への送信に失敗; "info" にエラー情報が含まれる。 プロパティの動作: - 読み取り専用
lastsent	timestamp	最後に正常に送信されたレポートの Unix タイムスタンプ。 プロパティの動作: - 読み取り専用
info	string	エラーの説明または追加情報。 プロパティの動作: - 読み取り専用

ユーザー

ユーザーオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
userid	ID	レポートを送信するユーザーの ID。 プロパティの動作: - 必須
access_userid	ID	レポートを生成するユーザーの ID。 0 - *(デフォルト)* 受信者によってレポートを生成する。
exclude	integer	ユーザーをメーリングリストから除外するかどうか。 可能な値: 0 - *(デフォルト)* 含める; 1 - 除外する。

ユーザーグループ

ユーザーグループオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
usrgrpid	ID	レポートを送信するユーザーグループの ID。 プロパティの動作: - 必須
access_userid	ID	レポートを生成するユーザーの ID。 0 - *(デフォルト)* 受信者によってレポートを生成します。

report.create

説明

`object report.create(object/array reports)`

このメソッドは新しい定期レポートを作成するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object/array`) 作成するスケジュールレポート。

標準のスケジュールレポートプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
users	object/array	レポートを送信するユーザー。 パラメーターの動作: - `user_groups` が設定されていない場合は必須
user_groups	object/array	レポートを送信するユーザーグループ。 パラメーターの動作: - `users` が設定されていない場合は必須

戻り値

(`object`) `reportids` プロパティの下で作成されたスケジュールされたレポートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスケジュール済みレポートの順序と一致します。

例

スケジュールレポートの作成

2021-04-01 から 2021-08-31 まで、毎週月曜日から金曜日の 12:00 に前週分の週次レポートを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.create",
    "params": {
        "userid": "1",
        "name": "Weekly report",
        "dashboardid": "1",
        "period": "1",
        "cycle": "1",
        "start_time": "43200",
        "weekdays": "31",
        "active_since": "2021-04-01",
        "active_till": "2021-08-31",
        "subject": "Weekly report",
        "message": "Report accompanying text",
        "status": "1",
        "description": "Report description",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

参照

- Users
- User groups

ソース

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

説明

`object report.delete(array reportids)`

このメソッドは、定期レポートを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する定期レポートの ID。

戻り値

(object) reportids プロパティの下にある削除された定期レポートの ID を含むオブジェクトを返します。

例

複数の定期レポートの削除

2 つの定期レポートの削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

説明

`integer/array report.get(object parameters)`

このメソッドは、指定されたパラメータに従ってスケジュールされたレポートを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 期待される出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
reportids	ID/array	指定したレポート ID のスケジュールレポートのみを返します。
expired	boolean	true の場合は期限切れのスケジュールレポートのみ、false の場合はアクティブなスケジュールレポートのみを返します。
selectUsers	query	レポートの送信先として設定されているユーザーのusersプロパティを返します。
selectUserGroups	query	レポートの送信先として設定されているユーザーグループのuser_groupsプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: reportid, name, status。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

レポートデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": [
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Weekly report",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Weekly report",
```

```
            "message": "Report accompanying text",
            "status": "1",
            "description": "Report description",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
            "name": "Monthly report",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monthly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


参照

- Users
- User groups

ソース

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

説明

`object report.update(object/array reports)`

このメソッドは、既存のスケジュールレポートを更新するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するスケジュールレポートのプロパティ。

各スケジュールレポートには reportid プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のスケジュールレポートプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
users	object/array	スケジュールレポートに割り当てられている現在のユーザーを置き換えるユーザー。 パラメータの動作: - user_groups が設定されていない場合は必須
user_groups	object/array	スケジュールレポートに割り当てられている現在のユーザーグループを置き換えるユーザーグループ。 パラメータの動作: - users が設定されていない場合は必須

戻り値

(object) reportids プロパティの下で更新されたスケジュールされたレポートの ID を含むオブジェクトを返します。

例

スケジュールレポートの無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

参照

- Users

- User groups

ソース

CReport::update() in *ui/include/classes/api/services/CReport.php*.

Role

このクラスは、ユーザーロールを操作するためのものです。

オブジェクトリファレンス:

- Role
- Role rules
  - UI element
  - Service
  - Service tag
  - Module
  - Action

利用可能なメソッド:

- role.create - 新しいユーザーロールを作成する
- role.delete - ユーザーロールを削除する
- role.get - ユーザーロールを取得する
- role.update - ユーザーロールを更新する

Role オブジェクト

次のオブジェクトは、`roleAPI` に直接関連しています。

ロール

ロールオブジェクトには以下のプロパティがあります:

プロパティ	タイプ	説明
roleid	ID	ロールの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ロールの名前。 プロパティの動作: - 作成操作では必須
type	integer	ユーザータイプ。 可能な値: 1 - *(デフォルト)* ユーザー; 2 - 管理者; 3 - スーパ管理者。 プロパティの動作: - 作成操作では必須
readonly	integer	ロールが読み取り専用かどうか。 可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。 プロパティの動作: - 読み取り専用

ロールルール

ロールルールオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
ui	array	UI 要素オブジェクトの配列。
ui.default_access	integer	新しい UI 要素へのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
services.read.mode	integer	サービスへの読み取り専用アクセス。 可能な値: 0 - `services.read.list` で指定された、または `services.read.tag` で一致したサービスへの読み取り専用アクセス; 1 - *(デフォルト)* すべてのサービスへの読み取り専用アクセス。
services.read.list	array	サービスオブジェクトの配列。 指定されたサービス（子サービスを含む）に対して、ユーザーロールに読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み書きアクセスを上書きしません。 プロパティの動作: - `services.read.mode` が"0" に設定されている場合にサポート
services.read.tag	object	サービスタグオブジェクトの配列。 タグで一致したサービス（子サービスを含む）に対して、ユーザーロールに読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み書きアクセスを上書きしません。 プロパティの動作: - `services.read.mode` が"0" に設定されている場合にサポート
services.write.mode	integer	サービスへの読み書きアクセス。 可能な値: 0 - *(デフォルト)* `services.write.list` で指定された、または `services.write.tag` で一致したサービスへの読み書きアクセス; 1 - すべてのサービスへの読み書きアクセス。
services.write.list	array	サービスオブジェクトの配列。 指定されたサービス（子サービスを含む）に対して、ユーザーロールに読み書きアクセスが付与されます。読み書きアクセスは、サービスへの読み取り専用アクセスを上書きします。 プロパティの動作: - `services.write.mode` が"0" に設定されている場合にサポート
services.write.tag	object	サービスタグオブジェクトの配列。 タグで一致したサービス（子サービスを含む）に対して、ユーザーロールに読み書きアクセスが付与されます。読み書きアクセスは、サービスへの読み取り専用アクセスを上書きします。 プロパティの動作: - `services.write.mode` が"0" に設定されている場合にサポート
modules	array	モジュールオブジェクトの配列。
modules.default_access	integer	新しいモジュールへのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

プロパティ	型	説明
api.access	integer	API へのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
api.mode	integer	`api` プロパティにリストされた API メソッドの扱いモード。 可能な値: 0 - *(デフォルト)* 拒否リスト; 1 - 許可リスト。
api	array	API メソッドの配列。
actions	array	アクションオブジェクトの配列。
actions.default_access	integer	新しいアクションへのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

UI エレメント

UI エレメントオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	UI エレメントの名前。 Roleオブジェクトの type が「User」、「Admin」または「Super admin」に設定されている場合の値: monitoring.dashboard - ダッシュボード; monitoring.problems - 監視 → 障害; monitoring.hosts - 監視 → ホスト; monitoring.latest_data - 監視 → 最新データ; monitoring.maps - 監視 → マップ; services.services - サービス → サービス; services.sla_report - サービス → *SLA* レポート; inventory.overview - インベントリ → 概要; inventory.hosts - インベントリ → ホスト; reports.availability_report - レポート → 可用性レポート; reports.top_triggers - レポート → トリガートップ *100*。 Roleオブジェクトの type が「Admin」または「Super admin」に設定されている場合の値: monitoring.discovery - 監視 → ディスカバリ; services.sla - サービス → *SLA*; reports.scheduled_reports - レポート → スケジュールレポート; reports.notifications - レポート → 通知; configuration.template_groups - データ収集 → テンプレートグループ; configuration.host_groups - データ収集 → ホストグループ; configuration.templates - データ収集 → テンプレート; configuration.hosts - データ収集 → ホスト; configuration.maintenance - データ収集 → メンテナンス; configuration.discovery - データ収集 → ディスカバリ; configuration.trigger_actions - アラート → アクション → トリガーアクション; configuration.service_actions - アラート → アクション → サービスアクション; configuration.discovery_actions - アラート → アクション → ディスカバリアクション; configuration.autoregistration_actions - アラート → アクション → 自動登録アクション; configuration.internal_actions - アラート → アクション → 内部アクション。 Roleオブジェクトの type が「Super admin」に設定されている場合の値: reports.system_info - レポート → システム情報; reports.audit - レポート → 監査ログ; reports.action_log - レポート → アクションログ; configuration.event_correlation - データ収集 → イベント相関; administration.media_types - アラート → メディアタイプ; administration.scripts - アラート → スクリプト; administration.user_groups - ユーザー → ユーザーグループ; administration.user_roles - ユーザー → ユーザーロール; administration.users - ユーザー → ユーザー; administration.api_tokens - ユーザー → *API* トークン; administration.authentication - ユーザー → 認証; administration.general - 管理 → 一般; administration.audit_log - 管理 → 監査ログ; administration.housekeeping - 管理 → ハウスキーピング; administration.proxy_groups - 管理 → プロキシグループ; administration.proxies - 管理 → プロキシ; administration.macros - 管理 → マクロ; administration.queue - 管理 → キュー。 プロパティの動作: - 必須

プロパティ	タイプ	説明
status	integer	UI エレメントへのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

サービス

プロパティ	タイプ	説明
serviceid	ID	サービスの ID。 プロパティの動作: - 必須

サービスタグ

プロパティ	タイプ	説明
tag	string	タグ名。 空文字列を指定した場合、サービスタグはサービスのマッチングに使用されません。 プロパティの動作: - 必須
value	string	タグ値。 値が指定されていないか空文字列が指定されている場合、タグ名のみがサービスのマッチングに使用されます。

モジュール

モジュールオブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
moduleid	ID	モジュールの ID。 プロパティの動作: - 必須
status	integer	モジュールへのアクセスが有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

アクション

アクションオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	アクションの名前。 ロールオブジェクトの `type` が"User"、"Admin"、"Super admin" に設定されている場合の値: `edit_dashboards` - ダッシュボードの作成と編集; `edit_maps` - マップの作成と編集; `add_problem_comments` - 障害コメントの追加; `change_severity` - 障害の深刻度の変更; `acknowledge_problems` - 障害の確認; `suppress_problems` - 障害の抑制; `close_problems` - 障害のクローズ; `execute_scripts` - スクリプトの実行; `manage_api_tokens` - API トークンの管理; `edit_own_media` - 自分のメディアの作成/編集を許可。 ロールオブジェクトの `type` が"Admin" または"Super admin" に設定されている場合の値: `edit_maintenance` - メンテナンスの作成と編集; `manage_scheduled_reports` - スケジュールレポートの管理, `manage_sla` - SLA の管理。 ロールオブジェクトの `type` が"User" または"Admin" に設定されている場合の値: `invoke_execute_now` - ホストに対して読み取り権限のみを持つユーザーがアイテムチェックを実行できるようにします。 ロールオブジェクトの `type` が"Super admin" に設定されている場合の値: `edit_user_media` - ユーザーのメディアの作成/編集を許可。 プロパティの動作: - 必須
status	integer	アクションの実行権限が有効かどうか。 可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

role.create

説明

`object role.create(object/array roles)`

このメソッドは新しいロールを作成します。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するロール。

標準のロールプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
rules	array	ロールに作成するロールルール。

戻り値

(object) roleids プロパティの下に作成されたロールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたロールの順序と一致します。

例

ロールの作成

タイプ「User」で 2 つの UI 要素へのアクセスを拒否したロールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.create",
    "params": {
        "name": "Operator",
        "type": "1",
        "rules": {
            "ui": [
                {
                    "name": "monitoring.hosts",
                    "status": "0"
                },
                {
                    "name": "monitoring.maps",
                    "status": "0"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

参照

- Role rules
- UI element
- Module
- Action

ソース

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

説明

`object role.delete(array roleids)`

このメソッドは、ユーザーの役割を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するユーザーの役割の ID。

戻り値

(object) roleids プロパティの下にある削除されたユーザーの役割の ID を含むオブジェクトを返します。

例

複数のユーザーの役割の削除

2 つのユーザーの役割を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

説明

`integer/array role.get(object parameters)`

このメソッドは、指定されたパラメータに従ってロールを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
roleids	ID/array	指定した ID のロールのみを返します。
selectRules	query	ロールルールを持つ`rules`プロパティを返します。
selectUsers	query	ロールが割り当てられているユーザーを持つ`users`プロパティを返します。 ユーザータイプに基づく制限については、`user.get`を参照してください。

パラメーター	型	説明
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `roleid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ロールデータの取得

”Super admin role” ロールデータとそのアクセスルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
```

```
        },
        {
            "name": "monitoring.latest_data",
            "status": "1"
        },
        {
            "name": "monitoring.maps",
            "status": "1"
        },
        {
            "name": "services.services",
            "status": "1"
        },
        {
            "name": "services.sla_report",
            "status": "1"
        },
        {
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
```

```
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
        },
        {
            "name": "configuration.discovery_actions",
            "status": "1"
        },
        {
            "name": "configuration.autoregistration_actions",
            "status": "1"
        },
        {
            "name": "configuration.internal_actions",
            "status": "1"
        },
        {
            "name": "reports.system_info",
            "status": "1"
        },
        {
            "name": "reports.audit",
            "status": "1"
        },
        {
            "name": "reports.action_log",
            "status": "1"
        },
        {
            "name": "configuration.event_correlation",
            "status": "1"
        },
        {
            "name": "administration.media_types",
            "status": "1"
        },
        {
            "name": "administration.scripts",
            "status": "1"
        },
        {
            "name": "administration.user_groups",
            "status": "1"
        },
        {
            "name": "administration.user_roles",
            "status": "1"
```

```
        },
        {
            "name": "administration.users",
            "status": "1"
        },
        {
            "name": "administration.api_tokens",
            "status": "1"
        },
        {
            "name": "administration.authentication",
            "status": "1"
        },
        {
            "name": "administration.general",
            "status": "1"
        },
        {
            "name": "administration.audit_log",
            "status": "1"
        },
        {
            "name": "administration.housekeeping",
            "status": "1"
        },
        {
            "name": "administration.proxy_groups",
            "status": "1"
        },
        {
            "name": "administration.proxies",
            "status": "1"
        },
        {
            "name": "administration.macros",
            "status": "1"
        },
        {
            "name": "administration.queue",
            "status": "1"
        }
    ],
    "ui.default_access": "1",
    "services.read.mode": "1",
    "services.read.list": [],
    "services.read.tag": {
        "tag": "",
        "value": ""
    },
    "services.write.mode": "1",
    "services.write.list": [],
    "services.write.tag": {
        "tag": "",
        "value": ""
    },
    "modules": [],
    "modules.default_access": "1",
    "api.access": "1",
    "api.mode": "0",
    "api": [],
    "actions": [
        {
```

```
            "name": "edit_dashboards",
            "status": "1"
        },
        {
            "name": "edit_maps",
            "status": "1"
        },
        {
            "name": "acknowledge_problems",
            "status": "1"
        },
        {
            "name": "suppress_problems",
            "status": "1"
        },
        {
            "name": "close_problems",
            "status": "1"
        },
        {
            "name": "change_severity",
            "status": "1"
        },
        {
            "name": "add_problem_comments",
            "status": "1"
        },
        {
            "name": "execute_scripts",
            "status": "1"
        },
        {
            "name": "manage_api_tokens",
            "status": "1"
        },
        {
            "name": "edit_maintenance",
            "status": "1"
        },
        {
            "name": "manage_scheduled_reports",
            "status": "1"
        },
        {
            "name": "manage_sla",
            "status": "1"
        },
        {
            "name": "invoke_execute_now",
            "status": "1"
        },
        {
            "name": "change_problem_ranking",
            "status": "1"
        },
        {
            "name": "edit_own_media",
            "status": "1"
        },
        {
            "name": "edit_user_media",
            "status": "1"
```

```
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```

参照

- Role rules
- User

ソース

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

説明

`object role.update(object/array roles)`

このメソッドは、既存のロールを更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 更新するロールのプロパティ。

各ロールには `roleid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他は変更されません。

標準のロールプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
rules	array	ロールに対して更新するアクセスルール。

戻り値

(object) `roleids` プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

スクリプト実行機能の無効化

ID "5" のロールを更新し、スクリプト実行機能を無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
```

```
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

API へのアクセス制限

ID が”5” のロールを更新し、すべての”create”、”update”、”delete” メソッドの呼び出しを拒否します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::update() in *ui/include/classes/api/services/CRole.php*.

Script

このクラスは、スクリプトを操作するためのものです。

オブジェクトリファレンス:

- Script
  - Webhook parameters
- Debug
  - Log entry

利用可能なメソッド:

- script.create - 新しいスクリプトを作成する
- script.delete - スクリプトを削除する
- script.execute - スクリプトを実行する
- script.get - スクリプトを取得する
- script.getscriptsbyevents - イベントのスクリプトを取得する
- script.getscriptsbyhosts - ホストのスクリプトを取得する
- script.update - スクリプトを更新する

Script オブジェクト

次のオブジェクトは、scriptAPI に直接関連しています。

スクリプト

スクリプトオブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
scriptid	ID	スクリプトの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	スクリプトの名前。 プロパティの動作: - 作成操作時に必須
type	integer	スクリプトタイプ。 scope が「アクション操作」に設定されている場合の値: 0 - スクリプト; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook。 scope が「手動ホストアクション」または「手動イベントアクション」に設定されている場合の値: 6 - URL。 プロパティの動作: - 作成操作時に必須
command	string	実行するコマンド。 プロパティの動作: - type が「スクリプト」「IPMI」「SSH」「TELNET」「Webhook」のいずれかに設定されている場合に必須
scope	integer	スクリプトのスコープ。 値: 1 - アクション操作; 2 - 手動ホストアクション; 4 - 手動イベントアクション。 プロパティの動作: - 作成操作時に必須

プロパティ	タイプ	説明
execute_on	integer	スクリプトを実行する場所。 値: 0 - Zabbix エージェントで実行; 1 - Zabbix サーバーで実行。グローバルスクリプトの実行が Zabbix サーバーで有効な場合のみサポート; 2 - *(デフォルト)* Zabbix サーバーまたはプロキシで実行。
menu_path	string	プロパティの動作: - `type` が「スクリプト」に設定されている場合にサポート スラッシュで区切られたフォルダ。ホストまたはイベントをクリックした際にフロントエンドでメニューのようなナビゲーションを形成します。
authtype	integer	プロパティの動作: - `scope` が「手動ホストアクション」または「手動イベントアクション」に設定されている場合にサポート SSH スクリプトタイプで使用される認証方式。 値: 0 - パスワード; 1 - 公開鍵。
username	string	プロパティの動作: - `type` が「SSH」に設定されている場合にサポート 認証に使用されるユーザー名。
password	string	プロパティの動作: - `type` が「SSH」または「TELNET」に設定されている場合に必須 パスワード認証の SSH スクリプトおよび TELNET スクリプトで使用されるパスワード。
publickey	string	プロパティの動作: - `type` が「SSH」かつ `authtype` が「パスワード」、または `type` が「TELNET」に設定されている場合にサポート 公開鍵認証の SSH スクリプトで使用される公開鍵ファイル名。
privatekey	string	プロパティの動作: - `type` が「SSH」かつ `authtype` が「公開鍵」に設定されている場合に必須 公開鍵認証の SSH スクリプトで使用される秘密鍵ファイル名。
port	string	プロパティの動作: - `type` が「SSH」かつ `authtype` が「公開鍵」に設定されている場合に必須 SSH および TELNET スクリプトで使用されるポート番号。
groupid	ID	プロパティの動作: - `type` が「SSH」または「TELNET」に設定されている場合にサポート スクリプトを実行できるホストグループの ID。 「0」に設定すると、すべてのホストグループでスクリプトが利用可能になります。 デフォルト: 0。

プロパティ	タイプ	説明
usrgrpid	ID	スクリプトの実行を許可するユーザーグループの ID。 「0」に設定すると、すべてのユーザーグループでスクリプトが利用可能になります。 デフォルト: 0。 プロパティの動作: - scope が「手動ホストアクション」または「手動イベントアクション」に設定されている場合にサポート
host_access	integer	スクリプトの実行に必要なホスト権限。 値: 2 - *(デフォルト)* 読み取り; 3 - 書き込み。 プロパティの動作: - scope が「手動ホストアクション」または「手動イベントアクション」に設定されている場合にサポート
confirmation	string	確認ポップアップのテキスト。 Zabbix フロントエンドからスクリプトを実行しようとした際にポップアップが表示されます。 プロパティの動作: - scope が「手動ホストアクション」または「手動イベントアクション」に設定されている場合にサポート
timeout	string	Webhook スクリプトの実行タイムアウト（秒）。時間のサフィックスがサポートされています（例: 30s, 1m）。 値: 1-60s。 デフォルト: 30s。 プロパティの動作: - type が「Webhook」に設定されている場合に必須
parameters	array	Webhook 入力パラメータの配列。 プロパティの動作: - type が「Webhook」に設定されている場合にサポート
description	string	スクリプトの説明。
url	string	ユーザー定義の URL。 プロパティの動作: - type が「URL」に設定されている場合に必須
new_window	integer	URL を新しいウィンドウで開く。 値: 0 - いいえ; 1 - *(デフォルト)* はい。 プロパティの動作: - type が「URL」に設定されている場合にサポート
manualinput	integer	スクリプトがユーザー入力を受け付けるかどうかを示します。 値: 0 - *(デフォルト)* 無効; 1 - 有効; プロパティの動作: - scope が「手動ホストアクション」または「手動イベントアクション」に設定されている場合にサポート

プロパティ	タイプ	説明
manualinput_prompt	string	手動入力のプロンプトテキスト。 プロパティの動作: - `manualinput` が「有効」に設定されている場合に必須
manualinput_validator	string	ユーザー入力を検証するための文字列フィールド。正規表現またはカンマ区切りの値のセットで構成されます。 プロパティの動作: - `manualinput` が「有効」に設定されている場合に必須
manualinput_validator_type	integer	期待されるユーザー入力のタイプを決定します。 値: 0 - *(デフォルト)* 文字列。manualinput_validator を正規表現として扱うことを示します; 1 - リスト。manualinput_validator をカンマ区切りの入力値リストとして扱うことを示します。 プロパティの動作: - `manualinput` が「有効」に設定されている場合にサポート
manualinput_default_value	string	ユーザー入力の自動入力用のデフォルト値。 プロパティの動作: - `manualinput_validator_type` が「文字列」に設定されている場合にサポート

Webhook パラメータ

Webhook スクリプトが呼び出されたときに渡されるパラメータは、以下のプロパティを持ちます。

プロパティ	タイプ	説明
name	string	パラメータ名。 プロパティの動作: - 必須
value	string	パラメータ値。マクロをサポート。

デバッグ

実行された Webhook スクリプトのデバッグ情報。debug オブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
logs	array	ログエントリの配列。
ms	string	スクリプト実行にかかったミリ秒単位の時間。

ログエントリ

ログエントリオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
level	integer	ログレベル
ms	string	ログエントリが追加されるまでにスクリプト実行後経過したミリ秒数
message	string	ログメッセージ

script.create

説明

`object script.create(object/array scripts)`

このメソッドは新しいスクリプトを作成するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受け入れます

戻り値

(object) scriptids プロパティの下で作成されたスクリプトの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスクリプトの順序と一致します。

例

Webhook スクリプトの作成

外部サービスに HTTP リクエストを送信する Webhook スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

SSH スクリプトの作成

ホスト上で実行でき、コンテキストメニューを持つ公開鍵認証付きの SSH スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

カスタムスクリプトの作成

サーバーを再起動するカスタムスクリプトを作成します。このスクリプトはホストへの書き込み権限を必要とし、ユーザーに手動入力を促します。入力が正常に送信されると、スクリプトはフロントエンドに確認メッセージを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
        "manualinput_default_value": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
```

```
    },
    "id": 1
}
```

URL タイプのスクリプト作成

同じウィンドウに表示され、確認テキストを含む、ホストスコープ用の URL タイプのスクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

手動入力が可能な URL 型スクリプト作成

新しいウィンドウで開き、手動入力が可能なイベントスコープ用の URL 型スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
```

```
    "id": 1
}
```

ソース

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

説明

`object script.delete(array scriptIds)`

このメソッドは、スクリプトを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するスクリプトの ID。

戻り値

(object) scriptids プロパティの下にある削除されたスクリプトの ID を含むオブジェクトを返します。

例

複数のスクリプトの削除

2 つのスクリプトを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

ソース

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

説明

`object script.execute(object parameters)`

このメソッドは、ホストまたはイベント上でスクリプトを実行することを可能にします。URL タイプのスクリプトは除きます。それらは実行できません。

Note:
このメソッドは、どのタイプのユーザーでも利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 実行するスクリプトの ID、ホストの ID、またはイベントの ID と manualinput 値を含むパラメーター。

パラメーター	タイプ	説明
scriptid	ID	実行するスクリプトの ID。 パラメーターの動作: - 必須
hostid	ID	スクリプトを実行するホストの ID。 パラメーターの動作: - eventid が設定されていない場合は必須
eventid	ID	スクリプトを実行するイベントの ID。 パラメーターの動作: - hostid が設定されていない場合は必須
manualinput	string	スクリプトを実行するためにユーザーが指定した値。 {MANUALINPUT} マクロを置き換えます。

戻り値

(object) スクリプト実行の結果を返します。

プロパティ	タイプ	説明
response	string	スクリプトが正常に実行されたかどうか。 可能な値 - success。
value	string	スクリプトの出力。
debug	object	Webhook スクリプトが実行された場合、debug オブジェクトを含みます。他のスクリプトタイプの場合は空のオブジェクトを含みます。

例

Webhook スクリプトの実行

外部サービスに HTTP リクエストを送信する Webhook スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
        "debug": {
            "logs": [
                {
                    "level": 3,
                    "ms": 480,
```

```
                    "message": "[Webhook Script] HTTP status: 200."
                }
            ],
            "ms": 495
        }
    },
    "id": 1
}
```


カスタムスクリプトの実行

ホストで”ping” スクリプトを実行します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt
        "debug": []
    },
    "id": 1
}
```


手動でカスタムスクリプトを実行する

ホスト上でコマンド”ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac” を使用して”ping” スクリプトを実行します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```


ソース

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

説明

`integer/array script.get(object parameters)`

このメソッドは、指定されたパラメータに従ってスクリプトを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object`) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
groupids	ID/array	指定したホストグループで実行できるスクリプトのみを返します。
hostids	ID/array	指定したホストで実行できるスクリプトのみを返します。
scriptids	ID/array	指定した ID のスクリプトのみを返します。
usrgrpids	ID/array	指定したユーザーグループのユーザーが実行できるスクリプトのみを返します。
selectHostGroups	query	スクリプトを実行できるホストグループを持つ`hostgroups`プロパティを返します。
selectHosts	query	スクリプトを実行できるホストを持つ`hosts`プロパティを返します。
selectActions	query	スクリプトが関連付けられているアクションを持つ`actions`プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `scriptid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのスクリプトを取得

設定されているすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
```

```
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
```

```
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    {
        "scriptid": "5",
        "name": "URL",
        "command": "",
        "host_access": "2",
        "usrgrpid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "Go to {HOST.NAME}?",
        "type": "6",
        "execute_on": "1",
        "timeout": "30s",
        "scope": "4",
        "port": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "menu_path": "",
        "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
        "new_window": "0",
        "manualinput": "0",
        "manualinput_prompt": "",
        "manualinput_validator": "",
        "manualinput_validator_type": "0",
        "manualinput_default_value": "",
        "parameters": []
    },
    {
        "scriptid": "6",
        "name": "URL with user input",
        "command": "",
        "host_access": "2",
        "usrgrpid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "Open zabbix page {MANUALINPUT}?",
        "type": "6",
        "execute_on": "1",
        "timeout": "30s",
        "scope": "2",
        "port": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "menu_path": "",
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": "0",
        "manualinput": "1",
        "manualinput_prompt": "Select a page to open:",
```

```
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```

参照

- Host
- Host group

ソース

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

説明

`object script.getscriptsbyevents(object parameters)`

このメソッドは、指定されたイベントで利用可能なすべてのスクリプト、またはスクリプト ID が指定されている場合は特定のスクリプトを取得します。manualinput が指定されている場合は、{MANUALINPUT} マクロを指定された値に置き換えます。

> **Note:**
> このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

Parameters

(オブジェクト/配列) メソッドは、以下のパラメータを持つオブジェクトまたはオブジェクトの配列を受け入れます。

パラメータ	タイプ	説明
eventid	ID	スクリプトを返すイベントの ID 一意である必要があります。 パラメータの動作: - 必須
scriptid	ID	返すスクリプトの ID
manualinput	文字列	ユーザーが指定した {MANUALINPUT} マクロの値

戻り値

(`object`) は、イベント ID をプロパティとして、利用可能なスクリプトの配列を値として持つオブジェクトを返します。スクリプト ID が指定されている場合、関連付けられた値は特定のスクリプトを含む配列です。

> **Note:**
> このメソッドは、`confirmation` テキスト、`manualinput prompt` テキスト、および `url` 内のマクロを自動的に展開します。
> manualinput パラメータが指定されている場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

イベント ID でスクリプトを取得する

イベント"632" および"614" に使用できるすべてのスクリプトを取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
          "eventid":  "632"
```

```
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
```

```
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
```

```
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ],
        "614": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "1",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
```

```
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ]
```

```
    },
    "id": 1
}
```

manualinput 値を持つ特定のスクリプトの情報を取得します。

イベント”632” の ID”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

ソース

CScript::getScriptsByEvents() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

説明

`object script.getscriptsbyhosts(object parameters)`

このメソッドは、指定したホストで利用可能なすべてのスクリプト、またはスクリプト ID が指定された場合は特定のスクリプトを取得するためのものです。manualinput が指定された場合、{MANUALINPUT} マクロを指定された値で置き換えます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) このメソッドは、以下のパラメータを持つオブジェクトまたはオブジェクトの配列を受け付けます。

パラメータ	タイプ	説明
hostid	ID	スクリプトを返すホストの ID。 一意でなければなりません。 パラメータの動作: - 必須
scriptid	ID	返すスクリプトの ID。
manualinput	string	ユーザーが指定した {MANUALINPUT} マクロ値。

戻り値

(object) ホスト ID をプロパティ、利用可能なスクリプトの配列を値とするオブジェクトを返します。スクリプト ID が指定された場合、関連する値は特定のスクリプトを含む配列となります。

> **Note:**
> このメソッドは、`confirmation` テキスト、`manualinput prompt` テキスト、`url` 内のマクロを自動的に展開します。
> manualinput パラメータが指定された場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

ホスト ID によるスクリプトの取得

ホスト"30079" と"30073" で利用可能なすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
```

```
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
```

```
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "30073": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
```

```
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
```

```
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
                "new_window": "1",
                "manualinput": "1",
                "manualinput_prompt": "Zabbix page to open:",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "2",
                "name": "Traceroute",
                "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```


manualinput 値を持つ特定のスクリプトの情報を取得します。

ホスト”30079” 上の ID ”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
```

```
        }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "4",
                "name": "Open Zabbix page",
                "command": "",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "Are you sure you want to open page dashboard.view?",
                "type": "6",
                "execute_on": "2",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
                "new_window": "1",
                "manualinput": "1",
                "manualinput_prompt": "Zabbix page to open:",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

ソース

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

説明

`object script.update(object/array scripts)`

このメソッドは、既存のスクリプトを更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) スクリプトのプロパティが更新されます。

scriptid プロパティはスクリプトごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。例外は type プロパティが 5（Webhook）からその他に変更された場合です。その場合は parameters プロパティはクリーンアップされます。

戻り値

(object) scriptids プロパティの下で更新されたスクリプトの ID を含むオブジェクトを返します。

例

スクリプトコマンドの変更

スクリプトのコマンドを"/bin/ping -c 10 {HOST.CONN} 2>&1" に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

スクリプトコマンドを変更して、手動入力を追加します

スクリプトのコマンドを"/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1" に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Specify the number of ICMP packets to send with the ping command",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CScript::update() in *ui/include/classes/api/services/CScript.php*.

Service

このクラスは、IT インフラストラクチャ/ビジネスのサービスを操作するためのものです。

オブジェクトリファレンス:

- Service
- Status rule
- Service tag
- Service alarm
- Problem tag

利用可能なメソッド:

- service.create - 新しいサービスを作成する
- service.delete - サービスを削除する
- service.get - サービスを取得する
- service.update - サービスを更新する

Service オブジェクト

次のオブジェクトは、`service`API に直接関連しています。

サービス

サービスオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
serviceid	ID	サービスの ID プロパティの動作: - 読み取り専用 - 更新操作に必須
アルゴリズム	整数	ステータス計算ルール子サービスが存在する場合にのみ適用されます。 可能な値: 0 - ステータスを OK に設定します。 1 - すべての子サービスに問題がある場合、最も重大な状態です。 2 - 子サービスの中で最も重大な状態です。 プロパティの動作: - 作成操作では 必須です。
name	文字列	サービスの名前 プロパティの動作: - 作成操作では 必須です。
sortorder	整数	ソートに使用するサービスの位置 可能な値: 0～999 プロパティの動作: - 作成操作では 必須です。
weight	整数	サービスの重み 可能な値: 0～1000000 デフォルト: 0

プロパティ	タイプ	説明
propagation_rule	整数	ステータス伝搬ルール 可能な値: 0 - *(デフォルト)* サービスステータスをそのまま伝搬します (変更なし) 1 - 伝搬されたステータスを指定された propagation_value だけ増加 (重大度 1 ～5 ずつ) 2 - 伝搬されたステータスを指定された propagation_value だけ減少 (重大度 1 ～5 ずつ) 3 - このサービスを無視 (ステータスは親サービスにまったく伝搬されません) 4 - 指定された propagation_value を使用して固定のサービスステータスを設定します。 プロパティの動作: - 必須 propagation_value が設定されている場合
propagation_value	整数	ステータス伝播値 propagation_rule が”0” または”3” に設定されている場合の可能な値: 0 - 未分類 propagation_rule が”1” または”2” に設定されている場合の可能な値: 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害 propagation_rule が”4” に設定されている場合の可能な値: -1 - OK; 0 - 分類されていません; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害 プロパティの動作: - 必須 if propagation_rule が設定されています
status	整数	サービスが正常状態か問題状態かを示します。 サービスが問題状態の場合、status は次のいずれかになります。 - 最も重大な問題の重大度 - 問題状態にある子サービスの最高ステータス サービスが正常状態の場合、status は -1 になります。 プロパティの動作: - 読み取り専用
description	文字列	サービスの説明
uuid	文字列	インポートされたサービスを既存のサービスにリンクするために使用される、Universal Unique Identifier（UUID）指定されていない場合は自動生成されます。
created_at	整数	サービスが作成された Unix タイムスタンプ

プロパティ	タイプ	説明
readonly	boolean	サービスへのアクセス 可能な値: 0 - 読み取り/書き込み; 1 - 読み取り専用 プロパティの動作: - 読み取り専用

ステータスルール

ステータスルールオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	(新しいステータス) ステータスを設定するための条件 可能な値: 0 - 少なくとも (N) 個の子サービスが (ステータス) ステータス以上である場合 1 - 少なくとも (N%) 個の子サービスが (ステータス) ステータス以上である場合 2 - (N) 個未満の子サービスが (ステータス) ステータス以下である場合 3 - (N%) 個未満の子サービスが (ステータス) ステータス以下である場合 4 - (ステータス) ステータス以上の子サービスの重みが少なくとも (W) である場合 5 - (ステータス) ステータス以上の子サービスの重みが少なくとも (N%) である場合 6 - (ステータス) ステータス以下の子サービスの重みが (W) 未満である場合 7 - (ステータス) ステータス以下の子サービスの重みが (N%) 未満である場合。 ここで: - N (W) は `limit_value`; - (ステータス) は `limit_status` です。 - (新しいステータス) は `new_status` です。 プロパティの動作: - 必須
limit_value	整数	制限値 可能な値: - N および W の場合: 1～100000; - N% の場合: 1～100 プロパティの動作: - 必須
limit_status	整数	制限ステータス 可能な値: -1 - OK; 0 - 分類なし; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害 プロパティの動作: - 必須

プロパティ	タイプ	説明
new_status	整数	新しいステータス値 可能な値: 0 - 分類なし; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害 プロパティの動作: - 必須

サービスタグ

サービスタグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
タグ	文字列	サービスタグ名 プロパティの動作: - 必須
値	文字列	サービスタグの値

サービスアラーム

Note:
サービスアラームは、Zabbix API 経由で直接作成、更新、または削除することはできません。

サービスアラームオブジェクトは、サービスの状態変化を表します。以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	サービスの状態変化が発生した時刻
値	整数	サービスのステータス 使用可能な値の一覧については、サービスの `status` プロパティ を参照してください。

問題タグ

問題タグを使用すると、サービスと問題イベントを関連付けることができます。問題タグオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	問題タグ名。 プロパティの動作: - 必須
operator	integer	マッピング条件の演算子。 可能な値: 0 - *(デフォルト)* 等しい; 2 - 含む。
value	string	問題タグの値。

service.create

説明

`object service.create(object/array services)`

このメソッドは新しいサービスを作成します。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

作成する（object/array）サービス

標準のサービスプロパティに加えて、このメソッドは以下のパラメーターを受け入れます。

パラメーター	タイプ	説明
children	array	サービスにリンクする子サービス 子サービスには、serviceid プロパティのみを定義する必要があります。
parents	array	サービスにリンクする親サービス 親サービスには、serviceid プロパティのみを定義する必要があります。
tags	array	サービスに対して作成されるサービスタグ
problem_tags	array	サービスに対して作成される問題タグ
status_rules	array	サービスに対して作成されるステータスルール

戻り値

（object）serviceids プロパティの下で作成されたサービスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたサービスの順序と一致します。

例

サービスの作成

少なくとも 1 つの子プロセスに問題がある場合、問題状態に切り替えるサービスを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

説明

`object service.delete(array serviceIds)`

このメソッドは、サービスを削除することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するサービスの ID。

戻り値

(object) serviceids プロパティの下にある削除されたサービスの ID を含むオブジェクトを返します。

例

複数のサービスの削除

2 つのサービスを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

説明

`integer/array service.get(object parameters)`

このメソッドは、指定されたパラメータに従ってサービスを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
serviceids	ID/array	指定した ID のサービスのみを返します。
parentids	ID/array	指定した親サービスにリンクされているサービスのみを返します。
deep_parentids	flag	すべての直接および間接の子サービスを返します。parentids と一緒に使用します。
childids	ID/array	指定した子サービスにリンクされているサービスのみを返します。
evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	object/array	指定したタグを持つサービスのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのサービスを返します。 可能なoperator値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
problem_tags	object/array	指定した問題タグを持つサービスのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのサービスを返します。 可能なoperator値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
without_problem_tags	flag	問題タグのないサービスのみを返します。
slaids	ID/array	特定の SLA にリンクされているサービスのみを返します。
selectChildren	query	子サービスを含む `children` プロパティを返します。 `count` をサポートします。
selectParents	query	親サービスを含む `parents` プロパティを返します。 `count` をサポートします。
selectTags	query	サービスタグを含む`tags`プロパティを返します。 `count` をサポートします。
selectProblemEvents	query	問題イベントオブジェクトの配列を含む `problem_events` プロパティを返します。 問題イベントオブジェクトは以下のプロパティを持ちます: `eventid` - (`ID`) イベント ID; `severity` - (`string`) 現在のイベントの深刻度; `name` - (`string`) 解決済みイベント名。 `count` をサポートします。
selectProblemTags	query	問題タグを含む`problem_tags`プロパティを返します。 `count` をサポートします。
selectStatusRules	query	ステータスルールを含む`status_rules`プロパティを返します。 `count` をサポートします。

パラメーター	タイプ	説明
selectStatusTimeline	object/array	指定した期間のサービス状態変化を含む `status_timeline` プロパティを返します。 フォーマット `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` は開始日(含む; 整数タイムスタンプ)、`period_to` は終了日（含まない; 整数タイムスタンプ）です。 指定した期間内の状態変化について、`start_value` プロパティと`alarms`配列を含むエントリの配列を返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	これらのパラメーターはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

（integer/array）次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのサービスの取得

すべてのサービスとその依存関係に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
```

```
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
            "status": "-1",
```

```
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


ソース

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

説明

`object service.update(object/array services)`

このメソッドは、既存のサービスを更新するためのものです。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

更新する（object/array）サービスプロパティ

serviceid プロパティは各サービスに対して定義する必要があります。その他のプロパティはオプションです。指定されたプロパティのみが更新され、その他のプロパティは変更されません。

標準のサービスプロパティ に加えて、このメソッドは以下のパラメータを受け入れます。

パラメータ	型	説明
children	array	現在の子サービスを置き換える子サービス 子サービスには serviceid プロパティのみを定義する必要があります。
parents	array	現在の親サービスを置き換える親サービス 親サービスには、serviceid プロパティのみが定義されている必要があります。
tags	array	現在のサービスタグを置き換えるサービスタグ
problem_tags	array	現在の問題タグを置き換える問題タグ
status_rules	array	現在のステータスルールを置き換えるステータスルール

戻り値

（object）serviceids プロパティの下で更新されたサービスの ID を含むオブジェクトを返します。

例

サービスの親の設定

ID ”3” のサービスを、ID ”5” のサービスの親にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

スケジュールされたダウンタイムの追加

ID ”4” のサービスに対して、毎週月曜日 22:00 から火曜日 10:00 までのダウンタイムを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

ソース

CService::update() in *ui/include/classes/api/services/CService.php*.

Settings

このクラスは、一般的な管理設定を操作するためのものです。

オブジェクトリファレンス:

- Settings

利用可能なメソッド:

- settings.get - 設定を取得する
- settings.update - 設定を更新する

Settings オブジェクト

次のオブジェクトは、settings API に直接関連しています。

設定

設定オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
default_lang	string	デフォルトのシステム言語。 デフォルト: en_US。
default_timezone	string	デフォルトのシステムタイムゾーン。 デフォルト: system - システムデフォルト。 サポートされているタイムゾーンの完全なリストはPHP ドキュメントを参照してください。
default_theme	string	デフォルトのテーマ。 指定可能な値: blue-theme - *(デフォルト)* Blue; dark-theme - Dark; hc-light - High-contrast light; hc-dark - High-contrast dark。
search_limit	integer	検索およびフィルタ結果の上限。 デフォルト: 1000。
max_overview_table_size	integer	データ概要およびトリガー概要ダッシュボードウィジェットの最大列数および行数。 デフォルト: 50。
max_in_table	integer	テーブルセル内に表示する要素の最大数。 デフォルト: 50。
server_check_interval	integer	Zabbix サーバーがダウンしている場合に警告を表示。 指定可能な値: 0 - 警告を表示しない; 10 - *(デフォルト)* 警告を表示。
work_period	string	稼働時間。 デフォルト: 1-5,09:00-18:00。
show_technical_errors	integer	技術的なエラー（PHP/SQL）を、スーパ管理者以外のユーザーおよびデバッグモードが有効なユーザーグループに属していないユーザーに表示するかどうか。 指定可能な値: 0 - *(デフォルト)* 技術的なエラーを表示しない; 1 - 技術的なエラーを表示。

プロパティ	タイプ	説明
history_period	string	最新データ、Web、データ概要ダッシュボードウィジェットで履歴データを表示する最大期間。 秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 24h。
period_default	string	タイムフィルターのデフォルト期間。 月および年のサポート付きで、サフィックス付きの秒または時間単位を受け付けます（30s, 1m, 2h, 1d, 1M, 1y）。 デフォルト: 1h。
max_period	string	タイムフィルターの最大期間。 月および年のサポート付きで、サフィックス付きの秒または時間単位を受け付けます（30s, 1m, 2h, 1d, 1M, 1y）。 デフォルト: 2y。
severity_color_0	string	「未分類」深刻度の 16 進カラーコード。 デフォルト: 97AAB3。
severity_color_1	string	「情報」深刻度の 16 進カラーコード。 デフォルト: 7499FF。
severity_color_2	string	「警告」深刻度の 16 進カラーコード。 デフォルト: FFC859。
severity_color_3	string	「軽度障害」深刻度の 16 進カラーコード。 デフォルト: FFA059。
severity_color_4	string	「重度障害」深刻度の 16 進カラーコード。 デフォルト: E97659。
severity_color_5	string	「致命的障害」深刻度の 16 進カラーコード。 デフォルト: E45959。
severity_name_0	string	「未分類」深刻度の名称。 デフォルト: Not classified。
severity_name_1	string	「情報」深刻度の名称。 デフォルト: Information。
severity_name_2	string	「警告」深刻度の名称。 デフォルト: Warning。
severity_name_3	string	「軽度障害」深刻度の名称。 デフォルト: Average。
severity_name_4	string	「重度障害」深刻度の名称。 デフォルト: High。
severity_name_5	string	「致命的障害」深刻度の名称。 デフォルト: Disaster。
custom_color	integer	カスタムイベントステータスカラーを使用。 指定可能な値: 0 - *(デフォルト)* カスタムイベントステータスカラーを使用しない; 1 - カスタムイベントステータスカラーを使用。
ok_period	string	OK トリガーの表示期間。 秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 5m。

プロパティ	タイプ	説明
blink_period	string	ステータス変更時のトリガー点滅期間。 秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 2m。
problem_unack_color	string	未確認の PROBLEM イベントの 16 進カラーコード。 デフォルト: CC0000。
problem_ack_color	string	確認済みの PROBLEM イベントの 16 進カラーコード。 デフォルト: CC0000。
ok_unack_color	string	未確認の RESOLVED イベントの 16 進カラーコード。 デフォルト: 009900。
ok_ack_color	string	確認済みの RESOLVED イベントの 16 進カラーコード。 デフォルト: 009900。
problem_unack_style	integer	未確認の PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示。
problem_ack_style	integer	確認済みの PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示。
ok_unack_style	integer	未確認の RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示。
ok_ack_style	integer	確認済みの RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示。
url	string	フロントエンド URL。
discovery_groupid	ID	自動検出されたホストが自動的に配置されるホストグループの ID。
default_inventory_mode	integer	デフォルトのホストインベントリモード。 指定可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
alert_usrgrpid	ID	データベースダウンアラームメッセージを送信するユーザーグループの ID。 ”0” に設定すると、アラームメッセージは送信されません。
snmptrap_logging	integer	一致しない SNMP トラップをログに記録。 指定可能な値: 0 - 一致しない SNMP トラップをログに記録しない; 1 - *(デフォルト)* 一致しない SNMP トラップをログに記録。
login_attempts	integer	ログインフォームがブロックされるまでのログイン失敗回数。 デフォルト: 5。
login_block	string	login_attempts フィールドで定義されたログイン失敗回数を超えた場合にログインフォームがブロックされる時間間隔。 秒またはサフィックス付きの時間単位を受け付けます。 デフォルト: 30s。

プロパティ	タイプ	説明
validate_uri_schemes	integer	URI スキームの検証。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
uri_valid_schemes	string	有効な URI スキーム。 デフォルト: http,https,ftp,file,mailto,tel,ssh。
x_frame_options	string	X-Frame-Options HTTP ヘッダー。 デフォルト: SAMEORIGIN。
iframe_sandboxing_enabled	integer	iframe サンドボックス化を使用。 指定可能な値: 0 - 使用しない; 1 - *(デフォルト)* 使用する。
iframe_sandboxing_exceptions	string	iframe サンドボックス化の例外。
connect_timeout	string	Zabbix サーバーとの接続タイムアウト。 指定可能な値の範囲: 1-30s。 デフォルト: 3s。 プロパティの動作: - 必須
socket_timeout	string	ネットワークのデフォルトタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 3s。 プロパティの動作: - 必須
media_type_test_timeout	string	メディアタイプテストのネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 65s。 プロパティの動作: - 必須
item_test_timeout	string	アイテムテストのネットワークタイムアウト。 指定可能な値の範囲: 1-600s。 デフォルト: 60s。 プロパティの動作: - 必須
script_timeout	string	スクリプト実行のネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 プロパティの動作: - 必須

プロパティ	タイプ	説明
report_test_timeout	string	スケジュールレポートテストのネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 プロパティの動作: - 必須
auditlog_enabled	integer	監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
auditlog_mode	integer	サーバー（システムユーザー）によって実行される低レベルディスカバリ、ネットワークディスカバリ、自動登録アクティビティの監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
ha_failover_delay	string	フェイルオーバー遅延（秒単位）。 デフォルト: 1m。 プロパティの動作: - 読み取り専用
geomaps_tile_provider	string	ジオマップタイルプロバイダー。 指定可能な値: `OpenStreetMap.Mapnik` - *(デフォルト)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery。
geomaps_tile_url	string	カスタム値の `geomaps_tile_url`、`geomaps_max_zoom`、`geomaps_attribution` を指定するには空文字列をサポート。 ジオマップタイル URL。 プロパティの動作: - `geomaps_tile_provider` が空文字列の場合にサポート
geomaps_max_zoom	integer	ジオマップの最大ズームレベル。 指定可能な値: 0-30。 プロパティの動作: - `geomaps_tile_provider` が空文字列の場合にサポート
geomaps_attribution	string	ジオマップの帰属テキスト。 プロパティの動作: - `geomaps_tile_provider` が空文字列の場合にサポート
vault_provider	integer	Vault プロバイダー。 指定可能な値: 0 - *(デフォルト)* HashiCorp Vault; 1 - CyberArk Vault。
proxy_secrets_provider	integer	シークレットマクロ値の解決方法: 指定可能な値: 0 - *(デフォルト)* サーバーのみ; 1 - サーバーとプロキシで独立して。

プロパティ	タイプ	説明
timeout_zabbix_agent	string	`timeout_*` 秒以内に処理を完了。 秒またはサフィックス付きの時間単位（例: 30s, 1m）を受け付けます。 ユーザーマクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: 3s。 `timeout_browser` のデフォルト: 60s。 プロパティの動作: - 必須
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

説明

`object settings.get(object parameters)`

このメソッドは、指定されたパラメータに従って設定オブジェクトを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	タイプ	説明
output	query	このパラメータはリファレンス解説で説明されています。

戻り値

(object) 設定オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
```

```
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```

ソース

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

説明

`object settings.update(object settings)`

このメソッドは、既存の共通設定を更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプのみが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新する設定プロパティ。

戻り値

(array) 更新されたパラメータ名の配列を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

ソース

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

SLA

このクラスは、IT インフラストラクチャとビジネスサービスのパフォーマンスを推定するために使用される SLA（サービスレベルアグリーメント）オブジェクトを操作するためのものです。

オブジェクトリファレンス:

- SLA
- SLA schedule
- SLA excluded downtime
- SLA service tag

利用可能なメソッド:

- sla.create - 新しい SLA を作成する
- sla.delete - SLA を削除する
- sla.get - SLA を取得する
- sla.getsli - SLA のサービスレベルインジケータ (SLI) データを取得する
- sla.update - SLA を更新する

SLA オブジェクト

次のオブジェクトは、`sla`(Service Level Agreement) API に直接関連しています。

SLA

SLA オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
slaid	ID	SLA の ID プロパティの動作: - 読み取り専用 - 更新操作に必須
name	文字列	SLA の名前
period	整数	プロパティの動作: - 作成操作に必須 SLA のレポート期間 有効な値: 0 - 毎日; 1 - 毎週; 2 - 毎月; 3 - 四半期ごと; 4 - 毎年
slo	浮動小数点	プロパティの動作: - 作成操作では 必須 許容可能なサービスレベル目標の最小値（パーセント単位）サービスレベル指標 (SLI) がこの値を下回ると、SLA は問題あり/未達成状態とみなされます。 有効な値: 0～100（小数点以下最大 4 桁）
effective_date	整数	プロパティの動作: - 作成操作では 必須 SLA の有効日 有効な値: UTC での日付タイムスタンプ

プロパティ	タイプ	説明
timezone	文字列	レポートのタイムゾーン例: `Europe/London`、`UTC` サポートされているタイムゾーンの完全なリストについては、PHP ドキュメントを参照してください。 プロパティの動作: - 作成操作に必須
status	整数	SLA のステータス 有効な値: 0 - *(デフォルト)* SLA 無効; 1 - SLA 有効
description	文字列	SLA の説明

SLA スケジュール

SLA スケジュールオブジェクトは、接続されたサービスが稼働状態であることが予定されている期間を定義します。以下のプロパティを持ちます。

プロパティ	タイプ	説明
period_from	整数	定期的な週次期間の開始時刻（その日を含む） 指定可能な値: 秒数（日曜日からカウント） プロパティの動作: - 必須
period_to	整数	定期的な週次期間の終了時刻（その日を含まない） 指定可能な値: 秒数（日曜日からカウント） プロパティの動作: - 必須

SLA excluded downtime

excluded downtime オブジェクトは、接続されたサービスが SLI に影響を与えることなく、正常に機能していないスケジュールを定義します。例えば、計画的なメンテナンスなどです。以下のプロパティがあります。

プロパティ	タイプ	説明
name	文字列	除外するダウンタイムの名前 プロパティの動作: - 必須
period_from	整数	除外するダウンタイムの開始時刻（開始時刻を含む） 使用可能な値: タイムスタンプ プロパティの動作: - 必須
period_to	整数	除外されたダウンタイムの終了時刻（含まない） 可能な値: タイムスタンプ プロパティの動作: - 必須

SLA サービスタグ

SLA サービスタグオブジェクトは、SLA の計算に含めるサービスをリンクします。次のプロパティがあります。

プロパティ	タイプ	説明
tag	string	SLA サービスタグ名。 プロパティの動作: - 必須
operator	integer	SLA サービスタグの演算子。 可能な値： 0 - (デフォルト) 等しい; 2 - 含む。
value	string	SLA サービスタグの値。

sla.create

説明

`object sla.create(object/array SLAs)`

このメソッドは新しい SLA オブジェクトを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

作成する (object/array) SLA オブジェクト

標準 SLA プロパティに加えて、このメソッドは以下のパラメータを受け入れます。

パラメータ	タイプ	説明
service_tags	array	SLA 用に作成されるSLA サービスタグ パラメータの動作: - 必須
schedule	array	SLA に対して作成されるSLA スケジュール 空のパラメータを指定すると、24 時間 365 日のスケジュールとして解釈されます。 デフォルト: 24 時間 365 日のスケジュール
excluded_downtimes	array	SLA に対して作成されるSLA 除外ダウンタイム

戻り値

(object) 作成された SLA の ID を slaids プロパティに含むオブジェクトを返します。返される ID の順序は、渡された SLA の順序と一致します。

例

SLA の作成

以下のために SLA エントリを作成するよう指示します: * SQL エンジン関連サービスの稼働時間の追跡; * 土曜日の最終時間を除くすべての平日のカスタムスケジュール; * 2022 年の最終日を有効日とする; * 7 月 4 日午前 0 時から 1 時間 15 分の計画停止; * SLA の週次レポート計算を有効にする; * 最小許容 SLO は 99.9995% とする。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Database Uptime",
            "slo": "99.9995",
            "period": "1",
```

```
            "timezone": "America/Toronto",
            "description": "Provide excellent uptime for main database engines.",
            "effective_date": 1672444800,
            "status": 1,
            "schedule": [
                {
                    "period_from": 0,
                    "period_to": 601200
                }
            ],
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::create()。

sla.delete

説明

`object sla.delete(array slaids)`

このメソッドは、SLA エントリを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除する SLA の ID。

戻り値

(object) 削除された SLA の ID を slaids プロパティに含むオブジェクトを返します。

例

複数の SLA の削除

2 つの SLA エントリを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::delete()。

sla.get

説明

`integer/array sla.get(object parameters)`

このメソッドを使用すると、指定されたパラメーターに従って SLA オブジェクトを取得できます。

> **Note:**
> このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくはユーザーの役割を参照してください。

パラメータ

(object) は、必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
slaids	ID/配列	指定された ID の SLA のみを返します。
serviceids	ID/配列	特定のサービスに一致する SLA のみを返します。
selectSchedule	クエリ	SLA スケジュールを含む schedule プロパティを返します。 count をサポートします。
selectExcludedDowntimes	クエリ	SLA から除外されるダウンタイムを含む excluded_downtimes プロパティを返します。 count をサポートします。

パラメータ	タイプ	説明
selectServiceTags	クエリ	SLA サービスタグを含む service_tags プロパティを返します。 count をサポートします。
sortfield	文字列/配列	指定されたプロパティで結果を並べ替えます。 使用可能な値: slaid、name、period、slo、effective_date、timezone、status、description
countOutput	ブール値	これらのパラメータについては、リファレンス解説 で説明されています。
editable	ブール値	
excludeSearch	ブール値	
filter	object	
limit	integer	
output	クエリ	
preservekeys	ブール値	
search	object	
searchByAny	ブール値	
searchWildcardsEnabled	ブール値	
sortorder	文字列/配列	
startSearch	ブール値	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

すべての SLA の取得

すべての SLA とそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.get",
    "params": {
        "output": "extend",
        "selectSchedule": ["period_from", "period_to"],
        "selectExcludedDowntimes": ["name", "period_from", "period_to"],
        "selectServiceTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
```

```
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    },
    "id": 1
}
```


ソース

ui/include/classes/api/services/CSla.php の CSla:get()。

sla.getsli

説明

`object sla.getsli(object parameters)`

このメソッドを使用すると、サービスレベルアグリーメント (SLA) のサービスレベル指標 (SLI) データを計算できます。

Note:
このメソッドは、あらゆるタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメーター

(object) SLA ID、レポート期間、およびオプションで SLI を計算するサービスの ID を含むパラメーター。

パラメーター	タイプ	説明
slaid	ID	可用性情報を返す SLA の ID。 パラメーターの動作: - 必須
period_from	timestamp	SLI をレポートする開始タイムスタンプ（含む）。 可能な値: Unix タイムスタンプ
period_to	timestamp	SLI をレポートする終了タイムスタンプ（含む）。 可能な値: Unix タイムスタンプ
periods	integer	レポートする期間数。 可能な値: 1-100
serviceids	ID/array	SLI を返すサービスの ID。

期間の分割

次の表は、パラメータの組み合わせに基づく返される期間スライスの配置を示しています。

Note:
返される期間は、最初に利用可能な期間（SLA の有効日付に基づく）より前にはならず、現在の期間を超えることもありません。

パラメータ			返される期間
period_from	**period_to**	**periods**	
-	-	-	現在の期間を含む直近 20 期間。
-	-	N	直近 N 期間。
-	指定	-	period_to の直前の 20 期間。
-	指定	N	period_to の直前の N 期間。
指定	-	-	period_from から始まる最初の 20 期間。
指定	-	N	period_from から始まる最初の N 期間。
指定	指定	-	指定範囲内の最大 100 期間。
指定	指定	N	指定範囲内の N 期間。

戻り値

(object) 計算結果を返します。

プロパティ	タイプ	説明
periods	array	報告された期間のリスト。 各報告期間は、以下で構成されるオブジェクトとして表されます。 - period_from - (timestamp) 報告期間の開始タイムスタンプ（含む）。 - period_to - (timestamp) 報告期間の終了タイムスタンプ（含まない）。 期間は period_from でソートされ、最も早い期間が最初に表示されます。
serviceids	array	報告された期間内のサービス ID のリスト。 リストのソート順は定義されていません。serviceids パラメータが sla.getsli メソッドに渡された場合でも同様です。
sli	array	各報告期間およびサービスごとの SLI データ（二次元配列として）。 periods プロパティのインデックスが sli プロパティの第一次元として使用されます。 serviceids プロパティのインデックスが sli プロパティの第二次元として使用されます。

SLI データ

報告された各期間およびサービスごとに返される SLI データは以下の通りです。

プロパティ	タイプ	説明
uptime	integer	スケジュールされた稼働時間中にサービスが _OK_ 状態であった時間（除外されたダウンタイムを除く）。
downtime	integer	スケジュールされた稼働時間中にサービスが _OK でない _ 状態であった時間（除外されたダウンタイムを除く）。
sli	float	稼働時間とダウンタイムに基づく SLI（総稼働時間のパーセント）。
error_budget	integer	SLI と SLO に基づくエラーバジェット（秒単位）。

プロパティ	タイプ	説明
excluded_downtimes	array	このレポート期間中の除外されたダウンタイムの配列。 各オブジェクトは以下のパラメータを含みます。 - name - (string) 除外されたダウンタイムの名前。 - period_from - (timestamp) 除外されたダウンタイムの開始タイムスタンプ（含む）。 - period_to - (timestamp) 除外されたダウンタイムの終了タイムスタンプ（含まない）。 除外されたダウンタイムは period_from でソートされ、最も早い期間が最初に表示されます。

例

日次 SLA の SLI 計算

ID「1」の SLA にリンクされているサービス ID「1」と「4」の SLI データを取得します。「1761861599」(2025 年 10 月 30 日 23:59:59 GMT+0200）までの単一期間のデータを取得します。SLA のレポート期間が日次であるため、SLI データは「1761775200」(2025 年 10 月 30 日 00:00:00 GMT+0200）から「1761861600」(2025 年 10 月 31 日 00:00:00 GMT+0200）まで取得されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599,
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
                            "period_from": 1761825600,
```

```
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```

月次 SLA の SLI 計算

ID が”5” の SLA にリンクされている ID が”50”、”60”、”70” のサービスの SLI データを取得します。”1635724800”（2021 年 11 月 1 日 00:00:00 UTC）から始まる 3 期間分のデータを取得します。SLA のレポート期間が月次であるため、以下の 3 か月分の SLI データが取得されます。

- ”1635724800”（2021 年 11 月 1 日 00:00:00 UTC）から”1638316800”（2021 年 12 月 1 日 00:00:00 UTC）まで
- ”1638316800”（2021 年 12 月 1 日 00:00:00 UTC）から”1640995200”（2022 年 1 月 1 日 00:00:00 UTC）まで
- ”1640995200”（2022 年 1 月 1 日 00:00:00 UTC）から”1643673600”（2022 年 2 月 1 日 00:00:00 UTC）まで

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": "1635724800"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
            50,
```

```
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
```

```
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


ソース

CSla::getSli() in *ui/include/classes/api/services/CSla.php*

sla.update

Description

`object sla.update(object/array slaids)`

このメソッドで、既存の SLA エントリを更新できます。

Note:
このメソッドは *Admin* および *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

Parameters

更新する（オブジェクト/配列）SLA プロパティ。

`slaid` プロパティは各 SLA ごとに定義する必要があり、その他のプロパティはオプションです。指定されたプロパティのみが更新され、その他のプロパティは変更されません。

標準 SLA プロパティに加えて、このメソッドは以下のパラメータを受け入れます。

パラメータ	タイプ	説明
service_tags	配列	SLA サービスタグを使用して、現在の SLA サービスタグを置き換えます。
schedule	array	SLA スケジュール を指定して、現在のスケジュールを置き換えます。 パラメータを空に指定すると、24 時間 365 日のスケジュールとして解釈されます。
excluded_downtimes	array	SLA 除外ダウンタイム を指定して、現在のスケジュールを置き換えます。

戻り値

（`object`）`slaids` プロパティに更新された SLA の ID を含むオブジェクトを返します。

例

サービスタグの更新

NoSQL 関連サービスの SLA ID "5" を月次で計算するように設定し、スケジュールや除外するダウンタイムは変更せず、SLO を 95% に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": [
        {
            "slaid": "5",
            "name": "NoSQL Database engines",
            "slo": "95",
            "period": 2,
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "redis"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mongodb"
                }
            ]
        }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA のスケジュール変更

ID "5" の SLA を 24 時間 365 日のスケジュールに切り替えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA の除外ダウンタイムの変更

2022 年 4 月 6 日に 4 時間の RAM アップグレードのためのダウンタイムを計画的に追加します。ただし、ID "5" の SLA については、7 月 4 日に既に計画されているソフトウェアアップグレードは維持します（新たに定義する必要があります）

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "excluded_downtimes": [
            {
                "name": "Software version upgrade rollout",
                "period_from": "1648760400",
                "period_to": "1648764900"
            },
            {
                "name": "RAM upgrade",
                "period_from": "1649192400",
                "period_to": "1649206800"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::update()。

Task

このクラスは、タスク (設定の再ロードなしでアイテムをチェックしたり、ローレベルディスカバリルールをチェックしたりするなど) を操作するためのものです。

オブジェクトリファレンス:

- Task
  - 'Execute now' request object
  - 'Refresh proxy configuration' request object
  - 'Diagnostic information' request object
    * Statistic request object
    * Statistic result object

利用可能なメソッド:

- task.create - 新しいタスクを作成する
- task.get - タスクを取得する

タスクオブジェクト

以下のオブジェクトは task API に直接関連しています。

タスク

タスクオブジェクトには以下のプロパティがあります:

プロパティ	タイプ	説明
taskid	ID	タスクの ID。 プロパティの動作: - 読み取り専用
type	integer	タスクのタイプ。 可能な値: 1 - 診断情報; 2 - プロキシ設定の更新; 6 - 今すぐ実行。 管理者およびユーザータイプのユーザーは「今すぐ実行」タスクのみ作成できます。 task.get は常に「7」(タスク実行サマリー) を返すことに注意してください。 プロパティの動作: - 必須

プロパティ	タイプ	説明
status	integer	タスクのステータス。 可能な値: 1 - 新しいタスク; 2 - タスク進行中; 3 - タスク完了; 4 - タスク期限切れ。 プロパティの動作: - 読み取り専用
clock	timestamp	タスクが作成された時刻。 プロパティの動作: - 読み取り専用
ttl	integer	タスクが期限切れになるまでの秒数。 プロパティの動作: - 読み取り専用
proxyid	ID	診断情報統計が収集されるプロキシの ID。 プロパティの動作: - type が「診断情報」または「プロキシ設定の更新」に設定されている場合にサポート
request	object	タスクタイプに応じたタスクリクエストオブジェクト: 「今すぐ実行」タスクのオブジェクトは以下で詳細に説明; 「プロキシ設定の更新」タスクのオブジェクトは以下で詳細に説明; 「診断情報」タスクのオブジェクトは以下で詳細に説明。 プロパティの動作: - 必須
result	object	診断情報タスクの結果オブジェクト。 結果がまだ準備できていない場合は NULL を含むことがあります。 結果オブジェクトは以下で詳細に説明。 プロパティの動作: - 読み取り専用

' 今すぐ実行' リクエストオブジェクト

' 今すぐ実行' タスクリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
itemid	ID	アイテムおよびローレベルディスカバリールールの ID。 *Admin* および *User* タイプのユーザーは、*read-write* の権限を持つホスト、または *read* 権限とロールに対して `invoke_execute_now` アクションが有効になっている場合に、アイテムを' 今すぐ実行' できます。ローレベルディスカバリー（LLD）ルールについても *Admin* タイプのユーザーに同様に適用されます。

' プロキシ設定の更新' リクエストオブジェクト

' プロキシ設定の更新' タスクリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
proxyids	配列	プロキシ ID

'Diagnostic information' リクエストオブジェクト

診断情報タスクリクエストオブジェクトには、以下のプロパティがあります。すべてのタイプのプロパティの統計リクエストオブジェクトについては、以下で詳細に説明しています。

プロパティ	タイプ	説明
historycache	object	ヒストリキャッシュ統計リクエスト。サーバーおよびプロキシで利用可能。
valuecache	object	アイテムキャッシュ統計リクエスト。サーバーで利用可能。
preprocessing	object	前処理マネージャ統計リクエスト。サーバーおよびプロキシで利用可能。
alerting	object	アラートマネージャ統計リクエスト。サーバーで利用可能。
lld	object	LLD マネージャ統計リクエスト。サーバーで利用可能。

統計リクエストオブジェクト

統計リクエストオブジェクトは、サーバー/プロキシの内部プロセスに関してどのタイプの情報を収集するかを定義するために使用されます。以下のプロパティがあります。

プロパティ	タイプ	説明
stats	query	返される統計オブジェクトのプロパティ。 各タイプの診断情報統計に利用可能なフィールドのリストは以下で詳細に説明されています。 デフォルト: `extend` は利用可能なすべての統計フィールドを返します。
top	object	返される統計値をソートおよび制限するためのオブジェクト。 各タイプの診断情報統計に利用可能なフィールドのリストは以下で詳細に説明されています。 例: { “source.alerts”: 10 }

各種診断情報リクエストタイプで利用可能な統計フィールドの一覧

各種診断情報リクエストプロパティでリクエスト可能な統計フィールドは以下の通りです。

診断タイプ	利用可能なフィールド	説明
historycache	items	キャッシュされたアイテム数
	values	キャッシュされた値の数
	memory	共有メモリの統計情報 (空き領域、使用中チャンク数、空きチャンク数、最大空きチャンクサイズ)
	memory.data	ヒストリデータキャッシュ共有メモリの統計情報
	memory.index	ヒストリインデックスキャッシュ共有メモリの統計情報
valuecache	items	キャッシュされたアイテム数
	values	キャッシュされた値の数
	memory	共有メモリの統計情報 (空き領域、使用中チャンク数、空きチャンク数、最大空きチャンクサイズ)
	mode	値キャッシュモード
preprocessing	values	キューイングされた値の数
	preproc.values	前処理ステップを持つキューイングされた値の数
alerting	alerts	キューイングされたアラートの数
lld	rules	キューイングされたルールの数
	values	キューイングされた値の数

各種診断情報リクエストで利用可能なソートフィールドのリスト

要求された情報をソートおよび制限するために、以下の統計フィールドを使用できます。

診断タイプ	利用可能なフィールド	タイプ
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

統計結果オブジェクト

統計結果オブジェクトは、タスクオブジェクトの `result` フィールドで取得されます。

プロパティ	タイプ	説明
status	integer	タスク結果のステータス。 可能な値: -1 - タスクの実行中にエラーが発生; 0 - タスク結果が作成された。 プロパティの動作: *- read-only*
data	string/object	特定の診断情報タスクの統計リクエストオブジェクトに従った結果。 タスクの実行中にエラーが発生した場合は、エラーメッセージ文字列が含まれます。

task.create

説明

`object task.create(object/array tasks)`

このメソッドはタスクを作成するためのものです。

> **Note:**
> このメソッドはすべてのタイプのユーザーが利用できます。ユーザーロールの設定でメソッドの呼び出し権限を取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するタスク。

このメソッドは標準タスクプロパティを持つタスクを受け付けます。

「今すぐ実行」タスクは、以下のタイプのアイテム/ディスカバリールールに対してのみ作成できます:

- Zabbix エージェント（パッシブ）
- シンプルチェック
- SNMP エージェント（v1/v2/v3）
- Zabbix 内部
- 外部チェック
- データベース監視
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 従属アイテム
- スクリプト
- ブラウザ

アイテム/ディスカバリールールが「従属アイテム」タイプの場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

戻り値

(`object`) `taskids` プロパティの下で作成されたタスクの ID を含むオブジェクトを返します。アイテムおよびローレベルディスカバリルールごとに 1 つのタスクが作成されます。返される ID の順序は、渡された `itemids` の順序と一致します。

例

タスクの作成

アイテムとローレベルディスカバリールールの「今すぐ実行」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

2 つのプロキシの「プロキシ設定のリフレッシュ」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

「診断情報」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 1,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "proxyid": 0
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

参考

- Task
- ' 今すぐ実行' リクエストオブジェクト
- ' プロキシ設定のリフレッシュ' リクエストオブジェクト
- ' 診断情報' リクエストオブジェクト

ソース

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

説明

`integer/array task.get(object parameters)`

このメソッドは、指定されたパラメータに従ってタスクを取得することができます。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	タイプ	説明
taskids	ID/array	指定した ID のタスクのみを返します。
output	query	これらのパラメータはリファレンス解説で説明されています。
preservekeys	boolean	

戻り値

(integer/array) オブジェクトの配列を返します

例

ID でタスクを取得

ID が"1" のタスクに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "taskid": "1",
            "type": "7",
            "status": "3",
            "clock": "1601039076",
            "ttl": "3600",
            "proxyid": null,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "result": {
                "data": {
                    "alerting": {
                        "alerts": 0,
                        "top": {
                            "media.alerts": []
                        },
                        "time": 0.000663
```

```
                    },
                    "lld": {
                        "rules": 0,
                        "values": 0,
                        "top": {
                            "values": []
                        },
                        "time": 0.000442
                    }
                },
                "status": "0"
            }
        }
    ],
    "id": 1
}
```


参考

- Task
- ' 今すぐ実行' リクエストオブジェクト
- ' プロキシ設定のリフレッシュ' リクエストオブジェクト
- ' 診断情報' リクエストオブジェクト

ソース

CTask::get() in *ui/include/classes/api/services/CTask.php*.

Template

このクラスは、テンプレートを操作するためのものです。

オブジェクトリファレンス:

- Template
- Template tag

利用可能なメソッド:

- template.create - 新しいテンプレートを作成する
- template.delete - テンプレートを削除する
- template.get - テンプレートを取得する
- template.massadd - テンプレートへ関連オブジェクトを追加する
- template.massremove - テンプレートから関連オブジェクトを削除する
- template.massupdate - テンプレートの関連オブジェクトを置換または削除する
- template.update - テンプレートを更新する

Template オブジェクト

次のオブジェクトは、`templateAPI` に直接関連しています。

テンプレート

テンプレートオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
templateid	ID	テンプレートの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
description	text	テンプレートの説明。
host	string	テンプレートの技術名。 プロパティの動作: - 作成操作では必須

プロパティ	タイプ	説明
name	string	テンプレートの表示名。 デフォルト: `host` プロパティの値。
readme	text	ホストウィザードで表示するテンプレート固有の設定手順。Markdown書式に対応。
uuid	string	インポートされたテンプレートを既存のテンプレートにリンクするためのユニバーサル一意識別子。指定がない場合は自動生成。
vendor_name	string	テンプレートベンダー名。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空にする必要があります。更新操作では、データベースに値がある場合は `vendor_version` を空にできます。
vendor_version	string	テンプレートベンダーバージョン。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空にする必要があります。更新操作では、データベースに値がある場合は `vendor_name` を空にできます。
wizard_ready	integer	ホストウィザードでテンプレートを選択可能にするかどうか。 指定可能な値: 0 - *(デフォルト)* 利用不可。 1 - 利用可能。

テンプレートタグ

テンプレートタグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	テンプレートタグ名。 プロパティの動作: - 必須
value	string	テンプレートタグの値。

template.create

説明

`object template.create(object/array templates)`

このメソッドは新しいテンプレートを作成します。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するテンプレート。

標準テンプレートプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
groups	object/array	テンプレートを追加するテンプレートグループ。 テンプレートグループには `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
tags	object/array	テンプレートタグ。

パラメータ	タイプ	説明
templates	object/array	テンプレートにリンクするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。
macros	object/array	テンプレートに作成するユーザーマクロ。

戻り値

(`object`) `templateids` プロパティの下で作成されたテンプレートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートの順序と一致します。

例

テンプレートの作成

タグ付きのテンプレートを作成し、このテンプレートに 2 つのテンプレートをリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.create",
    "params": {
        "host": "Linux template",
        "groups": {
            "groupid": 1
        },
        "templates": [
            {
                "templateid": "11115"
            },
            {
                "templateid": "11116"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.delete

説明

`object template.delete(array templateIds)`

このメソッドは、テンプレートを削除することができます。

テンプレートを削除すると、すべてのテンプレートエンティティ (アイテム、トリガー、グラフなど) が削除されます。テンプレートエンティティをホストに残しつつ、テンプレート自体を削除するにはtemplate.update、template.massupdate、host.update、host.massupdateのいずれかを使用して、ホストからテンプレートのリンクを解除します。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するテンプレートの ID。

戻り値

(object) templateids プロパティの下にある削除されたテンプレートの ID を含むオブジェクトを返します。

例

複数のテンプレートの削除

2 つのテンプレートを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

template.get

説明

`integer/array template.get(object parameters)`

このメソッドは、指定されたパラメータに従ってテンプレートを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
templateids	ID/array	指定したテンプレート ID を持つテンプレートのみを返します。
groupids	ID/array	指定したテンプレートグループに属するテンプレートのみを返します。
parentTemplateids	ID/array	指定したテンプレートにリンクされているテンプレートのみを返します。
hostids	ID/array	指定したホスト/テンプレートにリンクされているテンプレートのみを返します。
graphids	ID/array	指定したグラフを含むテンプレートのみを返します。
itemids	ID/array	指定したアイテムを含むテンプレートのみを返します。
triggerids	ID/array	指定したトリガーを含むテンプレートのみを返します。
with_items	flag	アイテムを持つテンプレートのみを返します。
with_triggers	flag	トリガーを持つテンプレートのみを返します。
with_graphs	flag	グラフを持つテンプレートのみを返します。
with_httptests	flag	Web シナリオを持つテンプレートのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値： 0 - *(デフォルト)* And/Or; 2 - Or.
tags	object/array	指定したタグを持つテンプレートのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのテンプレートを返します。 指定可能なoperator値： 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
selectTags	query	`tags`プロパティでテンプレートタグを返します。
selectDiscoveryRules	query	テンプレート LLD ルールを持つ `discoveryRules` プロパティを返します。`count` をサポートします。
selectHosts	query	`hosts`プロパティでテンプレートにリンクされているホストを返します。 `count` をサポートします。
selectTemplateGroups	query	`templategroups`プロパティでテンプレートが属するテンプレートグループを返します。
selectTemplates	query	`templates`プロパティで指定したテンプレートがリンクされているテンプレートを返します。 `count` をサポートします。
selectParentTemplates	query	`parentTemplates` プロパティで指定したテンプレートにリンクされているテンプレートを返します。 `count` をサポートします。
selectHttpTests	query	`httpTests`プロパティでテンプレートから Web シナリオを返します。 `count` をサポートします。
selectItems	query	`items`プロパティでテンプレートからアイテムを返します。 `count` をサポートします。
selectTriggers	query	`triggers`プロパティでテンプレートからトリガーを返します。 `count` をサポートします。
selectGraphs	query	`graphs`プロパティでテンプレートからグラフを返します。 `count` をサポートします。
selectMacros	query	`macros`プロパティでテンプレートからマクロを返します。

パラメーター	型	説明
selectDashboards	query	dashboardsプロパティでテンプレートからダッシュボードを返します。 count をサポートします。
selectValueMaps	query	テンプレートの値のマップを持つvaluemapsプロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます： selectTemplates - 結果は name でソートされます; selectHosts - host でソート; selectParentTemplates - host でソート; selectItems - name でソート; selectDiscoveryRules - name でソート; selectTriggers - description でソート; selectGraphs - name でソート; selectDashboards - name でソート。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値：hostid, host, name, status。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	discoveries プロパティでテンプレートから低レベルディスカバリを返します。 count をサポートします。 このクエリは非推奨です。代わりに selectDiscoveryRules を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

テンプレート名によるテンプレートの取得

"Linux" と"Windows" という 2 つのテンプレートのすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
```

```
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": "This is an official Linux template. It requires Zabbix agent 8.0 or newer.\r\n
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Overview\r\n\r\nThis is an official Linux template. It requires Zabbix agent 8.0
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": "This is an official Windows template. It requires Zabbix agent 8.0 or newer.\r
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
```

```
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Overview\r\n\r\nThis is an official Windows template. It requires Zabbix agent 8
        }
    ],
    "id": 1
}
```


テンプレートグループの取得

テンプレート”Linux by Zabbix エージェント” が所属するテンプレートグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Response:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```


テンプレートによるホストの取得

”10001”(*Linux by Zabbix* エージェント) テンプレートがリンクされているホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
```

```
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートタグによる検索

タグ”host-name” が”{HOST.NAME}” と等しいテンプレートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
```

```
    ],
    "id": 1
}
```

参照

- テンプレートグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース

ソース

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

説明

`object template.massadd(object parameters)`

このメソッドは、指定したテンプレートに複数の関連オブジェクトを同時に追加することができます。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 更新するテンプレートの ID とテンプレートに追加するオブジェクトを含むパラメーター。

このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
templates	object/array	更新するテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。 パラメーターの動作: - 必須
groups	object/array	指定したテンプレートを追加するテンプレートグループ。 テンプレートグループには `groupid` プロパティのみを定義する必要があります。
macros	object/array	指定したテンプレートに作成するユーザーマクロ。
templates_link	object/array	指定したテンプレートにリンクするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(object) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートにグループをリンクする

テンプレートグループ"2" を 2 つのテンプレートに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
```

```
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

2 つのテンプレートを 1 つのテンプレートにリンクする

テンプレート”10106” と”10104” をテンプレート”10073” にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
```

```
    "id": 1
}
```

関連項目

- template.update
- ホスト
- テンプレートグループ
- ユーザーマクロ

ソース

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

説明

`object template.massremove(object parameters)`

このメソッドは、複数のテンプレートから関連オブジェクトを削除することができます。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するテンプレートの ID と削除するオブジェクトを含むパラメータ。

パラメータ	タイプ	説明
templateids	ID/array	更新するテンプレートの ID。 パラメータの動作: - 必須
groupids	ID/array	指定したテンプレートを削除するテンプレートグループの ID。
macros	string/array	指定したテンプレートから削除するユーザーマクロの ID。
templateids_clear	ID/array	指定したテンプレートからリンク解除およびクリアするテンプレートの ID（上流）。
templateids_link	ID/array	指定したテンプレートからリンク解除するテンプレートの ID（上流）。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートをグループから削除する

2 つのテンプレートをグループ”2” から削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

ホストからテンプレートのリンクを解除

テンプレート”10085” からテンプレート”10106” と”10104” のリンクを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

参照

- template.update
- User macro

ソース

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.massupdate

説明

`object template.massupdate(object parameters)`

このメソッドは、複数のテンプレートに関連するオブジェクトを同時に置換または削除し、プロパティを更新することができます。

> **Note:**
> このメソッドは *Admin* および *Super admin* ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 更新するテンプレートの ID とテンプレートの置換対象オブジェクトを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
templates	object/array	更新するテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	テンプレートが属する現在のテンプレートグループを置き換えるテンプレートグループ。 テンプレートグループには `groupid` プロパティのみを定義する必要があります。
macros	object/array	指定したテンプレートの現在のユーザーマクロをすべて置き換えるユーザーマクロ。
templates_clear	object/array	指定したテンプレートからリンク解除およびクリアするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。
templates_link	object/array	現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(`object`) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

指定されたテンプレートからテンプレート”10091” のリンクを解除してクリアします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "templates_clear": [
            {
                "templateid": "10091"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
```

```
        ]
    },
    "id": 1
}
```

ユーザーマクロの置換

複数のテンプレートで、すべてのユーザーマクロを指定したユーザーマクロに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10074"
            },
            {
                "templateid": "10075"
            },
            {
                "templateid": "10076"
            },
            {
                "templateid": "10077"
            }
        ],
        "macros": [
            {
                "macro": "{$AGENT.TIMEOUT}",
                "value": "5m",
                "description": "Timeout after which agent is considered unavailable. Works only for agents
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10074",
            "10075",
            "10076",
            "10077"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- template.massadd
- テンプレートグループ
- ユーザーマクロ

ソース

CTemplate::massUpdate() in *ui/include/classes/api/services/CTemplate.php*.

template.update

説明

`object template.update(object/array templates)`

このメソッドは、既存のテンプレートを更新するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するテンプレートのプロパティ。

各テンプレートには `templateid` プロパティを定義する必要があり、他のプロパティはオプションです。指定されたプロパティのみが更新され、他は変更されません。

標準のテンプレートプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	テンプレートが属する現在のテンプレートグループを置き換えるテンプレートグループ。 テンプレートグループには `groupid` プロパティのみを定義する必要があります。
tags	object/array	現在のテンプレートタグを置き換えるテンプレートタグ。
macros	object/array	指定したテンプレートの現在のユーザーマクロを置き換えるユーザーマクロ。
templates	object/array	現在リンクされているテンプレートを置き換えるテンプレート。渡されていないテンプレートはアンリンクのみされます。 テンプレートには `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	指定したテンプレートからアンリンクしてクリアするテンプレート。 テンプレートには `templateid` プロパティのみを定義する必要があります。

戻り値

(object) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

標準テンプレートプロパティの変更

テンプレートの技術名を「Linux by Zabbix agent Custom」、表示名を「My template」、テンプレートの説明を更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "host": "Linux by Zabbix agent Custom",
        "name": "My template",
        "description": "This is a custom Linux template."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートグループの更新

指定したテンプレートのすべてのテンプレートグループを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートタグの更新

すべてのテンプレートタグを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートマクロの更新

すべてのテンプレートマクロを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートのリンクされたテンプレートの更新

指定したテンプレートからすべてのテンプレートのリンクを解除し (クリアせず)、別のテンプレートをリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10087"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
```

```
    },
    "id": 1
}
```

テンプレートのリンク解除とクリア

指定したリンクされたテンプレートからテンプレートのリンクを解除し、クリアします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates_clear": [
            {
                "templateid": "10087"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

Template dashboard

このクラスは、テンプレートダッシュボードを操作するためのものです。

オブジェクトリファレンス:

- Template dashboard
- Template dashboard page
    - Template dashboard widget
        - Template dashboard widget field

利用可能なメソッド:

- templatedashboard.create - 新しいテンプレートダッシュボートを作成する
- templatedashboard.delete - テンプレートダッシュボードを削除する
- templatedashboard.get - テンプレートダッシュボードを取得する
- templatedashboard.update - テンプレートダッシュボードを更新する

Template dashboard オブジェクト

次のオブジェクトは、`templatedashboard`API に直接関連しています。

テンプレートダッシュボード

テンプレートダッシュボードオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
dashboardid	ID	テンプレートダッシュボードの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	テンプレートダッシュボードの名前。 プロパティの動作: - 作成操作時に必須
templateid	ID	ダッシュボードが属するテンプレートの ID。 プロパティの動作: - 定数 - 作成操作時に必須
display_period	integer	デフォルトのページ表示期間（秒単位）。 指定可能な値: 10, 30, 60, 120, 600, 1800, 3600。 デフォルト: 30。
auto_start	integer	スライドショーの自動開始。 指定可能な値: 0 - スライドショーを自動開始しない; 1 - *(デフォルト)* スライドショーを自動開始する。
uuid	string	インポートされたテンプレートダッシュボードを既存のものにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。

テンプレートダッシュボードページ

テンプレートダッシュボードページオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
dashboard_pageid	ID	ダッシュボードページの ID。 プロパティの動作: - 読み取り専用
name	string	ダッシュボードページ名。 デフォルト: 空文字列
display_period	integer	ダッシュボードページの表示期間（秒単位）。 可能な値: 0, 10, 30, 60, 120, 600, 1800, 3600 デフォルト: 0（デフォルトのページ表示期間を使用）
widgets	array	テンプレートダッシュボードウィジェットオブジェクトの配列。

テンプレートダッシュボードウィジェット

テンプレートダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
type	string	ダッシュボードウィジェットのタイプ。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリーステータス; favgraphs - お気に入りグラフ; favmaps - お気に入りマップ; gauge - ゲージ; graph - グラフ (クラシック); graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホスト可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemnavigator - アイテムナビゲーター; item - アイテム値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; plaintext - プレーンテキスト; problemhosts - 問題のあるホスト; problems - 問題; problemsbysv - 深刻度別の問題; scatterplot - 散布図; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - トップホスト; topitems - トップアイテム; toptriggers - トップトリガー; trigover - トリガー概要; url - URL; web - Web 監視。 プロパティの動作: - 必須
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左端からの水平位置。 指定可能な値は 0 から 71 まで。
y	integer	ダッシュボードの上端からの垂直位置。 指定可能な値は 0 から 63 まで。
width	integer	ウィジェットの幅。 指定可能な値は 1 から 72 まで。
height	integer	ウィジェットの高さ。 指定可能な値は 1 から 64 まで。
view_mode	integer	ウィジェットの表示モード。 指定可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダーを非表示にする;
fields	array	テンプレートダッシュボードウィジェットフィールドオブジェクトの配列。

テンプレートダッシュボードウィジェットフィールド

テンプレートダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	ウィジェットフィールドのタイプ。 指定可能な値: 0 - 整数; 1 - 文字列; 4 - アイテム; 5 - アイテムプロトタイプ; 6 - グラフ; 7 - グラフプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 プロパティの動作: - 必須
name	string	ウィジェットフィールド名。 指定可能な値: ダッシュボードウィジェットフィールドを参照してください。なお、一部のホスト関連パラメータ（例: 障害ウィジェットのホストグループ、除外ホストグループ、ホスト、ホスト可用性ウィジェットのホストグループなど）は、テンプレートダッシュボードでウィジェットを設定する際には利用できません。これは、テンプレートダッシュボードがテンプレートがリンクされているホストからのみデータを表示するためです。 プロパティの動作: - 必須
value	mixed	タイプに応じたウィジェットフィールドの値。 指定可能な値: ダッシュボードウィジェットフィールドを参照してください。なお、一部のホスト関連パラメータ（例: 障害ウィジェットのホストグループ、除外ホストグループ、ホスト、ホスト可用性ウィジェットのホストグループなど）は、テンプレートダッシュボードでウィジェットを設定する際には利用できません。これは、テンプレートダッシュボードがテンプレートがリンクされているホストからのみデータを表示するためです。 プロパティの動作: - 必須

templatedashboard.create

説明

`object templatedashboard.create(object/array templateDashboards)`

このメソッドは新しいテンプレートダッシュボードを作成します。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(`object/array`) 作成するテンプレートダッシュボード。

標準のテンプレートダッシュボードプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
pages	array	ダッシュボードに作成するテンプレートダッシュボードページ。ダッシュボードページは指定された順序で並べられます。 パラメータの動作: - 必須

戻り値

(object) dashboardids プロパティの下に作成されたテンプレートダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートダッシュボードの順序と一致します。

例

テンプレートダッシュボードの作成

1 つのダッシュボードページに 1 つのグラフウィジェットを持つ「Graphs」という名前のテンプレートダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.create",
    "params": {
        "templateid": "10318",
        "name": "Graphs",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 6,
                                "name": "graphid",
                                "value": "1123"
                            }
                        ]
                    }
                ]

            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

参照

- Template dashboard page
- Template dashboard widget
- Template dashboard widget field

ソース

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

説明

`object templatedashboard.delete(array templateDashboardIds)`

このメソッドは、テンプレートダッシュボードを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するテンプレートダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

複数のテンプレートダッシュボードの削除

2 つのテンプレートダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

ソース

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

説明

`integer/array templatedashboard.get(object parameters)`

このメソッドは、指定されたパラメータに従ってテンプレートダッシュボードを取得することができます。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
dashboardids	ID/array	指定した ID のテンプレートダッシュボードのみを返します。
templateids	ID/array	指定したテンプレートに属するテンプレートダッシュボードのみを返します。
selectPages	query	テンプレートダッシュボードページを正しい順序で持つpagesプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: dashboardid, name。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

テンプレートダッシュボードの取得

指定したテンプレートのウィジェット付きテンプレートダッシュボードをすべて取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "23",
            "name": "Docker overview",
```

```
            "templateid": "10001",
            "display_period": "30",
            "auto_start": "1",
            "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "220",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1125"
                                }
                            ]
                        },
                        {
                            "widgetid": "221",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1129"
                                }
                            ]
                        },
                        {
                            "widgetid": "222",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1128"
                                }
                            ]
                        },
                        {
```

```
                            "widgetid": "223",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1126"
                                }
                            ]
                        },
                        {
                            "widgetid": "224",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "10",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1127"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Template dashboard page
- Template dashboard widget
- Template dashboard widget field

ソース

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

説明

`object templatedashboard.update(object/array templateDashboards)`

このメソッドは、既存のテンプレートダッシュボードを更新するためのものです。

> **Note:**
> このメソッドは管理者およびスーパ管理者ユーザータイプのみが利用できます。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するテンプレートダッシュボードのプロパティ。

各ダッシュボードには dashboardid プロパティを指定する必要があり、他のすべてのプロパティはオプションです。指定されたプロパティのみが更新されます。

標準のテンプレートダッシュボードプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
pages	array	既存のダッシュボードページを置き換えるテンプレートダッシュボードページ。 ダッシュボードページは dashboard_pageid プロパティによって更新されます。dashboard_pageid プロパティがないオブジェクトには新しいダッシュボードページが作成され、既存のダッシュボードページは再利用されない場合は削除されます。ダッシュボードページは指定された順序で並べられます。ダッシュボードページの指定されたプロパティのみが更新されます。pages プロパティには少なくとも 1 つのダッシュボードページオブジェクトが必要です。

戻り値

(object) dashboardids プロパティの下で更新されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

テンプレートダッシュボードの名前変更

テンプレートダッシュボードの名前を「Performance graphs」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

テンプレートダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換え、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
```

```
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

参照

- Template dashboard widget
- Template dashboard widget field

ソース

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

Template group

このクラスは、テンプレートグループを操作するためのものです。

オブジェクトリファレンス:

- Template group

利用可能なメソッド:

- templategroup.create - 新しいテンプレートグループを作成する
- templategroup.delete - テンプレートグループを削除する
- templategroup.get - テンプレートグループを取得する
- templategroup.massadd - テンプレートグループへ関連オブジェクトを追加する
- templategroup.massremove - テンプレートグループから関連オブジェクトを削除する
- templategroup.massupdate - テンプレートグループの関連オブジェクトを置換または削除する
- templategroup.propagate -テンプレートグループのサブグループへの権限を伝達する
- templategroup.update - テンプレートグループを更新する

Template group オブジェクト

次のオブジェクトは、templategroupAPI に直接関連しています。

テンプレートグループ

テンプレートグループオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
groupid	ID	テンプレートグループの ID プロパティの動作: - 読み取り専用 - 更新操作では必須
name	文字列	テンプレートグループの名前 プロパティの動作: - 作成操作では必須
uuid	文字列	インポートしたテンプレートグループを既存のテンプレートグループにリンクするために使用される、Universal Unique Identifier（UUID）指定がない場合は自動生成されます。

templategroup.create

説明

`object templategroup.create(object/array templateGroups)`

このメソッドは、新しいテンプレートグループを作成します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object/array) 作成するテンプレートグループ。このメソッドは、標準テンプレートグループプロパティを持つテンプレートグループを受け入れます。

戻り値

(object) は、groupids プロパティに指定された、作成されたテンプレートグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートグループの順序と一致します。

例

テンプレートグループの作成

”Templates/Databases” という名前のテンプレートグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
```

```
    "id": 1
}
```

ソース

CTemplateGroup::create() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

説明

`object templategroup.delete(array templateGroupIds)`

このメソッドはテンプレートグループを削除するためのものです。

テンプレートグループに、そのグループのみに属するテンプレートが含まれている場合、そのテンプレートグループは削除できません。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(array) 削除するテンプレートグループの ID

戻り値

(object) groupids プロパティの下に削除されたテンプレートグループの ID を含むオブジェクトを返します。

例

複数のテンプレートグループの削除

二つのテンプレートグループを削除する。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

説明

`integer/array templategroup.get(object parameters)`

このメソッドは、指定されたパラメーターに従ってテンプレートグループを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) は、必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
graphids	ID/array	指定されたグラフを持つテンプレートを含むテンプレートグループのみを返します。
groupids	ID/array	指定されたテンプレートグループ ID を持つテンプレートグループのみを返します。
templateids	ID/array	指定されたテンプレートを含むテンプレートグループのみを返します。
triggerids	ID/array	指定されたトリガーを持つテンプレートを含むテンプレートグループのみを返します。
with_graphs	flag	グラフを持つテンプレートを含むテンプレートグループのみを返します。
with_graph_prototypes	flag	グラフプロトタイプを持つテンプレートを含むテンプレートグループのみを返します。
with_httptests	フラグ	Web チェックを含むテンプレートを含むテンプレートグループのみを返します。
with_items	フラグ	アイテムを含むテンプレートを含むテンプレートグループのみを返します。 with_simple_graph_items パラメータをオーバーライドします。
with_item_prototypes	フラグ	アイテムプロトタイプを含むテンプレートを含むテンプレートグループのみを返します。 with_simple_graph_item_prototypes パラメータをオーバーライドします。
with_simple_graph_item_prototypes	フラグ	作成が有効で、数値型の情報を持つアイテムプロトタイプを含むテンプレートを含むテンプレートグループのみを返します。
with_simple_graph_items	フラグ	数値アイテムを含むテンプレートを含むテンプレートグループのみを返します。
with_templates	フラグ	テンプレートを含むテンプレートグループのみを返します。
with_triggers	フラグ	トリガー付きのテンプレートを含むテンプレートグループのみを返します。
selectTemplates	クエリ	テンプレートグループに属するテンプレートを含むtemplates プロパティを返します。 count をサポートします。
limitSelects	整数	サブセレクトによって返されるレコード数を制限します。 次のサブセレクトに適用されます: selectTemplates - 結果は template でソートされます。
sortfield	文字列/配列	指定されたプロパティで結果をソートします。 可能な値: groupid、name
countOutput	ブール値	これらのパラメータについては、リファレンス解説 で説明されています。
editable	ブール値	
excludeSearch	ブール値	
filter	オブジェクト	
limit	整数	
output	クエリ	
preservekeys	ブール値	
search	オブジェクト	
searchByAny	ブール値	

パラメータ	型	説明
searchWildcardsEnabled	ブール値	
sortorder	文字列/配列	
startSearch	ブール値	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用された場合は、取得したオブジェクトの数

例

名前によるデータの取得

"Templates/Databases" と "Templates/Modules" という名前の二つのテンプレートグループについて、全てのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::get() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

説明

`object templategroup.massadd(object parameters)`

このメソッドは、指定されたテンプレートグループすべてに複数の関連オブジェクトを同時に追加することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 更新するテンプレートグループの ID と、すべてのテンプレートグループに追加するオブジェクトを含むパラメータ

このメソッドは以下のパラメータを受け入れます。

パラメータ	タイプ	説明
groups	object/array	更新するテンプレートグループ テンプレートグループには、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
templates	object/array	すべてのテンプレートグループに追加するテンプレート テンプレートには templateid プロパティのみが定義されている必要があります。 パラメーターの動作: - 必須

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループへテンプレートの追加

ID12 と 13 のテンプレートグループへ二つのテンプレートを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::massAdd() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

説明

`object templategroup.massremove(object parameters)`

このメソッドは、関連するオブジェクトを複数のテンプレートグループから削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 更新するテンプレートグループの ID と削除するオブジェクトを含むパラメータ

パラメータ	タイプ	説明
groupids	ID/配列	更新するテンプレートグループの ID パラメータの動作: - 必須
templateids	ID/配列	すべてのテンプレートグループから削除するテンプレートの ID パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループからテンプレートの削除

指定されたテンプレートグループから 2 つのテンプレートを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
```

```
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::massRemove() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massupdate

説明

`object templategroup.massupdate(object parameters)`

このメソッドは、複数のテンプレートグループの指定されたテンプレートを置き換えることができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 更新するテンプレートグループと更新対象のオブジェクトの ID を含むパラメータ

パラメータ	タイプ	説明
groups	object/array	更新するテンプレートグループ テンプレートグループには `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
templates	object/array	テンプレートを使用して、指定されたテンプレートグループの現在のテンプレートを置き換えます。 指定されたテンプレート以外のすべてのテンプレートは、テンプレートグループから除外されます。 テンプレートには、`templateid` プロパティのみが定義されている必要があります。 パラメーターの動作: - 必須

戻り値

(object) `groupids` プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループ内のテンプレートを置き換える

テンプレートグループ内のすべてのテンプレートを、指定のテンプレートに置き換えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "8"
            }
        ],
        "templates": [
            {
                "templateid": "40050"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
        ]
    },
    "id": 1
}
```

参照

- templategroup.update
- templategroup.massadd
- Template

ソース

CTemplateGroup::massUpdate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

説明

`object templategroup.propagate(object parameters)`

このメソッドは、すべてのテンプレートグループのサブグループにアクセス許可を適用することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) は、必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
groups	object/array	伝播するテンプレートグループ テンプレートグループには、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
permissions	boolean	権限を伝播する必要がある場合は、true に設定してください。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に伝搬されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの権限をそのサブグループへ伝搬する

テンプレートグループの権限をそのサブグループへ伝搬します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

参照

- templategroup.update
- templategroup.massadd
- Template

ソース

CTemplateGroup::propagate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

説明

`object templategroup.update(object/array templates)`

このメソッドは、既存のテンプレートグループを更新することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object/array) 更新するテンプレートグループのプロパティ。

groupid プロパティはテンプレートグループごとに定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの名前の変更

テンプレートグループの名前を”Templates/Databases” へ変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::update() in *ui/include/classes/api/services/CTemplateGroup.php*.

Token

このクラスは、トークンを操作するためのものです。

オブジェクトリファレンス:

- Token

利用可能なメソッド:

- token.create - 新しいトークンを作成する
- token.delete - トークンを削除する
- token.get - トークンを取得する
- token.update - トークンを更新する
- token.generate - トークンを生成する

Token オブジェクト

次のオブジェクトは、tokenAPI に直接関連しています。

トークン

トークンオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tokenid	ID	トークンの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
name	string	トークンの名前。 プロパティの動作: - 作成操作時に必須
description	text	トークンの説明。
userid	ID	トークンが割り当てられているユーザーの ID。 デフォルト: 現在のユーザー。 プロパティの動作: - 定数
lastaccess	timestamp	トークンが最後に認証された日時。 トークンが一度も認証されていない場合は”0”。 プロパティの動作: - 読み取り専用
status	integer	トークンのステータス。 設定可能な値: 0 - *(*デフォルト*)* 有効なトークン; 1 - 無効なトークン。
expires_at	timestamp	トークンの有効期限。 無期限トークンの場合は”0”。
created_at	timestamp	トークンの作成日時。 プロパティの動作: - 読み取り専用
creator_userid	ID	トークンを作成したユーザーの ID。 プロパティの動作: - 読み取り専用

token.create

説明

`object token.create(object/array tokens)`

このメソッドは、新しいトークンを作成することができます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

> **Attention:**
> このメソッドで作成されたトークンも、使用可能になる前に生成する必要があります。

パラメータ

`(object/array)` トークンの作成

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) tokenids プロパティの下で作成されたトークンの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトークンの順序と一致します。

例

トークンの作成

有効期限が切れることのない有効なトークンを作成し、ID2 のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

2021 年 1 月 21 日に有効期限が切れるトークンを作成します。このトークンは現在のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

ソース

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

説明

`object token.delete(array tokenids)`

このメソッドは、トークンを削除することができます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

パラメーター

(array) 削除するトークンの ID。

戻り値

(object) tokenids プロパティの下にある削除されたトークンの ID を含むオブジェクトを返します。

例

複数のトークンの削除

2 つのトークンを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

説明

`object token.generate(array tokenids)`

このメソッドはトークンを生成するためのものです。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーロールに *API* トークンの管理権限が必要です。

> **Attention:**
> このメソッドでトークンを生成できるのは、作成済みの場合のみです。

パラメータ

(array) 生成するトークンの ID

戻り値

(array) tokenid プロパティに生成されたトークンの ID を、token プロパティに生成された認証文字列を含むオブジェクトの配列を返します。

プロパティ	タイプ	説明
tokenid	ID	トークンの ID。
token	string	このトークンのために生成された認証文字列。

例

複数のトークンを生成する

2 つのトークンを生成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

ソース

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

説明

`integer/array token.get(object parameters)`

このメソッドでは、指定されたパラメーターに従ってトークンを取得できます。

> **Note:**
> *Super admin* ユーザータイプのみが他のユーザーのトークンを表示できます。

パラメーター

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメーター	タイプ	説明
tokenids	ID/array	指定された ID のトークンのみを返します。
userids	ID/array	指定されたユーザー用に作成されたトークンのみを返します。

パラメーター	タイプ	説明
token	string	指定された *Auth* トークン用に作成されたトークンのみを返します。
valid_at	timestamp	指定された日時に有効な (期限の切れていない) トークンのみを返します。
expired_at	timestamp	指定された日時に無効な (期限切れになった) トークンのみを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます 利用可能な値: `tokenid`、`name`、`lastaccess`、`status`、`expires_at`、`created_at`。
countOutput	boolean	これらのパラメーターについては、リファレンスの解説で解説されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トークンの取得

ID"2" のトークンのデータをすべて取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
    ],
    "id": 1
}
```

ソース

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

説明

`object token.update(object/array tokens)`

このメソッドは、既存のトークンを更新することができます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

パラメータ

(`object/array`) 更新されるトークンプロパティ

`tokenid` プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(`object`) `tokenids` プロパティの下で更新されたトークンの ID を含むオブジェクトを返します。

例

トークンの有効期限変更

トークンから有効期限を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CToken::update() in *ui/include/classes/api/services/CToken.php*.

Trend

このクラスは、トレンドデータを操作するためのものです。

オブジェクトリファレンス:

- Float trend
- Integer trend

利用可能なメソッド:

- trend.get -トレンドを取得する

Trend オブジェクト

次のオブジェクトは、trendAPI に直接関連しています。

> **Note:**
> トレンドオブジェクトは、アイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API によって変更することはできません。

浮動小数点トレンド

浮動小数点トレンドオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値が計算された時間のタイムスタンプ。例えば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間に計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その時間に利用可能だった値の数。
value_min	float	1 時間あたりの最小値。
value_avg	float	1 時間あたりの平均値。
value_max	float	1 時間あたりの最大値。

整数トレンド

整数トレンドオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値が計算された時間のタイムスタンプ。例えば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間に計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その時間に利用可能だった値の数。
value_min	integer	1 時間あたりの最小値。
value_avg	integer	1 時間あたりの平均値。
value_max	integer	1 時間あたりの最大値。

trend.get

説明

`integer/array trend.get(object parameters)`

このメソッドは、指定されたパラメータに従ってトレンドデータを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。
 このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。
 詳細はユーザーロールを参照してください。

パラメーター

(object) 取得したい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
itemids	ID/array	指定したアイテム ID のトレンドのみを返します。
time_from	timestamp	指定した時刻以降に収集された値のみを返します。
time_till	timestamp	指定した時刻以前に収集された値のみを返します。
countOutput	boolean	取得したオブジェクトの数をカウントします。

パラメーター	タイプ	説明
limit	integer	取得するオブジェクトの数を制限します。
output	query	返すTrend オブジェクトのプロパティを設定します。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイテムのトレンドデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

ソース

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

Trigger

このクラスは、トリガーを操作するためのものです。

オブジェクトリファレンス:

- Trigger
- Trigger tag

利用可能なメソッド:

- trigger.create - 新しいトリガーを作成する
- trigger.delete -トリガーを削除する
- trigger.get -トリガーを取得する
- trigger.update -トリガーを更新する

Trigger オブジェクト

次のオブジェクトは、`triggerAPI` に直接関連しています。

トリガー

トリガーオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
triggerid	ID	トリガーの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
description	string	トリガーの名前。 プロパティの動作: - 作成操作では必須
expression	string	簡略化されたトリガー式。 プロパティの動作: - 作成操作では必須
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガーの追加説明。
error	string	トリガーの状態更新時に問題が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用
flags	integer	トリガーの由来。 可能な値: 0 - *(デフォルト)* 通常のトリガー; 4 - プロトタイプから変換されたトリガー。 プロパティの動作: - 読み取り専用
lastchange	timestamp	トリガーの状態が最後に変更された時刻。 プロパティの動作: - 読み取り専用
priority	integer	トリガーの深刻度。 可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度; 4 - 高度; 5 - 障害。

プロパティ	型	説明
state	integer	トリガーの状態。 可能な値: 0 - *(デフォルト)* トリガーの状態は最新; 1 - 現在のトリガー状態は不明。 プロパティの動作: - 読み取り専用
status	integer	トリガーが有効か無効か。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
templateid	ID	親テンプレートトリガーの ID。 プロパティの動作: - 読み取り専用
type	integer	トリガーが複数の問題イベントを生成できるかどうか。 可能な値: 0 - *(デフォルト)* 複数のイベントを生成しない; 1 - 複数のイベントを生成する。
url	string	トリガーに関連付けられた URL。
url_name	string	トリガーに関連付けられた URL のラベル。
value	integer	トリガーが OK か問題状態か。 可能な値: 0 - *(デフォルト)* OK; 1 - 問題。 プロパティの動作: - 読み取り専用
recovery_mode	integer	OK イベント生成モード。 可能な値: 0 - *(デフォルト)* 式; 1 - 復旧式; 2 - なし。
recovery_expression	string	簡略化されたトリガー復旧式。
correlation_mode	integer	OK イベントのクローズ。 可能な値: 0 - *(デフォルト)* すべての問題; 1 - タグ値が一致するすべての問題。
correlation_tag	string	一致用タグ。
manual_close	integer	手動クローズを許可。 可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。
uuid	string	インポートされたトリガーを既存のものにリンクするためのユニバーサル一意識別子。指定しない場合は自動生成。 プロパティの動作: - トリガーがテンプレートに属する場合のみサポート

トリガータグ

トリガータグオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag	string	トリガータグ名。 プロパティの動作: - 必須
value	string	トリガータグの値。

trigger.create

説明

`object trigger.create(object/array triggers)`

このメソッドは新しいトリガーを作成します。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するトリガー。

標準のトリガープロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
dependencies	array	トリガーが依存しているトリガー。 トリガーには `triggerid` プロパティのみが定義されている必要があります。
tags	array	トリガータグ。

Attention:
トリガー式は展開された形で指定する必要があります。

戻り値

(object) `triggerids` プロパティの下で作成されたトリガーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガーの順序と一致します。

例

トリガーの作成

単一のトリガー依存関係を持つトリガーを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Processor load is too high on {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Service status",
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
```

```
                {
                    "triggerid": "17368"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

説明

`object trigger.delete(array triggerIds)`

このメソッドは、トリガーを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するトリガーの ID。

戻り値

(object) triggerids プロパティの下にある削除されたトリガーの ID を含むオブジェクトを返します。

例

複数のトリガーの削除

2 つのトリガーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
```

```
        "12002",
        "12003"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

説明

`integer/array trigger.get(object parameters)`

このメソッドは、指定したパラメーターに応じたトリガーを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	タイプ	説明
triggerids	ID/array	指定した ID のトリガーのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのトリガーのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガーのみを返します。
hostids	ID/array	指定したホストに属するトリガーのみを返します。
itemids	ID/array	指定したアイテムを含むトリガーのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧については、サポートされている関数ページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに属するホストまたはテンプレートのトリガーのみを返します。
host	string	指定した技術名のホストに属するトリガーのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたトリガーのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するトリガーのみを返します。
dependent	boolean	`true` に設定すると、依存関係を持つトリガーのみを返します。`false` に設定すると、依存関係を持たないトリガーのみを返します。
monitored	flag	監視対象ホストに属し、有効なアイテムのみを含む有効なトリガーのみを返します。
active	flag	監視対象ホストに属する有効なトリガーのみを返します。
maintenance	boolean	`true` に設定すると、メンテナンス中のホストに属する有効なトリガーのみを返します。

パラメーター	タイプ	説明
withUnacknowledgedEvents	flag	未確認イベントを持つトリガーのみを返します。
withAcknowledgedEvents	flag	すべてのイベントが確認済みのトリガーのみを返します。
withLastEventUnacknowledged	flag	最後のイベントが未確認のトリガーのみを返します。
skipDependent	flag	他のトリガーに依存している問題状態のトリガーをスキップします。なお、他のトリガーが無効、無効なアイテム、または無効なアイテムホストの場合は無視されます。
lastChangeSince	timestamp	指定した時刻以降に状態が変化したトリガーのみを返します。
lastChangeTill	timestamp	指定した時刻以前に状態が変化したトリガーのみを返します。
only_true	flag	最近問題状態になったトリガーのみを返します。
min_severity	integer	指定した深刻度以上の深刻度を持つトリガーのみを返します。
evaltype	integer	タグの評価方法。 可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグを持つトリガーのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのトリガーを返します。 可能なoperatorの値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
expandComment	flag	トリガーの説明でマクロを展開します。
expandDescription	flag	トリガー名でマクロを展開します。
expandExpression	flag	トリガー式で関数とマクロを展開します。
selectHostGroups	query	`hostgroups`プロパティでトリガーが属するホストグループを返します。
selectHosts	query	`hosts`プロパティでトリガーが属するホストを返します。
selectItems	query	`items`プロパティでトリガーに含まれるアイテムを返します。
selectFunctions	query	`functions` プロパティでトリガーで使用されている関数を返します。 関数オブジェクトはトリガー式で使用されている関数を表し、以下のプロパティを持ちます: `functionid` - (ID) 関数の ID; `itemid` - (ID) 関数で使用されているアイテムの ID; `function` - (string) 関数名; `parameter` - (string) 関数に渡されるパラメーター。クエリパラメーターは返される文字列内で `$` 記号に置き換えられます。
selectDependencies	query	`dependencies` プロパティでトリガーが依存しているトリガーを返します。

パラメーター	タイプ	説明
selectDiscoveryData	query	トリガーのディスカバリーオブジェクトデータを持つ `discoveryData` プロパティを返します。トリガーディスカバリーオブジェクトは、検出されたトリガーを、それが検出されたトリガープロトタイプにリンクします。 以下のプロパティを持ちます: `parent_triggerid` - (ID) トリガーが作成されたトリガープロトタイプの ID; `status` - (int) トリガーディスカバリーステータス: 0 - *(デフォルト)* トリガーが検出されている; 1 - トリガーがもはや検出されていない; `ts_delete` - (timestamp) もはや検出されていないトリガーが削除される時刻; `ts_disable` - (timestamp) もはや検出されていないトリガーが無効化される時刻; `disable_source` - (int) トリガーが LLD ルールまたは手動で無効化されたかどうかの指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	`discoveryRule` プロパティでトリガーを作成した低レベルディスカバリールールを返します。
selectLastEvent	query	`lastEvent`プロパティで最後の重要なトリガーイベントを返します。
selectTags	query	`tags`プロパティでトリガータグを返します。
selectTemplateGroups	query	`templategroups`プロパティでトリガーが属するテンプレートグループを返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 `text`データ型のプロパティはサポートされません。 追加プロパティをサポートします: `host` - トリガーが属するホストの技術名; `hostid` - トリガーが属するホストの ID。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `host` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	`triggerDiscovery` プロパティでトリガーディスカバリーオブジェクトを返します。トリガーディスカバリーオブジェクトは、トリガーをそれが作成されたトリガープロトタイプにリンクします。 このクエリは非推奨です。代わりに `selectDiscoveryData` を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

トリガー ID によるデータの取得

トリガー"14062" で使用されているすべてのデータと関数を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "14062",
            "expression": "{13513}<10m",
            "description": "{HOST.NAME} has been restarted (uptime < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "The host uptime is less than 10 minutes",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "event_name": "",
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

障害状態のトリガーの取得

障害状態にあるすべてのトリガーの ID、名前、深刻度を取得し、深刻度の降順で並べ替えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

タグ付きの特定のトリガーの取得

タグ付きの特定のトリガーを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Discovery rule
- Item
- Host
- Host group
- Template group

ソース

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

説明

`object trigger.update(object/array triggers)`

このメソッドは、既存のトリガーを更新するためのものです。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 更新するトリガーのプロパティ。

各トリガーには `triggerid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のトリガープロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
dependencies	array	トリガーが依存しているトリガー。
tags	array	トリガーには `triggerid` プロパティのみを定義する必要があります。 トリガータグ。

Attention:
トリガーの式は展開された形式で指定する必要があります。

戻り値

(object) triggerids プロパティの下で更新されたトリガーの ID を含むオブジェクトを返します。

例

トリガーの有効化

トリガーを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

トリガータグの置換

トリガーのタグを置換します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "tags": [
            {
                "tag": "service",
                "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
            },
            {
                "tag": "error",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

依存関係の置き換え

トリガーの依存関係を置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

Trigger prototype

このクラスは、トリガープロトタイプを操作するためのものです。

オブジェクトリファレンス:

- Trigger prototype
- Trigger prototype tag

利用可能なメソッド:

- triggerprototype.create - 新しいトリガープロトタイプを作成する
- triggerprototype.delete -トリガープロトタイプを削除する
- triggerprototype.get -トリガープロトタイプを取得する
- triggerprototype.update -トリガープロトタイプを更新する

Trigger prototype オブジェクト

次のオブジェクトは、triggerprototypeAPI に直接関連しています。

トリガープロトタイプ

トリガープロトタイプオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
triggerid	ID	トリガープロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
description	string	トリガープロトタイプの名前。 プロパティの動作: - 作成操作では必須
expression	string	簡略化されたトリガー式。 プロパティの動作: - 作成操作では必須
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガープロトタイプへの追加コメント。
priority	integer	トリガープロトタイプの深刻度。 可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度; 4 - 高度; 5 - 障害。
status	integer	トリガープロトタイプが有効か無効か。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
flags	integer	トリガープロトタイプの起源。 可能な値: 2 - トリガープロトタイプ; 6 - 検出されたトリガープロトタイプ。 プロパティの動作: - 読み取り専用
templateid	ID	親テンプレートトリガープロトタイプの ID。 プロパティの動作: - 読み取り専用
type	integer	トリガープロトタイプが複数の問題イベントを生成できるかどうか。 可能な値: 0 - *(デフォルト)* 複数のイベントを生成しない; 1 - 複数のイベントを生成する。
url	string	トリガープロトタイプに関連付けられた URL。
url_name	string	トリガープロトタイプに関連付けられた URL のラベル。
recovery_mode	integer	OK イベント生成モード。 可能な値: 0 - *(デフォルト)* 式; 1 - 復旧式; 2 - なし。
recovery_expression	string	簡略化されたトリガー復旧式。
correlation_mode	integer	OK イベントのクローズ条件。 可能な値: 0 - *(デフォルト)* すべての問題; 1 - タグ値が一致するすべての問題。

プロパティ	タイプ	説明
correlation_tag	string	一致させるためのタグ。
manual_close	integer	手動クローズを許可するかどうか。 可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。
discover	integer	トリガープロトタイプの検出ステータス。 可能な値: 0 - *(デフォルト)* 新しいトリガーが検出される; 1 - 新しいトリガーは検出されず、既存のトリガーはロストとしてマークされる。
uuid	string	インポートされたトリガープロトタイプを既存のものにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - トリガープロトタイプがテンプレートに属している場合サポート

トリガープロトタイプタグ

トリガープロトタイプタグオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	トリガープロトタイプタグ名。 プロパティの動作: - 必須
value	string	トリガープロトタイプタグの値。

triggerprototype.create

説明

`object triggerprototype.create(object/array triggerPrototypes)`

このメソッドは新しいトリガープロトタイプを作成します。

Note:
このメソッドは管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object/array)` 作成するトリガープロトタイプ。

標準のトリガープロトタイププロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。 トリガーには `triggerid` プロパティのみを定義する必要があります。
tags	array	トリガープロトタイプのタグ。

Attention:
トリガー式は展開された形式で指定する必要があり、少なくとも 1 つのアイテムプロトタイプを含める必要があります。

戻り値

`(object)` `triggerids` プロパティの下に作成されたトリガープロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガープロトタイプの順序と一致します。

例

トリガープロトタイプの作成

ファイルシステムの空きディスク容量が 20% 未満になった場合に検出するトリガープロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Free disk space is less than 20% on volume {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

説明

`object triggerprototype.delete(array triggerPrototypeIds)`

このメソッドは、トリガーのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するトリガープロトタイプの ID。

戻り値

(object) triggerids プロパティの下にある削除されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

複数のトリガープロトタイプの削除

2 つのトリガープロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

説明

`integer/array triggerprototype.get(object parameters)`

このメソッドは、指定したパラメーターに応じたトリガーのプロトタイプを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`object`) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
active	flag	監視対象ホストに属する有効なトリガープロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するトリガープロトタイプのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧はサポートされている関数ページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに属するホストまたはテンプレートに属するトリガープロトタイプのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートに属するトリガープロトタイプのみを返します。
host	string	指定した名前のホストに属するトリガープロトタイプのみを返します。
hostids	ID/array	指定したホストに属するトリガープロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたトリガープロトタイプのみを返します。
maintenance	boolean	`true` に設定すると、メンテナンス中のホストに属する有効なトリガープロトタイプのみを返します。

パラメーター	型	説明
min_severity	integer	指定した深刻度以上の深刻度を持つトリガープロトタイプのみを返します。
monitored	flag	監視対象ホストに属し、有効なアイテムのみを含む有効なトリガープロトタイプのみを返します。
templated	boolean	true に設定すると、テンプレートに属するトリガープロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガープロトタイプのみを返します。
triggerids	ID/array	指定した ID のトリガープロトタイプのみを返します。
expandExpression	flag	トリガー式内の関数とマクロを展開します。
selectDependencies	query	dependencies プロパティでトリガープロトタイプが依存しているトリガープロトタイプおよびトリガーを返します。
selectDiscoveryData	query	トリガープロトタイプのディスカバリーオブジェクトデータを持つ discoveryData プロパティを返します。トリガープロトタイプのディスカバリーオブジェクトは、検出されたトリガープロトタイプを、そこから検出されたトリガープロトタイプにリンクします。 以下のプロパティがあります: parent_triggerid - (ID) トリガープロトタイプが作成された元のトリガープロトタイプの ID; status - (int) トリガープロトタイプのディスカバリーステータス: 0 - *(デフォルト)* トリガープロトタイプが検出されている; 1 - トリガープロトタイプがもはや検出されていない; ts_delete - (timestamp) もはや検出されていないトリガープロトタイプが削除される時刻; ts_disable - (timestamp) もはや検出されていないトリガープロトタイプが無効化される時刻; disable_source - (int) トリガープロトタイプが LLD ルールによって無効化されたか手動で無効化されたかの指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	トリガープロトタイプが属するLLD ルールを discoveryRule プロパティで返します。
selectDiscoveryRulePrototype	query	トリガープロトタイプが属する親 LLD ルールプロトタイプを持つdiscoveryRulePrototypeプロパティを返します。
selectFunctions	query	functions プロパティでトリガープロトタイプで使用されている関数を返します。 関数オブジェクトはトリガー式で使用されている関数を表し、以下のプロパティがあります: functionid - (ID) 関数の ID; itemid - (ID) 関数で使用されているアイテムの ID; function - (string) 関数名; parameter - (string) 関数に渡されるパラメーター。クエリパラメーターは返される文字列内で $ 記号に置き換えられます。
selectHostGroups	query	トリガープロトタイプが属するホストグループをhostgroupsプロパティで返します。
selectHosts	query	トリガープロトタイプが属するホストをhostsプロパティで返します。
selectItems	query	トリガープロトタイプで使用されているアイテムおよびアイテムプロトタイプをitemsプロパティで返します。
selectTags	query	トリガープロトタイプのタグをtagsプロパティで返します。
selectTemplateGroups	query	トリガープロトタイプが属するテンプレートグループをtemplategroupsプロパティで返します。
filter	object	指定したフィルターと完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる単一値または値の配列です。 textデータ型のプロパティはサポートされません。 追加プロパティをサポートします: host - トリガープロトタイプが属するホストの技術名; hostid - トリガープロトタイプが属するホストの ID。

パラメーター	型	説明
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `host` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

LLD ルールからトリガープロトタイプを取得する

LLD ルールからすべてのトリガープロトタイプとその関数を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
```

```
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


タグで特定のトリガープロトタイプを取得

リクエスト：


```
{
    "jsonrpc": "2.0",
```

```
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

参照

- Discovery rule
- Item
- Host
- Host group
- Template group

ソース

ui/include/classes/api/services/CTriggerPrototype.php の CTriggerPrototype::get()。

triggerprototype.update

説明

`object triggerprototype.update(object/array triggerPrototypes)`

このメソッドは、既存のトリガープロトタイプを更新するためのものです。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するトリガープロトタイプのプロパティ。

各トリガープロトタイプには triggerid プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のトリガープロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。
tags	array	トリガーには triggerid プロパティのみを定義する必要があります。 トリガープロトタイプのタグ。

Attention:
トリガー式は展開された形式で指定する必要があり、少なくとも 1 つのアイテムプロトタイプを含める必要があります。

戻り値

(object) triggerids プロパティの下で更新されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

トリガープロトタイプの有効化

トリガープロトタイプを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

トリガープロトタイプタグの置換

1 つのトリガープロトタイプのタグを置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

User

このクラスは、ユーザーを操作するためのものです。

オブジェクトリファレンス:

- User
- Media

利用可能なメソッド:

- user.checkauthentication - ユーザーセッションの確認と延長をする
- user.create - 新しいユーザーを作成する
- user.delete - ユーザーを削除する
- user.get - ユーザーを取得する
- user.login - API にログインする
- user.logout - API からログアウトする
- user.provision - LDAP ユーザーのプロビジョニングをする
- user.resettotp - ユーザーの TOTP シークレットをリセットする
- user.unblock - ユーザーブロックを解除する
- user.update - ユーザーを更新する

User オブジェクト

次のオブジェクトは、userAPI に直接関連しています。

ユーザー

ユーザーオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
userid	ID	ユーザーの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須

プロパティ	タイプ	説明
username	string	ユーザー名。 プロパティの動作: - 作成操作時に必須 - ユーザーディレクトリにリンクされているプロビジョニングユーザーの場合、ユーザーディレクトリプロビジョニングステータスが有効（ユーザーディレクトリオブジェクトの `provision_status` が”1”）、かつすべての LDAP または SAML プロビジョニングの認証ステータスが有効（認証オブジェクトの `ldap_jit_status` が” 設定済み LDAP IdP で有効” または `saml_jit_status` が” 設定済み SAML IdP で有効”）で、`userdirectoryid` が”0” 以外の有効な値に設定されている場合は読み取り専用
passwd	string	ユーザーのパスワード。 このパラメータの値は、ユーザーディレクトリにリンクされている場合、空文字列にすることができます。 プロパティの動作: - 書き込み専用
roleid	ID	ユーザーのロール ID。 ロールのないユーザーは、Zabbix で設定されたユーザーグループマッピングと一致する LDAP/SAML 情報がある場合にのみ、LDAPまたはSAML認証を使用して Zabbix にログインできます。
attempt_clock	timestamp	最後のログイン失敗時刻。 プロパティの動作: - 読み取り専用
attempt_failed	integer	最近のログイン失敗回数。 プロパティの動作: - 読み取り専用
attempt_ip	string	最後のログイン失敗が発生した IP アドレス。 プロパティの動作: - 読み取り専用
autologin	integer	自動ログインを有効にするかどうか。 設定可能な値: 0 - *(デフォルト)* 自動ログイン無効; 1 - 自動ログイン有効。
autologout	string	ユーザーセッションの有効期間。秒または単位付きの時間を指定可能。0s に設定するとセッションは期限切れになりません。 デフォルト: 15m。
lang	string	ユーザーの言語コード。例: `en_US`。 デフォルト: `default` - システムデフォルト。
name	string	ユーザーの名前。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 設定可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニング済み。 プロパティの動作: - 読み取り専用
refresh	string	自動リフレッシュ間隔。秒または単位付きの時間を指定可能（例: 30s, 90s, 1m, 1h）。 デフォルト: 30s。

プロパティ	タイプ	説明
rows_per_page	integer	1 ページあたりに表示するオブジェクト行数。 デフォルト: 50。
surname	string	ユーザーの姓。
theme	string	ユーザーのテーマ。 設定可能な値: `default` - *(デフォルト)* システムデフォルト; `blue-theme` - ブルー; `dark-theme` - ダーク。
ts_provisioned	timestamp	最新のプロビジョニング操作が行われた時刻。 プロパティの動作: - 読み取り専用 - スーパ管理者タイプのユーザーでサポート
url	string	ログイン後にユーザーをリダイレクトするページの URL。
userdirectoryid	ID	ユーザーがリンクされているユーザーディレクトリの ID。 プロビジョニング（作成または更新）や、ユーザーディレクトリにリンクされたユーザーのログインに使用されます。 ログイン操作では、このプロパティの値が、ユーザーが所属するユーザーグループの `userdirectoryid` プロパティよりも優先されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用 - スーパ管理者タイプのユーザーでサポート
timezone	string	ユーザーのタイムゾーン。例: `Europe/London`, `UTC`。 デフォルト: `default` - システムデフォルト。 サポートされているタイムゾーンの一覧はPHP ドキュメントを参照してください。

メディア

メディアオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
mediaid	ID	ユーザーのメディアの ID。 プロパティの動作: - 読み取り専用
mediatypeid	ID	ユーザーのメディアで使用されるメディアタイプの ID。 プロパティの動作: - 必須
sendto	string/array	受信者のアドレス、ユーザー名、またはその他の識別子。 メディアタイプの `type` が「Email」に設定されている場合、値は配列で表されます。他のメディアタイプの場合、値は文字列で表されます。 プロパティの動作: - 必須
active	integer	メディアが有効かどうか。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。

プロパティ	型	説明
severity	integer	通知を送信するトリガーの深刻度。 可能なビットマップ値: 1 - 未分類; 2 - 情報; 4 - 警告; 8 - 軽度; 16 - 高度; 32 - 障害。 これはビットマスクフィールドであり、可能なビットマップ値の任意の合計が許容されます（例: 48 は軽度、高度、障害）。 デフォルト: 63。
period	string	通知を送信できる時間を時間帯またはユーザーマクロでセミコロン区切りで指定。 デフォルト: 1-7,00:00-24:00。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニングされている。
userdirectory_mediaid	ID	プロビジョニングされたメディアのユーザーディレクトリメディアマッピング ID。 プロパティの動作: - 読み取り専用 - スーパ管理者タイプのユーザーでサポート

user.checkAuthentication

説明

`object user.checkAuthentication`

このメソッドはユーザーセッションをチェックし、延長します。

Attention:
`sessionid` パラメータを使用して `user.checkAuthentication` メソッドを呼び出すと、デフォルトでユーザーセッションが延長されます。

パラメータ

このメソッドは以下のパラメータを受け付けます。

パラメータ	タイプ	説明
extend	boolean	ユーザーセッションを延長するかどうか。 デフォルト値: "true" 値を"false" に設定すると、セッションを延長せずにユーザーセッションをチェックできます。
sessionid	string	パラメータの動作: - `sessionid` が設定されている場合は サポート ユーザーの認証トークン。
secret	string	パラメータの動作: - `token` が設定されていない場合は 必須 ランダムな 32 文字の文字列。ユーザーログイン時に生成されます。

パラメータ	タイプ	説明
token	string	ユーザーのAPI トークン。 パラメータの動作: - sessionid が設定されていない場合は 必須

戻り値

(object) ユーザーに関する情報を含むオブジェクトを返します。

標準のユーザーのプロパティに加えて、以下の情報が返されます。

プロパティ	型	説明
auth_type	integer	ユーザーのデフォルト認証。 可能な値の一覧は、認証オブジェクトの authentication_type プロパティを参照してください。
debug_mode	integer	ユーザーのデバッグモードが有効か無効か。 可能な値の一覧は、ユーザーグループオブジェクトの debug_mode プロパティを参照してください。
deprovisioned	boolean	ユーザーが無効化されたユーザーグループに属しているかどうか。
gui_access	string	フロントエンドへのユーザーの認証方法。 可能な値の一覧は、ユーザーグループオブジェクトの gui_access プロパティを参照してください。
secret	string	ランダムな 32 文字の文字列。ユーザーログイン時に生成されます。 API トークンを使用してユーザーセッションを確認した場合、secret プロパティは返されません。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。 API トークンを使用してユーザーセッションを確認した場合、sessionid プロパティは返されません。
type	integer	ユーザータイプ。 可能な値の一覧は、ロールオブジェクトの type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

例

認証トークンを使用し認証を確認

ユーザー認証トークンを使用してユーザーセッションを確認し、延長します。また、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
```

```
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
        "secret": "0e329b933e46984e49a5c1051ecd0751"
    },
    "id": 1
}
```

API トークンを使用して認証を確認します

ユーザー API トークンを使用してユーザーセッションを確認し、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
```

```
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "1",
        "deprovisioned": false,
        "auth_type": 0
    },
    "id": 1
}
```

ソース

CUser::checkAuthentication() in *ui/include/classes/api/services/CUser.php*。

user.create

説明

`object user.create(object/array users)`

このメソッドは新しいユーザーを作成します。

Note:
このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

Note:
ユーザーパスワードの強度は、認証 API で定義されたパスワードポリシールールに従って検証されます。詳細は認証 APIを参照してください。

パラメータ

(object/array) 作成するユーザー。

標準のユーザーのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
usrgrps	array	ユーザーを追加するユーザーグループ。 ユーザーグループには `usrgrpid` プロパティのみを定義する必要があります。
medias	array	作成するユーザーメディア。

戻り値

(object) `userids` プロパティの下で作成されたユーザーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーの順序と一致します。

例

ユーザーの作成

新しいユーザーを作成し、ユーザーグループに追加し、新しいメディアを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe123",
        "roleid": "5",
        "usrgrps": [
```

```
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

参照

- Authentication
- Media
- User group
- Role

ソース

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

説明

`object user.delete(array users)`

このメソッドは、ユーザーを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するユーザーの ID。

戻り値

(object) userids プロパティの下にある削除されたユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーの削除

2 つのユーザーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

説明

`integer/array user.get(object parameters)`

このメソッドは、指定されたパラメータに従ってユーザーを取得することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

Note:
ユーザーメディア、権限、またはロール情報をリクエストする場合、*Admin* および *User* タイプのユーザーは自分自身のユーザーに関するデータのみ取得できます。例については、*Admin* としてユーザーを取得するを参照してください。

パラメーター	型	説明
mediaids	ID/array	指定したメディアを使用しているユーザーのみを返します。
mediatypeids	ID/array	指定したメディアタイプを使用しているユーザーのみを返します。
userids	ID/array	指定した ID のユーザーのみを返します。
usrgrpids	ID/array	指定したユーザーグループに属するユーザーのみを返します。
getAccess	flag	ユーザーの権限に関する追加情報を追加します。 各ユーザーに以下のプロパティを追加します: `gui_access` - `(integer)` ユーザーのフロントエンド認証方式。可能な値の一覧はユーザーグループオブジェクトの `gui_access` プロパティを参照してください。 `debug_mode` - `(integer)` ユーザーのデバッグが有効かどうか。可能な値: 0 - デバッグ無効、1 - デバッグ有効。 `users_status` - `(integer)` ユーザーが無効かどうか。可能な値: 0 - ユーザー有効、1 - ユーザー無効。

パラメーター	型	説明
selectMedias	query	`medias`プロパティでユーザーが使用しているメディアを返します。
selectMediatypes	query	`mediatypes`プロパティでユーザーが使用しているメディアタイプを返します。 ユーザータイプによる制限については、`mediatype.get`を参照してください。
selectUsrgrps	query	`usrgrps`プロパティでユーザーが属するユーザーグループを返します。 ユーザータイプによる制限については、`usergroup.get`を参照してください。
selectRole	query	`role`プロパティでユーザーロールを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名、値は一致させる値または値の配列です。 `text`データ型のプロパティはサポートしません。 *Admin* および *User* タイプのユーザーが自分のユーザーグループ内のユーザーのデータをリクエストする場合のユーザーオブジェクトの可能なプロパティ: `userid`, `name`, `surname`, `username`。
output	query	返されるユーザーオブジェクトのプロパティ。 *Admin* および *User* タイプのユーザーは、以下のプロパティのみ取得できます: - 自分自身のユーザーの場合: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`。 - 自分のユーザーグループ内のユーザーの場合: `userid`, `name`, `surname`, `username`。 デフォルト: `extend`。
search	object	指定したパターンに一致する結果を返します（大文字小文字を区別しません）。 オブジェクトを受け入れ、キーはプロパティ名、値は検索する文字列です。追加オプションがない場合、`LIKE "%…%"` 検索を実行します。 `string` および `text`データ型のプロパティのみサポートします。 *Admin* および *User* タイプのユーザーが自分のユーザーグループ内のユーザーのデータをリクエストする場合のユーザーオブジェクトの可能なプロパティ: `name`, `surname`, `username`。
sortfield	string/array	指定したプロパティで結果をソートします。 可能な値: `userid`, `username`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ユーザーの取得

設定されているすべてのユーザーを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "3",
```

```
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```

管理者としてユーザーを取得する

管理者タイプのユーザーとして、自分のユーザーに関する詳細データと、ユーザーグループ内のユーザーの限定データを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "userid": "3",
```

```
            "username": "database-admin",
            "name": "John",
            "surname": "Doe",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "2",
            "provisioned": "0",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```

ユーザーデータの取得

ID ”12” のユーザーのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
      }
  ],
  "id": 1
}
```

参照

- Media
- Media type
- User group
- Role

ソース

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.login

説明

`string/object user.login(object parameters)`

このメソッドは、API にログインして認証トークンを生成するためのものです。

> **Warning:**
> このメソッドを使用する場合は、多数のオープンセッションレコードの生成を防ぐために、user.logoutも実行する必要があります。

> **Attention:**
> このメソッドは、多要素認証が有効なユーザーグループに属していない未認証ユーザーのみが利用できます。

パラメーター

(object) ユーザー名とパスワードを含むパラメーター。

このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
password	string	ユーザーパスワード。 パラメーターの動作: - 必須
username	string	ユーザー名。 パラメーターの動作: - 必須
userData	flag	認証されたユーザーに関する情報を返します。

戻り値

(string/object) userData パラメータが使用された場合、認証されたユーザーに関する情報を含むオブジェクトを返します。

標準のユーザー属性に加えて、以下の情報が返されます。

プロパティ	タイプ	説明
auth_type	integer	ユーザーのデフォルト認証。 利用可能な値の一覧は、認証オブジェクトの authentication_type プロパティを参照してください。
debug_mode	integer	ユーザーのデバッグモードが有効か無効か。 利用可能な値の一覧は、ユーザーグループオブジェクトの debug_mode プロパティを参照してください。
deprovisioned	boolean	ユーザーが無効化されたユーザーグループに属しているかどうか。
gui_access	string	フロントエンドへのユーザーの認証方法。 利用可能な値の一覧は、ユーザーグループオブジェクトの gui_access プロパティを参照してください。
mfaid	integer	ログイン時にユーザーが使用するMFA メソッドの ID。 MFA がグローバルまたはユーザーが属するすべてのユーザーグループで無効化されている場合は「0」を返します。
secret	string	ランダムな 32 文字の文字列。ユーザーログイン時に生成されます。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。
type	integer	ユーザータイプ。 利用可能な値の一覧は、ロールオブジェクトの type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

Note:
1 回以上の認証失敗の後にユーザーが正常に認証された場合、このメソッドは attempt_clock、attempt_failed、attempt_ip プロパティの現在値を返し、その後リセットします。

userData パラメータが使用されていない場合、このメソッドは認証に必要な認証トークンを返します。

例

ユーザーの認証

ユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証されたユーザー情報のリクエスト

認証を行い、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "mfaid": "1",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "5b56eee8be445e98f0bd42b435736e42",
        "secret": "cd0ba923319741c6586f3d866423a8f4"
    },
    "id": 1
}
```

参照

- user.logout

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.logout

説明

`string/object user.logout(array)`

このメソッドを使用すると、API からログアウトして、現在の認証トークンを無効にすることができます。

> **Note:**
> このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーの役割設定で取り消すことができます。詳しくはユーザーの役割を参照してください。

パラメータ

(array) このメソッドは空の配列を受け入れます。

戻り値

(boolean) ユーザーが正常にログアウトされた場合は `true` を返します

例

ログアウト

API からログアウトします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

参照

- user.login

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

説明

`object user.provision(object/array users)`

このメソッドは、LDAP ユーザーをプロビジョニングできます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーの役割設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(array) プロビジョニングするユーザーの ID

戻り値

(object) userids プロパティの下でプロビジョニングされたユーザーの ID を含むオブジェクトを返します。

例

複数ユーザーのプロビジョニング

2 つのユーザーをプロビジョニングします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::provision() in *ui/include/classes/api/services/CUser.php*.

user.resettotp

説明

`object user.resettotp(object/array users)`

このメソッドは、ユーザーの TOTP シークレットをリセットします。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割をご覧ください。

パラメータ

(配列) TOTP シークレットをリセットするユーザーの ID

> **Note:**
> 指定されたユーザーのユーザーセッションも削除されます（リクエストを送信したユーザーを除く）。

戻り値

(object) TOTP シークレットがリセットされたユーザーの ID を userids プロパティに格納したオブジェクトを返します。

例

複数のユーザーの TOTP シークレットのリセット

2 人のユーザーの TOTP シークレットをリセットします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

参照

- MFA object

ソース

CUser::resettotp() in *ui/include/classes/api/services/CUser.php*.

user.unblock

説明

`object user.unblock(array userids)`

このメソッドは、ユーザーのブロックを解除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(array) ユーザーブロックを解除する ID

戻り値

(object) userids プロパティの下にあるブロックされていないユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーのブロック解除

2 人のユーザーのブロックを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

説明

`object user.update(object/array users)`

このメソッドは既存のユーザーを更新します。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

Note:
ユーザーパスワードの強度は、Authentication API で定義されたパスワードポリシールールに従って検証されます。詳細はAuthentication APIを参照してください。

パラメーター

(`object/array`) 更新するユーザーのプロパティ。

各ユーザーには `userid` プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のユーザープロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	タイプ	説明
current_passwd	string	ユーザーの現在のパスワード。 このパラメーターの値は、ユーザーがユーザーディレクトリにリンクされている場合、空文字列にすることができます。 パラメーターの動作: - *write-only* - User objectの `passwd` が設定されていて、ユーザー自身がパスワードを変更する場合は必須
usrgrps	array	既存のユーザーグループを置き換えるユーザーグループ。 ユーザーグループには `usrgrpid` プロパティのみを定義する必要があります。
medias	array	既存のプロビジョニングされていないメディアを置き換えるユーザーメディア。メディアを更新する際、プロビジョニングされたメディアは省略できます。

戻り値

(`object`) `userids` プロパティの下で更新されたユーザーの ID を含むオブジェクトを返します。

例

ユーザー名の変更

ユーザー名を John Doe に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

ユーザーロールの変更

ユーザーのロールを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

参照

- Authentication

ソース

CUser::update() in *ui/include/classes/api/services/CUser.php*.

User directory

このクラスは、ユーザーディレクトリを操作するためのものです。

オブジェクトリファレンス:

- User directory
- Media type mappings
- Provisioning groups mappings

利用可能なメソッド:

- userdirectory.create - 新しいユーザーディレクトリを作成する
- userdirectory.delete - ユーザーディレクトリを削除する
- userdirectory.get - ユーザーディレクトリを取得する
- userdirectory.update - ユーザーディレクトリを更新する
- userdirectory.test - ユーザーディレクトリの接続をテストする

User directory オブジェクト

次のオブジェクトは、userdirectoryAPI に直接関連しています。

ユーザーディレクトリ

ユーザーディレクトリオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
userdirectoryid	ID	ユーザーディレクトリの ID。 ユーザーディレクトリが削除された場合、削除されたユーザーディレクトリにリンクされているすべてのユーザーのユーザーオブジェクトプロパティ `userdirectoryid` の値は「0」に設定されます。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
idp_type	integer	ユーザーディレクトリのアイデンティティプロバイダーで使用される認証プロトコルのタイプ。 タイプ SAML のユーザーディレクトリは 1 つしか存在できないことに注意してください。 可能な値: 1 - LDAP タイプのユーザーディレクトリ; 2 - SAML タイプのユーザーディレクトリ。 プロパティの動作: - 作成操作時に必須
group_name	string	LDAP/SAML ユーザーディレクトリ属性で、LDAP/SAML ユーザーディレクトリと Zabbix 間でグループをマッピングするために使用されるグループ名を含みます。 例: *cn* プロパティの動作: - `provision_status` が「有効」に設定され、認証オブジェクトの `saml_jit_status` が「構成済み SAML IdP で有効」に設定されている場合に必須
user_username	string	ユーザーのプロビジョニング時にユーザーオブジェクトプロパティ `name` の値として使用されるユーザー名を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	ユーザーのプロビジョニング時にユーザーオブジェクトプロパティ `surname` の値として使用されるユーザーの姓を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *sn*, *surname*, *lastName*

プロパティ	タイプ	説明
provision_status	integer	ユーザーディレクトリのプロビジョニングステータス。 可能な値: 0 - *(デフォルト)* 無効（このユーザーディレクトリで作成されたユーザーのプロビジョニングは無効）； 1 - 有効（このユーザーディレクトリで作成されたユーザーのプロビジョニングは有効。また、LDAP または SAML プロビジョニング（認証オブジェクトの `ldap_jit_status` または `saml_jit_status`）のステータスも有効である必要があります）。
provision_groups	array	LDAP/SAML ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングするためのプロビジョニンググループマッピングオブジェクトの配列。 プロパティの動作: - `provision_status` が「有効」に設定されている場合に必須
provision_media	array	通知送信のためにユーザーの LDAP/SAML メディア属性（例: メール）を Zabbix ユーザーメディアにマッピングするためのメディアタイプマッピングオブジェクトの配列。
LDAP固有のプロパティ**:**		
name	string	ユーザーディレクトリの一意の名前。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合に必須
host	string	LDAP サーバーのホスト名、IP または URI。 URI にはスキーマ（`ldap://`または `ldaps://`）、ホスト、およびポート（オプション）を含める必要があります。 例: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合に必須
port	integer	LDAP サーバーのポート。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合に必須
base_dn	string	ユーザーアカウントへの LDAP ユーザーディレクトリのベースパス。 例: *ou=Users,dc=example,dc=org* *ou=Users,ou=system*（OpenLDAP の場合） *DC=company,DC=com*（Microsoft Active Directory の場合） *uid=%{user},dc=example,dc=com*（直接ユーザーバインディングの場合。「*%{user}*」プレースホルダーは必須） プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合に必須

プロパティ	タイプ	説明
search_attribute	string	ログインリクエストで提供された情報からユーザーアカウントを識別するための LDAP ユーザーディレクトリ属性。 例: *uid*（OpenLDAP の場合） *sAMAccountName*（Microsoft Active Directory の場合） プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合に必須
bind_dn	string	LDAP サーバー上でバインドおよび検索を行うための LDAP サーバーアカウント。 直接ユーザーバインディングおよび匿名バインディングの場合、`bind_dn` は空にする必要があります。 例: *uid=ldap_search,ou=system*（OpenLDAP の場合） *CN=ldap_search,OU=user_group,DC=company,DC=com*（Microsoft Active Directory の場合） *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
bind_password	string	LDAP サーバー上でバインドおよび検索を行うためのアカウントの LDAP パスワード。 直接ユーザーバインディングおよび匿名バインディングの場合、`bind_password` は空にする必要があります。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
description	string	ユーザーディレクトリの説明。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_basedn	string	グループへの LDAP ユーザーディレクトリのベースパス。LDAP ユーザーディレクトリでユーザーのメンバーシップチェックを構成するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 例: *ou=Groups,dc=example,dc=com* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
group_filter	string	ユーザーがメンバーである LDAP ユーザーディレクトリグループを取得するためのフィルタ文字列。LDAP ユーザーディレクトリでユーザーのメンバーシップチェックを構成するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 サポートされている `group_filter` プレースホルダー: *%{attr}* - 検索属性（`search_attribute` プロパティ値に置換）; *%{groupattr}* - グループ属性（`group_member` プロパティ値に置換）; *%{host}* - LDAP サーバーのホスト名、IP または URI（`host` プロパティ値に置換）; *%{user}* - Zabbix ユーザーのユーザー名。 デフォルト: *(%{groupattr}=%{user})* 例: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* は、LDAP グループオブジェクトが値「*uid=User1,ou=Users,dc=example,dc=com*」の「*member*」属性を含む場合、「User1」と一致し、「User1」がメンバーであるグループを返します; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* は、LDAP グループオブジェクトが `group_member` プロパティで指定された属性に値「*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*」を含む場合、「User1」と一致し、「User1」がメンバーであるグループを返します。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_member	string	グループメンバーに関する情報を含む LDAP ユーザーディレクトリ属性。LDAP ユーザーディレクトリでユーザーのメンバーシップチェックを構成するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_membership	string	ユーザーが所属するグループに関する情報を含む LDAP ユーザーディレクトリ属性。 例: *memberOf* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
search_filter	string	ログインリクエストで提供された情報に基づいて LDAP ユーザーディレクトリ内のユーザーを検索および認証するために使用されるカスタムフィルタ文字列。 サポートされている `search_filter` プレースホルダー: *%{attr}* - 検索属性名（例: *uid*, *sAMAccountName*）; *%{user}* - Zabbix ユーザーのユーザー名。 デフォルト: *(%{attr}=%{user})* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
start_tls	integer	LDAP サーバーとの通信をトランスポートレイヤーセキュリティ（TLS）で保護できる LDAP サーバー構成オプション。 `ldaps://`プロトコルを使用するホストの場合、`start_tls` は「無効」に設定する必要があります。 可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
user_ref_attr	string	ユーザーオブジェクトを参照するために使用される LDAP ユーザーディレクトリ属性。`user_ref_attr` の値は、ユーザーディレクトリ内の指定された属性から値を取得し、それらを `group_filter` 文字列の*%{ref}* プレースホルダーの代わりに配置するために使用されます。 例: *cn*, *uid*, *member*, *uniqueMember* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
SAML固有のプロパティ:		
idp_entityid	string	アイデンティティプロバイダーを識別し、SAML メッセージでアイデンティティプロバイダーと通信するために使用される URI。 例: *https://idp.example.com/idp* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合に必須
sp_entityid	string	アイデンティティプロバイダーのサービスプロバイダーを識別する URL または任意の文字列。 例: *https://idp.example.com/sp* *zabbix* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合に必須
username_attribute	string	認証時にユーザーオブジェクトプロパティ `username` の値と比較されるユーザー名を含む SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *uid*, *userprincipalname*, *samaccountname*, *username*, *userusername*, *urn:oid:0.9.2342.19200300.100.1.1*, *urn:oid:1.3.6.1.4.1.5923.1.1.1.13*, *urn:oid:0.9.2342.19200300.100.1.44* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合に必須

プロパティ	タイプ	説明
sso_url	string	Zabbix が SAML 認証リクエストを送信するアイデンティティプロバイダーの SAML シングルサインオンサービスの URL。 例: *http://idp.example.com/idp/sso/saml* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合に必須
slo_url	string	Zabbix が SAML ログアウトリクエストを送信するアイデンティティプロバイダーの SAML シングルログアウトサービスの URL。 例: *https://idp.example.com/idp/slo/saml* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
encrypt_nameid	integer	SAML の name ID を暗号化するかどうか。 可能な値: 0 - *(デフォルト)* name ID を暗号化しない; 1 - name ID を暗号化する。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
encrypt_assertions	integer	SAML アサーションを暗号化するかどうか。 可能な値: 0 - *(デフォルト)* アサーションを暗号化しない; 1 - アサーションを暗号化する。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
nameid_format	string	SAML アイデンティティプロバイダーのサービスプロバイダーの Name ID フォーマット。 例: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
scim_status	integer	SAML の SCIM プロビジョニングが有効か無効か。 可能な値: 0 - *(デフォルト)* SCIM プロビジョニングは無効; 1 - SCIM プロビジョニングは有効。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
sign_assertions	integer	SAML アサーションを SAML 署名で署名するかどうか。 可能な値: 0 - *(デフォルト)* アサーションに署名しない; 1 - アサーションに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_authn_requests	integer	SAML AuthN リクエストを SAML 署名で署名するかどうか。 可能な値: 0 - *(デフォルト)* AuthN リクエストに署名しない; 1 - AuthN リクエストに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_messages	integer	SAML メッセージを SAML 署名で署名するかどうか。 可能な値: 0 - *(デフォルト)* メッセージに署名しない; 1 - メッセージに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_logout_requests	integer	SAML ログアウトリクエストを SAML 署名で署名するかどうか。 可能な値: 0 - *(デフォルト)* ログアウトリクエストに署名しない; 1 - ログアウトリクエストに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_logout_responses	integer	SAML ログアウトレスポンスを SAML 署名で署名するかどうか。 可能な値: 0 - *(デフォルト)* ログアウトレスポンスに署名しない; 1 - ログアウトレスポンスに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
idp_certificate	string	SAML シングルサインオン（SSO）サービスセットアップ用のサービスプロバイダー（SP）証明書の内容。 例: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で `$SSO['CERT_STORAGE']` が `database` に設定されている場合にサポート - 書き込み専用

プロパティ	タイプ	説明
sp_private_key	string	SAML シングルサインオン（SSO）サービスセットアップ用のサービスプロバイダー（SP）秘密鍵の内容。アイデンティティプロバイダー（IdP）との安全な認証およびデータ交換を提供します。 例: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 書き込み専用
sp_certificate	string	SAML シングルサインオン（SSO）サービスセットアップ用のサービスプロバイダー（SP）証明書の内容。 例: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 書き込み専用
idp_certificate_hash	string	idp_certificate 値の md5 ハッシュ。idp_certificate が空の場合は空文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用
sp_private_key_hash	string	sp_private_key 値の md5 ハッシュ。sp_private_key が空の場合は空文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用
sp_certificate_hash	string	sp_certificate 値の md5 ハッシュ。sp_certificate が空の場合は空文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、zabbix.conf.php で $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用

メディアタイプマッピング

メディアタイプマッピングオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
userdirectory_mediaid	ID	メディアタイプマッピング ID プロパティの動作: - 読み取り専用
name	文字列	メディアタイプマッピングのリストに表示される名前 プロパティの動作: - 必須
mediatypeid	ID	作成するメディアタイプの ID メディアオブジェクト プロパティ `mediatypeid` の値として使用されます。 プロパティの動作: - 必須
attribute	string	LDAP/SAML ユーザーディレクトリ属性（`scim_status` が”SCIM プロビジョニングが有効” に設定されている場合は SCIM 属性）で、ユーザーのメディア（例：*user@example.com*）が含まれます。このメディアは、メディアオブジェクト プロパティ `sendto` の値として使用されます。 LDAP/SAML アイデンティティプロバイダーから受信したデータに存在し、値が空でない場合、プロビジョニングされたユーザーのメディア作成がトリガーされます。 プロパティの動作: - 必須
active	integer	プロビジョニングされたユーザーのメディアが作成されるときのユーザーメディアの `active` プロパティ値 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効
severity	整数	プロビジョニングされたユーザーに対してメディアが作成される際のユーザーメディアの `severity` プロパティ値 デフォルト: 63
period	文字列	プロビジョニングされたユーザーに対してメディアが作成される際のユーザーメディアの `period` プロパティ値 デフォルト: 1～7、00:00～24:00

プロビジョニンググループのマッピング

プロビジョニンググループのマッピングには、以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	LDAP/SAML ユーザーディレクトリ（`scim_status` が”SCIM プロビジョニングが有効” に設定されている場合は SCIM も）内のグループのフルネーム（例：*Zabbix administrators*） ワイルドカード文字”*” をサポートします。 すべてのプロビジョニンググループのマッピングで一意です。 プロパティの動作: - 必須
roleid	ID	ユーザーに割り当てるユーザーロールの ID 複数のプロビジョニンググループマッピングが一致した場合、最も高いユーザータイプ（*User*、*Admin*、または *Super admin*）のロールがユーザーに割り当てられます。同じユーザータイプのロールが複数ある場合は、最初のロール（アルファベット順）がユーザーに割り当てられます。 プロパティの動作: - 必須

プロパティ	タイプ	説明
user_groups	array	Zabbix ユーザーグループ ID オブジェクトの配列各オブジェクトには次のプロパティがあります: usrgrpid - (ID) ユーザーに割り当てる Zabbix ユーザーグループの ID 複数のプロビジョニンググループマッピングが一致する場合、一致したすべてのマッピングの Zabbix ユーザーグループがユーザーに割り当てられます。 プロパティの動作: - 必須

userdirectory.create

説明

`object userdirectory.create(object/array userDirectory)`

このメソッドは、新しいユーザーディレクトリを作成します。

> **Note:**
> このメソッドは、*Super admin* のユーザーのみ利用可能です。

パラメータ

(`object/array`) 作成するユーザーディレクトリ。

このメソッドは、標準ユーザーディレクトリプロパティを持つユーザーディレクトリを受け入れます。

戻り値

(`object`) は、`userdirectoryids` プロパティに作成されたユーザーディレクトリの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーディレクトリの順序と一致します。

例

ユーザーディレクトリの作成

StartTLS over LDAP を使用してユーザーを認証するためのユーザーディレクトリを作成します。LDAP 経由でユーザーを認証するには、LDAP 認証 を有効にする必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
        "idp_type": "1",
        "name": "LDAP API server #1",
        "host": "ldap://local.ldap",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "ldapsecretpassword",
        "search_attribute": "uid",
        "start_tls": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

ユーザーディレクトリの作成（JIT プロビジョニング有効）

LDAP 経由でユーザーを認証するためのユーザーディレクトリを作成します。(JIT プロビジョニング有効）LDAP 経由でユーザーを認証するには、LDAP 認証 が有効になっている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

説明

`object userdirectory.delete(array userDirectoryIds)`

このメソッドを使用すると、ユーザーディレクトリを削除できます。ユーザーディレクトリが 1 つでもユーザーグループに直接使用されている場合は、削除できません。
`authentication.ldap_configured` が 1 に設定されている場合、またはユーザーディレクトリがまだ残っている場合、デフォルトの LDAP ユーザーディレクトリは削除できません。

> **Note:**
> このメソッドが使用できるのは、*Super admin* ユーザータイプのみです。

パラメータ

(`array`) 削除するユーザーディレクトリの ID。

戻り値

(`object`) `userdirectoryids` プロパティの下にある削除されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

複数のユーザーディレクトリの削除

2 つのユーザーディレクトリを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
```

```
    "id": 1
}
```

ソース

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

説明

`integer/array userdirectory.get(object parameters)`

このメソッドは、指定したパラメーターに応じたユーザーディレクトリを取得できます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。

パラメータ

(`object`) 出力を定義するパラメータ

このメソッドは以下のパラメータをサポートしています。

パラメータ	タイプ	説明
userdirectoryids	ID/配列	指定された ID を持つユーザーディレクトリのみを返します。
selectUsrgrps	クエリ	ユーザーディレクトリに関連付けられたユーザーグループを持つ `usrgrps` プロパティを返します。 `count` をサポートします。
selectProvisionMedia	クエリ	ユーザーディレクトリに関連付けられたメディアタイプマッピングを持つ `provision_media` プロパティを返します。
selectProvisionGroups	クエリ	ユーザーディレクトリに関連付けられたプロビジョニンググループマッピング を持つ `provision_groups` プロパティを返します。
sortfield	文字列/配列	指定されたプロパティで結果を並べ替えます。 使用可能な値: `name`
filter	オブジェクト	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が単一の値または値の配列であるオブジェクトを受け入れます。 サポートされるプロパティ: `userdirectoryid`、`idp_type`、`provision_status`
search	オブジェクト	指定されたパターンに一致する結果を返します（大文字と小文字は区別されません） キーがプロパティ名、値が検索する文字列であるオブジェクトを受け入れます。追加オプションが指定されていない場合は、`LIKE "%...%"` 検索が実行されます。 サポートされているプロパティ: `name`、`description` SAML タイプのユーザーディレクトリでは、`name` プロパティと `description` プロパティの両方が空の値になります。どちらのプロパティも `userdirectory.update` 操作で変更できます。
countOutput	boolean	これらのパラメータについては、リファレンス解説 で説明されています。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

パラメータ	タイプ	説明
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ユーザーディレクトリの取得

各ユーザーディレクトリに関連付けられたメディアタイプマッピングとプロビジョニンググループマッピングを表示する追加プロパティを含むすべてのユーザーディレクトリを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
            "name": "",
            "provision_status": "1",
            "description": "",
            "group_name": "groups",
            "user_username": "",
            "user_lastname": "",
            "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
            "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
            "slo_url": "",
            "username_attribute": "uid",
            "sp_entityid": "zabbix",
            "nameid_format": "",
            "sign_messages": "0",
            "sign_assertions": "0",
            "sign_authn_requests": "0",
            "sign_logout_requests": "0",
            "sign_logout_responses": "0",
            "encrypt_nameid": "0",
            "encrypt_assertions": "0",
            "scim_status": "1",
            "provision_media": [
                {
                    "userdirectory_mediaid": "1",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
```

```
                "severity": "63",
                "period": "1-7,00:00-24:00"
            }
        ],
        "provision_groups": [
            {
                "name": "*",
                "roleid": "1",
                "user_groups": [
                    {
                        "usrgrpid": "13"
                    }
                ]
            }
        ]
    },
    {
        "userdirectoryid": "2",
        "idp_type": "1",
        "name": "AD server",
        "provision_status": "1",
        "description": "",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "start_tls": "0",
        "search_filter": "",
        "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
        "group_name": "CN",
        "group_member": "member",
        "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
        "group_membership": "",
        "user_username": "givenName",
        "user_lastname": "sn",
        "user_ref_attr": "CN",
        "provision_media": [
            {
                "userdirectory_mediaid": "2",
                "name": "example.com",
                "mediatypeid": "1",
                "attribute": "user@example.com",
                "active": "0",
                "severity": "63",
                "period": "1-7,00:00-24:00"
            }
        ],
        "provision_groups": [
            {
                "name": "*",
                "roleid": "4",
                "user_groups": [
                    {
                        "usrgrpid": "8"
                    }
                ]
            },
            {
                "name": "Zabbix administrators",
                "roleid": "2",
                "user_groups": [
```

```
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
    "id": 1
}
```


参照

- User group

ソース

CUserDirectory::get() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

説明

`object userdirectory.test(array userDirectory)`

このメソッドは、ユーザーディレクトリの接続設定をテストします。

Note:
このメソッドは、ユーザープロビジョニングのユーザーディレクトリ設定と一致する設定データ（例：ユーザーに割り当てられるユーザーロール、ユーザーグループ、ユーザーメディア）をテストすることもできます。このタイプのテストでは、`provision_status` が有効に設定されているユーザーディレクトリ に対して API リクエストを送信する必要があります。

Note:
このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。

パラメータ

(object) ユーザーディレクトリのプロパティ

userdirectory.get API は bind_password フィールドを返さないため、userdirectoryid または bind_password を指定する必要があります。標準のユーザーディレクトリのプロパティに加えて、このメソッドは以下のパラメータを受け入れます。

パラメータ	タイプ	説明
test_username	文字列	ユーザーディレクトリでテストするユーザー名
test_password	文字列	ユーザーディレクトリでテストするユーザー名に関連付けられたパスワード

戻り値

(bool) 成功した場合は true を返します。

例

既存のユーザーのユーザーディレクトリのテスト

"user1" のユーザーディレクトリ"3" をテストします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "password",
        "test_username": "user1",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

存在しないユーザーのユーザーディレクトリのテスト

存在しない"user2" でユーザーディレクトリ"3" のテストをします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "test_username": "user2",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

ユーザープロビジョニングのユーザーディレクトリのテスト

"user3" プロビジョニングのユーザーディレクトリ設定と一致する構成されたデータのユーザーディレクトリ"3" をテストします (例えば、何かユーザーの役割、ユーザーグループ、ユーザーメディアがユーザーに割り当てられます)。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "2",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "test_username": "user3",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

ソース

CUserDirectory::test() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

説明

`object userdirectory.update(object/array userDirectory)`

このメソッドは、既存のユーザーディレクトリを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。

パラメーター

(object/array) 更新するユーザーディレクトリのプロパティ。

userdirectoryid プロパティは各ユーザーディレクトリに対して定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

戻り値

(object) userdirectoryids プロパティの下にある更新されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

ユーザーディレクトリのバインドパスワードの更新

ユーザーディレクトリの新しいバインドパスワードを設定します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

ユーザーディレクトリのマッピングの更新

ユーザーディレクトリ"2" のプロビジョニンググループマッピングとメディアタイプマッピングを更新します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "2",
        "provision_media": [
                {
                    "userdirectory_mediaid": "2"
                }
            ],
            "provision_groups": [
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
```

```
                        {
                            "usrgrpid": "8"
                        },
                        {
                            "usrgrpid": "11"
                        }
                    ]
                }
            ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::update() in *ui/include/classes/api/services/CUserDirectory.php*.

User group

このクラスは、ユーザーグループを操作するためのものです。

オブジェクトリファレンス:

- User group
- Permission
- Tag-based permission

利用可能なメソッド:

- usergroup.create - 新しいユーザーグループを作成する
- usergroup.delete - ユーザーグループを削除する
- usergroup.get - ユーザーグループを取得する
- usergroup.update - ユーザーグループを更新する

User group オブジェクト

次のオブジェクトは、usergroupAPI に直接関連しています。

ユーザーグループ

ユーザーグループオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ユーザーグループの名前。 プロパティの動作: - 作成操作では必須

プロパティ	タイプ	説明
debug_mode	integer	デバッグモードが有効か無効か。 可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
gui_access	integer	グループ内ユーザーのフロントエンド認証方式。 可能な値: 0 - *(デフォルト)* システムのデフォルト認証方式を使用; 1 - 内部認証を使用; 2 - LDAP 認証を使用; 3 - フロントエンドへのアクセスを無効化。
mfa_status	integer	グループ内ユーザーの MFA が有効か無効か。 可能な値: 0 - 無効（すべての設定済み MFA 方式に対して）; 1 - 有効（すべての設定済み MFA 方式に対して）。
mfaid	ID	グループ内ユーザーに使用されるMFA 方式。 プロパティの動作: - スーパ管理者タイプのユーザーでサポート - 認証オブジェクトの `mfa_status` が「有効」に設定されている場合にサポート
users_status	integer	ユーザーグループが有効か無効か。 解除済みユーザーの場合、ユーザーグループは有効にできません。 可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
userdirectoryid	ID	認証に使用されるユーザーディレクトリの ID。 プロパティの動作: - スーパ管理者タイプのユーザーでサポート - `gui_access` が「システムのデフォルト認証方式を使用」または「LDAP 認証を使用」に設定されている場合にサポート

Permission

permission オブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
id	ID	権限を追加するホストグループまたはテンプレートグループの ID。 プロパティの動作: - 必須
permission	integer	ホストグループまたはテンプレートグループへのアクセスレベル。 可能な値: 0 - アクセス拒否; 2 - 読み取り専用アクセス; 3 - 読み取り/書き込みアクセス。 プロパティの動作: - 作成操作時に必須

タグベースの権限

タグベースの権限オブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
groupid	ID	権限を追加するホストグループの ID。 プロパティの動作: - 必須
tag	string	タグ名。
value	string	タグ値。

usergroup.create

説明

`object usergroup.create(object/array userGroups)`

このメソッドは新しいユーザーグループを作成します。

Note:
このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 作成するユーザーグループ。

標準のユーザーグループプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
hostgroup_rights	object/array	ユーザーグループに割り当てるホストグループの権限。
templategroup_rights	object/array	ユーザーグループに割り当てるテンプレートグループの権限。
tag_filters	array	ユーザーグループに割り当てるタグベースの権限。
users	object/array	ユーザーグループに追加するユーザー。 ユーザーには `userid` プロパティのみが定義されている必要があります。

戻り値

(object) usrgrpids プロパティの下で作成されたユーザーグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーグループの順序と一致します。

例

ユーザーグループの作成

ホストグループ"2" へのアクセスが拒否されたユーザーグループ *Operation managers* を作成し、ユーザーを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

参照

- Permission

ソース

CUserGroup::create() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

説明

`object usergroup.delete(array userGroupIds)`

このメソッドは、ユーザーグループを削除することができます。

Attention:
プロビジョニング解除されたユーザーグループ (認証の `disabled_usrgrpid` に指定されたユーザーグループ) は削除できません。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除するユーザーグループの ID。

戻り値

(`object`) `usrgrpids` プロパティの下にある削除されたユーザーグループの ID を含むオブジェクトを返します。

例

複数のユーザーグループの削除

2 つのユーザーグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
```

```
            "21"
        ]
    },
    "id": 1
}
```

ソース

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

説明

`integer/array usergroup.get(object parameters)`

このメソッドは、指定されたパラメータに従ってユーザーグループを取得することができます。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 取得したい情報を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
mfaids	ID/array	指定した MFA 方式を持つユーザーグループのみを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
mfa_status	integer	指定した MFA ステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧はユーザーグループページを参照してください。
status	integer	指定したステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧はユーザーグループページを参照してください。
userids	ID/array	指定したユーザーを含むユーザーグループのみを返します。
usrgrpids	ID/array	指定した ID のユーザーグループのみを返します。
selectTagFilters	query	ユーザーグループのタグベースの権限を`tag_filters`プロパティで返します。
selectUsers	query	ユーザーグループのユーザーを`users`プロパティで返します。 ユーザータイプによる制限については`user.get`を参照してください。
selectHostGroupRights	query	ユーザーグループのホストグループ権限を `hostgroup_rights` プロパティで返します。 ホストグループへのアクセスレベルの一覧はユーザーグループページを参照してください。
selectTemplateGroupRights	query	ユーザーグループのテンプレートグループ権限を `templategroup_rights` プロパティで返します。 テンプレートグループへのアクセスレベルの一覧はユーザーグループページを参照してください。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。

パラメーター	型	説明
output	query	返されるユーザーグループオブジェクトのプロパティ。 *Admin* および *User* タイプのユーザーは、`usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status` のみ取得可能です。 デフォルト: `extend`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `usrgrpid`, `name`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

有効なユーザーグループの取得

すべての有効なユーザーグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
```

```
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


参照

- User

ソース

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

説明

`object usergroup.update(object/array userGroups)`

このメソッドは、既存のユーザーグループを更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object/array) 更新するユーザーグループのプロパティ。

各ユーザーグループには usrgrpid プロパティを定義する必要があり、他のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のすべては変更されません。

標準のユーザーグループプロパティに加えて、このメソッドは以下のパラメーターを受け付けます。

パラメーター	型	説明
hostgroup_rights	object/array	ユーザーグループに割り当てられている現在の権限を置き換えるホストグループの権限。
templategroup_rights	object/array	ユーザーグループに割り当てられている現在の権限を置き換えるテンプレートグループの権限。
tag_filters	array	ユーザーグループに割り当てられている現在の権限を置き換えるタグベースの権限。
users	object/array	ユーザーグループに割り当てられている現在のユーザーを置き換えるユーザー。 ユーザーには userid プロパティのみを定義する必要があります。

戻り値

(object) usrgrpids プロパティの下で更新されたユーザーグループの ID を含むオブジェクトを返します。

例

ユーザーグループの有効化と権限の更新

ユーザーグループを有効にし、ホストグループ"2" および"4" への読み書きアクセス権を付与します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
```

```
    },
    "id": 1
}
```

参照

- Permission

ソース

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

Value map

このクラスは、バリューマップを操作するためのものです。

オブジェクトリファレンス:

- Value map
- Value mappings

利用可能なメソッド:

- valuemap.create - 新しいバリューマップを作成する
- valuemap.delete - バリューマップを削除する
- valuemap.get - バリューマップを取得する
- valuemap.update - バリューマップを更新する

Value map オブジェクト

次のオブジェクトは、valuemapAPI に直接関連しています。

値のマップ

値のマップオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
valuemapid	ID	値のマップの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
hostid	ID	値のマップが属するホストまたはテンプレートの ID。 プロパティの動作: - 定数 - 作成操作時に必須
name	string	値のマップの名前。 プロパティの動作: - 作成操作時に必須
mappings	array	現在の値のマップの値のマッピング。マッピングオブジェクトは下記で詳細に説明されています。 プロパティの動作: - 作成操作時に必須
uuid	string	インポートされた値のマップを既存のものにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - 値のマップがテンプレートに属している場合にサポート

値のマッピング

値のマッピングオブジェクトは、値マップの値のマッピングを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	マッピングの一致タイプ。 可能な値: 0 - (デフォルト) 値が等しい場合にマッピングを適用; 1 - 値が以上の場合にマッピングを適用[1]; 2 - 値が以下の場合にマッピングを適用[1]; 3 - 値が範囲内の場合にマッピングを適用 (範囲は両端を含む; 複数の範囲はカンマ区切りで定義可能)[1]; 4 - 値が正規表現に一致する場合にマッピングを適用[2]; 5 - 一致が見つからない場合、マッピングは適用されず、デフォルト値が使用される。 type が"0"、"1"、"2"、"3"、"4" に設定されている場合、value は空にできません。 type が"5" に設定されている場合、value は空でなければなりません。
value	string	元の値。 プロパティの動作: - type が"1"、"2"、"3"、"4" に設定されている場合は必須 - type が"5" に設定されている場合はサポート
newvalue	string	元の値がマッピングされる値。 プロパティの動作: - 必須

[1] 値タイプが"numeric unsigned"、"numeric float" のアイテムのみサポート。
[2] 値タイプが"character" のアイテムのみサポート。

valuemap.create

説明

`object valuemap.create(object/array valuemaps)`

このメソッドは、新しいバリューマップを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) バリューマップ作製

このメソッドは、標準バリューマップのプロパティを持つバリューマップを受け入れます。

戻り値

(object) valuemapids プロパティの下に作成されたバリューマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたバリューマップの順序と一致します。

例

バリューマップの作成

2 つのマッピングを使用して 1 つのバリューマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
```

```
        "hostid": "50009",
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

説明

`object valuemap.delete(array valuemapids)`

このメソッドは、バリューマップを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するバリューマップの ID。

戻り値

(object) valuemapids プロパティの下にある削除されたバリューマップの ID を含むオブジェクトを返します。

例

複数のバリューマップの削除

2 つのバリューマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
    "params": [
        "1",
        "2"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

説明

`integer/array valuemap.get(object parameters)`

このメソッドは、指定されたパラメータに従って値マップを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 望ましい出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
valuemapids	ID/array	指定した ID の値のマップのみを返します。
selectMappings	query	現在の値のマップの値のマッピングを`mappings`プロパティで返します。
sortfield	string/array	`count` をサポートします。 指定したプロパティで結果をソートします。 指定可能な値: `valuemapid`, `name`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

値のマップの取得

すべての設定済みの値のマップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status"
        },
        {
            "valuemapid": "5",
            "name": "APC Battery Status"
        },
        {
            "valuemapid": "7",
            "name": "Dell Open Manage System Status"
        }
    ],
    "id": 1
}
```

マッピング付きで 1 つの値のマップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
                },
                {
```

```
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
        }
    ],
    "id": 1
}
```


ソース

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

説明

`object valuemap.update(object/array valuemaps)`

このメソッドは、既存の値のマッピングを更新するためのものです。

> **Note:**
> このメソッドはスーパ管理者ユーザータイプでのみ利用可能です。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールをご覧ください。

パラメータ

(object/array) バリューマップのプロパティ が更新されます。

valuemapid プロパティは、バリューマップごとに設定する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) valuemapids プロパティの下にある更新されたバリューマップの ID を含むオブジェクトを返します。

例

値のマップ名の変更

値のマップ名を「Device status」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

1 つの値のマッピングを変更する。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

Web scenario

このクラスは、Web シナリオを操作するためのものです。

オブジェクトリファレンス:

- Web scenario
- Scenario step
  - HTTP field
- Web scenario tag

利用可能なメソッド:

- httptest.create - 新しい Web シナリオを作成する
- httptest.delete - Web シナリオを削除する
- httptest.get - Web シナリオを取得する
- httptest.update - Web シナリオを更新する

Web scenario オブジェクト

次のオブジェクトは、`webcheck`API に直接関連しています。

Web シナリオ

Web シナリオオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
httptestid	ID	Web シナリオの ID。 プロパティの動作: - 読み取り専用 - 更新操作時に必須
hostid	ID	Web シナリオが属するホストの ID。 プロパティの動作: - 定数 - 作成操作時に必須
name	string	Web シナリオの名前。 プロパティの動作: - 作成操作時に必須
agent	string	Web シナリオで使用されるユーザーエージェント文字列。 デフォルト: Zabbix
authentication	integer	Web シナリオで使用される認証方式。 指定可能な値: 0 - *(デフォルト)* なし; 1 - 基本 HTTP 認証; 2 - NTLM 認証。
delay	string	Web シナリオの実行間隔。 秒またはサフィックス付きの時間単位（例: 30s, 1m, 2h, 1d）、またはユーザーマクロを受け付けます。 デフォルト: 1m。
headers	array	リクエスト実行時に送信されるHTTP ヘッダー。
http_password	string	基本 HTTP または NTLM 認証に使用されるパスワード。
http_proxy	string	Web シナリオで使用されるプロキシ。 *http://[username[:password]@]proxy.example.com[:port]* の形式で指定。
http_user	string	基本 HTTP または NTLM 認証に使用されるユーザー名。
retries	integer	Web シナリオが各ステップを失敗するまでに実行を試みる回数。 デフォルト: 1。
ssl_cert_file	string	クライアント認証に使用される SSL 証明書ファイル名（PEM 形式である必要があります）。

プロパティ	タイプ	説明
ssl_key_file	string	クライアント認証に使用される SSL 秘密鍵ファイル名（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵のパスワード。
status	integer	Web シナリオが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
templateid	ID	親テンプレート Web シナリオの ID。 プロパティの動作: - 読み取り専用
variables	array	Web シナリオの変数。
verify_host	integer	接続先のホスト名がホストの証明書のものと一致するかどうかを検証するか。 指定可能な値: 0 - *(デフォルト)* ホスト検証をスキップ; 1 - ホストを検証。
verify_peer	integer	ホストの証明書が正当かどうかを検証するか。 指定可能な値: 0 - *(デフォルト)* ピア検証をスキップ; 1 - ピアを検証。
uuid	string	インポートされた Web シナリオを既存のものにリンクするために使用されるグローバル一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - Web シナリオがテンプレートに属する場合にサポート

シナリオステップ

シナリオステップオブジェクトは、特定の Web シナリオチェックを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	シナリオステップの名前。 プロパティの動作: - 必須
no	integer	Web シナリオ内のステップのシーケンス番号。 プロパティの動作: - 必須
url	string	チェックする URL。 プロパティの動作: - 必須
follow_redirects	integer	HTTP リダイレクトを追跡するかどうか。 可能な値: 0 - リダイレクトを追跡しない; 1 - *(デフォルト)* リダイレクトを追跡する。
headers	array	リクエスト実行時に送信されるHTTP ヘッダー。シナリオステップのヘッダーは、Web シナリオで指定されたヘッダーを上書きします。
posts	string/array	HTTP POST 変数。文字列 (生の POST データ) またはHTTP フィールドの配列 (フォームフィールドデータ) として指定します。
required	string	レスポンス内に存在しなければならないテキスト。

プロパティ	タイプ	説明
retrieve_mode	integer	シナリオステップが取得する必要がある HTTP レスポンスの部分。 可能な値: 0 - *(デフォルト)* ボディのみ; 1 - ヘッダーのみ; 2 - ヘッダーとボディ。
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定。
timeout	string	リクエストのタイムアウト (秒)。秒数、単位付きの時間、またはユーザーマクロを指定可能。 デフォルト: 15s。最大: 1h。最小: 1s。
variables	array	シナリオステップの変数。
query_fields	array	クエリフィールド - リクエスト実行時に URL に追加されるHTTP フィールドの配列。

HTTP フィールド

HTTP フィールドオブジェクトは、Web シナリオの変数、HTTP ヘッダー、POST フィールドまたはクエリフィールドを指定するために使用される名前と値を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
name	string	ヘッダー/変数/POST または GET フィールドの名前。 プロパティの動作: - 必須
value	string	ヘッダー/変数/POST または GET フィールドの値。 プロパティの動作: - 必須

Web シナリオタグ

Web シナリオタグオブジェクトには、以下のプロパティがあります。

プロパティ	タイプ	説明
tag	string	Web シナリオタグ名。 プロパティの動作: - 必須
value	string	Web シナリオタグの値。

httptest.create

説明

`object httptest.create(object/array webScenarios)`

このメソッドは新しい Web シナリオを作成するためのものです。

Note:
Web シナリオを作成すると、自動的にWeb 監視アイテム のセットが作成されます。

Note:
このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成する Web シナリオ。

標準の Web シナリオプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
steps	array	シナリオステップ。 パラメータの動作: - 必須
tags	array	Web シナリオタグ。

戻り値

(object) httptestids プロパティの下で作成された Web シナリオの ID を含むオブジェクトを返します。返される ID の順序は、渡された Web シナリオの順序と一致します。

例

Web シナリオの作成

会社のホームページを監視するための Web シナリオを作成します。シナリオには 2 つのステップがあり、ホームページと「About」ページをチェックし、HTTP ステータスコード 200 が返されることを確認します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.create",
    "params": {
        "name": "Homepage check",
        "hostid": "10085",
        "steps": [
            {
                "name": "Homepage",
                "url": "http://example.com",
                "status_codes": "200",
                "no": 1
            },
            {
                "name": "Homepage / About",
                "url": "http://example.com/about",
                "status_codes": "200",
                "no": 2
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

参照

- Scenario step

ソース

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

説明

`object httptest.delete(array webScenarioIds)`

このメソッドは、Web シナリオを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する Web シナリオの ID。

戻り値

(object) httptestids プロパティの下にある削除された Web シナリオの ID を含むオブジェクトを返します。

例

複数の Web シナリオの削除

2 つの Web シナリオを削除します

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

ソース

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

説明

`integer/array httptest.get(object parameters)`

このメソッドは、指定されたパラメータに従って Web シナリオを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(object) 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
groupids	ID/array	指定したホストグループに属する Web シナリオのみを返します。
hostids	ID/array	指定したホストに属する Web シナリオのみを返します。
httptestids	ID/array	指定した ID の Web シナリオのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承された Web シナリオのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効な Web シナリオのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属する Web シナリオのみを返します。
templateids	ID/array	指定したテンプレートに属する Web シナリオのみを返します。
expandName	flag	Web シナリオ名のマクロを展開します。
expandStepName	flag	シナリオステップ名のマクロを展開します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグを持つ Web シナリオのみを返します。 フォーマット: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべての Web シナリオを返します。 指定可能なoperator値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
selectHosts	query	Web シナリオが属するホストを`hosts`プロパティの配列として返します。
selectSteps	query	Web シナリオのステップを`steps`プロパティで返します。 `count` をサポートします。
selectTags	query	Web シナリオのタグを`tags`プロパティで返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `httptestid`, `name`.
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

`(integer/array)` 次のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用された場合は、取得したオブジェクトの数

例

ウェブシナリオの取得

ウェブシナリオ”9” のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "variables": [],
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "headers": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
```

```
                    "no": "2",
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Host
- Scenario step

ソース

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

説明

`object httptest.update(object/array webScenarios)`

このメソッドは、既存の Web シナリオを更新するためのものです。

> **Note:**
> このメソッドは、管理者およびスーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新する Web シナリオのプロパティ。

各 Web シナリオには `httptestid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準の Web シナリオプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
steps	array	既存のステップを置き換えるシナリオステップ。
tags	array	Web シナリオタグ。

戻り値

(object) `httptestid` プロパティの下で更新された Web シナリオの ID を含むオブジェクトを返します。

例

Web シナリオの有効化

Web シナリオを有効にするには、status を”0” に設定します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
```

```
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

参照

- Scenario step

ソース

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

ユーザーマクロ

このクラスは、ホストレベルおよびグローバルユーザーマクロで動作するように設計されています。

オブジェクト参照:

- グローバルマクロ
- ホストマクロ
    - マクロ設定

利用可能なメソッド:

- usermacro.create - 新しいホストマクロの作成
- usermacro.createglobal - 新しいグローバルマクロの作成
- usermacro.delete - ホストマクロの削除
- usermacro.deleteglobal - グローバルマクロの削除
- usermacro.get - ホストおよびグローバルマクロの取得
- usermacro.update - ホストマクロの更新
- usermacro.updateglobal - グローバルマクロの更新

User macro オブジェクト

次のオブジェクトは、usermacroAPI に直接関連しています。

グローバルマクロ

グローバルマクロオブジェクトには以下のプロパティがあります。

プロパティ	タイプ	説明
globalmacroid	ID	グローバルマクロの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
description	string	マクロの説明。
macro	string	マクロ文字列。 プロパティの動作: - 作成操作では必須

プロパティ	タイプ	説明
type	integer	マクロのタイプ。 可能な値: 0 - *(デフォルト)* テキストマクロ; 1 - シークレットマクロ; 2 - Vault シークレット。
value	string	マクロの値。 プロパティの動作: - type が” シークレットマクロ” の場合は書き込み専用 - 作成操作では必須

ホストマクロ

ホストマクロオブジェクトは、ホスト、ホストプロトタイプ、またはテンプレートで利用可能なマクロを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
hostmacroid	ID	マクロの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
automatic	integer	マクロがディスカバリールールによって制御されているかどうかを定義します。 可能な値: 0 - *(デフォルト)* マクロはユーザーによって管理される; 1 - マクロはディスカバリールールによって管理される。 ユーザーは自動マクロを作成できません。 自動マクロを更新するには、手動に変換する必要があります。
config	object/array	ホストウィザードでマクロの表示方法を決定するマクロ設定。
description	string	マクロの説明。
hostid	ID	マクロが属するホスト、ホストプロトタイプ、またはテンプレートの ID。 プロパティの動作: - 定数 - 作成操作では必須
macro	string	マクロ文字列。 プロパティの動作: - 作成操作では必須
type	integer	マクロのタイプ。 可能な値: 0 - *(デフォルト)* テキストマクロ; 1 - シークレットマクロ; 2 - Vault シークレット。
value	string	マクロの値。 プロパティの動作: - type が「シークレットマクロ」に設定されている場合は書き込み専用 - 作成操作では必須

マクロ設定

マクロ設定オブジェクトは、ホストウィザードでマクロがどのように表示されるかを定義します。

プロパティ	型	説明
type	integer	マクロ入力フィールドのタイプ。 設定可能な値: 0 - ホストウィザードでマクロを使用しない; 1 - テキストボックス; 2 - リスト; 3 - チェックボックス。 プロパティの動作: - 必須
label	string	マクロ入力フィールドのラベル。 プロパティの動作: - type が”Textbox”、”List”、”Checkbox” に設定されている場合は必須
description	text	マクロ入力フィールドの横に表示されるヘルプテキスト。Markdown 形式に対応。
priority	integer	マクロリスト内でのマクロ入力フィールドの位置。 priority が設定されていない場合、マクロはグループ化されていないマクロリストの最後に追加されます。
required	integer	マクロを必須としてマークします。 設定可能な値: 0 - 必須ではない; 1 - 必須。 プロパティの動作: - type が”Textbox” または”List” に設定されている場合にサポート
regex	string	テキストボックスフィールドでのユーザー入力を検証するための正規表現。 プロパティの動作: - type が”Textbox” に設定されている場合にサポート
section_name	string	マクロがグループ化される折りたたみセクションのラベル。 section_name が設定されていない場合、マクロはグループ化されません。グループ化されていないマクロが最初に表示され、グループ化されたマクロは各折りたたみセクション内で priority 順に並べられて下に表示されます。
options	text	リスト項目またはチェックボックス値を定義する JSON 文字列。 リストの場合: value と text プロパティを持つオブジェクトの配列。 例: [{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}] チェックボックスの場合: checked と unchecked プロパティを持つオブジェクト。 例: {"checked": true, "unchecked": false} プロパティの動作: - type が”List” または”Checkbox” に設定されている場合は必須

usermacro.create

説明

`object usermacro.create(object/array hostMacros)`

このメソッドは、新しいホストマクロを作成することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成するホストマクロ

このメソッドは標準のホストマクロプロパティ を持つホストマクロを受け入れます。

戻り値

(object) hostmacroids プロパティの下で作成されたホストマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストマクロの順序と一致します。

例

ホストマクロの作成

ホスト”10198” で値”public” を持つホストマクロ”{$SNMP_COMMUNITY}” を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

説明

`object usermacro.createglobal(object/array globalMacros)`

このメソッドは、新しいグローバルマクロを作成することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成するグローバルマクロ

このメソッドは標準のグローバルマクロプロパティ を持つグローバルマクロを受け入れます。

戻り値

(object) globalmacroids プロパティの下で作成されたグローバルマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグローバルマクロの順序と一致します。

例

グローバルマクロの作成

値が”public” のグローバルマクロ”{$SNMP_COMMUNITY}” を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

説明

`object usermacro.delete(array hostMacroIds)`

このメソッドは、ホストマクロを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストマクロの ID。

戻り値

(object) hostmacroids プロパティの下にある削除されたホストマクロの ID を含むオブジェクトを返します。

例

複数のホストマクロの削除

2 つのホストマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

説明

`object usermacro.deleteglobal(array globalMacroIds)`

このメソッドは、グローバルマクロを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグローバルマクロの ID。

戻り値

(object) globalmacroids プロパティの下にある削除されたグローバルマクロの ID を含むオブジェクトを返します。

例

複数のグローバルマクロの削除

2 つのグローバルマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
```

```
    "id": 1
}
```

ソース

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

説明

`integer/array usermacro.get(object parameters)`

このメソッドは、指定されたパラメータに従ってホストおよびグローバルマクロを取得することができます。

> **Note:**
> このメソッドは、すべての種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

`(object)` 望ましい出力を定義するパラメーター。

このメソッドは以下のパラメーターをサポートします。

パラメーター	型	説明
globalmacro	flag	ホストマクロの代わりにグローバルマクロを返します。
globalmacroids	ID/array	指定した ID のグローバルマクロのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのマクロのみを返します。
hostids	ID/array	指定したホストまたはテンプレートに属するマクロのみを返します。
hostmacroids	ID/array	指定した ID のホストマクロのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたホストプロトタイプユーザーマクロのみを返します。
selectHostGroups	query	ホストマクロが属するホストグループを`hostgroups`プロパティで返します。 ホストマクロ取得時のみ使用されます。
selectHosts	query	ホストマクロが属するホストを`hosts`プロパティで返します。 ホストマクロ取得時のみ使用されます。
selectTemplateGroups	query	テンプレートマクロが属するテンプレートグループを`templategroups`プロパティで返します。 テンプレートマクロ取得時のみ使用されます。
selectTemplates	query	ホストマクロが属するテンプレートを`templates`プロパティで返します。 ホストマクロ取得時のみ使用されます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `macro`。
countOutput	boolean	これらのパラメーターはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

パラメーター	型	説明
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストのホストマクロの取得

ホスト"10198" に定義されているすべてのホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "9",
            "hostid": "10198",
            "macro": "{$INTERFACE}",
            "value": "eth0",
            "description": "",
            "type": "0",
            "automatic": "0"
        },
        {
            "hostmacroid": "11",
            "hostid": "10198",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0",
            "automatic": "0"
        }
    ],
    "id": 1
}
```

テンプレートのホストマクロの取得

マクロ名に"STATUS" を含むテンプレート"10265" に定義されたホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
```

```
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "The hostname or IP address of the Apache status page host.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "Apache status host",
                "description": "Hostname or IP address of the Apache status page host.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "The URL path of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Apache status page path",
                "description": "URL path of the Apache status page.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "The port of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Apache status page port",
                "description": "In the range from 1 to 65535 inclusive.",
```

```
                "required": "1",
                "regex": "^-?([0-9]+|([[0-9]+)\\.([0-9]+))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "The request scheme, which may be either HTTP or HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Request scheme",
                "description": "Request scheme, which may be either HTTP or HTTPS.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```


グローバルマクロの取得

すべてのグローバルマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
```

```
    ],
    "id": 1
}
```

ソース

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

説明

`object usermacro.update(object/array hostMacros)`

このメソッドは、既存のホストマクロを更新するためのものです。

> **Note:**
> このメソッドは 管理者および スーパ管理者ユーザータイプでのみ利用可能です。メソッドの呼び出し権限はユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するホストマクロプロパティ

ホストマクロごとに hostmacroid プロパティを定義する必要があります。他のプロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) hostmacroids プロパティの下で更新されたホストマクロの ID を含むオブジェクトを返します。

例

ホストマクロの値の変更

ホストマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

検出ルールによって作成されたマクロの値を変更します。

検出ルールによって作成された” 自動” マクロを” 手動” マクロに変換し、その値を” 新しい値” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
```

```
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

説明

`object usermacro.updateglobal(object/array globalMacros)`

このメソッドは、既存のグローバルマクロを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新するグローバルマクロプロパティ

グローバルマクロごとに globalmacroid プロパティを定義する必要があります。他のプロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) globalmacroids プロパティの下で更新されたグローバルマクロの ID を含むオブジェクトを返します。

例

グローバルマクロの値の変更

グローバルマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
```

```
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

付録 **1.** 参考文献の解説

表記法　データ型

Zabbix API は、以下のデータ型を入力としてサポートしています。

型	説明
ID	エンティティを参照するために使用される一意の識別子。
boolean	ブール値 (`true` または `false`)。
flag	渡された値は、`null` と等しくない場合に `true` とみなされ、そうでない場合は `false` とみなされます。
integer	整数。
float	浮動小数点数。
string	テキスト文字列。
text	より長いテキスト文字列。
timestamp	Unix タイムスタンプ。
array	順序付けられた値のシーケンス (単純な配列)。
object	連想配列。
query	返されるデータを定義する値。(特定のプロパティのみを返す) プロパティ名の配列として定義するか、定義済みの値のいずれかとして定義できます。 `extend` - すべてのオブジェクトプロパティを返します。 `count` - 取得したレコードの数を返します。特定のサブセレクトでのみサポートされます。

Attention:
Zabbix API は常に文字列または配列としてのみ値を返します。

プロパティの動作

一部のオブジェクトプロパティには、その動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - プロパティの値は自動的に設定され、特定の状況下でもユーザーが定義または変更することはできません (例: 継承オブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。
- 書き込み専用 - プロパティの値は設定できますが、作成後にアクセスすることはできません。
- 定数 - プロパティの値はオブジェクトの作成時に設定できますが、作成後は変更できません。
- サポート - プロパティの値の設定は必須ではありませんが、特定の状況下では設定が許可されます (例: `type` が" シンプルチェック"、" 外部チェック"、"SSH エージェント"、"TELNET エージェント"、または"HTTP エージェント" に設定されている場合、サポート)。ただし、サポートのプロパティは、状況に関係なくデフォルト値に設定される場合があることに注意してください。
- 必須 - プロパティの値は、すべての操作 (取得操作を除く) または特定の条件 (例: 作成操作の場合は必須、`operationtype` が" グローバルスクリプト" に設定され、`opcommand_hst` が設定されていない場合は必須) で設定する必要があります。

Note:
更新操作の場合、プロパティは更新操作中に設定されると" 設定済み" とみなされます。

ラベルが付いていないプロパティはオプションです。

パラメーターの動作

一部の操作パラメーターには、操作における動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - パラメーターの値は自動的に設定され、特定の条件下でもユーザーが定義または変更することはできません (例: 継承されたオブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。

- 書き込み専用 - パラメーターの値は設定できますが、設定後はアクセスできません。
- サポート - パラメーターの値の設定は必須ではありませんが、特定の条件下では設定が許可されます (例: Proxy オブジェクトの `operating_mode` が”パッシブプロキシ”に設定されている場合はサポート)。ただし、サポートされたパラメーターは、条件に関わらずデフォルト値に設定される場合があります。
- 必須 - パラメーターの値の設定は必須です。

ラベルが付いていないパラメーターはオプションです。

予約 **ID** 値**”0”**　予約 ID 値”0” は、要素をフィルタリングしたり、参照されているオブジェクトを削除したりするために使用できます。たとえば、ホストから参照されているプロキシを削除するには、proxyid を 0 に設定する必要があります (”proxyid”: ”0”)。また、サーバーオプション proxyids によって監視されるホストをフィルタリングするには、proxyids を 0 に設定する必要があります (”proxyids”: ”0”)。

共通の**”get”** メソッドパラメータ　すべての `get` メソッドでサポートされているパラメータは以下の通りです。

パラメータ	型	説明
countOutput	boolean	実際のデータの代わりに、結果のレコード数を返します。
editable	boolean	`true` に設定すると、ユーザーが書き込み権限を持つオブジェクトのみを返します。 デフォルト: `false`。
excludeSearch	boolean	`search` パラメータで指定した条件に一致しない結果を返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 オブジェクトを受け入れ、キーはプロパティ名 (例: `host.get` ではホストオブジェクトのプロパティ、`item.get` ではアイテムオブジェクトのプロパティなど)、値は一致させる単一値または値の配列です。 `text` データ型のプロパティはサポートされません。 一部のメソッドでは、このパラメータに固有の機能があり、メソッドページで説明されています (例: host.get の `filter` パラメータはホストインターフェースのプロパティもサポートします)。
limit	integer	返されるレコード数を制限します。
output	query	返すオブジェクトのプロパティ。 デフォルト: `extend`。
preservekeys	boolean	結果配列のキーとして ID を使用します。
search	object	指定したパターンに一致する結果を返します (大文字・小文字は区別しません)。 オブジェクトを受け入れ、キーはプロパティ名 (例: `host.get` ではホストオブジェクトのプロパティ、`item.get` ではアイテムオブジェクトのプロパティなど)、値は検索する文字列です。追加オプションが指定されていない場合、`LIKE "%…%"` 検索が実行されます。 `string` および `text` データ型のプロパティのみサポートします。 一部のメソッドでは、このパラメータに固有の機能があり、メソッドページで説明されています (例: host.get の `search` パラメータはホストインターフェースのプロパティもサポートします)。
searchByAny	boolean	`true` に設定すると、`filter` または `search` パラメータで指定したすべての条件ではなく、いずれかの条件に一致する結果を返します。 デフォルト: `false`。
searchWildcardsEnabled	boolean	`true` に設定すると、`search` パラメータでワイルドカード文字として”*” の使用が有効になります。 デフォルト: `false`。
sortfield	string/array	指定したプロパティで結果をソートします。ソートに使用できるプロパティの一覧は、各 API get メソッドの説明を参照してください。ソート前にマクロは展開されません。 値が指定されていない場合、データはソートされずに返されます。

パラメータ	型	説明
sortorder	string/array	ソート順。配列が渡された場合、各値は `sortfield` パラメータで指定した対応するプロパティにマッチします。 指定可能な値: ASC - (デフォルト) 昇順; DESC - 降順。
startSearch	boolean	`search` パラメータでフィールドの先頭を比較し、LIKE "…%" 検索を実行します。 `searchWildcardsEnabled` が `true` の場合は無視されます。

エンティティの起源フラグ　Get メソッドは、低レベルディスカバリ（LLD ルール/LLD ルールプロトタイプ、アイテム/アイテムプロトタイプなど）に関連するエンティティに対して `flags` プロパティを返します。このプロパティは、エンティティがディスカバリされたものかどうかを示すのに便利です。なぜなら、ディスカバリされたエンティティの編集は制限されるためです。

`flags` プロパティは、これらの値の組み合わせ（"+" 演算）に基づいて結果を返します。

値	説明
0	基本エンティティ（アイテム、トリガー、グラフ、ホスト）
1	低レベルディスカバリルール
2	任意のプロトタイプ（アイテムプロトタイプ、トリガープロトタイプ、LLD ルールプロトタイプなど）
4	ディスカバリされたエンティティ（ディスカバリされたアイテム、トリガー、グラフ、ホスト、LLD ルール）

`flags` プロパティによって返される組み合わせ値は次のとおりです。

値	組み合わせ	説明
0	0	通常のエンティティ（アイテム、トリガー、グラフ、ホスト）。
2	2	エンティティプロトタイプ（アイテムプロトタイプ、トリガープロトタイプなど）。
6	2+4	ディスカバリされたアイテム、トリガー、グラフ、ホスト（プロトタイプから変換）。
1	1	低レベルディスカバリルール。
3	1+2	低レベルディスカバリルールプロトタイプ。
5	1+4	ディスカバリされた低レベルディスカバリルール（プロトタイプから変換）。
7	1+2+4	ディスカバリされた低レベルディスカバリルールプロトタイプ。

例　ユーザー権限のチェック

ユーザーは、名前が"MySQL" または"Linux" で始まるホストに書き込む権限を持っていますか？

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
”result” が”0” の場合は、読み取り/書き込み権限を持つホストが存在しないことを意味します。

不一致を数える

名前に部分文字列”ubuntu” が含まれていないホストの数を数えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

ワイルドカードを使用したホストの検索

名前に”server” という単語が含まれ、インターフェースポートが”10050” または”10071” であるホストを検索します。結果をホスト名で降順に並べ替え、5 つに限定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
```

```
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    ],
    "id": 1
}
```

”preservekeys” を含むワイルドカードを使用したホストの検索

前のリクエストにパラメーター”preservekeys” を追加すると、結果はオブジェクトの ID をキーとする連想配列として返されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
```

```
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5,
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "50003": {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50005": {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50004": {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "99032": {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        "99061": {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    },
    "id": 1
}
```

付録 **2. 7.4** から **8.0** への変更点

後方互換性のない変更点

その他の変更点およびバグフィックス　dashboard

ZBXNEXT-9424 新しいダッシュボードウィジェットタイプscatterplot を追加しました。
ZBXNEXT-8160 geomap ウィジェットに新しいダッシュボードウィジェットフィールド clustering_mode と clustering_zoom_level を追加しました。

templatedashboard

ZBXNEXT-9424 新しいテンプレートダッシュボードウィジェットタイプscatterplot を追加しました。

userdirectory

ZBXNEXT-9834 ユーザーディレクトリオブジェクトに idp_certificate、sp_private_key、sp_certificate、idp_certificate_hash sp_private_key_hash、sp_certificate_hash プロパティを追加しました。

付録 **3. 8.0** の変更点

8.0.1　このバージョンはまだリリースされていません。

19 拡張機能

概要　Zabbix は多くの機能を提供していますが、追加機能の余地は常にあります。拡張機能は、Zabbix のソースコードを変更することなく、監視機能を修正・強化する便利な方法です。

Zabbix の機能は、組み込みの拡張オプション（トラッパーアイテム、ユーザーパラメータなど）を使用するか、カスタム拡張機能（ロード可能モジュール、プラグインなど）を使用または作成することで拡張できます。

このセクションでは、Zabbix を拡張するためのすべてのオプションの概要と参照を提供します。

カスタムコマンドによるデータ収集　トラッパーアイテム

トラッパーアイテムは、データを取得する代わりに受信するアイテムです。トラッパーアイテムは、たとえば長時間実行されるユーザースクリプトの場合の定期的な可用性やパフォーマンスデータなど、特定のデータを Zabbix サーバーまたはプロキシに送信するのに便利です。Zabbix サーバーまたはプロキシへのデータ送信は、Zabbix senderユーティリティまたは Zabbix senderプロトコルを使用して可能です。Zabbix サーバーへのデータ送信は、history.pushAPI メソッドを使用しても可能です。

外部チェック

外部チェックは、実行可能ファイル (例えば、シェルスクリプトやバイナリ) を実行してチェックを行うアイテムです。

外部チェックは Zabbix サーバーまたはプロキシ (ホストがプロキシによって監視されている場合) によって実行され、監視対象ホストでエージェントを実行する必要はありません。

ユーザーパラメータ

ユーザーパラメータは、ユーザー定義のキーに関連付けられたユーザー定義のコマンドであり、実行時に Zabbix エージェントが稼働しているホストから必要なデータを取得できます。ユーザーパラメータは、Zabbix で事前定義されていないエージェントまたはエージェント 2 のアイテムを設定する際に便利です。

system.run[] Zabbix エージェントアイテム

system.run[] Zabbix エージェントアイテムは、ユーザー定義コマンド用のアイテムです（事前定義されたキー system.run[] に関連付けられています。例えば、system.run[myscript.sh]）。このコマンドは、Zabbix エージェントが稼働しているホスト上で実行できます。

注意: system.run[] アイテムはデフォルトで無効になっており、使用する場合は Zabbix エージェントまたは agent 2 の設定ファイル（AllowKey 設定パラメータ）で許可し、定義する必要があります。

Attention:
外部チェック、ユーザーパラメーター、system.run[] Zabbix エージェントアイテムなどのアイテムでのユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーから実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が必要です。

HTTP エージェントアイテム

HTTP エージェントアイテムは、HTTP/HTTPS 経由でデータリクエストを実行するためのアイテムです。HTTP エージェントアイテムは、*Elasticsearch* や *OpenWeatherMap* などのサービスからデータを取得するために HTTP エンドポイントにリクエストを送信したり、Zabbix API のステータスや Apache や Nginx Web サーバーのステータスを確認したりするのに便利です。HTTP エージェントアイテム（トラッピングを有効にした場合）は、トラッパーアイテムとしても機能します。

スクリプトアイテム

スクリプトアイテムは、HTTP/HTTPS 経由でデータを取得するユーザー定義の JavaScript コードを実行するためのアイテムです。スクリプトアイテムは、HTTP エージェントアイテムで提供される機能だけでは十分でない場合に便利です。例えば、複数のステップや複雑なロジックを必要とする高度なデータ収集シナリオでは、スクリプトアイテムを設定して HTTP コールを実行し、受信したデータを処理し、変換した値を 2 回目の HTTP コールに渡すことができます。

Note:
HTTP エージェントアイテムとスクリプトアイテムは Zabbix サーバーおよびプロキシでサポートされており、監視対象ホストでエージェントを実行する必要はありません。

高度な拡張機能　ロード可能モジュール

C 言語で記述されたロード可能モジュールは、UNIX プラットフォーム上で Zabbix コンポーネント（サーバー、プロキシ、エージェント）の機能を拡張するための多用途かつパフォーマンス重視のオプションです。ロード可能モジュールは基本的に Zabbix デーモンによって使用され、起動時にロードされる共有ライブラリです。ライブラリには特定の関数を含める必要があり、Zabbix プロセスがそのファイルが実際にロードして動作できるモジュールであることを検出できるようにします。

ロード可能モジュールには、新しいメトリックの追加やその他のロジックの実装（例：Zabbix の履歴データエクスポート）などの機能拡張、優れたパフォーマンス、提供される機能の開発・利用・共有のオプションなど、多くの利点があります。これにより、メンテナンスが容易になり、Zabbix のコアコードベースとは独立して新機能を簡単に提供できるようになります。

ロード可能モジュールは、複雑な監視構成で特に有用です。組み込みシステムの監視、大量の監視パラメータ、複雑なロジックや起動時間の長い重いスクリプトを使用する場合、ユーザーパラメータ、system.run[] Zabbix エージェントアイテム、外部チェックなどの拡張機能はパフォーマンスに影響を与えます。ロード可能モジュールは、パフォーマンスを犠牲にすることなく Zabbix の機能を拡張する方法を提供します。

プラグイン

プラグインは、Zabbix agent 2 の監視機能を拡張します。Go プログラミング言語で記述されており、(C で記述された) ロード可能なモジュールの代替手段を提供します。

プラグインは、構造を定義し、1 つまたは複数のプラグインインターフェース (*Exporter*、*Collector*、*Configurator*、*Runner*、*Watcher*) を実装する Go パッケージです。Zabbix agent 2 では、2 種類のプラグインがサポートされています:

- 組み込みプラグイン (Zabbix 4.4.0 以降でサポート)
- ロード可能プラグイン (Zabbix 6.0.0 以降でサポート)

Zabbix が提供するプラグインの一覧を参照してください。

独自のプラグインを作成するための手順やチュートリアルについては、Developer centerを参照してください。

アラートのカスタマイズ　Webhook

Webhookは、Zabbix のメディアタイプであり、Zabbix のアラート機能をヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアに拡張するオプションを提供します。スクリプトアイテムと同様に、Webhook はカスタム JavaScript コードを使用して HTTP コールを行うのに便利であり、たとえば Microsoft Teams、Discord、Jira などのさまざまなプラットフォームに通知をプッシュすることができます。また、作成されたヘルプデスクチケットに関する情報などのデータを返し、それを Zabbix に表示することも可能です。

既存の Webhook は、Zabbix のGit リポジトリで利用できます。カスタム Webhook の開発については、Webhook 開発ガイドラインを参照してください。

アラートスクリプト

アラートスクリプトは、Zabbix のメディアタイプの一つで、Zabbix アラートを処理するための独自の方法（スクリプト）を作成するオプションを提供します。アラートスクリプトは、Zabbix でアラートを送信する既存のメディアタイプに満足できない場合に便利です。

フロントエンドのカスタマイズ　カスタムテーマ

カスタムテーマを使用して、Zabbix フロントエンドの外観を変更することができます。独自のテーマの作成と適用方法については、手順を参照してください。

フロントエンドモジュール

フロントエンドモジュールは、サードパーティ製モジュールの追加や独自開発により、Zabbix フロントエンドの機能を拡張するオプションを提供します。フロントエンドモジュールを使用すると、新しいメニュー項目やそれに対応するビュー、アクションなどを追加できます。

グローバルスクリプト　グローバルスクリプトは、設定されたスコープとユーザー権限に応じて、監視対象で実行できるユーザー定義のコマンドセットです（シェル（/bin/sh）インタープリターによる）。グローバルスクリプトは、以下のアクションに設定できます。

- アクションオペレーション
- 手動ホストアクション
- 手動イベントアクション

グローバルスクリプトは多くのケースで有用です。例えば、アクションオペレーションや手動ホストアクションに設定した場合、グローバルスクリプトを使用して、アプリケーション（Web サーバー、ミドルウェア、CRM など）の再起動やディスクスペースの解放（古いファイルの削除、`/tmp` のクリーンアップなど）などのリモートコマンドを自動または手動で実行できます。また、手動イベントアクションに設定した場合は、外部システムでの問題チケットの管理などにも利用できます。

グローバルスクリプトは、Zabbix サーバー、プロキシ、エージェントによって実行できます。

> **Attention:**
> ユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーから実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が必要です。

Zabbix API　Zabbix APIは、Zabbix フロントエンドの一部である HTTP ベースの API です。Zabbix API を使用すると、以下の操作をプログラムで実行できます。

- Zabbix の設定をプログラムで取得・変更する。
- Zabbix の設定をインポートおよびエクスポートする。
- Zabbix の履歴データやトレンドデータにアクセスする。
- アプリケーションを Zabbix と連携するように設定する。
- Zabbix をサードパーティ製ソフトウェアと統合する。
- 定型作業を自動化する。

Zabbix API は、多数のメソッドで構成されており、名目上は個別の API にグループ化されています。各メソッドは特定のタスクを実行します。利用可能なメソッドや、Zabbix API が提供する機能の概要については、Zabbix API のメソッドリファレンスを参照してください。

1 ローダブルモジュール

概要

ロード可能なモジュールは、Zabbix の機能を拡張するためのパフォーマンス重視のオプションを提供します。

Zabbix の機能は、ユーザーパラメータ、外部チェック、system.run Zabbix エージェントアイテムなど、さまざまな方法で拡張できます。これらは非常によく機能しますが、主な欠点は fork() です。Zabbix はユーザーメトリックを処理するたびに新しいプロセスを fork する必要があり、これはパフォーマンスにとって良くありません。通常は大きな問題ではありませんが、組み込みシステムの監視や、多数の監視パラメータ、複雑なロジックや起動時間の長い重いスクリプトを使用する場合には深刻な問題となる可能性があります。

ロード可能なモジュールのサポートにより、パフォーマンスを犠牲にすることなく、Zabbix エージェント、サーバー、プロキシの拡張が可能になります。

ロード可能なモジュールは基本的に Zabbix デーモンによって使用され、起動時にロードされる共有ライブラリです。このライブラリには特定の関数が含まれている必要があり、Zabbix プロセスがそのファイルが実際にロードして動作できるモジュールであることを検出できるようになっています。

ロード可能なモジュールには多くの利点があります。優れたパフォーマンスや任意のロジックを実装できる能力も非常に重要ですが、おそらく最も重要な利点は、Zabbix モジュールを開発、利用、共有できることです。これにより、トラブルのないメンテナンスが可能となり、Zabbix のコアコードベースとは独立して新しい機能をより簡単に提供できるようになります。

モジュールのライセンスおよびバイナリ形式での配布は AGPL-3.0 ライセンスによって規定されています（モジュールは実行時に Zabbix とリンクし、Zabbix ヘッダーを使用します。Zabbix の全コードは Zabbix 7.0 以降 AGPL-3.0 ライセンスで提供されています）。バイナリ互換性は Zabbix によって保証されません。

モジュール API の安定性は、1 つの Zabbix LTS（長期サポート）リリースサイクルの間保証されます。Zabbix API の安定性は保証されません（技術的にはモジュールから Zabbix の内部関数を呼び出すことは可能ですが、そのようなモジュールが動作する保証はありません）。

モジュール API

共有ライブラリを Zabbix モジュールとして扱うためには、いくつかの関数を実装してエクスポートする必要があります。現在、Zabbix モジュール API には 6 つの関数があり、そのうち 1 つだけが必須で、他の 5 つはオプションです。

必須インターフェース

唯一必須の関数は **zbx_module_api_version()** です:

```
int zbx_module_api_version(void);
```

この関数は、このモジュールで実装されている API バージョンを返す必要があり、モジュールがロードされるためには、このバージョンが Zabbix でサポートされているモジュール API バージョンと一致している必要があります。Zabbix でサポートされているモジュール API のバージョンは ZBX_MODULE_API_VERSION です。したがって、この関数はこの定数を返す必要があります。この目的で使用されていた古い定数 ZBX_MODULE_API_VERSION_ONE は、ソース互換性を維持するために ZBX_MODULE_API_VERSION と等しく定義されていますが、使用は推奨されません。

オプションインターフェース

オプションの関数は **zbx_module_init()**、**zbx_module_item_list()**、**zbx_module_item_timeout()**、**zbx_module_history_write_cbs()**、**zbx_module_uninit()** です:

```
int zbx_module_init(void);
```

この関数は、モジュールに必要な初期化処理 (必要な場合) を実行する必要があります。成功した場合は ZBX_MODULE_OK を返す必要があります。失敗した場合は ZBX_MODULE_FAIL を返す必要があります。後者の場合、Zabbix は起動しません。

```
ZBX_METRIC  *zbx_module_item_list(void);
```

この関数は、モジュールがサポートするアイテムのリストを返す必要があります。各アイテムは ZBX_METRIC 構造体で定義されます。詳細は下記のセクションを参照してください。リストは、"key" フィールドが NULL の ZBX_METRIC 構造体で終了します。

```
void    zbx_module_item_timeout(int timeout);
```

モジュールが **zbx_module_item_list()** をエクスポートしている場合、この関数は Zabbix によって、モジュールが実装するアイテムチェックが従うべき Zabbix 設定ファイルのタイムアウト設定を指定するために使用されます。ここで、"timeout" パラメータは秒単位です。

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

この関数は、Zabbix サーバーがさまざまなデータ型の履歴をエクスポートするために使用するコールバック関数を返す必要があります。コールバック関数は ZBX_HISTORY_WRITE_CBS 構造体のフィールドとして提供され、モジュールが特定の型の履歴に関心がない場合はフィールドを NULL にできます。

```
int zbx_module_uninit(void);
```

この関数は、割り当てられたリソースの解放やファイルディスクリプタのクローズなど、必要な後処理 (必要な場合) を実行する必要があります。

すべての関数は、モジュールがロードされたときに Zabbix の起動時に 1 回呼び出されます。ただし、zbx_module_uninit() のみは、モジュールがアンロードされたときに Zabbix のシャットダウン時に 1 回呼び出されます。

アイテムの定義

各アイテムは ZBX_METRIC 構造体で定義されます:

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
```

```
}
ZBX_METRIC;
```

ここで、**key** はアイテムキー（例：”dummy.random”）、**flags** は CF_HAVEPARAMS または 0（アイテムがパラメータを受け入れるかどうかによる）、**function** はアイテムを実装する C 関数（例：”zbx_module_dummy_random”）、**test_param** は Zabbix エージェントが”-p” フラグで起動されたときに使用するパラメータリスト（例：”1,1000”、NULL も可）です。定義例は次のようになります:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

アイテムを実装する各関数は、最初のパラメータが AGENT_REQUEST 型、2 番目のパラメータが AGENT_RESULT 型の 2 つのポインタパラメータを受け取る必要があります:

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

これらの関数は、アイテム値が正常に取得できた場合は SYSINFO_RET_OK を返す必要があります。そうでない場合は SYSINFO_RET_FAIL を返す必要があります。AGENT_REQUEST から情報を取得する方法や AGENT_RESULT に情報を設定する方法の詳細については、以下の”dummy” モジュールの例を参照してください。

履歴エクスポートコールバックの提供

Attention:
モジュールによる履歴エクスポートは、Zabbix プロキシではサポートされなくなりました。

モジュールは、型ごとに履歴データをエクスポートする関数を指定できます: 数値 (float)、数値 (unsigned)、文字列、テキスト、ログ:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

それぞれの関数は、”history_num” 個の要素からなる”history” 配列を引数として受け取る必要があります。エクスポートする履歴データの型に応じて、”history” はそれぞれ以下の構造体の配列となります:

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    double      value;
}
ZBX_HISTORY_FLOAT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;
```

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
    const char  *source;
    int     timestamp;
    int     logeventid;
    int     severity;
}
ZBX_HISTORY_LOG;
```

コールバックは、Zabbix サーバーの履歴同期プロセスによって、履歴同期処理の最後に、データが Zabbix データベースに書き込まれ、値キャッシュに保存された後に使用されます。

Attention:
履歴エクスポートモジュールで内部エラーが発生した場合、モジュールは全体の監視をブロックせず、データを破棄して Zabbix サーバーの動作を継続できるように記述することを推奨します。

モジュールのビルド

モジュールは現在、Zabbix のソースツリー内でビルドすることを想定しています。なぜなら、モジュール API が Zabbix ヘッダで定義されているいくつかのデータ構造に依存しているためです。

ロード可能なモジュールにとって最も重要なヘッダは **include/module.h** であり、これがこれらのデータ構造を定義しています。**include/module.h** が正しく動作するために必要な他のシステムヘッダは **stdlib.h** と **stdint.h** です。

この情報を踏まえれば、モジュールをビルドする準備は整っています。モジュールは **stdlib.h**、**stdint.h**、**module.h** をインクルードし、ビルドスクリプトはこれらのファイルがインクルードパスにあることを確認する必要があります。詳細については、下記の「ダミー」モジュールの例を参照してください。

もう 1 つ便利なヘッダは **include/zbxcommon.h** で、これはログ記録やデバッグ目的で使用できる **zabbix_log()** 関数を定義しています。

設定パラメータ

Zabbix エージェント、サーバー、プロキシは、モジュールを扱うために 2 つのパラメータをサポートしています。

- LoadModulePath – ロード可能なモジュールの場所へのフルパス
- LoadModule – 起動時にロードするモジュール。モジュールは LoadModulePath で指定されたディレクトリに配置するか、パスをモジュール名の前に付ける必要があります。前置されたパスが絶対パス（'/' で始まる）である場合、LoadModulePath は無視されます。複数の LoadModule パラメータを含めることができます。

例えば、Zabbix エージェントを拡張するために、以下のパラメータを追加できます。

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
```

```
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

エージェントの起動時に、/usr/local/lib/zabbix/agent ディレクトリから mariadb.so、apache.so、kernel.so モジュールがロードされ、dummy.so は/usr/local/lib/zabbix からロードされます。モジュールが存在しない場合、不適切なパーミッションの場合、または共有ライブラリが Zabbix モジュールでない場合、エージェントは起動に失敗します。

フロントエンドの設定

Zabbix エージェント、サーバー、プロキシはロード可能なモジュールをサポートしています。したがって、Zabbix フロントエンドでのアイテムタイプは、モジュールがどこにロードされているかによって異なります。モジュールがエージェントにロードされている場合、アイテムタイプは「Zabbix エージェント」または「Zabbix エージェント（アクティブ)」である必要があります。モジュールがサーバーまたはプロキシにロードされている場合、アイテムタイプは「シンプルチェック」である必要があります。

Zabbix モジュールによる履歴エクスポートには、フロントエンドの設定は必要ありません。サーバーがモジュールのロードに成功し、**zbx_module_history_write_cbs()** 関数が少なくとも 1 つの NULL 以外のコールバック関数を返す場合、履歴エクスポートは自動的に有効になります。

ダミーモジュール

Zabbix には C 言語で書かれたサンプルモジュールが含まれています。モジュールは src/modules/dummy にあります。

```
alex@alex:~trunk/src/modules/dummy$ ls -l
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c
-rw-rw-r-- 1 alex alex   67 Apr 24 17:54 Makefile
-rw-rw-r-- 1 alex alex  245 Apr 24 17:54 README
```

このモジュールは十分にドキュメント化されており、独自のモジュールのテンプレートとして使用できます。

上記のように Zabbix ソースツリーのルートで./configure を実行した後、**make** を実行するだけで **dummy.so** がビルドされます。

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
```

```
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;
```

```
    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
```

```
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
```

```c
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
```

```
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

このモジュールは 3 つの新しいアイテムをエクスポートします。

- dummy.ping - 常に'1' を返します
- dummy.echo[param1] - 最初のパラメータをそのまま返します。例えば、dummy.echo[ABC] は ABC を返します
- dummy.random[param1, param2] - param1-param2 の範囲内のランダムな数値を返します。例えば、dummy.random[1,1000000]

制限事項

ロード可能なモジュールのサポートは Unix プラットフォームのみで実装されています。つまり、Windows エージェントでは動作しません。

場合によっては、モジュールが *zabbix_agentd.conf* からモジュール関連の設定パラメータを読み取る必要があるかもしれません。これは現在サポートされていません。モジュールでいくつかの設定パラメータを使用する必要がある場合は、モジュール固有の設定ファイルの解析を実装する必要があるでしょう。

2 プラグイン

概要

プラグインは、Zabbix agent 2 の監視機能を拡張します。Go プログラミング言語で記述されており、ロード可能モジュール（C で記述）や、Zabbix の機能を拡張するためのその他の方法の代替手段を提供します。

これらの機能は、agent 2 およびそのプラグインに固有のものです。

- パッシブおよびアクティブチェックの両方に対するスケジュールされた柔軟な間隔
- スケジュールとタスクの同時実行性を考慮したタスクキュー管理
- プラグインレベルのタイムアウト
- 起動時の Zabbix agent 2 とそのプラグインの互換性チェック

Zabbix agent 2 がサポートするすべてのメトリクスは、プラグインによって収集されます。このページでは、ユーザーの視点からプラグインの設定原則について説明し、Zabbix が提供する組み込みおよびロード可能なプラグインの両方を一覧表示します。

> **Note:**
> Zabbix 6.0 以降、プラグインは agent 2 に組み込む必要がなくなりました。ロード可能なプラグインとして追加できるため、新しいメトリクス用のプラグイン作成が簡単になります。独自のプラグイン作成手順については、Developer centerおよびZabbix agent 2 用のサンプルプラグインを参照してください。

プラグインの設定

このセクションでは、一般的なプラグインの設定原則とベストプラクティスについて説明します。

すべてのプラグインは *Plugins.** パラメータを使用して設定されます。これは、Zabbix agent 2 の設定ファイルの一部である場合と、プラグイン独自の設定ファイルである場合があります。プラグインが個別の設定ファイルを使用する場合は、このファイルへのパスを Zabbix agent 2 の設定ファイルの Include パラメータで指定する必要があります。

一般的なプラグインパラメータは、以下の構造を持ちます。

Plugins.<PluginName>.<Parameter>=<Value>

さらに、2 つの特定のパラメータグループがあります。

- デフォルトパラメータ値を定義するための *Plugins.<PluginName>.Default.<Parameter>=<Value>* 。
- 名前付きセッションを介して異なる監視対象ごとに個別のパラメータセットを定義するための *Plugins.<PluginName>.<SessionName>.<*
  。

すべてのパラメータ名は、以下の要件に従う必要があります。

- プラグイン名は大文字にすることを推奨します
- パラメータは大文字にする必要があります
- 特殊文字は使用できません
- ネストの最大レベルに制限はありません
- パラメータ数に制限はありません

たとえば、アクティブチェックで、*Scheduling* 更新間隔を持たないものを、エージェントの再起動直後に Uptime プラグインのみに対して実行するには、設定ファイルで `Plugins.Uptime.System.ForceActiveChecksOnStart=1` を設定します。同様に、CPU プラグインの同時チェック数のカスタム上限を設定するには、設定ファイルで `Plugins.CPU.System.Capacity=N` を設定します。

デフォルト値

接続関連のパラメータ（URI、ユーザー名、パスワードなど）のデフォルト値は、設定ファイルで以下の形式で設定できます。

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

例えば、`Plugins.Mysql.Default.Username=zabbix`、`Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017` などです。

このようなパラメータの値がアイテムキーや名前付きセッションパラメータで指定されていない場合、プラグインはデフォルト値を使用します。デフォルトパラメータも未定義の場合は、ハードコードされたデフォルト値が使用されます。

> **Note:**
> アイテムキーにパラメータが指定されていない場合、Zabbix agent 2 はデフォルトパラメータセクションで定義された値を使用してメトリックの収集を試みます。

名前付きセッション

名前付きセッションは、プラグインパラメータの追加レベルを表し、監視対象の各インスタンスごとに個別の認証パラメータセットを指定するために使用できます。各名前付きセッションパラメータは、以下の構造を持つ必要があります。

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

セッション名は、URI、ユーザー名、および/またはパスワードを個別に指定する代わりに、connString アイテムキーのパラメータとして使用できます。

アイテムキーでは、最初のパラメータは connString または URI のいずれかです。最初のキーのパラメータがセッション名と一致しない場合、それは URI として扱われます。アイテムキーで埋め込み URI 認証情報を渡すことはサポートされていないため、代わりに名前付きセッションパラメータを使用してください。

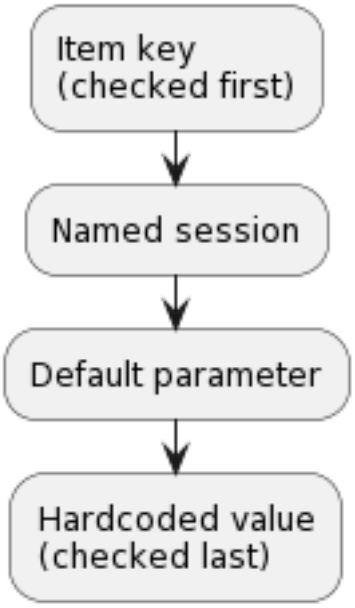
利用可能な名前付きセッションパラメータのリストは、プラグインによって異なります。

アイテムキーのパラメータで新しい値を指定することで、セッションパラメータを上書きすることができます（例を参照）。

名前付きセッションにパラメータが定義されていない場合、Zabbix agent 2 はデフォルトのプラグインパラメータで定義された値を使用します。

パラメータの優先順位

Zabbix agent 2 プラグインは、接続に関連するパラメータ値を以下の順序で検索します:



1. 最初のアイテムキーのパラメータがセッション名と比較されます。一致が見つからない場合は、実際の値として扱われます。この場合、ステップ 3 はスキップされます。一致が見つかった場合、パラメータ値 (通常は URI) は名前付きセッションで定義されている必要があります。
2. 他のパラメータは、定義されていればアイテムキーから取得されます。
3. アイテムキーのパラメータ (例: password) が空の場合、プラグインは対応する名前付きセッションパラメータを探します。
4. セッションパラメータも指定されていない場合は、対応するデフォルトパラメータで定義された値が使用されます。
5. それでも値が見つからない場合、プラグインはハードコードされたデフォルト値を使用します。

例 1

2 つのインスタンス “MySQL1” と “MySQL2” の監視。

設定パラメータ:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

この設定の結果､各セッション名はitem キーの connString として使用できます｡例: `mysql.ping[MySQL1]` や `mysql.ping[MySQL2]`。

例 2

アイテムキーで一部のパラメータを指定する。

設定パラメータ:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

アイテムキー: `pgsql.ping[session1,new_username,,postgres]`

この設定の結果、エージェントは以下のパラメータで PostgreSQL に接続します:

- セッションパラメータからの URI: `192.0.2.234:5432`
- アイテムキーからのユーザー名: `new_username`
- セッションパラメータからのパスワード (アイテムキーで省略されているため): `session_password`
- アイテムキーからのデータベース名: `postgres`

例 3

デフォルトの構成パラメータを使用してメトリックを収集します。

構成パラメータ:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

アイテムキー: `pgsql.ping[,,,postgres]`

この構成の結果、エージェントは以下のパラメータを使用して PostgreSQL に接続します:

- デフォルト URI: `192.0.2.234:5432`
- デフォルトユーザー名: `zabbix`
- デフォルトパスワード: `password`
- アイテムキーから取得したデータベース名: `postgres`

接続

一部のプラグインは、複数のインスタンスから同時にメトリクスを収集することをサポートしています。ローカルおよびリモートのインスタンスの両方を監視できます。TCP および Unix ソケット接続がサポートされています。

プラグインを設定して、インスタンスへの接続をオープン状態に保つことを推奨します。その利点は、接続数が少ないため、ネットワークの混雑、遅延、CPU およびメモリの使用量が削減されることです。クライアントライブラリがこれを処理します。

Note:
未使用の接続をオープン状態にしておく期間は、*Plugins.<PluginName>.KeepAlive* パラメータで決定できます。例: *Plugins.Memcached.KeepAlive*

組み込みプラグイン

Zabbix agent 2 の組み込みプラグインは、すぐに利用できます。プラグイン名をクリックすると、追加情報が記載されたプラグインリポジトリに移動します。

プラグイン名	説明	サポートされているアイテムキー	コメント
Agent	使用中の Zabbix エージェントのメトリクス。	agent.hostname, agent.ping, agent.version	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
CPU	システム CPU の監視 (CPU/CPU コア数、検出された CPU、使用率)。	system.cpu.discovery, system.cpu.num, system.cpu.util	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Docker	Docker コンテナの監視。	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	詳細は以下も参照: 設定パラメータ
File	ファイルメトリクスの収集。	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Kernel	カーネルの監視。	kernel.maxfiles, kernel.maxproc	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Log	ログファイルの監視。	log, log.count, logrt, logrt.count	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。 詳細は以下も参照: プラグイン設定パラメータ (Unix/Windows)
Memcached	Memcached サーバーの監視。	memcached.ping, memcached.stats	

プラグイン名	説明	サポートされているアイテムキー	コメント
Modbus	Modbus データの読み取り。	modbus.get	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
MQTT	MQTT トピックの公開値の受信。	mqtt.get	MQTT ブローカーへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定します。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。
MySQL	MySQL およびそのフォークの監視。	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定します。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。
NetIf	ネットワークインターフェースの監視。	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Oracle	Oracle Database の監視。	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	プラグインを使用する前にOracle Instant Clientをインストールしてください。
Proc	プロセスの CPU 使用率。	proc.cpu.util	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
Redis	Redis サーバーの監視。	redis.config, redis.info, redis.ping, redis.slowlog.count	Redis への暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定します。TLS パラメータはアイテムキーのパラメータとして渡すことはできません。TLS の設定が誤っている、または無効な場合、Zabbix agent 2 の起動ができなくなることがあるため、証明書ファイル、権限、パスを有効化前に確認してください。

プラグイン名	説明	サポートされているアイテムキー	コメント

Smart	S.M.A.R.T. 監視。	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	必要な smartctl の最小バージョンは 7.1 です。
			Zabbix agent 2 を実行するユーザーには、smartctl への sudo/root 権限が必要です。プラグインは以下のコマンドのみを使用します:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			サポートされているkeysは、Linux/Windows上の Zabbix agent 2 でのみ、パッシブ・アクティブチェックの両方で使用できます。
			詳細は以下も参照:
			設定パラメータ

SW	インストール済みパッケージの一覧取得。	system.sw.packages, system.sw.packages.get	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
Swap	スワップ領域のサイズ (バイト/パーセンテージ)。	system.swap.size	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
SystemRun	指定したコマンドの実行。	system.run	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。 詳細は以下も参照: プラグイン設定パラメータ (Unix/Windows)
Systemd	systemd サービスの監視。	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP 接続の可用性チェック。	net.tcp.port	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
UDP	UDP サービスの可用性とパフォーマンスの監視。	net.udp.service, net.udp.service.perf	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Uname	システム情報の取得。	system.hostname, system.sw.arch, system.uname	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Uptime	システム稼働時間メトリクスの収集。	system.uptime	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
VFSDev	VFS メトリクスの収集。	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
WebCertificate	TLS/SSL ウェブサイト証明書の監視。	web.certificate.get	
WebPage	ウェブページの監視。	web.page.get, web.page.perf, web.page.regexp	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
ZabbixAsync	非同期メトリクスの収集。	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。

プラグイン名	説明	サポートされているアイテムキー	コメント
ZabbixStats	Zabbix サーバー/プロキシの内部メトリクスまたはキュー内の遅延アイテム数。	zabbix.stats	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
ZabbixSync	同期メトリクスの収集。	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。

ロード可能なプラグイン

Zabbix agent 2 のロード可能なプラグインは、デフォルトでは利用できず、別途インストールが必要です。

- Linux では、パッケージを利用できます(例: Ubuntu で Ember+ をインストールするには `apt install zabbix-agent2-plugin-embe` を実行)、またはプラグインをビルドします
- Windows では、MSI からインストールするか、プラグインをビルドします

Attention:
プラグインをインストールする前に、README ファイルを確認してください。特定の要件やインストール手順が記載されている場合があります。

プラグイン名をクリックすると、追加情報を含むプラグインの README ファイルがあるプラグインリポジトリに移動します。

プラグイン名	説明	サポートされているアイテムキー	コメント
Ceph	Ceph の監視。	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Ceph プラグインの設定パラメータも参照してください。
Ember+	Ember+ の監視。	ember.get	Ember+ プラグインの設定パラメータも参照してください。

プラグイン名	説明	サポートされているアイテムキー	コメント
MongoDB	MongoDB サーバーおよびクラスター（ドキュメントベースの分散データベース）の監視。	mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションパラメータとして指定します。 現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 MongoDB プラグインの設定パラメータも参照してください。
MSSQL	MSSQL データベースの監視。	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定します。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 MSSQL プラグインの設定パラメータも参照してください。
NVIDIA GPU	NVIDIA GPU の監視。	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	NVIDIA GPU プラグインの設定パラメータも参照してください。

プラグイン名	説明	サポートされているアイテムキー	コメント
PostgreSQL	PostgreSQL およびそのフォークの監視。	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定します。 現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 PostgreSQL プラグインの設定パラメータも参照してください。

Note:
ロード可能なプラグインは、以下のオプションで起動できます:
 - `-V --version` - プラグインのバージョンとライセンス情報を表示;
 - `-h --help` - ヘルプ情報を表示;
 - `-t, --test <item key>` — テスト用にプラグインを起動（プラグイン設定は無視されます）。

3 フロントエンドモジュール

概要

Zabbix のフロントエンド機能は、サードパーティ製のモジュールを追加したり、独自のモジュールを開発したりすることで拡張できます。Zabbix のソースコードを変更する必要はありません。

モジュールのコードは Zabbix のソースコードと同じ権限で実行されることに注意してください。これは以下のことを意味します:

- サードパーティ製のモジュールは有害な場合があります。インストールするモジュールを信頼できるものにしてください。
- サードパーティ製モジュールのコードにエラーがあると、フロントエンドがクラッシュする可能性があります。その場合は、フロントエンドからモジュールのコードを削除してください。Zabbix フロントエンドをリロードすると、いくつかのモジュールが存在しないというメッセージが表示されます。モジュール管理（管理 → 一般 → モジュール）に移動し、ディレクトリのスキャンを再度クリックして、データベースから存在しないモジュールを削除してください。

インストール

特定のモジュールのインストールマニュアルを必ずお読みください。新しいモジュールは 1 つずつインストールすることを推奨します。そうすることで、失敗を簡単に特定できます。

モジュールをインストールする直前に:

- 信頼できるソースからモジュールをダウンロードしたことを確認してください。有害なコードをインストールすると、データ損失などの結果を招く可能性があります
- 同じモジュール（同じ ID）の異なるバージョンを並行してインストールできますが、有効化できるのは 1 つのバージョンのみです

モジュールをインストールする手順:

- Zabbix フロントエンドの `modules` フォルダ内に、モジュール専用のフォルダとしてモジュールを展開します
- モジュールフォルダに少なくとも manifest.json ファイルが含まれていることを確認します
- モジュール管理に移動し、ディレクトリのスキャンボタンをクリックします
- 新しいモジュールが、バージョン、作成者、説明、ステータスとともにリストに表示されます
- ステータスをクリックしてモジュールを有効化します

トラブルシューティング:

問題	解決策
モジュールがリストに表示されない	モジュールがリストに表示されない場合は、modules/your-module/に manifest.json が存在することを確認してください。存在しない場合は、モジュールを誤ったディレクトリに展開した可能性があります。存在する場合は、モジュールが Zabbix のバージョンと互換性がない可能性があります。また、Web サーバーユーザーがモジュールフォルダおよびすべてのサブディレクトリに対して少なくとも読み取り・検索権限（`r-x`）、内部のすべてのファイルに対して読み取り権限（`r--`）を持っていることを確認してください。
フロントエンドがクラッシュした	モジュールのコードが現在の Zabbix バージョンまたはサーバー構成と互換性がありません。モジュールファイルを削除し、フロントエンドをリロードしてください。いくつかのモジュールが存在しないという通知が表示されます。モジュール管理に移動し、ディレクトリのスキャンを再度クリックして、存在しないモジュールをデータベースから削除してください。
同一の名前空間、*ID*、アクションに関するエラーメッセージが表示される	新しいモジュールが、他の有効なモジュールですでに登録されている名前空間、ID、アクションを登録しようとしました。新しいモジュールを有効化する前に、(エラーメッセージで示された）競合するモジュールを無効化してください。
技術的なエラーメッセージが表示される	モジュールの開発者にエラーを報告してください。

モジュールの開発

カスタムモジュールの開発の詳細については、開発者センターを参照してください。

20 付録

サイドバーから付録セクションの内容にアクセスしてください。

1 インストールとセットアップ

サイドバーから、参照したい項目を選択してください。

1 データベースの作成

概要

Zabbix サーバーまたはプロキシのインストール時に、Zabbix データベースを作成する必要があります。

このセクションでは、Zabbix データベースの作成手順を説明します。サポートされている各データベースごとに、個別の手順が用意されています。

Note:
最小限の権限を持つデータベースロール/ユーザーを作成してデータベースのセキュリティを向上させるには、各サポートデータベースのデータベース作成ベストプラクティスを参照してください:

- MySQL/MariaDB
- PostgreSQL/TimescaleDB

安全な TLS 接続の設定については、データベースへの安全な接続を参照してください。

Zabbix がサポートする唯一のエンコーディングは UTF-8 です。これは、セキュリティ上の欠陥なく動作することが知られています。他のエンコーディングを使用する場合、既知のセキュリティ問題があることに注意してください。UTF-8 への切り替えについては、

Zabbix データベースの文字セットと照合順序の修復を参照してください。また、utf8mb4 照合順序によるフィルタリングの制限も参照してください。

> **Note:**
> Zabbix Git リポジトリからインストールする場合は、次の手順に進む前に以下のコマンドを実行する必要があります。
>

 `make dbschema`

MySQL/MariaDB

Zabbix サーバー/プロキシが MySQL データベースで正しく動作するためには、utf8（別名 utf8mb3）および utf8mb4 の文字セット（それぞれ utf8_bin および utf8mb4_bin の照合順序）がサポートされています。新規インストールには utf8mb4 の使用を推奨します。

スキーマのインポート時に決定的なトリガーを作成する必要があります。MySQL および MariaDB では、バイナリログが有効になっていて、スーパーユーザー権限がなく、MySQL の設定ファイルで log_bin_trust_function_creators = 1 が設定されていない場合、`GLOBAL log_bin_trust_function_creators = 1` を設定する必要があります。

Zabbix をパッケージからインストールする場合は、ご利用のプラットフォームの手順に従ってください。

Zabbix をソースからインストールする場合:

- データベースとユーザーを作成し、設定します。

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- データベースにデータをインポートし、デフォルトの文字セットとして utf8mb4 を設定します。Zabbix プロキシのデータベースの場合は、`schema.sql` のみをインポートしてください（images.sql や data.sql は不要です）。

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### Zabbix プロキシ用のデータベースを作成する場合はここで終了
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

スキーマのインポートが正常に完了した後は、`log_bin_trust_function_creators` を無効にできます。

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

データベースオブジェクトを作成する権限を持つデータベースユーザーが必要です。

Zabbix パッケージからインストールする場合は、ご利用のプラットフォームの手順に従ってください。

Zabbix をソースからインストールする場合:

- データベースユーザーを作成します。

次のシェルコマンドは、ユーザー `zabbix` を作成します。プロンプトが表示されたらパスワードを指定し、パスワードを再入力してください（最初に `sudo` のパスワードを求められる場合があります）。

```
sudo -u postgres createuser --pwprompt zabbix
```

- データベースを作成します。

次のシェルコマンドは、先ほど作成したユーザーを所有者（`-O zabbix`）として、データベース `zabbix`（最後のパラメータ）を作成します。

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- 初期スキーマとデータをインポートします（Zabbix ソースのルートディレクトリにいることを前提としています）。Zabbix プロキシデータベースの場合は、`schema.sql` のみをインポートしてください（images.sql や data.sql は不要です）。

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### Zabbix プロキシ用のデータベースを作成する場合はここで終了
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
上記のコマンドは、ほとんどの GNU/Linux インストールで動作する例として提供されています。システムやデータベースの設定によっては、異なるコマンドを使用することもできます。例:

 `psql -U <username>`

 データベースのセットアップに問題がある場合は、データベース管理者にご相談ください。

TimescaleDB

TimescaleDB の作成と設定の手順は、別のセクションに記載されています。

SQLite

Zabbix プロキシでのみ SQLite の使用がサポートされています！

データベースは存在しない場合、自動的に作成されます。

インストールセクションに戻る。

2 Zabbix データベースのキャラクターセットと照合の修正

MySQL/MariaDB

従来、MySQL およびその派生製品では、'utf8' は utf8mb3 のエイリアスとして使用されていました。utf8mb3 は MySQL 独自の 3 バイト実装であり、標準の UTF8（4 バイト）とは異なります。MySQL 8.0.28 および MariaDB 10.6.1 以降では、'utf8mb3' 文字セットは非推奨となり、将来的にはサポートが削除され、'utf8' は'utf8mb4' を参照するようになります。Zabbix 6.0 以降では、'utf8mb4' がサポートされています。将来的な問題を回避するためにも、'utf8mb4' の使用を強く推奨します。'utf8mb4' に切り替えるもう一つの利点は、補助 Unicode 文字のサポートです。

Warning:
Zabbix 6.0 以前のバージョンは utf8mb4 を認識しないため、utf8mb4 への変換を実行する前に、必ず Zabbix サーバーと DB スキーマを 6.0.x 以降にアップグレードしてください。

1. データベースの文字セットと照合順序を確認します。

例:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2_general_ci    |
+--------------------------+----------------------+
```

または:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

ここで、文字セットが'utf8mb4' ではなく、照合順序も'utf8mb4_bin' ではないことが分かりますので、修正が必要です。

2. Zabbix を停止します。

3. データベースのバックアップコピーを作成します！

4. データベースレベルで文字セットと照合順序を修正します:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

修正後の値:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. テーブルおよびカラムレベルの文字セットと照合順序を修正するためにスクリプト をロードします:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. スクリプトを実行します:

```
               SET @ZABBIX_DATABASE = '<your DB name>';
MariaDBの場合 →  set innodb_strict_mode = OFF;
               CALL zbx_convert_utf8();
MariaDBの場合 →  set innodb_strict_mode = ON;
               drop procedure zbx_convert_utf8;
```

'utf8mb4' はディスク容量をわずかに多く消費することにご注意ください。

7. エラーがなければ、修正後のデータベースのバックアップコピーを作成することをお勧めします。

8. Zabbix を起動します。

3 プライマリキーへのデータベースのアップグレード

概要

このセクションでは、既存のインストールでテーブルを主キーに手動でアップグレードする手順を説明します。

主キーへのアップグレードは、データのインデックス化とアクセス方法を最適化し、クエリの高速化やスペースの節約につながる場合があります。また、クラスタ構成でのデータ管理や同期を改善し、スケーリングを支援し、一部のサーバーが障害を起こしてもシステムの信頼性を維持します。

> **Attention:**
> このページで提供されている手順は上級ユーザー向けに設計されており、特定の構成に合わせて調整が必要な場合があります。主キーへのアップグレードは、時間とリソースを大量に消費する可能性があります。十分な空きディスク容量があることを確認してください。データベースのサイズや保存されているデータによっては、履歴テーブルで現在使用されている容量の最大 2.5 倍のスペースが必要になる場合があります。

Zabbix 6.0 以降の新規インストールでは、すべてのテーブルで主キーが使用されています。

主キーへの自動データベースアップグレードはありませんが、既存のインストールは Zabbix サーバーを 6.0 以降にアップグレードした後に手動でアップグレードできます。

> **Attention:**
> Zabbix 7.0 以降、テーブルを主キーにアップグレードすると同時に、テーブルが倍精度データ型を使用するようにアップグレードされます。

 Zabbix 7.0（またはそれ以降）をお使いの場合、テーブルはすでに倍精度を使用しています。ただし、このページの手順に従って、すでに倍精度を使用しているテーブルに影響を与えることなく、テーブルを主キーにアップグレードすることができます。

 Zabbix 6.4（またはそれ以前）をお使いの場合は、最初にテーブルを倍精度にアップグレードすることを検討してください。詳細については、Zabbix 7.0 ドキュメントの拡張範囲の数値値へのアップグレードを参照してください。

手順は以下のデータベース用に用意されています:

- MySQL
- PostgreSQL
- TimescaleDB

重要な注意事項

データベースのアップグレードを実行するには:

1. Zabbix サーバーを停止します。

アップグレード中は Zabbix サーバーを停止することを強く推奨します。ただし、どうしても必要な場合は、サーバーが稼働している間にアップグレードを実行することもできます（MySQL、MariaDB、および TimescaleDB なしの PostgreSQL のみ）。

2. データベースのバックアップを取ります。

3. Zabbix のバージョンに対応した最新の zabbix-sql-scripts パッケージをインストールします(例：RHEL の場合：`dnf install zabbix-sql-scripts`)。
4. データベース用のスクリプトを実行します。
5. Zabbix サーバーを起動します。

Warning:
サーバーデータベースのみに対してスクリプトを実行してください。プロキシにはこのアップグレードの恩恵はありません。

データベースでパーティションを使用している場合は、DB 管理者または Zabbix サポートにお問い合わせください。

CSV ファイルは、主キーへのアップグレードが正常に完了した後に削除できます。

オプションで、Zabbix フロントエンドをメンテナンスモードに切り替えることができます。

MySQL

セッションがドロップされないように、エクスポートとインポートは tmux/screen で実行する必要があります。

参照: 重要事項

mysqlsh を使用した MySQL 8.4+

この方法は稼働中の Zabbix サーバーでも使用できますが、アップグレード中はサーバーを停止することを推奨します。MySQL Shell (*mysqlsh*) がインストールされており、DB に接続できる必要があります。

- root（推奨）または FILE 権限を持つ任意のユーザーで MySQL コンソールにログインします。
- local_infile変数を有効にして MySQL を起動します。
- `history_upgrade_prepare.sql` を実行して古いテーブルのリネームと新しいテーブルの作成を行います。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- データのエクスポートとインポートを行います。

mysqlsh で接続します。ソケット接続を使用する場合は、パスの指定が必要な場合があります。

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

JavaScript モードに切り替えます。

```
\js
```

次のコードを実行します（CSVPATH は必要に応じて変更可能)。

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

「JavaScript is not supported」というメッセージが表示された場合は、MySQL Shell インストールに JS サポートがありません。その場合は、JavaScript モードが有効になるように、Oracle の公式MySQL Shell パッケージをインストールするか、ソースからビルドしてください。

- 移行後の手順に従って古いテーブルを削除します。

mysqlsh を使用しない MariaDB/MySQL 8.4+

このアップグレード方法は時間がかかるため、*mysqlsh* でのアップグレードが不可能な場合のみ使用してください。

テーブルのアップグレード

- root（推奨）または FILE 権限を持つ任意のユーザーで MySQL コンソールにログインします。
- Zabbix サーバーを稼働したまま移行を行う場合は、local_infile変数を有効にして MySQL を起動します。
- history_upgrade_prepare.sql を実行して古いテーブルのリネームと新しいテーブルの作成を行います。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

サーバー停止時の移行

移行中のタイムアウトを回避するため、データ移行前に max_execution_time（MySQL の場合）または max_statement_time（MariaDB の場合）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

移行後の手順に従って古いテーブルを削除してください。

サーバー稼働時の移行

インポート/エクスポートが有効なパスを確認します。

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

secure_file_priv の値がディレクトリへのパスの場合、エクスポート/インポートはそのディレクトリ内のファイルに対して実行されます。この場合、クエリ内のファイルパスを適宜編集するか、アップグレード時のみ *secure_file_priv* の値を空文字列に設定してください。

secure_file_priv の値が空の場合、任意の場所からエクスポート/インポートが可能です。

secure_file_priv の値が NULL の場合は、エクスポートしたテーブルデータを含むパス（上記例では'/var/lib/mysql-files/'）に設定してください。

詳細はMySQL ドキュメントまたはMariaDB ドキュメントを参照してください。

エクスポート中のタイムアウトを回避するため、データエクスポート前に max_execution_time（MySQL の場合）または max_statement_time（MariaDB の場合）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

移行後の手順に従って古いテーブルを削除してください。

PostgreSQL

エクスポートとインポートは、セッションが切断されないように tmux/screen で実行する必要があります。TimescaleDB を使用している場合は、このセクションをスキップし、PostgreSQL + TimescaleDBに進んでください。

参考: 重要な注意事項

テーブルのアップグレード

- history_upgrade_prepare.sql を使用してテーブル名を変更します:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

サーバー停止時の移行

- 現在の履歴をエクスポートし、一時テーブルにインポートしてから、重複を無視して新しいテーブルにデータを挿入します:

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

INSERT のパフォーマンス向上のヒントについては、PostgreSQL: Bulk Loading Huge Amounts of Data、Checkpoint Distance and Amount of WALを参照してください。

- 移行後の手順に従って古いテーブルを削除します。

サーバー稼働中の移行

- 現在のヒストリーデータをエクスポートし、一時テーブルにインポートしてから、重複を無視して新しいテーブルにデータを挿入します:

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    DOUBLE PRECISION DEFAULT '0.0000'          NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    numeric(20)     DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_str (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    varchar(255)    DEFAULT ''                NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
```

```
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,n

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    timestamp                integer         DEFAULT '0'               NOT NULL,
    source                   varchar(64)     DEFAULT ''                NOT NULL,
    severity                 integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    logeventid               integer         DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- 移行後の手順に従って古いテーブルを削除します。

PostgreSQL + TimescaleDB

エクスポートとインポートは、セッションが切断されないように tmux/screen で実行する必要があります。アップグレード中は Zabbix サーバーを停止してください。

参照: 重要な注意事項

- history_upgrade_prepare.sql を使用してテーブル名を変更します。
  - 圧縮が有効(デフォルトインストール時)な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option- のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - 圧縮が無効な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compre のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- 圧縮設定に応じて TimescaleDB ハイパーテーブルのマイグレーションスクリプトを実行します:
  - 圧縮が有効(デフォルトインストール時)な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option- のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
    ```

  - 圧縮が無効な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compre のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
    ```

INSERT パフォーマンス向上のためのTipsも参照してください。

- マイグレーション後の手順に従い、古いテーブルを削除します。

移行後

すべてのデータベースで、移行が完了したら、次のことを行います。

- すべてが期待通りに動作することを確認します。
- 古いテーブルを削除します。

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- TimescaleDB の場合は、次の古いテーブルも削除します。

```
DROP TABLE trends_old;
```

こちらもご参照ください

- パーティショニングのための auditlog テーブルの準備

4 監査ログテーブルのパーティション化の準備

概要

一部のデータベース（例えば、MySQL）では、パーティション分割カラムがテーブルの一意制約の一部である必要があります。そのため、auditlog テーブルを時間でパーティション分割するには、主キーを auditid から複合キー auditid + clock に変更する必要があります。

このセクションでは、auditlog テーブルの主キーを変更する手順について説明します。

> **Attention:**
> このページで提供されている手順は、上級ユーザー向けに設計されています。これらの手順は、お使いの構成に合わせて調整が必要な場合がありますのでご注意ください。主キーの変更は、将来のアップグレードパッチと互換性がない場合もあるため、将来のアップグレードを手動で処理する必要があるかもしれません。

 主キーの変更は、auditlog テーブルのサイズによっては多くのリソースを消費し、時間がかかる操作となる場合があります。変更作業中は Zabbix サーバーを停止し、Zabbix フロントエンドをメンテナンスモードに切り替えることを推奨します。ただし、どうしても必要な場合は、ダウンタイムなしで主キーを変更する方法もあります（下記参照）。

auditlog テーブルのパーティショニングは、大規模なセットアップでのハウスキーピングなどを改善できます。Zabbix のハウスキーピングは現在、(TimescaleDB を除き）パーティショニングされたテーブルを利用できませんが、Zabbix のハウスキーピングを無効にし、スクリプトを使用してパーティションを削除することができます。

Zabbix 7.0 以降、TimescaleDB 用の auditlog テーブルはハイパーテーブルに変換され、ハウスキーパーがチャンク単位でデータを削除できるようになりました。既存の auditlog テーブルをハイパーテーブルにアップグレードするには、TimescaleDB スキーマのアップグレードを参照してください。

MySQL

インデックスの再構築に関する重要な注意事項

MySQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。ただし、最適なデータベースパフォーマンスを確保するために、OPTIMIZE TABLE ステートメントで手動でインデックスを再構築することも強く推奨します。

インデックスの再構築には、一時的にテーブル自体が使用するのと同じくらいの追加ディスク容量が必要になる場合があります。データとインデックスの現在のサイズを取得するには、次のステートメントを実行できます。

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

利用可能なディスク容量が懸念される場合は、ダウンタイムなしで主キーを変更する手順に従ってください。他にも次のような選択肢があります:

- インデックスを手動で再構築する際のディスク使用量を減らすために、sort_buffer_size MySQL パラメータを増やすことが役立つ場合があります。ただし、この変数を変更すると、データベース全体のメモリ使用量に影響を与える可能性があります。
- 不要なデータを削除して空き容量を確保することを検討してください。
- ハウスキーパーを実行する前に、データ保存期間housekeeperパラメータを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. インデックスを再構築します（オプションですが強く推奨します。インデックス再構築に関する重要な注意事項を参照してください）。

```
OPTIMIZE TABLE auditlog;
```

ダウンタイムなしで主キーを変更する

主キーを手動で変更する方法について説明します。または、Percona のpt-online-schema-changeツールキットを使用することもできます。このツールキットは、auditlog テーブルの変更に使用されるスペースを最小限に抑えながら、以下のアクションを自動的に実行します。

1. 新しい主キーで新しいテーブルを作成し、インデックスを作成します。

```
CREATE TABLE `auditlog_new` (
  `auditid`             varchar(25)                             NOT NULL,
  `userid`              bigint unsigned                         NULL,
  `username`            varchar(100)    DEFAULT ''              NOT NULL,
  `clock`               integer         DEFAULT '0'             NOT NULL,
  `ip`                  varchar(39)     DEFAULT ''              NOT NULL,
  `action`              integer         DEFAULT '0'             NOT NULL,
  `resourcetype`        integer         DEFAULT '0'             NOT NULL,
  `resourceid`          bigint unsigned                         NULL,
  `resource_cuid`       varchar(25)                             NULL,
  `resourcename`        varchar(255)    DEFAULT ''              NOT NULL,
  `recordsetid`         varchar(25)                             NOT NULL,
  `details`             longtext                                NOT NULL,
  PRIMARY KEY (auditid,clock)
) ENGINE=InnoDB;
CREATE INDEX `auditlog_1` ON `auditlog_new` (`userid`,`clock`);
CREATE INDEX `auditlog_2` ON `auditlog_new` (`clock`);
CREATE INDEX `auditlog_3` ON `auditlog_new` (`resourcetype`,`resourceid`);
```

2. テーブルを入れ替えます。

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. 古いテーブルから新しいテーブルにデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

リソースの過剰な使用を避けるために、必要に応じて WHERE clock 句を使用した複数の INSERT INTO ステートメントでチャンク単位で実行できます。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

PostgreSQL

インデックス再構築に関する重要な注意事項

PostgreSQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。ただし、最適なデータベースパフォーマンスを確保するために、REINDEX TABLE CONCURRENTLY ステートメントを使用して手動でインデックスを再構築することを強く推奨します。

インデックスの再構築には、一時的に現在インデックスで使用されているディスク容量の最大 3 倍が必要になる場合があります。現在のインデックスサイズを取得するには、次のクエリを実行できます。

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

利用可能なディスク容量が懸念される場合は、ダウンタイムなしで主キーを変更する手順に従ってください。他にも以下のようなオプションがあります:

- インデックスを手動で再構築する際にディスク容量の使用量を減らすために、maintenance_work_mem PostgreSQL パラメータを増やすことが役立つ場合があります。ただし、この変数を変更すると、データベース全体のメモリ使用量に影響を与える可能性があります。

- より多くの空き容量がある別のディスクやテーブルスペースがある場合は、インデックス再構築時の一時的な保存場所を変更することを検討できます。一時オブジェクト用の別のテーブルスペースを指定するには、temp_tablespaces PostgreSQL パラメータを設定できます。
- 不要なデータを削除して空き容量を確保することを検討してください。
- ハウスキーパーを実行する前に、データ保存期間housekeeperパラメータを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. インデックスを再構築します（オプションですが強く推奨します。インデックス再構築に関する重要な注意事項を参照してください）。

```
REINDEX TABLE CONCURRENTLY auditlog;
```

ダウンタイムなしで主キーを変更する

主キーを手動で変更する方法について説明します。また、pg_repack 拡張機能を使用して新しいテーブルを作成し、データをコピーしてテーブルを入れ替えることも検討できます。

1. 新しい主キーで新しいテーブルを作成し、インデックスを作成します。

```
CREATE TABLE auditlog_new (
  auditid               varchar(25)                             NOT NULL,
  userid                bigint                                  NULL,
  username              varchar(100)    DEFAULT ''              NOT NULL,
  clock                 integer         DEFAULT '0'             NOT NULL,
  ip                    varchar(39)     DEFAULT ''              NOT NULL,
  action                integer         DEFAULT '0'             NOT NULL,
  resourcetype          integer         DEFAULT '0'             NOT NULL,
  resourceid            bigint                                  NULL,
  resource_cuid         varchar(25)                             NULL,
  resourcename          varchar(255)    DEFAULT ''              NOT NULL,
  recordsetid           varchar(25)                             NOT NULL,
  details               text            DEFAULT ''              NOT NULL,
  PRIMARY KEY (auditid,clock)
);
CREATE INDEX auditlog_new_1 ON auditlog_new (userid,clock);
CREATE INDEX auditlog_new_2 ON auditlog_new (clock);
CREATE INDEX auditlog_new_3 ON auditlog_new (resourcetype,resourceid);
```

2. テーブルを入れ替えます。

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. 古いテーブルから新しいテーブルにデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

リソースの過剰な使用を避けるために、必要に応じて WHERE clock 句を使用した複数の INSERT INTO 文で分割して実行することもできます。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

参考

- データベースの主キーへのアップグレード

5 データベースへの安全な接続

概要

このセクションでは、以下の間で安全な TLS 接続を行うための Zabbix のセットアップ手順と設定例を提供します。

データベース	Zabbix コンポーネント
MySQL	Zabbix フロントエンド、Zabbix サーバー、Zabbix プロキシ
PostgreSQL	Zabbix フロントエンド、Zabbix サーバー、Zabbix プロキシ

DBMS 内での接続暗号化の設定については、詳細は公式ベンダーのドキュメントを参照してください。

- MySQL: ソースおよびレプリカのレプリケーションデータベースサーバー。
- MySQL: グループレプリケーションなどのデータベースサーバー。
- PostgreSQL 暗号化オプション。

すべての例は、CentOS 8 の公式リポジトリで利用可能な MySQL CE（8.0）および PgSQL（13）の GA リリースに基づいています。

要件

暗号化を設定するには、次が必要です。

- デベロッパーサポートされている OS と OpenSSL1.1.X 以上または OpenSSL の代用品。

Note:
特に新規インストールの場合は、サポート終了した OS は避けることをお勧めします。

- デベロッパーが提供する公式リポジトリからインストールおよび保守されるデータベースエンジン（RDBMS）やオペレーティングシステムには、暗号化サポートが実装されていない古いバージョンのデータベースソフトウェアが付属していることがよくあります。たとえば、RHEL7 ベースのシステムでの PostgreSQL9.2 や暗号化サポートのない MariaDB5.5 などです。

用語

このオプションを設定すると、Zabbix サーバー/プロキシおよびフロントエンドからデータベースへの TLS 接続が強制されます。

- `required` - アイデンティティチェックなしで TLS をトランスポートモードとして接続
- `verify_ca` - TLS で接続し、証明書を検証
- `verify_full` - TLS で接続し、証明書を検証し、DBHost で指定されたデータベースのアイデンティティ（CN）が証明書と一致することを検証

Zabbix の設定

Web インターフェースからデータベース

Web インターフェースのインストール中に、データベースへの安全な接続を設定できます。

- トランスポート暗号化を有効にするにはデータベース接続設定ステップでデータベースの *TLS* 暗号化チェックボックスをオンにします。
- 証明書による暗号化を有効にするには、*TLS* 暗号化フィールドがオンになっている場合に表示されるデータベースの証明書確認チェックボックスをオンにします。

Note:
MySQL の場合、データベースホストが localhost に設定されていると、ソケットファイル（Unix の場合）または共有メモリ（Windows の場合）を使用する接続を暗号化できないため、データベースの *TLS* 暗号化チェックボックスが無効になります。PostgreSQL の場合、データベースホストフィールドの値がスラッシュで始まるか、フィールドが空の場合、*TLS* 暗号化チェックボックスは無効になります。

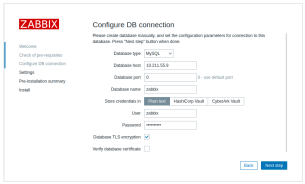
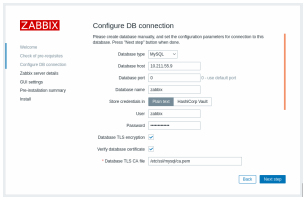
次のパラメータは、証明書モードの TLS 暗号化で使用可能になります。(両方のチェックボックスがマークされている場合)

パラメータ	説明
データベースの *TLS CA* ファイル	有効な TLS 認証局（CA）ファイルへのフルパスを指定します。
データベースの *TLS* キーファイル	有効な TLS キーファイルへのフルパスを指定します。
データベースの *TLS* 証明書ファイル	有効な TLS 証明書ファイルへのフルパスを指定します。
データベースのホスト検証	このチェックボックスをオンにすると、ホスト検証がアクティブになります。 PHP MySQL ライブラリではピア証明書の検証手順をスキップできないため、MySQL では無効になっています。
データベースの *TLS* 暗号化方式リスト	有効な暗号のカスタムリストを指定します。暗号リストの形式は、OpenSSL 標準に準拠している必要があります。 MySQL でのみ使用できます。

Attention:
TLS パラメータは有効なファイルを指している必要があります。存在しないファイルまたは無効なファイルを指している場合は、認証エラーが発生します。証明書ファイルが書き込み可能である場合、Web インターフェースはシステム情報レポートに「データベースの TLS 証明書ファイルは読み込み専用にしてください」という警告を表示します。(PHP ユーザーが証明書の所有者である場合にのみ表示されます)。
パスワードで保護された証明書はサポートされていません。

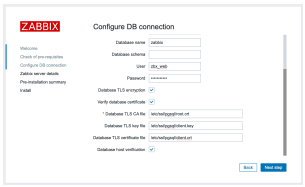
使用例

Zabbix フロントエンドは、GUI インターフェースを使用して可能なオプションを定義します: `required`、`verify_ca`、`verify_full`。インストールウィザードの *DB* 接続の設定ステップで必要なオプションを指定します。これらのオプションは、設定ファイル (zabbix.conf.php) に次のようにマッピングされます:

GUI 設定	設定ファイル	説明	結果
	... // TLS 接続に使用。 $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	データベース *TLS* 暗号化をチェック データベース証明書の検証はチェックしない	`required` モードを有効化
	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem'; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック 2. データベース *TLS CA* ファイルのパスを指定	`verify_ca` モードを有効化

GUI 設定	設定ファイル	説明	結果

	...
	// 厳密に定義された Cipher リストで TLS 接続に使用。
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '<cipher_list>';
	...
	
	または:
	
	...
	// Cipher リストが定義されていない TLS 接続に使用 - MySQL サーバーによって選択される
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック
		2. データベース *TLS* キー・ファイルのパスを指定
		3. データベース *TLS CA* ファイルのパスを指定
		4. データベース *TLS* 証明書ファイルのパスを指定
		5. データベース *TLS* 暗号スイートリストを指定 (任意)
			MySQL の `verify_full` モードを有効化

	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = '<key_file_path>'; $DB['CERT_FILE'] = '<key_file_path>'; $DB['CA_FILE'] = '<key_file_path>'; $DB['VERIFY_HOST'] = true; $DB['CIPHER_LIST'] = ' '; ...	1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック 2. データベース *TLS* キー・ファイルのパスを指定 3. データベース *TLS CA* ファイルのパスを指定 4. データベース *TLS* 証明書ファイルのパスを指定 5. データベースホストの検証をチェック	PostgreSQL の `verify_full` モードを有効化

関連情報: MySQL の暗号化設定例、PostgreSQL の暗号化設定例

Zabbix サーバー/プロキシの設定

データベースへの安全な接続は、Zabbix のサーバーおよび/またはプロキシの設定ファイルで、それぞれのパラメータを使用して設定できます。

設定	結果
なし	暗号化なしでデータベースに接続します。
1. DBTLSConnect=required を設定	サーバー/プロキシはデータベースに TLS 接続を行います。暗号化されていない接続は許可されません。
1. DBTLSConnect=verify_ca を設定 2. DBTLSCAFile を設定 - TLS 証明書認証局ファイルを指定	サーバー/プロキシは、データベース証明書を検証した後、データベースに TLS 接続を行います。
1. DBTLSConnect=verify_full を設定 2. DBTLSCAFile を設定 - TLS 証明書認証局ファイルを指定	サーバー/プロキシは、データベース証明書およびデータベースホストの識別を検証した後、データベースに TLS 接続を行います。
1. DBTLSCAFile を設定 - TLS 証明書認証局ファイルを指定 2. DBTLSCertFile を設定 - クライアント公開鍵証明書ファイルを指定 3. DBTLSKeyFile を設定 - クライアント秘密鍵ファイルを指定	サーバー/プロキシは、データベースに接続する際にクライアント証明書を提供します。
1. DBTLSCipher を設定 - クライアントが TLS 1.2 までのプロトコルでの接続に許可する暗号スイートのリスト または DBTLSCipher13 - クライアントが TLS 1.3 プロトコルでの接続に許可する暗号スイートのリスト	(MySQL) 提供されたリストから暗号スイートを使用して TLS 接続が行われます。 (PostgreSQL) このオプションを設定するとエラーとみなされます。

1 MySQL の暗号化設定

概要

このセクションでは、CentOS 8.2 および MySQL 8.4.0 の暗号化設定例をいくつか紹介しており、データベースへの接続を暗号化するためのクイックスタートガイドとして使用できます。

Attention:
MySQL ホストが localhost に設定されている場合、暗号化オプションは利用できません。この場合、Zabbix フロントエンドとデータベース間の接続はソケットファイル（Unix の場合）または共有メモリ（Windows の場合）を使用し、暗号化できません。

Note:
暗号化の組み合わせのリストは、このページに記載されているものに限定されません。他にも多くの組み合わせが利用可能です。

前提条件

公式リポジトリから MySQL データベースをインストールします。

MySQL リポジトリの使用方法については、MySQL ドキュメントを参照してください。

MySQL サーバーは自己署名証明書を使用して安全な接続を受け入れる準備ができています。

暗号化された接続を使用しているユーザーを確認するには、次のクエリを実行します（Performance Schema が ON になっている必要があります）。

```
SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, processli
FROM performance_schema.status_by_thread  AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

転送のみの暗号化

MySQL の設定

最新バージョンのデータベースは、required 暗号化モードに対応しており、すぐに利用できます。初期セットアップと起動後にサーバー側の証明書が作成されます。

主要コンポーネント用のユーザーとロールを作成します:

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
```

```
 'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

X.509 プロトコルはアイデンティティの確認には使用されませんが、ユーザーは暗号化された接続のみを使用するように設定されています。ユーザーの設定の詳細については、MySQL ドキュメントを参照してください。

接続を確認するには、以下を実行します（ソケット接続はセキュア接続のテストには使用できません）：

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

現在のステータスと利用可能な暗号スイートを確認します:

```
mysql> status
--------------
mysql Ver 8.4.0 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384

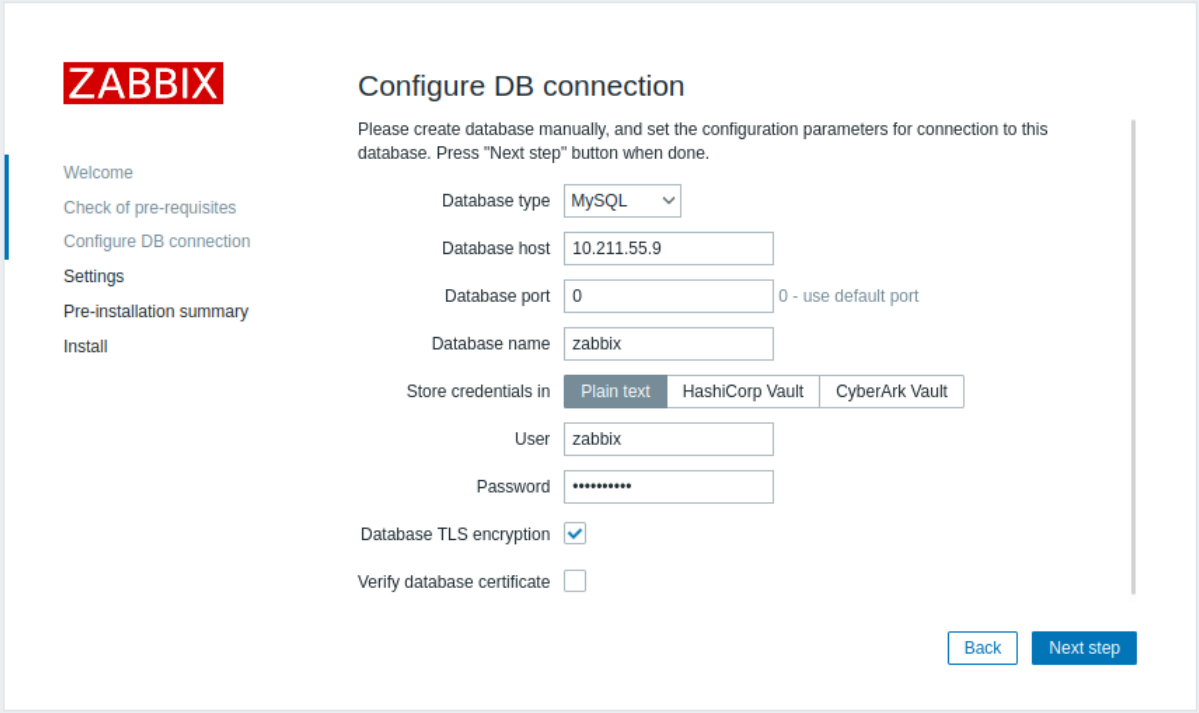

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには:

- データベース *TLS* 暗号化をチェックします
- データベース証明書の検証はチェックしません



サーバー

サーバーとデータベース間の接続でトランスポートのみの暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

証明書認証局による暗号化

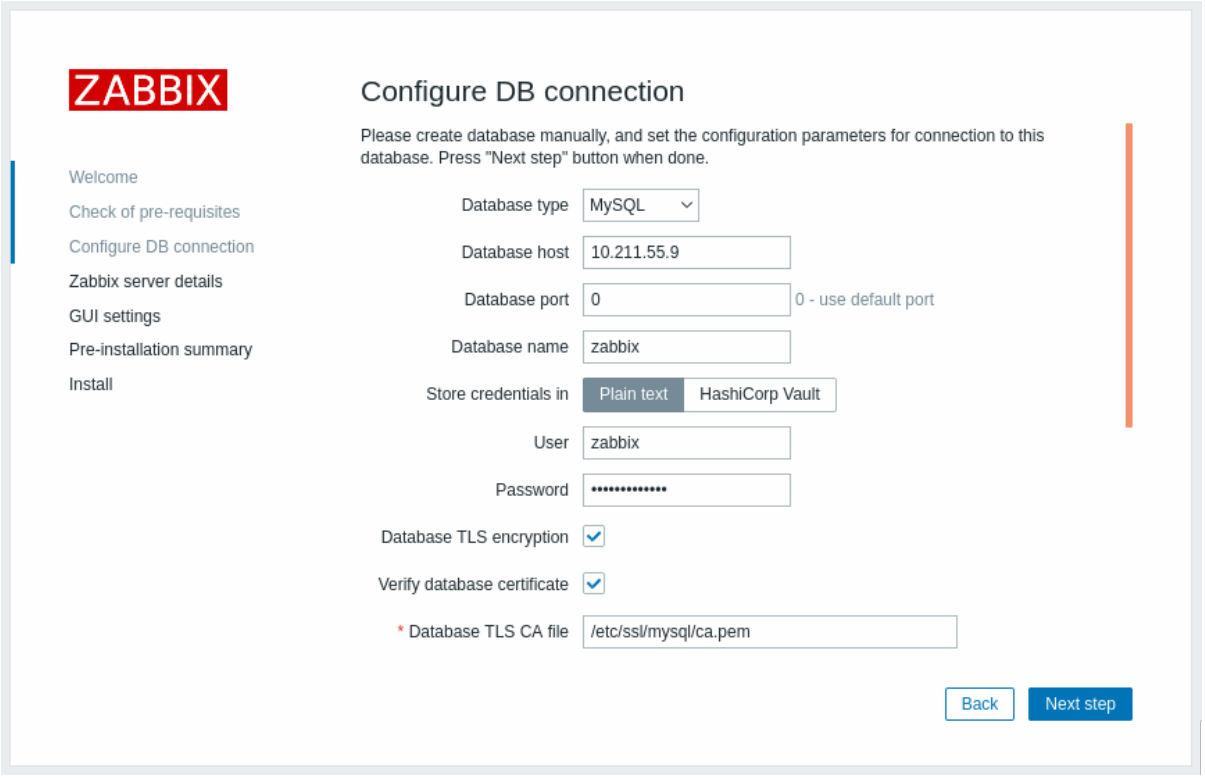
必要な MySQL CA を Zabbix フロントエンドサーバーにコピーし、Web サーバーがこのファイルを読み取れるように適切な権限を割り当てます。

Note:
このモードは、古い MySQL ライブラリのため RHEL 7 では動作しません。

フロントエンド

Zabbix フロントエンドとデータベース間の接続で証明書検証付きの暗号化を有効にするには、以下の手順を実行します。

- *Database TLS encryption* と *Verify database certificate* をチェックします
- Database TLS CA ファイルへのパスを指定します



または、*/etc/zabbix/web/zabbix.conf.php* で設定することもできます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

コマンドラインツールを使用して、必要なユーザーで接続できるかどうかを確認します。

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

サーバー

Zabbix サーバーとデータベース間の接続で証明書検証付きの暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

完全検証による暗号化

MySQL の設定

MySQL CE サーバーの設定オプション（*/etc/my.cnf.d/server-tls.cnf*）を以下のように設定します。

```
[mysqld]
...
##### この例では、キーは MySQL CE の datadir ディレクトリに配置されています
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
```

```
tls_version=TLSv1.3
...
```

MySQL CE サーバーおよびクライアント（Zabbix フロントエンド）用のキーは、MySQL CE のドキュメントに従って手動で作成する必要があります：MySQL を使用した SSL および RSA 証明書とキーの作成 または openssl を使用した SSL 証明書とキーの作成

Attention:
MySQL サーバー証明書には、Common Name フィールドに FQDN 名を設定する必要があります。Zabbix フロントエンドは、データベースとの通信に DNS 名またはデータベースホストの IP アドレスを使用します。

MySQL ユーザーを作成します。

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

そのユーザーでログインできるか確認します。

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

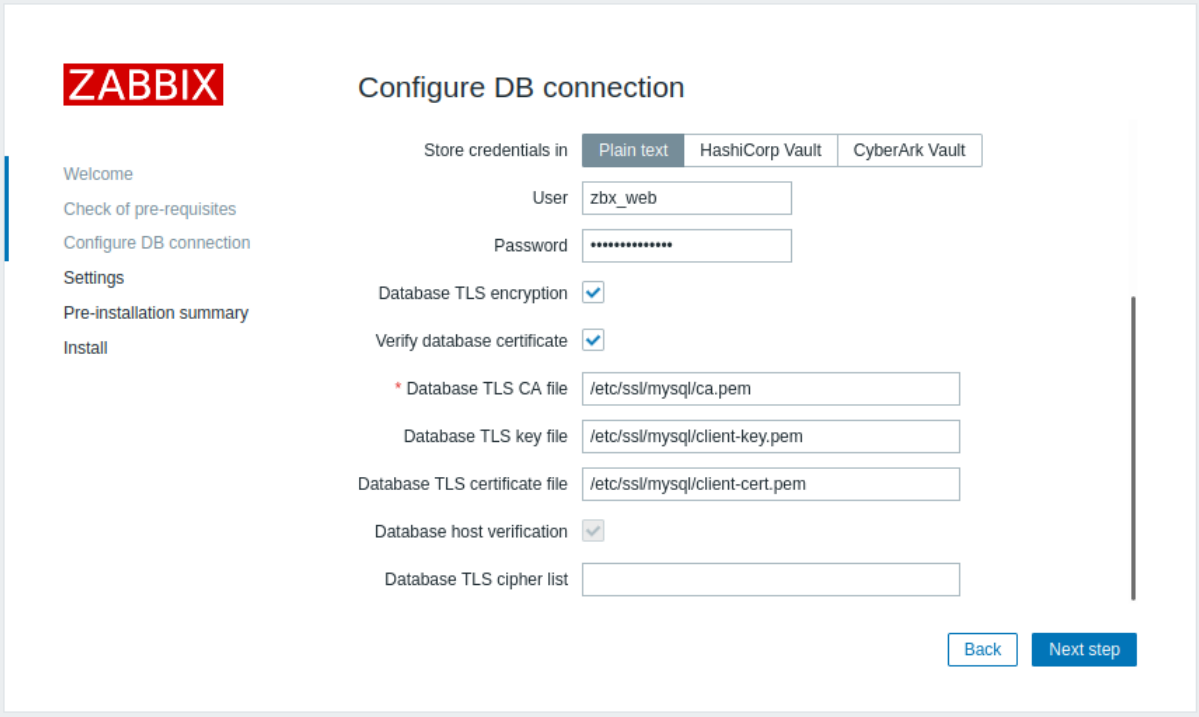
フロントエンド

Zabbix フロントエンドとデータベース間の接続で完全な検証付き暗号化を有効にするには、以下の手順を実行します。

- *Database TLS encryption* と *Verify database certificate* をチェックします
- *Database TLS key file* のパスを指定します
- *Database TLS CA file* のパスを指定します
- *Database TLS certificate file* のパスを指定します

Database host verification はチェックされており、グレーアウトされています。MySQL の場合、このステップはスキップできません。

Warning:
Database TLS cipher list フィールドを空のままにした場合、フロントエンド（クライアント）とサーバーの両方で許可されている共通の暗号が有効になります。または、暗号設定要件に従って、暗号を明示的に設定することもできます。



または、*/etc/zabbix/web/zabbix.conf.php* で設定することもできます。

```
...
// 厳密に定義された暗号リストで TLS 接続を使用する場合
$DB['ENCRYPTION'] = true;
```

```
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// または
...
// 暗号リストを定義せずに TLS 接続を使用する場合 - MySQL サーバーによって選択されます
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続で完全な検証による暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 PostgreSQL の暗号化設定

概要

このセクションでは、CentOS8.2 および PostgreSQL13 の暗号化設定例をいくつか示します。

> **Note:**
> *Database host* フィールドの値がスラッシュで始まるか、空の場合、Zabbix フロントエンドと PostgreSQL 間の接続を暗号化できません（GUI のパラメーターが無効になっています）

前提条件

公式リポジトリを使用して PostgreSQL データベースをインストールします。

PostgreSQL はデフォルトで TLS 接続を受け付けるようには設定されていません。postgresql.conf での証明書準備および ph_hba.conf によるユーザーアクセス制御については、PostgreSQL のドキュメントの指示に従ってください。

デフォルトでは、PostgreSQL ソケットは localhost にバインドされています。ネットワークのリモート接続の場合は、実際のネットワークインターフェースでリッスンできるようにしてください。

すべてのモードに対する PostgreSQL の設定例は次のようになります。

/var/lib/pgsql/13/data/postgresql.conf:

```
...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
ssl_min_protocol_version = 'TLSv1.3'
...
```

アクセス制御のために*/var/lib/pgsql/13/data/pg_hba.conf* を調整します:

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

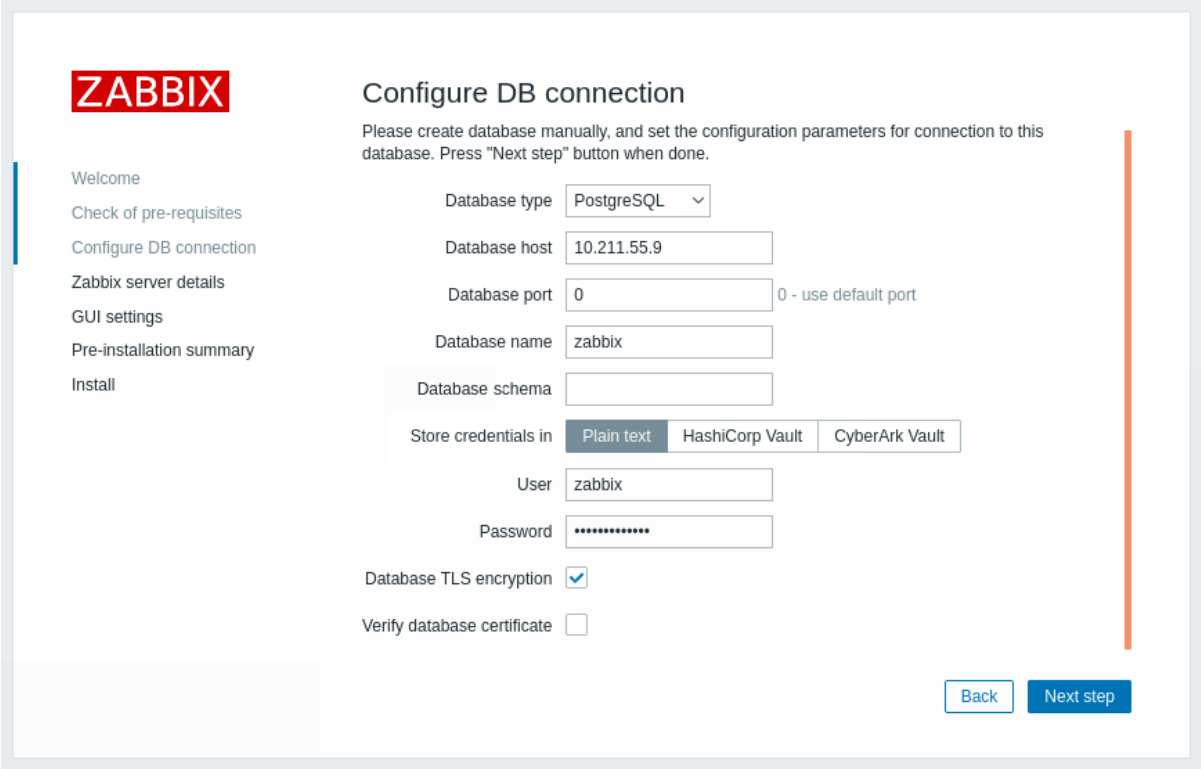
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

トランスポートのみの暗号化

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには:

- データベースの *TLS* 暗号化をチェックします
- データベース証明書の確認はチェックしません



サーバー

サーバーとデータベース間の接続に対してトランスポートのみの暗号化を有効にするには*/etc/zabbix/zabbix_server.conf* を設定します:

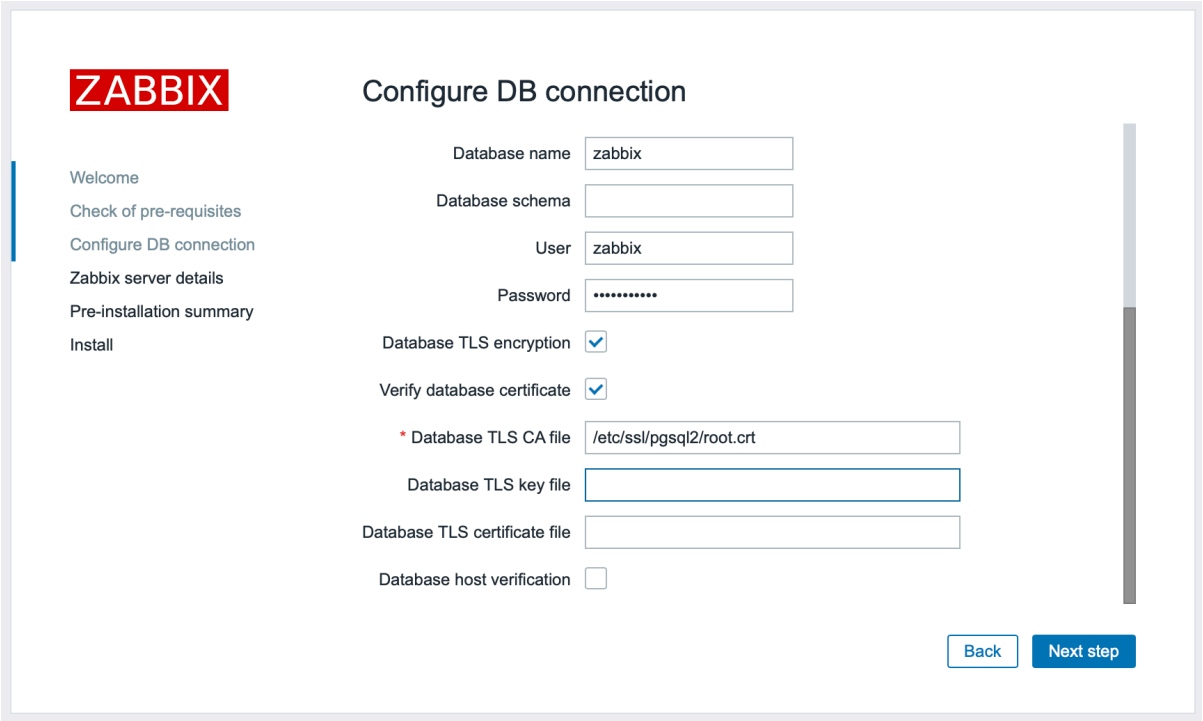
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

認証局検証による暗号化

フロントエンド

Zabbix フロントエンドとデータベース間の接続の認証局検証による暗号化を有効にするには:

- データベース *TLS* 暗号化とデータベース証明書の確認をチェック
- データベース *TLS CA* ファイルへのパスを指定



または、これは*/etc/zabbix/web/zabbix.conf.php:* で設定できます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書検証による暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します。

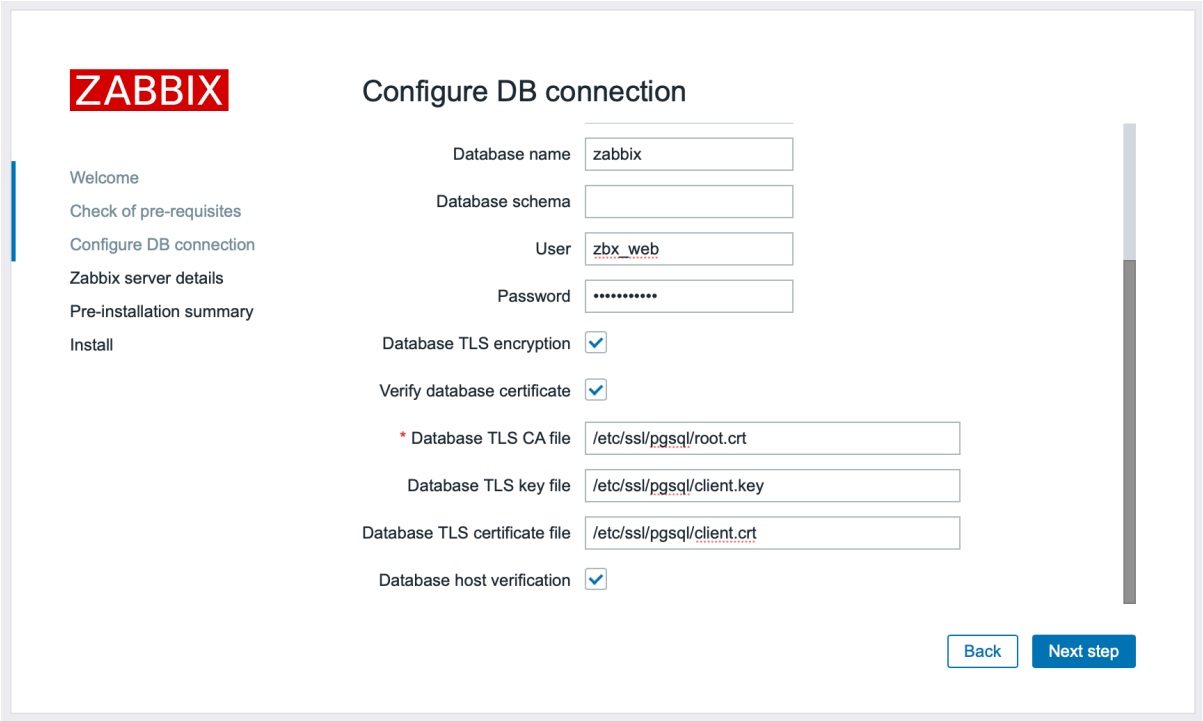
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

完全な検証を伴う暗号化

フロントエンド

Zabbix フロントエンドとデータベース間接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには：

- データベース *TLS* 暗号化とデータベース証明書の確認をチェック
- データベース *TLS* キーファイルへのパスを指定
- データベース *TLSCA* ファイルへのパスを指定
- データベース *TLS* 証明書ファイルへのパスを指定
- データベースホストの検証をチェック



または*/etc/zabbix/web/zabbix.conf.php* で設定：

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには*/etc/zabbix/zabbix_server.conf* を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

6 フロントエンドへのセキュアな接続

概要

このセクションでは、Zabbix フロントエンドと Zabbix サーバー間の安全な TLS 接続のための Zabbix セットアップ手順と設定例を示します。

設定

デフォルトでは、Zabbix フロントエンドと Zabbix サーバー間の通信は暗号化されていません。セキュリティを強化するために、両方で TLS を有効にしてください。以下は、その最も簡単な方法の例です。

1. 証明書と鍵を生成します。

作業ディレクトリを作成します:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

CA 証明書を作成します（MyZabbixCA の値は実際のコモンネームに合わせて調整してください）：

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Zabbix サーバー用の秘密鍵と証明書を生成します（zabbix-server.example.com の値は実際のコモンネームに合わせて調整してください）：

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Zabbix フロントエンド用の秘密鍵と証明書を生成します（zabbix-frontend-node の値は実際のコモンネームに合わせて調整してください）：

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. 適切なパーミッションを設定します。

Zabbix サーバー用（ディストリビューションの Zabbix サーバーデーモンユーザーに合わせて所有者/グループを調整してください）：

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

フロントエンド用（ディストリビューションの Web サーバーユーザーに合わせて所有者/グループを調整してください）：

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Zabbix サーバーを設定します。

zabbix_server.conf に以下を追加します:

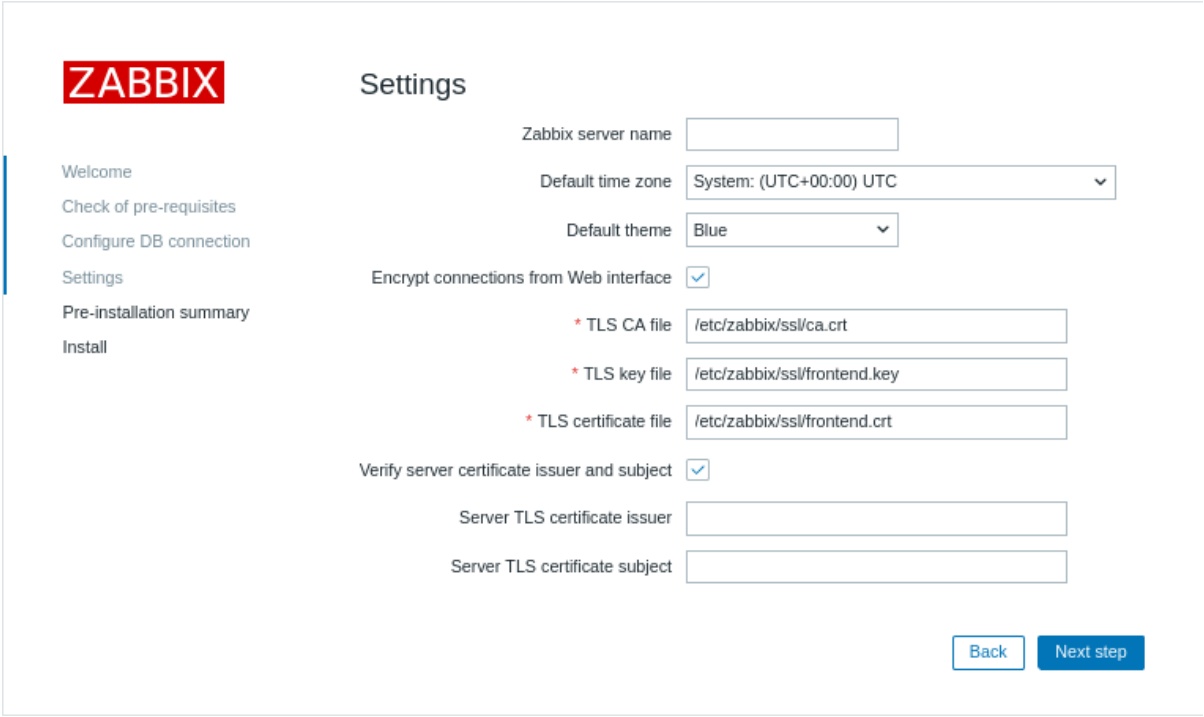
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### オプション:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

その後、サーバーを再起動します:

```
sudo systemctl restart zabbix-server
```

4. Zabbix フロントエンドを設定します。

Web インターフェースのインストール時に、*Web* インターフェースからの接続の暗号化オプション（必要に応じてサーバー証明書の発行者とサブジェクトの検証オプション）を有効にし、*TLS CA* ファイル、*TLS* 鍵ファイル、*TLS* 証明書ファイルの各フィールド（必要に応じてサーバー *TLS* 証明書の発行者およびサーバー *TLS* 証明書のサブジェクトフィールド）に入力します:



パラメータ	説明
TLS CA ファイル	サーバー証明書の検証に使用する認証局（CA）証明書ファイルへのフルパスを指定します。
TLS 鍵ファイル	クライアント証明書に対応するクライアント秘密鍵ファイルへのフルパスを指定します。
TLS 証明書ファイル	相互 TLS 認証が必要な場合は、クライアント証明書ファイルへのフルパスを指定します。
サーバー *TLS* 証明書の発行者	サーバー証明書と照合する発行者の識別名（DN）を指定します。
サーバー *TLS* 証明書のサブジェクト	サーバー証明書と照合するサブジェクトの識別名（DN）を指定します。

既存のインストールでは、`zabbix.conf.php` の以下のフィールドを編集します:

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// オプション:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Zabbix フロントエンドまたは Zabbix サーバーログファイルにエラーメッセージが表示されていないことを確認して、暗号化を確認します:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 TimescaleDB のセットアップ

概要

Zabbix は、スケール時のパフォーマンス向上をサポートするために、データを自動的に時間ベースのチャンクにパーティション分割する PostgreSQL ベースのデータベースソリューションである TimescaleDB をサポートしています。

> **Warning:**
> 現在、Zabbix プロキシでは TimescaleDB はサポートされていません。

このページの手順は、以下のシナリオで使用できます。

- TimescaleDB データベースの作成、または既存の PostgreSQL テーブルから TimescaleDB への移行（設定を参照）。
- Zabbix のアップグレード時に既存の TimescaleDB データベーススキーマをアップグレードする（TimescaleDB スキーマのアップグレードを参照）。

設定

前提条件: データベースサーバーにサポートされているバージョンの TimescaleDB 拡張機能がインストールされていること。インストール手順については、TimescaleDB ドキュメントを参照してください。

> **Warning:**
> TimescaleDB をインストールする前に、公式 PostgreSQLリポジトリからサポートされている PostgreSQL リリースをインストールしてください。

特定のデータベースで TimescaleDB 拡張機能を有効にするには、次のコマンドを実行します。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

このコマンドを実行するには、データベース管理者権限が必要です。

> **Note:**
> 'public' 以外のデータベーススキーマを使用している場合は、上記のコマンドに SCHEMA 句を追加する必要があります。例: `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

次に、`postgresql/timescaledb/schema.sql` スクリプトを実行します。新規インストールの場合は、通常の PostgreSQL データベースが初期スキーマ/データで作成された後にスクリプトを実行する必要があります（データベース作成を参照）。

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> TimescaleDB バージョン 2.9.0 以降で `schema.sql` スクリプトを実行する際に、ベストプラクティスに従っていないことを通知する警告メッセージは無視してください。この警告に関係なく、設定は正常に完了します。

既存の履歴、トレンド、監査ログデータの移行には多くの時間がかかる場合があります。移行期間中は Zabbix サーバーとフロントエンドを停止する必要があります。

`schema.sql` スクリプトは、次のハウスキーピングパラメータを設定します。

- アイテム履歴期間の上書き
- アイテムトレンド期間の上書き

履歴とトレンドのパーティション化されたハウスキーピングを使用するには、これらのオプションの両方を有効にする必要があります。履歴のみ、またはトレンドのみの上書きを個別に有効にすることも可能です。

PostgreSQL および TimescaleDB の場合、`postgresql/timescaledb/schema.sql` スクリプトはさらに 2 つのパラメータを設定します。

- 圧縮を有効にする
- 7 日より古いレコードを圧縮する

ハウスキーパーによって圧縮データを正常に削除するには、アイテムの履歴保存期間を上書きとアイテムのトレンド保存期間を上書きの両方のオプションを有効にする必要があります。上書きが無効で、テーブルに圧縮チャンクがある場合、ハウスキーパーはこれらのテーブルからデータを削除せず、ハウスキーピングおよびシステム情報セクションに不正な設定に関する警告が表示されます。

これらのパラメータはすべて、インストール後に 管理 > ハウスキーピング で変更できます。

> **Note:**
> `postgresql.conf` の PostgreSQL 構成パラメータを最適化するために、TimescaleDB が提供する timescaledb-tune ツールを実行することをお勧めします。

TimescaleDB スキーマのアップグレード

新しい TimescaleDB ハイパーテーブルを含むバージョンに Zabbix をアップグレードする場合、Zabbix サーバーはそれらのハイパーテーブルを自動的に設定しません（例えば、Zabbix 6.4 から 8.0 にアップグレードする場合、バージョン 7.0.0 および 7.0.2 で新しいハイパーテーブルが導入されています）。

新しい TimescaleDB ハイパーテーブルを設定するには、以下の手順に従ってください。

1. Zabbix サーバーを起動します。これにより既存のデータベースがアップグレードされます。
2. サーバーログファイルでデータベースのアップグレードが完了したことを確認し、完了したら Zabbix サーバーを停止します。サーバーがハイパーテーブルでないテーブルに対して圧縮を有効にしようとした場合、サーバーログに警告が記録されることに注意してください。

3. `postgresql/timescaledb/schema.sql` スクリプトを実行します。これにより新しい TimescaleDB ハイパーテーブルが設定されます。Zabbix 7.0.0 以降、スクリプトの場所と名前が `postgresql/timescaledb.sql` から `postgresql/timescaledb/schema.sql` に変更されていることに注意してください。

Attention:
TimescaleDB バージョン 2.9.0 以降で `schema.sql` スクリプトを実行する際に、ベストプラクティスに従っていないことを通知する警告メッセージは無視してください。この警告に関係なく、設定は正常に完了します。

TimescaleDB 圧縮

ネイティブの TimescaleDB 圧縮は、TimescaleDB ハイパーテーブルであるすべての Zabbix テーブルでサポートされています。アップグレードや TimescaleDB への移行時に、大きなテーブルの初期圧縮には多くの時間がかかる場合があります。

圧縮は「timescale」Timescale Community ライセンスの下でサポートされており、「apache」Apache 2.0 ライセンスの下ではサポートされていないことに注意してください。Zabbix が圧縮がサポートされていないことを検出した場合、警告メッセージが Zabbix サーバーログに書き込まれ、ユーザーはフロントエンドで圧縮を有効にできません。

Note:
圧縮を使用する前に、TimescaleDB ドキュメントで圧縮について理解することをお勧めします。

圧縮には特定の制限があることに注意してください。具体的には:

- 圧縮されたチャンクの変更（挿入、削除、更新）は許可されていません
- 圧縮されたテーブルのスキーマ変更は許可されていません。

圧縮設定は、Zabbix フロントエンドの管理 > ハウスキーピングセクションの履歴、トレンド、監査ログの圧縮ブロックで変更できます。

パラメータ	デフォルト	コメント
圧縮を有効にする	有効	チェックボックスのオン/オフは、圧縮を即座に有効/無効にするものではありません。圧縮はハウスキーパーによって処理されるため、変更は最大で `HousekeepingFrequency` 時間（zabbix_server.confで設定）×2 以内に反映されます。 圧縮を無効にした後、圧縮期間に該当する新しいチャンクは圧縮されません。ただし、以前に圧縮されたすべてのデータは圧縮されたままです。以前に圧縮されたチャンクの圧縮を解除するには、TimescaleDB ドキュメントの手順に従ってください。
指定日数より古いレコードを圧縮	7d	TimescaleDB サポート付きの古いバージョンの Zabbix からアップグレードする場合、デフォルトで圧縮は有効になりません。 このパラメータは 7 日未満にはできません。 圧縮チャンクの不変性のため、すべての遅延データ（例: プロキシによって遅延したデータ）でこの値より古いものは破棄されます。

トレンドの更新パフォーマンスを向上させるために、`trends` および `trends_uint` テーブルの「chunk_time_interval」を 30 日から 7 日またはそれ以下に下げることをお勧めします (トレンドを使用するアイテム数に応じて調整)。この設定の目的は、TimescaleDB のベストプラクティスに従い、チャンクサイズがシステムの利用可能なリソース内に収まるようにすることです。

8 Elasticsearch のセットアップ

Attention:
Elasticsearch のサポートは実験的なものです！

Zabbix は、データベースの代わりに Elasticsearch を使用して履歴データを保存することをサポートしています。ユーザーは、互換性のあるデータベースと Elasticsearch のどちらに履歴データを保存するかを選択できます。このセクションで説明するセットアップ手順は、Elasticsearch バージョン 7.X に適用されます。より古い、または新しいバージョンの Elasticsearch を使用する場合、一部の機能が意図したとおりに動作しない可能性があります。

Warning:
1. すべての履歴データが Elasticsearch に保存されている場合、トレンドは計算も保存もされません。トレンドが計算・保存されない場合、履歴の保存期間を延長する必要があるかもしれません。
 2. Elasticsearch を使用する場合、データベースから値を取得する範囲クエリは、データ保存期間のタイムスタンプによって制限されます。

設定

関連するすべての要素間の適切な通信を確保するには、サーバー構成ファイルとフロントエンド構成ファイルのパラメーターが適切に構成されていることを確認してください。

Zabbix サーバーとフロントエンド

更新するパラメーターを含む Zabbix サーバー構成ファイルの草案：

```
### Option: HistoryStorageURL
# History storage HTTP[S] URL.
#
# Mandatory: no
# Default:
# HistoryStorageURL=
### Option: HistoryStorageTypes
# Comma separated list of value types to be sent to the history storage.
#
# Mandatory: no
# Default:
# HistoryStorageTypes=uint,dbl,str,log,text
```

Zabbix サーバー構成ファイルに入力するパラメーター値の例：

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,text
```

この設定により、Zabbix サーバーは数値タイプの履歴値を対応するデータベースに保存し、テキスト履歴データを Elasticsearch に保存します。

Elasticsearch は次のアイテムタイプをサポートしています：

```
uint,dbl,str,log,text
```

サポートされているアイテムタイプの説明：

アイテム値のタイプ	データベーステーブル	**Elasticsearch** でのタイプ
符号なし数値型	history_uint	uint
浮動小数点数型	history	dbl
固定長文字列	history_str	str
ログ	history_log	log
テキスト	history_text	text

更新するパラメータを含む Zabbix フロントエンド設定ファイル（conf/zabbix.conf.php）の草案：

```
// Elasticsearch url (can be string if same url is used for all types).
$HISTORY['url']   = [
      'uint' => 'http://localhost:9200',
      'text' => 'http://localhost:9200'
];
// Value types stored in Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Zabbix フロントエンド設定ファイルに入力するパラメータ値の例：

```
$HISTORY['url']   = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

この設定により、Elasticsearch に Text、Character、および Log の履歴値が強制的に保存されます。

また、すべてが正しく機能するように conf/zabbix.conf.php で $HISTORY をグローバルにする必要があります。(方法については conf/zabbix.conf.php.example を参照してください)

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;
```

Elasticsearch のインストールとマッピングの作成

最後の 2 つのステップは、Elasticsearch 自体のインストールとマッピングプロセスの作成です。

Elasticsearch のインストールについては、Elasticsearch インストールガイドを参照してください。

> **Note:**
> マッピングは Elasticsearch のデータ構造（データベースのテーブルに類似）です。すべての履歴データタイプのマッピングはここにあります：database/elasticsearch/elasticsearch.map

> **Warning:**
> マッピングの作成は必須です。指示通りにマッピングを作成しないと、一部の機能が動作しません。

text タイプのマッピングを作成するには、以下のリクエストを Elasticsearch に送信します。

```
curl -X PUT \
 http://your-elasticsearch.here:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "fields": {
               "analyzed": {
                  "index": true,
                  "type": "text",
                  "analyzer": "standard"
               }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

同様のリクエストを、対応するタイプの修正を行いながら、Character および Log 履歴値のマッピング作成にも実行する必要があります。

> **Note:**
> Elasticsearch を使用する場合は、要件を参照してください。

> **Note:**
> ハウスキーパーは、Elasticsearch からデータを削除しません。

複数の日付ベースのインデックスに履歴データを保存する

このセクションでは、パイプラインと取り込みノードを操作するために必要な追加の手順について説明します。

まず、インデックスのテンプレートを作成する必要があります。

次の例は、uint テンプレートを作成するためのリクエストです：

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/uint_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "uint*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "type": "long"
         }
      }
   }
}'
```

他のテンプレートを作成するには、URL を変更し（最後の部分はテンプレートの名前です）、「index_patterns」フィールドをインデックス名と一致するように変更して、データベース/elasticsearch/elasticsearch.map から取得できる有効なマッピングを設定する必要があります。

たとえば、次のコマンドを使用して、テキストインデックスのテンプレートを作成できます。

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "text*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "fields": {
               "analyzed": {
                  "index": true,
                  "type": "text",
                  "analyzer": "standard"
```

```
            }
         },
         "index": false,
         "type": "text"
      }
   }
}'
```

これは、Elasticsearch が自動的に作成されたインデックスに有効なマッピングを設定できるようにするために必要です。次に、パイプライン定義を作成する必要があります。パイプラインは、データをインデックスに入れる前の、ある種のデータの前処理です。次のコマンドを使用して作成できます uint インデックスのパイプライン：

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily uint index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [
               "UNIX"
            ],
            "index_name_prefix": "uint-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

ユーザーは、丸めパラメーター (”date_rounding”) を変更して、特定のインデックスローテーション期間を設定できます。他のパイプラインを作成するには、URL（最後の部分はパイプラインの名前）を変更し”index_name_prefix” フィールドをインデックス名と一致するように変更する必要があります。

Elasticsearch ドキュメントも参照してください

さらに、履歴データを複数の日付ベースのインデックスに保存することも、Zabbix サーバー構成の新しいパラメーターで有効にする必要があります。

```
### Option: HistoryStorageDateIndex
# Enable preprocessing of history values in history storage to store values in different indices based on
# 0 - disable
# 1 - enable
#
# Mandatory: no
# Default:
# HistoryStorageDateIndex=0
```

トラブルシューティング

次の手順は Elasticsearch のセットアップに関する問題のトラブルシューティングに役立つ場合があります。

1. マッピングが正しいかどうかを確認します (`http://localhost:9 00/uint` などの必要なインデックス URL への GET リクエスト)。
2. シャードが失敗状態になっていないかどうかを確認します (Elasticsearch の再起動が役立つはずです)。
3. Elasticsearch の設定を確認します。Zabbix フロントエンドホストと Zabbix サーバーホストからのアクセスを許可する必要があります。
4. Elasticsearch ログを確認します。
5. LogSlowQueriesを使用して、ElasticSearch データベースで遅いクエリを確認できます。

それでもインストールで問題が発生する場合は、このリストのすべての情報 (マッピング、エラーログ、構成、バージョンなど) を含むバグレポートを作成してください。

9 Zabbix 用 **Nginx** のセットアップに関するディストリビューション固有の注意事項

RHEL

RHEL 8、9、10 では、Nginx は AppStream に含まれており、追加のリポジトリは不要です。以下のコマンドで有効化およびインストールできます。

```
dnf module list nginx
dnf -y install nginx
```

RHEL 7 では、Nginx は CentOS 7 Extras リポジトリ（CentOS を使用している場合）からインストールできます。

```
yum -y install nginx
```

最新のアップストリームビルドを利用したい場合は、公式の NGINX リポジトリを `/etc/yum.repos.d/nginx.repo` に設定してください。

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

SUSE Linux Enterprise Server 15 では、`php-fpm` を設定する必要があります（設定ファイルのパスはサービスパックによって若干異なる場合があります）：

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

10 root としてエージェントを実行する

Zabbix **5.0.0** 以降、公式パッケージの Zabbix エージェント用 systemd サービスファイルには、`User` および `Group` のディレクティブが明示的に含まれています。両方とも `zabbix` に設定されています。

Zabbix エージェントがどのユーザーとして実行されるかを `zabbix_agentd.conf` ファイルで設定することはできなくなりました。なぜなら、エージェントはこの設定をバイパスし、systemd サービスファイルで指定されたユーザーとして実行されるためです。Zabbix エージェントを root として実行するには、以下に記載されている変更を行う必要があります。

Zabbix エージェント

Zabbix エージェントのデフォルトのユーザーとグループをオーバーライドするには、次のコマンドを実行します。

```
systemctl edit zabbix-agent
```

次に以下のコンテンツを追加します。

```
[Service]
User=root
Group=root
```

デーモンをリロードし、zabbix-agent サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix エージェントの場合、これにより、`zabbix_agentd.conf` ファイルでユーザーを設定する機能が再度有効になります。次に、エージェントの設定ファイルで `User=root` および `AllowRoot=1` 設定パラメーターを設定する必要があります。

Zabbix agent 2

Zabbix agent 2 のデフォルトのユーザーとグループを上書きするには、次のコマンドを実行します。

```
systemctl edit zabbix-agent2
```

次に、以下の内容を追加します。

```
[Service]
User=root
```

```
Group=root
```

デーモンをリロードし、zabbix-agent2 サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Zabbix agent2 の場合、これで完全に実行ユーザーが決定されます。追加の修正は必要ありません。

11 Microsoft Windows 上の Zabbix エージェント

エージェントの設定

両世代の Zabbix エージェントは Windows サービスとして実行されます。Zabbix agent 2 の場合は、以下の手順で *agentd* を *agent2* に置き換えてください。

Microsoft Windows ホスト上で、Zabbix エージェントの単一インスタンスまたは複数インスタンスを実行できます。単一インスタンスの場合、以下のいずれかを使用できます。

- エージェントバイナリと同じディレクトリにあるデフォルトの設定ファイル
- コマンドラインで指定した設定ファイル

複数インスタンスの場合、各エージェントインスタンスには独自の設定ファイルが必要です（インスタンスの 1 つはデフォルトの設定ファイルを使用できます）。

設定ファイルのサンプルは、Zabbix ソースアーカイブ内に以下のように用意されています。

- Zabbix agent 用は `conf/zabbix_agentd.conf`
- Zabbix agent2 用は `conf/zabbix_agent2.conf`

アーカイブから Windows 用の Zabbix agent/agent 2 を設定ファイルを明示的に指定せずにサービスとしてインストールしたい場合は、エージェントをインストールする前に以下を行ってください。

- `conf/zabbix_agentd.conf` を zabbix_agentd.exe をインストールするディレクトリに手動でコピーする
- `conf/zabbix_agent2.conf` および `conf/zabbix_agent2.d` ディレクトリを zabbix_agent2.exe をインストールするディレクトリに手動でコピーする

Zabbix Windows エージェントの設定の詳細については、設定ファイルのオプションを参照してください。

Hostname パラメータ

ホストでアクティブチェックを実行するには、Zabbix エージェントでホスト名を定義する必要があります。さらに、エージェント側で設定されたホスト名の値は、フロントエンドでホストに設定された”ホスト名” と完全に一致している必要があります。

エージェント側のホスト名の値は、エージェントの設定ファイル内の **Hostname** または **HostnameItem** パラメータのいずれかで定義できます。これらのパラメータが指定されていない場合は、デフォルト値が使用されます。

HostnameItem パラメータのデフォルト値は、”system.hostname” エージェントキーによって返される値です。Windows の場合、gethostname() 関数の結果を返します。この関数は名前空間プロバイダーに問い合わせてローカルホスト名を決定します。名前空間プロバイダーが応答しない場合は、NetBIOS 名が返されます。

Hostname のデフォルト値は、HostnameItem パラメータによって返される値です。したがって、これらのパラメータが両方とも指定されていない場合、実際のホスト名はホストの NetBIOS 名になります。Zabbix エージェントは NetBIOS ホスト名を使用して Zabbix サーバーからアクティブチェックのリストを取得し、結果を送信します。

”system.hostname” キーは、*type* と *transform* の 2 つのオプションパラメータをサポートしています。

type は、アイテムが返す名前のタイプを決定します:

- *netbios*（デフォルト）- 15 文字に制限され、すべて大文字の NetBIOS ホスト名を返します。
- *host* - 大文字小文字を区別し、完全な実際の Windows ホスト名（ドメインなし）を返します。
- *shorthost* - 最初のドットの前のホスト名の部分を返します。名前にドットが含まれていない場合は、完全な文字列を返します。
- *fqdn* - 完全修飾ドメイン名（末尾のドットなし）を返します。

transform は、ホスト名に追加の変換ルールを指定できます:

- *none*（デフォルト）- 元の大文字小文字を使用します。
- *lower* - テキストを小文字に変換します。

したがって、zabbix_agentd.conf ファイルの設定を簡素化し、統一するために、次の 3 つのアプローチが使用できます:

1. **Hostname** または **HostnameItem** パラメータを未定義のままにすると、Zabbix エージェントは NetBIOS ホスト名をホスト名として使用します。

2. **Hostname** パラメータを未定義のままにし、**HostnameItem** を次のように定義します:
   HostnameItem=system.hostname[host] - Zabbix エージェントが完全な実際の（大文字小文字を区別する）Windows ホスト名をホスト名として使用する場合
   HostnameItem=system.hostname[shorthost,lower] - Zabbix エージェントが最初のドットの前のホスト名の部分のみを小文字に変換して使用する場合
   HostnameItem=system.hostname[fqdn] - Zabbix エージェントが完全修飾ドメイン名をホスト名として使用する場合

ホスト名は、Windows サービス名の一部としても使用されます。これは、Windows サービスのインストール、開始、停止、アンインストールに使用されます。たとえば、Zabbix エージェントの設定ファイルで `Hostname=Windows_db_server` と指定されている場合、エージェントは Windows サービス"`Zabbix Agent [Windows_db_server]`" としてインストールされます。したがって、各 Zabbix エージェントインスタンスに異なる Windows サービス名を持たせるには、各インスタンスが異なるホスト名を使用する必要があります。

Windows サービスとしてのエージェントのインストール

エージェントをインストールする前に、`conf/zabbix_agentd.conf` を zabbix_agentd.exe をインストールするディレクトリに手動でコピーしてください。

デフォルトの設定ファイルで Zabbix エージェントの単一インスタンスをインストールするには:

```
zabbix_agentd.exe --install
```

Attention:
64 ビットシステムでは、64 ビットプロセスの実行に関連するすべてのチェックが正しく動作するために、64 ビット版の Zabbix エージェントが必要です。

デフォルト以外の設定ファイルを使用したい場合は、サービスインストールのために以下のコマンドを使用してください:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

設定ファイルへのフルパスを指定する必要があります。

Zabbix エージェントの複数のインスタンスを以下のようにサービスとしてインストールできます:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

インストールされたサービスはコントロールパネルで確認できます。

エージェントの起動

エージェントサービスを開始するには、コントロールパネルを使用するか、コマンドラインから実行します。

デフォルトの設定ファイルで Zabbix エージェントの単一インスタンスを起動するには：

```
zabbix_agentd.exe --start
```

Zabbix エージェントの単一インスタンスを別の設定ファイルで起動するには：

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Zabbix エージェントの複数インスタンスのうち 1 つを開始するには：

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

エージェントの停止

エージェントサービスを停止するには、コントロールパネルを使用するか、コマンドラインからコマンドを実行します。

デフォルトの設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
zabbix_agentd.exe --stop
```

別の設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Zabbix エージェントの複数インスタンスのうち 1 つを停止するには

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Windows サービスとしてのエージェントのアンインストール

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --uninstall
```

デフォルト以外の設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Windows サービスから Zabbix エージェントの複数のインスタンスをアンインストールするには

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

制限

Windows 用 Zabbix エージェントは、CPU が NUMA ノード全体に不均一に分散されている非標準の Windows 構成をサポートしていません。論理 CPU が不均一に分散されている場合、一部の CPU では CPU パフォーマンスメトリックが使用できない場合があります。たとえば、2 つの NUMA ノードを持つ 72 個の論理 CPU がある場合、両方のノードにそれぞれ 36 個の CPU が必要です。

12 Microsoft Entra ID での SAML 設定

概要

このセクションでは、SAML 2.0 認証を使用して Microsoft Entra ID（旧 Microsoft Azure Active Directory）から Zabbix へのシングルサインオンおよびユーザーのプロビジョニングを設定するためのガイドラインを提供します。

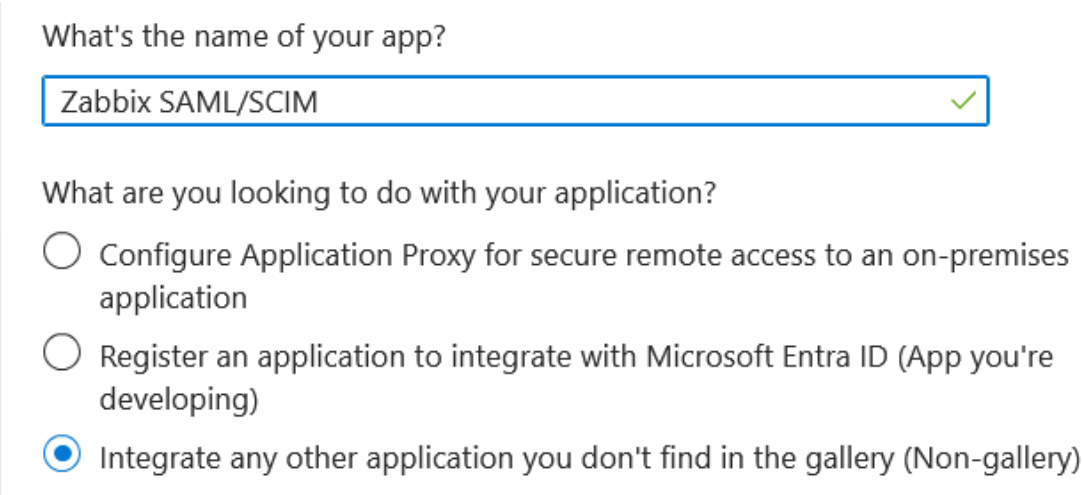
Microsoft Entra ID の設定

アプリケーションの作成

1. Microsoft Entra IDで Microsoft Entra 管理センターにログインします。テスト目的の場合は、Microsoft Entra ID で無料トライアルアカウントを作成できます。

2. Microsoft Entra 管理センターで、アプリケーション -> エンタープライズアプリケーション -> 新しいアプリケーション -> 独自のアプリケーションを作成を選択します。

3. アプリケーション名を追加し、他のアプリケーションを統合... オプションを選択します。その後、作成をクリックします。

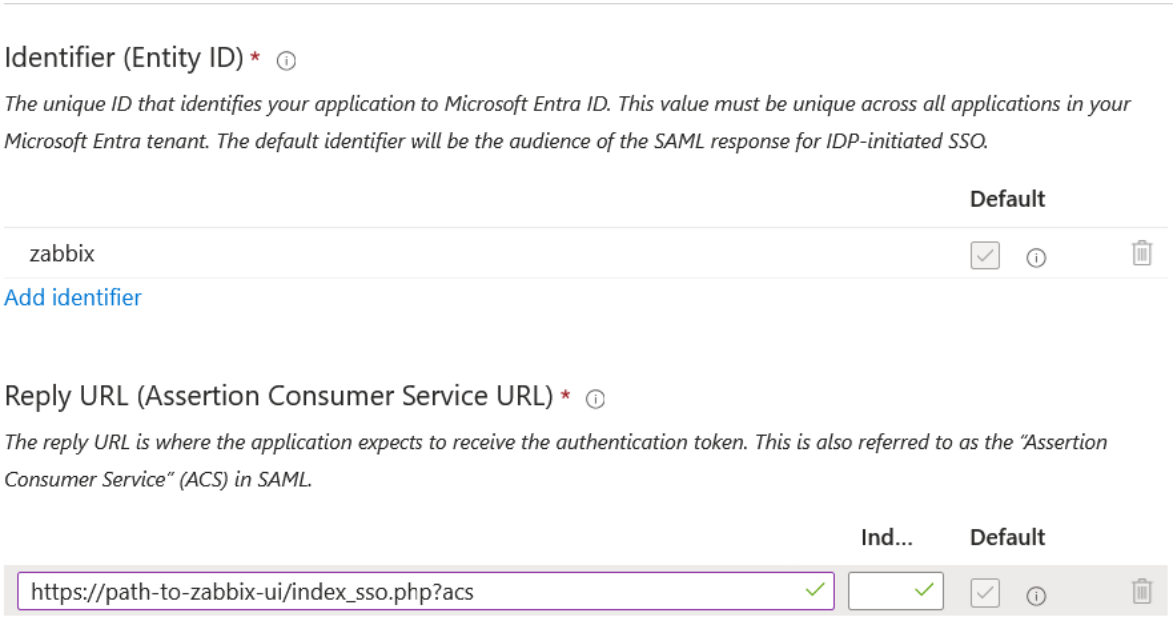


シングルサインオンの設定

1. アプリケーションページで、*Set up single sign on* に移動し、*Get started* をクリックします。次に *SAML* を選択します。

2. *Basic SAML Configuration* を編集します:

- *Identifier (Entity ID)* には、Microsoft Entra ID でアプリを識別するための一意の名前（例：`zabbix`）を設定します。
- *Reply URL (Assertion Consumer Service URL)* には、Zabbix のシングルサインオンエンドポイント `https://<path-to-zabbix-ui>/ir` を設定します。



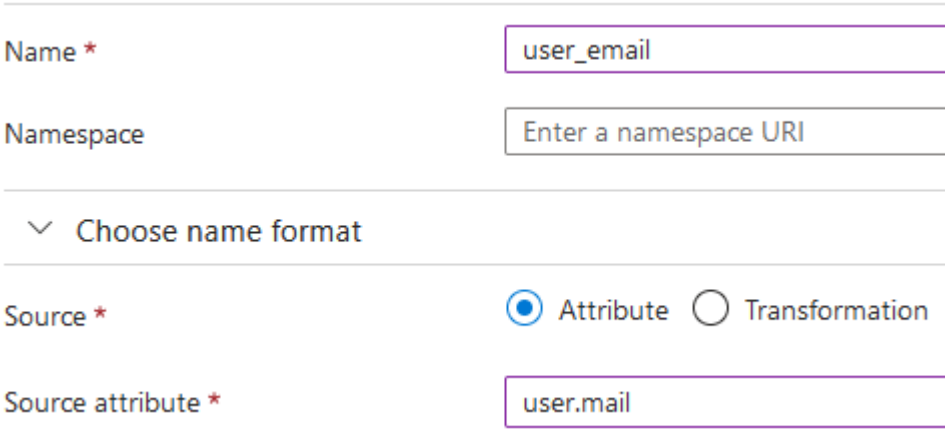
"https" が必須であることに注意してください。Zabbix でこれを機能させるには、`conf/zabbix.conf.php` に以下の行を追加する必要があります。

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

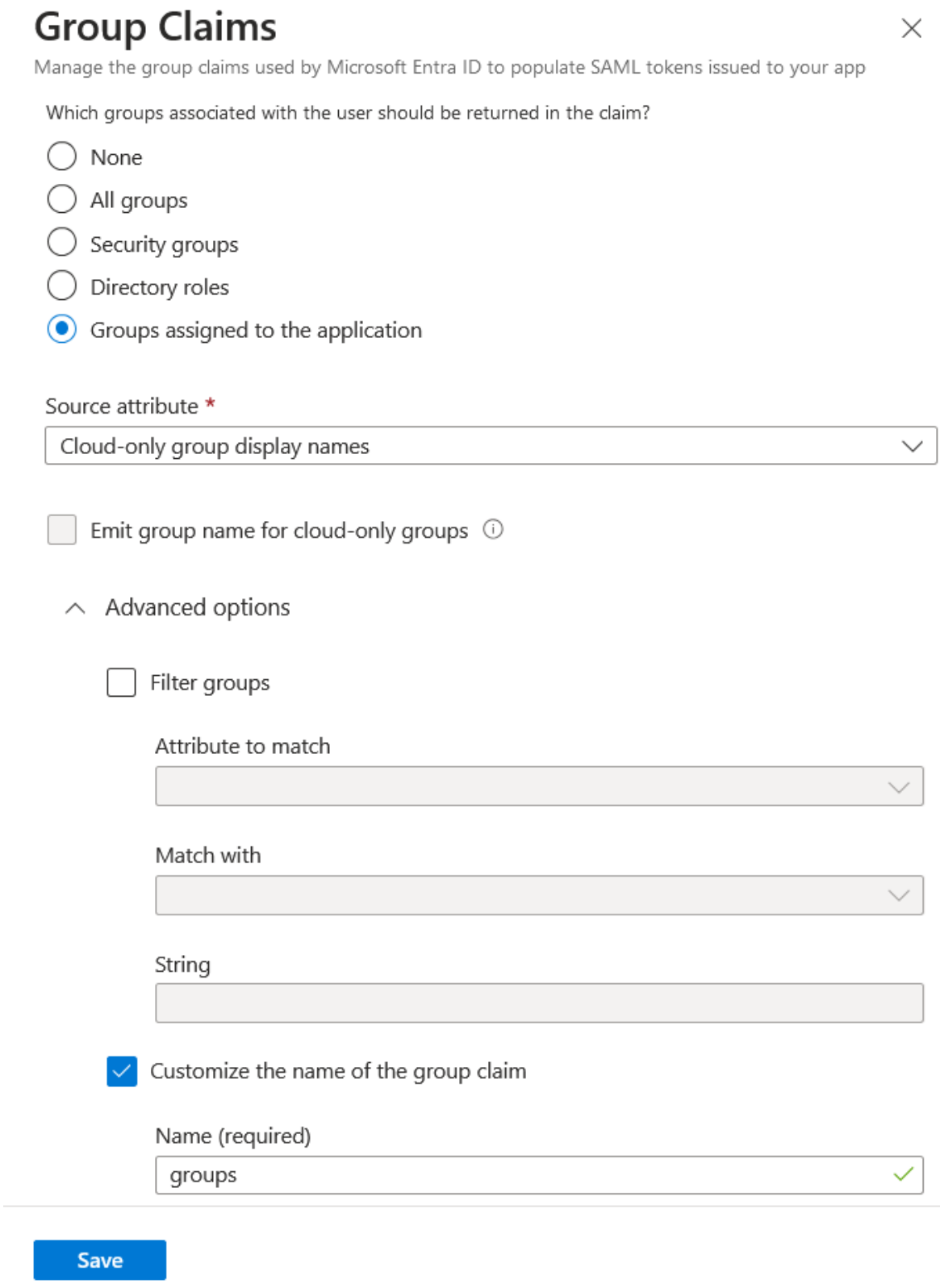
3. *Attributes & Claims* を編集します。Zabbix に渡したいすべての属性（user_name, user_lastname, user_email, user_mobile, groups）を追加する必要があります。

属性名は任意です。異なる属性名を使用することもできますが、Zabbix の SAML 設定の該当フィールド値と一致している必要があります。

- *Add new claim* をクリックして属性を追加します。



- *Add a group claim* をクリックして、グループを Zabbix に渡すための属性を追加します。



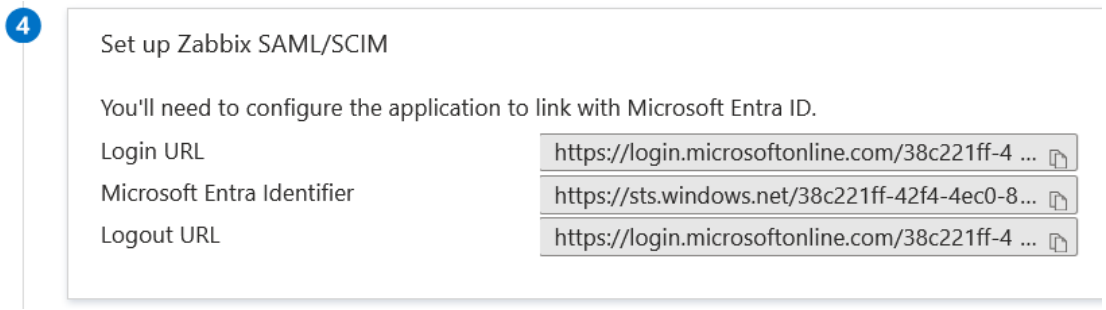
このクレームでは、選択した *Source attribute* によってグループ名（グループ ID ではなく）が Zabbix に渡されることが重要です。そうでない場合、JIT ユーザープロビジョニングが正しく機能しません。

4. *SAML Certificates* で、Entra ID から提供された Base64 証明書をダウンロードし、Zabbix フロントエンドインストールの `conf/certs` に配置します。

以下のコマンドで 644 パーミッションを設定します。

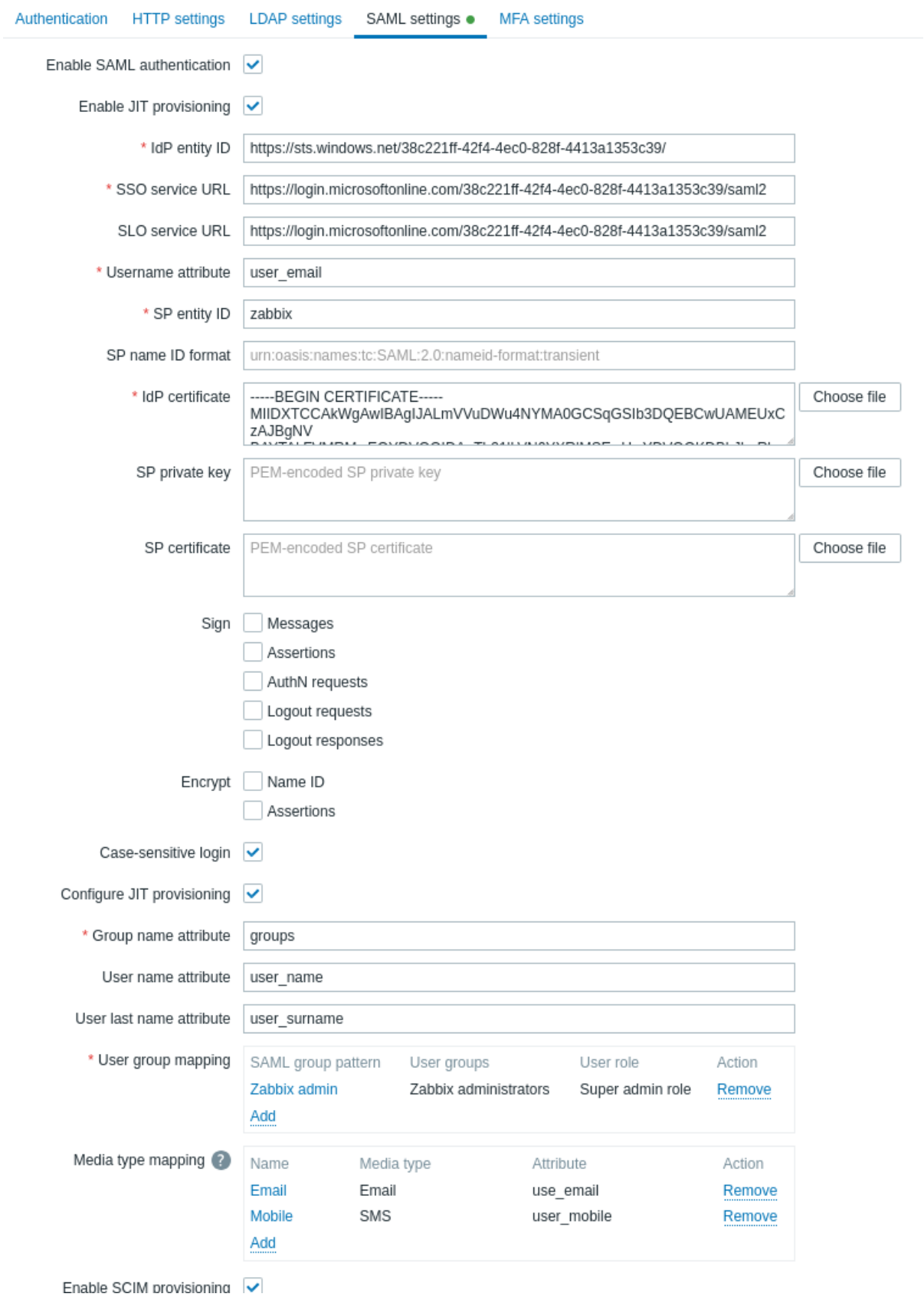
```
chmod 644 entra.cer
```

5. Entra ID の *Set up <your app name>* の値を使用して、Zabbix の SAML 認証を設定します（次のセクションを参照）。



Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Entra ID の設定に基づいて設定オプションを入力します。



Zabbix フィールド	Entra ID の設定フィールド	サンプル値
IdP エンティティ *ID*	Microsoft Entra 識別子	
SSO サービス *URL*	ログイン URL	
SLO サービス *URL*	ログアウト URL	
SP エンティティ *ID*	識別子（エンティティ ID）	
ユーザー名属性	カスタム属性（クレーム）	`user_email`
グループ名属性	カスタム属性（クレーム）	`groups`
ユーザー名属性	カスタム属性（クレーム）	`user_name`

Zabbix フィールド	Entra ID の設定フィールド	サンプル値
ユーザー姓属性	カスタム属性（クレーム）	`user_lastname`

2. Entra ID から提供された Base64 証明書を追加します。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'database'`が設定されている場合、SAML 設定時にフロントエンドで証明書テキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルは不要です。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'file'`が設定されている場合、証明書はファイルシステム上（デフォルトでは ui/conf/certs または zabbix.conf.php で設定されたパス）に配置する必要があり、フロントエンドは証明書をデータベースに保存しません。
 `$SSO['CERT_STORAGE']` が設定されていない、またはコメントアウトされている場合は、ファイルストレージが想定され、証明書は ui/conf/certs から読み込まれます。

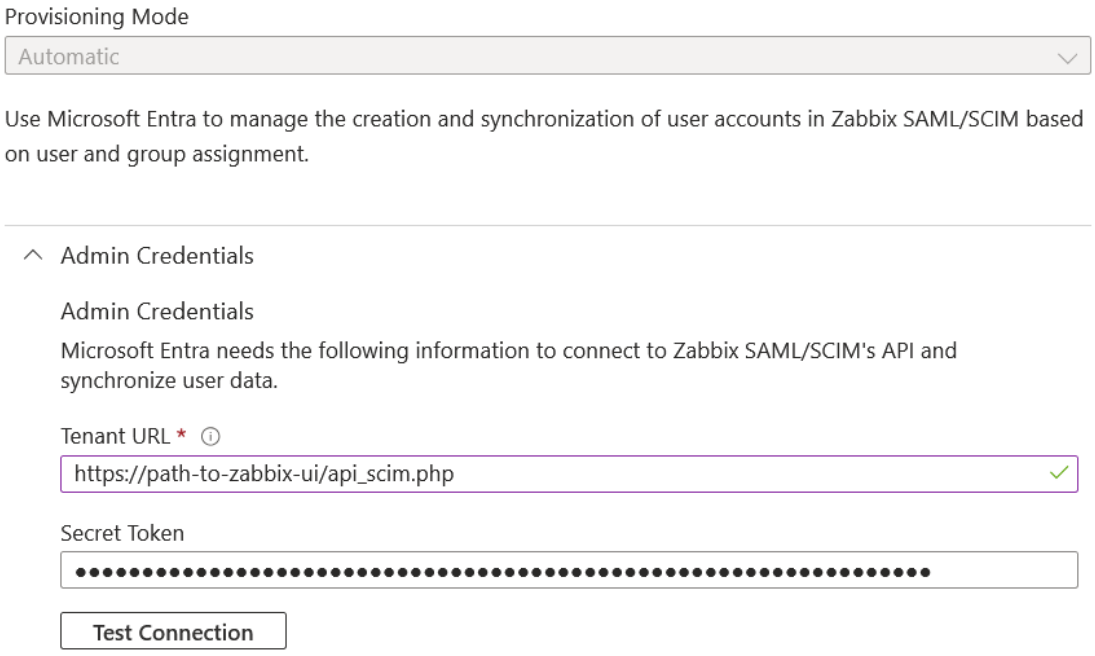
ユーザーグループのマッピングも設定する必要があります。
 メディアのマッピングはオプションです。

3. 更新ボタンを押してこれらの設定を保存します。

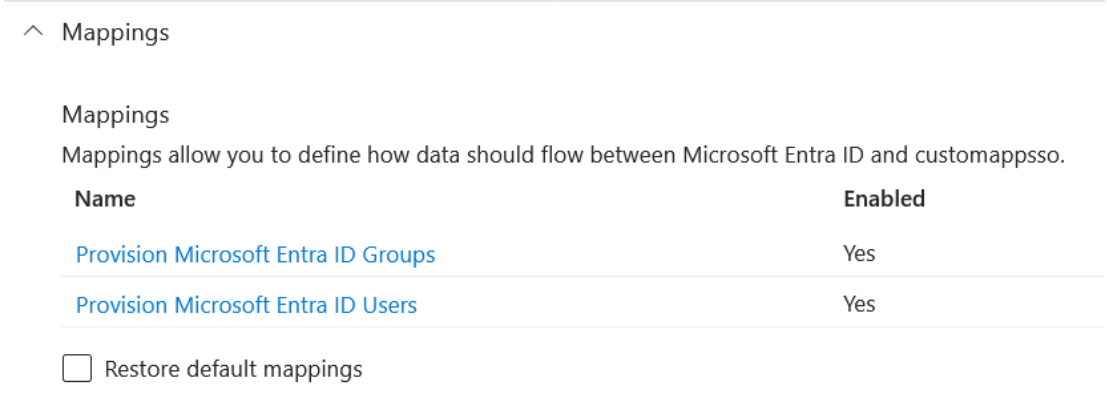
SCIM ユーザーのプロビジョニング

1. Entra ID アプリケーションページのメインメニューから、プロビジョニングページを開きます。*Get started* をクリックし、自動プロビジョニングモードを選択します:

- *Tenant URL* には、次の値を設定します: `https://<path-to-zabbix-ui>/api_scim.php`
- *Secret token* には、スーパ管理者権限を持つ Zabbix API トークンを入力します。
- *Test connection* をクリックして、接続が確立されているか確認します。

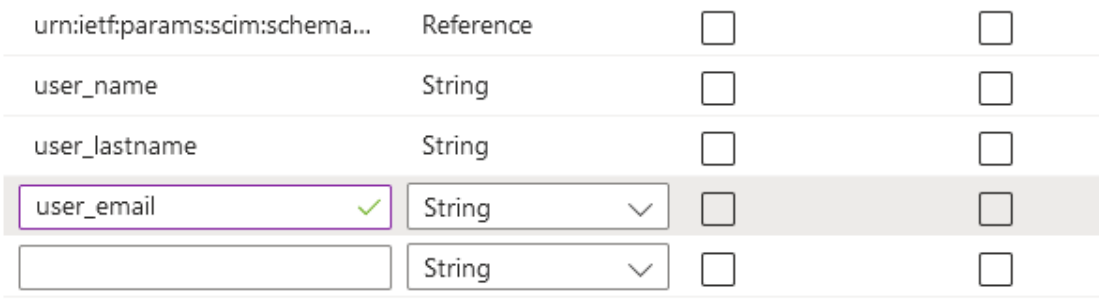


2. これで、SCIM で Zabbix に渡されるすべての属性を追加できます。そのためには、*Mappings* をクリックし、*Provision Microsoft Entra ID Users* をクリックします。



属性マッピングリストの一番下で *Show advanced options* を有効にし、*Edit attribute list for customappsso* をクリックします。

属性リストの一番下に、タイプ「String」で独自の属性を追加します:



リストを保存します。

3. 追加した属性のマッピングを追加できます。属性マッピングリストの一番下で *Add New Mapping* をクリックし、以下のようにマッピングを作成します:



すべてのマッピングを追加したら、マッピングリストを保存します。



4. Zabbix へのユーザープロビジョニングの前提条件として、Entra ID でユーザーとグループが設定されている必要があります。

そのためには、*Microsoft Entra admin center* に移動し、該当するユーザーおよびグループページでユーザー/グループを追加します。

5. Entra ID でユーザーとグループが作成されたら、アプリケーションの *Users and groups* メニューに移動し、アプリに追加できます。

6. アプリの *Provisioning* メニューに移動し、*Start provisioning* をクリックして、ユーザーを Zabbix にプロビジョニングします。

Entra ID の Users PATCH リクエストはメディアの変更をサポートしていないことに注意してください。

認証リクエストの署名

Entra ID を構成して、署名付き認証リクエストの署名を検証することができます。

これを機能させるには、公開鍵/秘密鍵を作成します:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

権限を割り当てます:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Zabbix フロントエンドの設定を更新し、以下を追加します:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

トラブルシューティング

ユーザーが SAML 経由で Zabbix にログインしようとした際、すでに MicroSoft Edge プロファイルでログインしている場合、MicroSoft Edge ブラウザで認証の問題が発生することがあります。この問題の兆候として、ユーザーが MicroSoft Edge のプライベートモードを使用して Zabbix にログインできる場合があります。

この場合の認証問題を回避するには、Zabbix フロントエンドの設定ファイル（*zabbix.conf.php*）で `requestedAuthnContext` を”false” に設定する必要がある場合があります。

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 Okta による **SAML** のセットアップ

このセクションでは、Oktaを設定して、Zabbix の SAML 2.0 認証とユーザーのプロビジョニングを有効にするためのガイドラインを提供します。

Okta の設定

1. https://developer.okta.com/signup/ にアクセスし、アカウントを登録またはサインインします。

2. Okta の Web インターフェースで、*Applications* → *Applications* に移動します。

3. *Create App Integration* をクリックします。

サインイン方法として「SAML 2.0」を選択し、*Next* をクリックします。

4. 一般設定でアプリ名を入力し、*Next* をクリックします。

5. SAML 設定で、以下の値を入力し、*Next* をクリックします。

- **General** に以下を追加します:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs`
    リクエストで `acs` パラメータが切り捨てられないよう、「https」ではなく「http」を使用していることに注意してください。*Use this for Recipient URL and Destination URL* のチェックボックスもマークしてください。
  - *Audience URI (SP Entity ID)*: `zabbix`
    この値は SAML アサーション内で一意のサービスプロバイダー識別子として使用されます（一致しない場合、操作は拒否されます）。このフィールドには URL や任意の文字列を指定できます。
  - *Default RelayState*:
    このフィールドは空白のままにしてください。カスタムリダイレクトが必要な場合は、Zabbix のユーザー > ユーザー設定で追加できます。
  - その他のフィールドは好みに応じて入力してください。
- **Attribute Statements/Group Attribute Statements** に以下を追加します:

これらの属性ステートメントは、Zabbix と共有される SAML アサーションに挿入されます。

ここで使用されている属性名は任意の例です。異なる属性名を使用することもできますが、Zabbix の SAML 設定の各フィールド値と一致している必要があります。

JIT ユーザープロビジョニングなしで Zabbix への SAML サインインを設定したい場合は、email 属性のみが必要です。

> **Note:**
> 暗号化接続を使用する場合は、秘密鍵と公開暗号化証明書を生成し、公開証明書を Okta にアップロードしてください。*Assertion Encryption* を「Encrypted」に設定すると（このパラメータは *Show Advanced Settings* をクリックすると表示されます）、証明書アップロードフォームが表示されます。

6. 次のタブで「I’m a software vendor. I’d like to integrate my app with Okta」を選択し、*Finish* をクリックします。

7. 新しく作成したアプリケーションの「Assignments」タブに移動し、*Assign* ボタンをクリックして、ドロップダウンから「Assign to People」を選択します。

8. 表示されたポップアップで、SAML 2.0 を使用して Zabbix に認証するユーザーにアプリを割り当て、*Save and go back* をクリックします。

9. IdP 証明書をダウンロードします。以下のコマンドを実行して 644 パーミッションを設定します:

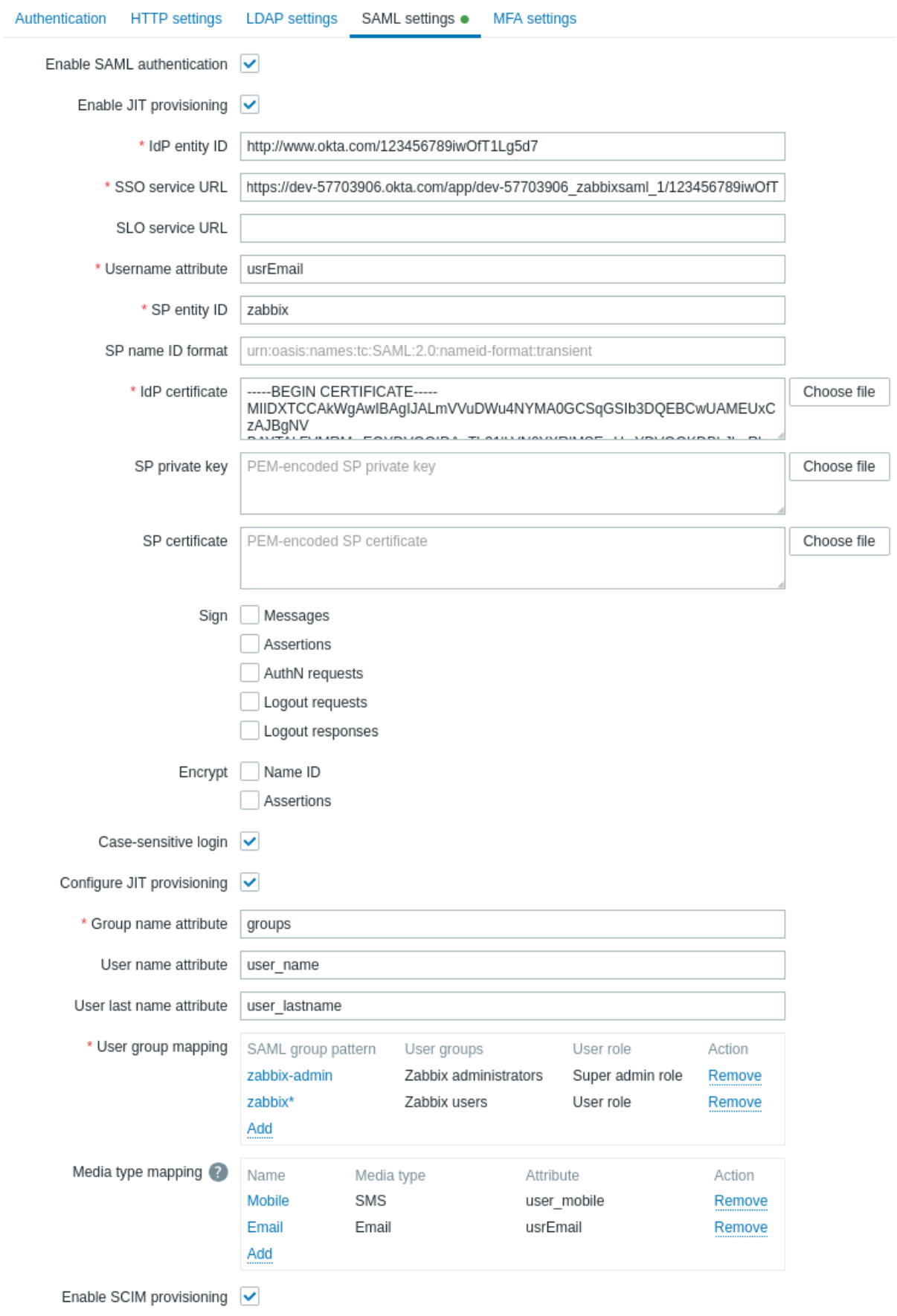
```
chmod 644 idp.crt
```

10. 「Sign On」タブに移動し、*View Setup Instructions* ボタンをクリックします。

セットアップ手順が新しいタブで開きます。Zabbix の設定中はこのタブを開いたままにしてください。

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Okta のセットアップ手順に基づいて設定オプションを入力します。



Zabbix フィールド	Okta のセットアップフィールド	サンプル値
IdP エンティティ ID	Identity Provider Issuer	
SSO サービス URL	Identity Provider Single Sign-On URL	
ユーザー名属性	Attribute name	`usrEmail`
SP エンティティ ID	Audience URI	`zabbix`
グループ名属性	Attribute name	`groups`

Zabbix フィールド	Okta のセットアップフィールド	サンプル値
ユーザー名属性	Attribute name	`user_name`
ユーザー姓属性	Attribute name	`user_lastname`

ユーザーグループとメディアのマッピングも設定する必要があります。

2. Okta の SAML セットアップ手順で提供された Base64 証明書を追加します。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'database'`が設定されている場合、SAML 設定時にフロントエンドで証明書テキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルは必要ありません。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'file'`が設定されている場合、証明書はファイルシステム上（デフォルトでは ui/conf/certs、または zabbix.conf.php で設定されたパス）に配置する必要があり、フロントエンドは証明書をデータベースに保存しません。`$SSO['CERT_STORAGE']` が設定されていない、またはコメントアウトされている場合は、ファイルストレージが想定され、証明書は ui/conf/certs から読み込まれます。

3. Okta で *Assertion Encryption* が「Encrypted」に設定されている場合、Zabbix でも *Encrypt* パラメータの「Assertions」チェックボックスにチェックを入れる必要があります。

4. 更新ボタンを押して、これらの設定を保存します。

SCIM プロビジョニング

1. SCIM プロビジョニングを有効にするには、Okta のアプリケーションの「General」->「App Settings」に移動します。

Enable SCIM provisioning チェックボックスをオンにします。
その結果、新しい *Provisioning* タブが表示されます。

2. 「Provisioning」タブに移動して SCIM 接続を設定します:

- *SCIM connector base URL* には Zabbix フロントエンドへのパスを指定し、`api_scim.php` を追加します。例: `https://<your-zabbix-url>/zabbix/api_scim.php`
- *Unique identifier field for users*: `email`
- *Authentication mode*: `HTTP header`
- *Authorization* には、スーパ管理者権限を持つ有効な API トークンを入力します

Attention:
認証に問題がある場合は、Authorization ヘッダーの転送を参照してください。

3. *Test Connector Configuration* をクリックして接続をテストします。
すべてが正しければ、成功メッセージが表示されます。

4. 「Provisioning」->「To App」で、次のチェックボックスがオンになっていることを確認します:

- Create Users
- Update User Attributes
- Deactivate Users

これにより、これらのリクエストタイプが Zabbix に送信されるようになります。

5. SAML で定義されているすべての属性が SCIM でも定義されていることを確認します。
「Provisioning」->「To App」で *Go to Profile Editor* をクリックすると、アプリのプロファイルエディタにアクセスできます。

Add Attribute をクリックします。
Display name、*Variable name*、*External name* には SAML 属性名（例: `user_name`）を入力します。

External namespace にはユーザースキーマと同じ値を指定します: `urn:ietf:params:scim:schemas:core:2.0:User`

6. アプリケーションの「Provisioning」->「To App」->「Attribute Mappings」に移動します。
下部の *Show Unmapped Attributes* をクリックします。
新しく追加した属性が表示されます。

7. 追加した各属性をマッピングします。

8. 「Assignments」タブでユーザーを追加します。
事前に *Directory -> People* でユーザーを追加しておく必要があります。
これらの割り当てはすべて Zabbix へのリクエストとして送信されます。

9. 「Push Groups」タブでグループを追加します。
Zabbix の SAML 設定でのユーザーグループマッピングパターンが、ここで指定したグループと一致している必要があります。
一致しない場合、ユーザーは Zabbix に作成できません。

グループメンバーに関する情報は、何らかの変更が行われるたびに送信されます。

14 OneLogin での SAML 設定

概要

このセクションでは、SAML 2.0 認証を使用してOneLoginから Zabbix へのシングルサインオンおよびユーザーのプロビジョニングを設定するためのガイドラインを提供します。

OneLogin の設定

アプリケーションの作成

1. OneLogin のアカウントにログインします。テスト目的の場合は、OneLogin で無料の開発者アカウントを作成できます。

2. OneLogin の Web インターフェースで、*Applications* → *Applications* に移動します。

3. 「Add App」をクリックし、適切なアプリを検索します。このページのガイドラインは、*SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)* アプリの例に基づいています。

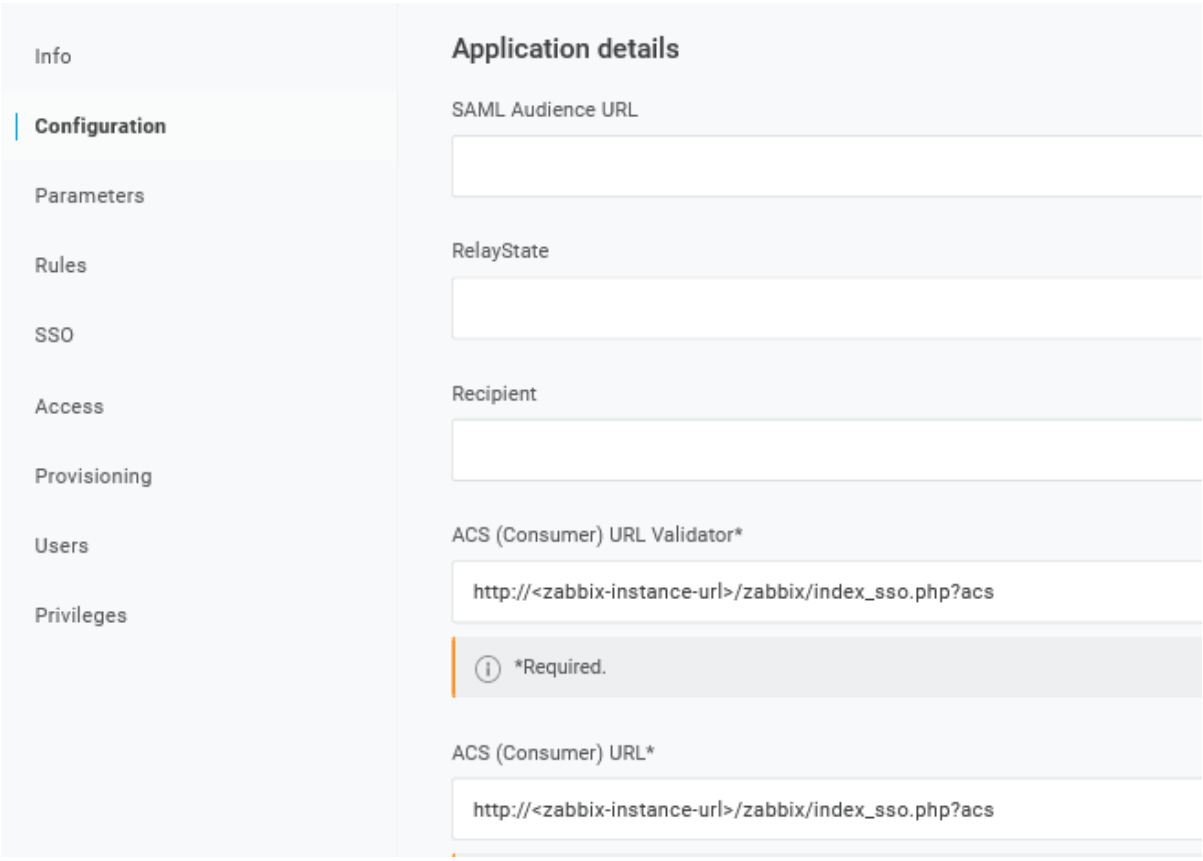
4. まず、アプリの表示名をカスタマイズすることをお勧めします。アイコンやアプリの詳細を追加することもできます。その後、*Save* をクリックします。

SSO/SCIM プロビジョニングの設定

1. *Configuration* -> *Application details* で、Zabbix シングルサインオンエンドポイント `http://<zabbix-instance-url>/zabbix/index_s` を以下のフィールドの値として設定します:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

リクエストで `acs` パラメータが切り取られないように、”https” ではなく” http” を使用していることに注意してください。



” https” を使用することも可能です。Zabbix でこれを機能させるには、`conf/zabbix.conf.php` に以下の行を追加する必要があります:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

他のオプションはデフォルト値のままにします。

2. *Configuration* -> *API connection* で、以下の値を設定します:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`

- *SCIM JSON Template*: SCIM 経由で Zabbix に渡したいすべてのカスタム属性（`user_name`、`user_lastname`、`user_email`、`user_mobile` など）を含める必要があります:

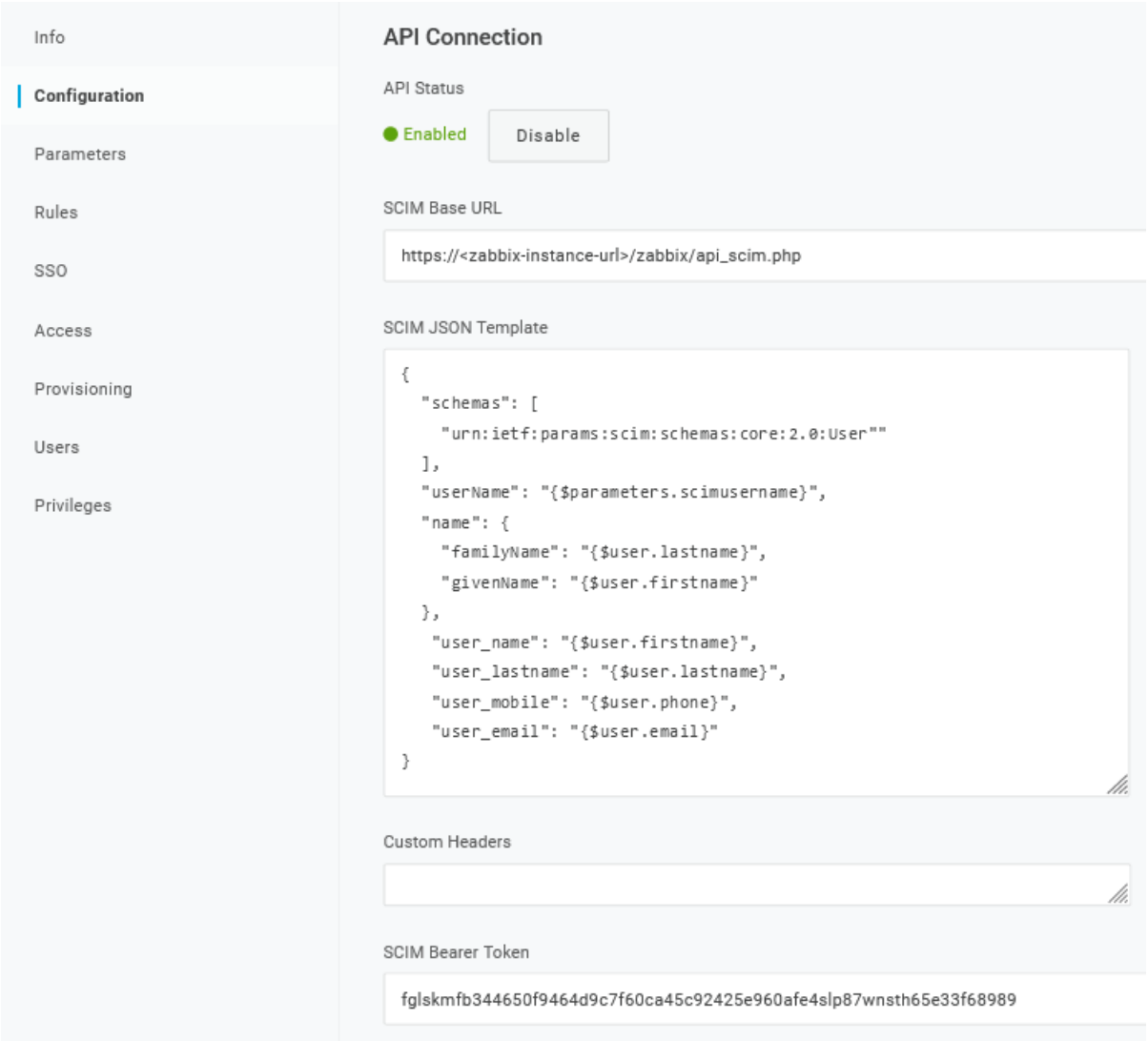
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

属性名は任意です。異なる属性名を使用することもできますが、Zabbix の SAML 設定の該当フィールド値と一致する必要があります。

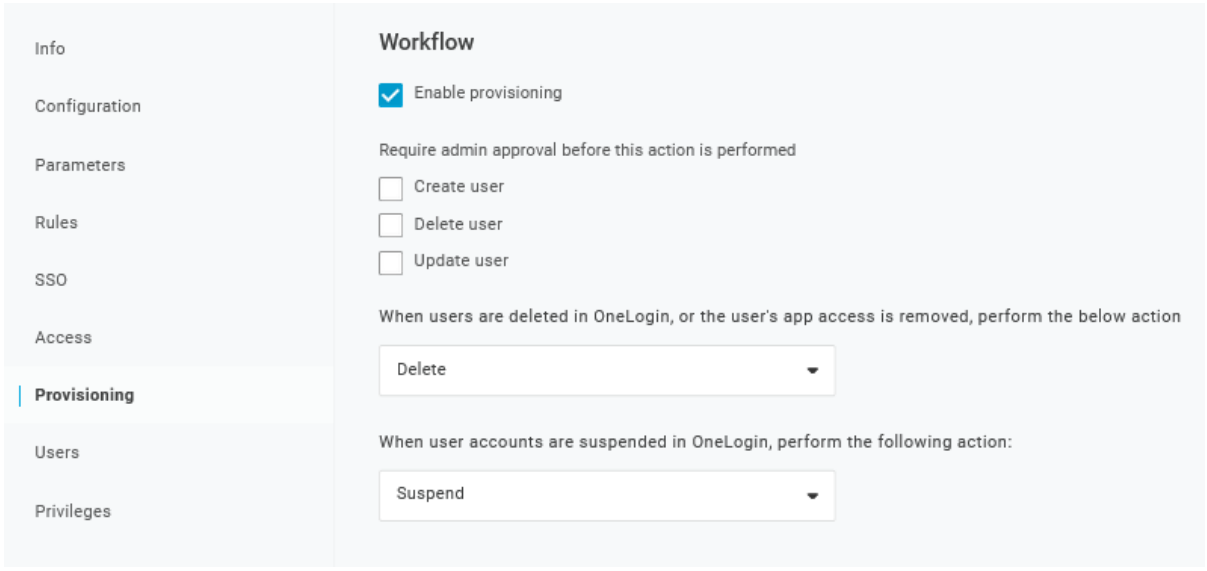
ユーザープロビジョニングを機能させるには、OneLogin がレスポンスで'name' 属性（'givenName' と'familyName' を含む）を受け取る必要があることに注意してください。これはサービスプロバイダーによって必須でなくても必要です。したがって、アプリケーション設定部分のスキーマでこれを指定する必要があります。

- *SCIM Bearer Token*: Super admin 権限を持つ Zabbix API トークンを入力します。

Enable をクリックして接続を有効にします。

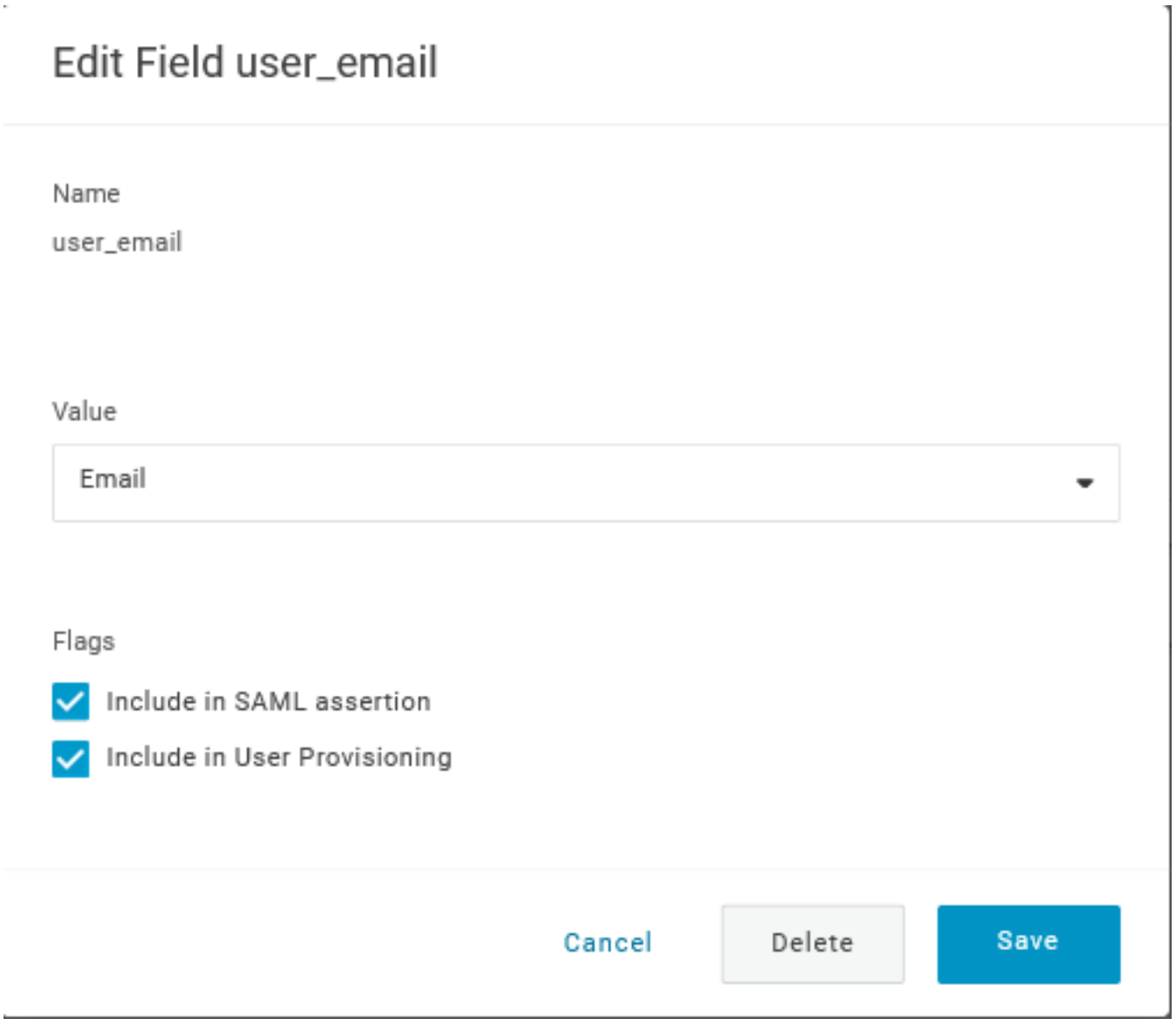


3. *Provisioning* ページで、Provisioning オプションを有効にします:



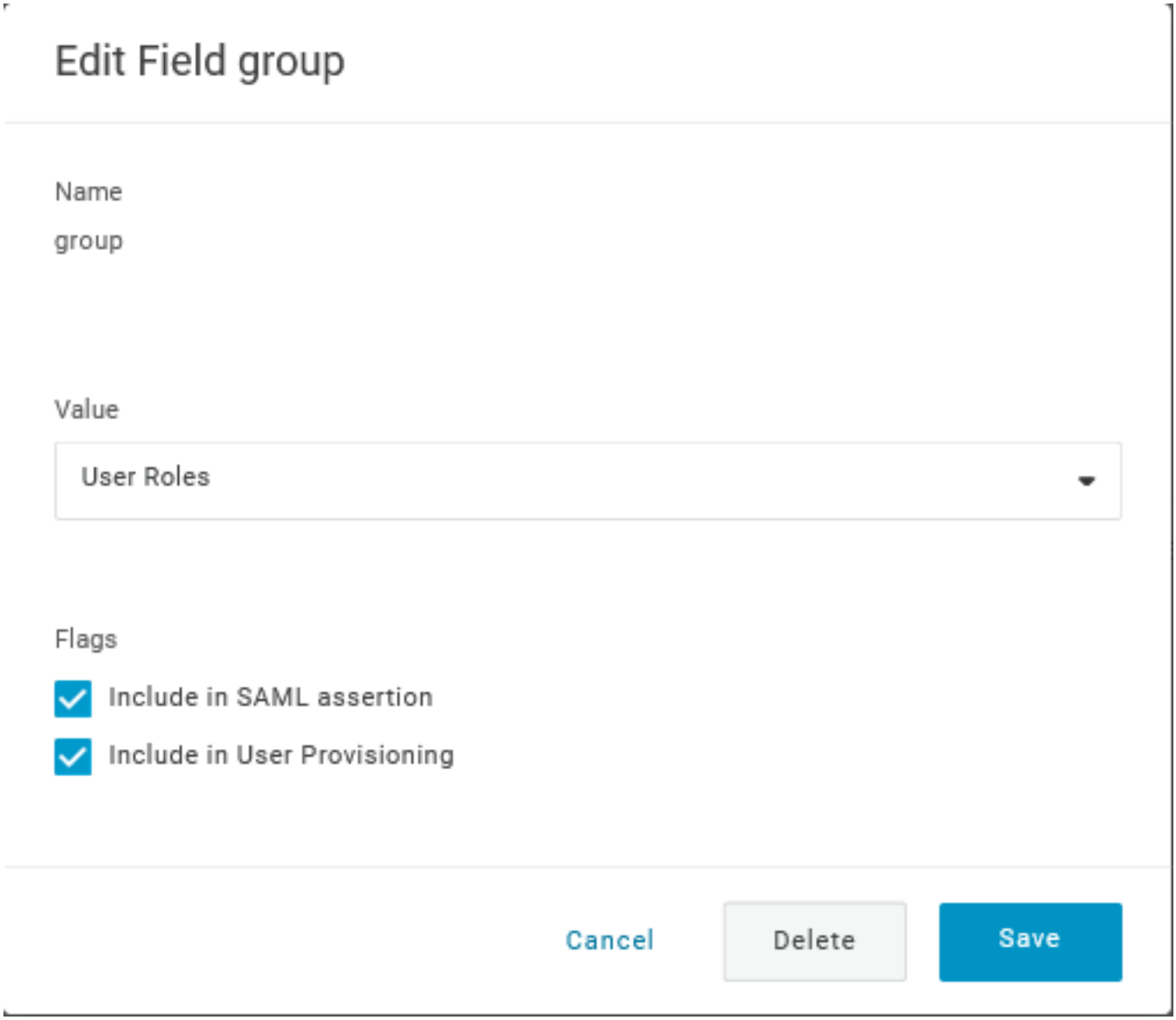
4. *Parameters* ページにはデフォルトパラメータのリストがあります:

   - 'scimusername' が OneLogin のユーザーログイン値（例: メールアドレス）と一致していることを確認します;
   - 'Groups' パラメータの *Include in User Provisioning* オプションにチェックを入れます;
   - "+" をクリックして、`user_name`、`user_lastname`、`user_email`、`user_mobile` など、SAML アサーションやユーザープロビジョニングに必要なカスタムパラメータを作成します:

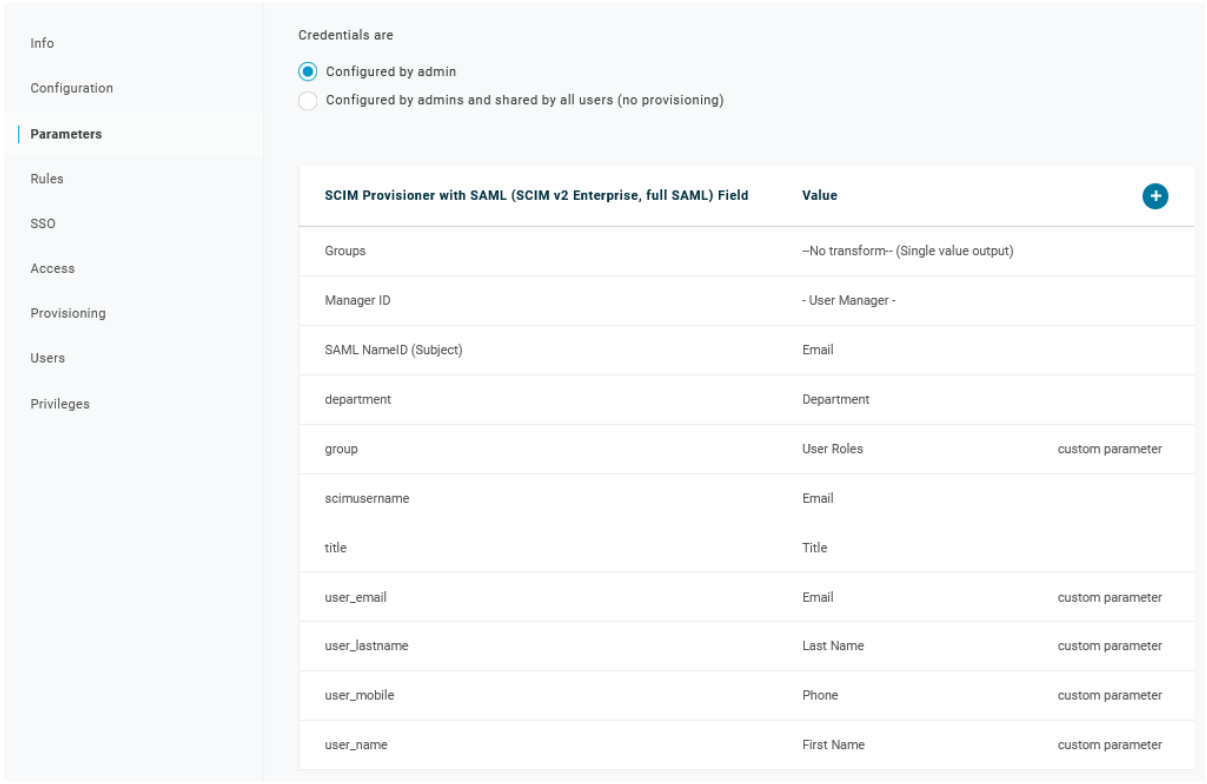


パラメータを追加する際は、*Include in SAML assertion* と *Include in User Provisioning* の両方にチェックを入れてください。

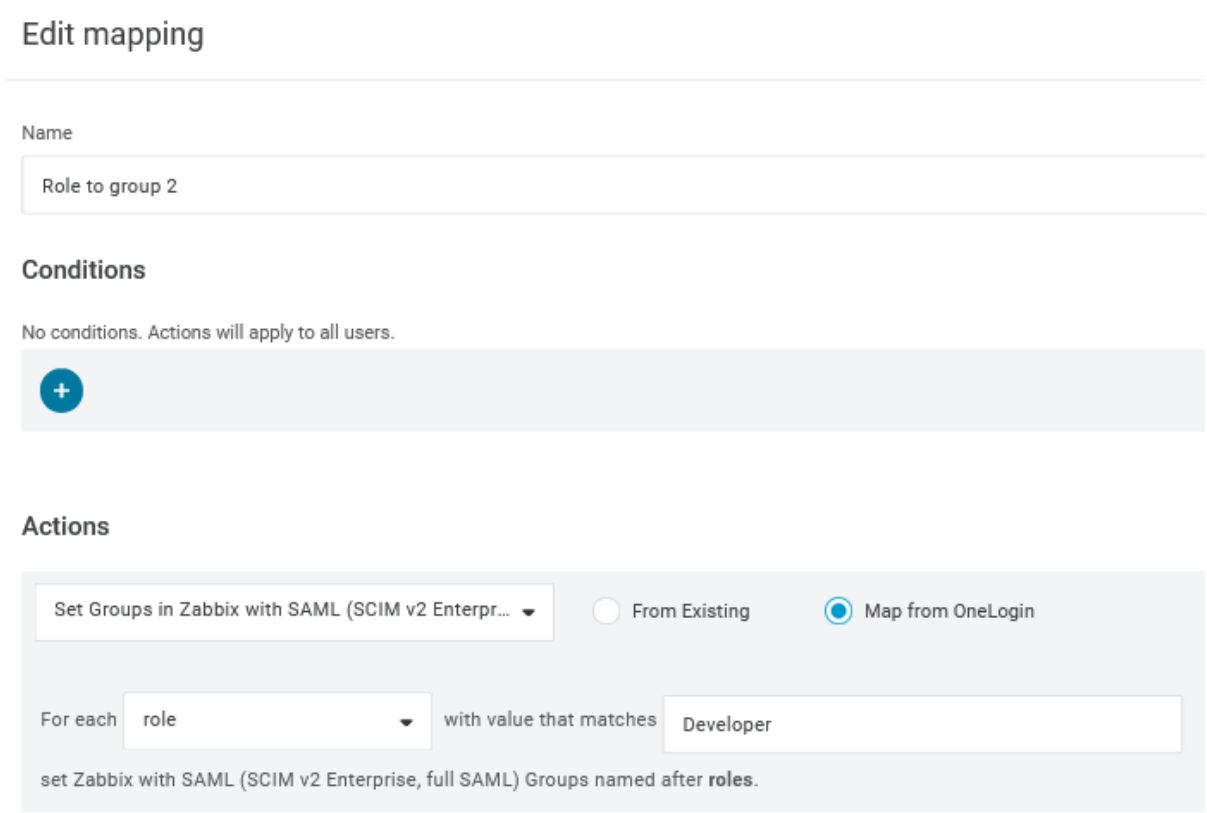
- OneLogin のユーザーロールに一致する'group' パラメータを追加します。ユーザーロールはセミコロン；で区切られた文字列として渡されます。OneLogin のユーザーロールは、Zabbix でユーザーグループを作成する際に使用されます:



パラメータのリストを確認します:



5. *Rules* ページで、デフォルトの Groups パラメータにユーザーロールのマッピングを作成します。



特定のロールをグループとして渡すために正規表現を使用することができます。OneLogin は複数のロールを含む属性を送信する際にセミコロン;を区切り文字として使用するため、ロール名に;を含めないでください。

6. IdP 証明書をダウンロードします。以下のコマンドで 644 パーミッションを設定します:

```
chmod 644 idp.crt
```

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、OneLogin の設定に基づいて設定オプションを入力します。



Zabbix フィールド	OneLogin の設定フィールド	サンプル値
IdP エンティティ *ID*	Issuer URL (OneLogin のアプリケーションの SSO タブを参照)	
SSO サービス *URL*	SAML 2.0 Endpoint (HTTP) (OneLogin のアプリケーションの SSO タブを参照)	

Zabbix フィールド	OneLogin の設定フィールド	サンプル値
SLO サービス URL	SLO Endpoint (HTTP) (OneLogin のアプリケーションの SSO タブを参照)	
ユーザー名属性	カスタムパラメータ	`user_email`
グループ名属性	カスタムパラメータ	`group`
ユーザー名属性	カスタムパラメータ	`user_name`
ユーザー姓属性	カスタムパラメータ	`user_lastname`

ユーザーグループのマッピングも設定する必要があります。メディアマッピングはオプションです。更新をクリックしてこれらの設定を保存します。

2. OneLogin から提供された Base64 証明書を追加します。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'database'`が設定されている場合、SAML 設定時にフロントエンドで証明書テキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルは必要ありません。

zabbix.conf.php で `$SSO['CERT_STORAGE'] = 'file'`が設定されている場合、証明書はファイルシステム上（デフォルトでは ui/conf/certs または zabbix.conf.php で設定されたパス）に配置する必要があり、フロントエンドは証明書をデータベースに保存しません。`$SSO['CERT_STORAGE']` が設定されていないかコメントアウトされている場合は、ファイルストレージが想定され、証明書は ui/conf/certs から読み込まれます。

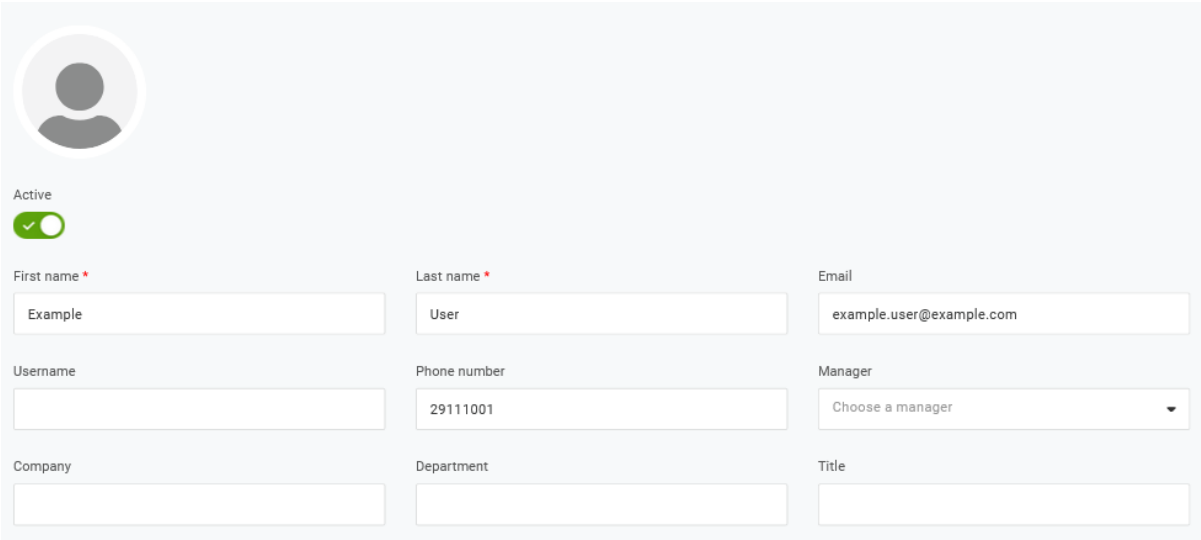
OneLogin で証明書をダウンロードするには、*Applications* -> *SSO* -> 現在の証明書の *View details* をクリックします。

3. 更新ボタンを押してこれらの設定を保存します。

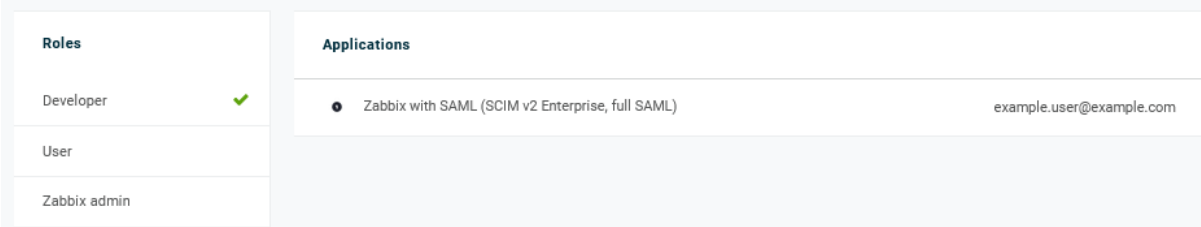
SCIM ユーザーのプロビジョニング

ユーザーのプロビジョニングを有効にすると、OneLogin でユーザーとそのロールを追加/更新し、それらを即座に Zabbix にプロビジョニングすることが可能になります。

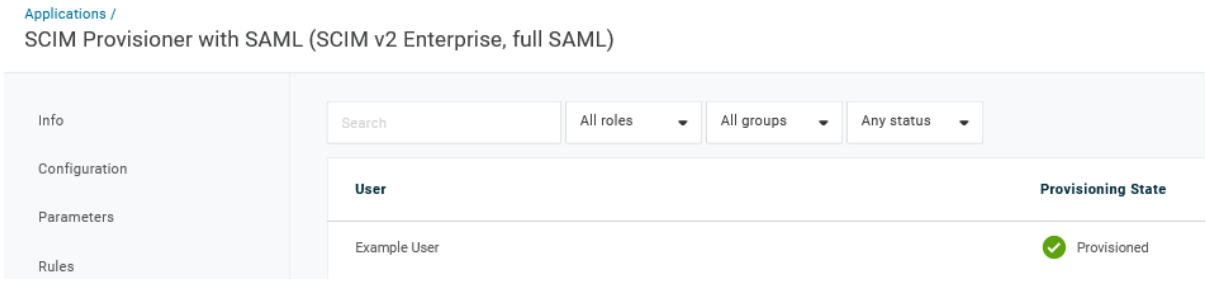
例えば、新しいユーザーを作成できます。



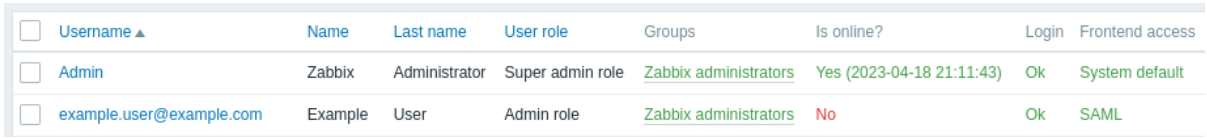
ユーザーロールとユーザーをプロビジョニングするアプリケーションに追加します。



ユーザーを保存すると、Zabbix にプロビジョニングされます。[Application] -> [Users] で、現在のアプリケーションユーザーのプロビジョニング状況を確認できます。



正常にプロビジョニングされると、Zabbix のユーザーリストにユーザーが表示されます。



15 定期レポートの設定

概要

このセクションでは、Zabbix Web サービスのインストール方法と、スケジュールレポートの生成を有効にするための Zabbix の設定方法について説明します。

インストール

スケジュールレポートの生成を有効にするには、新しいZabbix Web サービスプロセスとGoogle Chrome ブラウザをインストールする必要があります。Web サービスは、Zabbix サーバーがインストールされているマシンと同じマシン、または別のマシンにインストールできます。Google Chrome ブラウザは、Web サービスがインストールされているのと同じマシンにインストールする必要があります。

公式の zabbix-web-service パッケージは、Zabbix リポジトリで入手できます。Google Chrome ブラウザはこれらのパッケージには含まれていないため、別途インストールする必要があります。

Zabbix Web サービスをソースからコンパイルするには、Zabbix Web サービスのインストールを参照してください。

インストール後、Web サービスがインストールされているマシンで zabbix_web_service を実行します。

```
zabbix_web_service
```

設定

すべての要素間で適切な通信を行うために、サーバーの設定ファイルとフロントエンドの設定パラメータが正しく設定されていることを確認してください。

Zabbix サーバー

Zabbix サーバーの設定ファイルで、*WebServiceURL* と *StartReportWriters* のパラメータを更新する必要があります。

WebServiceURL

このパラメータは、Web サービスとの通信を有効にするために必要です。URL は `http[s]://host:port/report` の形式で指定します。

- デフォルトでは、Web サービスはポート 10053 で待ち受けます。Web サービスの設定ファイルで異なるポートを指定できます。
- `/report` パスの指定は必須です（パスはハードコードされており、変更できません）。

例:

```
WebServiceURL=http://localhost:10053/report
```

> **Attention:**
> Zabbix サーバーと Zabbix Web サービス間の通信には、証明書を使用した暗号化を設定することを強く推奨します。デフォルトでは、Zabbix サーバーと Zabbix Web サービス間で送信されるデータは暗号化されていないため、不正アクセスにつながる可能性があります。

StartReportWriters

このパラメータは、起動するレポートライタープロセスの数を決定します。設定しない場合や 0 の場合は、レポート生成が無効になります。必要なレポートの数や頻度に応じて、1～100 のレポートライタープロセスを有効にできます。

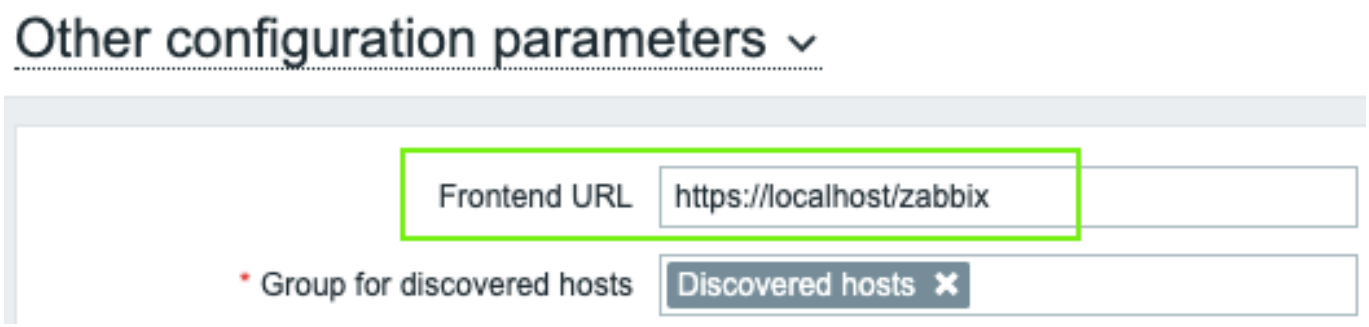
例:

```
StartReportWriters=3
```

Zabbix フロントエンド

Zabbix フロントエンドと Zabbix Web サービス間の通信を有効にするには、フロントエンド *URL* パラメータを設定する必要があります。

- フロントエンドの管理 > 一般 > その他メニューセクションに進みます
- フロントエンド *URL* パラメータに Zabbix Web インターフェースの完全な URL を指定します



Note:
セットアップ手順が完了したら、テストレポートを設定して送信し、すべてが正しく動作することを確認することをお勧めします。

16 追加のフロントエンド言語

概要

Zabbix Web インターフェースで英語以外の言語を使用するには、そのロケールを Web サーバーにインストールする必要があります。さらに翻訳を機能させるには、PHP の gettext 拡張機能が必要です。

ロケールのインストール

インストールされているすべての言語のリストを表示するには、次のコマンドを実行します。

```
locale -a
```

必要な言語がリストにない場合は*/etc/locale.gen* ファイルを開き、必要なロケールのコメントを解除します。Zabbix は UTF-8 エンコーディングを使用するため、UTF-8 文字セットを使用してロケールを選択する必要があります。

次に以下を実行します。

```
locale-gen
```

Web サーバーを再起動してください。

これでロケールがインストールされました。新しい言語を表示するには、Ctrl+F5 でブラウザ上の Zabbix Web インターフェースページを再読み込みする必要がある場合があります。

Zabbix のインストール

Zabbix をZabbix git リポジトリから直接インストールする場合は、翻訳ファイルを手動で生成する必要があります。翻訳ファイルを生成するには、以下を実行します。

```
make gettext
locale/make_mo.sh
```

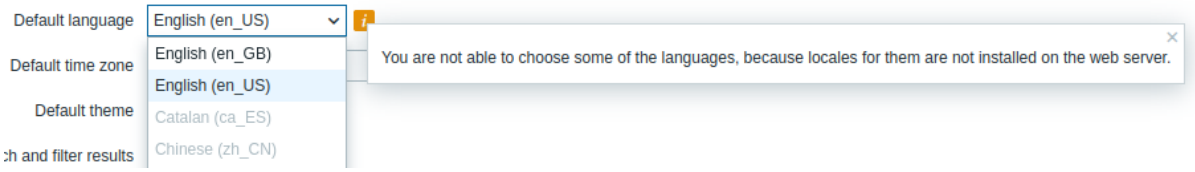
Zabbix をパッケージまたはソース tar.gz ファイルからインストールする場合、この手順は必要ありません。

言語選択

Zabbix Web インターフェースで言語を選択する方法はいくつかあります。

- Web インターフェースをインストールする時 - フロントエンドインストールウィザードで選択した言語がシステムデフォルトとして設定されます。
- インストール後、システムのデフォルト言語は管理 → 一般 → *GUI* のメニューセクションで変更できます。
- 特定のユーザーの言語はユーザープロファイルで変更できます。

対象の言語ロケールがマシンにインストールされていない場合、その言語は Zabbix 言語セレクターでグレー表示されます。少なくとも 1 つのロケールが欠落している場合、言語セレクターの横に赤いアイコンが表示されます。このアイコンを押すと、次のメッセージが表示されます: ”Web サーバーにロケールがインストールされていないため、いくつかの言語は利用できません。”



2 プロセスの設定

サイドバーから、参照したい項目を選択してください。

1 Zabbix サーバー

概要

Zabbix サーバーの設定ファイル（zabbix_server.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしでリストされています。パラメータをクリックすると詳細が表示されます。

パラメータ	説明
AlertScriptsPath	カスタムアラートスクリプトの場所。
AllowRoot	サーバーを「root」として実行することを許可します。
AllowSoftwareUpdateCheck	Zabbix UI が zabbix.com からソフトウェアアップデート情報を受信することを許可します。
AllowUnsupportedDBVersions	サポートされていないデータベースバージョンでのサーバーの動作を許可します。
CacheSize	設定キャッシュのサイズ。
CacheUpdateFrequency	設定キャッシュの更新を何秒ごとに実行するかを決定します。
DBHost	データベースホスト名。
DBName	データベース名。
DBPassword	データベースパスワード。
DBPort	ローカルソケットを使用しない場合のデータベースポート。
DBSchema	データベーススキーマ名。PostgreSQL で使用されます。
DBSocket	MySQL ソケットファイルへのパス。
DBUser	データベースユーザー。
DBTLSConnect	このオプションを指定した値に設定すると、データベースへの TLS 接続の使用が強制されます。
DBTLSCAFile	データベース証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。
DBTLSCertFile	データベース認証用の Zabbix サーバー証明書を含むファイルのフルパス名。
DBTLSKeyFile	データベース認証用の秘密鍵を含むファイルのフルパス名。
DBTLSCipher	TLS v1.2 までの TLS プロトコルで Zabbix サーバーが許可する暗号化方式のリスト。MySQL のみサポート。
DBTLSCipher13	TLS v1.3 プロトコルで Zabbix サーバーが許可する暗号化スイートのリスト。MySQL 8.0.16 以降のみサポート。
DebugLevel	デバッグレベルを指定します。
EnableGlobalScripts	Zabbix サーバーでグローバルスクリプトを有効にします。
ExportDir	イベント、履歴、トレンドを改行区切りの JSON 形式でリアルタイムエクスポートするディレクトリ。設定するとリアルタイムエクスポートが有効になります。
ExportFileSize	エクスポートファイルごとの最大サイズ（バイト単位）。
ExportType	リアルタイムエクスポートするエンティティタイプ（events, history, trends）のカンマ区切りリスト（デフォルトはすべてのタイプ）。
ExternalScripts	外部スクリプトの場所。
Fping6Location	fping6 の場所。
FpingLocation	fping の場所。
FrontendAllowedIP	フロントエンドからの接続を許可する IP アドレスまたは CIDR 範囲のカンマ区切りリスト。
HANodeName	高可用性クラスタノード名。
HistoryCacheSize	履歴キャッシュのサイズ。
HistoryIndexCacheSize	履歴インデックスキャッシュのサイズ。
HistoryStorageDateIndex	履歴ストレージで履歴値を前処理し、日付に基づいて異なるインデックスに値を保存することを有効にします。
HistoryStorageURL	履歴ストレージの HTTP[S] URL。

パラメータ	説明
HistoryStorageTypes	履歴ストレージに送信する値タイプのカンマ区切りリスト。
HousekeepingFrequency	ハウスキーピング処理を何時間ごとに実行するかを決定します。
Include	設定ファイルに個別ファイルまたはディレクトリ内のすべてのファイルを含めることができます。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。
JavaGatewayPort	Zabbix Java ゲートウェイがリッスンするポート。
ListenBacklog	TCP キュー内の保留中の接続の最大数。
ListenIP	トラッパーがリッスンする IP アドレスのカンマ区切りリスト。
ListenPort	トラッパーのリッスンポート。
LoadModule	サーバー起動時にロードするモジュール。
LoadModulePath	サーバーモジュールの場所へのフルパス。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogSlowQueries	データベースクエリがログに記録されるまでの最大時間（ミリ秒単位）。
LogType	ログ出力のタイプ。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラー、SNMP ポーラーが同時に実行できる非同期チェックの最大数。
MaxHousekeeperDelete	1 つのハウスキーピングサイクルの 1 つのタスクで削除される行数（[tablename], [field], valueに対応）は「MaxHousekeeperDelete」を超えません。
NodeAddress	フロントエンドがサーバーに接続する方法を上書きするための、オプションのポート付き IP またはホスト名。
PidFile	PID ファイル名。
ProblemHousekeepingFrequency	削除されたトリガーの問題を Zabbix が削除する頻度を決定します。
ProxyConfigFrequency	Zabbix サーバーが Zabbix プロキシに設定データを送信する頻度を決定します。
ProxyDataFrequency	Zabbix サーバーが Zabbix プロキシから履歴データを要求する頻度を決定します。
ServiceManagerSyncFrequency	Zabbix がサービスマネージャーの設定を同期する頻度を決定します。
SMSDevices	Zabbix サーバーで使用を許可するモデムファイルのカンマ区切りリスト。
SNMPTrapperFile	SNMP トラップデーモンからサーバーにデータを渡すための一時ファイル。
SocketDir	内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリ。
SourceIP	送信元 IP アドレス。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵および秘密鍵の場所。
SSLCertLocation	クライアント認証用の SSL クライアント証明書ファイルの場所。
SSLKeyLocation	クライアント認証用の SSL 秘密鍵ファイルの場所。
SSLCALocation	SSL サーバー証明書検証用の認証局（CA）ファイルの場所を上書きします。
StartAgentPollers	非同期 Zabbix エージェントポーラーの事前フォークインスタンス数。
StartAlerters	アラートプロセスの事前フォークインスタンス数。
StartBrowserPollers	ブラウザーアイテムポーラーの事前フォークインスタンス数。
StartConnectors	コネクターワーカーの事前フォークインスタンス数。
StartDBSyncers	履歴同期プロセスの事前フォークインスタンス数。
StartDiscoverers	ディスカバリーワーカーの事前フォークインスタンス数。
StartEscalators	エスカレーターの事前フォークインスタンス数。
StartHistoryPollers	履歴ポーラーの事前フォークインスタンス数。
StartHTTPAgentPollers	非同期 HTTP エージェントポーラーの事前フォークインスタンス数。
StartHTTPPollers	HTTP ポーラーの事前フォークインスタンス数。
StartIPMIPollers	IPMI ポーラーの事前フォークインスタンス数。
StartJavaPollers	Java ポーラーの事前フォークインスタンス数。
StartLLDProcessors	低レベルディスカバリ（LLD）ワーカーの事前フォークインスタンス数。
StartODBCPollers	ODBC ポーラーの事前フォークインスタンス数。
StartPingers	ICMP ピンガーの事前フォークインスタンス数。
StartPollersUnreachable	到達不能ホスト用ポーラー（IPMI および Java を含む）の事前フォークインスタンス数。
StartPollers	ポーラーの事前フォークインスタンス数。
StartPreprocessors	前処理ワーカーの事前起動インスタンス数。
StartProxyPollers	パッシブプロキシ用ポーラーの事前フォークインスタンス数。
StartReportWriters	レポートライターの事前フォークインスタンス数。
StartSNMPPollers	非同期 SNMP ポーラーの事前フォークインスタンス数。
StartSNMPTrapper	1 に設定すると、SNMP トラッパープロセスが開始されます。
StartTimers	タイマーの事前フォークインスタンス数。
StartTrappers	トラッパーの事前フォークインスタンス数。
StartVMwareCollectors	VMware コレクターの事前フォークインスタンス数。
StatsAllowedIP	外部 Zabbix インスタンスの IP アドレス（CIDR 表記または DNS 名も可）のカンマ区切りリスト。ここにリストされたアドレスからのみ統計リクエストが受け付けられます。

パラメータ	説明
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）、`icmpping[*]` アイテムの待機時間（秒）を指定します。
TLSCAFile	Zabbix コンポーネント間の暗号化通信でピア証明書検証に使用する最上位 CA 証明書を含むファイルのフルパス名。
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用するサーバー証明書または証明書チェーンを含むファイルのフルパス名。
TLSCipherAll	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。証明書および PSK ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書および PSK ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。証明書ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。PSK ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCRLFile	失効証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信で使用されます。
TLSFrontendAccept	フロントエンドから受け入れる着信接続。
TLSFrontendCertIssuer	許可されたフロントエンド証明書発行者。
TLSFrontendCertSubject	許可されたフロントエンド証明書サブジェクト。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用するサーバー秘密鍵を含むファイルのフルパス名。
TLSListen	トラッパーソケットでの TLS を制御します。
TmpDir	一時ディレクトリ。
TrapperTimeout	以下のタイムアウト（秒）を指定します： - Zabbix プロキシからの履歴データ取得； - Zabbix プロキシへの設定データ送信； - Zabbix サーバーでのグローバルスクリプト実行またはリモートコマンド実行。
TrendCacheSize	トレンドキャッシュのサイズ。
TrendFunctionCacheSize	トレンド関数キャッシュのサイズ。
UnavailableDelay	到達不能期間中にホストの可用性をチェックする頻度を決定します。
UnreachableDelay	到達不能期間中にホストの可用性をチェックする頻度を決定します。
UnreachablePeriod	到達不能状態が何秒続いたらホストを到達不能と見なすかを決定します。
User	システム上の特定の既存ユーザーに権限をドロップします。
ValueCacheSize	履歴値キャッシュのサイズ。
Vault	ボールトプロバイダーを指定します。
VaultDBPath	データベース認証情報をキーで取得する場所を指定します。
VaultPrefix	ボールトパスまたはクエリのカスタムプレフィックス。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイル名。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイル名。
VaultToken	HashiCorp ボールト認証トークン。
VaultURL	ボールトサーバーの HTTP[S] URL。
VMwareCacheSize	VMware データ保存用の共有メモリサイズ。
VMwareFrequency	単一の VMware サービスからデータを収集する間隔（秒単位）。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンタ統計を取得する間隔（秒単位）。
VMwareTimeout	VMware コレクターが VMware サービスからの応答を待機する最大秒数。
WebDriverURL	WebDriver インターフェースの HTTP[S] URL。
WebServiceURL	Zabbix Web サービスへの HTTP[S] URL（形式：<host:port>/report）。

パラメータは、必須であることが明示的に記載されていない限り、すべて必須ではありません。

注意：

- デフォルト値はデーモンのデフォルト値であり、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は、BOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- 「#」で始まるコメントは、行頭のみサポートされます。

パラメータの詳細

AlertScriptsPath

カスタムアラートスクリプトの場所（*datadir* コンパイル時インストール変数に依存）。

デフォルト：`/usr/local/share/zabbix/alertscripts`

AllowRoot

サーバーを'root' として実行することを許可します。無効にした場合、サーバーが'root' によって起動された場合、代わりに'zabbix' ユーザーに切り替えようとします。通常のユーザーで起動した場合は効果がありません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

AllowSoftwareUpdateCheck

Zabbix UI が zabbix.com からソフトウェアアップデート情報を受信することを許可します。

デフォルト: 1
 値: 0 - 許可しない; 1 - 許可する

AllowUnsupportedDBVersions

サーバーがサポートされていないデータベースバージョンで動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

CacheSize

設定キャッシュのサイズ（バイト単位）。ホスト、アイテム、トリガーのデータを格納する共有メモリのサイズ。

デフォルト: 32M
 範囲: 128K-64G

CacheUpdateFrequency

このパラメータは、Zabbix が構成キャッシュの更新を何秒ごとに実行するかを決定します。ランタイム制御オプションも参照してください。

デフォルト：10
 範囲：1-3600

DBHost

データベースのホスト名。
MySQL の場合、`localhost` または空文字列はソケットの使用を意味します。PostgreSQL の場合、空文字列のみがソケットの使用を試みます。

デフォルト: `localhost`

DBName

データベース名。

必須: はい

DBPassword

データベースのパスワード。パスワードを使用しない場合は、この行をコメントアウトしてください。

DBPort

ローカルソケットを使用しない場合のデータベースポート。[3]

MySQL のデフォルト: 3306

PostgreSQL のデフォルト: 5432

範囲: 1024-65535

DBSchema

データベーススキーマ名。
PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパス。[3]

DBUser

データベースユーザー。

DBTLSConnect

このオプションを次の値に設定すると、データベースへの TLS 接続が強制されます:
required - TLS を使用して接続
verify_ca - TLS を使用して接続し、証明書を検証
verify_full - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベースの識別子が証明書と一致することを検証

MySQL 5.7.11 以降および PostgreSQL では、`required`、

verify_ca、verify_full の値がサポートされています。
MariaDB 10.2.6 以降では、required および verify_full の値がサポートされています。
 デフォルトではどのオプションにも設定されておらず、動作はデータベースの設定に依存します。

DBTLSCAFile

データベース証明書の検証のためのトップレベル CA 証明書を含むファイルのフルパス名。

必須: いいえ（DBTLSConnect が *verify_ca* または *verify_full* に設定されている場合は「はい」）

DBTLSCertFile

データベースへの認証のための Zabbix サーバー証明書を含むファイルのフルパス名。

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を含むファイルのフルパス名。

DBTLSCipher

Zabbix サーバーが TLS v1.2 までの TLS プロトコルで許可する暗号化アルゴリズムのリストです。MySQL のみサポートされています。

DBTLSCipher13

Zabbix サーバーが TLS v1.3 プロトコルで許可する暗号スイートのリストです。MySQL のみサポートされており、バージョン 8.0.16 以降で利用可能です。

DebugLevel

デバッグレベルを指定します:
0 - Zabbix プロセスの開始と停止に関する基本情報
1 - 重大な情報;
2 - エラー情報;
3 - 警告;
4 - デバッグ用 (大量の情報を出力);
5 - 拡張デバッグ (さらに多くの情報を出力)。
 また、ランタイム制御オプションも参照してください。

デフォルト: 3
 範囲: 0-5

EnableGlobalScripts

Zabbix サーバーでグローバルスクリプトを有効にします。
 注: グローバルスクリプトの実行はデフォルトで有効になっています。ただし、Zabbix 7.0 以降の新規インストールでは、EnableGlobalScripts が明示的に 0（無効）に設定されています。

デフォルト: 1
 値: 0 - 無効; 1 - 有効

ExportDir

イベント、ヒストリ、トレンドのリアルタイムエクスポートを改行区切りの JSON 形式で出力するディレクトリ。設定すると、リアルタイムエクスポートが有効になります。

ExportFileSize

エクスポートファイルごとの最大サイズ (バイト単位)。ExportDir が設定されている場合にローテーションに使用されます。

デフォルト: 1G
 範囲: 1M-1G

ExportType

リアルタイムエクスポートのためのカンマ区切りのエンティティタイプ（events、history、trends）のリスト（デフォルトですべてのタイプ）。ExportDir が設定されている場合のみ有効です。
ExportType が指定されているが ExportDir が指定されていない場合は、設定エラーとなり、サーバーは起動しませんのでご注意ください。

ヒストリとトレンドのエクスポート例:

```
ExportType=history,trends
```

イベントエクスポートのみの例:

```
ExportType=events
```

ExternalScripts

外部スクリプトの場所（datadir のコンパイル時インストール変数に依存）。

デフォルト：/usr/local/share/zabbix/externalscripts

Fping6Location

fping6 の場所。fping6 バイナリが root 所有で SUID フラグが設定されていることを確認してください。fping ユーティリティが IPv6 アドレスを処理できる場合は空にしてください（”Fping6Location=”）。

デフォルト：/usr/sbin/fping6

FpingLocation

fping の場所。fping バイナリが root 所有で SUID フラグが設定されていることを確認してください。

デフォルト: `/usr/sbin/fping`

FrontendAllowedIP

フロントエンドからの接続を許可する IP アドレスまたは CIDR 範囲のカンマ区切りリスト。このパラメータが設定されている場合、ここにリストされたアドレスからのみフロントエンド接続が受け入れられます。デフォルトでは、すべてのフロントエンドリクエストの接続が受け入れられます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。任意の IPv4 アドレスを許可するには'0.0.0.0/0' を使用できます。

例:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

高可用性クラスタノード名。空の場合、サーバーはスタンドアロンモードで動作し、空の名前のノードが作成されます。

HistoryCacheSize

ヒストリキャッシュのサイズ (バイト単位)。ヒストリデータを保存するための共有メモリサイズ。

デフォルト: 16M
 範囲: 128K-16G

HistoryIndexCacheSize

ヒストリインデックスキャッシュのサイズ (バイト単位)。ヒストリキャッシュに保存されたヒストリデータをインデックス化するための共有メモリサイズ。1 アイテムをキャッシュするのに約 100 バイトのインデックスキャッシュサイズが必要です。

デフォルト: 4M
 範囲: 128K-16G

HistoryStorageDateIndex

履歴ストレージで履歴値の前処理を有効にし、日付に基づいて異なるインデックスに値を保存します。

デフォルト: 0
 値: 0 - 無効; 1 - 有効

HistoryStorageURL

履歴ストレージの HTTP[S] URL。このパラメータはElasticsearchのセットアップに使用されます。

HistoryStorageTypes

履歴ストレージに送信される値のタイプをカンマ区切りで指定します。このパラメータはElasticsearchのセットアップに使用されます。

デフォルト: `uint,dbl,str,log,text`

HousekeepingFrequency

このパラメータは、Zabbix がハウスキーピング処理を何時間ごとに実行するかを決定します。ハウスキーピングとは、データベースから古い情報を削除することです。
 注: ハウスキーパーの過負荷を防ぐために (例えば、ヒストリやトレンドの保持期間を大幅に短縮した場合など)、1 回のハウスキーピングサイクルで削除される古い情報は、各アイテムごとに HousekeepingFrequency 時間の 4 倍を超えません。したがって、HousekeepingFrequency が 1 の場合、1 サイクルで削除される古い情報は最大 4 時間分 (最も古いエントリから) となります。
 注: サーバ起動時の負荷を軽減するため、ハウスキーピングはサーバ起動後 30 分間延期されます。したがって、HousekeepingFrequency が 1 の場合、サーバ起動後最初のハウスキーピング処理は 30 分後に実行され、その後は 1 時間ごとに繰り返されます。
HousekeepingFrequency を 0 に設定することで、自動ハウスキーピングを無効にすることができます。この場合、ハウスキーピング処理は *housekeeper_execute* ランタイム制御オプションでのみ開始でき、1 回のハウスキーピングサイクルで削除される古い情報の期間は、前回のハウスキーピングサイクルからの期間の 4 倍ですが、4 時間未満にはならず、4 日を超えることもありません。
 ランタイム制御オプションについてはこちらも参照してください。

デフォルト: 1
 範囲: 0-24

Include

個々のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。指定したディレクトリ内の関連ファイルのみを含めるために、アスタリスクワイルドカード文字によるパターンマッチングがサポートされています。制限事項については特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。Java ポーラーが起動されている場合のみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイがリッスンするポート。

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値もシステムに依存し、値が高すぎる場合は「実装で指定された最大値」に暗黙的に切り捨てられることがあります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーが待ち受ける IP アドレスのカンマ区切りリスト。
 このパラメータが指定されていない場合、トラッパーはすべてのネットワークインターフェースで待ち受けます。

デフォルト: 0.0.0.0

ListenPort

trapper のリッスンポート。

デフォルト: 10051
 範囲: 1024-32767

LoadModule

サーバーの起動時にロードするモジュール。モジュールはサーバーの機能を拡張するために使用されます。モジュールは、LoadModulePath で指定されたディレクトリに配置するか、パスをモジュール名の前に付ける必要があります。パスが絶対パス（'/' で始まる）である場合は、LoadModulePath は無視されます。
 書式:
LoadModule=<module.so>
LoadModule=<path/module.so>
LoadMod
複数の LoadModule パラメータを含めることができます。

LoadModulePath

サーバーモジュールの場所へのフルパス。デフォルトはコンパイルオプションに依存します。

LogFile

ログファイルの名前。

必須: LogType が *file* に設定されている場合は必須。それ以外は不要

LogFileSize

ログファイルの最大サイズ (MB 単位)。
0 - 自動ログローテーションを無効にします。
 注意: ログファイルサイズの上限に達し、何らかの理由でファイルのローテーションに失敗した場合、既存のログファイルは切り詰められ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024
 必須: LogType が *file* に設定されている場合は必須。それ以外は不要

LogSlowQueries

データベースクエリがログに記録されるまでの時間をミリ秒単位で指定します。
0 - 遅いクエリを記録しません。
 このオプションは DebugLevel=3 から有効になります。

デフォルト: 0
 範囲: 0-3600000

LogType

ログ出力のタイプ：
file - LogFile パラメータで指定されたファイルにログを書き込む;
system - syslog にログを書き込む;
console - 標準出力にログを書き込む。

デフォルト：file

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーによって一度に実行できる非同期チェックの最大数です。StartHTTPAgentPollers、StartAgentPollers、StartSNMPPollersを参照してください。

デフォルト: 1000
 範囲: 1-1000

MaxHousekeeperDelete

1 つのハウスキーピングサイクルで 1 つのタスクごとに削除される行数（[tablename]、[field]、[value] に対応）は、'MaxHousekeeperDelete' を超えません。
0 に設定した場合、制限は一切ありません。この場合、データベースの過負荷 [2] にならないように、何

をしているのかを十分に理解している必要があります。
 このパラメータは、すでに削除されたアイテムの履歴およびトレンドの削除にのみ適用されます。

デフォルト: 5000
 範囲: 0-1000000

NodeAddress

フロントエンドがサーバーに接続する方法を上書きするための、オプションのポートを含む IP またはホスト名。
 フォーマット: <address>[:<port>]

IP またはホスト名が設定されていない場合は、ListenIP の値が使用されます。ListenIP が設定されていない場合は、`localhost` が使用されます。
 ポートが設定されていない場合は、ListenPort の値が使用されます。ListenPort が設定されていない場合は、10051 が使用されます。

 このオプションは、フロントエンドの設定で指定されたアドレスによって上書きできます。

 関連項目: HANodeName パラメータ; 高可用性の有効化。

デフォルト: `localhost:10051`

PidFile

PID ファイルの名前。

デフォルト: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Zabbix が削除されたトリガーの問題を削除する頻度（秒単位）を決定します。

デフォルト: 60
 範囲: 1-3600

ProxyConfigFrequency

Zabbix サーバーが Zabbix プロキシに構成データを送信する頻度（秒単位）を決定します。パッシブモードのプロキシにのみ使用されます。

デフォルト: 10
 範囲: 1-604800

ProxyDataFrequency

Zabbix サーバーが Zabbix プロキシからヒストリデータを要求する頻度を秒単位で指定します。パッシブモードのプロキシでのみ使用されます。

デフォルト: 1
 範囲: 1-3600

ServiceManagerSyncFrequency

Zabbix がサービスマネージャの設定を何秒ごとに同期するかを決定します。

デフォルト: 60
 範囲: 1-3600

SMSDevices

Zabbix サーバーで使用を許可するモデムファイルのカンマ区切りリスト。
 このパラメータが設定されていない場合、SMS の送信はできません。

例:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

SNMP トラップデーモンからサーバーにデータを渡すために使用される一時ファイルです。
zabbix_trap_receiver.pl または SNMPTT の設定ファイルと同じでなければなりません。

デフォルト: `/tmp/zabbix_traps.tmp`

SocketDir

内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリ。

デフォルト：`/tmp`

SourceIP

以下の送信元 IP アドレス:

- Zabbix プロキシおよび Zabbix エージェントへの外部接続
- エージェントレス接続 (VMware、SSH、JMX、SNMP、Telnet、シンプルチェック)
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- 前処理の JavaScript HTTP リクエスト
- 通知メールの送信 (SMTP サーバーへの接続)

- Webhook 通知 (JavaScript HTTP 接続)
- Vault への接続

SSHKeyLocation

SSH チェックとアクションのための公開鍵と秘密鍵の場所。

SSLCertLocation

クライアント認証用の SSL クライアント証明書ファイルの場所。
 このパラメータは Web 監視でのみ使用されます。

SSLKeyLocation

クライアント認証用の SSL 秘密鍵ファイルの場所。
 このパラメータは Web 監視でのみ使用されます。

SSLCALocation

SSL サーバー証明書の検証のための認証局 (CA) ファイルの場所を上書きします。設定しない場合は、システム全体のディレクトリが使用されます。
 このパラメータの値は libcurl オプション CURLOPT_CAPATH として設定されます。libcurl 7.42.0 より前のバージョンでは、libcurl が OpenSSL を使用するようにコンパイルされている場合のみ有効です。詳細はcURL の Web ページを参照してください。
 このパラメータは Web 監視および SMTP 認証で使用されます。

StartAgentPollers

Zabbix エージェントpollersのプリフォークインスタンス数。MaxConcurrentChecksPerPollerも参照してください。

デフォルト: 1
 範囲: 0-1000

StartAlerters

alertersのプリフォークインスタンス数。

デフォルト：3
 範囲：1-100

StartBrowserPollers

ブラウザーアイテムpollersのプリフォークされたインスタンスの数。

デフォルト: 1
 範囲: 0-1000

StartConnectors

コネクタワーカーのプリフォークインスタンスの数。コネクタワーカーが起動されると、コネクタマネージャプロセスも自動的に起動されます。

デフォルト: 0
 範囲: 0-1000

StartDBSyncers

ヒストリーシンカーの事前フォークインスタンスの数。
 注意: この値を変更する際は注意してください。増やすことで逆効果になる場合があります。概算で、デフォルト値は最大 4000 NVPS まで処理できるはずです。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

ディスカバリーワーカーの事前フォークされたインスタンスの数[1]。

デフォルト: 5
 範囲: 0-1000

StartEscalators

エスカレーターのプリフォークインスタンスの数。

デフォルト: 1
 範囲: 1-100

StartHistoryPollers

ヒストリーポーラーの事前フォークされたインスタンス数。
 計算チェックにのみ必要です。

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP エージェントpollersの事前フォークされたインスタンスの数。MaxConcurrentChecksPerPollerも参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP ポーラーのプリフォークインスタンス数[1]。

デフォルト: 1
 範囲: 0-1000

StartIPMIPollers

IPMI ポーラーの事前フォークされたインスタンス数。

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java ポーラーの事前フォークインスタンス数[1]。

デフォルト: 0
 範囲: 0-1000

StartLLDProcessors

低レベルディスカバリ (LLD) ワーカー[1] のプリフォークインスタンス数。
LLD ワーカープロセスが開始されると、LLD マネージャープロセスも自動的に開始されます。

デフォルト: 2
 範囲: 0-100

StartODBCPollers

ODBC ポーラーのプリフォークされたインスタンス数[1]。

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP ピンガーの事前フォークインスタンス数[1]。

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能なホストのポーラー (IPMI および Java を含む) のプリフォークされたインスタンス数[1]。
 通常のポーラー、IPMI ポーラー、Java ポーラーのいずれかが起動している場合、少なくとも 1 つの到達不能ホスト用ポーラーが必要です。

デフォルト: 1
 範囲: 0-1000

StartPollers

ポーラーのプリフォークされたインスタンス数[1]。

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

事前処理ワーカー[1] スレッドの事前起動インスタンス数は、利用可能な CPU コア数以上に設定する必要があります。事前処理が CPU に依存せず、ネットワークリクエストが多い場合は、より多くのワーカーを設定してください。

デフォルト: 16
 範囲: 1-1000

StartProxyPollers

パッシブプロキシ用のポーラーの事前フォークされたインスタンス数[1]。

デフォルト: 1
 範囲: 0-250

StartReportWriters

レポートライターの事前フォークされたインスタンスの数。
0 に設定すると、スケジュールされたレポート生成が無効になります。
 レポートマネージャープロセスは、レポートライターが起動されると自動的に開始されます。

デフォルト: 0
 範囲: 0-100

StartSNMPPollers

SNMP ポーラーのプリフォークされたインスタンスの数。MaxConcurrentChecksPerPollerも参照してください。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

1 に設定すると、SNMP トラッパープロセスが起動されます。

デフォルト: 0
 範囲: 0-1

StartTimers

タイマーのプリフォークインスタンスの数。
 タイマープロセスはメンテナンス期間を処理します。

デフォルト：1
 範囲：1-1000

StartTrappers

トラッパーのプリフォークされたインスタンス数[1]。
 トラッパーは、Zabbix sender、アクティブエージェント、およびアクティブプロキシからの着信接続を受け付けます。

デフォルト: 5
 範囲: 1-1000

StartVMwareCollectors

事前にフォークされたVMware コレクタインスタンスの数。

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

カンマ区切りの IP アドレス（CIDR 表記も可）または外部 Zabbix インスタンスの DNS 名のリストです。ここにリストされたアドレスからのみ統計リクエストが受け付けられます。このパラメータが設定されていない場合、統計リクエストは受け付けられません。
IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。

例:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

タイムアウト

Zabbix プロキシ、エージェント、Web サービス、レガシー SNMP チェック（単一の OID 番号または文字列）、および `icmpping[*]` アイテムとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。

このパラメータは、さまざまな通信操作の期間を定義します：

- Zabbix エージェントでのリモートコマンド実行
- SSH/Telnet コマンドの実行
- Zabbix Web サービスへのリクエスト
- メディアタイプのテストリクエストおよび `history.push` メソッドの通信タイムアウト
- IPMI インターフェースが利用できなくなった場合のアイテムの再スケジューリング
- 権限や暗号化の問題でデータ交換に失敗した場合に Zabbix プロキシに応答を送信
- ICMP ピンガー
- IPC 非同期ソケットおよびランタイム制御オプションのデッドライン
- JMX 接続
- リモート Zabbix プロキシまたはサーバーからの統計情報の取得
- Zabbix フロントエンドへの応答の送信
- 非同期ポーラーの DNS リクエスト
- アクティブチェックのハートビート応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- トラッパーがリクエストの処理に失敗した場合の応答
- Zabbix エージェントへのアクティブチェックリストの送信

このタイムアウトは、フロントエンドで柔軟なタイムアウト設定（グローバル、プロキシ、またはアイテム単位）が構成されているチェックには使用されません。
 たとえば、SNMP の `walk[OID]` および `get[OID]` アイテムは、フロントエンドで設定されたタイムアウトを使用しますが、レガシー SNMP チェックはサーバーのタイムアウト値を使用します。

デフォルト：3
 範囲：1-30

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書の検証用の最上位 CA 証明書を含むファイルのフルパス名。

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用される、サーバー証明書または証明書チェーンを含むファイルのフルパス名。

TLSCipherAll

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例：

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書ベースおよび PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例：

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例：

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号化方式文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。PSK ベースの暗号化のために、デフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSFrontendAccept

フロントエンドから受け入れる接続。

複数の値をカンマ区切りで指定できます。

- unencrypted - 暗号化なしの接続を受け入れます。
- cert - TLS および証明書で保護された接続を受け入れます。

デフォルト: `unencrypted`

TLSFrontendCertIssuer

許可されたフロントエンド証明書発行者。

TLSFrontendCertSubject

許可されるフロントエンド証明書のサブジェクト。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるサーバーの秘密鍵を含むファイルのフルパス名。

TLSListen

trapper ソケットでの TLS を制御します。

サポートされている値:

- required - TLS 接続のみを受け付ける

TmpDir

一時ディレクトリ。

デフォルト: `/tmp`

TrapperTimeout

以下のタイムアウト（秒）を指定します。

- Zabbix プロキシからの履歴データの取得
- Zabbix プロキシへの設定データの送信
- Zabbix サーバーでのグローバルスクリプト実行またはリモートコマンド実行

デフォルト：300
 範囲：1-300

TrendCacheSize

トレンドキャッシュのサイズ (バイト単位)。
 トレンドデータを保存するための共有メモリサイズ。

デフォルト: `4M`
 範囲: 128K-16G

TrendFunctionCacheSize

トレンド関数キャッシュのサイズ (バイト単位)。
 計算されたトレンド関数データをキャッシュするための共有メモリサイズ。

デフォルト: `4M`
 範囲: 128K-2G

UnavailableDelay

到達不能期間中にホストの可用性をチェックする間隔 (秒) を指定します。

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能期間中にホストの可用性をチェックする間隔 (秒) を指定します。

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能になってから何秒後にホストを利用不可とみなすかを決定します。

デフォルト: 45
 範囲: 1-3600

ユーザー

システム上の特定の既存ユーザーに権限を落とします。
「root」として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: `zabbix`

ValueCacheSize

ヒストリ値キャッシュのサイズ (バイト単位)。
 アイテムのヒストリデータリクエストをキャッシュするための共有メモリサイズ。
0 に設定すると値キャッシュが無効になります (非推奨)。
 値キャッシュが共有メモリを使い果たすと、5 分ごとに警告メッセージがサーバーログに書き込まれます。

デフォルト: `8M`
 範囲: 0,128K-64G

Vault

ボールトプロバイダーを指定します。
HashiCorp - HashiCorp KV Secrets Engine バージョン 2
CyberArk - CyberArk Central Credential Provider
 フロントエンドで設定されたボールトプロバイダーと一致する必要があります。

デフォルト: `HashiCorp`

VaultDBPath

データベースの資格情報をキーで取得するために、Vault から取得するパスまたはクエリ（Vault によって異なります）。

HashiCorpで使用されるキーは「password」と「username」です。

VaultPrefix=/v1/secret/data/zabbix/を指定した場合のパス例:

```
database
```

VaultPrefix を指定しない場合のパス例:

```
secret/zabbix/database
```

CyberArkで使用されるキーは「Content」と「UserName」です。

例:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

このオプションは、DBUser および DBPassword が指定されていない場合のみ使用できます。

VaultPrefix

Vault の種類に応じて、Vault のパスまたはクエリのカスタムプレフィックスを指定します。指定しない場合は、最適なデフォルト値が使用されます。
HashiCorp の場合、VaultPrefix が指定されていない場合は、'data' が自動的にマウントポイントの後ろに追加されることに注意してください。

Hashicorp のプレフィックス例:

```
v1/secret/data/zabbix/
```

Cyberark のプレフィックス例:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイルの名前
 証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL キーファイルフィールドを空のままにします。
 このファイルを含むディレクトリは、構成パラメータ SSLCertLocation で指定されます。
 このオプションは省略できますが、CyberArkCCP ボールトには推奨されます。

VaultTLSKeyFile

クライアント認証に使用する SSL 秘密鍵ファイルの名前。
 秘密鍵ファイルは PEM1 形式である必要があります。
 このファイルを含むディレクトリは、設定パラメータ SSLKeyLocation で指定します。
 このオプションは省略可能ですが、CyberArkCCP ボールトでは推奨されます。

VaultToken

Vault macrosで指定されたパスへの読み取り専用権限と、オプションの VaultDBPath 構成パラメータで指定されたパスへの読み取り専用権限を持つ Zabbix サーバー専用に生成された HashiCorp Vault 認証トークン。
VaultToken と VAULT_TOKEN 環境変数が同時に定義されている場合はエラーとなります。

必須: Vault が *HashiCorp* に設定されている場合は必須、それ以外は不要

VaultURL

Vault サーバーの HTTP[S] URL。SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリサイズ。
VMware 内部チェック zabbix[vmware,buffer,...] を使用して VMware キャッシュの使用状況を監視できます（内部チェックを参照）。
 起動するように設定された vmware コレクターインスタンスがない場合、共有メモリは割り当てられません。

デフォルト: 8M
 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからデータを収集する間隔 (秒単位)。
 この間隔は、VMware 監視アイテムの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンタ統計を取得する間隔（秒単位）。この間隔は、VMware パフォーマンスカウンタを使用する VMware 監視itemの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

VMware コレクターが VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待つ最大秒数。

デフォルト: 10
 範囲: 1-300

WebServiceURL

Zabbix Web サービスへの HTTP[S] URL。形式は `http[s]://host:port/report` です。

例:

```
WebServiceURL=http://localhost:10053/report
```

注意: スキーム（`http://`）は非 TLS（HTTP）接続の場合のみ省略可能です。TLS が設定されている場合は、`https://`を使用する必要があります。

WebDriverURL

WebDriver インターフェースの HTTP[S] URL。

例 (Selenium WebDriver スタンドアロン・サーバーで使用):

```
WebDriverURL=http://localhost:4444
```

脚注

[1] データ収集プロセス（ポーラー、到達不能ポーラー、ODBC ポーラー、HTTP ポーラー、Java ポーラー、ピンガー、トラッパー、プロキシポーラー）に加えて、IPMI マネージャー、SNMP トラッパー、前処理ワーカー、ディスカバリーワーカーが多すぎると、前処理マネージャーのプロセスごとのファイルディスクリプタ制限を使い果たす可能性があります。

> **Warning:**
> ファイルディスクリプタ制限を使い果たすと、Zabbix サーバーは停止します。これは通常、起動直後に発生しますが、場合によっては時間がかかることもあります。このような問題を回避するために、Zabbix サーバーの設定ファイルを見直し、同時チェック数やプロセス数を最適化してください。また、必要に応じて、システムの制限を確認・調整し、ファイルディスクリプタ制限が十分に高く設定されていることを確認してください。

[2] 多くのアイテムが削除されると、ハウスキーパーがこれらのアイテムの履歴データをすべて削除する必要があるため、データベースへの負荷が増加します。例えば、テンプレートからアイテムプロトタイプを 1 つだけ削除する場合でも、このテンプレートが 50 台のホストにリンクされていて、各ホストでプロトタイプが 100 個の実アイテムに展開されている場合、合計で 5000 個のアイテム (1*50*100）を削除する必要があります。MaxHousekeeperDelete（MaxHousekeeperDelete=500）を 500 に設定している場合、ハウスキーパーは 1 サイクルで削除されたアイテムの履歴およびトレンドテーブルから最大 2500000 個（5000*500）の値を削除する必要があります。

[3] サーバー設定では、DBSocket と DBPort は排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

2 Zabbix プロキシ

概要

Zabbix プロキシの設定ファイル（zabbix_proxy.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしでリストされています。パラメータをクリックすると、詳細情報が表示されます。

パラメータ	説明
AllowRoot	プロキシを'root' として実行することを許可します。
AllowUnsupportedDBVersions	サポートされていないデータベースバージョンでプロキシを動作させることを許可します。
CacheSize	設定キャッシュのサイズ。
ConfigFrequency	このパラメータは非推奨です（代わりに ProxyConfigFrequency を使用してください）。 プロキシが Zabbix サーバーから設定データを取得する間隔（秒単位）。
DataSenderFrequency	プロキシが収集したデータをサーバーに送信する間隔（秒単位）。
DBHost	データベースのホスト名。
DBName	データベース名または SQLite3 の場合はデータベースファイルへのパス。
DBPassword	データベースのパスワード。
DBPort	ローカルソケットを使用しない場合のデータベースポート。
DBSchema	データベーススキーマ名。PostgreSQL で使用します。
DBSocket	MySQL ソケットファイルのパス。
DBUser	データベースユーザー。
DBTLSConnect	このオプションを指定した値に設定すると、データベースへの TLS 接続の使用を強制します。
DBTLSCAFile	データベース証明書の検証用の最上位 CA 証明書を含むファイルのフルパス名。

パラメータ	説明
DBTLSCertFile	データベース認証用の Zabbix プロキシ証明書を含むファイルのフルパス名。
DBTLSKeyFile	データベース認証用の秘密鍵を含むファイルのフルパス名。
DBTLSCipher	Zabbix プロキシが TLS v1.2 までの TLS プロトコルで許可する暗号化アルゴリズムのリスト。 MySQL のみサポート。
DBTLSCipher13	Zabbix プロキシが TLS v1.3 プロトコルで許可する暗号化スイートのリスト。MySQL（バージョン 8.0.16 以降）のみサポート。
DebugLevel	デバッグレベル。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可するかどうか。
ExternalScripts	外部スクリプトの場所。
Fping6Location	fping6 の場所。
FpingLocation	fping の場所。
HistoryCacheSize	ヒストリキャッシュのサイズ。
HistoryIndexCacheSize	ヒストリインデックスキャッシュのサイズ。
Hostname	一意で大文字小文字を区別するプロキシ名。
HostnameItem	Hostname が未定義の場合に設定するためのアイテム。
HousekeepingFrequency	Zabbix がハウスキーピング処理を実行する間隔（時間単位）。
Include	設定ファイルに個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。
JavaGatewayPort	Zabbix Java ゲートウェイがリッスンするポート。
ListenBacklog	TCP キュー内の保留中の接続の最大数。
ListenIP	トラッパーがリッスンするカンマ区切りの IP アドレスリスト。
ListenPort	トラッパーのリッスンポート。
LoadModule	プロキシ起動時にロードするモジュール。
LoadModulePath	プロキシモジュールの場所へのフルパス。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録するかどうか。
LogSlowQueries	データベースクエリがログに記録されるまでの最大時間。
LogType	ログ出力の種類。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラーまたは SNMP ポーラーが同時に実行できる非同期チェックの最大数。
PidFile	PID ファイル名。
ProxyBufferMode	ヒストリ、ディスカバリ、自動登録データの保存メカニズム（disk/memory/hybrid）を指定します。
ProxyConfigFrequency	プロキシが Zabbix サーバーから設定データを取得する間隔（秒単位）。
ProxyLocalBuffer	プロキシは、データがすでにサーバーと同期されていても、N 時間ローカルにデータを保持します。
ProxyMemoryBufferAge	プロキシメモリバッファ内のデータの最大保持期間（秒単位）。
ProxyMemoryBufferSize	収集したヒストリ、ディスカバリ、自動登録データ用の共有メモリキャッシュのサイズ。
ProxyMode	プロキシの動作モード（アクティブ/パッシブ）。
ProxyOfflineBuffer	Zabbix サーバーとの接続がない場合にプロキシがデータを保持する期間（時間単位）。
Server	ProxyMode がアクティブモードの場合：設定データの取得およびデータ送信先の Zabbix サーバーの IP アドレスまたは DNS 名（address:port）またはクラスタ（address:port;address2:port）。 ProxyMode がパッシブモードの場合：カンマ区切りの IP アドレス（CIDR 表記も可）または Zabbix サーバーの DNS 名のリスト。
SNMPTrapperFile	SNMP トラップデーモンからプロキシにデータを渡すための一時ファイル。
SocketDir	内部 Zabbix サービスで使用される IPC ソケットを格納するディレクトリ。
SourceIP	送信元 IP アドレス。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵および秘密鍵の場所。
SSLCertLocation	クライアント認証用の SSL クライアント証明書ファイルの場所。
SSLKeyLocation	クライアント認証用の SSL 秘密鍵ファイルの場所。
SSLCALocation	SSL サーバー証明書検証用の認証局（CA）ファイルの場所を上書きします。
StartAgentPollers	非同期 Zabbix エージェントポーラーの事前フォークインスタンス数。
StartBrowserPollers	ブラウザーアイテムポーラーの事前フォークインスタンス数。
StartDBSyncers	ヒストリ同期プロセスの事前フォークインスタンス数。
StartDiscoverers	ディスカバリワーカーの事前フォークインスタンス数。
StartHTTPAgentPollers	非同期 HTTP エージェントポーラーの事前フォークインスタンス数。
StartHTTPPollers	HTTP ポーラーの事前フォークインスタンス数。
StartIPMIPollers	IPMI ポーラーの事前フォークインスタンス数。
StartJavaPollers	Java ポーラーの事前フォークインスタンス数。
StartODBCPollers	ODBC ポーラーの事前フォークインスタンス数。
StartPingers	ICMP ピンガーの事前フォークインスタンス数。

パラメータ	説明
StartPollersUnreachable	到達不能ホスト用のポーラー（IPMI および Java を含む）の事前フォークインスタンス数。
StartPollers	ポーラーの事前フォークインスタンス数。
StartPreprocessors	前処理ワーカーの事前起動インスタンス数。
StartSNMPPollers	非同期 SNMP ポーラーの事前フォークインスタンス数。
StartSNMPTrapper	1 に設定すると、SNMP トラッパープロセスが開始されます。
StartTrappers	トラッパーの事前フォークインスタンス数。
StartVMwareCollectors	VMware コレクターの事前フォークインスタンス数。
StatsAllowedIP	カンマ区切りの IP アドレス（CIDR 表記も可）または外部 Zabbix インスタンスの DNS 名のリスト。統計リクエストはここにリストされたアドレスからのみ受け付けられます。
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）、`icmpping[*]` アイテムの待機時間（秒単位）を指定します。
TLSAccept	Zabbix サーバーから受け入れる着信接続の種類。
TLSCAFile	Zabbix コンポーネント間の暗号化通信で使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用される、サーバー証明書または証明書チェーンを含むファイルのフルパス名。
TLSCipherAll	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。証明書ベースおよび PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書ベースおよび PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS 優先度文字列または OpenSSL（TLS 1.2）暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	プロキシが Zabbix サーバーに接続する方法。
TLSCRLFile	失効証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信で使用されます。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用される、プロキシ秘密鍵を含むファイルのフルパス名。
TLSListen	トラッパーソケットでの TLS を制御します。
TLSPSKFile	Zabbix サーバーとの暗号化通信で使用される、プロキシ事前共有鍵を含むファイルのフルパス名。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信で使用される事前共有鍵の識別文字列。
TLSServerCertIssuer	許可されるサーバー証明書の発行者。
TLSServerCertSubject	許可されるサーバー証明書のサブジェクト。
TmpDir	一時ディレクトリ。
TrapperTimeout	以下のタイムアウト（秒単位）を指定します： - Zabbix サーバーからの設定データ取得； - グローバルスクリプト実行またはリモートコマンド実行。
UnavailableDelay	到達不能期間中にホストの可用性をチェックする間隔。
UnreachableDelay	到達不能期間中にホストの可用性をチェックする間隔。
UnreachablePeriod	到達不能になってからホストを到達不能と見なすまでの秒数。
User	システム上の特定の既存ユーザーに権限をドロップします。
Vault	ボールトプロバイダー。
VaultDBPath	データベース認証情報をキーで取得する場所。
VaultPrefix	ボールトパスまたはクエリのカスタムプレフィックス。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイル名。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイル名。
VaultToken	HashiCorp ボールト認証トークン。
VaultURL	ボールトサーバーの HTTP[S] URL。
VMwareCacheSize	VMware データ保存用の共有メモリサイズ。
VMwareFrequency	単一の VMware サービスからデータを収集する間隔（秒単位）。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンタ統計を取得する間隔（秒単位）。
VMwareTimeout	VMware コレクターが VMware サービスからの応答を待機する最大秒数。
WebDriverURL	WebDriver インターフェースの HTTP[S] URL。

パラメータが必須であると明示的に記載されていない限り、すべてのパラメータは必須ではありません。

注意：

- デフォルト値はデーモンのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは行頭でのみサポートされます。

パラメータの詳細

AllowRoot

プロキシを’root’ として実行できるようにします。無効にした場合、プロキシが’root’ で開始された場合は’zabbix’ ユーザーに切り替えようとします。通常のユーザーで開始した場合は効果がありません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

AllowUnsupportedDBVersions

サポートされていないデータベースバージョンでプロキシが動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

CacheSize

設定キャッシュのサイズ（バイト単位）。ホストおよびアイテムデータを格納するための共有メモリサイズ。

デフォルト: 32M
 範囲: 128K-64G

ConfigFrequency

このパラメータは非推奨です（代わりに ProxyConfigFrequency を使用してください）。
 プロキシが Zabbix サーバーから設定データを取得する間隔（秒単位）。
 アクティブプロキシ用のパラメータ。パッシブプロキシの場合は無視されます（ProxyMode パラメータを参照）。

デフォルト：3600
 範囲：1-604800

DataSenderFrequency

プロキシは収集したデータを N 秒ごとにサーバーに送信します。アクティブプロキシは、リモートコマンドタスクのために毎秒 Zabbix サーバーにポーリングし続けることに注意してください。
 アクティブプロキシ用のパラメータです。パッシブプロキシでは無視されます（ProxyMode パラメータを参照）。

デフォルト: 1
 範囲: 1-3600

DBHost

データベースのホスト名。
MySQL の場合、`localhost` または空文字列はソケットの使用を意味します。PostgreSQL の場合、空文字列のみがソケットの使用を試みます。

デフォルト: `localhost`

DBName

SQLite3 のデータベース名またはデータベースファイルへのパス（Zabbix のマルチプロセスアーキテクチャでは、インメモリデータベース は使用できません。例: `:memory:`、`file::memory:?cache=shared`、`file:memdb1?mode=memory&cache=shared`）
 警告: Zabbix サーバーが使用しているデータベースと同じデータベースを使用しないでください。

必須: はい

DBPassword

データベースのパスワード。パスワードを使用しない場合は、この行をコメントアウトしてください。SQLite の場合は無視されます。

DBPort

ローカルソケットを使用しない場合のデータベースポート。SQLite では無視されます。

範囲: 1024-65535

DBSchema

データベーススキーマ名です。PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパス。[1]
 ローカルソケットを使用しない場合のデータベースポート。SQLite の場合は無視されます。

デフォルト: 3306

DBUser

データベースユーザー。SQLite の場合は無視されます。

DBTLSConnect

このオプションを設定すると、強制的にデータベースへ TLS を使用して接続します。
required - TLS を使用して接続します。
verify_ca - TLS を使用して接続し、証明書を検証します。
verify_full - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID がその証明書と一致することを確認します。
MySQL 5.7.11 以降および PostgreSQL では、次の値がサポートされています。”required”、”verify”、”verify_full”
MariaDB バージョン 10.2.6 以降では、”required” および”verify_full” がサポートされています。
 デフォルトではどのオプションも設定されておらず、動作はデータベースの設定によって異なります。

DBTLSCAFile

データベース証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。

必須: いいえ（DBTLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須）

DBTLSCertFile

データベースへの認証に使用する Zabbix プロキシ証明書を含むファイルのフルパス名。

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を格納しているファイルのフルパス名。

DBTLSCipher

Zabbix プロキシが TLS v1.2 までの TLS プロトコルで許可する暗号化方式のリスト。MySQL のみサポートされています。

DBTLSCipher13

Zabbix プロキシが TLS v1.3 プロトコルで許可する暗号スイートのリスト。MySQL のみサポートされており、バージョン 8.0.16 以降で利用可能です。

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

外部スクリプト

外部スクリプトの場所（コンパイル時のインストール変数 `datadir` に依存します）

デフォルト: `/usr/local/share/zabbix/externalscripts`

Fping6Location

fping6 の場所。fping6 バイナリが root 所有で SUID フラグが設定されていることを確認してください。fping ユーティリティが IPv6 アドレスを処理できる場合は空にしてください（”Fping6Location=”）。

デフォルト：`/usr/sbin/fping6`

FpingLocation

fping の場所。fping バイナリが root 所有で SUID フラグが設定されていることを確認してください。

デフォルト：`/usr/sbin/fping`

HistoryCacheSize

ヒストリキャッシュのサイズ (バイト単位)。ヒストリデータを保存するための共有メモリサイズ。

デフォルト: `16M`
 範囲: 128K-16G

HistoryIndexCacheSize

ヒストリーインデックスキャッシュのサイズ（バイト単位）。ヒストリーキャッシュに格納されたヒストリーデータをインデックス化するための共有メモリサイズ。インデックスキャッシュは、1 アイテムをキャッシュするのに約 100 バイト必要です。

デフォルト: `4M`
 範囲: 128K-16G

ホスト名

一意で大文字と小文字を区別するプロキシ名。プロキシ名がサーバーに認識されていることを確認してください。
 使用可能な文字: 英数字、'.'、' '、'_'、'-' 最大長: 128

デフォルト: HostnameItem で設定

HostnameItem

Hostname が未定義の場合に Hostname を設定するために使用されるアイテム（これはエージェントと同様にプロキシ上で実行されます）。Hostname が設定されている場合は無視されます。
UserParameters、パフォーマンスカウンター、エイリアスはサポートされていませんが、system.run[] はサポートされています。

デフォルト：system.hostname

HousekeepingFrequency

Zabbix がハウスキーピング処理を実行する頻度（時間単位）。ハウスキーピングは、データベースから古い情報を削除することです。
 注意: プロキシの起動時の負荷を下げるため、ハウスキーピングはプロキシ起動後 30 分間延期されます。したがって、HousekeepingFrequency が 1 の場合、プロキシ起動後の最初のハウスキーピング処理は 30 分後に実行され、その後は毎時繰り返されます。
HousekeepingFrequency を 0 に設定することで自動ハウスキーピングを無効にすることも可能です。この場合、ハウスキーピング処理は *housekeeper_execute* ランタイム制御オプションでのみ開始できます。

デフォルト: 1
 範囲: 0-24

Include

個別のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。
 指定したディレクトリ内の関連ファイルのみを含めるために、パターンマッチング用のアスタリスクワイルドカード文字がサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。Java ポーラーが起動されている場合のみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイが listen するポート

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' に自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーがリッスンする IP アドレスのカンマ区切りリスト
 このパラメータが指定されていない場合、トラッパーはすべてのネットワークインターフェースをリッスンします。

デフォルト: 0.0.0.0

ListenPort

トラッパーのリッスンポート

デフォルト: 10051
 範囲: 1024-32767

LoadModule

プロキシの起動時にロードするモジュール。モジュールはプロキシの機能を拡張するために使用されます。モジュールは、LoadModulePath で指定されたディレクトリに配置するか､パスをモジュール名の前に付ける必要があります｡前置パスが絶対パス（'/' で始まる）である場合､LoadModulePath は無視されます｡
 フォーマット:
LoadModule=<module.so>
LoadModule=<path/module.so>
複数の LoadModule パラメータを含めることができます。

LoadModulePath

プロキシモジュールの場所へのフルパス。デフォルトはコンパイルオプションに依存します。

LogFile

ログファイル名

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ。(MB)
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうか。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

LogSlowQueries

データベースクエリがログに記録されるまでの所要時間。(ミリ秒単位)
0 - スロークエリをログに記録しません。
 このオプションは、DebugLevel=3 以降で有効になります。

デフォルト: 0
 範囲: 0-3600000

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーによって一度に実行できる非同期チェックの最大数。StartHTTPAgentPollers、StartAgentPollers、StartSNMPPollersも参照してください。

デフォルト: 1000
 範囲: 1-1000

PidFile

PID ファイルの名前

デフォルト: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

ヒストリ、ネットワークディスカバリ、自動登録データの保存メカニズムを指定します。*disk* - データはデータベースに保存され、データベースからアップロードされます。*memory* - データはメモリに保存され、メモリからアップロードされます。バッファのメモリが不足した場合、古いデータは破棄されます。シャットダウン時にバッファは破棄されます。*hybrid* - プロキシバッファは通常、メモリモードと同様に動作しますが、メモリが不足するか、最も古いレコードが設定された期間を超えると、バッファがデータベースにフラッシュされ、すべてのデータがアップロードされるまでディスクモードのように動作し、その後再びメモリで動作します。シャットダウン時にはメモリバッファがデータベースにフラッシュされます。

詳細はプロキシのメモリバッファも参照してください。

デフォルト: `disk`
 値: disk; memory; hybrid

ProxyConfigFrequency

プロキシが Zabbix サーバーから設定データを取得する間隔 (秒単位)。
 アクティブプロキシ用のパラメータです。パッシブプロキシの場合は無視されます (ProxyMode パラメータを参照)。

デフォルト: 10
 範囲: 1-604800

ProxyLocalBuffer

データがサーバーと同期済みであっても、プロキシはデータをローカルに N 時間保持します。
 このパラメータは、ローカルデータがサードパーティ製アプリケーションで使用される場合に使用できます。

デフォルト: 0
 範囲: 0-720

ProxyMemoryBufferAge

プロキシメモリバッファ内のデータの最大経過時間 (秒単位)。有効 (ゼロ以外) で、プロキシメモリバッファ内のレコードがこれより古い場合、すべてのレコードがサーバーにアップロードされるまで、プロキシバッファをデータベースモードに強制的に切り替えます。このパラメータは、ProxyOfflineBuffer パラメータ以下でなければなりません。

デフォルト: 0
 範囲: 0;600-864000

ProxyMemoryBufferSize

収集したヒストリ、ディスカバリ、および自動登録データ用の共有メモリキャッシュのサイズ（バイト単位）。有効（ゼロ以外）の場合、プロキシはキャッシュがいっぱいになるか、保存されたレコードが ProxyMemoryBufferAge で定義された期間より古くなるまで、ヒストリ、ディスカバリ、および自動登録データをメモリ内に保持します。このパラメータは、ProxyLocalBuffer パラメータと同時に使用できません。

デフォルト: 0
 範囲: 0;128K-2G

ProxyMode

プロキシの動作モード。
0 - アクティブモードのプロキシ
1 - パッシブモードのプロキシ
 注意アクティブプロキシを使用する場合、(機密性の高い) プロキシの設定データが Zabbix サーバーのトラッパーポートにアクセスできる第三者に利用可能になる可能性があります。これは、誰でもアクティブプロキシを装って設定データを要求できるためです。認証は行われません。

デフォルト: 0
 範囲: 0-1

ProxyOfflineBuffer

Zabbix サーバーとの接続が切断された場合、プロキシは N 時間データを保持します。
 古いデータは失われます。

デフォルト: 1
 範囲: 1-720

サーバー

ProxyMode がアクティブモードに設定されている場合:
Zabbix サーバーの IP アドレスまたは DNS 名（アドレス: ポート）、またはクラスタ（アドレス: ポート; アドレス 2: ポート）。ここから設定データを取得し、データを送信します。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
 クラスタノードはセミコロンで区切る必要があります。

ProxyMode がパッシブモードに設定されている場合:
Zabbix サーバーの IP アドレス（CIDR 表記も可）または DNS 名のカンマ区切りリスト。ここに記載されたアドレスからの着信接続のみを受け付けます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われます。
'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: yes

SNMPTrapperFile

SNMP トラップデーモンからプロキシへデータを渡すために使用される一時ファイル
zabbix_trap_receiver.pl または SNMPTT 設定ファイルと同じである必要があります。

デフォルト: `/tmp/zabbix_traps.tmp`

SocketDir

Zabbix 内部サービスで使用される IPC ソケットを格納するディレクトリ

デフォルト: `/tmp`

SourceIP

以下の送信元 IP アドレス:

- Zabbix サーバーへの外部接続
- エージェントレス接続 (VMware、SSH、JMX、SNMP、Telnet、シンプルチェック)
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- 前処理の JavaScript HTTP リクエスト
- Vault への接続

SSHKeyLocation

SSH チェックおよびアクションで使用する公開鍵と秘密鍵の格納場所

SSLCertLocation

クライアント認証用の SSL クライアント証明書ファイルの格納場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLKeyLocation

クライアント認証用の SSL 秘密鍵ファイルの格納場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLCALocation

SSL サーバー証明書の検証のための証明書認証局 (CA) ファイルの場所。
 このパラメータの値は CURLOPT_CAPATH libcurl オプションとして設定されます。libcurl バージョン 7.42.0 より前では、libcurl が OpenSSL を使用するようにコンパイルされている場合のみ有効です。詳細はcURL の Web ページを参照してください。
 このパラメータは Web 監視および SMTP 認証で使用されます。

StartAgentPollers

Zabbix エージェントpollersの事前フォークされたインスタンスの数。MaxConcurrentChecksPerPollerも参照してください。

デフォルト: 1
 範囲: 0-1000

StartBrowserPollers

ブラウザアイテムポーラー の pre-forked インスタンスの数

デフォルト: 1
 範囲: 0-1000

StartDBSyncers

history syncersの pre-forked インスタンスの数。
 注: この値を変更する際は注意してください。値を大きくすると、メリットよりもデメリットが大きくなる可能性があります。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

検出ワーカーの pre-forked インスタンスの数

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP エージェントpollersの事前フォークされたインスタンス数。MaxConcurrentChecksPerPollerも参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP ポーラーの pre-forked インスタンスの数

デフォルト: 1 | 範囲: 0-1000

StartIPMIPollers

IPMI ポーラーの pre-forked インスタンスの数

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java ポーラーの pre-forked インスタンスの数

デフォルト: 0
 範囲: 0-1000

StartODBCPollers

ODBC ポーラーの pre-forked インスタンスの数

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP pingers の pre-forked インスタンス数

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能ホスト用ポーラー（IPMI および Java を含む）の事前フォークされたインスタンス数。通常のポーラー、IPMI ポーラー、または Java ポーラーが起動されている場合、少なくとも 1 つの到達不能ホスト用ポーラーが実行されている必要があります。

デフォルト: 1
 範囲: 0-1000

StartPollers

ポーラー の pre-forked インスタンスの数

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

事前に起動するworkerスレッドのインスタンス数を、利用可能な CPU コア数以上に設定する必要があります。前処理が CPU に依存せず、ネットワークリクエストが多い場合は、より多くのワーカーを設定する必要があります。

デフォルト: 16
 範囲: 1-1000

StartSNMPPollers

SNMP ポーラーのプリフォークインスタンス数。MaxConcurrentChecksPerPollerも参照。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

1 に設定すると、SNMP トラッパープロセスが起動します。

デフォルト: 0
 範囲: 0-1

StartTrappers

トラッパー の pre-forked インスタンスの数。
 トラッパーは、Zabbix sender とアクティブエージェントからの受信接続を受け入れます。

デフォルト: 5
 範囲: 0-1000

StartVMwareCollectors

VMware Collector pre-forked インスタンスの数。

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

カンマ区切りの IP アドレス（オプションで CIDR 表記）、または外部 Zabbix インスタンスの DNS 名のリストです。stats リクエストは、ここにリストされたアドレスからのみ受け付けます。このパラメータが設定されていない場合、stats リクエストは受け付けません。
IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。

例:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

タイムアウト

Zabbix サーバー、エージェント、Web サービス、レガシー SNMP チェック（単一の OID 番号または文字列）、および `icmpping[*]` アイテムとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。

このパラメータは、さまざまな通信操作の期間を定義します:

- Zabbix エージェントでのリモートコマンド実行
- SSH / Telnet コマンドの実行
- IPMI インターフェースが利用できなくなった場合のアイテムの再スケジューリング
- ICMP ピンガー
- 権限や暗号化の問題でデータ交換に失敗した場合に Zabbix サーバーに応答を送信
- IPC 非同期ソケットおよびランタイム制御オプションの締め切り
- 非同期ポーラの DNS リクエスト
- アクティブチェックのハートビート応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- アクティブチェックリストを Zabbix エージェントに送信

このタイムアウトは、フロントエンドで柔軟なタイムアウト設定が構成されているチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。たとえば、SNMP の `walk[OID]` および `get[OID]` アイテムは、フロントエンドで設定されたタイムアウトを使用しますが、レガシー SNMP チェックは引き続きサーバーのタイムアウト値を使用します。

デフォルト: 3
 範囲: 1-30

TLSAccept

Zabbix サーバーから受け入れる着信接続。パッシブプロキシで使用され、アクティブプロキシでは無視されます。複数の値をカンマで区切って指定できます:
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
psk - TLS および事前共有鍵（PSK）による接続を受け入れる
cert - TLS および証明書による接続を受け入れる

必須: TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続でも）、パッシブプロキシでは必須。それ以外は不要。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用されるピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用されるプロキシ証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号化方式文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降のための暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

プロキシが Zabbix サーバーに接続する方法。アクティブプロキシで使用され、パッシブプロキシでは無視されます。1 つの値のみ指定可能です:
unencrypted - 暗号化せずに接続（デフォルト）
psk - TLS および事前共有鍵（PSK）を使用して接続
cert - TLS および証明書を使用して接続

必須: アクティブプロキシで、TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも必須）、それ以外は不要

TLSCRLFile

失効した証明書を含むファイルのフルパス名です。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるプロキシ秘密鍵を格納するファイルのフルパス名

TLSListen

trapper ソケットでの TLS を制御します。

サポートされている値:

- required - TLS 接続のみを受け付ける

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるプロキシ事前共有キーを格納するファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー証明書の発行者

TLSServerCertSubject

許可されたサーバー証明書のサブジェクト

TmpDir

テンポラリディレクトリ

デフォルト: `/tmp`

TrapperTimeout

以下のタイムアウトを秒単位で指定します。
 - Zabbix サーバーからの設定データの取得
 - グローバルスクリプトの実行またはリモートコマンドの実行

デフォルト: 300
 範囲: 1-300

UnavailableDelay

非可用性 期間中にホストへの可用性をチェックする頻度（秒単位）

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能 期間中にホストへの可用性をチェックする頻度（秒単位）

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能が何秒間続いた後にホストを利用不可とみなすかの指定

デフォルト: 45
 範囲: 1-3600

User

システム上の特定の既存ユーザーに権限を落とします。
'root' として実行し、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: `zabbix`

Vault

Vault プロバイダー：
HashiCorp - HashiCorp KV Secrets Engine バージョン 2
CyberArk - CyberArk Central Credential Provider
 フロントエンドで設定された Vault プロバイダーと一致する必要があります。

デフォルト：`HashiCorp`

VaultDBPath

データベースの認証情報をキーで取得するための Vault のパスまたはクエリ（Vault によって異なります）。

HashiCorpで使用されるキーは「password」と「username」です。

VaultPrefix=/v1/secret/data/zabbix/を指定した場合のパス例:

```
database
```

VaultPrefix を指定しない場合のパス例:

```
secret/zabbix/database
```

CyberArkで使用されるキーは「Content」と「UserName」です。

例:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます。

VaultPrefix

Vault に応じて、Vault パスまたはクエリのカスタムプレフィックス。指定しない場合は、最も適切なデフォルト値が使用されます。
HashiCorp の場合、VaultPrefix が指定されていない場合は、'data' が自動的にマウントポイントの後に追加されることに注意してください。
HashiCorp の場合、VaultPrefix が指定されていない場合は、'data' が自動的にマウントポイントの後に追加されることに注意してください。

Hashicorp のプレフィックス例:

```
v1/secret/data/zabbix/
```

Cyberark のプレフィックス例:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイルの名前です。証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL キーのファイルフィールドを空にしてください。
 このファイルを含むディレクトリは、SSLCertLocation 構成パラメータで指定します。
 このオプションは省略できますが、CyberArkCCP ボールトでは推奨されます。

VaultTLSKeyFile

クライアント認証に使用する SSL 秘密鍵ファイルの名前。
 秘密鍵ファイルは PEM1 形式でなければなりません。
 このファイルを含むディレクトリは、SSLKeyLocation 設定パラメータで指定します。
 このオプションは省略可能ですが、CyberArkCCP ボールトでは推奨されます。

VaultToken

Zabbix プロキシ専用に生成され、オプションの VaultDBPath 構成パラメータで指定されたパスへの読み取り専用権限を持つ HashiCorp vault 認証トークン。
VaultToken と VAULT_TOKEN 環境変数が同時に定義されている場合はエラーとなります。

必須: Vault が *HashiCorp* に設定されている場合は必須、それ以外は不要

VaultURL

Vault サーバーの HTTP[S] URL。SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリサイズ。
VMware の内部チェック zabbix[vmware,buffer,...] を使用して VMware キャッシュの使用状況を監視できます（内部チェックを参照）。
 開始するように設定された vmware コレクターインスタンスがない場合、共有メモリは割り当てられません。

デフォルト: 8M
 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからデータを収集する間隔 (秒単位)。
 この間隔は、VMware 監視アイテムの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンタの統計情報を取得するまでの秒数の遅延。
 この遅延は、VMware パフォーマンスカウンタを使用する VMware 監視itemの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

VMware コレクターが VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待つ最大秒数。

デフォルト: 10
 範囲: 1-300

WebDriverURL

WebDriver インターフェースの HTTP[S] URL。

例 (Selenium WebDriver スタンドアロン・サーバーと併用):

```
WebDriverURL=http://localhost:4444
```

脚注

[1] プロキシの設定では、DBSocket と DBPort は排他的です。どちらか一方のみを指定するか、両方とも未定義にしてください。

3 Zabbix エージェント (UNIX)

概要

Zabbix エージェントの設定ファイル（zabbix_agentd.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしでリストされています。パラメータをクリックすると、詳細情報が表示されます。

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowRoot	エージェントを'root' として実行することを許可します。
BufferSend	バッファ内のデータを N 秒以上保持しません。
BufferSize	メモリバッファ内の値の最大数です。
DebugLevel	デバッグレベルです。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可するかどうか。
HeartbeatFrequency	ハートビートメッセージの頻度（秒単位）です。
HostInterface	ホストインターフェースを定義するオプションパラメータです。
HostInterfaceItem	ホストインターフェースを取得するために使用するアイテムを定義するオプションパラメータです。
HostMetadata	ホストメタデータを定義するオプションパラメータです。
HostMetadataItem	ホストメタデータを取得するために使用する Zabbix エージェントアイテムを定義するオプションパラメータです。
Hostname	ホスト名を定義するオプションパラメータです。
HostnameItem	ホスト名を取得するために使用する Zabbix エージェントアイテムを定義するオプションパラメータです。
Include	個別のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。
ListenBacklog	TCP キュー内の保留中の接続の最大数です。
ListenIP	エージェントがリッスンするカンマ区切りの IP アドレスのリストです。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LoadModule	エージェント起動時にロードするモジュールです。
LoadModulePath	エージェントモジュールの場所へのフルパスです。
LogFile	ログファイル名です。
LogFileSize	ログファイルの最大サイズです。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録します。
LogType	ログ出力のタイプです。
MaxLinesPerSecond	'log' および'logrt' のアクティブチェックを処理する際に、エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシに送信する新しい行の最大数です。
PidFile	PID ファイル名です。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度です。
Server	カンマ区切りの IP アドレス（CIDR 表記も可）、または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリストです。
ServerActive	アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ設定です。
SourceIP	送信元 IP アドレスです。
StartAgents	パッシブチェックを処理する zabbix_agentd の事前フォークインスタンス数です。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する秒数を指定します。
TLSAccept	受け入れる着信接続の種類です。
TLSCAFile	ピア証明書の検証に使用する、最上位 CA 証明書を含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信に使用されます。

パラメータ	説明
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	GnuTLS の優先文字列または OpenSSL（TLS 1.2）の暗号文字列です。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列です。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS の優先文字列または OpenSSL（TLS 1.2）の暗号文字列です。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列です。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS の優先文字列または OpenSSL（TLS 1.2）の暗号文字列です。PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列です。PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。
TLSCRLFile	失効した証明書を含むファイルのフルパス名です。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	Zabbix サーバーとの暗号化通信に使用する、エージェントの事前共有鍵を含むファイルのフルパス名です。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信に使用する事前共有鍵の識別文字列です。
TLSServerCertIssuer	許可されるサーバー（プロキシ）の証明書発行者です。
TLSServerCertSubject	許可されるサーバー（プロキシ）の証明書サブジェクトです。
UnsafeUserParameters	ユーザー定義パラメータへの引数にすべての文字を渡すことを許可します。
User	システム上の特定の既存ユーザーに権限をドロップします。
UserParameter	監視するユーザー定義パラメータです。
UserParameterDir	UserParameter コマンドのデフォルトの検索パスです。

パラメータが必須であると明示的に記載されていない限り、すべてのパラメータは必須ではありません。

注意:

- デフォルト値はデーモンのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- ”#” で始まるコメントは行頭のみサポートされます。

パラメータの詳細

Alias

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えるために使用できます。
 複数の *Alias* パラメータを指定できます。同じ *Alias* キーの複数のパラメータは許可されていません。
 異なる *Alias* キーが同じアイテムキーを参照することができます。
 エイリアスは *HostMetadataItem* で使用できますが、*HostnameItem* パラメータでは使用できません。

例 1: ユーザー'zabbix' の ID を取得する。

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

これで、**zabbix.userid** の短縮キーを使用してデータを取得できます。

例 2: デフォルトおよびカスタムパラメータで CPU 使用率を取得する。

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

これにより、**cpu.util** キーを使用してデフォルトパラメータで CPU 使用率のパーセンテージを取得したり、**cpu.util[all, idle, avg15]** を使用して CPU 使用率に関する特定のデータを取得したりできます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、**vfs.fs.discovery** を使用して、各ルールに異なるパラメータ (**vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など) で複数のディスカバリルールを設定できるようになりました。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、”*” 文字を使用して任意の文字列に一致させることができます。
DenyKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述された順に 1 つずつ処理されます。参考: エージェントチェックの制限

AllowRoot

エージェントを’root’ として実行することを許可します。無効にした場合、エージェントが’root’ によって開始された場合、代わりに’zabbix’ ユーザーに切り替えようとします。通常のユーザーで開始した場合は効果がありません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

BufferSend

N 秒を超えてバッファ内にデータを保持しません。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンはワイルドカード式であり、”*” 文字を使用して任意の文字列に一致させることができます。
AllowKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述順に 1 つずつ処理されます。参考: エージェントチェックの制限

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください。
 値に応じて AllowKey/DenyKey パラメータの内部エイリアスです。
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

デフォルト: 0
 値: 0 - 許可しない, 1 - 許可する

HeartbeatFrequency

ハートビートメッセージの頻度（秒単位）。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストの自動登録プロセス中に使用されるホストインターフェース（IP アドレスまたは DNS 名）を定義するオプションパラメーターです。この値は新しく作成されたホストのインターフェースに設定され、IP または DNS アドレスのいずれかを明示的に設定できます。詳細はDNS をデフォルトインターフェースとして使用するを参照してください。

定義されていない場合は、HostInterfaceItem から値が取得されます。

値が 255 文字を超える場合、エージェントはエラーを出して起動しません。

範囲：0～255 文字

HostInterfaceItem

ホストの自動登録プロセス中にホストインターフェース（IP アドレスまたは DNS 名）を決定するために使用されるアイテムを定義するオプションパラメーターです。この値は、HostInterface が定義されていない場合にのみ使用されます。詳細については、DNS をデフォルトインターフェースとして使用するを参照してください。

自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録プロセス (アクティブエージェント) 中にホストを識別または区別するために使用されるメタデータを定義するオプションパラメータです。HostMetadata を使用すると、ホスト名を超えてホストを区別できます。

定義されていない場合、値は HostMetadataItem から取得されます。

指定された値が 2034 バイトを超える場合や、UTF-8 以外の文字列の場合、エージェントはエラーを発行し、起動しません。パラメータが IP アドレスまたは DNS 名を期待する場合、UTF-8 として有効でも IP または DNS 名として無効な値は拒否され、無効として報告されます。

範囲: 0-2034 バイト

HostMetadataItem

Zabbix エージェントアイテムを定義するオプションパラメータで、ホストメタデータの取得に使用されます。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。HostMetadataItem を使用すると、ホスト名以外でホストを区別できます。

ユーザーパラメータとエイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

自動登録リクエスト中に、指定されたアイテムによって返される値が 65535 UTF-8 コードポイントの制限を超えている場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列でなければならず、そうでない場合は無視されます。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は拒否され、無効として報告されます。

Hostname

カンマ区切りの一意で大文字と小文字を区別するホスト名のリスト。アクティブチェックに必要で、サーバーで設定されたホスト名と一致する必要があります。未定義の場合は HostnameItem から値が取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-' 最大長: ホスト名ごとに 128 文字、1 行全体で 2048 文字。

デフォルト: HostnameItem で設定

HostnameItem

ホスト名を取得するために使用される Zabbix エージェントアイテムを定義するオプションパラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、AllowKey/DenyKey の値に関係なく system.run[]アイテムはサポートされています。

デフォルト: `system.hostname`

Include

個々のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。指定したディレクトリ内の関連ファイルのみを含めるために、パターンマッチング用のアスタリスクワイルドカード文字がサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' で自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

エージェントが待受する IP アドレスのカンマ区切りリスト

デフォルト: 0.0.0.0

ListenPort

エージェントは、サーバーからの接続をこのポートで待受します。

デフォルト: 10050
 範囲: 1024-32767

LoadModule

エージェントの起動時にロードするモジュール。モジュールは、エージェントの機能を拡張するために使用されます。モジュールは、LoadModulePath で指定されたディレクトリに配置するか、モジュール名の前にパスを付ける必要があります。先頭のパスが絶対パス（'/' で始まる）場合は、LoadModulePath は無視されます。
 フォーマッ

ト:
LoadModule=<module.so>
LoadModule=<path/module.so>
LoadModule=</abs_path/module.so>
 複数の LoadModule パラメータを含めることができます。

LoadModulePath

エージェントモジュールの場所へのフルパス。デフォルトはコンパイルオプションによって異なります。

LogFile

ログファイル名

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうかを設定します。コマンドはリモートで実行された場合のみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

MaxLinesPerSecond

'log' および'logrt' のアクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。指定された値は、'log' または'logrt' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテムで必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

PidFile

PID ファイルの名前

デフォルト: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

アクティブチェックのリストをリフレッシュする間隔 (秒)。アクティブチェックのリフレッシュに失敗した場合、次のリフレッシュは 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーおよび Zabbix プロキシの IP アドレス（カンマ区切り）、CIDR 表記、または DNS 名のリスト。ここにリストされたホストからの着信接続のみが受け入れられます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' を使用すると、任意の IPv4 アドレスを許可できます。「IPv4 互換 IPv6 アドレス」（0000::/96 プレフィックス）はサポートされていますが、RFC4291で非推奨となっています。スペースも許可されます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: StartAgents が明示的に 0 に設定されていない場合は必須

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ構成。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、コロンで区切られたオプションのポートです。
 クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシをカンマで区切って指定できます。各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定すべきではありません。Zabbix プロキシが指定されている場合、そのプロキシの Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りのアドレスを指定することで、複数の独立した

Zabbix サーバーを並列で使用できます。スペースも許可されます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでホストのポートを指定する場合は、角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーによる高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下のための送信元 IP アドレス:

- Zabbix サーバーまたは Zabbix プロキシへの外部接続
- 一部のアイテム（web.page.get、net.tcp.port など）を実行する際の接続

StartAgents

パッシブチェックを処理する zabbix_agentd の事前フォークインスタンス数。0 に設定するとパッシブチェックが無効になり、エージェントは TCP ポートで待ち受けしません。

デフォルト: 10
 範囲: 0-100

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。

このパラメータは、以下を含むさまざまな通信操作の期間を定義します。

- Zabbix サーバーからの応答を待つ
- アイテム設定リクエストやアクティブチェックのアイテムデータなど、Zabbix サーバーへのリクエスト送信
- logfile を介したログデータの取得
- ハートビートメッセージの送信
- `vfs.*` チェックの最大継続時間
- Zabbix エージェントモジュールによる使用
- タイムアウトなしでチェックを送信するバージョン 7.0 より前のサーバーまたはプロキシからのフォールバックとして機能

このタイムアウトは、フロントエンドでタイムアウト設定が可能なエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる着信接続の種類。パッシブチェックに使用されます。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
psk - TLS および事前共有鍵（PSK）による接続を受け入れる
cert - TLS および証明書による接続を受け入れる

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも必須）、それ以外は不要

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 での OpenSSL 1.1.1 以降の暗号文字列。証明書ベースおよび PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先順位文字列または OpenSSL（TLS 1.2）の暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントが Zabbix サーバーまたはプロキシにどのように接続するかを指定します。アクティブチェックで使用されます。1 つの値のみ指定できます:
unencrypted - 暗号化せずに接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）；それ以外は不要

TLSCRLFile

失効した証明書を含むファイルのフルパス名です。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータへの引数としてすべての文字の使用を許可します。次の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 さらに、改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

ユーザー

システム上の特定の既存ユーザーの権限を削除します。
'root' として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: zabbix

UserParameter

監視するユーザー定義パラメータ。ユーザー定義パラメータは複数指定できます。
 フォーマット: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返してはいけません。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスを使用できます。

例:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。このパラメータを使用すると、エージェントはコマンドを実行する前にここで指定された作業ディレクトリに移動します。これにより、UserParameter コマンドはフルパスの代わりに相対パスの ./プレフィックスを使用できます。
1 つだけ指定できます。

例:

```
UserParameterDir=/opt/myscripts
```

参考

1. バージョン 2.0.0 以降のアクティブチェックとパッシブチェックの Zabbix エージェント設定の違い

4 Zabbix エージェント **2 (UNIX)**

概要

Zabbix agent 2 は Zabbix エージェントの新世代であり、Zabbix エージェントの代わりに使用できます。

Zabbix agent 2 の設定ファイル（zabbix_agent2.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしで一覧表示されています。パラメータをクリックすると詳細が表示されます。

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファ内のデータを N 秒以上保持しません。
BufferSize	メモリバッファ内の最大値数。
ControlSocket	'-R' オプションでランタイムコマンドを送信するためのコントロールソケット。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテムのローカル永続ストレージの使用を有効にします。
ForceActiveChecksOnStart	再起動後、最初に受信した設定に対してアクティブチェックを即座に実行します。
HeartbeatFrequency	ハートビートメッセージの頻度（秒単位）。
HostInterface	ホストインターフェースを定義するオプションパラメータ。
HostInterfaceItem	ホストインターフェースを取得するために使用されるアイテムを定義するオプションパラメータ。
HostMetadata	ホストメタデータを定義するオプションパラメータ。
HostMetadataItem	ホストメタデータを取得するために使用される Zabbix エージェントアイテムを定義するオプションパラメータ。
Hostname	ホスト名を定義するオプションパラメータ。
HostnameItem	ホスト名を取得するために使用される Zabbix エージェントアイテムを定義するオプションパラメータ。
Include	設定ファイルに個別ファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenIP	エージェントがリッスンするカンマ区切りの IP アドレスリスト。

パラメータ	説明
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
PersistentBufferFile	Zabbix agent 2 が SQLite データベースを保持するファイル。
PersistentBufferPeriod	サーバーまたはプロキシへの接続がない場合にデータを保存する期間。
PidFile	PID ファイル名。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェックの上限。
Plugins.Log.MaxLinesPerSecond	'log' および 'logrt' アクティブチェック処理時に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたり送信する新しい行の最大数。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録するかどうか。
PluginSocket	ロード可能なプラグイン通信のための UNIX ソケットのパス。
PluginTimeout	ロード可能なプラグインとの接続タイムアウト（秒単位）。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度。
Server	カンマ区切りの IP アドレス（CIDR 表記または DNS 名も可）のリスト。Zabbix サーバーおよび Zabbix プロキシ。
ServerActive	アクティブチェックを取得するための Zabbix サーバー/プロキシアドレスまたはクラスタ設定。
SourceIP	送信元 IP アドレス。
StatusPort	設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待ち受けます。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）。
TLSAccept	受け入れる着信接続の種類。
TLSCAFile	ピア証明書の検証用の最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	OpenSSL 暗号文字列（TLS 1.2）。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列（TLS 1.3）。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	OpenSSL 暗号文字列（TLS 1.2）。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列（TLS 1.3）。証明書ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	OpenSSL 暗号文字列（TLS 1.2）。PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列（TLS 1.3）。PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェントの事前共有鍵を含むファイルのフルパス名。Zabbix サーバーとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列。Zabbix サーバーとの暗号化通信に使用されます。
TLSServerCertIssuer	許可されるサーバー（プロキシ）証明書の発行者。
TLSServerCertSubject	許可されるサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

パラメータが必須であると明示的に記載されていない限り、すべてのパラメータは必須ではありません。

注意:

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- ”#” で始まるコメントは行頭でのみサポートされます。

パラメータの詳細

Alias

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えるために使用できます。
 複数の *Alias* パラメータを指定できます。同じ *Alias* キーの複数のパラメータは許可されません。
 異なる *Alias* キーが同じアイテムキーを参照することができます。
 エイリアスは *HostMetadataItem* で使用できますが、*HostnameItem* パラメータでは使用できません。

例 1: ユーザー'zabbix' の ID を取得する。

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

これで、**zabbix.userid** の省略キーを使用してデータを取得できます。

例 2: デフォルトおよびカスタムパラメータで CPU 使用率を取得する。

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

これにより、**cpu.util** キーを使用してデフォルトパラメータで CPU 使用率のパーセンテージを取得したり、**cpu.util[all, idle, avg15]** を使用して CPU 使用率に関する特定のデータを取得したりできます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、**vfs.fs.discovery** を使用して各ルールに異なるパラメータを指定した複数のディスカバリルールを設定できるようになりました。例: **vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、"*" 文字を使用して任意の文字列に一致させることができます。
DenyKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述された順に 1 つずつ処理されます。参考: エージェントチェックの制限

BufferSend

バッファから Zabbix サーバーに値が送信される頻度を決定する秒単位の時間間隔です。バッファがいっぱいの場合は、データがより早く送信されることに注意してください。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。このパラメータは、永続バッファが無効になっている場合（*EnablePersistentBuffer=0*）のみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

コントロールソケット、'-R' オプションを指定したランタイムコマンドの送信に使用します。

デフォルト: `/tmp/agent.sock`

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンはワイルドカード式であり、"*" 文字を使用して任意の文字列に一致させることができます。
AllowKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述された順に 1 つずつ処理されます。参考: エージェントチェックの制限

EnablePersistentBuffer

アクティブアイテムのローカル永続ストレージの使用を有効にします。永続ストレージが無効な場合、メモリバッファが使用されます。

デフォルト: 0
 値: 0 - 無効, 1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した設定に対してアクティブチェックを即座に実行します。また、プラグインごとの設定パラメータとしても利用可能です。例: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

デフォルト: 0
 値: 0 - 無効, 1 - 有効

HeartbeatFrequency

ハートビートメッセージの頻度（秒単位）。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストの自動登録プロセス中に使用されるホストインターフェース（IP アドレスまたは DNS 名）を定義するオプションパラメーターです。この値は新しく作成されたホストのインターフェースに設定され、IP または DNS アドレスのいずれかを明示的に設定できます。詳細はデフォルトインターフェースとして DNS を使用するを参照してください。

定義されていない場合は、HostInterfaceItem から値が取得されます。

値が 255 文字を超える場合、エージェントはエラーを出して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録プロセス中にホストインターフェース（IP アドレスまたは DNS 名）を決定するために使用されるアイテムを定義するオプションパラメーターです。この値は、HostInterface が定義されていない場合にのみ使用されます。詳細については、DNS をデフォルトインターフェースとして使用するを参照してください。

自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録プロセス中にホストを識別または区別するために使用されるメタデータを定義するオプションパラメータです。HostMetadata を使用すると、ホスト名以外でもホストを区別できます。

定義されていない場合は、HostMetadataItem から値が取得されます。

指定された値が 2034 バイトを超える場合や、UTF-8 以外の文字列の場合、エージェントはエラーを出して起動しません。パラメータが IP アドレスまたは DNS 名を期待する場合、UTF-8 としては有効でも IP や DNS 名として無効な値は、無効として拒否され報告されます。

範囲: 0-2034 バイト

HostMetadataItem

Zabbix エージェントアイテムを定義するオプションパラメータで、ホストメタデータの取得に使用されます。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録プロセスでのみ使用されます。HostMetadataItem を使用すると、ホスト名以外でホストを区別できます。

ユーザーパラメータとエイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

自動登録リクエスト時に、指定されたアイテムによって返される値が 65535 UTF-8 コードポイントの制限を超えている場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列でなければならず、そうでない場合は無視されます。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は拒否され、無効として報告されます。

Hostname

カンマ区切りで指定された、ユニークで大文字小文字を区別するホスト名のリスト。アクティブチェックに必要であり、サーバーで設定されたホスト名と一致する必要があります。未定義の場合は HostnameItem から値が取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-' 最大長: ホスト名ごとに 128 文字、1 行全体で 2048 文字。

デフォルト: HostnameItem で設定

HostnameItem

ホスト名の取得に使用するアイテムを定義するオプションパラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、AllowKey/DenyKey の値に関係なくsystem.run[]アイテムはサポートされています。

デフォルト：system.hostname

Include

個々のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。インストール時に、Zabbix はコンパイル時に変更されていない限り、/usr/local/etc に include ディレクトリを作成します。パスは *zabbix_agent2.conf* ファイルの場所からの相対パスにすることができます。
 指定したディレクトリ内の関連ファイルのみを含めるために、パターンマッチング用のアスタリスクワイルドカード文字がサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

エージェントがリッスンする IP アドレスのカンマ区切りリスト。アクティブチェックのリストを取得するために Zabbix サーバーに接続する場合、最初の IP アドレスが Zabbix サーバーに送信されます。

デフォルト: 0.0.0.0

ListenPort

エージェントは、サーバーからの接続をこのポートで待ち受けします。

デフォルト: 10050
 範囲: 1024-32767

LogFile

ログファイル名

デフォルト: /tmp/zabbix_agent2.log
 必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: file

PersistentBufferFile

Zabbix agent 2 が SQLite データベースを保持するファイル。完全なファイル名でなければなりません。このパラメータは、永続バッファが有効になっている場合 (*EnablePersistentBuffer=1*) のみ使用されます。

PersistentBufferPeriod

サーバーまたはプロキシへの接続がない場合にデータを保存する期間。古いデータは失われます。ログデータは保持されます。このパラメータは、永続バッファが有効になっている場合 (*EnablePersistentBuffer=1*) にのみ使用されます。

デフォルト: 1h
 範囲: 1m-365d

PidFile

PID ファイルの名前

デフォルト: /tmp/zabbix_agent2.pid

Plugins.<PluginName>.System.Capacity

<PluginName> プラグインごとに同時に実行できるチェックの上限。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

'log' および'logrt' のアクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。指定された値は、'log' および'logrt' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテムで必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうかを設定します。コマンドはリモートで実行された場合のみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信用の UNIX ソケットへのパス

デフォルト: `/tmp/agent.plugin.sock`

PluginTimeout

読み込み可能なプラグインとの接続のタイムアウト（秒単位）

デフォルト: `Timeout`
 範囲: 1-30

RefreshActiveChecks

アクティブチェックのリストをリフレッシュする間隔 (秒)。アクティブチェックのリフレッシュに失敗した場合、次のリフレッシュは 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーまたは Zabbix プロキシの IP アドレス (カンマ区切り、CIDR 表記も可) または DNS 名のリスト。ここに記載されたホストからの接続のみを受け付けます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。スペースも許可されます。このパラメータが指定されていない場合、パッシブチェックは無効になり、エージェントは TCP ポートで待ち受けしません。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ構成を指定します。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、コロンで区切られたオプションのポート番号です。
 クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシをカンマで区切って指定できます。各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定すべきではありません。Zabbix プロキシを指定した場合、そのプロキシの Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りのアドレスを指定することで、複数の独立した Zabbix サーバーを並列で利用できます。スペースも許可されます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでホストのポートを指定する場合は、角括弧で囲む必要があります。ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータを指定しない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーによる高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下のための送信元 IP アドレス:

- Zabbix サーバーまたは Zabbix プロキシへの外部接続
- 一部のアイテム (web.page.get、net.tcp.port など) の実行時の接続

StatusPort

設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待受します。

範囲: 1024-32767

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。

このパラメータは、以下を含むさまざまな通信操作の期間を定義します。

- Zabbix サーバーからの応答を待つ
- アイテム設定リクエストやアクティブチェックのアイテムデータなど、Zabbix サーバーへのリクエスト送信
- logfile を介したログデータの取得
- ハートビートメッセージの送信
- `vfs.*` チェックの最大実行時間
- タイムアウトなしでチェックを送信するバージョン 7.0 より前のサーバーまたはプロキシとのやり取り時のフォールバック

このタイムアウトは、フロントエンドでタイムアウト設定が可能なエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる着信接続。パッシブチェックで使用されます。複数の値をカンマで区切って指定できます:
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
psk - TLS および事前共有鍵（PSK）による接続を受け入れる
cert - TLS および証明書による接続を受け入れる

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも必要）；それ以外は不要

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

OpenSSL の暗号文字列（TLS 1.2）。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列（TLS 1.3）。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

OpenSSL の暗号文字列（TLS 1.2）。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

このパラメータは `TLSAccept=cert,psk` と一緒に使用できないことに注意してください。証明書接続（`TLSConnect=cert`）の場合は、代わりに TLSCipherAll を使用してください。

TLSCipherCert13

OpenSSL 1.1.1 以降の OpenSSL 暗号文字列（TLS 1.3）。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

このパラメータは `TLSAccept=cert,psk` と一緒に使用できないことに注意してください。証明書接続（`TLSConnect=cert`）の場合は、代わりに TLSCipherAll13 を使用してください。

TLSCipherPSK

OpenSSL の暗号文字列（TLS 1.2）。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列（TLS 1.3）。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントが Zabbix サーバーまたはプロキシにどのように接続するかを指定します。アクティブチェックで使用されます。指定できる値は 1 つだけです。
unencrypted - 暗号化せずに接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）。それ以外は不要

TLSCRLFile

失効した証明書を含むファイルのフルパス名です。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。次の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 さらに、改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。ユーザー定義パラメータは複数指定できます。
 フォーマット: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返してはいけません。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスを使用できます。

例:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。このパラメータを使用すると、エージェントはコマンドを実行する前にここで指定した作業ディレクトリに移動します。これにより、UserParameter コマンドはフルパスの代わりに相対パス ./プレフィックスを使用できます。
1 つだけ指定できます。

例:

```
UserParameterDir=/opt/myscripts
```

5 Zabbix エージェント **(Windows)**

概要

Windows Zabbix エージェントの設定ファイル（zabbix_agentd.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしでリストされています。パラメータをクリックすると詳細が表示されます。

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファ内のデータを N 秒以上保持しません。
BufferSize	メモリバッファ内の最大値数。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可するかどうか。
HeartbeatFrequency	ハートビートメッセージの頻度（秒単位）。
HostInterface	ホストインターフェースを定義するオプションパラメータ。
HostInterfaceItem	ホストインターフェースを取得するために使用するアイテムを定義するオプションパラメータ。
HostMetadata	ホストメタデータを定義するオプションパラメータ。
HostMetadataItem	ホストメタデータを取得するために使用する Zabbix エージェントアイテムを定義するオプションパラメータ。
Hostname	ホスト名を定義するオプションパラメータ。
HostnameItem	ホスト名を取得するために使用する Zabbix エージェントアイテムを定義するオプションパラメータ。
Include	個別のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。
ListenBacklog	TCP キュー内の保留中の接続の最大数。
ListenIP	エージェントがリッスンするカンマ区切りの IP アドレスのリスト。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録します。
LogType	ログ出力の種類。
MaxLinesPerSecond	'log' および'logrt' アクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。
PerfCounter	指定した時間（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。
PerfCounterEn	指定した時間（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter パスは英語でなければなりません。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度。
Server	カンマ区切りの IP アドレス（オプションで CIDR 表記）、または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリスト。
ServerActive	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスタ構成。
SourceIP	送信元 IP アドレス。
StartAgents	パッシブチェックを処理する zabbix_agentd の事前フォークインスタンス数。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。
TLSAccept	受け入れる着信接続の種類。
TLSCAFile	ピア証明書の検証用の最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有鍵を含むファイルのフルパス名。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信に使用される事前共有鍵の識別文字列。
TLSServerCertIssuer	許可されたサーバー（プロキシ）証明書の発行者。
TLSServerCertSubject	許可されたサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータへの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

パラメータが必須であると明示的に記載されていない限り、すべてのパラメータは必須ではありません。

注意:

- デフォルト値はデーモンのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は UTF-8 エンコーディング（BOMなし）のみの設定ファイルをサポートします。
- ”#” で始まるコメントは行頭のみサポートされます。

パラメータの詳細

Alias

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、短くて簡単なものに置き換えるために使用できます。
 複数の *Alias* パラメータを指定できます。同じ *Alias* キーの複数のパラメータは許可されていません。
 異なる *Alias* キーが同じアイテムキーを参照することができます。
 エイリアスは *HostMetadataItem* で使用できますが、*HostnameItem* や *PerfCounter* パラメータでは使用できません。

例 1: サーバーからページファイルの使用率（パーセンテージ）を取得する。

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

この場合、短縮キー **pg_usage** を使用してデータを取得できます。

例 2: デフォルトおよびカスタムパラメータで CPU 負荷を取得する。

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

これにより、**cpu.load** キーを使用してデフォルトパラメータで CPU 負荷を取得したり、**cpu.load[percpu,avg15]** を使用して CPU 負荷の特定データを取得したりできます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、**vfs.fs.discovery** を使用して、各ルールごとに異なるパラメータ（例：**vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など）で複数のディスカバリルールを設定できます。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、”*” 文字を使用して任意の文字列に一致させることができます。
DenyKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述された順に 1 つずつ処理されます。参考: エージェントチェックの制限

BufferSend

N 秒を超えてバッファ内にデータを保持しません。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンはワイルドカード式で、「*」文字を使用して任意の文字列に一致させることができます。
AllowKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述順に 1 つずつ処理されます。参考: エージェントチェックの制限

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください。
 値に応じて AllowKey/DenyKey パラメータの内部エイリアスです。
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

デフォルト: 0
 値: 0 - 許可しない, 1 - 許可する

HeartbeatFrequency

ハートビートメッセージの頻度（秒単位）。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストの自動登録プロセス中に使用されるホストインターフェース（IP アドレスまたは DNS 名）を定義するオプションパラメーターです。この値は新しく作成されたホストのインターフェースに設定され、IP または DNS アドレスのいずれかを明示的に設定できます。詳細はDNS をデフォルトインターフェースとして使用するを参照してください。

定義されていない場合は、HostInterfaceItem から値が取得されます。

値が 255 文字を超える場合、エージェントはエラーを出して起動しません。

範囲: 0～255 文字

HostInterfaceItem

ホストの自動登録プロセス中にホストインターフェース（IP アドレスまたは DNS 名）を決定するために使用されるアイテムを定義するオプションパラメーターです。この値は、HostInterface が定義されていない場合にのみ使用されます。詳細については、デフォルトインターフェースとして DNS を使用するを参照してください。

自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[]アイテムは、AllowKey/DenyKey 設定に関係なくサポートされます。

HostMetadata

ホストの自動登録プロセス（アクティブエージェント）中にホストを識別または区別するために使用されるメタデータを定義するオプションパラメータです。HostMetadata を使用すると、ホスト名以外でホストを区別できます。

定義されていない場合は、HostMetadataItem から値が取得されます。

指定された値が 2034 バイトを超える場合や、UTF-8 以外の文字列の場合、エージェントはエラーを出して起動しません。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は無効として拒否され、報告されます。

複数行のメタデータはサポートされていません。出力は最初の改行で切り捨てられます。

範囲: 0-2034 バイト

HostMetadataItem

Zabbix エージェントアイテムを定義するオプションパラメータで、ホストメタデータの取得に使用されます。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録試行時に取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。HostMetadataItem を使用すると、ホスト名以外でもホストを区別できます。

ユーザーパラメータ、パフォーマンスカウンタ、エイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

自動登録リクエスト時に、指定されたアイテムによって返される値が 65535 UTF-8 コードポイントの制限を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列でなければならず、そうでない場合は無視されます。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされていません — 出力は最初の改行で切り捨てられます。

Hostname

カンマ区切りで指定された、ユニークで大文字小文字を区別するホスト名のリスト。アクティブチェックに必要であり、サーバーで設定されたホスト名と一致する必要があります。未定義の場合は HostnameItem から値が取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-' 最大長: ホスト名ごとに 128 文字、1 行全体で 2048 文字。

デフォルト: HostnameItem で設定

HostnameItem

ホスト名の取得に使用する Zabbix エージェントアイテムを定義するオプションパラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータ、パフォーマンスカウンタ、エイリアスはサポートされていませんが、system.run[]アイテムは AllowKey/DenyKey の値に関係なくサポートされています。
より詳細な説明も参照してください。

デフォルト: system.hostname

Include

個別のファイルやディレクトリ内のすべてのファイルを、設定ファイルに含めることができます（Zabbix エージェントが Windows MSI インストーラーパッケージを使用してインストールされた場合はデフォルトで C:\Program Files\Zabbix Agent にあります。Zabbix エージェントが zip アーカイブとしてインストールされた場合は、インストール時に指定したフォルダにあります）。含めるすべてのファイルは正しい構文でなければならず、そうでない場合はエージェントが起動しません。
 指定したディレクトリ内の関連ファイルのみを含めるために、アスタリスクワイルドカード文字によるパターンマッチングがサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' で自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

エージェントが待受する IP アドレスのカンマ区切りリスト

デフォルト: 0.0.0.0

ListenPort

エージェントは、サーバーからの接続をこのポートで待受します。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイル名。

デフォルト: c:\zabbix_agentd.log
 必須: LogType が *file* に設定されている場合は必須、それ以外は不要

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうかを設定します。コマンドはリモートで実行された場合のみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効, 1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - Windows イベントログにログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: file

MaxLinesPerSecond

'log'、'logrt'、'eventlog' のアクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数です。指定された値は、'log'、'logrt'、'eventlog' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注意: Zabbix は、*MaxLinesPerSecond* で設定された値の 10 倍の新しい行を処理して、ログアイテムで必要な文字列を検索します。

デフォルト: 20
 範囲: 1-1000

PerfCounter

指定された期間 <period>（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。
 構文: <parameter_name>,"<perf_counter_path>",<period>

例えば、過去 1 分間のプロセッサ割り込み数の平均値を取得したい場合は、次のように新しいパラメータ"interrupts" を定義できます:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターパスの周囲に二重引用符が必要であることに注意してください。パラメータ名（interrupts）は、アイテム作成時のアイテムキーとして使用します。平均値を計算するためのサンプルは毎秒取得されます。
Windows で利用可能なすべてのパフォーマンスカウンターのリストを取得するには、”typeperf -qx” を実行してください。

PerfCounterEn

指定された時間間隔 <period>（秒単位）で、システムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter パスは英語でなければなりません。**Windows Server 2008/Vista** 以降でのみサポートされています。
 構文: <parameter_name>,”<perf_counter_path>”,<period>

たとえば、過去 1 分間のプロセッサ割り込み数の平均値を取得したい場合は、次のように新しいパラメータ”interrupts” を定義できます:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンタパスの周囲に二重引用符が必要であることに注意してください。パラメータ名（interrupts）は、アイテム作成時のアイテムキーとして使用します。平均値を計算するためのサンプルは、毎秒取得されます。
 英語の文字列のリストは、次のレジストリキーを参照してください: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`。

RefreshActiveChecks

アクティブチェックのリストをリフレッシュする間隔 (秒)。アクティブチェックのリフレッシュに失敗した場合、次のリフレッシュは 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーまたは Zabbix プロキシの IP アドレス（カンマ区切り、CIDR 表記も可）または DNS 名のリスト。ここに記載されたホストからの接続のみ受け付けます。IPv6 サポートが有効な場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ はすべての IPv4 または IPv6 アドレスを許可します。’0.0.0.0/0’ はすべての IPv4 アドレスを許可するために使用できます。「IPv4 互換 IPv6 アドレス」（0000::/96 プレフィックス）はサポートされていますが、RFC4291で非推奨となっていますのでご注意ください。スペースも許可されます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: StartAgents が明示的に 0 に設定されていない場合は必須

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ構成。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、コロンで区切られたオプションのポートです。
 クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシをカンマで区切って指定できます。各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定すべきではありません。Zabbix プロキシが指定されている場合、そのプロキシの Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りのアドレスを指定することで、複数の独立した Zabbix サーバーを並列で使用できます。スペースも許可されます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでホストのポートを指定する場合は、角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーによる高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

範囲: (*)

SourceIP

以下のための送信元 IP アドレス:

- Zabbix サーバーまたは Zabbix プロキシへの外部接続
- 一部のアイテム (web.page.get、net.tcp.port など) を実行する際の接続

StartAgents

パッシブチェックを処理する zabbix_agentd の事前フォークインスタンスの数。0 に設定すると、パッシブチェックが無効になり、エージェントは TCP ポートで待ち受けしません。

デフォルト: 10
 範囲: 0-100 (*)

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換の待機時間（秒単位）を指定します。

このパラメータは、以下を含むさまざまな通信操作の期間を定義します。

- Zabbix サーバーからの応答を待つ
- アイテム設定リクエストやアクティブチェックでのアイテムデータなど、Zabbix サーバーへのリクエスト送信
- logfile または Windows イベントログ監視によるログデータの取得
- ハートビートメッセージの送信
- `vfs.*` チェックの最大実行時間
- Zabbix エージェントモジュールで使用
- バージョン 7.0 より前のサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとして使用

このタイムアウトは、フロントエンドでタイムアウト設定が可能なエージェントチェック（グローバル、プロキシ、アイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる着信接続。パッシブチェックで使用されます。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
psk - TLS と事前共有鍵（PSK）による接続を受け入れる
cert - TLS と証明書による接続を受け入れる

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）。それ以外は不要

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSConnect

エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。アクティブチェックで使用されます。1 つの値のみ指定できます:
unencrypted - 暗号化せずに接続（デフォルト）
psk - TLS および事前共有鍵（PSK）を使用して接続
cert - TLS および証明書を使用して接続

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）; それ以外は不要

TLSCRLFile

失効した証明書を含むファイルのフルパス名です。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータへの引数としてすべての文字の使用を許可します。次の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 さらに、改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。ユーザー定義パラメータは複数指定できます。
 フォーマット: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返してはいけません。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスでも指定できます。

例:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。使用する場合、エージェントはコマンドを実行する前にここで指定された作業ディレクトリに変更します。これにより、UserParameter コマンドはフルパスの代わりに相対的な ./プレフィックスを持つことができます。1 つのエントリのみ許可されます。

例:

```
UserParameterDir=C:\opt\myscripts
```

Note:
(*) ServerActive にリストされているアクティブサーバーの数と、StartAgents で指定されたパッシブチェック用のプリフォークインスタンスの数の合計は 64 未満でなければなりません。

6 Zabbix エージェント 2 (Windows)

概要

Zabbix agent 2 は Zabbix エージェントの新世代であり、Zabbix エージェントの代わりに使用することができます。

Windows Zabbix agent 2 の設定ファイル（zabbix_agent2.conf）でサポートされているパラメータをこのセクションに一覧表示します。

パラメータは追加情報なしでリストされています。パラメータをクリックすると、詳細が表示されます。

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファ内のデータを N 秒以上保持しません。
BufferSize	メモリバッファ内の値の最大数です。
ControlSocket	'-R' オプションでランタイムコマンドを送信するために使用されるコントロールソケットです。
DebugLevel	デバッグレベルです。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテムのローカル永続ストレージの使用を有効にします。
ForceActiveChecksOnStart	再起動後、最初に受信した設定に対してアクティブチェックを即座に実行します。
HeartbeatFrequency	ハートビートメッセージの頻度（秒単位）です。
HostInterface	ホストインターフェースを定義するオプションのパラメータです。
HostInterfaceItem	ホストインターフェースを取得するために使用されるアイテムを定義するオプションのパラメータです。
HostMetadata	ホストメタデータを定義するオプションのパラメータです。
HostMetadataItem	ホストメタデータを取得するために使用される Zabbix エージェントアイテムを定義するオプションのパラメータです。
Hostname	ホスト名を定義するオプションのパラメータです。
HostnameItem	ホスト名を取得するために使用される Zabbix エージェントアイテムを定義するオプションのパラメータです。
Include	設定ファイルに個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenIP	エージェントがリッスンすべきカンマ区切りの IP アドレスのリストです。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名です。
LogFileSize	ログファイルの最大サイズです。

パラメータ	説明
LogType	ログ出力のタイプです。
PersistentBufferFile	Zabbix agent 2 が SQLite データベースを保持するファイルです。
PersistentBufferPeriod	サーバーまたはプロキシへの接続がない場合にデータを保存する期間です。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェックの上限です。
Plugins.Log.MaxLinesPerSecond	'log' および 'logrt' のアクティブチェック処理時に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたり送信する新しい行の最大数です。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録するかどうかを有効にします。
PluginSocket	ロード可能なプラグイン通信のための UNIX ソケットのパスです。
PluginTimeout	ロード可能なプラグインとの接続のタイムアウト（秒単位）です。
PerfCounter	指定した期間 <period>（秒単位）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。
PerfCounterEn	指定した期間 <period>（秒単位）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter パスは英語でなければなりません。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度です。
Server	カンマ区切りの IP アドレス（オプションで CIDR 表記）、または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリストです。
ServerActive	アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ設定です。
SourceIP	送信元 IP アドレスです。
StatusPort	設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待ち受けます。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。
TLSAccept	受け入れる着信接続の種類です。
TLSCAFile	ピア証明書の検証のための最上位 CA 証明書を含むファイルのフルパス名で、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名で、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll13	OpenSSL 暗号文字列（TLS 1.3）。証明書ベースおよび PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL 暗号文字列（TLS 1.3）。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL 暗号文字列（TLS 1.3）。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシにどのように接続するかを指定します。
TLSCRLFile	失効した証明書を含むファイルのフルパス名です。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルのフルパス名で、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有鍵を含むファイルのフルパス名です。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信に使用される事前共有鍵の識別文字列です。
TLSServerCertIssuer	許可されたサーバー（プロキシ）証明書の発行者です。
TLSServerCertSubject	許可されたサーバー（プロキシ）証明書のサブジェクトです。
UnsafeUserParameters	ユーザー定義パラメータへの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータです。
UserParameterDir	UserParameter コマンドのデフォルトの検索パスです。

パラメータが必須であると明示的に記載されていない限り、すべてのパラメータは必須ではありません。

注意:

- デフォルト値はプロセスのデフォルト値を反映しており、出荷時の設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- "#" で始まるコメントは行頭でのみサポートされます。

パラメータの詳細

Alias

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、短くて簡単なものに置き換えるために使用できます。

複数の *Alias* パラメータを指定できます。同じ *Alias* キーを持つ複数のパラメータは許可されていません。
 異なる *Alias* キーが同じアイテムキーを参照することは可能です。
 エイリアスは *HostMetadataItem* で使用できますが、*HostnameItem* パラメータでは使用できません。

例 1: サーバーからページングファイルの使用率 (%) を取得する。

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

これで、短縮キー **pg_usage** を使用してデータを取得できます。

例 2: デフォルトおよびカスタムパラメータで CPU 負荷を取得する。

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

これにより、**cpu.load** キーを使用してデフォルトパラメータで CPU 負荷を取得したり、**cpu.load[percpu,avg15]** を使用して CPU 負荷の特定のデータを取得したりできます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、各ルールに異なるパラメータ (**vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など) を指定して、**vfs.fs.discovery** を使用する複数のディスカバリルールを設定できます。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、”*” 文字を使用して任意の文字列に一致させることができます。
DenyKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、出現順に 1 つずつ処理されます。参考: エージェントチェックの制限。

BufferSend

バッファから Zabbix サーバーへ値を送信する間隔（秒単位）です。
 バッファがいっぱいの場合、データはより早く送信されます。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいの場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。
 このパラメータは、永続バッファが無効になっている場合 (*EnablePersistentBuffer=0*) のみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

’-R’ オプションを指定したランタイムコマンドの送信に使用する制御ソケット

デフォルト: `\\.\pipe\agent.sock`

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンはワイルドカード式であり、”*” 文字を使用して任意の文字列に一致させることができます。
AllowKey と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記述された順に 1 つずつ処理されます。参考: エージェントチェックの制限

EnablePersistentBuffer

アクティブアイテムのローカル永続ストレージの使用を有効にします。永続ストレージが無効になっている場合は、メモリバッファが使用されます。

デフォルト: 0
 値: 0 - 無効, 1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した設定に対してアクティブチェックを直ちに実行します。また、プラグインごとの設定パラメータとしても利用可能です。例: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

デフォルト: 0
 値: 0 - 無効, 1 - 有効

HeartbeatFrequency

ハートビートメッセージの間隔（秒単位）。
 アクティブチェックの可用性を監視するために使用します。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストの自動登録プロセス中に使用されるホストインターフェース (IP アドレスまたは DNS 名) を定義するオプションパラメーターです。この値は新しく作成されたホストのインターフェースに設定され、IP または DNS アドレスのいずれかを明示的に設定できます。詳細はデフォルトインターフェースとして DNS を使用するを参照してください。

定義されていない場合は、HostInterfaceItem から値が取得されます。

値が 255 文字の制限を超える場合、エージェントはエラーを発行し、起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録プロセス中にホストインターフェース（IP アドレスまたは DNS 名）を決定するために使用されるアイテムを定義するオプションパラメーターです。この値は、HostInterface が定義されていない場合にのみ使用されます。詳細については、DNS をデフォルトインターフェースとして使用するを参照してください。

自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録プロセス（アクティブエージェント）中にホストを識別または区別するために使用されるメタデータを定義するオプションパラメータです。HostMetadata を使用すると、ホスト名以外でホストを区別できます。

定義されていない場合は、HostMetadataItem から値が取得されます。

指定された値が 2034 バイトを超える場合や UTF-8 以外の文字列の場合、エージェントはエラーを出して起動しません。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は無効として拒否され、報告されます。

複数行のメタデータはサポートされていません。出力は最初の改行で切り捨てられます。

範囲: 0-2034 バイト

HostMetadataItem

Zabbix エージェントアイテムを定義するオプションパラメータで、ホストメタデータの取得に使用されます。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。HostMetadataItem を使用すると、ホスト名以外でホストを区別できます。

ユーザーパラメータとエイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の設定に関係なくサポートされます。

自動登録リクエスト中に、指定されたアイテムによって返される値が 65535 UTF-8 コードポイントの制限を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列でなければならず、そうでない場合は無視されます。パラメータが IP アドレスまたは DNS 名を期待する場合、有効な UTF-8 であっても有効な IP または DNS 名でない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされていません — 出力は最初の改行で切り捨てられます。

Hostname

カンマ区切りで指定された一意の大文字小文字を区別するホスト名のリスト。アクティブチェックに必要であり、サーバーで設定されたホスト名と一致する必要があります。未定義の場合は HostnameItem から値が取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: ホスト名ごとに 128 文字、1 行あたり 2048 文字。

デフォルト: HostnameItem で設定

HostnameItem

ホスト名を取得するために使用されるアイテムを定義するオプションパラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、AllowKey/DenyKey の値に関係なくsystem.run[]アイテムはサポートされています。

デフォルト: `system.hostname`

Include

個別のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます（Zabbix エージェントが Windows MSI インストーラーパッケージを使用してインストールされた場合、デフォルトで `C:\Program Files\Zabbix Agent 2` にあります。Zabbix エージェントが zip アーカイブとしてインストールされた場合は、インストール時に指定したフォルダーにあります）。含めるすべてのファイルは正しい構文である必要があります。そうでない場合、エージェントは起動しません。パスは *zabbix_agent2.conf* ファイルの場所からの相対パスで指定できます（例：`Include=.\zabbix_agent2.d\plugins.d\*.conf`）。
 指定したディレクトリ内の関連ファイルのみを含めるために、アスタリスクワイルドカード文字によるパターンマッチングがサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

エージェントがリッスンする IP アドレスのカンマ区切りリスト。アクティブチェックのリストを取得するために Zabbix サーバーに接続する場合、最初の IP アドレスが Zabbix サーバーに送信されます。

デフォルト: `0.0.0.0`

ListenPort

エージェントは、サーバーからの接続をこのポートで待ち受けします。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイル名。

デフォルト: `c:\zabbix_agent2.log`
 必須: LogType が *file* に設定されている場合は必須、それ以外は不要

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
console - 標準出力にログを書き出します。

デフォルト: `file`

PersistentBufferFile

Zabbix agent 2 が SQLite データベースを保持するファイル。完全なファイル名でなければなりません。このパラメータは、永続バッファが有効になっている場合 (*EnablePersistentBuffer=1*) のみ使用されます。

PersistentBufferPeriod

サーバーまたはプロキシへの接続がない場合にデータを保存する期間。古いデータは失われます。ログデータは保持されます。このパラメータは、永続バッファが有効になっている場合（*EnablePersistentBuffer=1*）のみ使用されます。

デフォルト: 1h
 範囲: 1m-365d

Plugins.<PluginName>.System.Capacity

`<PluginName>` プラグインごとに同時に実行できるチェックの上限。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

'log'、'logrt'、'eventlog' のアクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。指定された値は、'log'、'logrt'、'eventlog' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテムで必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうかを指定します。コマンドはリモートで実行された場合のみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信用の UNIX ソケットへのパス

デフォルト: `\\.\pipe\agent.plugin.sock`

PluginTimeout

読み込み可能なプラグインとの接続のタイムアウト（秒単位）

デフォルト: `Timeout`
 範囲: 1-30

PerfCounter

指定された時間間隔 <period>（秒単位）に対して、システムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。
 構文: <parameter_name>,”<perf_counter_path>”,<period>

例えば、過去 1 分間のプロセッサ割り込み回数の平均値を取得したい場合は、次のように新しいパラメータ”interrupts” を定義できます。

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンタパスの周囲に二重引用符が必要であることに注意してください。パラメータ名（interrupts）は、アイテム作成時のアイテムキーとして使用します。平均値を計算するためのサンプルは毎秒取得されます。
Windows で利用可能なすべてのパフォーマンスカウンタのリストを取得するには、”typeperf -qx” を実行してください。

PerfCounterEn

指定された時間間隔 <period>（秒単位）に対して、システムパフォーマンスカウンター <perf_counter_path> の平均値を表す新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter のパスは英語でなければなりません。**Windows Server 2008/Vista** 以降でのみサポートされます。
 構文: <parameter_name>,”<perf_counter_path>”,<period>

例えば、過去 1 分間のプロセッサ割り込み回数の平均値を取得したい場合は、次のように新しいパラメータ”interrupts” を定義できます:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターパスの周囲のダブルクォーテーションに注意してください。パラメータ名(interrupts)は、アイテム作成時のアイテムキーとして使用します。平均値を計算するためのサンプルは毎秒取得されます。
 英語の文字列のリストは、次のレジストリキーを参照してください: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`。

RefreshActiveChecks

アクティブチェックのリストをリフレッシュする間隔 (秒)。アクティブチェックのリフレッシュに失敗した場合、次のリフレッシュは 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーまたは Zabbix プロキシの IP アドレス (カンマ区切り、CIDR 表記も可) または DNS 名のリスト。ここに記載されたホストからの接続のみ受け付けます。IPv6 サポートが有効な場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ は任意の IPv4 または IPv6 アドレスを許可します。’0.0.0.0/0’ は任意の IPv4 アドレスを許可するために使用できます。スペースも許可されます。このパラメータが指定されていない場合、パッシブチェックは無効になり、エージェントは TCP ポートで待ち受けしません。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレスまたはクラスタ構成。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、コロンで区切られたオプションのポートです。
 クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシをカンマで区切って指定できます。各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定すべきではありません。Zabbix プロキシが指定されている場合、そのプロキシの Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りのアドレスを指定することで、複数の独立した Zabbix サーバーを並列で使用できます。スペースも許可されます。
 ポートが指定されていない場合は、デフォルトのポートが

使用されます。
IPv6 アドレスでホストのポートを指定する場合は、角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーによる高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下のための送信元 IP アドレス:

- Zabbix サーバーまたは Zabbix プロキシへの外部接続
- 一部のアイテム (web.page.get、net.tcp.port など) を実行する際の接続

StatusPort

設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待受します。

範囲: 1024-32767

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒単位）を指定します。

このパラメータは、以下を含むさまざまな通信操作の期間を定義します。

- Zabbix サーバーからの応答を待つ
- アイテム設定リクエストやアクティブチェックでのアイテムデータなど、Zabbix サーバーへのリクエスト送信
- logfile または Windows イベントログ監視によるログデータの取得
- ハートビートメッセージの送信
- `vfs.*` チェックの最大継続時間
- タイムアウトなしでチェックを送信するバージョン 7.0 より前のサーバーまたはプロキシからのフォールバックとして機能

このタイムアウトは、フロントエンドでタイムアウト設定が可能なエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる着信接続。パッシブチェックに使用されます。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
psk - TLS および事前共有鍵（PSK）による接続を受け入れる
cert - TLS および証明書による接続を受け入れる

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続の場合でも）。それ以外の場合は不要

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll13

OpenSSL 暗号文字列（TLS 1.3）。証明書および PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

OpenSSL 暗号文字列（TLS 1.3）。証明書ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

このパラメータは TLSAccept=cert,psk と一緒に使用できないことに注意してください。証明書接続（TLSConnect=cert）の場合は、代わりに TLSCipherAll13 を使用してください。

TLSCipherPSK13

OpenSSL の暗号文字列（TLS 1.3）。PSK ベースの暗号化のためのデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。アクティブチェックで使用されます。1 つの値のみ指定できます:
unencrypted - 暗号化せずに接続（デフォルト）
psk - TLS および事前共有鍵（PSK）を使用して接続
cert - TLS および証明書を使用して接続

必須: TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）; それ以外は不要

TLSCRLFile

失効した証明書を含むファイルのフルパス名です。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメーターの引数にすべての文字を渡すことを許可します。次の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 さらに、改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。ユーザー定義パラメータは複数指定できます。
 フォーマット: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返してはいけません。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスを持つことができます。

例:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。使用する場合、エージェントはコマンドを実行する前にここで指定された作業ディレクトリに変更します。これにより、UserParameter コマンドはフルパスの代わりに相対的な .\プレフィックスを持つことができます。
1 つのエントリのみ許可されます。

例:

```
UserParameterDir=C:\opt\myscripts
```

7 Zabbix エージェント **2** プラグイン

概要

このセクションには、Zabbix エージェント 2 プラグインの設定ファイルパラメータの説明が含まれています。特定のプラグインに関する情報にアクセスするには、サイドバーを使用してください。

1 Ceph プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Ceph プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：`ceph.conf`）に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は UTF-8 エンコーディング（BOMなし）のみ設定ファイルとしてサポートします。
- 「#」で始まるコメントは、行頭でのみサポートされます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.Ceph.Default.ApiKey				Ceph への接続に使用するデフォルトの API キー。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Ceph.Default.Mode		native / restful	restful (非推奨)	Ceph への接続に使用するデフォルトのモード。 native モードは Linux および Ceph 16 以降でのみサポートされます。restful モードは Ceph バージョン 20 以降では動作しません。 ユーザー認証情報セットは各モードで異なり、互換性がないことに注意してください。
Plugins.Ceph.Default.User				Ceph への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Ceph.Default.Uri			https://localhost:8003	Ceph への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 `https` スキームのみサポートされます。スキームは省略可能です。 ポートは省略可能です（デフォルト =8003）。 例: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify		false / true	false	http クライアントがサーバーの証明書チェーンとホスト名を検証するかどうかを決定します。 *true* の場合、TLS はサーバーによって提示された任意の証明書と、その証明書内の任意のホスト名を受け入れます。このモードでは、TLS は中間者攻撃に対して脆弱です（テスト目的でのみ使用してください）。
Plugins.Ceph.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.Ceph.Sessions.<SessionName>.ApiKey				名前付きセッションの API キー。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Ceph.Sessions.<SessionName>.Mode		native / restful	restful (非推奨)	Ceph への接続に使用する名前付きセッションのモード。 native モードは Linux および Ceph 16 以降でのみサポートされます。restful モードは Ceph バージョン 20 以降では動作しません。 ユーザー認証情報セットは各モードで異なり、互換性がないことに注意してください。
Plugins.Ceph.Sessions.<SessionName>.User				名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメーター	必須	範囲	デフォルト	説明
Plugins.Ceph.Sessions.<SessionName>.Uri	いいえ			名前付きセッションの接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 `https` スキームのみサポートされます。スキームは省略可能です。 ポートは省略可能です（デフォルト =8003）。 例: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト（リクエストが完了するまでに待機する秒数）。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

2 Docker プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Docker プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル (例: `docker.conf`) に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点:

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみ設定ファイルをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker デーモンの unix ソケットの場所。 スキームが含まれている必要があります (`unix://`のみサポート)。
Plugins.Docker.Timeout	いいえ	1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの数秒の時間)。

参照:

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix エージェント 2 (UNIX) / Zabbix エージェント 2 (Windows)
- プラグインの設定手順

3 Ember+ プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Ember+ プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル（例：`emberplus.conf`）に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

Ember+ プラグインはロード可能なプラグインであり、Ember+ プラグインリポジトリで利用可能かつ完全に説明されています。

このプラグインは現在、ソースからビルドすることでのみ利用可能です（Unix および Windows の両方）。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- ”#” で始まるコメントは、行の先頭でのみサポートされます。

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テスト用にプラグインを起動します（プラグイン設定は無視されます）。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.EmberPlus.Default.Uri			tcp://localhost:9092	接続するデフォルトの URI。サポートされているスキーマは tcp:// のみです。スキーマは省略可能です。埋め込み認証情報は無視されます。
Plugins.EmberPlus.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9092	指定したセッション名で接続する URI。サポートされているスキーマは tcp:// のみです。スキーマは省略可能です。埋め込み認証情報は無視されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.EmberPlus.System.Path				Ember+ プラグイン実行ファイルへのパス。 使用例: Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-
Plugins.EmberPlus.Timeout		1-30	グローバルタイムアウト	最初の接続時およびセッション内の後続操作でサーバーの応答を待つ秒数。

参照:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

4 Memcached プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Memcached プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル (例: memcached.conf) に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

注意点:

- デフォルト値はプロセスのデフォルトを反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートしています。
- Zabbix は UTF-8 エンコーディングでBOMなしの設定ファイルのみをサポートしています。
- ”#” で始まるコメントは、行頭でのみサポートされています。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.Memcached.Default.Password	いいえ			Memcached への接続に使用するデフォルトのパスワード。アイテムキーや名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Memcached.Default.Uri	いいえ		tcp://localhost:11211	Memcached への接続に使用するデフォルトの URI。アイテムキーや名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 サポートされているスキーム：tcp、unix。スキームは省略可能です。 ポートは省略可能です（デフォルト =11211）。 例：tcp://localhost:11211 localhost unix:/var/run/memcached.sock
Plugins.Memcached.Default.User	いいえ			Memcached への接続に使用するデフォルトのユーザー名。アイテムキーや名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Memcached.KeepAlive	いいえ	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.Memcached.Sessions.<SessionName>.Password	いいえ			名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Memcached.Sessions.<SessionName>.Uri	いいえ			名前付きセッションの接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 サポートされているスキーム：tcp、unix。スキームは省略可能です。 ポートは省略可能です（デフォルト =11211）。 例：tcp://localhost:11211 localhost unix:/var/run/memcached.sock
Plugins.Memcached.Sessions.<SessionName>.User	いいえ			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Memcached.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト（リクエストが完了するまで待機する秒数。タイムアウト後にリクエストをシャットダウンします）。

また、以下も参照してください：

- 一般的な Zabbix agent 2 の設定パラメーターの説明：Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

5 Modbus プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Modbus プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：modbus.conf）に記述し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートしています。
- Zabbix はBOMなしの UTF-8 エンコーディングのみ設定ファイルとしてサポートします。

- 「#」で始まるコメントは行頭でのみサポートされます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Modbus.Sessions.<SessionName>.Endpoint				エンドポイントは、プロトコルスキーム、ホストアドレス、およびポートまたはシリアルポート名と属性からなる接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Modbus.Sessions.<SessionName>.SlaveID				名前付きセッションのスレーブ ID。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 例: `Plugins.Modbus.Sessions.MB1.SlaveID=20` この名前付きセッションパラメータは、アイテムキーのスレーブ ID パラメータで値が指定されていない場合のみチェックされることに注意してください。
Plugins.Modbus.Sessions.<SessionName>.Timeout				名前付きセッションのタイムアウト（秒単位）。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 例: `Plugins.Modbus.Sessions.MB1.Timeout=2`

リクエスト実行のタイムアウト（リクエストの完了を待つ時間）を設定する必要がある場合は、アイテムの設定フォームを使用してください。

関連情報:

- Zabbix agent 2 の一般的な設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

6 MongoDB プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、MongoDB プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：`mongo.conf`）に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

MongoDB プラグインはロード可能なプラグインであり、MongoDB プラグインリポジトリで利用可能で、完全に説明されています。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートしています。
- Zabbix はBOMなしの UTF-8 エンコーディングのみ設定ファイルをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テストのためにプラグインを起動します（プラグイン設定は無視されます）。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.MongoDB.Default.Password				MongoDB への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

パラメーター	必須	範囲	デフォルト	説明
Plugins.MongoDB.Default.Uri	いいえ			MongoDB への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 tcp スキームのみサポートされます。スキームは省略可能です。 ポートは省略可能です（デフォルト =27017）。 例: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`
Plugins.MongoDB.Default.User	いいえ			MongoDB への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MongoDB.KeepAlive	いいえ	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.MongoDB.Sessions.<SessionName>.Password	いいえ			名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	いいえ (Plugins.MongoDB.Sessions.<SessionName>.TLSConnect が verify_ca または verify_full に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信で使用するピア証明書の検証用の最上位 CA 証明書を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	いいえ (Plugins.MongoDB.Sessions.<SessionName>.TLSConnect が指定されている場合は必須			Zabbix agent 2 と監視対象データベース間の暗号化通信で使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	いいえ			Zabbix agent 2 と監視対象データベース間の通信の暗号化タイプ。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - TLS 接続を必須とする; *verify_ca* - 証明書を検証する; *verify_full* - 証明書と IP アドレスを検証する。 プラグインバージョン 1.2.1 以降でサポート
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	いいえ (Plugins.MongoDB.Sessions.<SessionName>.TLSConnect が指定されている場合は必須			Zabbix agent 2 と監視対象データベース間の暗号化通信で使用される、データベースの秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.Uri	いいえ			名前付きセッションの接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 tcp スキームのみサポートされます。スキームは省略可能です。 ポートは省略可能です（デフォルト =27017）。 例: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`
Plugins.MongoDB.Sessions.<SessionName>.User	いいえ			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメーター	必須	範囲	デフォルト	説明
Plugins.MongoDB.System.Path	いいえ			MongoDB プラグイン実行ファイルへのパス。 使用例: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト（リクエストが完了するまで待機する秒数。経過後はリクエストをシャットダウンします）。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

7 MQTT プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、MQTT プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：`mqtt.conf`）に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングのみ設定ファイルをサポートします。
- 「#」で始まるコメントは、行頭のみサポートされます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.MQTT.DefaultPassword	いいえ			MQTT への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.DefaultTLSCAFile	いいえ			Zabbix agent 2 と MQTT ブローカー間の暗号化通信のためのピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.DefaultTLSCertFile	いいえ			Zabbix agent 2 と MQTT ブローカー間の暗号化通信のためのエージェント証明書または証明書チェーンを含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.DefaultTLSKeyFile	いいえ			Zabbix agent 2 と MQTT ブローカー間の暗号化通信のための MQTT 秘密鍵を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.DefaultTopic	いいえ			MQTT サブスクリプションのデフォルトトピック。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				トピックにはワイルドカード（「+」「#」）を含めることができます。
				例: `path/to/file`
				`path/to/#`
				`path/+/topic`

パラメーター	必須	範囲	デフォルト	説明
Plugins.MQTT.DefaultUrl			tcp://localhost:1883	デフォルトの MQTT ブローカー接続文字列。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 クエリパラメータを含めないでください。 URL 形式に一致する必要があります。 サポートされているスキーム: tcp（デフォルト）、ws、tls。 スキームは省略可能です。 ポートは省略可能です（デフォルト =1883）。 例: tcp://host:1883 localhost ws://host:8080
Plugins.MQTT.DefaultUser				MQTT への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.Sessions.<SessionName>.Password				名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile				Zabbix agent 2 と MQTT ブローカー間の暗号化通信のためのピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				Zabbix agent 2 と MQTT ブローカー間の暗号化通信のためのエージェント証明書または証明書チェーンを含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				Zabbix agent 2 と MQTT ブローカー間の暗号化通信のための MQTT 秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.Topic				MQTT サブスクリプションの名前付きセッショントピック。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 トピックにはワイルドカード（「+」「#」）を含めることができます。 例: path/to/file path/to/# path/+/topic
Plugins.MQTT.Sessions.<SessionName>.Url				名前付きセッションの接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 クエリパラメータを含めないでください。 URL 形式に一致する必要があります。 サポートされているスキーム: tcp（デフォルト）、ws、tls。 スキームは省略可能です。 ポートは省略可能です（デフォルト =1883）。 例: tcp://host:1883 localhost ws://host:8080
Plugins.MQTT.Sessions.<SessionName>.User				名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

リクエストの実行タイムアウト（リクエストの完了を待機する時間）を設定する必要がある場合は、アイテムの設定フォームを使用してください。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

8 MSSQL プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、MSSQL プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：`mssql.conf`）に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

MSSQL プラグインはロード可能なプラグインであり、MSSQL プラグインリポジトリで利用可能で、完全に説明されています。

注意事項：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は UTF-8 エンコーディング（BOMなし）の設定ファイルのみをサポートします。
- 「#」で始まるコメントは、行の先頭でのみサポートされます。

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テストのためにプラグインを起動します（プラグイン設定は無視されます）。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.MSSQL.CustomQueriesDir			Unix システムの場合は`/usr/local/share/zabbix/custom-queries/mssql` Windows システムの場合は`*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL`（`*`は ProgramFiles 環境変数のドライブ名）	プラグインが実行できるカスタムクエリを含むユーザー定義の.sql ファイルが格納されているディレクトリへのファイルパスを指定します。プラグイン起動時に設定されたディレクトリ内のすべての.sql ファイルを読み込みます。つまり、カスタムクエリファイルへの変更は、プラグインを再起動するまで反映されません。プラグインは Zabbix agent 2 と一緒に起動および停止します。
Plugins.MSSQL.CustomQueriesEnabled			false	設定されている場合、`mssql.custom.query` アイテムキーの実行を有効にします。無効にすると、カスタムクエリディレクトリパスからクエリは読み込まれません。
Plugins.MSSQL.Default.CACertPath				MSSQL サーバーの証明書を発行した認証局（CA）の公開鍵証明書へのデフォルトのファイルパス。証明書は PEM 形式である必要があります。
Plugins.MSSQL.Default.Database				接続するデフォルトのデータベース名。

パラメータ	必須	範囲	デフォルト	説明
Plugins.MSSQL.Default.Encrypt				デフォルトの接続暗号化タイプを指定します。可能な値: *true* - プラグインとサーバー間のデータ送信が暗号化されます; *false* - ログインパケット以外のプラグインとサーバー間のデータ送信は暗号化されません; *strict* - TDS8を使用してプラグインとサーバー間のデータ送信が E2E で暗号化されます; *disable* - プラグインとサーバー間のデータ送信は暗号化されません。
Plugins.MSSQL.Default.HostNameInCertificate				デフォルトでの MSSQL サーバー証明書のコモンネーム（CN）。
Plugins.MSSQL.Default.Password				デフォルトで保護された MSSQL サーバーに送信されるパスワード。
Plugins.MSSQL.Default.TLSMinVersion				デフォルトで使用する最小 TLS バージョン。可能な値: `1.0`, `1.1`, `1.2`, `1.3`。
Plugins.MSSQL.Default.TrustServerCertificate				デフォルトでプラグインがサーバー証明書を検証せずに信頼するかどうか。可能な値: `true`, `false`。
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	デフォルトの接続 URI。サポートされているスキーマは `sqlserver://`のみです。スキーマは省略可能です。埋め込み認証情報は無視されます。
Plugins.MSSQL.Default.User				デフォルトで保護された MSSQL サーバーに送信されるユーザー名。
Plugins.MSSQL.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				指定したセッションの MSSQL サーバー証明書を発行した認証局（CA）の公開鍵証明書へのファイルパス。証明書は PEM 形式である必要があります。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Database				指定したセッションで接続するデータベース名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				指定したセッションの接続暗号化タイプを指定します。可能な値は: *true* - プラグインとサーバー間のデータ送信が暗号化されます; *false* - ログインパケット以外のプラグインとサーバー間のデータ送信は暗号化されません; *strict* - TDS8を使用してプラグインとサーバー間のデータ送信が E2E で暗号化されます; *disable* - プラグインとサーバー間のデータ送信は暗号化されません。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				指定したセッションの MSSQL サーバー証明書のコモンネーム（CN）。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Password				指定したセッションで保護された MSSQL サーバーに送信されるパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				指定したセッションで使用する最小 TLS バージョン。可能な値は: `1.0`, `1.1`, `1.2`, `1.3`。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				指定したセッションでプラグインがサーバー証明書を検証せずに信頼するかどうか。可能な値: `true`, `false`。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost:1433	指定したセッションの接続 URI。サポートされているスキーマは `sqlserver://`のみです。スキーマは省略可能です。埋め込み認証情報は無視されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメータ	必須	範囲	デフォルト	説明
Plugins.MSSQL.Sessions.<SessionName>.User				指定したセッションで保護された MSSQL サーバーに送信されるユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.System.Path				MSSQL プラグイン実行ファイルへのパス。 MSSQL プラグインのグローバル設定。すべての接続に適用されます。 使用例: Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen
Plugins.MSSQL.Timeout		1-30	グローバルタイムアウト	最初の接続時およびセッション内の後続操作でサーバーの応答を待機する秒数。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

9 MySQL プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、MySQL プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：mysql.conf）に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は UTF-8 エンコーディング（BOMなし）のみ設定ファイルとしてサポートします。
- 「#」で始まるコメントは、行頭でのみサポートされます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Mysql.CallTimeout		1-30	グローバルタイムアウト	リクエストが完了するまで待機する最大時間（秒単位）。
Plugins.Mysql.CustomQueriesPath			Unix システムの場合は/usr/local/share/zabbix/custom-queries/mysql Windows システムの場合は *:\Program Files\Zabbix Agent 2\Custom Queries\Mysql (* は ProgramFiles 環境変数のドライブ名)	カスタムクエリを含む.sql ファイルが格納されているディレクトリのフルパス名。
Plugins.Mysql.CustomQueriesEnabled			false	設定すると、mysql.custom.query アイテムキーの実行が有効になります。無効の場合、カスタムクエリディレクトリパスからクエリはロードされません。

パラメータ	必須	範囲	デフォルト	説明
Plugins.Mysql.Default.Password	いいえ			MySQL への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSCAFile	いいえ (Plugins.Mysql.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSCertFile	いいえ (Plugins.Mysql.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのエージェント証明書または証明書チェーンを含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSConnect	いいえ			Zabbix agent 2 と監視対象データベース間の通信の暗号化タイプ。名前付きセッションで値が指定されていない場合に使用されます。 サポートされる値: *required* - TLS 接続を必須とする; *verify_ca* - 証明書を検証する; *verify_full* - 証明書と IP アドレスを検証する。
Plugins.Mysql.Default.TLSKeyFile	いいえ (Plugins.Mysql.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのデータベース秘密鍵を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Mysql.Default.Uri	いいえ		tcp://localhost:3306	MySQL への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（無視されます）。
				URI 形式に一致する必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略可能です。
				ポートは省略可能です（デフォルト =3306）。
				例: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	いいえ			MySQL への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.KeepAlive	いいえ	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.Mysql.Sessions.<SessionName>.Password	いいえ			名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメータ	必須	範囲	デフォルト	説明
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	セッション (Plugins.Mysql.Sessions.<SessionName>.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	セッション (Plugins.Mysql.Sessions.<SessionName>.TLSCertFile が指定されている場合は必須			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのエージェント証明書または証明書チェーンを含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	セッション			Zabbix agent 2 と監視対象データベース間の通信の暗号化タイプ。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - TLS 接続を必須とする; *verify_ca* - 証明書を検証する; *verify_full* - 証明書と IP アドレスを検証する。
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	セッション (Plugins.Mysql.Sessions.<SessionName>.TLSCertFile が指定されている場合は必須			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのデータベース秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.Mysql.Sessions.<SessionName>.Uri	セッション			名前付きセッションの接続文字列。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（無視されます）。
				URI 形式に一致する必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略可能です。
				ポートは省略可能です（デフォルト =3306）。
				例: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	セッション			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Timeout	いいえ	1-30	グローバルタイムアウト	接続が確立されるまでの最大待機時間（秒単位）。

また、以下も参照してください:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

10 NVIDIA GPU プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、NVIDIA GPU プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル（例：`nvidiagpu.conf`）に指定し、このファイルを Zabbix agent 2 の設定に追加するためにIncludeディレクティブを使用することを推奨します。

NVIDIA GPU プラグインはロード可能なプラグインであり、NVIDIA GPU プラグインリポジトリで利用可能で、完全に説明されています。

注意事項:

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートしています。
- Zabbix はBOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートしています。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.NVIDIA.System.Path	いいえ			NVIDIA GPU プラグイン実行ファイルへのパス。 使用例: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout	いいえ	1-30	グローバルタイムアウト	接続試行時およびセッション内の後続操作時にサーバー応答を待機する最大時間（秒単位）を指定します。グローバルなアイテムタイプのタイムアウトまたは個々のアイテムタイムアウトがこの値より大きい場合は、それらが優先されます。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

11 Oracle プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Oracle プラグインの操作に対応しています。

これらのパラメータは独自の設定ファイル（例：`oracle.conf`）に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- Zabbix は UTF-8 エンコーディング（BOMなし）のみの設定ファイルをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.Oracle.CallTimeout	いいえ	1-30	グローバルタイムアウト	リクエストが完了するまでの最大待機時間（秒単位）。
Plugins.Oracle.ConnectTimeout	いいえ	1-30	グローバルタイムアウト	接続が確立されるまでの最大待機時間（秒単位）。

パラメーター	必須	範囲	デフォルト	説明
Plugins.Oracle.CustomQueriesPath			Unix システムの場合は/usr/local/share/zabbix/oracle/custom_queries、Windows システムの場合は *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle、ここで * は ProgramFiles 環境変数のドライブ名	カスタムクエリを含む.sql ファイルが格納されているディレクトリのフルパス名。 例: /etc/zabbix/custom_queries/oracle
Plugins.Oracle.CustomQueriesEnabled			false	設定されている場合、oracle.custom.query アイテムキーの実行を有効にします。無効の場合、カスタムクエリディレクトリパスからクエリはロードされません。
Plugins.Oracle.Default.Password				Oracle への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Oracle.Default.Service				Oracle への接続に使用するデフォルトのサービス名（SID はサポートされません）。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Oracle.Default.Uri				Oracle への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 tcp および tcps スキームのみがサポートされます。スキームは省略可能です。 ポートも省略可能です（デフォルト =1521）。 TNS キーまたは TNS 値を接続文字列として指定することも可能です。TNS 値は空白なしで構成する必要があります。 例: tcp://127.0.0.1:1521 localhost zbx_tns_example（TNS キー） (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS 値)
Plugins.Oracle.Default.User				Oracle への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Oracle.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.Oracle.ResolveTNS		true / false	true	Oracle サーバーへの接続時に接続文字列（ConnString）をどのように解釈するかを指定します。 false に設定すると、デフォルトのスキームとポートが ConnString に追加されます（存在しない場合）。true に設定すると、デフォルトのスキームとポートは省略され（ConnString で明示的に指定されていない限り）、ConnString はそのまま Oracle クライアントに渡されます。Oracle クライアントが tnsnames.ora ファイル内で ConnString を見つけた場合、見つかった接続記述子が Oracle サーバーへの接続に使用されます。
Plugins.Oracle.Sessions.<SessionName>.Password				名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメーター	必須	範囲	デフォルト	説明
Plugins.Oracle.Sessions.<SessionName>.Service	いいえ			接続に使用する名前付きセッションのサービス名（SID はサポートされません）。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Oracle.Sessions.<SessionName>.Uri	いいえ			Oracle 用の名前付きセッション接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（無視されます）。 URI 形式に一致する必要があります。 `tcp` および `tcps` スキームのみがサポートされます。スキームは省略可能です。 ポートも省略可能です（デフォルト =1521）。 TNS キーまたは TNS 値を接続文字列として指定することも可能です。TNS 値は空白なしで構成する必要があります。 例: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example`（TNS キー） `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR`（TNS 値）
Plugins.Oracle.Sessions.<SessionName>.User	いいえ			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

参照:

- 一般的な Zabbix agent 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

12 PostgreSQL プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、PostgreSQL プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル (例: `postgresql.conf`) に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

PostgreSQL プラグインはロード可能なプラグインであり、PostgreSQL プラグインリポジトリで利用可能で、完全に説明されています。

注意:

- デフォルト値は、出荷時の設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートしています。
- Zabbix は、BOMなしの UTF-8 エンコーディングのみの設定ファイルをサポートします。
- ”#” で始まるコメントは、行頭のみサポートされます。

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テストのためにプラグインを起動します（プラグイン設定は無視されます）。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.PostgreSQL.Default.CacheMode			prepare	PostgreSQL 接続のキャッシュモード。 サポートされる値: *prepare*（デフォルト）- PostgreSQL サーバー上でプリペアドステートメントを作成します。 *describe* - サーバー上でステートメントを作成せずに、匿名のプリペアドステートメントを使用してステートメントを記述します。 「describe」は、PgBouncer のようなコネクションプーラーを実行している場合など、プリペアドステートメントが許可されていない環境で主に有用です。
Plugins.PostgreSQL.CallTimeout		1-30	グローバルタイムアウト	リクエストが完了するまでの最大待機時間（秒単位）。
Plugins.PostgreSQL.CustomQueriesPath			Unix システムの場合は`/usr/local/share/zabbix/custom-queries/postgreSQL` Windows システムの場合は`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL`（* は ProgramFiles 環境変数のドライブ名）	カスタムクエリを含む`.sql` ファイルが格納されているディレクトリのフルパス名。
Plugins.PostgreSQL.CustomQueriesEnabled			false	設定すると、`postgresql.custom.query` アイテムキーの実行を有効にします。無効にすると、カスタムクエリディレクトリパスからクエリは読み込まれません。
Plugins.PostgreSQL.Default.Database				PostgreSQL への接続に使用するデフォルトのデータベース。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.Password				PostgreSQL への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.TLSCAFile	(Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のためのピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.TLSCertFile	(Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のための PostgreSQL 証明書または証明書チェーンを含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。

パラメーター	必須	範囲	デフォルト	説明
Plugins.PostgreSQL.Default.TLSConnect	いいえ			Zabbix agent 2 と監視対象データベース間の通信の暗号化タイプ。名前付きセッションで値が指定されていない場合に使用されます。 サポートされる値: *required* - ID チェックなしで TLS をトランスポートモードとして使用して接続します。 *verify_ca* - TLS を使用して接続し、証明書を検証します。 *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID（CN）が証明書と一致することを検証します。 暗号化タイプが未定義の場合は、暗号化されていない接続となります。
Plugins.PostgreSQL.Default.TLSKeyFile	いいえ (Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			Zabbix agent 2 と監視対象データベース間の暗号化通信のための PostgreSQL 秘密鍵を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。

Plugins.PostgreSQL.Default.Uri	いいえ			PostgreSQL への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（無視されます）。
				URI 形式に一致する必要があります。
				サポートされるスキーム: `tcp`, `unix`。
				例: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	いいえ			PostgreSQL への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.KeepAlive	いいえ	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒単位）。
Plugins.PostgreSQL.Sessions.<SessionName>.CachePreparedMode	いいえ			PostgreSQL 接続のキャッシュモード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *prepare*（デフォルト）- PostgreSQL サーバー上でプリペアドステートメントを作成します。 *describe* - サーバー上でステートメントを作成せずに、匿名のプリペアドステートメントを使用してステートメントを記述します。 「describe」は、PgBouncer のようなコネクションプーラーを実行している場合など、プリペアドステートメントが許可されていない環境で主に有用です。
Plugins.PostgreSQL.Sessions.<SessionName>.Database	いいえ			セッション接続用のデータベース。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.Password	いいえ セッション名が形式に一致する必要があります。			セッション接続用のパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

パラメーター	必須	範囲	デフォルト	説明
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	いいえ (Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須)			最上位 CA 証明書ピア証明書検証を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	いいえ (Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect が指定されている場合は必須			PostgreSQL 証明書または証明書チェーンを含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	いいえ			PostgreSQL 接続の暗号化タイプ。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - ID チェックなしで TLS をトランスポートモードとして使用して接続します。 *verify_ca* - TLS を使用して接続し、証明書を検証します。 *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID（CN）が証明書と一致することを検証します。 暗号化タイプが未定義の場合は、暗号化されていない接続となります。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	いいえ (Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect が指定されている場合は必須			PostgreSQL 秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.PostgreSQL.Sessions.<SessionName>.Uri	いいえ			名前付きセッションの接続文字列。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（無視されます）。
				URI 形式に一致する必要があります。
				サポートされるスキーム: `tcp`, `unix`。
				例: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User	いいえ			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.System.Path	いいえ			PostgreSQL プラグイン実行ファイルへのパス。 使用例: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト（リクエストが完了するまで待機する秒数）。

参照:

- 一般的な Zabbix agent 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

13 Redis プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、Redis プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル（例：`redis.conf`）に指定し、Includeディレクティブを使用してこのファイルを Zabbix agent 2 の設定に追加することを推奨します。

注意点：

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートしています。
- Zabbix は UTF-8 エンコーディング（BOMなし）のみ設定ファイルとしてサポートします。
- 「#」で始まるコメントは、行頭でのみサポートされます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Redis.DefaultPassword				Redis への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Redis.DefaultTLSConnect				クライアント (Zabbix agent 2) の TLS 検証要件。 サポートされる値: `required` - 暗号化、未検証 (テストのみ) `verify_ca` - 暗号化、サーバ証明書 `verify_full` - 暗号化、サーバ証明書およびサーバ名を SAN で検証 空または未設定の場合は TLS なしとみなされます。
Plugins.Redis.DefaultTLSCAFile				トップレベル CA 証明書を含むファイルのデフォルトのフルパス名。
Plugins.Redis.DefaultTLSCertFile				エージェントの証明書を含むファイルのデフォルトのフルパス名。
Plugins.Redis.DefaultTLSKeyFile				エージェントの秘密鍵を含むファイルのデフォルトのフルパス名。

Plugins.Redis.DefaultUri			tcp://localhost:6379	Redis への接続に使用するデフォルトの URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報は含めないでください (無視されます)。
				URI 形式に一致する必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略可能です。
				ポートは省略可能 (デフォルト =6379)。
				例: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.DefaultUser			default	保護された Redis サーバーに送信するデフォルトのユーザー。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Redis.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)。
Plugins.Redis.Sessions.<SessionName>.Password				名前付きセッションのパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.TLSConnect				クライアント (Zabbix agent 2) の TLS 検証要件。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: `required` - 暗号化、未検証 (テストのみ) `verify_ca` - 暗号化、サーバ証明書 `verify_full` - 暗号化、サーバ証明書およびサーバ名を SAN で検証 空または未設定の場合は TLS なしとみなされます。

パラメータ	必須	範囲	デフォルト	説明
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	いいえ (TLSConnect が verify_ca または verify_full の場合は必須)			Redis サーバ証明書の検証用トップレベル CA 証明書を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	いいえ			クライアント認証用のエージェント証明書を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	いいえ			クライアント認証用のエージェント秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.Uri	いいえ		localhost:6379	名前付きセッションの接続文字列。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報は含めないでください (無視されます)。 URI 形式に一致する必要があります。 サポートされるスキーム: tcp, unix。スキームは省略可能です。 ポートは省略可能 (デフォルト =6379)。 例: tcp://localhost:6379 localhost unix:/var/run/redis.sock
Plugins.Redis.Sessions.<SessionName>.User	いいえ		default	保護された Redis サーバーに送信するユーザー。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト (リクエストが完了するまで待機する秒数)。

関連情報:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

14 SMART プラグイン

概要

Zabbix agent 2 の設定ファイルは、プラグインの設定に使用されます。これらの Zabbix agent 2 の設定パラメータは、SMART プラグインの操作をサポートしています。

これらのパラメータは独自の設定ファイル (例: smart.conf) に指定し、このファイルを Zabbix agent 2 の設定にIncludeディレクティブで追加することを推奨します。

注意点:

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。
- smartctl 実行ファイルへのパスは、そのディレクトリをシステムの PATH 環境変数に追加するか、Plugins.Smart.Path を設定することで指定する必要があります。これは Linux と Windows の両方に適用されます。
- Zabbix はBOMなしの UTF-8 エンコーディングのみ設定ファイルをサポートします。
- "#" で始まるコメントは、行頭でのみサポートされます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Smart.Path	いいえ		smartctl	smartctl 実行ファイルへのパス。
Plugins.Smart.Timeout	いいえ	1-30	グローバルタイムアウト	リクエスト実行のタイムアウト（リクエストが完了するまでに待機する秒数）。

参照:

- 一般的な Zabbix agent 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグインの設定手順

8 Zabbix Java ゲートウェイ

Zabbix Java ゲートウェイの起動に `startup.sh` および `shutdown.sh` スクリプトを使用する場合は、`settings.sh` ファイルで必要な設定パラメータを指定できます。起動スクリプトとシャットダウンスクリプトは設定ファイルを取得し、シェル変数（最初の列にリストされている）を Java プロパティ（2 番目の列にリストされている）に変換します。

`java` を直接実行して ZabbixJava ゲートウェイを手動で起動する場合は、コマンドラインで対応する Java プロパティを指定します。

|Variable|Property|Mandatory|Range|Default|Description| | 変数 | プロパティ | 必須 | 範囲 | デフォルト | 説明 | |--------|--------|---------|-----|-------|-----------| |LISTEN_IP|zabbix.listenIP|no|<|0.0.0.0| 待ち受け IP アドレス | |LISTEN_PORT|zabbix.listenPort|no|1024-32767|10052| 待ち受けポート番号 | |PID_FILE|zabbix.pidFile|no|<|/tmp/zabbix_java.pid|PID ファイル名。省略した場合、ZabbixJava ゲートウェイはコンソールアプリケーションとして起動されます。| |PROPERTIES_FILE|zabbix.propertiesFile|no|<|<| プロパティファイルの名前。コマンドラインに表示されないようにキー値形式を使用して追加のプロパティを設定したり、既存のプロパティを上書きしたりするために使用できます。
 例:"javax.net.ssl.trustStorePassword=<password>"| |START_POLLERS|zabbix.startPollers|no|1-1000|5| 開始するワーカースレッドの数 | |TIMEOUT|zabbix.timeout|no|1-30|3| ネットワーク操作を待機する時間 |

Warning:
ポート 10052 はIANA 登録済みではありません

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部 Web サービスとの通信に使用されるプロセスです。

このセクションでは、Zabbix Web サービスの設定ファイル（zabbix_web_service.conf）でサポートされているパラメータを一覧表示しています。

パラメータは追加情報なしでリストされています。パラメータをクリックすると、詳細が表示されます。

パラメータ	説明
AllowedIP	カンマ区切りの IP アドレス（CIDR 表記も可）または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリスト。
DebugLevel	デバッグレベル。
IgnoreURLCertErrors	フロントエンド URL にアクセスする際の TLS 証明書検証エラーの処理方法を指定します。
Include	設定ファイルに個別ファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenPort	サーバーからの接続をこのポートで待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
Timeout	ダッシュボードの PDFレポートのフォーマットに費やす最大時間（秒単位）。
TLSAccept	受け入れる着信接続の種類。
TLSCAFile	Zabbix コンポーネント間の暗号化通信で使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用される、サービス証明書または証明書チェーンを含むファイルのフルパス名
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用される、サービス秘密鍵を含むファイルのフルパス名。

パラメータは、必須であることが明示されていない限り、すべて任意です。

注意:

- デフォルト値はプロセスのデフォルト値を反映しており、配布されている設定ファイルの値ではありません。
- 値は環境変数をサポートします。

- Zabbix は、BOMなしの UTF-8 エンコーディングのみ設定ファイルをサポートします。
- 「#」で始まるコメントは、行頭のみサポートされます。

パラメータの詳細

AllowedIP

Zabbix サーバーおよび Zabbix プロキシの IP アドレス（カンマ区切り、CIDR 表記も可）または DNS 名のリスト。ここにリストされたホストからのみ接続を受け付けます。
IPv6 サポートが有効な場合、`127.0.0.1`、`::127.0.0.1`、`::ffff:127.0.0.1` は同等に扱われ、`::/0` は任意の IPv4 または IPv6 アドレスを許可します。任意の IPv4 アドレスを許可するには `0.0.0.0/0` を使用できます。

例:

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: yes

DebugLevel

デバッグレベルを指定します:
0 - Zabbix プロセスの開始と停止に関する基本情報
1 - 重大な情報;
2 - エラー情報;
3 - 警告;
4 - デバッグ用 (大量の情報を出力);
5 - 拡張デバッグ (さらに多くの情報を出力)。

デフォルト: 3
 範囲: 0-5

IgnoreURLCertErrors

フロントエンド URL にアクセスする際の TLS 証明書検証エラーの処理を指定します:
0 - 証明書エラーを無視しない;
1 - 証明書エラーを無視する。

デフォルト: 0
 範囲: 0-1

Include

個々のファイルやディレクトリ内のすべてのファイルを設定ファイルに含めることができます。インストール時に、Zabbix はコンパイル時に変更されていない限り、/usr/local/etc に include ディレクトリを作成します。パスは *zabbix_web_service.conf* ファイルの場所からの相対パスにすることができます。
 指定したディレクトリ内の関連ファイルのみを含めるために、アスタリスクワイルドカード文字によるパターンマッチングがサポートされています。
 制限事項については特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

サービスはこのポートでサーバーからの接続を待ち受けます。

デフォルト：10053
 範囲：1024-32767

LogFile

ログファイルの名前。

例:

```
/tmp/zabbix_web_service.log
```

必須: LogType が *file* に設定されている場合は必須。それ以外の場合は不要

LogFileSize

ログファイルの最大サイズ (MB 単位)。
0 - 自動ログローテーションを無効にします。
 注意: ログファイルサイズの上限に達し、何らかの理由でファイルのローテーションに失敗した場合、既存のログファイルは切り詰められ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力のタイプ：
file - LogFile パラメータで指定されたファイルにログを書き込む;
system - syslog にログを書き込む;
console - 標準出力にログを書き込む。

デフォルト：`file`

タイムアウト

ダッシュボードの PDFレポートのフォーマットに費やす最大時間（秒）。

デフォルト: 10
 範囲: 1-30

TLSAccept

受け入れる接続の種類：
unencrypted - 暗号化なしの接続を受け入れる（デフォルト）
cert - TLS および証明書付きの接続を受け入れる

デフォルト：`unencrypted`

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書の検証のための最上位 CA 証明書を含むファイルのフルパス名。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるサービス証明書または証明書チェーンを含むファイルのフルパス名。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるサービス秘密鍵を含むファイルのフルパス名。

10 環境変数

概要

環境変数を使用すると、設定ファイルに値をハードコーディングすることなく、Zabbix コンポーネントを設定できます。これにより、Docker などの動的な環境で設定を簡単に管理でき、Docker では、実行時に変数を渡すことでさまざまな設定に適応させることができます。

最も単純なケースでは、Zabbix サーバーのDebugLevel 設定パラメータ値を環境変数に設定することで、起動時にサーバーの設定に使用できます。

```
#### Zabbix サーバー設定ファイル:
DebugLevel=${NewDebugLevel}

#### Zabbix サーバーの起動:
NewDebugLevel=5 /usr/sbin/zabbix_server
```

環境変数は、以下の Zabbix コンポーネントでサポートされています。

- Server
- Proxy
- エージェント (UNIX またはWindows)
- エージェント 2 (UNIX またはWindows)、プラグインを含む
- Web サービス
- Zabbix Sender (-c, --config オプション 使用時)

重要事項

- 設定パラメータが環境変数に設定されており、コンポーネント実行時にその環境変数が指定されていない場合、パラメータのデフォルト値が使用されます。
- ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、以前に使用した環境変数を指定する必要があります。これは、Zabbix コンポーネントが設定ファイルを使用してランタイムコマンドを実行するためです。環境変数が省略されている場合、設定パラメータのデフォルト値が使用されます。例を参照してください。
- `userparameter_reload` ランタイムコマンドは、環境変数の再読み込みをサポートしていません。再読み込み中は変数は無視され、通常の値を持つパラメータのみが再読み込みされます。
- 設定ファイルで使用されていたプロセスの現在の環境変数は、Zabbix コンポーネントの起動後にクリアされます。これにより、子プロセス（例：Zabbix によって実行されるリモートスクリプト）はこれらの変数にアクセスできなくなります。ただし、プロセス初期変数は引き続き取得できることに注意してください。(たとえば、`/proc/<PID>/environ` ファイル経由)

構文

環境変数は、`${英数字/アンダースコア}` という構文を使用する必要があります。

変数名には、英字 (a-z、A-Z)、アンダースコア (_)、数字 (0-9) のみを使用できます。また、数字で始まってはなりません。

必須の構文に一致しない変数、または通常の値と組み合わせられた変数は通常の値として扱われ、エラーが発生する可能性があります。

正しい変数構文:

```
DebugLevel=${NewDebugLevel}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

誤った変数構文:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> Windows では、環境変数名の大文字と小文字は区別されません。

例

以下の例では、Zabbix コンポーネントでの環境変数の設定および使用方法を示しています。

例 1: Zabbix エージェントの設定とテスト

1. エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. 設定ファイルをテストします。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. 環境変数を有効にしてエージェントを起動します。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、以前に使用した環境変数を再度指定する必要があります。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

これは、エージェントがランタイムコマンドを実行するために設定ファイルを使用するためです。環境変数が省略されている場合、設定パラメータのデフォルト値が使用されます。

あるいは、エージェント設定ファイルで環境変数を設定した後、プロセスでそれらを有効に出来るようにすることもできます。（例：export コマンドを使用）これにより、変数の欠落や誤った設定による予期しない動作のリスクを軽減できます。

```
export ZBX_HOSTNAME=" 新しい Zabbix エージェント"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

例 2: コンテナ用の Zabbix エージェントの設定

Zabbix コンポーネント（例: Zabbix エージェント）用のカスタムイメージを独自に作成・設定する場合には、環境変数を使用して設定パラメータを定義し、その変数を有効にしてコンテナを起動できます。

1. イメージを準備する際に、エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. コンテナイメージをビルドしたら、環境変数を指定してエージェントコンテナ（例：Docker）を起動します。

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、コンテナシェルにアクセスし、以下のランタイムコマンドを実行します。

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

userparameter_reload ランタイムコマンドは、環境変数の再読み込みをサポートしていません。再読み込み中は変数は無視され、通常の値を持つパラメータのみが再読み込みされます。

11 インクルード

概要

追加のファイルやディレクトリは、Include パラメータを使用して、サーバー、プロキシ、エージェント、および Web サービスの設定に含めることができます。

インクルード時の注意

ファイルを含めるために Include パラメーターを使用する場合、ファイルは読み取り可能である必要があります。

ディレクトリを含めるために Include パラメーターが使用される場合:

- ディレクトリ内のすべてのファイルが読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません (たとえばファイルはアルファベット順で読み込まれません)。したがって複数の" インクルード" ファイルで 1 つのパラメーターを定義しないでください (たとえば特定の設定で一般設定をオーバーライドするため等)。
- ディレクトリ内のすべてのファイルが設定に含まれます。
- 一部のテキストエディタによって自動的に作成されるファイルバックアップコピーに注意してください。たとえば"include/my_specific.conf" ファイルを編集すると、バックアップコピー"include/my_specific_conf.BAK" が生成される場合、両方のファイルがインクルード対象に含まれます。"include/my_specific.conf.BAK" を"Include" ディレクトリから移動してください。Linux では"Include" ディレクトリの内容を"ls -al" コマンドで不要なファイルをチェックできます。

パターンを使用してファイルをインクルードするために Include パラメーターが使用される場合:

- パターンに一致するすべてのファイルは読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません (たとえばファイルはアルファベット順で読み込まれません)。したがって複数の" インクルード" ファイルで 1 つのパラメーターを定義しないでください (たとえば特定の設定で一般設定をオーバーライドするため等)。

3 プロトコル

サイドバーから、参照したい項目を選択してください。

1 サーバープロキシデータ交換プロトコル

概要

サーバー - プロキシ間のデータ交換は JSON 形式に基づいています。

要求メッセージと応答メッセージはヘッダーとデータの長さに則って始まる必要があります。

パッシブプロキシ

設定リクエスト

サーバーは最初に空の proxy config リクエストを送信します。このリクエストは ProxyConfigFrequency（サーバーの設定パラメータ）秒ごとに送信されます。

プロキシは現在のプロキシバージョン、セッショントークン、および設定リビジョンで応答します。サーバーは更新が必要な設定データで応答します。

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。
session	*string*	プロキシ設定セッショントークン。
config_revision	*number*	プロキシ設定リビジョン。
server→proxy:		
full_sync	*number*	フル設定データが送信される場合は 1、そうでない場合は省略（オプショ
data	*array*	テーブルデータのオブジェクト。設定が変更されていない場合は省略（オ
<table>	*object*	<table> データを持つ 1 つ以上のオブジェクト（変更に応じてオプショ
fields	*array*	フィールド名の配列。

name	value type	description
-	*string*	フィールド名。
data	*array*	行の配列。
-	*array*	列の配列。
-	*string,number*	データベーススキーマの列タイプに応じた型の列値。
macro.secrets	*object*	シークレットマクロ情報。ボールトマクロに変更がない場合は省略（オ
config_revision	*number*	設定キャッシュリビジョン - 設定データとともに送信（オプション）。
del_hostids	*array*	削除された hostid の配列（オプション）。
-	*number*	ホスト識別子。
del_macro_hostids	*array*	すべてのマクロが削除された hostid の配列（オプション）。
-	*number*	ホスト識別子。
proxy→server:		
response	*string*	リクエスト成功情報（'success' または'failed'）。
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。

例:

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
```

```
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
```

```
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
```

```
}
},
"config_revision": 2
}
```

proxy→server:

```
{
  "response": "success",
  "version": "8.0.0"
}
```

データリクエスト

`proxy data` リクエストは、プロキシからホストインターフェースの可用性、履歴、ディスカバリ、および自動登録データを取得するために使用されます。このリクエストは、`ProxyDataFrequency` (サーバーの設定パラメータ) 秒ごとに送信されます。

名前	値の型	説明
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	データセッションのトークン
interface availability	*array*	*(オプション)* インターフェース可用性データオブジェクトの配列
interfaceid	*number*	インターフェースの識別子
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースのエラーメッセージまたは空文字列
history data	*array*	*(オプション)* 履歴データオブジェクトの配列
itemid	*number*	アイテムの識別子
clock	*number*	アイテム値のタイムスタンプ (秒)
ns	*number*	アイテム値のタイムスタンプ (ナノ秒)
value	*string*	*(オプション)* アイテムの値
id	*number*	値の識別子 (データセッション内で一意な昇順カウンタ)
timestamp	*number*	*(オプション)* ログ型アイテムのタイムスタンプ
source	*string*	*(オプション)* イベントログアイテムのソース値
severity	*number*	*(オプション)* イベントログアイテムの重大度値
eventid	*number*	*(オプション)* イベントログアイテムの eventid 値
state	*string*	*(オプション)* アイテムの状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(オプション)* ログ型アイテムの最終ログサイズ
mtime	*number*	*(オプション)* ログ型アイテムの修正時刻
discovery data	*array*	*(オプション)* ディスカバリデータオブジェクトの配列
clock	*number*	ディスカバリデータのタイムスタンプ
druleid	*number*	ディスカバリルールの識別子
dcheckid	*number*	ディスカバリチェックの識別子、またはディスカバリルールデータの場合は null

名前	値の型	説明
type	*number*	ディスカバリチェックタイプ: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック
ip	*string*	ホストの IP アドレス
dns	*string*	ホストの DNS 名
port	*number*	*(オプション)* サービスのポート番号
key_	*string*	*(オプション)* タイプ **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(オプション)* サービスから受信した値。ほとんどのサービスでは空の場合あり
status	*number*	*(オプション)* サービスの状態: **0**, *DOBJECT_STATUS_UP* - サービス UP **1**, *DOBJECT_STATUS_DOWN* - サービス DOWN
auto registration	*array*	*(オプション)* 自動登録データオブジェクトの配列
clock	*number*	自動登録データのタイムスタンプ
host	*string*	ホスト名
ip	*string*	*(オプション)* ホストの IP アドレス
dns	*string*	*(オプション)* IP アドレスから解決された DNS 名
port	*string*	*(オプション)* ホストのポート
host_metadata	*string*	*(オプション)* エージェントから送信されたホストメタデータ (エージェントの HostMetadata または HostMetadataItem 設定パラメータに基づく)
tasks	*array*	*(オプション)* タスクの配列
type	*number*	タスクタイプ: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンドの結果
status	*number*	リモートコマンドの実行状態: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドが正常に完了 **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドが失敗
error	*string*	*(オプション)* エラーメッセージ
parent_taskid	*number*	親タスク ID
more	*number*	*(オプション)* 1 - 送信する履歴データがさらにある
clock	*number*	*(オプション)* データ転送のタイムスタンプ (秒)
ns	*number*	*(オプション)* データ転送のタイムスタンプ (ナノ秒)
version	*string*	プロキシのバージョン (<major>.<minor>.<build>)
server→proxy:		
response	*string*	リクエストの成功情報 ('success' または'failed')
tasks	*array*	*(オプション)* タスクの配列

名前	値の型	説明
type	*number*	タスクタイプ: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド
clock	*number*	タスク作成時刻
ttl	*number*	タスクの有効期限 (秒)
commandtype	*number*	リモートコマンドのタイプ: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用 (現在はカスタムスクリプトと機能的に同等)
command	*string*	実行するリモートコマンド
execute_on	*number*	カスタムスクリプトの実行対象: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェントでスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバーでスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシでスクリプトを実行
port	*number*	*(オプション)* Telnet および SSH コマンド用のポート
authtype	*number*	*(オプション)* SSH コマンド用の認証タイプ
username	*string*	*(オプション)* Telnet および SSH コマンド用のユーザー名
password	*string*	*(オプション)* Telnet および SSH コマンド用のパスワード
publickey	*string*	*(オプション)* SSH コマンド用の公開鍵
privatekey	*string*	*(オプション)* SSH コマンド用の秘密鍵
parent_taskid	*number*	親タスク ID
hostid	*number*	ターゲットホスト ID

例:

server→proxy:

```
{
  "request": "proxy data"
}
```

proxy→server:

```
{
    "session": "12345678901234567890123456789012"
    "interface availability": [
        {
            "interfaceid": 1,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 2,
            "available": 2,
            "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
    },
        {
            "interfaceid": 3,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 4,
            "available": 1,
            "error": ""
    }
    ],
```

```
    "history data":[
        {
            "itemid":"12345",
            "clock":1478609647,
            "ns":332510044,
            "value":"52956612",
            "id": 1
        },
        {
            "itemid":"12346",
            "clock":1478609647,
            "ns":330690279,
            "state":1,
            "value":"Cannot find information for this network interface in /proc/net/dev.",
            "id": 2
        }
    ],
    "discovery data":[
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":3,
            "type":12,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        },
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":null,
            "type":-1,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        }
    ],
    "auto registration":[
        {
            "clock":1478608371,
            "host":"Logger1",
            "ip":"10.3.0.1",
            "dns":"localhost",
            "port":"10050"
        },
        {
            "clock":1478608381,
            "host":"Logger2",
            "ip":"10.3.0.2",
            "dns":"localhost",
            "port":"10050"
        }
    ],
    "tasks":[
        {
            "type": 0,
            "status": 0,
            "parent_taskid": 10
        },
        {
            "type": 0,
            "status": 1,
```

```
            "error": "No permissions to execute task.",
            "parent_taskid": 20
        }
    ],
    "version":"8.0.0"
}
```

server→proxy:

```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
      }
  ]
}
```

アクティブプロキシ

設定リクエスト

`proxy config` リクエストは、プロキシの設定データを取得するためにアクティブプロキシから送信されます。このリクエストは、`ProxyConfigFrequency`（プロキシ設定パラメータ）秒ごとに送信されます。

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	プロキシ名
version	*string*	プロキシバージョン（<major>.<minor>.<build>）
session	*string*	プロキシ設定セッショントークン
config_revision	*number*	プロキシ設定リビジョン
server→proxy:		
fullsync	*number*	フル設定データが送信された場合は 1、それ以外は省略（オプション）
data	*array*	テーブルデータのオブジェクト。設定が変更されていない場合は省略（オ

name				value type	description
	<table>			*object*	1 つ以上の <table> データのオブジェクト（変更に応じてオプション）
		fields		*array*	フィールド名の配列
			-	*string*	フィールド名
		data		*array*	行の配列
			-	*array*	列の配列
			-	*string,number*	データベーススキーマの列タイプに応じた型の列値
macro.secrets				*object*	シークレットマクロ情報。ボールトマクロに変更がない場合は省略（オ
config_revision				*number*	設定キャッシュリビジョン - 設定データとともに送信（オプション）
del_hostids				*array*	削除された hostid の配列（オプション）
	-			*number*	ホスト識別子
del_macro_hostids				*array*	すべてのマクロが削除された hostid の配列（オプション）
	-			*number*	ホスト識別子

例:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
```

```
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
```

```
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


データリクエスト

proxy data リクエストは、プロキシがホストインターフェースの可用性、履歴、ディスカバリ、および自動登録データを提供するために送信されます。このリクエストは、DataSenderFrequency(プロキシの設定パラメータ) 秒ごとに送信されます。アクティブプロキシは、リモートコマンドタスクのために (空の proxy data リクエストで) 毎秒 Zabbix サーバーにポーリングし続けることに注意してください。

名前	値の型	説明
proxy→server:		
request	*string*	'proxy data'
host	*string*	プロキシ名
session	*string*	データセッションのトークン
interface availability	*array*	*(省略可能)* インターフェース可用性データオブジェクトの配列
interfaceid	*number*	インターフェース識別子
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースのエラーメッセージまたは空文字列
history data	*array*	*(省略可能)* 履歴データオブジェクトの配列
itemid	*number*	アイテム識別子
clock	*number*	アイテム値のタイムスタンプ (秒)
ns	*number*	アイテム値のタイムスタンプ (ナノ秒)
value	*string*	*(省略可能)* アイテム値
id	*number*	値の識別子 (昇順カウンタ、1 つのデータセッション内で一意)
timestamp	*number*	*(省略可能)* ログタイプアイテムのタイムスタンプ
source	*string*	*(省略可能)* イベントログアイテムのソース値
severity	*number*	*(省略可能)* イベントログアイテムの重大度値
eventid	*number*	*(省略可能)* イベントログアイテムの eventid 値
state	*string*	*(省略可能)* アイテムの状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(省略可能)* ログタイプアイテムの最終ログサイズ
mtime	*number*	*(省略可能)* ログタイプアイテムの修正時刻
discovery data	*array*	*(省略可能)* ディスカバリデータオブジェクトの配列
clock	*number*	ディスカバリデータのタイムスタンプ
druleid	*number*	ディスカバリルール識別子
dcheckid	*number*	ディスカバリチェック識別子、またはディスカバリルールデータの場合は null
type	*number*	ディスカバリチェックタイプ: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック

名前	値の型	説明
ip	*string*	ホストの IP アドレス
dns	*string*	ホストの DNS 名
port	*number*	*(省略可能)* サービスのポート番号
key_	*string*	*(省略可能)* タイプ **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(省略可能)* サービスから受信した値。ほとんどのサービスでは空の場合あり
status	*number*	*(省略可能)* サービスの状態: **0**, *DOBJECT_STATUS_UP* - サービス UP **1**, *DOBJECT_STATUS_DOWN* - サービス DOWN
autoregistration	*array*	*(省略可能)* 自動登録データオブジェクトの配列
clock	*number*	自動登録データのタイムスタンプ
host	*string*	ホスト名
ip	*string*	*(省略可能)* ホストの IP アドレス
dns	*string*	*(省略可能)* IP アドレスから解決された DNS 名
port	*string*	*(省略可能)* ホストのポート
host_metadata	*string*	*(省略可能)* エージェントから送信されたホストメタデータ (エージェントの HostMetadata または HostMetadataItem 設定パラメータに基づく)
tasks	*array*	*(省略可能)* タスクの配列
type	*number*	タスクタイプ: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンドの結果
status	*number*	リモートコマンドの実行ステータス: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドが正常に完了 **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドが失敗
error	*string*	*(省略可能)* エラーメッセージ
parent_taskid	*number*	親タスク ID
more	*number*	*(省略可能)* 1 - 送信する履歴データがさらにある
clock	*number*	*(省略可能)* データ転送のタイムスタンプ (秒)
ns	*number*	*(省略可能)* データ転送のタイムスタンプ (ナノ秒)
version	*string*	プロキシバージョン (<major>.<minor>.<build>)
server→proxy:		
response	*string*	リクエスト成功情報 ('success' または'failed')
upload	*string*	履歴データ (履歴、自動登録、ホスト可用性、ネットワークディスカバリ) のアップロード制御 可能な値: **enabled** - 通常動作 **disabled** - サーバーがデータを受け付けていない (内部キャッシュが上限を超えている可能性あり)
tasks	*array*	*(省略可能)* タスクの配列
type	*number*	タスクタイプ: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド
clock	*number*	タスク作成時刻
ttl	*number*	タスクの有効期限 (秒)
commandtype	*number*	リモートコマンドのタイプ: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用 (現在はカスタムスクリプトと機能的に同等)
command	*string*	実行するリモートコマンド

名前	値の型	説明
execute_on	*number*	カスタムスクリプトの実行対象: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェント上でスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバー上でスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシ上でスクリプトを実行
port	*number*	*(省略可能)* Telnet および SSH コマンド用のポート
authtype	*number*	*(省略可能)* SSH コマンド用の認証タイプ
username	*string*	*(省略可能)* Telnet および SSH コマンド用のユーザー名
password	*string*	*(省略可能)* Telnet および SSH コマンド用のパスワード
publickey	*string*	*(省略可能)* SSH コマンド用の公開鍵
privatekey	*string*	*(省略可能)* SSH コマンド用の秘密鍵
parent_taskid	*number*	親タスク ID
hostid	*number*	対象ホスト ID

例:

proxy→server:


```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
```

```
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```


2 Zabbix エージェント/エージェント **2** のプロトコル

Zabbix エージェントおよび Zabbix エージェント 2 のプロトコルの詳細については、パッシブおよびアクティブエージェントチェックページを参照してください。

4 Zabbix エージェント **2** プラグインプロトコル

Zabbix エージェント 2 プロトコルはコード、サイズ、データモデルに基づいています。

コード

タイプ	サイズ	コメント
Byte	4	ペイロードタイプ。現在サポートされているのは JSON のみです。

サイズ

タイプ	サイズ	コメント
バイト	4	現在のペイロードのサイズ（バイト単位）。

ペイロードデータ

タイプ	サイズ	コメント
バイト	*Size* フィールドで定義	JSON 形式のデータ。

ペイロードデータの定義

共通データ

これらのパラメータはすべてのリクエスト/レスポンスに存在します:

名前	タイプ	コメント
id	uint32	リクエストの場合 - レスポンスとリクエストを関連付けるために使用されるインクリメント識別子。リクエストの方向（つまり、エージェントからプラグイン、またはプラグインからエージェント）内で一意です。 レスポンスの場合 - 対応するリクエストの ID。
type	uint32	リクエストタイプ。

ログリクエスト

プラグインがエージェントのログファイルにログメッセージを書き込むために送信するリクエスト。

direction	plugin → agent
response	なし

ログリクエストに固有のパラメータ:

名前	型	コメント
severity	uint32	メッセージの重大度（ログレベル）。
message	string	ログに記録するメッセージ。

例:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

登録リクエスト

プラグインを登録するために、エージェントの起動フェーズ中にエージェントから送信されるリクエストです。

direction	agent → plugin
response	yes

登録リクエストに固有のパラメータ:

名前	型	コメント
version	string	プロトコルバージョン <major>.<minor>

例:

```
{"id":1,"type":2,"version":"1.0"}
```

レスポンスの登録

register リクエストに対するプラグインのレスポンス。

direction	plugin → agent
response	n/a

register レスポンスに固有のパラメータ:

名前	型	コメント
name	string	プラグイン名。
metrics	文字列の配列 (オプション)	プラグインで使用される説明付きのメトリクス。RegisterMetrics() を返します。エラーが返された場合は存在しません。
interfaces	uint32 (オプション)	プラグインがサポートするインターフェースのビットマスク。エラーが返された場合は存在しません。
error	string (オプション)	プラグインを開始できない場合に返されるエラーメッセージ。メトリクスが返される場合は存在しません。

例:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

または

```
{"id":2,"type":3,"error":"error message"}
```

Start リクエスト

Runner インターフェースの Start 関数を実行するリクエスト。

direction	agent → plugin
response	なし

リクエストには特定のパラメータはなく、共通データパラメータのみが含まれます。

例:

```
{"id":3,"type":4}
```

リクエストの終了

プラグインをシャットダウンするためにエージェントから送信されるリクエスト。

direction	agent → plugin
response	no

このリクエストには特定のパラメータはなく、共通データパラメータのみが含まれます。

例:

```
{"id":3,"type":5}
```

エクスポートリクエスト

Exporter インターフェースの Export 関数を実行するためのリクエストです。

direction	agent → plugin
response	no

エクスポートリクエストに固有のパラメータ:

名前	型	コメント
key	string	プラグインキー
parameters	文字列の配列 (オプション)	Export 関数のパラメータ

例:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

エクスポートレスポンス

Exporter インターフェースの Export 関数からのレスポンス。

direction	plugin → agent
response	n/a

エクスポートレスポンスに固有のパラメータ:

名前	型	コメント
value	string (省略可能)	Export 関数からのレスポンス値。エラーが返された場合は存在しません。
error	string (省略可能)	Export 関数が正常に実行されなかった場合のエラーメッセージ。値が返された場合は存在しません。

例:

```
{"id":5,"type":7,"value":"response"}
```

または

```
{"id":5,"type":7,"error":"error message"}
```

設定リクエスト

Configurator インターフェースの *Configure* 関数を実行するリクエスト。

direction	agent → plugin
response	n/a

Configure リクエストに固有のパラメータ:

名前	型	コメント
global_options	JSON オブジェクト	グローバルなエージェント設定オプションを含む JSON オブジェクト
private_options	JSON オブジェクト (省略可)	指定された場合、プライベートなプラグイン設定オプションを含む JSON オブジェクト

例:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

リクエストの検証

Configurator インターフェースの *Validate* 関数を実行するリクエスト。

direction	agent → plugin
response	yes

Validate リクエストに固有のパラメータ:

名前	型	コメント
private_options	JSON オブジェクト（オプション）	プライベートなプラグイン設定オプションを含む JSON オブジェクト。指定された場合のみ。

例:

```
{"id":7,"type":9,"private_options":{...}}
```

Validate response

Configurator インターフェースの *Validate* 関数からのレスポンス。

direction	plugin → agent
response	n/a

Validate レスポンスに固有のパラメータ:

名前	型	コメント
error	string (オプション)	Validate 関数が正常に実行されなかった場合に返されるエラーメッセージ。正常に実行された場合は省略されます。

例:

```
{"id":8,"type":10}
```

または

```
{"id":8,"type":10,"error":"error message"}
```

5 Zabbix sender プロトコル

概要

Zabbix サーバーと Zabbix プロキシは、Zabbix sender からデータを受信するのに JSON ベースの通信プロトコルを使用します。トラッパーアイテム、またはトラッピングを有効にしたHTTP エージェントアイテムの助けを借りてデータを受信できます。

リクエストメッセージとレスポンスメッセージは、ヘッダーとデータの長さで始まる必要があります。

Zabbix sender リクエスト

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Zabbix server レスポンス

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

タイムスタンプ付きの Zabbix sender リクエスト

また、Zabbix 送信者はタイムスタンプとナノ秒を含むリクエストを送信できます。

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
```

```
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Zabbix server レスポンス

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

6 ヘッダー

概要

ヘッダーは、Zabbix コンポーネント間のすべての要求および応答メッセージに存在します。ヘッダーの情報からメッセージの長さ、圧縮されているかどうか、大きなパケットであるかどうかを判断できます。

Zabbix 通信プロトコルには、接続ごとに 1GB のパケットサイズ制限があります。受信パケットのデータ長、圧縮前のデータ長ともに 1GB の制限が適用されます。

構成を Zabbix プロキシに送信する場合、大きな構成を同期できるように、パケットサイズの制限が 4 GB に増加します。圧縮前のデータ長が 4GB を超えると、Zabbix サーバーは自動的に大きなパケット形式 (0x04 フラグ) を使用して起動し、パケットサイズの制限を 16GB に増やします。

任意のデータの送信に大きなパケット形式を使用できますが、現在、1GB を超えるパケットを処理できるのは Zabbix プロキシ構成シンカーのみであることに注意してください。

構造

ヘッダーは 4 つのフィールドで構成されます。ヘッダー内のすべての数値は、リトルエンディアンとしてフォーマットされています。

フィールド	サイズ	サイズ (ラージパケット)	説明
<PROTOCOL>	4	4	"ZBXD" または 5A 42 58 44
<FLAGS>	1	1	Protocol フラグ: 0x01 - Zabbix 通信プロトコル 0x02 - 圧縮 0x04 - ラージパケット
<DATALEN>	4	8	データ長
<RESERVED>	4	8	圧縮を使用する場合 (0x02 フラグ) - 非圧縮データの長さ 圧縮を使用しない場合 - 00 00 00 00

例

以下は、Zabbix に送信する必要があるパケットを取得して正しく解釈されるようにするために、送信するデータに Zabbix プロトコルヘッダーを追加する方法を示すコードスニペットです。これらのコードスニペットは、データが 1GB を超えないことを前提としているため、ラージパケットフォーマットは使用されません。

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

または

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
```

```
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

または

```
sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }

    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;
```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

または

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
```

```
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

7 改行区切り JSON エクスポートプロトコル

このセクションでは、以下で使用される改行区切り JSON 形式のエクスポートプロトコルの詳細を示します。

- ファイルへのデータエクスポート
- 外部システムへのストリーミング

以下のデータをエクスポートできます。

- トリガーイベント
- アイテム値
- トレンド (ファイルへのエクスポートのみ)

すべてのファイルは.ndjson 拡張子を持ちます。エクスポートファイルの各行は JSON オブジェクトです。

トリガーイベント

障害イベントについては、以下の情報がエクスポートされます。

フィールド			型	説明
clock			number	障害が検出された時点の Epoch からの秒数（整数部）。
ns			number	障害検出時刻を正確に取得するために `clock` に加算するナノ秒数。
value			number	1（常に）。
eventid			number	障害イベント ID。
name			string	障害イベント名。
severity			number	障害イベントの深刻度（0 - 未分類、1 - 情報、2 - 警告、3 - 軽度、4 - 高度、5 - 障害）。
hosts			array	トリガー式に関与するホストのリスト。配列には少なくとも 1 つの要素が必要です。
	-		object	
		host	string	ホスト名。
		name	string	表示用ホスト名。
groups			array	トリガー式に関与するすべてのホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要
	-		string	ホストグループ名。
tags			array	障害タグのリスト（空の場合あり）。
	-		object	
		tag	string	タグ名。
		value	string	タグ値（空の場合あり）。

復旧イベントについては、以下の情報がエクスポートされます。

フィールド	型	説明
clock	number	障害が復旧した時点の Epoch からの秒数（整数部）。
ns	number	障害復旧時刻を正確に取得するために `clock` に加算するナノ秒数。
value	number	0（常に）。
eventid	number	復旧イベント ID。
p_eventid	number	障害イベント ID。

例

障害:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

リカバリ:

{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}

障害 (多重障害イベントの生成):

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

リカバリ:

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}

アイテム値

収集されたアイテム値については、以下の情報がエクスポートされます。

フィールド	型	説明
host	object	アイテムホストのホスト名
host	string	ホスト名
name	string	表示用ホスト名
groups	array	アイテムホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要です。
-	string	ホストグループ名
item_tags	array	アイテムタグのリスト（空の場合あり）
-	object	
tag	string	タグ名
value	string	タグ値（空の場合あり）
itemid	number	アイテム ID
name	string	表示用アイテム名
clock	number	値が収集された時点の Epoch からの秒数（整数部）
ns	number	値の収集時刻を正確に得るために `clock` に加えるナノ秒数
timestamp (*Log* のみ)	number	利用できない場合は 0
source (*Log* のみ)	string	利用できない場合は空文字列
severity (*Log* のみ)	number	利用できない場合は 0
eventid (*Log* のみ)	number	利用できない場合は 0
value	数値アイテムの場合は number、テキストアイテムの場合は string	収集されたアイテム値
type	number	収集された値の型： 0 - 数値（浮動小数点）、1 - 文字列、2 - ログ、3 - 数値（符号なし）、4 - テキスト、5 - バイナリ

例

数値（符号なし）値:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

数値（浮動小数点）値:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

文字、テキスト値:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

ログ値:

{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

トレンド

計算されたトレンド値については、次の情報がエクスポートされます。

フィールド			型	説明
host			object	アイテムのホスト名
	host		string	ホスト名
	name		string	表示用ホスト名
groups			array	アイテムのホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要です。
	-		string	ホストグループ名
item_tags			array	アイテムタグのリスト（空の場合あり）
	-		object	
		tag	string	タグ名
		value	string	タグ値（空の場合あり）
itemid			number	アイテム ID
name			string	表示用アイテム名
clock			number	値が収集された時点の Epoch からの秒数（整数部）
count			number	指定した 1 時間に収集された値の数
min			number	指定した 1 時間の最小アイテム値
avg			number	指定した 1 時間の平均アイテム値
max			number	指定した 1 時間の最大アイテム値
type			number	値の型: 0 - 数値（浮動小数点）, 3 - 数値（符号なし）

例

数値（符号なし）値：

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

数値（浮動小数点）値：

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

4 アイテム

サイドバーから、参照したい項目を選択してください。

1 vm.memory.size パラメーター

概要

このセクションでは、vm.memory.size[<mode>] エージェントアイテムのパラメータの詳細について説明します。

パラメータ

このアイテムには以下のパラメータがあります:

- **active** - 現在使用中またはごく最近使用されたメモリで、RAM に存在する
- **anon** - ファイルに関連付けられていないメモリ（ファイルから再読み込みできない）
- **available** - 利用可能なメモリ。プラットフォームによって計算方法が異なる（下表参照）
- **buffers** - ファイルシステムメタデータなどのキャッシュ
- **cached** - 様々なもののキャッシュ

- **exec** - 実行可能なコード。通常は（プログラム）ファイルから
- **file** - 最近アクセスされたファイルの内容のキャッシュ
- **free** - メモリを要求する任意のエンティティがすぐに利用できるメモリ
- **inactive** - 未使用としてマークされたメモリ
- **pavailable** - 'total' に対する'available' メモリの割合（`available/total*100` で計算）
- **pinned** - 'wired' と同じ
- **pused** - 'total' に対する'used' メモリの割合（`used/total*100` で計算）
- **shared** - 複数のプロセスが同時にアクセスできるメモリ
- **slab** - カーネルが自身の使用のためにデータ構造をキャッシュするために使用するメモリの合計量
- **total** - 利用可能な物理メモリの合計
- **used** - 使用中のメモリ。プラットフォームによって計算方法が異なる（下表参照）
- **wired** - 常に RAM に保持されるようにマークされたメモリ。ディスクに移動されることはない。

Warning:
これらのパラメータの一部はプラットフォーム固有であり、お使いのプラットフォームでは利用できない場合があります。詳細はZabbix エージェントアイテムを参照してください。

available および **used** のプラットフォーム別計算方法:

プラットフォーム	**"available"**	**"used"**
AIX	free + cached	使用中の実メモリ
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (RHEL 7 の 3.10 にもバックポート済み)	/proc/meminfo、詳細は Linux カーネルドキュメントの"MemAvailable" を参照。 free + buffers + cached は、すべてのページキャッシュが解放できるわけではなく、計算に low watermark が使われるため、もはや'available' と等しくないことに注意。	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
vm.memory.size[used] と *vm.memory.size[available]* の合計が必ずしも total と等しくなるとは限りません。例えば FreeBSD では、
* Active、inactive、wired、cached メモリは有用な情報を保持しているため、使用中と見なされます。
* 同時に inactive、cached、free メモリは、より多くのメモリを要求するプロセスに即座に割り当てることができるため、利用可能と見なされます。

したがって、inactive メモリは同時に使用中かつ利用可能となります。このため、*vm.memory.size[used]* アイテムは情報提供のみを目的として設計されており、*vm.memory.size[available]* はトリガーで使用することを目的としています。

以下も参照

1. 異なる OS でのメモリ計算に関する追加の詳細

2 パッシブおよびアクティブエージェントチェック

概要

このセクションでは、Zabbix エージェントおよびZabbix エージェント 2によって実行されるパッシブチェックとアクティブチェックの詳細について説明します。

Zabbix は、エージェントとの通信に JSON ベースの通信プロトコルを使用しています。

Zabbix エージェントと Zabbix エージェント 2 のプロトコルは、Zabbix 7.0 以降で統一されました。Zabbix エージェントと Zabbix エージェント 2 のリクエスト/レスポンスの違いは、"variant" タグの値で表されます。

パッシブチェック

パッシブチェックは単純なデータリクエストです。Zabbix サーバーまたはプロキシが何らかのデータ（例：CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。

パッシブチェックは非同期で実行されます。1 つのリクエストへの応答を受信する前に他のチェックを開始する必要はありません。DNS 解決も非同期です。

エージェントポーラーは、DNS ルックアップで返されたすべてのアドレスへの接続を試みます。これにより、1 つの IP アドレスに到達できない場合でも、ポーラーは次に利用可能なアドレスを試み、接続成功の可能性が高まります。この機能強化は、Zabbix サーバーとプロキシの両方に適用されます。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPollerで定義）。

非同期エージェントポーラーの数は、StartAgentPollersパラメータで定義されます。

サーバーリクエスト

ヘッダーとデータ長の定義については、プロトコルの詳細を参照してください。

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

フィールド		型	必須	値
request		*string*	yes	"passive checks"
data		*array of object*	yes	パッシブチェックアイテム
	key	*string*	yes	展開済みマクロを含むアイテムキー
	timeout	*string*	yes	アイテムのタイムアウト

エージェント応答

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

フィールド		型	必須	値
version		*string*	yes	エージェントのバージョン番号
variant		*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）
data		*array of object*	yes	チェック結果を含む
	value	*string*	no	チェックが成功した場合のアイテム値
	error	*string*	no	チェックが失敗した場合のエラーメッセージ

サポートされているアイテムの場合の例：

1. サーバーが TCP 接続を開く
2. サーバーが **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}** を送信
3. エージェントがリクエストを読み取り、**<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”value”:1}]}** で応答
4. サーバーがデータを処理し、値（この場合は’1’）を取得
5. TCP 接続を閉じる

サポートされていないアイテムの場合：

1. サーバーが TCP 接続を開く

2. サーバーが **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"vfs.fs.size[/nono]","timeout":3}]}** を送信
3. エージェントがリクエストを読み取り、**<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"error":"Unsupported item key."}]}** で応答
4. サーバーがデータを処理し、指定されたエラーメッセージでアイテムの状態をサポートされていないに変更
5. TCP 接続を閉じる

古いプロトコルへのフェイルオーバー

Zabbix サーバーまたはプロキシが、プレーンテキストプロトコルを持つ 7.2 以前のバージョンのエージェントと連携できるように、古いプロトコルへのフェイルオーバーが実装されています。

パッシブチェックは、再起動後またはインターフェース構成が変更された場合に、JSON プロトコル（7.0 以降）を使用して実行されます。応答で有効な JSON が受信されなかった場合（エージェントが"ZBX_NOTSUPPORTED" を送信した場合）、Zabbix はインターフェースを古いプロトコルとしてキャッシュし、再試行としてアイテムキーのみを送信してチェックを行います。

Zabbix サーバー/プロキシは、すべてのインターフェースで毎時新しいプロトコルでの動作を再試行し、必要に応じて古いプロトコルにフォールバックすることに注意してください。

アクティブチェック

アクティブチェックは、より複雑な処理を必要とします。エージェントは、独立した処理のために、最初にサーバー/プロキシからアイテムやリモートコマンドのリストを取得する必要があります。

アクティブチェックを取得するサーバー/プロキシは、エージェントの設定ファイルの「ServerActive」パラメータに記載されています。これらのチェックを要求する頻度は、同じ設定ファイルの「RefreshActiveChecks」パラメータで設定します。ただし、アクティブチェックの更新に失敗した場合は、ハードコードされた 60 秒後に再試行されます。

Note:
Zabbix 6.4 以降、エージェント（アクティブモード）は、サーバー/プロキシから 2 分ごと（デフォルト）に設定の完全なコピーを受信しなくなりました。代わりに、ネットワークトラフィックとリソース使用量を減らすために、5 秒ごと（デフォルト）に増分設定同期が実行され、サーバー/プロキシは、エージェントがまだ受信していない場合、またはホスト設定、グローバルマクロ、グローバル正規表現に何か変更があった場合にのみ設定の完全なコピーを提供します。

エージェントは定期的に新しい値をサーバーに送信します。エージェントが実行するリモートコマンドを受信した場合、実行結果も送信されます。アクティブエージェントでのリモートコマンド実行のサポートは、Zabbix agent 7.0 からです。

Note:
エージェントがファイアウォールの背後にある場合は、アクティブチェックのみを使用することを検討してください。この場合、初期の着信接続を許可するためにファイアウォールを変更する必要がありません。

アイテムのリストの取得

エージェントリクエスト

アクティブチェックリクエストは、エージェントが処理するアクティブチェックを取得するために使用されます。このリクエストは、エージェントの起動時およびRefreshActiveChecksの間隔でエージェントから送信されます。

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

フィールド	型	必須	値
request	*string*	yes	`active checks`
host	*string*	yes	ホスト名
host_metadata	*string*	no	設定パラメータ HostMetadata または HostMetadataItem のメトリック値
interface	*string*	no	設定パラメータ HostInterface または HostInterfaceItem のメトリック値
ip	*string*	no	設定パラメータ ListenIP が設定されている場合の最初の IP

フィールド	型	必須	値
port	*number*	no	設定パラメータ ListenPort が設定されており、デフォルトのエージェントリスニングポートでない場合の値
version	*string*	yes	エージェントのバージョン番号
variant	*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）
config_revision	*number*	no	増分設定同期用の設定識別子
session	*string*	no	増分設定同期用のセッション識別子

サーバーレスポンス

アクティブチェックレスポンスは、アクティブチェックリクエストの処理後にサーバーからエージェントに返されます。

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

フィールド	型	必須	値
response	*string*	yes	`success` | `failed`
info	*string*	no	失敗時のエラー情報
data	*array of objects*	no	アクティブチェックアイテム。ホスト設定が変更されていない場合は省略されます。
key	*string*	no	展開されたマクロを含むアイテムキー
itemid	*number*	no	アイテム識別子
delay	*string*	no	アイテムの更新間隔 柔軟/スケジューリング間隔は、Zabbix 7.0 以降、Zabbix agent および Zabbix agent 2 の両方でサポートされています。
lastlogsize	*number*	no	アイテムの lastlogsize
mtime	*number*	no	アイテムの mtime
timeout	*string*	no	アイテムのタイムアウト
refresh_unsupported	*number*	no	サポートされていないアイテムのリフレッシュ間隔
regexp	*array of objects*	no	グローバル正規表現

フィールド		型	必須	値
	name	*string*	no	グローバル正規表現名
	expression	*string*	no	グローバル正規表現
	expression_type	*number*	no	グローバル正規表現タイプ
	exp_delimiter	*string*	no	グローバル正規表現のデリミタ
	case_sensitive	*number*	no	グローバル正規表現の大文字小文字の区別設定
commands		*array of objects*	no	実行するリモートコマンド。リモートコマンドの実行がアクションオペレーションまたは手動スクリプトの実行によってトリガーされた場合に含まれます。アクティブエージェントでのリモートコマンド実行は Zabbix agent 7.0 以降でサポートされています。古いアクティブエージェントは、アクティブチェックサーバーレスポンスに含まれるリモートコマンドを無視します。
	command	*string*	no	リモートコマンド
	id	*number*	no	リモートコマンド識別子
	wait	*number*	no	リモートコマンドの実行モード（アクションオペレーションからのコマンドの場合は”0”（nowait）、手動スクリプトの実行からのコマンドの場合は”1”（wait））
config_revision		*number*	no	増分設定同期用の設定識別子。ホスト設定が変更されていない場合は省略されます。ホスト設定が変更された場合はインクリメントされます。

サーバーは必ず success で応答する必要があります。

例:

1. エージェントが TCP 接続を開く
2. エージェントがチェックのリストを要求する
3. サーバーがアイテムと実行するリモートコマンドのリストで応答する
4. エージェントがレスポンスを解析する
5. TCP 接続が閉じられる
6. エージェントが定期的なデータ収集とリモートコマンドの実行を開始する（Zabbix agent 7.0 以降でサポート）

Attention:
アクティブチェックを使用する場合、（機密性の高い）設定データが Zabbix サーバートラッパーポートにアクセスできる第三者に利用可能になる可能性があることに注意してください。これは、誰でもアクティブエージェントを装ってアイテム設定データを要求できるためです。暗号化オプションを使用しない限り、認証は行われません。

収集したデータの送信

エージェントが送信

エージェントデータリクエストには、収集されたアイテムの値と実行されたリモートコマンドの値（存在する場合）が含まれます。

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
  "commands": [
```

```
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "1234456akdsjhfoui",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

フィールド	型	必須	値
request	*string*	yes	`agent data`
data	*array of objects*	yes	アイテムの値。
id	*number*	yes	値の識別子（ネットワークの問題で重複した値をチェックするために使用されるインクリメンタルカウンター）。
itemid	*string*	yes	アイテムの識別子。
value	*string*	no	アイテムの値。
lastlogsize	*number*	no	アイテムの lastlogsize。
mtime	*number*	no	アイテムの mtime。
state	*number*	no	アイテムの状態。
source	*string*	no	値のイベントログソース。
eventid	*number*	no	値のイベントログ eventid。
severity	*number*	no	値のイベントログ severity。
timestamp	*number*	no	値のイベントログ timestamp。
clock	*number*	yes	値のタイムスタンプ（Epoch からの秒数）。
ns	*number*	yes	値のタイムスタンプのナノ秒。
commands	*array of objects*	no	リモートコマンドの実行結果。アクティブエージェントでのリモートコマンド実行は Zabbix agent 7.0 以降でサポートされています。古いアクティブエージェントは、アクティブチェックサーバーの応答に含まれるリモートコマンドを無視します。
id	*number*	no	リモートコマンドの識別子。
value	*string*	no	実行が成功した場合のリモートコマンドの実行結果。
error	*string*	no	実行が失敗した場合のリモートコマンドの実行エラーメッセージ。
session	*string*	yes	エージェントが起動されるたびに生成される一意のセッション識別子。
host	*string*	yes	ホスト名。
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）。

各値には仮想 ID が割り当てられます。値 ID は単純な昇順カウンターで、1 つのデータセッション（セッショントークンで識別）内で一意です。この ID は、接続性が悪い環境で送信される可能性のある重複値を破棄するために使用されます。

サーバーの応答

エージェントデータの応答は、サーバーがエージェントデータリクエストを処理した後、エージェントに返されます。

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

フィールド	型	必須	値
response	*string*	yes	`success` | `failed`
info	*string*	yes	アイテム処理結果。

Attention:
サーバー側で一部の値の送信に失敗した場合（たとえば、ホストやアイテムが無効化または削除された場合）、エージェントはそれらの値の再送信を行いません。

例:

1. エージェントが TCP 接続を開く
2. エージェントが値のリストを送信する
3. サーバーがデータを処理し、ステータスを返す
4. TCP 接続が閉じられる

上記の例では、vfs.fs.size[/nono] の not supported ステータスが、”state” 値が 1 であり、”value” プロパティにエラーメッセージが含まれていることで示されていることに注意してください。

Attention:
エラーメッセージはサーバー側で 2048 文字に切り詰められます。

ハートビートメッセージ

エージェントが送信

ハートビートメッセージは、アクティブエージェントから Zabbix サーバー/プロキシに対して、Zabbix agent/agent 2の設定ファイルで設定された HeartbeatFrequency 秒ごとに送信されます。

これはアクティブチェックの可用性を監視するために使用されます。

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

フィールド	型	必須	値
request	*string*	yes	`active check heartbeat`
host	*string*	yes	ホスト名
heartbeat_freq	*number*	yes	エージェントのハートビート間隔（HeartbeatFrequency 設定パラメータ）
version	*string*	yes	エージェントのバージョン番号
variant	*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）

リダイレクト応答

ホストが再割り当てされた場合、サーバーはエージェントに対してハートビート（およびその後のアクティブチェック）を別のプロキシまたはサーバーインスタンスにリダイレクトするよう指示することがあります。

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.0:10055"
    }
  }
```

フィールド		型	必須	値
response		*string*	yes	`success` | `failed`
redirect		*object*	yes	リダイレクト指示
	revision	*number*	yes	設定リビジョン識別子
	address	*string*	yes	転送先サーバー/プロキシのアドレス

古い XML プロトコル

Note:
Zabbix は、Base64 にエンコードされた XML のデータを 16MB まで取得しますが、デコードされる値は 1 つが 64KB 以下でなければなりません。そうでないと、デコード中にそれが 64KB に切り捨てられます。

3 Windows エージェントアイテムの最小アクセス許可レベル

概要

エージェントを使用してシステムを監視する場合は、エージェントがインストールされているホストからメトリックを取得することをお勧めします。最小特権の原則を使用するには、エージェントから取得するメトリックを判別する必要があります。

このドキュメントの表では、Zabbix エージェントの正しい動作を保証するための最小権限を選択できます。

エージェントの実行ユーザーを「LocalSystem」ではなく別のユーザーにした場合、エージェントを Windows サービスとして実行するために対象ユーザーは「ローカルポリシー」→「ユーザー権限の割り当て」から「サービスとしてログオン」の権限と、Zabbix エージェントのログファイルを作成、書き込み、削除する権限を持っている必要があります。対象ユーザーを ActiveDirectory の *PerformanceMonitorUsers* グループに追加する必要があります。

Note:
「技術的に許容できる最小限の」グループに基づくエージェントの権利を使用する場合は、監視対象のオブジェクトに対する権利を事前に提供する必要があります。

Windows でサポートされている一般的なエージェントアイテム

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests

アイテムキー	ユーザーグループ	
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows 固有のアイテムキー

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

4 戻り値のエンコード

Zabbix サーバーは、UTF8 エンコーディングで返されるすべてのテキスト値を想定しています。これは Zabbix エージェント、SSH、Telnet などのあらゆるタイプのチェックに関連しています。

監視対象のさまざまなシステム/デバイスおよびチェックで、値に非 ASCII 文字が返される場合があります。このような場合、ほとんどすべての **zabbix** キーには、追加のアイテムキーパラメーター - **<encoding>** が含まれています。このキーパラメーターはオプションですが、戻り値が UTF8 エンコーディングではなく、非 ASCII 文字が含まれている場合は指定する必要があります。指定していないと結果が予期せぬ値になる可能性があります。

このような場合のさまざまなデータベースバックエンドでの動作の説明は次のとおりです。

MySQL

少なくとも MySQL バージョン 5.1.61 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、この文字と次の文字は、データベースが値を格納するときに破棄されます。警告メッセージは *zabbix_server.log* に書き込まれません。

PostgreSQL

少なくとも PostgreSQL バージョン 9.1.3 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、SQL クエリが失敗し (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) データは保存されません。適切な警告メッセージが *zabbix_server.log* に書き込まれます。

5 ラージファイルサポート

ラージファイルサポートとは、多くの場合 LFS と略され、32bit オペレーティングシステムで 2GB を超えるファイルを処理する機能に適用される用語です。ラージファイルサポートは、少なくともログファイルの監視とすべてのvfs.file.* アイテムに影響します。ラージファイルサポートは、Zabbix コンパイル時のシステムの機能に依存しますが、procfs および swapctl との互換性がないため、32bit Solaris では完全に無効になっています。

各センサーチップは、sysfs /sys/devices ツリー内に独自のディレクトリを持ちます。すべてのセンサーチップを見つけるには、/sys/class/hwmon/hwmon* からデバイスのシンボリックリンクをたどる方が簡単です（* は実際の番号（0,1,2,...））。

センサーの読み取り値は、仮想デバイスの場合は /sys/class/hwmon/hwmon*/ ディレクトリ、非仮想デバイスの場合は /sys/class/hwmon/hwmon*/device ディレクトリにあります。hwmon* または hwmon*/device ディレクトリ内にある name というファイルには、チップの名前が含まれており、これはセンサーチップで使用されるカーネルドライバーの名前に対応しています。

ファイルごとに 1 つのセンサー読み取り値のみが含まれます。上記のいずれかのディレクトリ内でセンサー読み取り値を含むファイルの一般的な命名規則は次のとおりです：<type><number>_<item>、ここで

- **type** - センサーチップの場合は”in”（電圧）、”temp”（温度）、”fan”（ファン）など
- **item** - ”input”（測定値）、”max”（上限値）、”min”（下限値）など
- **number** - 複数存在する可能性のある要素には常に使用されます（通常は 1 から始まりますが、電圧の場合は 0 から始まります）。ファイルが特定の要素を参照していない場合は、番号のない単純な名前になります。

ホストで利用可能なセンサーに関する情報は、**sensor-detect** および **sensors** ツール（lm-sensors パッケージ：http://lm-sensors.org/）を使用して取得できます。**Sensors-detect** は、利用可能なセンサーに必要なモジュールを特定するのに役立ちます。モジュールがロードされると、**sensors** プログラムを使用してすべてのセンサーチップの読み取り値を表示できます。このプログラムで使用されるセンサー読み取り値のラベル付けは、一般的な命名規則（<type><number>_<item>）とは異なる場合があります：

- <type><number>_label というファイルがある場合は、このファイル内のラベルが <type><number><item> 名の代わりに使用されます
- <type><number>_label ファイルがない場合は、/etc/sensors.conf（/etc/sensors3.conf など、異なる場合もあり）内で名前の置換を検索します

このラベル付けにより、ユーザーはどのようなハードウェアが使用されているかを特定できます。<type><number>_label ファイルも設定ファイル内のラベルもない場合は、name 属性（hwmon*/device/name）によってハードウェアの種類を特定できます。zabbix_agent が受け付けるセンサーの実際の名前は、**sensors** プログラムを -u パラメータ（**sensors -u**）で実行することで取得できます。

sensor プログラムでは、利用可能なセンサーはバスタイプ（ISA アダプタ、PCI アダプタ、SPI アダプタ、仮想デバイス、ACPI インターフェース、HID アダプタ）ごとに区切られています。

Linux 2.4 の場合:

(センサーの読み取り値は /proc/sys/dev/sensors ディレクトリから取得されます)

- **device** - デバイス名（<mode> が使用されている場合は正規表現）；
- **sensor** - センサー名（<mode> が使用されている場合は正規表現）；
- **mode** - 可能な値: avg, max, min（このパラメータが省略された場合、device と sensor はそのまま扱われます）。

キーの例: sensor[w83781d-i2c-0-2d,temp1]

Linux 2.6 以降の場合：

(センサーの読み取り値は/sys/class/hwmon ディレクトリから取得されます)

- **device** - デバイス名（非正規表現）デバイス名はデバイスの実際の名前（0000:00:18.3 など）またはセンサープログラムを使用して取得した名前（k8temp-pci-00c3 など）の場合があります。使用する名前はユーザーが選択してください。
- **sensor** - センサー名（非正規表現）
- **mode** - 使用可能な値: avg, max, min (このパラメーターを省略した場合、デバイスとセンサーは逐語的に扱われます).

キーの例：

sensor[k8temp-pci-00c3,temp,max] or sensor[0000:00:18.3,temp1]

sensor[smsc47b397-isa-0880,in,avg] or sensor[smsc47b397.2176,in1]

センサー名の取得

sensors コマンドで表示されるセンサーラベルは、各センサーチップベンダーごとにラベルの命名が異なる場合があるため、必ずしも直接使用できるとは限りません。例えば、*sensors* の出力には以下のような行が含まれている場合があります。

```
$ sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

このうち、直接使用できるラベルは 1 つだけです。

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

他のラベル（*Vcore* や *+12V* など）を使用しようとしても動作しません。

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Zabbix がセンサー値を取得するために使用できる実際のセンサー名を調べるには、*sensors -u* を実行します。出力には以下のような内容が表示されます。

```
$ sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

したがって、*Vcore* は *in1*、*+12V* は *in4* として問い合わせる必要があります。仕様によると、これらはチップピンの電圧であり、一般的にはスケーリングが必要な場合があります。

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix では、電圧（in）だけでなく、電流（curr）、温度（temp）、ファン速度（fan）の値も取得できます。

7 proc.mem アイテムの **memtype** パラメーターに関する注意

概要

memtype パラメーターは、Linux,AIX,FreeBSD, および Solaris でサポートされています。

'memtype' の 3 つの一般的な値である `pmem`、`rss`、`vsize` は、これらすべてのプラットフォームでサポートされています。さらに一部のプラットフォームでは固有の'memtype' 値がサポートされています。

AIX

以下の表が AIX の'memtype' パラメーターでサポートされている値です。

サポート値	説明	procentry64 構造体のソース	互換
vsize [1]	仮想メモリサイズ	pi_size	
pmem	実メモリのパーセンテージ	pi_prm	ps -o pmem
rss	物理メモリ消費量	pi_trss + pi_drss	ps -o rssize
size	プロセスのサイズ (コード + データ)	pi_dvm	"ps gvw" SIZE column
dsize	データサイズ	pi_dsize	
tsize	テキスト (コード) サイズ	pi_tsize	"ps gvw" TSIZ column
sdsize	シェアードライブラリのデータサイズ	pi_sdsize	
drss	データの物理メモリ消費量	pi_drss	
trss	テキストの物理メモリ消費量	pi_trss	

AIX に関する注意事項:

1. AIX で proc.mem[] アイテムキーのパラメータを選択する際は、できるだけ絞り込んだプロセス選択条件を指定してください。そうしないと、proc.mem[] の結果に不要なプロセスがカウントされるリスクがあります。

例:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

この例は、コマンドライン (正規表現で一致させる) パラメータだけを指定した場合、Zabbix エージェント自身がカウントされてしまうことを示しています。おそらく、これは望んでいる結果ではありません。

2. ”ps -ef” を使用してプロセスを閲覧しないでください。これはカーネル以外のプロセスのみを表示します。Zabbix エージェントが見ることができるすべてのプロセスを表示するには、”ps -Af” を使用してください。

3. Zabbix エージェントの proc.mem[] がどのようにプロセスを選択するか、’topasrec’ の例で説明します。

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

proc.mem[] には以下の引数があります:

```
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]
```

1 つ目の基準はプロセス名（引数 <name>）です。この例では、Zabbix エージェントはこれを「topasrec」として認識します。一致させるには、「topasrec」を指定するか、空のままにする必要があります。2 つ目の基準はユーザー名（引数 <user>）です。一致させるには、「root」を指定するか、空のままにする必要があります。プロセス選択に使用される 3 つ目の基準は引数 <cmdline> です。Zabbix エージェントはその値を「/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023」として認識します。一致させるには、この文字列に一致する正規表現を指定するか、空のままにする必要があります。

引数 <mode> と <memtype> は、上記の 3 つの基準を使用した後に適用されます。

FreeBSD

以下の表が FreeBSD の’memtype’ パラメーターでサポートされている値です。

サポート値	説明	kinfo_proc 構造体のソース	互換
vsize	仮想メモリサイズ	kp_eproc.e_vm.vm_map.size or ki_size	ps -o vsz
pmem	実メモリのパーセンテージ	calculated from rss	ps -o pmem
rss	物理メモリ消費量	kp_eproc.e_vm.vm_rssize or ki_rssize	ps -o rss
size [1]	プロセスのサイズ (コード + データ + スタック)	tsize + dsize + ssize	
tsize	テキスト (コード) サイズ	kp_eproc.e_vm.vm_tsize or ki_tsize	ps -o tsiz
dsize	データサイズ	kp_eproc.e_vm.vm_dsize or ki_dsize	ps -o dsiz
ssize	スタックサイズ	kp_eproc.e_vm.vm_ssize or ki_ssize	ps -o ssiz

Linux

以下の表が Linux の’memtype’ パラメーターでサポートされている値です。

サポート値	説明	/proc/<pid>/status ファイルのソース
vsize [1]	仮想メモリサイズ	VmSize
pmem	実メモリのパーセンテージ	(VmRSS/total_memory) * 100
rss	物理メモリ消費量	VmRSS
data	データセグメントサイズ	VmData
exe	コードセグメントサイズ	VmExe
hwm	物理メモリ消費のピーク	VmHWM
lck	ロックメモリのサイズ	VmLck
lib	シェアードライブラリのサイズ	VmLib
peak	仮想メモリサイズのピーク	VmPeak
pin	固定されたページのサイズ	VmPin
pte	ページテーブルエントリのサイズ	VmPTE
size	プロセスコードとデータとスタックセグメントの合計値	VmExe + VmData + VmStk
stk	スタックセグメントのサイズ	VmStk
swap	スワップスペース使用量	VmSwap

Linux に関する注意：

1. 古い Linux カーネルではサポートされていない `memtype` 値があります。たとえば Linux 2.4 カーネルは、`hwm`、`pin`、`peak`、`pte`、および `swap` の値をサポートしていません。
2. `proc.mem[...,...,...,...,data]` を使用した Zabbix エージェントのアクティブチェックプロセス自己監視は、エージェントの/proc/<pid>/status ファイルの `VmData` 行で報告される値よりも 4kB 大きい値が示されていることがわかりました。自己測定時にエージェントのデータセグメントは 4 kB 増加し、その後前のサイズに戻ります。

Solaris

以下の表が Solaris の'memtype' パラメーターでサポートされている値です。

サポート値	説明	psinfo 構造のソース	互換
vsize [1]	プロセスイメージのサイズ	pr_size	ps -o vsz
pmem	実メモリのパーセンテージ	pr_pctmem	ps -o pmem
rss	物理メモリ消費量 過小評価されている可能性があります。"man ps" の rss の説明を参照してください	pr_rssize	ps -o rss

脚注

[1] デフォルト値

8 proc.mem および **proc.num** アイテムでのプロセスの選択に関する注意

コマンドラインを変更するプロセス

一部のプログラムは、現在のアクティビティを表示する方法としてコマンドラインの変更を使用します。ユーザーは ps や top コマンドを実行することでアクティビティを確認できます。このようなプログラムの例としては、*PostgreSQL*、*Sendmail*、*Zabbix* などがあります。

Linux の例を見てみましょう。Zabbix エージェントプロセスの数を監視したいと仮定します。

ps コマンドは、対象のプロセスを次のように表示します。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/ZBXNEXT-1078/zabbix_age
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

名前とユーザーでプロセスを選択すると、目的を達成できます。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

ここで、zabbix_agentd 実行ファイルの名前を zabbix_agentd_30 に変更し、再起動します。

ps の表示は次のようになります。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/ZBXNEXT-1078/zabbix_
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

この状態で名前とユーザーでプロセスを選択すると、誤った結果になります。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

なぜ実行ファイルの名前を長くしただけで、まったく異なる結果になったのでしょうか？

Zabbix エージェントはまずプロセス名をチェックします。/proc/<pid>/status ファイルを開き、Name 行を確認します。この場合、Name 行は次のようになります。

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

status ファイルのプロセス名は 15 文字に切り詰められます。

同様の結果は ps コマンドでも確認できます。

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

明らかに、これは proc.num[] の name パラメータ値 zabbix_agentd_30 とは一致しません。status ファイルのプロセス名と一致しなかったため、Zabbix エージェントは /proc/<pid>/cmdline ファイルを参照します。

エージェントが"cmdline" ファイルをどのように認識するかは、次のコマンドで確認できます。

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/ZBXNEXT-1078/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

この場合、/proc/<pid>/cmdline ファイルには、*C* 言語で文字列を終了するために使用される不可視の非表示のヌルバイトが含まれています。この例ではヌルバイトは"<NUL>" として表示されています。

Zabbix エージェントはメインプロセスの"cmdline" をチェックし、zabbix_agentd_30 を取得します。これは name パラメータ値 zabbix_agentd_30 と一致します。したがって、メインプロセスは proc.num[zabbix_agentd_30,zabbix] アイテムでカウントされます。

次のプロセスをチェックする際、エージェントは cmdline ファイルから zabbix_agentd_30: collector [idle 1 sec] を取得しますが、これは name パラメータ zabbix_agentd_30 とは一致しません。したがって、コマンドラインを変更しないメインプロセスのみがカウントされます。他のエージェントプロセスはコマンドラインを変更しているため無視されます。

この例から、proc.mem[] や proc.num[] の name パラメータは、このケースでプロセスを選択するために使用できないことがわかります。

Note:
proc.get[] アイテムの場合、Zabbix エージェントがプロセス名の"cmdline" をチェックするときは、名前の最後のスラッシュから最初のスペースまたはコロン記号までの部分のみを使用します。cmdline ファイルから取得したプロセス名は、その先頭部分が status ファイルの短縮プロセス名と完全に一致する場合にのみ使用されます。フィルター内のプロセス名と JSON 出力内のプロセス名の両方でアルゴリズムは同じです。

cmdline パラメータに適切な正規表現を使用すると、正しい結果が得られます。

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

コマンドラインを変更するプログラムの監視には、proc.get[]、proc.mem[]、proc.num[] アイテムの使用に注意してください。

proc.get[]、proc.mem[]、proc.num[] アイテムに name や cmdline パラメータを設定する前に、proc.num[] アイテムや ps コマンドを使ってパラメータをテストすることをお勧めします。

Linux カーネルスレッド

proc.get[]、proc.mem[]、proc.num[] アイテムの cmdline パラメータでスレッドを選択できません

カーネルスレッドの 1 つを例にとってみましょう:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

これはプロセスの name パラメータで選択できます:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

しかし、プロセスの cmdline パラメータによる選択は機能しません:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

その理由は、Zabbix エージェントが cmdline パラメータで指定された正規表現をプロセスの/proc/<pid>/cmdline の内容に適用するためです。カーネルスレッドの場合、/proc/<pid>/cmdline ファイルは空です。そのため、cmdline パラメータは一致しません。

proc.mem[] および proc.num[] アイテムのスレッドカウント

Linux カーネルスレッドは proc.num[] アイテムによってカウントされますが、proc.mem[] アイテムのメモリは報告されません。

例：

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

しかし、カーネルスレッドと同じ名前のユーザープロセスがある場合は次のようになります。

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num [] はカーネルスレッドとユーザープロセスの両方をカウントしました。proc.mem [] はユーザープロセスのメモリのみを報告し、カーネルスレッドメモリを 0 であるかのようにカウントします。これは ZBX_NOTSUPPORTED が報告された上記の場合とは異なります。

プログラム名がスレッドの 1 つと一致する場合は、proc.mem[] および proc.num[] アイテムを使用するときに注意してください。

パラメータを proc.mem[] および proc.num[] アイテムに入れる前に、proc.num[] アイテムと ps コマンドを使用してパラメータをテストすることをお勧めします。

9 net.tcp.service および **net.udp.service** チェックの実装の詳細

このページでは net.tcp.service および net.udp.service チェック実装における、service パラメーターで指定されたさまざまなサービスについて詳しく説明しています。

net.tcp.service アイテムのパラメータ

ftp

TCP 接続を作成し、応答の最初の 4 文字が"220 " であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 21 が使用されます。

http

特に何もせず TCP 接続を作成します。ポートが指定されていない場合はデフォルトのポート 80 が使用されます。

https

libcurl を使用し（そしてそれとともにのみ機能し）、証明書の信頼性を検証せず、SSL 証明書のホスト名を検証せず、応答ヘッダー（HEAD 要求）のみを読み出します。ポートが指定されていない場合はデフォルトポートの 443 が使用されます。

imap

TCP 接続を作成し、応答の最初の 4 文字が”* OK” であることを検証して”a1 LOGOUT\r\n” を送信します。ポートが指定されていない場合はデフォルトのポート 143 が使用されます。

ldap

LDAP サーバーへの接続を開き、フィルターを (objectClass=*) に設定して LDAP 検索操作を実行して、最初のエントリの最初の属性が正常に取得できることを検証します。ポートが指定されていない場合はデフォルトのポート 389 が使用されます。

nntp

TCP 接続を作成し、応答の最初の 3 文字が”200” または”201” であることを検証して”QUIT\r\n” を送信します。ポートが指定されていない場合はデフォルトのポート 119 が使用されます。

pop

TCP 接続を作成し、応答の最初の 3 文字が”+OK” であることを検証して”QUIT\r\n” を送信します。ポートが指定されていない場合はデフォルトのポート 110 が使用されます。

smtp

TCP 接続を作成し、応答の最初の 3 文字が”220” であり、その後にスペース、行末、またはダッシュが続くことを検証します。ダッシュを含む行は複数行の応答で、ダッシュのない行が受信されるまで応答が再読み取りされます。次に”QUIT\r\n” を送信します。ポートが指定されていない場合はデフォルトのポート 25 が使用されます。

ssh

TCP 接続を作成します。接続が確立されている場合、両側で識別文字列（SSH-major.minor-XXXX）が交換されます。ここでのメジャーとマイナーはプロトコルバージョンであり、XXXX は文字列です。Zabbix は仕様に一致する文字列が見つかったかどうかを確認し、不一致の場合は文字列”SSH-major.minor-zabbix_agent\r\n” または”0\n” を送り返します。ポートが指定されていない場合は、デフォルトのポート 22 が使用されます。

tcp

特に何もせず TCP 接続を作成します。他のチェックとは異なり、ポート指定必須です。

telnet

TCP 接続を作成し、ログインプロンプト (の最後に’:’) が返ってくることを検証します。ポートが指定されていない場合はデフォルトのポート 23 が使用されます。

net.udp.service アイテムのパラメータ

ntp

UDP を介して SNTP パケットを送信し、RFC 4330、セクション 5に従って応答を検証します。ポートが指定されていない場合は、デフォルトのポート 123 が使用されます。

10 proc.get パラメーター

概要

proc.get[<name>,<user>,<cmdline>,<mode>] アイテムは、Linux、Windows、FreeBSD、OpenBSD、NetBSD でサポートされています。

アイテムによって返されるプロセスパラメータのリストは、オペレーティングシステムと’mode’ 引数の値によって異なります。

Linux

各モードで Linux で返されるプロセスパラメータは以下の通りです。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親プロセスの PID	ppid: 親プロセスの PID	processes: プロセス数
name: プロセス名	name: プロセス名	vsize: 仮想メモリサイズ
cmdline: 引数付きコマンド	user: プロセスが実行されているユーザー (実ユーザー)	pmem: 実メモリの割合
user: プロセスが実行されているユーザー (実ユーザー)	group: プロセスが実行されているグループ (実グループ)	rss: 常駐セットサイズ

mode=process	mode=thread	mode=summary
group: プロセスが実行されているグループ (実グループ)	uid: ユーザー ID	data: データセグメントのサイズ
uid: ユーザー ID	gid: プロセスが実行されているグループの ID	exe: コードセグメントのサイズ
gid: プロセスが実行されているグループの ID	tid: スレッド ID	lib: 共有ライブラリのサイズ
vsize: 仮想メモリサイズ	tname: スレッド名	lck: ロックされたメモリのサイズ
pmem: 実メモリの割合	cputime_user: ユーザーモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)	pin: ピン留めされたページのサイズ
rss: 常駐セットサイズ	cputime_system: システムモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)	pte: ページテーブルエントリのサイズ
data: データセグメントのサイズ	state: スレッドの状態	size: プロセスのコード + データ + スタックセグメントのサイズ
exe: コードセグメントのサイズ	ctx_switches: コンテキストスイッチの回数	stk: スタックセグメントのサイズ
hwm: ピーク常駐セットサイズ	page_faults: ページフォールトの回数	swap: 使用中のスワップ領域のサイズ
lck: ロックされたメモリのサイズ		cputime_user: ユーザーモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)
lib: 共有ライブラリのサイズ		cputime_system: システムモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)
peak: ピーク仮想メモリサイズ		ctx_switches: コンテキストスイッチの回数
pin: ピン留めされたページのサイズ		threads: スレッド数
pte: ページテーブルエントリのサイズ		page_faults: ページフォールトの回数
size: プロセスのコード + データ + スタックセグメントのサイズ		pss: 比例セットサイズメモリ
stk: スタックセグメントのサイズ		
swap: 使用中のスワップ領域のサイズ		
cputime_user: ユーザーモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)		
cputime_system: システムモードで消費した合計 CPU 時間 (クロックティック単位 — 秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割る)		
state: プロセスの状態 (procfs から透過的に取得、ロングフォーム)		
ctx_switches: コンテキストスイッチの回数		
threads: スレッド数		
page_faults: ページフォールトの回数		
pss: 比例セットサイズメモリ		

BSD 系 OS

各モードで FreeBSD、OpenBSD、NetBSD で返されるプロセスパラメータは以下の通りです。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親プロセスの PID	ppid: 親プロセスの PID	processes: プロセス数
jid: ジェイルの ID(FreeBSD のみ)	jid: ジェイルの ID(FreeBSD のみ)	vsize: 仮想メモリサイズ
jname: ジェイル名 (FreeBSD のみ)	jname: ジェイル名 (FreeBSD のみ)	pmem: 実メモリの割合 (FreeBSD のみ)
name: プロセス名	name: プロセス名	rss: 常駐セットサイズ
cmdline: 引数付きコマンド	user: プロセスが実行されているユーザー (実ユーザー)	size: プロセスサイズ (コード + データ + スタック)
user: プロセスが実行されているユーザー (実ユーザー)	group: プロセスが実行されているグループ (実グループ)	tsize: テキスト (コード) サイズ
group: プロセスが実行されているグループ (実グループ)	uid: ユーザー ID	dsize: データサイズ
uid: ユーザー ID	gid: プロセスが実行されているグループの ID	ssize: スタックサイズ
gid: プロセスが実行されているグループの ID	tid: スレッド ID	cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)
vsize: 仮想メモリサイズ	tname: スレッド名	cputime_system: システムモードで消費した CPU 時間 (秒単位)
pmem: 実メモリの割合 (FreeBSD のみ)	cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)	ctx_switches: コンテキストスイッチ数
rss: 常駐セットサイズ	cputime_system: システムモードで消費した CPU 時間 (秒単位)	threads: スレッド数 (NetBSD は非対応)
size: プロセスサイズ (コード + データ + スタック)	state: スレッド状態	stk: スタックセグメントサイズ
tsize: テキスト (コード) サイズ	ctx_switches: コンテキストスイッチ数	page_faults: ページフォールト数
dsize: データサイズ	io_read_op: システムが入力を実行しなければならなかった回数	fds: ファイルディスクリプタ数 (OpenBSD のみ)
ssize: スタックサイズ	io_write_op: システムが出力を実行しなければならなかった回数	swap: 使用スワップサイズ
cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)		io_read_op: システムが入力を実行しなければならなかった回数
cputime_system: システムモードで消費した CPU 時間 (秒単位)		io_write_op: システムが出力を実行しなければならなかった回数
state: プロセス状態 (ディスクスリープ/実行中/スリープ/トレース停止/ゾンビ/その他)		
ctx_switches: コンテキストスイッチ数		
threads: スレッド数 (NetBSD は非対応)		
page_faults: ページフォールト数		
fds: ファイルディスクリプタ数 (OpenBSD のみ)		
swap: 使用スワップサイズ		
io_read_op: システムが入力を実行しなければならなかった回数		
io_write_op: システムが出力を実行しなければならなかった回数		

Windows

各モードで Windows で返されるプロセスパラメータは以下の通りです。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親プロセスの PID	ppid: 親プロセスの PID	processes: プロセス数
name: プロセス名	name: プロセス名	vmsize: 仮想メモリサ
user: プロセスを実行しているユーザー	user: プロセスを実行しているユーザー	wkset: プロセスのワー
sid: ユーザー SID	sid: ユーザー SID	cputime_user: ユーザー
vmsize: 仮想メモリサイズ	tid: スレッド ID	cputime_system: シス
wkset: プロセスのワーキングセットサイズ		threads: スレッド数
cputime_user: ユーザーモードで消費した CPU 時間の合計 (ミリ秒単位)		page_faults: ページフ

mode=process	mode=thread	mode=summary
cputime_system: システムモードで消費した CPU 時間の合計 (ミリ秒単位) threads: スレッド数 page_faults: ページフォールト数 handles: ハンドル数 io_read_b: IO で読み込んだバイト数 io_write_b: IO で書き込んだバイト数 io_read_op: IO の読み込み操作数 io_write_op: IO の書き込み操作数 io_other_b: 読み書き以外の IO で転送されたバイト数 io_other_op: 読み書き以外の IO 操作数		handles: ハンドル数 io_read_b: IO で読み込 io_write_b: IO で書き込 io_read_op: IO の読み込 io_write_op: IO の書き込 io_other_b: 読み書き以 io_other_op: 読み書き以

11 到達不能/使用不可のホストインターフェース設定

概要

いくつかの設定パラメータは、エージェントチェック（Zabbix、SNMP、IPMI、JMX）が失敗し、ホストインターフェイスに到達できなくなった場合の Zabbix サーバーの動作を定義します。

到達不能なインターフェース

ホストインターフェースは、Zabbix、SNMP、IPMI、または JMX エージェントによるチェックが失敗した場合（ネットワークエラー、タイムアウト）に到達不能と見なされます。Zabbix 6.2.0 以降、アクティブ Zabbix エージェントチェックもインターフェースの可用性に影響します。アクティブチェックが利用できなくなると、エージェントインターフェースの全体的な可用性ステータスに影響します。

インターフェースが到達不能になった瞬間から、**UnreachableDelay**は、アイテム（LLD ルールを含む）のいずれかを使用して再チェックされる頻度を定義します。これらの再チェックは、unreachable poller（または IPMI チェックの場合は IPMI poller）によって実行されます。デフォルトでは、連続する到達可能性チェックの間隔は 15 秒です。

> **Attention:**
> 非同期 poller によって実行されるチェックは、unreachable poller に移動されません。

Zabbix サーバーログでは、到達不能は次のようなメッセージで示されます:

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

ログメッセージには、失敗した正確なアイテムとそのタイプ（Zabbix エージェント）が指定されます。

> **Note:**
> *Timeout* パラメータも、到達不能時にインターフェースがどれだけ早く再チェックされるかに影響します。Timeout が 20 秒、UnreachableDelay が 30 秒に設定されている場合、次のチェックは最初の試行から 50 秒後になります。

UnreachablePeriod パラメータは、到達不能期間の合計時間を定義します。デフォルトでは UnreachablePeriod は 45 秒です。この値は、インターフェースが利用不可とマークされる前に複数回再チェックされるように、UnreachableDelay よりも数倍大きくする必要があります。

内部アイテムzabbix[host,active_agent,available]を使用すると、到達不能なシナリオでアクティブチェックの可用性を監視できます。

インターフェイスを到達可能に戻す

到達不能期間が終了すると、インターフェイスが再度ポーリングされ、インターフェイスを到達不能状態にしたアイテムの優先度が低下します。到達不能なインターフェイスが再び到達可能になると、監視は自動的に通常に戻ります。

```
resuming Zabbix agent checks on host "New host": connection restored
```

> **Note:**
> インターフェイスが使用可能になっても、ホストは次の 2 つの理由ですべてのアイテムをすぐにポーリングしません。
> - ホストが過負荷になる可能性があります。
> - インターフェイスの復元時間は、計画されたアイテムのポーリングスケジュール時間と常に一致するとは限りません。
>
> そのため、インターフェイスが使用可能になった後、アイテムはすぐにはポーリングされず、次のポーリングラウンドに再スケジュールされます。

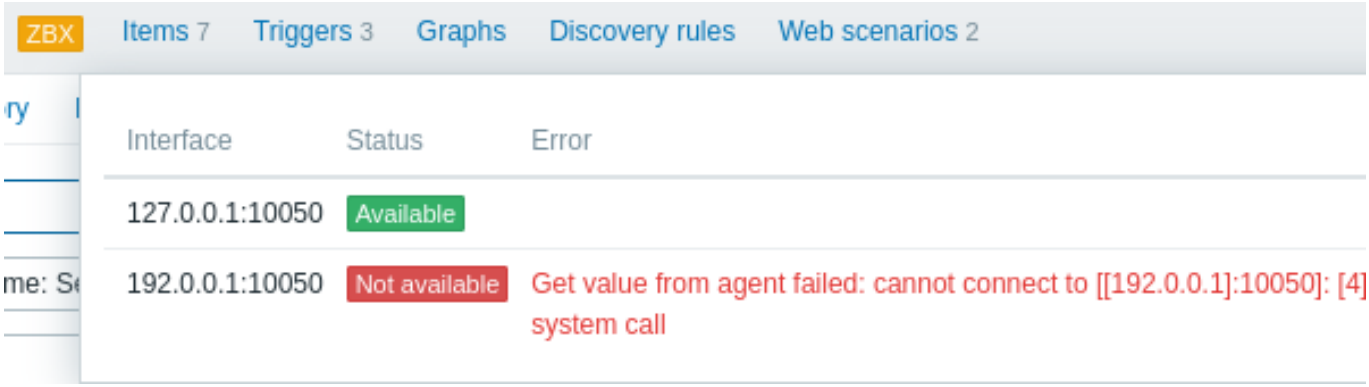
利用できないインターフェース

UnreachablePeriod が終了し、インターフェースが再び利用可能にならなかった場合、そのインターフェースは利用できないものとして扱われます。

サーバーログには、次のようなメッセージが表示されます。

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

また、フロントエンド では、ホストの可用性アイコンが緑/グレーから黄/赤に変わります（利用できないインターフェースの詳細は、ホストの可用性アイコンにマウスを合わせたときに表示されるヒントボックスで確認できます）。



UnavailableDelay パラメータは、インターフェースが利用できない間にどのくらいの頻度でインターフェースをチェックするかを定義します。

デフォルトでは 60 秒です（この場合、上記のログメッセージの「temporarily disabling」は、1 分間チェックを無効にすることを意味します）。

インターフェースへの接続が復旧すると、監視も自動的に通常状態に戻ります。

```
enabling Zabbix agent checks on host "New host": interface became available
```

12 Zabbix 統計のリモート監視

概要

Zabbix サーバーやプロキシの一部の内部メトリクスを、別の Zabbix インスタンスやサードパーティーツールからリモートでアクセスできるようにすることが可能です。これにより、サポーターやサービスプロバイダーがクライアントの Zabbix サーバー/プロキシをリモートで監視したり、Zabbix が主要な監視ツールでない組織において、Zabbix の内部メトリクスをサードパーティーシステムで傘型監視構成で監視したりすることができます。

Zabbix の内部統計情報は、新しい'StatsAllowedIP' server/proxy パラメータでリストされた設定可能なアドレスセットに公開されます。リクエストはこれらのアドレスからのみ受け付けられます。

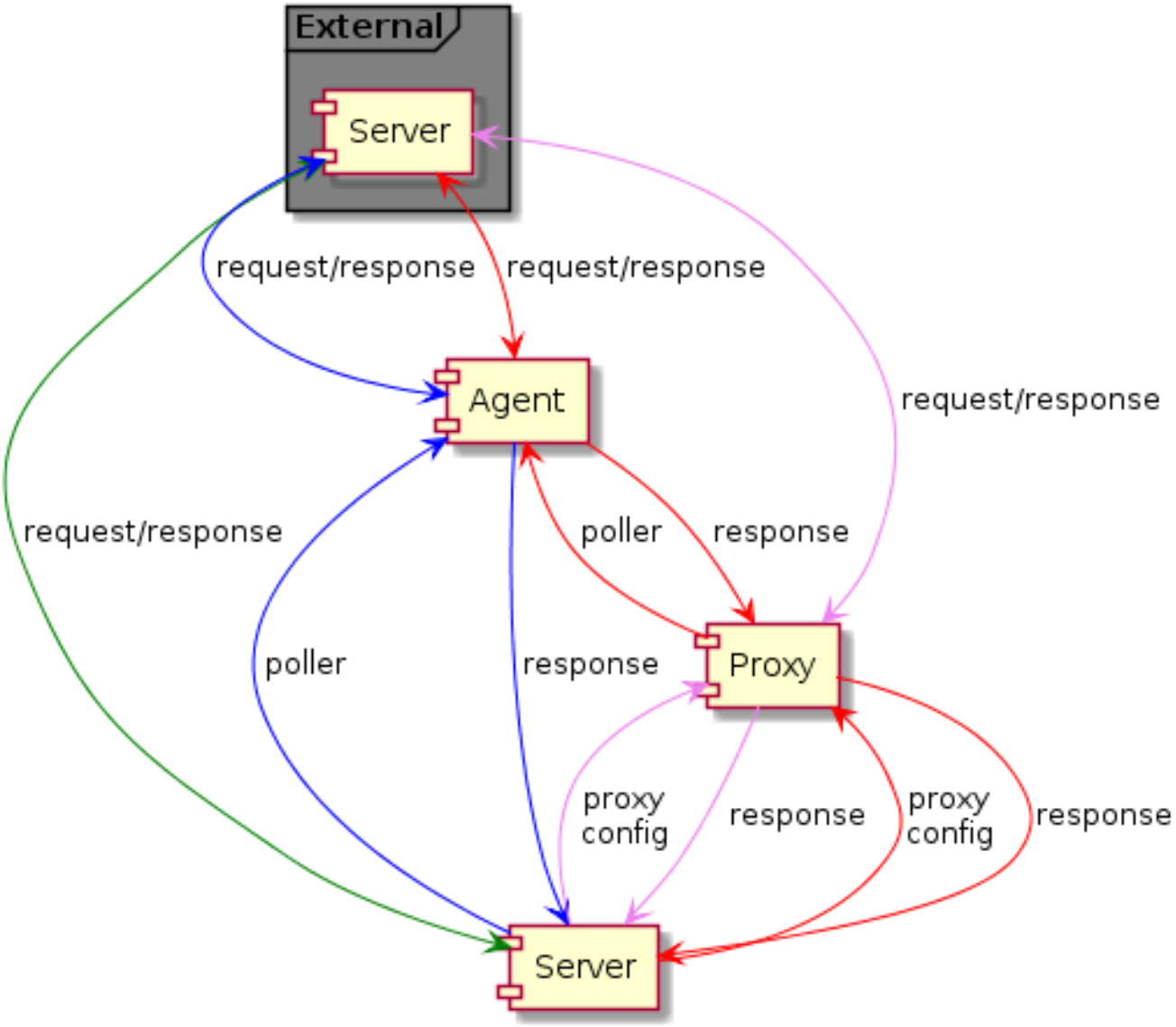
アイテム

別の Zabbix インスタンスで内部統計のクエリを設定するには、次の 2 つの項目を使用します。

- `zabbix[stats,<ip>,<port>]` internal item - Zabbix サーバー/プロキシの直接リモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用します
- `zabbix.stats[<ip>,<port>]` agent item - Zabbix サーバー/プロキシのエージェントベースのリモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用されます。

参考情報: 内部項目, Zabbix エージェントアイテム

次の図はコンテキストに応じた、いずれかのアイテムの使用法を示しています。



- - Server → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → proxy → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])
- - Server → proxy → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])

ターゲットインスタンスが外部インスタンスによるクエリを許可するには、ターゲットインスタンスの'StatsAllowedIP' パラメータに外部インスタンスのアドレスをリストします。

収集されるメトリクス

stats アイテムは統計情報をまとめて収集し、JSON 形式で返します。これは依存アイテムがデータを取得するための基礎となります。以下の内部メトリクスが、いずれかのアイテムによって返されます。

- zabbix[boottime]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[preprocessing] (サーバーのみ)
- zabbix[preprocessing_queue] (サーバーのみ)
- zabbix[process,<type>,<mode>,<state>] (プロセスタイプベースの統計のみ)
- zabbix[rcache,<cache>,<mode>]
- zabbix[requiredperformance]
- zabbix[triggers] (サーバーのみ)
- zabbix[uptime]
- zabbix[vcache,buffer,<mode>] (サーバーのみ)

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] ('trends' キャッシュタイプはサーバーのみ)

テンプレート

テンプレートは、外部インスタンスから Zabbix サーバーまたはプロキシの内部メトリクスをリモート監視するために利用できます:

- リモート Zabbix サーバーヘルス
- リモート Zabbix プロキシヘルス

複数の外部インスタンスのリモート監視用テンプレートを使用するには、各外部インスタンス監視ごとに個別のホストが必要であることに注意してください。

トラッパープロセス

外部の Zabbix インスタンスからの内部メトリックリクエストの受信は、トラッパープロセスによって処理されます。このプロセスはリクエストを検証し、メトリックを収集し、JSON データバッファを作成し、準備した JSON を返します。例えば、サーバーからは以下のようになります。

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
```

```
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```


内部キューアイテム

別の Zabbix インスタンスで内部キュー統計をリモートクエリできるようにする別の 2 つの項目もあります。

- zabbix[stats,<ip>,<port>,queue,<from>,<to>] internal item - リモート Zabbix サーバー/プロキシへの直接内部キュークエリ用
- zabbix.stats[<ip>,<port>,queue,<from>,<to>] agent item - リモート Zabbix サーバー/プロキシへのエージェントベースの内部キュークエリ用

参考情報: 内部項目, Zabbix エージェントアイテム

13 Zabbix を使用した **Kerberos** の設定

概要

Kerberos 認証は、Zabbix の Web 監視や HTTP アイテムで使用できます。

このページでは、Debian/Ubuntu で Zabbix プロセス用の Kerberos プリンシパルを使用して、Zabbix サーバーが `www.example.com` の Web 監視を実行するための Kerberos の設定例について説明します。

設定

1. KDC とクライアントユーティリティをインストールします:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

パッケージのセットアップ中にプロンプトに答えます。例:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost（または FQDN）
Administrative server for your Kerberos realm: localhost（または FQDN）
```

2. フレンドリーホスト名を割り当てます（オプション、ローカルテスト用）。

DNS がない場合は、/etc/hosts を編集し、DC と Web サーバーのエントリを追加します:

```
sudo vi /etc/hosts
```

追加する例:

```
192.168.1.100  dc01.example.com dc01
```

3. Kerberos クライアントと KDC レルムを設定します:

```
sudo vi /etc/krb5.conf
```

設定例:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

`.localdomain` やその他の非公開名を使用する場合は、ホスト名 → レルムのマッピングが機能するように明示的なドメイン → レルムのマッピングを追加してください。ここで不一致があると `Server not found in Kerberos database` エラーが発生します。

4. Kerberos データベースを初期化します（KDC ホストで一度だけ実行）。プロンプトが表示されたら安全なマスターパスワードを設定します:

```
sudo krb5_newrealm
```

5. クライアントが使用する正確なホスト名で HTTP/host.fqdn@REALM プリンシパルを作成します。小文字を推奨します（例: HTTP/web.example.com@EXAMPLE.COM）。大文字/小文字や名前の不一致は Server not found in Kerberos database の原因となります。

```
sudo kadmin.local
```

kadmin.local 内で:

```
addprinc kerb-admin@EXAMPLE.COM      # 管理用プリンシパル
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

keytab を Web ホストに移動（同一マシンの場合はそのまま）し、Apache が使用できるようにパーミッションを設定します:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### 確認
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Apache GSSAPI モジュールをインストールして有効化します:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> すべての mod_auth_gssapi バージョンがすべての Gssapi* ディレクティブをサポートしているわけではありません。Apache が Invalid command 'GssapiCredStore'で失敗する場合は、サポートされていないディレクティブを削除するか、モジュールをアップグレードしてください。

7. VirtualHost を設定します（DocumentRoot/Zabbix UI へのパスを調整）:

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

zabbix.conf 内で:

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Apache を再起動します:

```
sudo systemctl restart apache2
```

8. KDC サービスを有効化/起動し、リッスンポートを確認します（KDC ホスト）:

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # または: sudo netstat -tnlp | grep :80
```

9. テスト用に TGT を取得します（チケットを使用するユーザーで実行）。

チケットリストに krbtgt/EXAMPLE.COM@EXAMPLE.COM が表示されることを期待します。チケットが必要な OS ユーザー（Web

チェック用の zabbix や、ブラウザ SSO テスト用の www-data/Apache など）で kinit を実行します。異なる OS ユーザーに発行されたチケットは、KRB5CCNAME とパーミッションを調整しない限り表示されません。

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. curl で SPNEGO 交換をテストします（有効な TGT を持つクライアントから）。200 OK（またはアプリへのリダイレクト）が表示されれば SPNEGO は成功です:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. オプションで、Zabbix UI が HTTP 認証ログインを受け入れる場合は、Zabbix フロントエンド（ui/conf/zabbix.conf.php）で HTTP 認証を有効にします:

```
$ALLOW_HTTP_AUTH = true;
```

Web UI でユーザー > 認証に移動し、*HTTP* 設定タブに移動します。*HTTP* 認証を有効にするチェックボックスをマークし、ポップアップで *OK* をクリックします。デフォルトのログインフォームドロップダウンで「HTTP ログインフォーム」を選択します。大文字と小文字を区別するログインがディレクトリポリシーに合うかどうかを決定します。更新ボタンをクリックして完了します。

12. ブラウザの設定（例として Firefox を使用）: network.negotiate-auth.trusted-uris を Negotiate を実行するホスト（dc01.example.com）に設定し、ブラウザが自動的に Kerberos トークンを送信するようにします。

about:config 内で:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

これで http://dc01.example.com にアクセスすると、フォームなしで Zabbix に直接ログインできるはずです。

13. キー/チケットを常に新鮮に保ちます。Kerberos チケットのデフォルト有効期間は約 10 時間です。有効期限切れを防ぐために cron/systemd タイマーを追加します:

```
####Web サービス用
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
#### 監視ユーザー用
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. 保守チェック:

- klist -k /etc/apache2/http.keytab — keytab にサービスプリンシパルが存在することを確認します。
- sudo tail -f /var/log/apache2/error.log — GSSAPI エラーを監視します（gss_acquire_cred[_from]() failed to get server creds は keytab/パーミッションまたはプリンシパルの不足を意味します）。
- curl --negotiate が 401/403 を返す場合は、プリンシパルの間違い、チケットの不在、ホストヘッダの不一致、ファイルシステムのパーミッション問題などが考えられます。ログと/etc/krb5.conf のドメインマッピングを確認してください。

セキュリティとファイルパーミッションに関する注意事項

Keytab ファイルは、それを必要とするアカウントのみが読み取り可能でなければなりません。例: zabbix ユーザーの keytab の場合は zabbix:zabbix 所有の 0400、Apache の keytab の場合は root:www-data 所有の 0440。

ホストに長期間有効な平文パスワードを保存するのは避けてください。可能な場合は keytab やドメイン参加済みのマシンプリンシパルを使用してください。

KRB5CCNAME を設定したり keytab をコピーするテストやスクリプトを実行する場合、操作後に所有権とパーミッションを再確認してください。Web サーバーが認証情報を拒否する場合、ファイルパーミッションの問題であることがよくあります。

14 modbus.get パラメーター

概要

以下の表は、アイテムmodbus.get のパラメーターの詳細を示しています。

パラメーター

パラメーター	説明	デフォルト	例
endpoint	プロトコルとエンドポイントのアドレス。 `protocol://connection_string` で定義 プロトコルの値: *rtu*, *ascii* (Agent 2 のみ), *tcp* コネクションストリングのフォーマット: *tcp* の場合 - `address:port` シリアルラインの場合: *rtu*, *ascii* - `port_name:speed:params` ここで 'speed' - 1200, 9600 など 'params' - データビット (5,6,7 または 8)、パリティ (n,e,o のいずれか、none/even/odd)、ストップビット (1 または 2)	プロトコル: なし *rtu/ascii* プロトコル: port_name: なし speed: 115200 params: 8n1 *tcp* プロトコル: address: なし port: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
slave id	対象デバイスの Modbus アドレス (1～247)。MODBUS Messaging Implementation Guide (23 ページ) を参照 tcp デバイス (GW 以外) はこのフィールドを無視	serial: 1 tcp: 255 (0xFF)	2
function	空、またはサポートされている関数の値: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	空	3
address	最初のレジスタ、コイル、または入力のアドレス。 'function' が空の場合、'address' は以下の範囲で指定: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 'function' が空でない場合、'address' フィールドは 0～65535 の範囲で、修正なしで使用 (PDU)	function が空: 00001 function が空でない: 0	9999
count	デバイスから読み取る連続した'type' の数。 Coil または Discrete input の場合、'type' = 1 ビット その他の場合: (count*sizeof(type))/2 = 実際に読み取るレジスタ数 'offset' が 0 でない場合、その値が' 実際の数' に加算される 許容範囲は 1:65535	1	2
type	データ型: Read Coil, Read Discrete Input の場合 - *bit* Read Holding Registers, Read Input Registers の場合: *int8* - 8 ビット *uint8* - 8 ビット (符号なし) *int16* - 16 ビット *uint16* - 16 ビット (符号なし) *int32* - 32 ビット *uint32* - 32 ビット (符号なし) *float* - 32 ビット *uint64* - 64 ビット (符号なし) *double* - 64 ビット	bit uint16	uint64

パラメーター	説明	デフォルト	例
endianness	エンディアンタイプ: *be* - ビッグエンディアン *le* - リトルエンディアン *mbe* - ミッドビッグエンディアン *mle* - ミッドリトルエンディアン 制限: 1 ビットの場合 - be 8 ビットの場合 - be,le 16 ビットの場合 - be,le	be	le
offset	'address' から始まるレジスタ数で、その結果は破棄される。 各レジスタのサイズは 16 ビット (ランダムリードアクセスをサポートしない機器のために必要)	0	4

15 VMware のカスタムパフォーマンスカウンター名作成

概要

VMware のパフォーマンスカウンターパスは `group/counter[rollup]` という形式であり、以下のようになります。

- `group` - パフォーマンスカウンターグループ（例：*cpu*）
- `counter` - パフォーマンスカウンター名（例：*usagemhz*）
- `rollup` - パフォーマンスカウンターのロールアップタイプ（例：*average*）

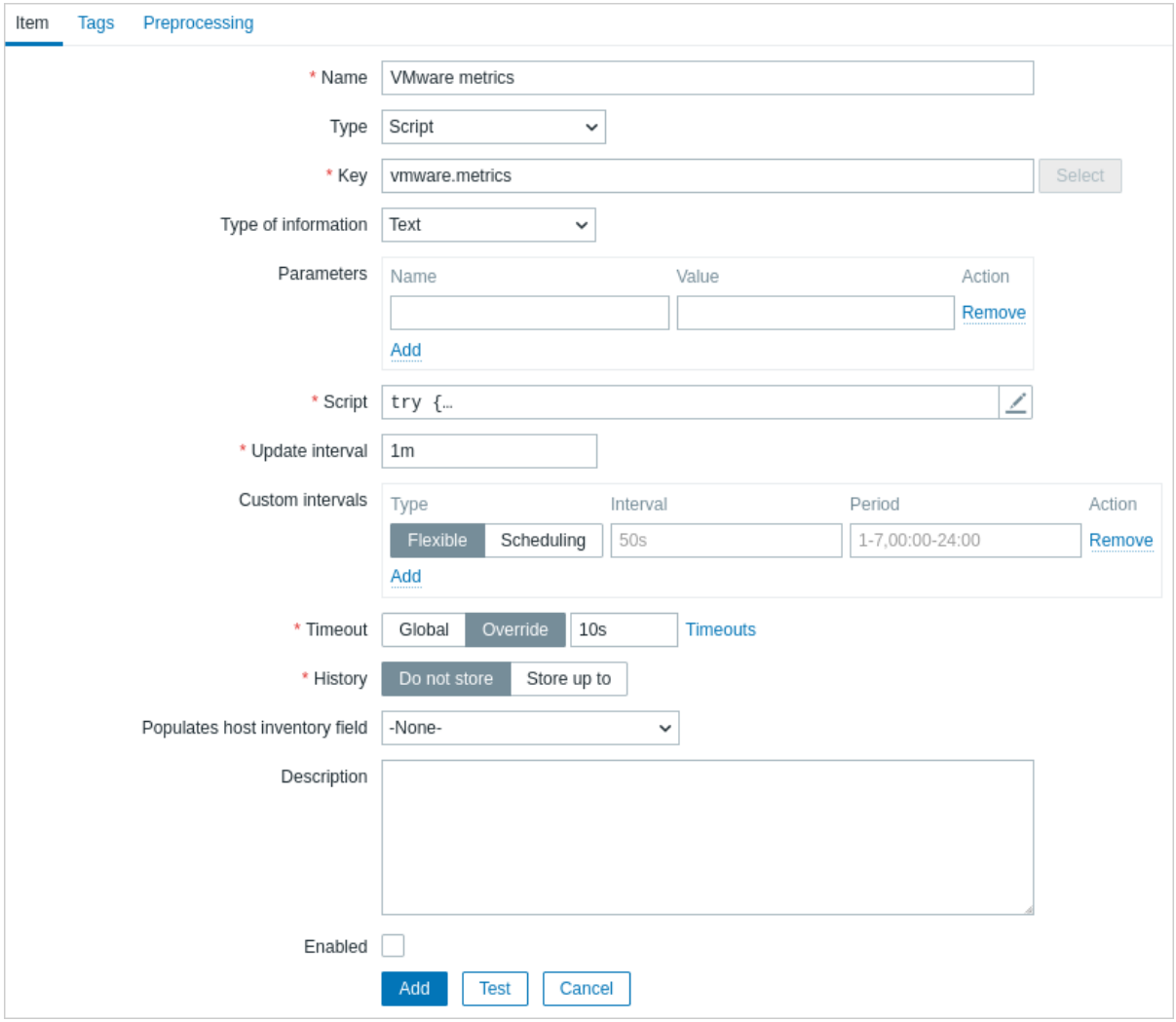
したがって、上記の例では次のようなカウンターパスになります：`cpu/usagemhz[average]`

パフォーマンスカウンターグループの説明、カウンター名、ロールアップタイプについては、VMware のドキュメントを参照してください。

Zabbix のスクリプトアイテムを使用することで、内部名を取得したり、カスタムのパフォーマンスカウンター名を作成したりすることができます。

設定

1. メインの VMware ホスト（**eventlog[]** アイテムが存在するホスト）で、以下のパラメータで無効化されたスクリプトアイテムを作成します。



- 名前: VMware metrics
- タイプ: スクリプト
- キー: vmware.metrics
- 情報のタイプ: テキスト
- スクリプト: 下記のスクリプトをコピー＆ペースト
- タイムアウト: 10
- 履歴: 保存しない
- 有効: チェックを外す

スクリプト

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

アイテムを設定したら、テストボタンを押し、値の取得を押します。



取得した XML を任意の XML フォーマッタにコピーし、目的のメトリックを探します。

1 つのメトリックに対する XML の例:

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
    <key>6</key>
```

```
    <nameInfo>
        <label>Usage in MHz</label>
        <summary>CPU usage in megahertz during the interval</summary>
        <key>usagemhz</key>
    </nameInfo>
    <groupInfo>
        <label>CPU</label>
        <summary>CPU</summary>
        <key>cpu</key>
    </groupInfo>
    <unitInfo>
        <label>MHz</label>
        <summary>Megahertz</summary>
        <key>megaHertz</key>
    </unitInfo>
    <rollupType>average</rollupType>
    <statsType>rate</statsType>
    <level>1</level>
    <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

取得した XML からカウンターパスを抽出するには XPath を使用します。上記の例の場合、XPath は以下のようになります。

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

この場合のパフォーマンスカウンターパスは: `cpu/usagemhz[average]` です。

16 system.sw.packages.get の戻り値

概要

このセクションでは、`system.sw.packages.get` Zabbix エージェントアイテムの戻り値の詳細について説明します。

詳細

このアイテムの出力は、以下のキーを含むオブジェクトの配列です:

- **name** - パッケージ名
- **manager** - このデータを報告したパッケージマネージャー（rpm、dpkg、pacman、pkgtool、または portage）
- **version** - パッケージバージョン
- **size** - バイト単位の非圧縮パッケージサイズ（利用できない場合は 0 を設定）
- **arch** - パッケージアーキテクチャ
- **buildtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがビルドされた UNIX タイムスタンプ（利用できない場合は 0 を設定）
  - **value** - パッケージがビルドされた日時（利用できない場合は空文字列を設定）
- **installtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがインストールされた UNIX タイムスタンプ（利用できない場合は 0 を設定）
  - **value** - パッケージがインストールされた日時（利用できない場合は空文字列を設定）

例:

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
```

```
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

17 net.dns.get の戻り値

概要

このセクションでは、net.dns.get Zabbix agent 2 アイテムの戻り値の詳細を説明します。

詳細

このアイテムの出力は、アイテムキーで指定されたパラメータに基づいて DNS レコード情報を含むオブジェクトです。

例えば、net.dns.get[,example.com] アイテムは、拒否されたクエリの以下の JSON を返す場合があります:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

DNS サーバーの IP アドレスを指定することで、net.dns.get[192.0.2.0,example.com] アイテムは以下の JSON を返す場合があります:

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
                "ns": "ns.example.org.",
                "refresh": 7200,
```

```
                "retry": 3600,
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


接続に問題がある場合、net.dns.get[192.0.2.0,example.com] アイテムはエラーを返す場合があります:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


以下の種類のエラーコードが考えられます:

シナリオ	"zbx_error_code"	"zbx_error_msg"
エラーなし、かつ DNS 応答が受信され解析された	0	
DNS がダウンしている	-1	"Communication error"
JSON 解析中にエラーが発生	-2	"Received unexpected response"

追加パラメータを指定すると、net.dns.get[192.0.2.0,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,aafla
アイテムは以下の JSON を返す場合があります:


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
            ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
```

```
                "a": "192.0.2.0"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


関連情報

DNS レコードの詳細については、以下を参照してください。

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

18 Windows の system.cpu.util アイテムに関する注意事項

system.cpu.util アイテムは、CPU 使用率のパーセンテージを提供します。

Windows 用 Zabbix エージェントでコレクタープロセスが開始されると、N 個の論理プロセッサ (スレッド) ごとに N 個のアイテム用のバッファが初期化されます。値は Windows パフォーマンスカウンターを使用してバッファ内で定期的に更新されます。これらの値は、0 <= n < N の範囲で system.cpu.util[n] が要求されたときに報告されます。

バッファ内のもう 1 つのアイテムは、system.cpu.util[total] に使用されます。

Zabbix は、NUMA システムとプロセッサグループをサポートしない古い Windows バージョンの両方をサポートするために、異なるパフォーマンスカウンターを自動的に選択します。

論理プロセッサ (スレッド) 数が 64 以下で、プロセッサグループ数が 1 の場合は、次のようになります。

```
N は論理プロセッサ（スレッド）数です。

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

それ以外の場合:

```
G はプロセッサグループ数、N はグループ内の論理プロセッサ（スレッド）数です。

\Processor Information(_Total)\% Processor Time
\Processor Information(0,0)\% Processor Time
\Processor Information(0,1)\% Processor Time
\Processor Information(0,2)\% Processor Time
...
\Processor Information(0,N-1)\% Processor Time
...
\Processor Information(G-1,0)\% Processor Time
\Processor Information(G-1,1)\% Processor Time
\Processor Information(G-1,2)\% Processor Time
...
\Processor Information(G-1,N-1)\% Processor Time
```

5 サポートされている関数

詳細については、それぞれの関数グループをクリックしてください。

関数グループ		関数
集計関数		avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
	Foreach 関数	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
ビット演算関数		bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
日付・時刻関数		date, dayofmonth, dayofweek, now, time
履歴関数		change, changecount, count, countunique, find, first, firstclock, fuzzytime, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate

関数グループ	関数
トレンド関数	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
数学関数	abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
演算子関数	between, in
予測関数	forecast, timeleft
文字列関数	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

特に記載がない限り、これらの関数は以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

Foreach 関数は集計計算でのみサポートされています。

1 集計関数

特に記載がない限り、ここに記載されているすべての関数は以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

集計関数は、以下のいずれかで動作します:

- アイテムの履歴、例: `min(/host/key,1h)`
- foreach 関数を唯一のパラメータとして、例: `min(last_foreach(/*/key))`（計算アイテムのみ; トリガーでは使用できません）

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
avg	定義された評価期間内のアイテムの平均値。
bucket_percentile	ヒストグラムのバケットからパーセンタイルを計算します。
count	foreach 関数によって返された配列内の値の数。
histogram_quantile	ヒストグラムのバケットから φ-分位数を計算します。
item_count	フィルタ条件に一致する設定内の既存アイテムの数。
kurtosis	定義された評価期間内の収集値の確率分布の「尖度」。
mad	定義された評価期間内の収集値の中央値絶対偏差。

関数	説明
max	定義された評価期間内のアイテムの最大値。
min	定義された評価期間内のアイテムの最小値。
skewness	定義された評価期間内の収集値の確率分布の非対称性。
stddevpop	定義された評価期間内の収集値の母集団標準偏差。
stddevsamp	定義された評価期間内の収集値の標本標準偏差。
sum	定義された評価期間内の収集値の合計。
sumofsquares	定義された評価期間内の収集値の二乗和。
varpop	定義された評価期間内の収集値の母集団分散。
varsamp	定義された評価期間内の収集値の標本分散。

共通パラメータ

- /host/key は、ホストアイテムの履歴を参照する関数の共通の必須最初のパラメータです。
- (sec|#num)<:time shift> は、ホストアイテムの履歴を参照する関数の共通の 2 番目のパラメータで、以下の通りです:
  - **sec** - 最大評価期間（秒単位、サフィックスが使用可能）、または
  - **#num** - 最新の収集値の最大評価範囲（ハッシュ記号が先頭に付く場合）
  - **time shift**（オプション）は、評価ポイントを過去に移動することができます。タイムシフトの指定方法については詳細をご覧ください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数で説明されています
- /host/key および (sec|#num)<:time shift> パラメータは決して引用符で囲まないでください

avg(/host/key,(sec|#num)<:time shift>)

定義された評価期間内のアイテムの平均値。
 サポートされている値の型: *Float*、*Integer*。
 サポートされているforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*。

パラメータ: 共通パラメータを参照。

time shift は、現在の平均値と過去の平均値を比較する必要がある場合に便利です。

例:

```
avg(/host/key,1h) #現在までの直近1時間の平均値
avg(/host/key,1h:now-1d) #現在から25時間前から24時間前までの1時間の平均値
avg(/host/key,#5) #最新5つの値の平均値
avg(/host/key,#5:now-1d) #直近24時間で受信した値を除く最新5つの値の平均値
```

bucket_percentile(item filter,time period,percentage)

ヒストグラムのバケットからパーセンタイルを計算します。

パラメータ:

- **item filter** - アイテムフィルタを参照;
- **time period** - 時間期間を参照;
- **percentage** - パーセンテージ (0-100)。

コメント:

- 計算アイテムでのみサポートされています;
- この関数は histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1)) のエイリアスです。

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

foreach 関数によって返された配列内の値の数。
 サポートされているforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ:

- **func_foreach** - 返された値の数をカウントする foreach 関数。foreach 関数の詳細を参照してください。count_foreach と bucket_rate_foreach は追加パラメータをサポートしていることに注意してください。
- **item filter** - アイテムフィルタを参照;

- **time period** - 時間期間を参照;
- **operator** (ダブルクォーテーションで囲む必要があります)。サポートされている operator:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字小文字を区別）
bitand - ビットごとの AND
regexp - pattern で指定された正規表現の大文字小文字を区別した一致
iregexp - pattern で指定された正規表現の大文字小文字を区別しない一致
- **pattern** - 必要なパターン（文字列引数はダブルクォーテーションで囲む必要があります）; *operator* が 3 番目のパラメータで指定されている場合にサポートされます。

コメント:

- **count()** を履歴関連の foreach 関数（max_foreach、avg_foreach など）と一緒に使用するとパフォーマンスに影響を与える可能性がありますが、**exists_foreach()**（構成データのみで動作）はそのような影響はありません。
- オプションのパラメータ *operator* または *pattern* は、カンマの後に空のままにすることはできず、完全に省略する必要があります。
- 3 番目のパラメータとして *bitand* を使用する場合、4 番目の pattern パラメータは「/」で区切られた 2 つの数値として指定できます：**number_to_compare_with/mask**。count() は値と *mask* から「ビットごとの AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビットごとの AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定すればよく（「/」なし）、*number_to_compare_with* は省略できます。
- 3 番目のパラメータとして *regexp* または *iregexp* を使用する場合、4 番目の pattern パラメータは通常の正規表現またはグローバル（「@」で始まる）正規表現にできます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定から継承されます。regexp マッチングの目的で、float 値は常に「.」の後に 4 桁の小数で表されます。また、大きな数値の場合、10 進数（データベースに保存）と 2 進数（Zabbix サーバーで使用）の表現の違いにより、4 番目の小数点以下の桁に影響を与える可能性があることに注意してください。

例:

```
count(max_foreach(/*/net.if.in[*],1h)) #今から直近1時間でデータを受信したnet.if.inアイテムの数
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #ディスク使用率が95%を超えるファイルシステムの数
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

ヒストグラムのバケットから φ-分位数を計算します。
 サポートされているforeach 関数: *bucket_rate_foreach*。

パラメータ:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - 手動で入力するパラメータのペア (2 つ以上) またはbucket_rate_foreachの応答。

コメント:

- 計算アイテムでのみサポートされます;
- 機能的には PromQL の'histogram_quantile' に対応します;
- 最後の'Infinity' バケット (*"+inf"*) の値が 0 の場合は-1 を返します。

例:

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

フィルター条件に一致する設定済みアイテムの数。
 サポートされる値の型: 整数。

パラメータ:

- **item filter** - アイテム選択のための条件。ホストグループ、ホスト、アイテムキー、タグによる参照が可能。ワイルドカードがサポートされている。詳細はitem filterを参照。

コメント:

- 計算アイテムでのみサポートされる;
- *count(exists_foreach(item_filter))* 関数のエイリアスとして動作する。

例:

```
item_count(/*/agent.ping?[group="Host group 1"]) #「Host group 1」内で*agent.ping*アイテムを持つホストの数
```

kurtosis(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の確率分布の「尖度」。詳細は Kurtosis を参照してください。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータ を参照してください。

例:

```
kurtosis(/host/key,1h) #直近1時間の尖度
```

mad(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の中央値絶対偏差。参照: 中央値絶対偏差。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
mad(/host/key,1h) #現在までの直近1時間の中央値絶対偏差
```

max(/host/key,(sec|#num)<:time shift>)

定義された評価期間内のアイテムの最大値。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
max(/host/key,1h) - min(/host/key,1h) #現在までの直近1時間の最大値と最小値の差分（値のデルタ）を計算する
```

min(/host/key,(sec|#num)<:time shift>)

定義された評価期間内のアイテムの最小値。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
max(/host/key,1h) - min(/host/key,1h) #現在までの直近1時間の最大値と最小値の差分(値のデルタ)を計算する
```

skewness(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の確率分布の非対称性。参考: Skewness。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
skewness(/host/key,1h) #現在までの直近1時間の歪度
```

stddevpop(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の母集団標準偏差。参考: 標準偏差。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
stddevpop(/host/key,1h) #直近1時間の母集団標準偏差
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の標本標準偏差です。参考: 標準偏差。
 サポートされている値のタイプ: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

この関数を動作させるには、少なくとも 2 つのデータ値が必要です。

例:

```
stddevsamp(/host/key,1h) #現在までの直近1時間の標本標準偏差
```

sum(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の合計。
 サポートされている値のタイプ: *Float*, *Integer*。
 サポートされているforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
sum(/host/key,1h) #現在までの直近1時間の値の合計
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の二乗和。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach関数: *last_foreach*。

パラメータ: 共通パラメータを参照。

例:

```
sumofsquares(/host/key,1h) #現在までの直近1時間の二乗和
```

varpop(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の母分散。分散も参照してください。
 サポートされている値の型: *Float*, *Integer*。
 サポートされているforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
varpop(/host/key,1h) #現在までの直近1時間の母分散
```

varsamp(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の標本分散。参考: 分散。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照。

この関数を動作させるには、少なくとも 2 つのデータ値が必要です。

例:

```
varsamp(/host/key,1h) #現在までの直近1時間の標本分散
```

サポートされているすべての関数を参照してください。

1 Foreach 関数

概要

Foreach 関数は集計計算で使用され、使用されたアイテムフィルタによって選択された各アイテムの集計値を 1 つ返します。値の配列が返されます。

たとえば、*avg_foreach* 関数は、指定された時間間隔内に選択されたアイテムの平均履歴値ごとに値の配列を返します。

item フィルタは foreach 関数で使用される構文の一部です。アイテムフィルタではワイルドカードの使用がサポートされているため、必要なアイテムを柔軟に選択できます。

サポートされている関数

関数	説明
avg_foreach	各アイテムの平均値を返します。
bucket_rate_foreach	histogram_quantile()関数で使用するのに適したペア（バケットの上限、レート値）を返します。ここで「バケットの上限」は、< パラメータ番号 > パラメータで定義されたアイテムキーのパラメータの値です。
count_foreach	各アイテムの値の数を返します。
exists_foreach	有効な各アイテムに対して「1」を返します。
last_foreach	各アイテムの最新値を返します。
max_foreach	各アイテムの最大値を返します。
min_foreach	各アイテムの最小値を返します。
sum_foreach	各アイテムの値の合計を返します。

関数の構文

Foreach 関数は、2 つの共通パラメータ、アイテムフィルタ（詳細は下記参照）と時間期間をサポートします。

```
foreach_function(item filter,time period)
```

例：

`avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)`

は、MySQL サーバーグループ内の各'mysql.qps' アイテムの 5 分間の平均値を返します。

一部の関数は追加のパラメータをサポートしていることに注意してください。

アイテムフィルタの構文

アイテムフィルタ:

`/host/key[parameters]?[conditions]`

は 4 つの部分で構成されます:

- host - ホスト名
- key - アイテムキー (パラメータなし)
- parameters - アイテムキーパラメータ
- conditions - ホストグループおよび/またはアイテムタグに基づく条件 (式として)

スペースは条件式の中でのみ許可されます。

ワイルドカードの使用

- ワイルドカードは、ホスト名、アイテムキー、または個々のアイテムキーパラメータを置き換えるために使用できます。
- ホストまたはアイテムキーのいずれかは、ワイルドカードなしで指定する必要があります。したがって、`/host/*` および `/*/key` は有効なフィルタですが、`/*/*` は無効です。
- ワイルドカードは、ホスト名、アイテムキー、アイテムキーパラメータの一部には使用できません。
- ワイルドカードは、1 つのアイテムキーパラメータ以上には一致しません。したがって、各パラメータごとにワイルドカードを個別に指定する必要があります (例: `key[abc,*,*]`)。

条件式

条件式は以下をサポートします:

- オペランド:
  - group - ホストグループ
  - tag - アイテムタグ
  - "<text>" - 文字列定数。" および \ をエスケープするには \ を使用
- 大文字小文字を区別する文字列比較演算子: =, <>
- 論理演算子: and, or, not
- 括弧によるグループ化: ( )

文字列定数の引用は必須です。大文字小文字を区別した完全一致のみがサポートされます。

Warning:
フィルタでタグを指定する場合 (例: `tag="tagname:value"`)、コロン":" は区切り文字として使用されます。コロン以降はすべてタグ値と見なされます。したがって、現時点では":" を含むタグ名の指定はサポートされていません。

例

アイテムキー、ホストグループ、タグを参照する複雑なフィルタを使用することができ、以下の例で示されています:

構文例	説明
`/host/key[abc,*]`	このホスト上の類似アイテムに一致します。
`/*/key`	任意のホストの同じアイテムに一致します。
`/*/key?[group="ABC" and tag="tagname:value"]`	ABC グループの任意のホストで'tagname:value' タグを持つ同じアイテムに一致します。
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	ABC または DEF グループの任意のホストで、それぞれのタグを持つ類似アイテムに一致します。

参照されるすべてのアイテムは存在し、データを収集している必要があります。有効なホスト上の有効なアイテムのみが計算に含まれます。サポートされていない状態のアイテムは含まれません。

Attention:
参照アイテムのアイテムキーが変更された場合、フィルタは手動で更新する必要があります。

親ホストグループを指定すると、親グループとすべてのネストされたホストグループおよびそのアイテムが含まれます。

期間

2 番目のパラメーターでは、集計の期間を指定できます。期間は時間でのみ表現でき、値の数（**#** で始まる）はサポートされていません。

このパラメーターには、利便性のためにサポートされている単位記号を使用できます。たとえば、'300s'（300 秒）の代わりに'5m'（5 分）、'86400'（86400 秒）の代わりに'1d'（1 日）などです。

last_foreach 関数の期間はオプションのパラメーターです（Zabbix 7.0 以降でサポート）。省略することもできます。

```
last_foreach(/*/key?[group="host group"])
```

exists_foreach 関数では期間はサポートされていません。

追加パラメータ

bucket_rate_foreach

bucket_rate_foreach 関数では、3 番目のオプションパラメータがサポートされています:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

ここで、<parameter number> はアイテムキー内の「bucket」値の位置です。例えば、myItem[aaa,0.2] の「bucket」値が'0.2' の場合、その位置は 2 です。

<parameter number> のデフォルト値は「1」です。

count_foreach

count_foreach 関数では、3 番目と 4 番目のオプションパラメータがサポートされています:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

ここで:

- **operator** はアイテム値の条件演算子（ダブルクォートで囲む必要があります）。サポートされている `operator`:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字小文字を区別）
bitand - ビットごとの AND
regexp - `pattern` で指定された正規表現に大文字小文字を区別して一致
iregexp - `pattern` で指定された正規表現に大文字小文字を区別せずに一致
- **pattern** は必要なパターン（文字列引数はダブルクォートで囲む必要があります）；*operator* が 3 番目のパラメータで指定されている場合にサポートされます。

コメント:

- オプションパラメータ *operator* または *pattern* は、カンマの後に空のままにすることはできません。完全に省略する必要があります。
- 3 番目のパラメータが *bitand* の場合、4 番目の `pattern` パラメータは「/」で区切った 2 つの数値として指定できます：**number_to_compare_with/mask**。count_foreach() は値と *mask* の「ビットごとの AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビットごとの AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます（「/」なし）。
- 3 番目のパラメータが *regexp* または *iregexp* の場合、4 番目の `pattern` パラメータは通常の正規表現またはグローバル（「@」で始まる）正規表現にできます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定から継承されます。正規表現マッチングの目的で、float 値は常に「.」の後に 4 桁の小数で表されます。また、大きな数値の場合、10 進数（データベースに保存）と 2 進数（Zabbix サーバーで使用）表現の違いが 4 桁目の小数に影響する場合があります。

foreach 関数の使用に関する詳細および例については、集計計算を参照してください。

可用性による動作

次の表は、ホスト/アイテムおよび履歴データの可用性が制限されている場合に、各関数がどのように動作するかを示しています。

関数	無効なホスト	データがある利用不可ホスト	データがない利用不可ホスト	無効なアイテム	非サポートアイテム
avg_foreach	無視	平均値を返す	無視	無視	無視
bucket_rate_foreach	無視	バケットレートを返す	無視	無視	無視
count_foreach	無視	カウントを返す	0	無視	無視
exists_foreach	無視	1	1	無視	1
last_foreach	無視	最新値を返す	無視	無視	無視
max_foreach	無視	最大値を返す	無視	無視	無視
min_foreach	無視	最小値を返す	無視	無視	無視
sum_foreach	無視	合計値を返す	無視	無視	無視

アイテムが無視された場合、集計には何も加算されません。

2 ビット演算関数

ここに記載されているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
bitand	アイテム値とマスクの「ビットごとの AND」の値。
bitlshift	アイテム値のビットごとの左シフト。
bitnot	アイテム値の「ビットごとの NOT」の値。
bitor	アイテム値とマスクの「ビットごとの OR」の値。
bitrshift	アイテム値のビットごとの右シフト。
bitxor	アイテム値とマスクの「ビットごとの排他的 OR」の値。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- パラメータとして式が受け入れられます
- オプションの関数パラメータ (またはパラメータの一部) は < > で示されます

bitand(value,mask)

アイテム値とマスクの「ビットごとの AND」の値。
 サポートされる値の型: 整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビットごとに行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

例:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3番目または4番目のビットがセットされているか
bitand(last(/host/key),20)=16 #3番目のビットがセットされておらず、5番目のビットがセットされている
```

bitlshift(value,bits to shift)

アイテム値のビット単位の左シフト。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitnot(value)

アイテム値の「ビット単位の NOT」の値。
 サポートされている値のタイプ: 整数。

パラメータ:

- **value** - チェックする値。

比較はビット単位で行われますが、すべての値は 10 進数で指定され、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitor(value,mask)

アイテム値とマスクの「ビット単位の OR」の値。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビット単位で行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitrshift(value,bits to shift)

アイテム値のビット単位での右シフト。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は 100 ではなく 4 と比較します。

bitxor(value,mask)

アイテム値とマスクの「ビットごとの排他的論理和」の値。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビットごとに行われますが、すべての値は 10 進数で指定され、返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

サポートされているすべての関数を参照してください。

3 日付と時刻の関数

ここにリストされているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

Attention:
日付と時刻の関数は、それ自体だけで式内で使用することはできません。ホストアイテムを参照する他のグループの関数を少なくとも 1 つ式に含める必要があります (nodata()関数を除く)。日付と時刻の関数が式内でどのように動作するかの詳細については、計算タイミングを参照してください。

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
date	現在の日付 (YYYYMMDD 形式)。
dayofmonth	1～31 の範囲の月の日。
dayofweek	1～7 の範囲の曜日。
now	エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数。
time	現在の時刻 (HHMMSS 形式)。

関数の詳細

date

現在の日付（YYYYMMDD 形式)。

例：

```
date()<20220101
```

dayofmonth

1 から 31 の範囲の月の日。

例:

```
dayofmonth()=1
```

dayofweek

曜日を 1 から 7 の範囲で指定します（Mon - 1、Sun - 7）。

例（平日のみ）：

`dayofweek()<6`

例（週末のみ）：

`dayofweek()>5`

now

エポック（1970 年 1 月 1 日 00:00:00 UTC）からの秒数。

例：

`now()<1640998800`

time

現在の時刻（HHMMSS 形式）。

例（夜間のみ、00:00-06:00）：

`time()<060000`

サポートされているすべての関数を参照してください。

4 ヒストリ関数

ここに記載されているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると、詳細が表示されます。

関数	説明
change	前回の値と最新の値の差分。
changecount	定義された評価期間内で隣接する値の変化回数。
count	定義された評価期間内の値の数。
countunique	定義された評価期間内のユニークな値の数。
find	定義された評価期間内で値の一致を検索。
first	定義された評価期間内の最初 (最も古い) の値。
firstclock	定義された評価期間内の最初 (最も古い) の値のタイムスタンプ。
fuzzytime	パッシブエージェントの時刻が Zabbix サーバー/プロキシの時刻とどれだけ異なるかを確認。
last	最新の値。
lastclock	定義された評価期間内の N 番目に新しい値のタイムスタンプ。
logeventid	最後のログエントリのイベント ID が正規表現に一致するかどうかを確認。
logseverity	最後のログエントリのログ重大度。
logsource	最後のログエントリのログソースが正規表現に一致するかどうかを確認。
logtimestamp	N 番目に新しいログアイテム値のログメッセージタイムスタンプ。
monodec	値が単調減少しているかどうかを確認。
monoinc	値が単調増加しているかどうかを確認。
nodata	データが受信されていないかどうかを確認。
percentile	期間の P パーセンタイル。P(パーセンテージ) は 3 番目のパラメータで指定。
rate	定義された期間内で単調増加カウンタの 1 秒あたりの平均増加率。

共通パラメータ

- `/host/key` は、ホストアイテムの履歴を参照する関数の共通の必須最初のパラメータです
- `(sec|#num)<:time shift>` は、ホストアイテムの履歴を参照する関数の共通の 2 番目のパラメータで、以下の通りです:
  - **sec** - 秒単位の最大評価期間（時間のサフィックスが使用可能）、または
  - **#num** - 最新の収集値での最大評価範囲（ハッシュマークが先頭にある場合）
  - **time shift**（オプション）は、評価ポイントを過去に移動することができます。タイムシフトの指定方法については詳細をご覧ください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示します
- 関数固有のパラメータは各関数で説明します
- /host/key および (sec|#num)<:time shift> パラメータは決して引用符で囲まないでください

change(/host/key)

前回値と最新値の差分の量。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。
 文字列の場合は、0 - 値が等しい; 1 - 値が異なる。

パラメータ: 共通パラメータを参照。

コメント:

- 数値の差分は、以下の受信値の例 (' 前回値' と ' 最新値' = 差分) のように計算されます:
'1' と '5' = +4
'3' と '1' = −2
'0' と '-2.5' = −2.5。
- 比較にはabsも参照してください。

例:

```
change(/host/key)>10
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

定義された評価期間内の隣接する値間の変化数。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォートで囲む必要あり) - 可能な値: *all* - すべての変化をカウント (デフォルト); *dec* - 減少のみカウント; *inc* - 増加のみカウント

数値以外の値の型の場合、*mode* パラメータは無視されます。

例:

```
changecount(/host/key,1w) # 現在までの直近 1 週間の値の変化数
changecount(/host/key,#10,"inc") # 直近 10 個の値の中で (隣接する値に対して) 増加した回数
changecount(/host/key,24h,"dec") # 現在までの直近 24 時間の間に (隣接する値に対して) 減少した回数
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内の値の数。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照
- **operator** (ダブルクォートで囲む必要あり) サポートされている operator:
eq - 等しい (integer, float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string, text, log のデフォルト) - パターンを含む場合に一致 (大文字小文字を区別)
bitand - ビットごとの AND
regexp - pattern で指定された正規表現に大文字小文字を区別して一致
iregexp - pattern で指定された正規表現に大文字小文字を区別せず一致
- **pattern** - 必要なパターン (文字列引数はダブルクォートで囲む必要あり)

コメント:

- Float アイテムは 2.22e-16 の精度で一致します。
- *like* は integer 値の operator としてはサポートされていません。
- *like* および *bitand* は float 値の operator としてはサポートされていません。
- string, text, log 値では *eq*, *ne*, *like*, *regexp*, *iregexp* のみがサポートされています。
- operator に *bitand* を指定した場合、4 番目の pattern パラメータは「/」で区切った 2 つの数値、**number_to_compare_with/mask** として指定できます。count() は値と *mask* の「ビットごとの AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビットごとの AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます（「/」なし）。
- operator に *regexp* または *iregexp* を指定した場合、4 番目の pattern パラメータは通常の正規表現またはグローバル（「@」で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定を継承します。regexp マッチングの目的で、float 値は常に「.」の後に 4 桁の小数で表現されます。また、大きな数値の場合、10 進数（データベースに格納）と 2 進数（Zabbix サーバーで使用）の表現の違いにより、4 桁目の小数点以下の値が影響を受ける場合があります。

例:

```
count(/host/key,10m) # 現在までの直近 10 分間の値
count(/host/key,10m,"like","error") # 現在までの直近 10 分間で「error」を含む値の数
count(/host/key,10m,,12) # 現在までの直近 10 分間で「12」と等しい値の数
count(/host/key,10m,"gt",12) # 現在までの直近 10 分間で「12」より大きい値の数
count(/host/key,#10,"gt",12) # 現在までの直近 10 個の値のうち「12」より大きい値の数
count(/host/key,10m:now-1d,"gt",12) # 現在から 24 時間前の 10 分間から 24 時間前までの間で「12」より大きい値の数
count(/host/key,10m,"bitand","6/7") # 現在までの直近 10 分間で下位 3 ビットが「110」(2 進数) の値の数
count(/host/key,10m:now-1d) # 現在から 24 時間前の 10 分間から 24 時間前までの間の値の数
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内のユニークな値の数。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照
- **operator** (ダブルクォーテーションで囲む必要があります)。サポートされている operator:
eq - 等しい (integer, float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string, text, log のデフォルト) - パターンを含む場合に一致 (大文字小文字を区別)
bitand - ビットごとの AND
regexp - pattern で指定された正規表現に大文字小文字を区別して一致
iregexp - pattern で指定された正規表現に大文字小文字を区別せずに一致
- **pattern** - 必要なパターン (文字列引数はダブルクォーテーションで囲む必要があります)

コメント:

- Float アイテムは 2.22e-16 の精度で一致します。
- integer 値の operator として *like* はサポートされていません。
- float 値の operator として *like* および *bitand* はサポートされていません。
- string, text, log 値では *eq*, *ne*, *like*, *regexp*, *iregexp* のみがサポートされています。
- operator に *bitand* を指定した場合、4 番目の pattern パラメータは「**number_to_compare_with/mask**」のように「/」で区切った 2 つの数値で指定できます。countunique() は値と *mask* の「ビットごとの AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビットごとの AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます (「/」なし)。
- operator に *regexp* または *iregexp* を指定した場合、4 番目の pattern パラメータは通常の正規表現またはグローバル (「@」で始まる) 正規表現を指定できます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定を継承します。regexp マッチングの目的で、float 値は常に「.」の後に 4 桁の小数で表現されます。また、大きな数値の場合、10 進数 (データベースに格納) と 2 進数 (Zabbix サーバーで使用) の表現の違いにより、4 番目の小数点以下の桁に影響する場合があります。

例:

```
countunique(/host/key,10m) # 直近 10 分間のユニークな値の数
countunique(/host/key,10m,"like","error") # 直近 10 分間で「error」を含むユニークな値の数
countunique(/host/key,10m,,12) # 直近 10 分間で「12」と等しいユニークな値の数
countunique(/host/key,10m,"gt",12) # 直近 10 分間で「12」より大きいユニークな値の数
countunique(/host/key,#10,"gt",12) # 直近 10 個の値のうち「12」より大きいユニークな値の数
countunique(/host/key,10m:now-1d,"gt",12) #24 時間前から 24 時間前 +10 分までの間で「12」より大きいユニークな値の数
countunique(/host/key,10m,"bitand","6/7") # 直近 10 分間で下位 3 ビットが「110」(2 進数) のユニークな値の数
countunique(/host/key,10m:now-1d) #24 時間前から 24 時間前 +10 分までの間のユニークな値の数
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内で値の一致を検索します。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。
 戻り値: 1 - 一致あり; 0 - それ以外。

パラメータ:

- 共通パラメータを参照
- **sec** または **#num** (省略可能) - 指定しない場合は最新値がデフォルト
- **operator** (ダブルクォートで囲む必要あり) サポートされている operator:
eq - 等しい (integer, float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string, text, log のデフォルト) - pattern で指定した文字列を含む場合に一致 (大文字小文字を区別)
bitand - ビット AND
regexp - pattern で指定した正規表現に大文字小文字を区別して一致
iregexp - pattern で指定した正規表現に大文字小文字を区別せず一致
- **pattern** - 必要なパターン (文字列引数はダブルクォートで囲む必要あり); operator が *regexp* または *iregexp* の場合はPerl 互換正規表現 (PCRE) の正規表現

コメント:

- 複数の値が処理される場合、一致する値が 1 つでもあれば'1' を返します。
- integer 値では *like* は operator としてサポートされていません。
- float 値では *like* および *bitand* は operator としてサポートされていません。
- string, text, log 値では *eq*, *ne*, *like*, *regexp*, *iregexp* のみが operator としてサポートされます。
- operator に *regexp* または *iregexp* を指定した場合、4 番目の `pattern` パラメータには通常の正規表現またはグローバル (先頭が'@') 正規表現を指定できます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定を継承します。

例:

```
find(/host/key,10m,"like","error") # 直近 10 分間で'error'を含む値を検索
```

first(/host/key,sec<:time shift>)

定義された評価期間内の最初 (最も古い) の値。
 サポートされている値のタイプ: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照

last()も参照。

例:

```
first(/host/key,1h) # 現在までの直近 1 時間で最も古い値を取得
```

firstclock(/host/key,sec<:time shift>)

定義された評価期間内で最も古い値のタイムスタンプ。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照。

指定した期間にデータが収集されていない場合、関数はエラーで失敗します。

lastclock()も参照してください。

例:

```
firstclock(/host/key,1h) #直近1時間で最も古い値のタイムスタンプを取得
firstclock(/host/key,1h:now-24h) #1日前の1時間で最も古い値のタイムスタンプを取得
```

fuzzytime(/host/key,sec)

パッシブエージェントの時刻が Zabbix サーバー/プロキシの時刻とどれだけ異なるかをチェックします。
 サポートされる値の型: *Float*, *Integer*。
 返り値: 1 - パッシブアイテムの値 (タイムスタンプとして) と Zabbix サーバー/プロキシのタイムスタンプ (値収集時のクロック) の差が *sec* 秒以下の場合; 0 - それ以外の場合。

パラメータ:

- 共通パラメータを参照してください。

コメント:

- 通常、'system.localtime' アイテムと組み合わせて、ローカル時刻が Zabbix サーバーのローカル時刻と同期しているかをチェックするために使用します。注意'system.localtime' は Zabbix エージェントではパッシブチェックとして設定する必要があります。Zabbix agent 2 ではアクティブチェックとして設定することもできます。
- ファイルが長期間更新されていないことをチェックするために、`vfs.file.time[/path/file,modify]` キーと組み合わせて使用することもできます。
- この関数は複雑なトリガー式 (複数のアイテムが関与するもの) での使用は推奨されません。予期しない結果を引き起こす可能性があるためです (時刻差は最新のメトリックで測定されます)。例: `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`。

例:

```
fuzzytime(/host/key,60s)=0 # 時刻差が 60 秒を超えた場合に問題を検出<br><br>
```

last(/host/key,<#num<:time shift»)

最新の値。
 サポートされている値のタイプ: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照

- **#num** (オプション) - N 番目に新しい値

コメント:

- ハッシュタグ付きの期間(#N)は、他の多くの関数とは異なる動作をすることに注意してください。例えば: `last(/host/key)` は常に `last(/host/key,#1)` と等しい; `last(/host/key,#3)` は 3 番目に新しい値（最新の *3* つの値ではない）。
- 1 秒以内に履歴に 2 つ以上の値が存在する場合、Zabbix は値の正確な順序を保証しません。
- first()も参照してください。

例:

```
last(/host/key) # 最新の値を取得
last(/host/key,#2) # 前回の値を取得
last(/host/key,#1) <> last(/host/key,#2) # 最新の値と前回の値が異なる
```

lastclock(/host/key,<#num<:time shift»)

定義された評価期間内で N 番目に新しい値のタイムスタンプ。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照;
- **#num** (オプション) - N 番目に新しい値。

指定された期間にデータが収集されていない場合や、N 番目の値が収集されていない場合は、エラーで失敗します。

firstclock()も参照してください。

例:

```
lastclock(/host/key) #最新値のタイムスタンプを取得
lastclock(/host/key,#2) #2番目に新しい値のタイムスタンプを取得
lastclock(/host/key,#3:now-1d) #1日前の3番目に新しい値のタイムスタンプを取得
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

最後のログエントリのイベント ID が正規表現に一致するかどうかを確認します。
 サポートされている値の型: *Log*。
 戻り値: 0 - 一致しない; 1 - 一致する。

パラメータ:

- 共通パラメータを参照
- **#num** (オプション) - 最新から N 番目の値
- **pattern** (オプション) - 必要なパターンを記述する正規表現、Perl 互換正規表現 (PCRE) スタイル（文字列引数はダブルクォーテーションで囲む必要があります）

logseverity(/host/key,<#num<:time shift»)

最後のログエントリのログ重大度。
 サポートされている値の型: *Log*。
 返り値: 0 - デフォルトの重大度; N - 重大度 (整数、Windows イベントログに便利: 1 - Information, 2 - Warning, 4 - Error, 7 - Failure Audit, 8 - Success Audit, 9 - Critical, 10 - Verbose)。

パラメータ:

- 共通パラメータを参照
- **#num** (省略可能) - 最新から N 番目の値

Zabbix は Windows イベントログの **Information** フィールドからログ重大度を取得します。

logsource(/host/key,<#num<:time shift»,<pattern>)

最後のログエントリのログソースが正規表現に一致するかどうかを確認します。
 サポートされている値の型: *Log*。
 戻り値: 0 - 一致しない; 1 - 一致する。

パラメータ:

- 共通パラメータを参照
- **#num** (省略可能) - 最新から N 番目の値
- **pattern** (省略可能) - 必要なパターンを記述する正規表現、Perl Compatible Regular Expression (PCRE) スタイル（文字列引数はダブルクォートで囲む必要があります）

通常、Windows イベントログで使用されます。

例:

```
logsource(/host/key,,"VMware Server")
```

logtimestamp(/host/key,<#num<:time shift»)

N 番目に新しいログアイテム値のログメッセージのタイムスタンプ。
 サポートされている値のタイプ: *Log*。

パラメータ:

- 共通パラメータを参照;
- **#num** (オプション) - N 番目に新しい値。

コメント:

- タイムシフトの計算は、ログメッセージのタイムスタンプではなく、アイテム値のクロックに基づいています。
- 次の場合、関数はエラーで失敗します:
  - ログタイプ以外のアイテムが受信された場合;
  - 指定された期間にデータが収集されていない場合;
  - N 番目の値が収集されていない場合;
  - ログメッセージにタイムスタンプ値が含まれていない場合。

例:

```
logtimestamp(/host/key) #最新のログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#2) #2番目に新しいログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#3:now-1d) #1日前の3番目に新しいログメッセージのタイムスタンプを取得
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調減少しているかどうかを確認します。
 サポートされている値の型: 整数。
 戻り値: 1 - 期間内のすべての要素が連続して減少している場合; 0 - それ以外の場合。

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォートで囲む必要あり) - *weak* (各値が前の値以下; デフォルト) または *strict* (各値が減少している)

例:

```
monodec(/Host1/system.swap.size[all,free],60s) + monodec(/Host2/system.swap.size[all,free],60s) + monodec(
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調増加しているかどうかをチェックします。
 サポートされている値の型: 整数。
 戻り値: 1 - 期間内のすべての要素が連続して増加している場合; 0 - それ以外の場合。

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォーテーションで囲む必要あり) - *weak* (すべての値が前の値以上; デフォルト) または *strict* (すべての値が増加している)

例:

```
monoinc(/Host1/system.localtime,#3,"strict")=0 # システムのローカルタイムが一貫して増加しているかどうかをチェック
```

nodata(/host/key,sec,<mode>)

データが受信されていないことをチェックします。
 サポートされている値の型: 整数、浮動小数点、文字列、テキスト、ログ。
 戻り値: 指定した期間内にデータが受信されなかった場合は 1、それ以外は 0。

パラメータ:

- 共通パラメータを参照
- **sec** - この関数はヒストリーシンカーが 30 秒ごとにのみ計算するため、期間は 30 秒未満にしないでください。`nodata(/host/key,0)` は許可されていません
- **mode** - *strict*（ダブルクォーテーションで囲む）を指定すると、プロキシの可用性に影響されなくなります（詳細はコメントを参照）

コメント:

- プロキシで監視されている'nodata' トリガーは、デフォルトでプロキシの可用性に影響されます。プロキシが利用できなくなった場合、'nodata' トリガーは接続が復旧した直後には発火せず、遅延期間のデータをスキップします。パッシブプロキシの場合、接続が 2 秒以上 15 秒以内に復旧した場合に抑制が有効になります。アクティブプロキシの場合、接続が 15 秒以上遅れて復旧した場合に抑制が有効になります。プロキシの可用性に影響されないようにするには、3 番目のパラメータを使用します。例: `nodata(/host/key,5m,"strict")`。この場合、評価期間（5 分）データがなければすぐに関数が発火します。

- この関数は、1 番目のパラメータで指定した期間内に:
- データがなく、Zabbix サーバーが再起動された
- データがなく、メンテナンスが完了した
- データがなく、アイテムが追加または再有効化された
 場合にエラーを表示します。
- エラーはトリガー設定の情報列に表示されます。
- Zabbix サーバー、プロキシ、エージェント間で時刻のずれがある場合、この関数は正しく動作しない場合があります。時刻同期の要件も参照してください。
- nodata() 関数は、それ単体では式で使用できません。別のグループの関数を少なくとも 1 つ、ホストアイテムを参照して式に含める必要があります（日付と時刻の関数を除く）。nodata() 関数が式内でどのように動作するかの詳細は、計算タイミングを参照してください。

percentile(/host/key,(sec|#num)<:time shift>,percentage)

期間の P パーセンタイル。P（パーセンテージ）は 3 番目のパラメータで指定します。
 サポートされている値の型：*Float*、*Integer*。

パラメータ:

- 共通パラメータを参照
- **percentage** - 小数点以下 4 桁までの 0 から 100（両端を含む）の浮動小数点数

rate(/host/key,sec<:time shift>)

指定した期間内で単調増加するカウンターの 1 秒あたりの平均増加率。
 サポートされている値の型: *Float*, *Integer*。

パラメータ:

- 共通パラメータを参照

PromQL の'rate' に機能的に対応。

例:

```
rate(/host/key,30s) #30 秒間で単調増加が 20 の場合、この関数は 0.67 を返します。
```

サポートされているすべての関数を参照してください。

5 トレンド関数

トレンド関数は、ヒストリ関数とは対照的に、計算にトレンドデータを使用します。

トレンドは 1 時間ごとの集計値を保存します。トレンド関数はこれらの 1 時間ごとの平均値を使用するため、長期的な分析に役立ちます。

トレンド関数の結果はキャッシュされるため、同じパラメータで同じ関数を複数回呼び出しても、データベースからの情報取得は 1 回だけです。トレンド関数のキャッシュは、TrendFunctionCacheSizeサーバーパラメータで制御されます。

トレンド関数のみを参照するトリガーは、式内の最小の時間間隔ごとに 1 回評価されます。たとえば、次のようなトリガーの場合

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

は 1 日ごとに評価されます。トリガーにトレンド関数とヒストリ関数（または日付と時刻、nodata()）の両方が含まれている場合は、通常の原則に従って計算されます。

ここに記載されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
baselinedev	直近のデータ期間と前のシーズンの同じデータ期間との間の偏差数（stddevpop アルゴリズムによる）を返します。
baselinewma	加重移動平均アルゴリズムを使用して、複数の等しい期間（「シーズン」）の同じ時間枠のデータを平均化することでベースラインを計算します。
trendavg	指定した期間内のトレンド値の平均。
trendcount	指定した期間内のトレンド値の計算に使用された正常に取得されたヒストリ値の数。
trendmax	指定した期間内のトレンド値の最大値。
trendmin	指定した期間内のトレンド値の最小値。
trendstl	検出期間中の異常値の割合を返します。これは 0 から 1 の間の小数値で、((異常値の数)/(値の総数)) です。

関数	説明
trendsum	指定した期間内のトレンド値の合計。

共通パラメータ

- /host/key は共通の必須の最初のパラメータです
- time period:time shift は共通の 2 番目のパラメータで、以下のようになります:
  - **time period** - 期間（最小値は'1h'）、<N><time unit> で定義されます。N は時間単位の数、time unit は h（時間）、d（日）、w（週）、M（月）、y（年）です。
  - **time shift** - 期間のオフセット（関数の例を参照）

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数で説明されています
- /host/key および time period:time shift パラメータは決して引用符で囲まないでください

baselinedev(/host/key,data period:time shift,season unit,num seasons)

直近のデータ期間と、過去のシーズンの同じデータ期間との間の偏差の数 (stddevpop アルゴリズムによる) を返します。

パラメータ:

- 共通パラメータを参照;
- **data period** - シーズン内のデータ収集期間。<N><time unit> で定義されます。
N - 時間単位の数
time unit - h (時間), d (日), w (週), M (月), y (年)。season 以下でなければなりません。
- **season unit** - 1 シーズンの期間 (h, d, w, M, y)。data period より小さくできません。
- **num seasons** - 評価するシーズン数。

例:

```
baselinedev(/host/key,1d:now/d,"M",6) #直近1日と、過去6ヶ月の同じ日との標準偏差(母集団)の数を計算します。
baselinedev(/host/key,1h:now/h,"d",10)  #直近1時間と、昨日までの10日間の同じ時間との標準偏差(母集団)の数を
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

重み付き移動平均アルゴリズムを使用して、複数の等しい期間（「シーズン」）の同じ時間枠からデータを平均化することでベースラインを計算します。

パラメータ:

- 共通パラメータを参照;
- **data period** - シーズン内でのデータ収集期間、<N><time unit> として定義されます。
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）、M（月）、y（年）、シーズン以下でなければなりません
Time shift - 期間のオフセット、シーズン内でのデータ収集期間の終了を定義します（例を参照）;
- **season unit** - 1 シーズンの期間（h, d, w, M, y）、data period より小さくできません;
- **num seasons** - 評価するシーズンの数。

例:

```
baselinewma(/host/key,1h:now/h,"d",3) #昨日で終了した3日間の期間内の最後の1時間に基づいてベースラインを計
baselinewma(/host/key,2h:now/h,"d",3) #昨日で終了した3日間の期間内の最後の2時間に基づいてベースラインを計
baselinewma(/host/key,1d:now/d,"M",4) #直近の完全な月の前の4か月間で「昨日」と同じ月の日に基づいてベースラ
```

trendavg(/host/key,time period:time shift)

指定した期間内のトレンド値の平均。

パラメータ:

- 共通パラメータを参照。

例:

```
trendavg(/host/key,1h:now/h) #直前1時間の平均（例: 12:00-13:00）
trendavg(/host/key,1h:now/h-1h) #2時間前の平均（11:00-12:00）
trendavg(/host/key,1h:now/h-2h) #3時間前の平均（10:00-11:00）
trendavg(/host/key,1M:now/M-1y) #1年前の直前1ヶ月の平均
```

trendcount(/host/key,time period:time shift)

定義された期間内にトレンド値を計算するために使用された、正常に取得された履歴値の数。

パラメータ:

- 共通パラメータを参照。

例:

```
trendcount(/host/key,1h:now/h) #直前の1時間の値の数（例：12:00-13:00）
trendcount(/host/key,1h:now/h-1h) #2時間前の値の数（11:00-12:00）
trendcount(/host/key,1h:now/h-2h) #3時間前の値の数（10:00-11:00）
trendcount(/host/key,1M:now/M-1y) #1年前の前月の値の数
```

trendmax(/host/key,time period:time shift)

定義された期間内のトレンド値の最大値。

パラメータ:

- 共通パラメータを参照。

例:

```
trendmax(/host/key,1h:now/h) #直前の1時間の最大値（例：12:00-13:00）
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前の1時間（12:00-13:00）の最大値と最小値の
trendmax(/host/key,1h:now/h-1h) #2時間前の最大値（11:00-12:00）
trendmax(/host/key,1h:now/h-2h) #3時間前の最大値（10:00-11:00）
trendmax(/host/key,1M:now/M-1y) #1年前の直前月の最大値
```

trendmin(/host/key,time period:time shift)

定義された期間内のトレンド値の最小値。

パラメータ:

- 共通パラメータを参照。

例:

```
trendmin(/host/key,1h:now/h) #直前1時間の最小値(例: 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前1時間(12:00-13:00)の最大値と最小値の差分
trendmin(/host/key,1h:now/h-1h) #2時間前(11:00-12:00)の最小値
trendmin(/host/key,1h:now/h-2h) #3時間前(10:00-11:00)の最小値
trendmin(/host/key,1M:now/M-1y) #1年前の直前月の最小値
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

検出期間中の異常値の割合を返します。値は 0 から 1 の間の小数で、((異常値の数)/(値の総数)) です。

パラメータ:

- 共通パラメータを参照;
- **eval period** - 分解する必要がある期間（最小'1h'）、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間), d (日), w (週), M (月), y (年)
- **detection period** - 異常値を計算するための eval period の終了前の期間（最小'1h'、eval period より長くできません）、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間), d (日), w (週)
- **season** - 繰り返しパターン（"season"）が予想される最短期間（最小'2h'、eval period より長くできません。eval period 内のエントリ数は、得られる頻度（season/h）の 2 倍より大きくなければなりません）、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間), d (日), w (週)
- **deviations** - 異常値と見なす偏差（devalg で計算、少数可、1 以上、デフォルトは 3）
- **devalg**（ダブルクォート必須）- 偏差アルゴリズム。*stddevpop*、*stddevsamp*、*mad*（デフォルト）から選択
- **s window** - 季節性抽出のための loess ウィンドウのスパン（ラグ数、デフォルトは"10 * eval period 内のエントリ数 + 1"）

例:

```
trendstl(/host/key,100h:now/h,10h,2h) #直近100時間のトレンドデータを分析し、その期間の直近10時間の異常率を
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #10時間前までの100時間のトレンドデータを分析し、その
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #1日前までの100日間のトレンドデータを分析し、その期間の直近
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #1年前の前月を分析し、その期間の最終日の異常率を求める
```

trendsum(/host/key,time period:time shift)

指定した期間内のトレンド値の合計。

パラメータ:

- 共通パラメータを参照。

例:

```
trendsum(/host/key,1h:now/h) #直前1時間の合計（例: 12:00-13:00）
trendsum(/host/key,1h:now/h-1h) #2時間前の合計（11:00-12:00）
trendsum(/host/key,1h:now/h-2h) #3時間前の合計（10:00-11:00）
trendsum(/host/key,1M:now/M-1y) #1年前の直前月の合計
```

サポートされているすべての関数を参照してください。

6 数学関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテム
- 式マクロ

特に記載がない限り、数学関数は float 型および integer 型の値タイプでサポートされています。

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
abs	値の絶対値。
acos	値のアークコサイン（ラジアンで表される角度）。
asin	値のアークサイン（ラジアンで表される角度）。
atan	値のアークタンジェント（ラジアンで表される角度）。
atan2	指定された座標（値）と横座標のアークタンジェント（ラジアンで表される角度）。
avg	参照されたアイテム値の平均値。
cbrt	値の立方根。
ceil	値を切り上げて最も近い大きいまたは等しい整数に丸める。
cos	値のコサイン（値はラジアンで表される角度）。
cosh	値の双曲線コサイン。
cot	値のコタンジェント（値はラジアンで表される角度）。
degrees	値をラジアンから度に変換。
e	オイラー数（2.718281828459045）。
exp	値のオイラー数のべき乗。
expm1	値のオイラー数のべき乗から 1 を引いた値。
floor	値を切り捨てて最も近い小さいまたは等しい整数に丸める。
log	自然対数。
log10	常用対数。
max	参照されたアイテム値の最大値。
min	参照されたアイテム値の最小値。
mod	除算の余り。
pi	円周率（3.14159265358979）。
power	値のべき乗。
radians	値を度からラジアンに変換。
rand	ランダムな整数値を返す。
round	値を小数点以下で丸める。
signum	値が負の場合は'-1'、ゼロの場合は'0'、正の場合は'1' を返す。
sin	値のサイン（値はラジアンで表される角度）。
sinh	値の双曲線サイン（値はラジアンで表される角度）。
sqrt	値の平方根。
sum	参照されたアイテム値の合計。
tan	値のタンジェント。
truncate	値を小数点以下で切り捨てる。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- パラメータとして式が受け入れられます

- オプションの関数パラメータ (またはパラメータの一部) は < > で示されます

abs(value)

値の絶対値（0 からの距離）を返します。

パラメータ:

- **value** - チェックする値

例えば、'3' または'-3' の絶対値は'3' になります。

例:

```
abs(last(/host/key))>10
```

acos(value)

値の逆余弦をラジアンで返します。

パラメータ:

- **value** - チェックする値

値は-1 から 1 の間でなければなりません。例えば、値'0.5' の逆余弦は'2.0943951' になります。

例:

```
acos(last(/host/key))
```

asin(value)

値の逆正弦をラジアンで表した角度。

パラメータ:

- **value** - チェックする値

値は-1 から 1 の間でなければなりません。例えば、値'0.5' の逆正弦は'-0.523598776' になります。

例:

```
asin(last(/host/key))
```

atan(value)

値のアークタンジェント（逆正接）をラジアンで返します。

パラメータ:

- **value** - チェックする値

値は-1 から 1 の間でなければなりません。例えば、値'1' のアークタンジェントは'0.785398163' になります。

例:

```
atan(last(/host/key))
```

atan2(value,abscissa)

指定された縦座標（value）と横座標（abscissa）のアークタンジェントをラジアンで表した角度を返します。

パラメータ:

- **value** - チェックする値
- **abscissa** - 横座標の値

例えば、値「1」の縦座標と横座標のアークタンジェントは「2.21429744」になります。

例:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

参照されたアイテム値の平均値。

パラメータ:

- **valueX** - アイテムの履歴を扱う他の関数によって返された値。

例:

```
avg(avg(/host/key),avg(/host2/key2))
```

cbrt(value)

値の立方根。

パラメータ:

- **value** - チェックする値

例えば、'64' の立方根は'4'、'63' の立方根は'3.97905721' となります。

例:

```
cbrt(last(/host/key))
```

ceil(value)

値を切り上げて、最も近い大きいまたは等しい整数にします。

パラメータ:

- **value** - チェックする値

例えば、'2.4' は'3' に切り上げられます。floor()も参照してください。

例:

```
ceil(last(/host/key))
```

cos(value)

値のコサイン。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' のコサインは'0.54030230586' になります。

例:

```
cos(last(/host/key))
```

cosh(value)

値の双曲線余弦。値を科学的記法ではなく実数として返します。

パラメータ:

- **value** - チェックする値

例えば、値'1' の双曲線余弦は'1.54308063482' になります。

例:

```
cosh(last(/host/key))
```

cot(value)

値の余接。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' の余接は'0.54030230586' になります。

例:

```
cot(last(/host/key))
```

degrees(value)

値をラジアンから度に変換します。

パラメータ:

- **value** - チェックする値

例えば、値'1' を度に変換すると'57.2957795' になります。

例:

```
degrees(last(/host/key))
```

e

オイラー数 (2.718281828459045)。

例:

```
e()
```

exp(value)

値のべき乗のネイピア数。

パラメータ:

- **value** - チェックする値

例えば、値'2' のべき乗のネイピア数は'7.38905609893065' になります。

例:

```
exp(last(/host/key))
```

expm1(value)

値の累乗のオイラー数から 1 を引いた値。

パラメータ:

- **value** - チェックする値

例えば、値'2' の累乗のオイラー数から 1 を引いた値は'6.38905609893065' となります。

例:

```
expm1(last(/host/key))
```

floor(value)

値を最も近い小さいまたは等しい整数に切り捨てます。

パラメータ:

- **value** - チェックする値

例えば、'2.6' は'2' に切り捨てられます。ceil()も参照してください。

例:

```
floor(last(/host/key))
```

log(value)

自然対数。

パラメータ:

- **value** - チェックする値

例えば、値'2' の自然対数は'0.69314718055994529' となります。

例:

```
log(last(/host/key))
```

log10(value)

常用対数（底 10 の対数）を返します。

パラメータ:

- **value** - チェックする値

例えば、値'5' の常用対数は'0.69897000433' になります。

例:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

参照されたアイテム値の中で最も高い値。

パラメータ:

- **valueX** - アイテムの履歴を扱う他の関数によって返された値

例:

```
max(avg(/host/key),avg(/host2/key2))
```

min(<value1>,<value2>,...)

参照されたアイテム値のうち最小の値。

パラメータ:

- **valueX** - アイテム履歴を扱う他の関数によって返された値

例:

```
min(avg(/host/key),avg(/host2/key2))
```

mod(value,denominator)

除算の余り。

パラメータ:

- **value** - チェックする値
- **denominator** - 除算の分母

例えば、値'5' を分母'2' で割った余りは'1' になります。

例:

```
mod(last(/host/key),2)
```

pi

円周率（3.14159265358979）。

例：

```
pi()
```

power(value,power value)

値のべき乗。

パラメータ:

- **value** - チェックする値
- **power value** - 使用する N 乗

例えば、値'2' の 3 乗は'8' になります。

例:

```
power(last(/host/key),3)
```

radians(value)

値を度からラジアンに変換します。

パラメータ:

- **value** - チェックする値

例えば、値「1」をラジアンに変換すると「0.0174532925」になります。

例:

```
radians(last(/host/key))
```

rand

ランダムな整数値を返します。時刻をシードにした擬似乱数を生成します (数学的な目的には十分ですが、暗号化には適していません)。

例:

`rand()`

round(value,decimal places)

値を小数点以下の桁数に丸めます。

パラメータ:

- **value** - チェックする値
- **decimal places** - 丸める小数点以下の桁数を指定します（0 も可能です）。

例えば、値'2.5482' を小数点以下 2 桁に丸めると'2.55' になります。

例:

`round(last(/host/key),2)`

signum(value)

値が負の場合は「-1」、値がゼロの場合は「0」、値が正の場合は「1」を返します。

パラメータ:

- **value** - チェックする値

例:

`signum(last(/host/key))`

sin(value)

値のサイン。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値「1」のサインは「0.8414709848」になります。

例:

`sin(last(/host/key))`

sinh(value)

値の双曲線正弦を返します。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' の双曲線正弦は'1.17520119364' になります。

例:

`sinh(last(/host/key))`

sqrt(value)

値の平方根。
 この関数は負の値では失敗します。

パラメータ:

- **value** - チェックする値

例えば、値'3.5' の平方根は'1.87082869339' になります。

例:

`sqrt(last(/host/key))`

sum(<value1>,<value2>,...)

参照されたアイテム値の合計。

パラメータ:

- **valueX** - アイテム履歴を扱う他の関数によって返される値。

例:

`sum(avg(/host/key),avg(/host2/key2))`

tan(value)

値のタンジェント。

パラメータ:

- **value** - チェックする値

例えば、値'1' のタンジェントは'1.55740772465' になります。

例:

```
tan(last(/host/key))
```

truncate(value,decimal places)

値を小数点以下で切り捨てます。

パラメータ:

- **value** - チェックする値
- **decimal places** - 切り捨てる小数点以下の桁数を指定します（0 も可能です）

たとえば、値'2.5482' を小数点以下 2 桁で切り捨てると'2.54' になります。

例:

```
truncate(last(/host/key),2)
```

サポートされているすべての関数を参照してください。

7 演算子関数

ここにリストされているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
between	値が指定した範囲に属しているかどうかをチェックします。
in	値がリストされた値の少なくとも 1 つと等しいかどうかをチェックします。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- パラメータとして式を受け付けます

between(value,min,max)

値が指定された範囲に属しているかどうかを確認します。
 サポートされている値の型: *Integer*, *Float*。
 戻り値: 1 - 範囲内; 0 - それ以外。

パラメータ:

- **value** - チェックする値;
- **min** - 最小値;
- **max** - 最大値。

例:

```
between(last(/host/key),1,10)=1 #値が1から10の間の場合にトリガー
```

in(value,value1,value2,...valueN)

値がリストされた値の少なくとも 1 つと等しいかどうかをチェックします。
 サポートされている値の型: 整数、浮動小数点、文字、テキスト、ログ。
 戻り値: 1 - 等しい場合; 0 - それ以外の場合。

パラメータ:

- **value** - チェックする値;

- **valueX** - リストされた値（文字列値はダブルクォートで囲む必要があります）。

値は、すべての値が数値に変換できる場合は数値として比較されます。それ以外の場合は文字列として比較されます。

例:

```
in(last(/host/key),5,10)=1 #最後の値が5または10の場合にトリガー
in("text",last(/host/key),last(/host/key,#2))=1 #"text"が直近2つの値のいずれかと等しい場合にトリガー
```

サポートされているすべての関数を参照してください。

8 予測関数

ここに記載されているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
forecast	アイテムの将来の値、最大値、最小値、デルタまたは平均値。
timeleft	アイテムが指定したしきい値に達するまでに必要な秒数。

共通パラメータ

- `/host/key` は、ホストアイテムの履歴を参照する関数の共通必須の最初のパラメータです
- `(sec|#num)<:time shift>` は、ホストアイテムの履歴を参照する関数の共通の 2 番目のパラメータです。ここで:
  - **sec** - 最大評価期間（秒単位。時間のサフィックスが使用可能）
  - **#num** - 最新の収集値の最大評価範囲（ハッシュ記号を前置）
  - **time shift**（オプション）は、評価ポイントを過去に移動するために使用します。タイムシフトの指定方法については詳細を参照してください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数で説明されています
- `/host/key` および `(sec|#num)<:time shift>` パラメータは決して引用符で囲まないでください

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

アイテムの将来の値、最大値、最小値、差分または平均値。
 サポートされている値の型: *Float*, *Integer*。

パラメータ:

- 共通パラメータを参照;
- **time** - 予測する期間（秒単位、時間のサフィックスが使用可能）。負の値もサポート;
- **fit** (オプション; ダブルクォート必須) - 過去データにフィットさせる関数。サポートされているフィット:
linear - 線形関数（デフォルト）
polynomialN - N 次多項式（1 <= N <= 6）
exponential - 指数関数
logarithmic - 対数関数
power - 累乗関数
 なお、*polynomial1* は *linear* と同等;
- **mode** (オプション; ダブルクォート必須) - 出力する値。サポートされているモード:
value - 値（デフォルト）
max - 最大値
min - 最小値
delta - *max-min*
avg - 平均値
 なお、*value* は `now` + `time` 時点のアイテム値を推定する。*max*、*min*、*delta*、*avg* は `now` から `now` + `time` までの区間でのアイテム値の推定値を調査する。

コメント:

- 返却値が 1.7976931348623158E+308 より大きい、または-1.7976931348623158E+308 より小さい場合、返却値はそれぞれ 1.7976931348623158E+308 または-1.7976931348623158E+308 に切り詰められる;
- 式の誤用（アイテムタイプの誤り、パラメータの不正）でのみ未サポートとなる。それ以外のエラー時は-1 を返す;
- 予測トリガー関数の追加情報も参照。

例:

```
forecast(/host/key,#10,1h) #直近10個の値から1時間後のアイテム値を予測
forecast(/host/key,1h,30m) #直近1時間のデータから30分後のアイテム値を予測
forecast(/host/key,1h:now-1d,12h) #1日前の1時間分のデータから12時間後のアイテム値を予測
```

```
forecast(/host/key,1h,10m,"exponential") #直近1時間のデータと指数関数で10分後のアイテム値を予測
forecast(/host/key,1h,2h,"polynomial3","max") #直近1時間のデータと3次多項式で今後2時間以内に到達しうる最大
forecast(/host/key,#2,-20m) #直近2個の値から20分前のアイテム値を推定（特にアイテムの更新頻度が低い場合、1
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

アイテムが指定されたしきい値に達するまでに必要な秒数。
 サポートされている値のタイプ: *Float*、*Integer*。

パラメータ:

- 共通パラメータを参照;
- **threshold** - 到達する値（単位サフィックスが使用可能）;
- **fit**（オプション; ダブルクォーテーションで囲む必要あり）- forecast()を参照。

コメント:

- 返す値が 1.7976931348623158E+308 より大きい場合、返り値は 1.7976931348623158E+308 に切り捨てられます;
- しきい値に到達できない場合は 1.7976931348623158E+308 を返します;
- 式で誤用された場合（アイテムタイプの誤り、パラメータの無効など）のみ未サポートとなり、それ以外のエラー時は-1 を返します;
- 予測トリガー関数に関する追加情報も参照してください。

例:

```
timeleft(/host/key,#10,0) #直近10個の値に基づき、アイテム値が0に達するまでの時間
timeleft(/host/key,1h,100) #直近1時間のデータに基づき、アイテム値が100に達するまでの時間
timeleft(/host/key,1h:now-1d,100) #1日前の1時間分のデータに基づき、アイテム値が100に達するまでの時間
timeleft(/host/key,1h,200,"polynomial2") #直近1時間のデータと、アイテムが2次多項式のように振る舞うという仮
```

サポートされているすべての関数を参照してください。

9 文字列関数

ここに記載されているすべての関数は、以下でサポートされています:

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
ascii	値の最左文字の ASCII コード。
bitlength	値のビット長。
bytelength	値のバイト長。
char	値を ASCII コードとして解釈し、文字を返します。
concat	参照されたアイテム値または定数値を連結した結果の文字列。
insert	指定した位置から文字列に指定した文字またはスペースを挿入します。
jsonpath	JSONPath の結果を返します。
left	値の最左文字を返します。
length	値の文字数。
ltrim	文字列の先頭から指定した文字を削除します。
mid	'start' で指定した文字位置から N 文字の部分文字列を返します。
repeat	文字列を繰り返します。
replace	値の中でパターンを検索し、置換文字列で置き換えます。
right	値の最右文字を返します。
rtrim	文字列の末尾から指定した文字を削除します。
trim	文字列の先頭と末尾から指定した文字を削除します。
xmlxpath	XML XPath の結果を返します。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- パラメータとして式を使用できます
- 文字列パラメータはダブルクォーテーションで囲む必要があります。そうしないと誤解される可能性があります

- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます

ascii(value)

値の最左文字の ASCII コード。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値

例えば、'Abc' のような値は'65'（'A' の ASCII コード）を返します。

例:

```
ascii(last(/host/key))
```

bitlength(value)

値のビット長。
 サポートされている値の型: *String*, *Text*, *Log*, *Integer*。

パラメータ:

- **value** - チェックする値

例:

```
bitlength(last(/host/key))
```

bytelength(value)

値のバイト数。
 サポートされている値の型: *String*, *Text*, *Log*, *Integer*。

パラメータ:

- **value** - チェックする値

例:

```
bytelength(last(/host/key))
```

char(value)

値を ASCII コードとして解釈して文字を返します。
 サポートされている値の型：整数。

パラメータ:

- **value** - チェックする値

値は 0-255 の範囲でなければなりません。例えば、「65」という値（ASCII コードとして解釈）は「A」を返します。

例:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

参照されたアイテム値または定数値を連結した結果の文字列。
 サポートされている値の型: *String*, *Text*, *Log*, *Float*, *Integer*。

パラメータ:

- **valueX** - ヒストリ関数のいずれかによって返された値、または定数値 (文字列、整数、または浮動小数点数)。少なくとも 2 つのパラメータを含める必要があります。

たとえば、「Zab」のような値に「bix」(定数文字列) を連結すると、「Zabbix」が返されます。

例:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15]))
```

insert(value,start,length,replacement)

指定した位置から文字列に指定した文字またはスペースを挿入します。
 対応する値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **start** - 開始位置
- **length** - 置換する位置数
- **replacement** - 置換する文字列

例えば、'Zabbbix' という値は、'bb' (開始位置 3、置換する位置数 2) を'b' に置換すると'Zabbix' になります。

例:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

JSONPath の結果を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **path** - パス (引用符で囲む必要があります);
- **default** - JSONPath クエリがデータを返さない場合のオプションのフォールバック値。他のエラーの場合は失敗が返されます (例: "unsupported construct")。

例:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

値の左端の文字を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **count** - 返す文字数。

例えば、返す左端の文字数を 3 に指定することで、'Zabbix' から'Zab' を返すことができます。right()も参照してください。

例:

```
left(last(/host/key),3) #左端の3文字を返す
```

length(value)

値の文字数。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値

例:

```
length(last(/host/key)) #最新値の文字数
length(last(/host/key,#3)) #3番目に新しい値の文字数
length(last(/host/key,#1:now-1d)) #1日前の最新値の文字数
```

ltrim(value,<chars>)

文字列の先頭から指定した文字を削除します。
 対応する値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **chars** (省略可能) - 削除する文字を指定します。

デフォルトでは空白文字が左側から削除されます（省略可能な文字が指定されていない場合）。関連項目: rtrim(), trim()。

例:

```
ltrim(last(/host/key)) #文字列の先頭から空白を削除
ltrim(last(/host/key),"Z") #文字列の先頭から'Z'をすべて削除
ltrim(last(/host/key)," Z") #文字列の先頭から空白と'Z'をすべて削除
```

mid(value,start,length)

'start' で指定された文字位置から N 文字の部分文字列を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **start** - 部分文字列の開始位置;
- **length** - 部分文字列として返す文字数。

例えば、開始位置が 2、返す文字数が 4 の場合、'Zabbix' のような値から'abbi' を返すことができます。

例:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

文字列を繰り返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **count** - 繰り返す回数

例:

```
repeat(last(/host/key),2) #値を2回繰り返す
```

replace(value,pattern,replacement)

値の中でパターンを見つけて、replacement で置き換えます。パターンのすべての出現箇所が置き換えられます。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **pattern** - 検索するパターン;
- **replacement** - パターンを置き換える文字列。

例:

```
replace(last(/host/key),"ibb","abb") # 'ibb'をすべて'abb'に置き換えます
```

right(value,count)

値の右端の文字を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **count** - 返す文字数。

例えば、返す右端の文字数を 3 に指定することで、'Zabbix' から'bix' を返すことができます。left()も参照してください。

例:

```
right(last(/host/key),3) #右端の3文字を返す
```

rtrim(value,<chars>)

文字列の末尾から指定した文字を削除します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **chars** (省略可能) - 削除する文字を指定

デフォルトでは空白が右側からトリムされます (省略可能な文字が指定されていない場合)。関連項目: ltrim(), trim()。

例:

```
rtrim(last(/host/key)) #文字列の末尾から空白を削除
rtrim(last(/host/key),"x") #文字列の末尾から'x'をすべて削除
rtrim(last(/host/key),"x ") #文字列の末尾から'x'と空白をすべて削除
```

trim(value,<chars>)

文字列の先頭と末尾から指定した文字を削除します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **chars** (省略可能) - 削除する文字を指定します。

デフォルトでは、(省略可能な文字が指定されていない場合) 両端の空白がトリミングされます。関連項目: ltrim(), rtrim()。

例:

```
trim(last(/host/key)) #文字列の先頭と末尾の空白を削除
trim(last(/host/key),"_") #文字列の先頭と末尾の'_'を削除
```

xmlxpath(value,path,<default>)

XML XPath の結果を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **path** - パス (引用符で囲む必要があります);
- **default** - XML XPath クエリが空のノードセットを返した場合のオプションのフォールバック値。空の結果がノードセットでない場合 (つまり空文字列) は返されません。他のエラーの場合は失敗が返されます (例: ”invalid expression”)。

例:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

サポートされているすべての関数を参照してください。

6 マクロ

サポートされているマクロとロケーションでサポートされているユーザーマクロをすぐに使用することができます。

1 ロケーションでサポートされているマクロ

概要

場所ごとにサポートされているマクロの表には、特定の場所でサポートされているすべてのマクロ（組み込み、式、ユーザー定義）のリファレンスが含まれています。

関連グループのすべてのマクロがサポートされている場合は、グループ名のみが表示されます。各マクロ名またはグループ名をクリックすると、マクロの詳細が表示されます。

マクロの詳細リストには、アプリケーション領域ごとにグループ化されたすべての組み込みマクロの詳細なリストが含まれています。マクロ値をカスタマイズするには（たとえば、短縮したり特定の部分文字列を抽出したりする場合）、マクロ関数を使用できます。

ロケーションでサポートされるマクロ

ロケーション	サポートされるマクロ	コメント
アクション 期間条件、デフォルトのオペレーションステップ期間、ステップ期間	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません
通知とコマンド		

ロケーション	サポートされるマクロ	コメント
トリガーベース	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ イベント更新マクロ: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} 式マクロ: {?EXPRESSION} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ターゲットホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR}を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} ユーザー定義マクロ	参照: メッセージでのマクロの使用
問題の更新	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ イベント更新マクロ: すべて（[{EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}を除く）] 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ターゲットホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR}を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} {USER.FULLNAME} ユーザー定義マクロ	
問題の復旧	復旧イベントマクロ {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	

ロケーション		サポートされるマクロ	コメント
	ディスカバリ	アクションマクロ 日付/時刻マクロ ディスカバリマクロ イベントマクロ: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} ターゲットホストマクロ（コマンドのみ） プロキシマクロ	
	自動登録	アクションマクロ 日付/時刻マクロ イベントマクロ: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} ホストマクロ: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} ターゲットホストマクロ（コマンドのみ） プロキシマクロ	
	サービスベース	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.NAME}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.STATUS}, {EVENT.TIME}, {EVENT.TIMESTAMP}, {EVENT.VALUE} サービスマクロ ユーザー定義マクロ	
	サービス復旧	イベントマクロ（{EVENT.ACK.STATUS}, {EVENT.OPDATA}を除く）、復旧イベントマクロ	
	サービス更新	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ（{EVENT.ACK.STATUS}, {EVENT.OPDATA}を除く） イベント更新マクロ: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.MESSAGE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} サービスマクロ ユーザー定義マクロ	
内部通知		アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ（{EVENT.NSEVERITY}, {EVENT.SEVERITY}を除く）、復旧イベントマクロ ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} プロキシマクロ	
	アイテムベース	アイテムマクロ: {ITEM.STATE}, {ITEM.STATE.ERROR}	
	LLD ルールベース	低レベルディスカバリ通知マクロ	

ロケーション		サポートされるマクロ	コメント
アラートスクリプトパラメータコネクタ	トリガーベース	トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATE}, {TRIGGER.STATE.ERROR}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME} ユーザー定義マクロ アラートマクロ ユーザー定義マクロ（グローバルのみ）	
	URL、ユーザー名、パスワード、ベアラトークン、*HTTP* プロキシ、*SSL* 証明書、*SSL* キーファイル、*SSL* キーパスワードフィールド	ユーザー定義マクロ（グローバルのみ）	
	タイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。

ロケーション		サポートされるマクロ	コメント
グラフ名		式マクロ: {?EXPRESSION}	このロケーションのこのマクロでは、パラメータに秒数を指定した **avg**、**last**、**max**、**min** 関数のみがサポートされます（時間のサフィックスが使用可能）。 マクロ内のホストとして {HOST.HOST<1-9>} マクロが使用できます（インデックス付きマクロも参照）。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
ホスト、ホストプロトタイプ	インターフェース *IP/DNS*	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} ユーザー定義マクロ [1]	[1] IP アドレスの場合、1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません
	インターフェースポート	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません
	SNMP コミュニティ *SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズ *IPMI* ユーザー名、パスワード	ユーザー定義マクロ	

ロケーション	サポートされるマクロ	コメント
アイテム、アイテムプロトタイプ、**LLD** ルール		
名前	ユーザー定義マクロ	LLD ルール名ではサポートされていません。
説明	ユーザー定義マクロ	
キーのパラメータ	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	サポートされている {HOST.*} マクロは、アイテムに選択されたインターフェースに解決されます。インターフェースのないアイテムで使用した場合、Zabbix エージェント、SNMP、JMX、IPMI インターフェースのいずれか（この優先順位）またはインターフェースがない場合は'UNKNOWN'に解決されます。
更新間隔、カスタム間隔、タイムアウト（サポートされているアイテムタイプで利用可能）、保存期間（ヒストリ/トレンド）フィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
失われたリソースの削除、失われたリソースの無効化フィールド	ユーザー定義マクロ	これらのフィールドは LLD ルールのみサポートされます。 1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。

ロケーション		サポートされるマクロ	コメント
	前処理ステップのパラメータ（カスタムスクリプトを含む）、カスタムエラーハンドリングパラメータ（値を設定およびエラーを設定フィールド）	ユーザー定義マクロ	
	フィルタの正規表現	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	これらのフィールドは LLD ルールのみサポートされます。
	オーバーライドフィルタの正規表現	ユーザー定義マクロ	これらのフィールドは LLD ルールのみサポートされます。
	オーバーライド操作: 更新間隔、ヒストリ保存期間、トレンド保存期間（アイテムプロトタイプ用）	ユーザー定義マクロ	これらのフィールドは LLD ルールのみサポートされます。 1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
アイテムタイプ			
	ブラウザアイテム パラメータ名と値	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	
	スクリプトアイテム パラメータ名と値		
	計算/集約アイテム式 (式定数、関数パラメータ、アイテムキーパラメータ、(集約アイテムのみ）フィルタ条件 (ホストグループ名とタグ名）)	ユーザー定義マクロ	
	データベースモニタアイテム ユーザー名、パスワード、*SQL* クエリフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム *URL*、クエリフィールド、リクエストボディ、ヘッダー（名前と値）、*SSL* 証明書ファイル、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	シークレットマクロを含む URL は機能しません。マクロは”******”として解決されます

ロケーション		サポートされるマクロ	コメント
	HTTP エージェントアイテム	アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	
	HTTP プロキシフィールド		
	HTTP エージェントアイテム 許可ステータスコード、*HTTP* 認証ユーザー名/パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
	許可ホストフィールド	ユーザー定義マクロ	
	JMX エージェントアイテム	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT}	
	エンドポイントフィールド	ユーザー定義マクロ	
	トラッパーアイテム 許可ホストフィールド		
	SNMP エージェントアイテム	ユーザー定義マクロ	
	SNMP OID フィールド		
	SSH および **Telnet** アイテム	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
	スクリプト	ユーザー定義マクロ	
	SSH エージェントアイテム ユーザー名、公開鍵ファイル、秘密鍵ファイル、パスワードフィールド	ユーザー定義マクロ	
	Telnet エージェントアイテム ユーザー名、パスワードフィールド	ユーザー定義マクロ	
マップ			
	URL、*URL* 名	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ {TRIGGER.ID}	

ロケーション		サポートされるマクロ	コメント
ネットワークディスカバリ	エレメントラベル	式マクロ: {?EXPRESSION} [1] ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ トリガーマクロ: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] このロケーションのこのマクロでは、パラメータに秒数を指定した **avg**、**last**、**max**、**min** 関数のみがサポートされます（時間のサフィックスが使用可能）。 マクロ内のホストとして {HOST.HOST<1-9>} マクロが使用できます。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
	リンクラベル	式マクロ: {?EXPRESSION} [1]	
	シェイプラベル	式マクロ: {?EXPRESSION} [1]	
	マップシェイプのテキストフィールド	{MAP.NAME}	
	更新間隔フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
プロキシ	*SNMP* コミュニティ、*SNMP OID* フィールド *SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズフィールド	ユーザー定義マクロ（グローバルのみ）	

ロケーション		サポートされるマクロ	コメント
プロキシグループ	アクティブエージェントのアドレス > ポート (プロキシがグループに属している場合) パッシブプロキシモードの場合: インターフェースアドレスおよびポートフィールド オーバーライドモードのアイテムタイプのタイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
	フェイルオーバー期間、最小プロキシ数フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
スクリプト	コマンド、確認テキスト (タイプ: スクリプト、手動ホストアクション)	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ [2] {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	[1] 確認テキストのみ [2] Zabbix サーバーおよび Zabbix プロキシでサポート

ロケーション		サポートされるマクロ	コメント
	コマンド (タイプ: スクリプト、手動イベントアクション)	日付/時刻マクロ イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ (復旧が発生した場合) イベント更新マクロ: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR}を除く） {MANUALINPUT} プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} ユーザー名マクロ ユーザー定義マクロ	
	確認テキスト (タイプ: スクリプト、手動イベントアクション)	イベントマクロ: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	
	URL、確認テキスト (タイプ: URL、手動ホスト/イベントアクション)		
タグ名と値		ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} トリガーマクロ: {TRIGGER.ID}（トリガータグ値のみ） ユーザー定義マクロ	タグ名と値のマクロは、イベント生成プロセス中のみ解決されます。
トリガー			
	名前	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} 位置指定マクロ/参照: $1...$9 ユーザー定義マクロ	

ロケーション		サポートされるマクロ	コメント
	イベント名	日付/時刻マクロ: {TIME}, {TIMESTAMP} 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} トリガーマクロ: {TRIGGER.EXPRESSION.EXPLAIN}	
	運用データ	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} ユーザー定義マクロ	
	式	{TRIGGER.VALUE} ユーザー定義マクロ [1]	[1] 定数および関数パラメータのみ。シークレットマクロはサポートされていません
	一致用タグ、メニューエントリ名、メニューエントリ *URL* フィールド	ユーザー定義マクロ	シークレットマクロを含む URL は機能しません。マクロは”******”として解決されます。
	説明	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} ユーザー定義マクロ	
	URL	{EVENT.ID} ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
ユーザーパラメータコマンド		位置指定マクロ/参照: $1...$9	
Web シナリオ			
	名前、エージェント、*HTTP* プロキシフィールド	ユーザー定義マクロ	

ロケーション		サポートされるマクロ	コメント
	更新間隔フィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
	変数、ヘッダー、*SSL* 証明書、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	ユーザー、パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	Web シナリオステップ名前、変数（値のみ）フィールド	ユーザー定義マクロ	
	Web シナリオステップタイムアウト、許可ステータスコードフィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
	Web シナリオステップ *URL*、ヘッダー（名前と値）、必須文字列フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	Web シナリオステップ *Post* フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Webhook メディアタイプ			
	パラメータ値	アラートマクロ トリガーベースの問題通知でサポートされるすべてのマクロ（上記参照）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
	JavaScript スクリプト、*JavaScript* スクリプトパラメータ名	ユーザー定義マクロ（グローバルのみ）	
	メニューエントリ名、メニューエントリ *URL*	{EVENT.TAGS.<tag name>}	
ウィジェット			

ロケーション		サポートされるマクロ	コメント
その他のロケーション	ゲージウィジェット説明フィールド	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ（{ITEM.STATE.ERROR}を除く） ユーザー定義マクロ（グローバルのみ）	
	ハニカムウィジェット一次/二次ラベルフィールド アイテム値ウィジェット説明フィールド トップホストウィジェットテキストデータ列	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ ユーザー定義マクロ（グローバルのみ）	
	URL（ダイナミック）ウィジェット *URL* フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	稼働時間フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされていません。
	アイテムタイプのタイムアウトフィールド（グローバル設定） ユーザーメディア有効期間フィールド		
	メールメディアタイプユーザー名、パスワードフィールド	ユーザー定義マクロ（グローバルのみ）	シークレットマクロの使用を推奨します。
	スケジュールレポート件名、メッセージフィールド	日付/時刻マクロ: {TIME}, {TIMESTAMP}	

マクロの詳細

このリストには、アプリケーション領域ごとにグループ化された各組み込みマクロの詳細が含まれています:

- アクションマクロ
- アラートマクロ
- 日付/時刻マクロ
- ディスカバリーマクロ
- イベントマクロ

    - イベント更新マクロ
    - 原因/症状イベントマクロ
    - リカバリエベントマクロ
- エスカレーションマクロ
- 関数マクロ
- ホストマクロ
    - ターゲットホストマクロ
- ホストグループマクロ
- ホストインベントリマクロ
- アイテムマクロ
- ローレベルディスカバリ通知マクロ
- マップマクロ
- プロキシマクロ
- スクリプトマクロ
- サービスマクロ
- トリガーマクロ
- ユーザー名マクロ

アクションマクロ

{ACTION.ID}

トリガーされたアクションの数値 ID。

{ACTION.NAME}

トリガーされたアクションの名前。

アラートマクロ

{ALERT.MESSAGE}

アクション設定のデフォルトメッセージの値。

{ALERT.SENDTO}

ユーザーメディア設定の送信先の値。

{ALERT.SUBJECT}

アクション設定のデフォルトの件名の値。

日付と時刻のマクロ

{DATE}

現在の日付（`yyyy.mm.dd` 形式）。

{TIME}

現在の時刻（`hh:mm:ss` 形式）

{TIMESTAMP}

現在の時刻を UNIX タイムスタンプ形式で表示します。

ディスカバリーマクロ

{DISCOVERY.DEVICE.IPADDRESS}

検出されたデバイスの IP アドレス。
 常に利用可能であり、ホストが追加されるかどうかには依存しません。

{DISCOVERY.DEVICE.DNS}

検出されたデバイスの DNS 名。
 常に利用可能であり、ホストが追加されるかどうかには依存しません。

{DISCOVERY.DEVICE.STATUS}

検出されたデバイスのステータス (UP/DOWN)。

{DISCOVERY.DEVICE.UPTIME}

特定のデバイスのディスカバリーステータスが最後に変更されてからの時間。
 例: 1h 29m 01s。
 DOWN ステータスのデバイスの場合は、ダウンタイムの期間です。

{DISCOVERY.RULE.NAME}

デバイス/サービスの存在または不在を検出したディスカバリールールの名前。

{DISCOVERY.SERVICE.NAME}

検出されたサービスの名前。
 例: HTTP。

{DISCOVERY.SERVICE.PORT}

検出されたサービスのポート。
 例: 80。

{DISCOVERY.SERVICE.STATUS}

検出されたサービスのステータス (UP/DOWN)。

{DISCOVERY.SERVICE.UPTIME}

特定のサービスの検出ステータスが最後に変更されてからの時間。
 例: 1h 29m 01s。
 DOWN ステータスのサービスの場合は、ダウンタイムの期間です。

エスカレーションマクロ

{ESC.HISTORY}

以前に送信された通知のログ、そのエスカレーションステップとステータス (送信済み、進行中、失敗)。

イベントマクロ

{EVENT.ACK.STATUS}

イベントの確認状態（はい/いいえ）。

{EVENT.AGE}

アクションをトリガーしたイベントの経過時間 (秒単位の精度)。
 エスカレーションメッセージで便利です。例: 9m 13s。

{EVENT.DATE}

アクションをトリガーしたイベントの日付。
 例: 2025.04.14.

{EVENT.DURATION}

イベントの継続時間（問題イベントと復旧イベントの時間差）を秒単位で表示します。
 問題の復旧メッセージで便利です。

{EVENT.ID}

アクションをトリガーしたイベントの数値 ID。

{EVENT.NAME}

アクションをトリガーしたイベントの名前。

{EVENT.NSEVERITY}

イベントの深刻度の数値。
 可能な値：*0* - 未分類、*1* - 情報、*2* - 警告、*3* - 軽度障害、*4* - 障害、*5* - 重度障害。

{EVENT.OBJECT}

イベントオブジェクトの数値。
 可能な値: *0* - トリガー, *1* - 検出されたホスト, *2* - 検出されたサービス, *3* - 自動登録, *4* - アイテム, *5* - ローレベルディスカバリルール。

{EVENT.OPDATA}

問題の根本となるトリガーの運用データ。

{EVENT.SEVERITY}

イベントの深刻度の名前。

{EVENT.SOURCE}

イベントソースの数値。
 可能な値: *0* - トリガー, *1* - ディスカバリ, *2* - 自動登録, *3* - 内部, *4* - サービス。

{EVENT.STATUS}

アクションをトリガーしたイベントの言語値。

{EVENT.TAGS}

イベントタグのカンマ区切りリスト。
 タグが存在しない場合は空文字列に展開されます。

{EVENT.TAGSJSON}

イベントタグのオブジェクトを含む JSON 配列。
 タグが存在しない場合は空の配列に展開されます。

{EVENT.TAGS.<tag name>}

タグ名で参照されるイベントタグの値。
 英数字以外の文字（非英語のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。

{EVENT.TIME}

アクションをトリガーしたイベントの時刻。
 例: 12:57:53。

{EVENT.TIMESTAMP}

アクションをトリガーしたイベントの UNIX タイムスタンプ。

{EVENT.VALUE}

アクションをトリガーしたイベントの数値。
 可能な値: *1* - 障害、*0* - 復旧。

原因/症状イベントマクロ

{EVENT.CAUSE.ACK.STATUS}

原因イベントの確認ステータス (Yes/No)。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.AGE}

原因イベントの経過時間（秒単位の精度）。
 エスカレーションメッセージで便利です。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.DATE}

原因イベントの日付。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.DURATION}

原因イベントの継続時間（問題イベントと復旧イベントの時間差）を秒単位で表示します。
 問題復旧メッセージで便利です。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.ID}

原因イベントの数値 ID。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NAME}

原因イベントの名前。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NSEVERITY}

原因イベントの深刻度の数値。
 可能な値：*0* - 未分類、*1* - 情報、*2* - 警告、*3* - 軽度障害、*4* - 障害、*5* - 重度障害。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.OBJECT}

原因イベントオブジェクトの数値。
 可能な値: *0* - トリガー, *1* - 検出されたホスト, *2* - 検出されたサービス, *3* - 自動登録, *4* - アイテム, *5* - ローレベルディスカバリルール。
 症状イベントのコンテキストで使用されます。

原因となる問題の基礎となるトリガーの運用データ。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SEVERITY}

原因イベントの深刻度の名前。
 可能な値：*Not classified*、*Information*、*Warning*、*Average*、*High*、*Disaster*。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SOURCE}

原因イベントソースの数値。
 可能な値：*0* - トリガー、*1* - ディスカバリ、*2* - 自動登録、*3* - 内部。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.STATUS}

原因イベントの文字列値。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS}

原因イベントタグのカンマ区切りリスト。
 タグが存在しない場合は空文字列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGSJSON}

原因イベントタグオブジェクトを含む JSON 配列。
 タグが存在しない場合は空の配列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS.<tag name>}

タグ名で参照される原因イベントタグの値。
 英数字以外の文字（非英語のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符とバックスラッシュは、引用符で囲まれたタグ名内ではバックスラッシュでエスケープする必要があります。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIME}

原因イベントの時刻。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIMESTAMP}

原因イベントの UNIX タイムスタンプ。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.UPDATE.HISTORY}

原因となる問題の更新履歴（確認応答など）。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.VALUE}

原因イベントの数値
 可能な値：*1* - 障害、*0* - 復旧
 症状イベントのコンテキストで使用されます。

{EVENT.SYMPTOMS}

症状イベントのリスト。ホスト名、イベント名、深刻度、経過時間、サービスタグと値などの詳細が含まれます。
 このマクロは原因イベントのコンテキストで使用され、症状イベントに関する情報を返します。

リカバリーイベントのマクロ

{EVENT.RECOVERY.DATE}

リカバリーイベントの日付。

{EVENT.RECOVERY.ID}

リカバリーイベントの数値 ID。

{EVENT.RECOVERY.NAME}

リカバリーイベントの名前。

{EVENT.RECOVERY.STATUS}

リカバリーイベントの文字列値。

{EVENT.RECOVERY.TAGS}

復旧イベントタグのカンマ区切りリスト。タグが存在しない場合は空文字列に展開されます。

{EVENT.RECOVERY.TAGSJSON}

復旧イベントタグオブジェクトを含む JSON 配列。タグが存在しない場合は空の配列に展開されます。

{EVENT.RECOVERY.TIME}

リカバリーイベントの時刻。

{EVENT.RECOVERY.TIMESTAMP}

リカバリーイベントの UNIX タイムスタンプ。

{EVENT.RECOVERY.VALUE}

リカバリエベントの数値。

イベント更新マクロ

{EVENT.UPDATE.ACTION}

問題の更新中に実行されたアクションの人間が読める名前。
 次の値に解決されます: *acknowledged*、*unacknowledged*、*commented*、*changed severity from (original severity) to (updated severity)*、*closed*（1 回の更新で実行されるアクションの数によって異なります）。

{EVENT.UPDATE.ACTIONJSON}

問題の更新時に実行されたアクションの詳細を含む JSON 配列。
 JSON プロパティの値の例:
- true (acknowledge, unacknowledge, close, unsuppress, cause, symptom プロパティの場合);
- < メッセージ文字列 > (message プロパティの場合);
- タイムスタンプ (suppress_until および timestamp プロパティの場合) または 0 (suppress_until プロパティが無期限抑止の場合);
- 0, 1, 2, 3, 4, 5 (old および new の深刻度プロパティの場合)。

 例: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851`

{EVENT.UPDATE.DATE}

イベント更新の日付 (アクノレッジメントなど)
{ACK.DATE} は非推奨であり、サポートされている場所で解決されない場合があります。

{EVENT.UPDATE.HISTORY}

問題の更新履歴（確認など）。
 {EVENT.ACK.HISTORY} は非推奨であり、サポートされている場所で解決されない場合があります。

{EVENT.UPDATE.MESSAGE}

問題の更新メッセージ。
 {ACK.MESSAGE} は非推奨であり、サポートされている場所で解決されない場合があります。

{EVENT.UPDATE.NSEVERITY}

問題の更新操作中に設定された新しいイベントの深刻度の数値。

{EVENT.UPDATE.SEVERITY}

問題の更新操作中に設定された新しいイベントの深刻度の名前。

{EVENT.UPDATE.STATUS}

問題の更新ステータスの数値。
 可能な値：*0* - 問題/復旧イベントによって Webhook が呼び出された、*1* - 更新操作。

{EVENT.UPDATE.TIME}

イベント更新（確認など）の時刻
{ACK.TIME} は非推奨であり、サポートされている場所で解決されない場合があります。

{EVENT.UPDATE.TIMESTAMP}

イベント更新（確認など）の UNIX タイムスタンプ

関数マクロ

{FUNCTION.VALUE}

イベント発生時のトリガー式内の N 番目のアイテムベースの関数の値。
 最初のパラメータが /host/key である関数のみがカウントされます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などの関数を指すために、数値インデックス（例：{FUNCTION.VALUE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{FUNCTION.RECOVERY.VALUE}

イベント発生時のリカバリエクスプレッション内の N 番目のアイテムベース関数の値。*
 最初のパラメータが/host/key である関数のみがカウントされます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などの関数を指すために、数値インデックス（例：{FUNCTION.RECOVERY.VALUE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

ホストマクロ

{HOST.CONN}

ホストの設定に応じて、ホストの IP アドレスまたは DNS 名。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースがない場合、このマクロは「UNKNOWN」に解決されます。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.CONN<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.DESCRIPTION}

ホストの説明。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.DESCRIPTION<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.DNS}

ホストの DNS 名。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、Web シナリオでは、マクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースがない場合、マクロは「UNKNOWN」に解決されます。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.DNS<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.HOST}

技術的なホスト名。
 このマクロが履歴関数の最初のパラメータのプレースホルダーとして使用される場合、マクロ関数はサポートされません。例: last(/{HOST.HOST}/{ITEM.KEY})。
 {HOSTNAME<1-9>} は非推奨であり、サポートされている

場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、{HOST.HOST<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.ID}

ホスト ID。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.ID<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.IP}

ホストの IP アドレス。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースがない場合、このマクロは「UNKNOWN」に解決されます。
 {IPADDRESS<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、{HOST.IP<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.METADATA}

ホストのメタデータ。

 アクティブエージェントの自動登録でのみ使用されます。

{HOST.NAME}

表示されるホスト名。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.NAME<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.PORT}

ホスト（エージェント）ポート。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、Web シナリオでは、マクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースがない場合、マクロは「UNKNOWN」に解決されます。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、{HOST.PORT<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

ターゲットホストのマクロ

{HOST.TARGET.CONN}

ターゲットホストの IP アドレスまたは DNS 名。

{HOST.TARGET.DNS}

ターゲットホストの DNS 名。

{HOST.TARGET.HOST}

ターゲットホストの技術名。

{HOST.TARGET.IP}

ターゲットホストの IP アドレス。

{HOST.TARGET.NAME}

ターゲットホストの表示名。

ホストグループマクロ

{HOSTGROUP.ID}

ホストグループ ID。

ホストインベントリマクロ

{INVENTORY.ALIAS}

ホストインベントリのエイリアスフィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.ALIAS<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.ASSET.TAG}

ホストインベントリの資産タグフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.ASSET.TAG<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.CHASSIS}

ホストインベントリのシャーシフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.CHASSIS<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.CONTACT}

ホストインベントリの連絡先フィールド。
 {PROFILE.CONTACT<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.CONTACT<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.CONTRACT.NUMBER}

ホストインベントリの契約番号フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.CONTRACT.NUMBER<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.DEPLOYMENT.STATUS}

ホストインベントリの展開状況フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.DEPLOYMENT.STATUS<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HARDWARE}

ホストインベントリのハードウェアフィールド。
 {PROFILE.HARDWARE<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HARDWARE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HARDWARE.FULL}

ホストインベントリのハードウェア (詳細) フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス (例: {INVENTORY.HARDWARE.FULL<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HOST.NETMASK}

ホストインベントリのホストサブネットマスクフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HOST.NETMASK<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HOST.NETWORKS}

ホストインベントリのネットワークフィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HOST.NETWORKS<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HOST.ROUTER}

ホストインベントリのホストルーターフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HOST.ROUTER<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HW.ARCH}

ホストインベントリの *Hardware architecture* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HW.ARCH<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.DECOMM}

ホストインベントリのハードウェア廃棄日フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HW.DATE.DECOMM<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.EXPIRY}

ホストインベントリのハードウェア保守期限フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HW.DATE.EXPIRY<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.INSTALL}

ホストインベントリのハードウェア設置日フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HW.DATE.INSTALL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.PURCHASE}

ホストインベントリのハードウェア購入日フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.HW.DATE.PURCHASE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.INSTALLER.NAME}

ホストインベントリのインストーラー名フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.INSTALLER.NAME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.LOCATION}

ホストインベントリの場所フィールド。
 {PROFILE.LOCATION<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.LOCATION<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.LOCATION.LAT}

ホストインベントリの場所の緯度フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.LOCATION.LAT<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.LOCATION.LON}

ホストインベントリの位置経度フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.LOCATION.LON<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.MACADDRESS.A}

ホストインベントリの *MAC* アドレス *A* フィールド。
 {PROFILE.MACADDRESS<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.MACADDRESS.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.MACADDRESS.B}

ホストインベントリの *MAC* アドレス *B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.MACADDRESS.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.MODEL}

ホストインベントリのモデルフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.MODEL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.NAME}

ホストインベントリの名前フィールド。
 {PROFILE.NAME<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは数値インデックス付きで使用できます。例: {INVENTORY.NAME<1-9> } は、トリガー式内で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

{INVENTORY.NOTES}

ホストインベントリの *Notes* フィールド。
 {PROFILE.NOTES<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.NOTES<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OOB.IP}

ホストインベントリの *OOB IP* アドレスフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OOB.IP<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OOB.NETMASK}

ホストインベントリの *OOB* サブネットマスクフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OOB.NETMASK<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OOB.ROUTER}

ホストインベントリの *OOB* ルーターフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OOB.ROUTER<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OS}

ホストインベントリの *OS* フィールド。
 {PROFILE.OS<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OS<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OS.FULL}

ホストインベントリの *OS (詳細)* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OS.FULL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.OS.SHORT}

ホストインベントリの *OS (Short)* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.OS.SHORT<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.CELL}

ホストインベントリの *Primary POC cell* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.CELL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.EMAIL}

ホストインベントリの *Primary POC email* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.EMAIL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.NAME}

ホストインベントリの主な *POC* 名フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.NAME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.NOTES}

ホストインベントリの主な *POC* メモフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.NOTES<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.PHONE.A}

ホストインベントリの *Primary POC phone A* フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.PHONE.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.PHONE.B}

ホストインベントリの *Primary POC phone B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.PHONE.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.SCREEN}

ホストインベントリの主な *POC* 画面名フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.PRIMARY.SCREEN<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.CELL}

ホストインベントリの *Secondary POC cell* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.CELL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.EMAIL}

ホストインベントリの *Secondary POC email* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.EMAIL<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.NAME}

ホストインベントリの *Secondary POC name* フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.NAME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.NOTES}

ホストインベントリのセカンダリ *POC* メモフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.NOTES<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.PHONE.A}

ホストインベントリの *Secondary POC phone A* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.PHONE.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.POC.SECONDARY.PHONE.B}

ホストインベントリの *Secondary POC phone B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.PHONE.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

ホストインベントリの *Secondary POC screen name* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.POC.SECONDARY.SCREEN<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SERIALNO.A}

ホストインベントリのシリアル番号 *A* フィールド。
 {PROFILE.SERIALNO<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SERIALNO.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SERIALNO.B}

ホストインベントリのシリアル番号 *B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SERIALNO.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ADDRESS.A}

ホストインベントリのサイトアドレス *A* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.ADDRESS.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ADDRESS.B}

ホストインベントリのサイトアドレス *B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.ADDRESS.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ADDRESS.C}

ホストインベントリのサイトアドレス *C* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.ADDRESS.C<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.CITY}

ホストインベントリのサイト都市フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.CITY<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

ホストインベントリのサイトの国フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.COUNTRY<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.NOTES}

ホストインベントリのサイトノートフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.NOTES<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.RACK}

ホストインベントリのサイトラックの場所フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.RACK<1-9>}）と共に使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.STATE}

ホストインベントリの都道府県フィールド。

 このマクロは、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.STATE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ZIP}

ホストインベントリの *Site ZIP/postal* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SITE.ZIP<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE}

ホストインベントリのソフトウェアフィールド。
 {PROFILE.SOFTWARE<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.A}

ホストインベントリのソフトウェアアプリケーション *A* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE.APP.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.B}

ホストインベントリのソフトウェアアプリケーション *B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE.APP.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.C}

ホストインベントリのソフトウェアアプリケーション *C* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE.APP.C<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.D}

ホストインベントリのソフトウェアアプリケーション *D* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE.APP.D<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.E}

ホストインベントリのソフトウェアアプリケーション *E* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.SOFTWARE.APP.E<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.FULL}

ホストインベントリのソフトウェア *(詳細)* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス (例: {INVENTORY.SOFTWARE.FULL<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.TAG}

ホストインベントリのタグフィールド。
 {PROFILE.TAG<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.TAG<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.TYPE}

ホストインベントリのタイプフィールド。
 {PROFILE.DEVICETYPE<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.TYPE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.TYPE.FULL}

ホストインベントリのタイプ *(詳細)* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス (例: {INVENTORY.TYPE.FULL<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.URL.A}

ホストインベントリの *URL A* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.URL.A<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.URL.B}

ホストインベントリの *URL B* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.URL.B<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.URL.C}

ホストインベントリの *URL C* フィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.URL.C<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{INVENTORY.VENDOR}

ホストインベントリのベンダーフィールド。

 このマクロは、トリガー式で最初、2 番目、3 番目などのホストを指すために、数値インデックス（例：{INVENTORY.VENDOR<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

アイテムマクロ

{ITEM.DESCRIPTION}

通知を発生させたトリガー式の N 番目のアイテムの説明。

 このマクロは、トリガー式の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックスを付けて {ITEM.DESCRIPTION<1-9>} として使用できます。インデックス付きマクロを参照してください。

{ITEM.DESCRIPTION.ORIG}

通知を発生させたトリガー式内の N 番目のアイテムの元の説明 (マクロ未解決) です。

 このマクロは、{ITEM.DESCRIPTION.ORIG<1-9>} のように数値インデックス付きで使用でき、トリガー式内の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

{ITEM.ID}

通知を発生させたトリガー式の N 番目のアイテムの数値 ID。

 このマクロは、トリガー式の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックスを付けて {ITEM.ID<1-9>} として使用できます。インデックス付きマクロを参照してください。

{ITEM.KEY}

通知を引き起こしたトリガー式の N 番目のアイテムのキー。
 このマクロは、ヒストリ関数の最初のパラメータのプレースホルダーとして使用される場合、マクロ関数はサポートされません。例: `last(/{HOST.HOST}/{ITEM.KEY})`。
 `{TRIGGER.KEY}` は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのホストを指すために、{ITEM.KEY<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{ITEM.KEY.ORIG}

通知を発生させたトリガー式内の N 番目のアイテムの元のキー (マクロが解決されていない状態)。

 このマクロは、{ITEM.KEY.ORIG<1-9>} のように数値インデックスとともに使用でき、トリガー式内の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE}

通知を発生させたトリガー式内の N 番目のアイテムの最新値。
 最新の履歴値が管理 → 一般メニューセクション (管理 → 一般) で設定された最大履歴表示期間よりも前に収集された場合、フロントエンドでは *UNKNOWN* に解決されます。
 問題名で使用した場合、問題イベントを表示するときにマクロは最新のアイテム値には解決されず、問題が発生した時点のアイテム値が保持されます。
 通知で使用した場合、トリガーが発生した時点で最新のアイテム値に解決されない場合があります。たとえば、アイテムが短時間で 2 つの値「A」と「B」を受信し、トリガーが「A」で発生した場合、通知作成時の処理遅延により、通知には「B」が最新値として表示されることがあります。これは、トリガーが発生した時点と通知が作成された時点の間に最新のアイテム値が変更されたためです。これを回避するには、{ITEM.VALUE} マクロを使用してください。このマクロはトリガーが発生した時点の値に解決され、通知に正しい値が使用されることを保証します。
 `last(/{HOST.HOST}/{ITEM.KEY})` のエイリアスです。
 テキスト/ログアイテムの解決値は、以下の場所でフロントエンドによって 20 文字に切り詰められます:
- 運用データ;
- トリガーの説明;
- トリガーの URL;
- トリガーの URL ラベル;
- アイテム値ウィジェットの説明。
 完全な値に解決するには、マクロ関数を使用できます。サーバー側では値は切り詰められません。例: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`

 このマクロは数値インデックス付きで使用できます。例: {ITEM.LASTVALUE<1-9>}‘。トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.AGE}

最新のアイテム値の収集とマクロの評価の間に経過した時間。
 人間が読める形式で表示されます (例：1m 45s)。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス (例：{ITEM.LASTVALUE.AGE<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.DATE}

最新のアイテム値が収集された日付。
 YYYYMMDD 形式で表示されます。
 トリガー名 (監視→障害リスト)、イベント名、タグ名および値にはサーバーのタイムゾーンが使用されます。それ以外の場合はユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス (例：{ITEM.LASTVALUE.DATE<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.TIME}

最新のアイテム値が収集された時刻。
 HHMMSS 形式で表示されます。
 トリガー名 (監視データ -> 障害一覧)、イベント名、タグ名および値にはサーバーのタイムゾーンが使用されます。それ以外の場合はユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス (例: {ITEM.LASTVALUE.TIME<1-9>}) とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.TIMESTAMP}

最新のアイテム値が収集された UNIX タイムスタンプ。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LASTVALUE.TIMESTAMP<1-9>}）と共に使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.AGE}

ログイベントの経過時間。
 秒単位の精度。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.AGE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.DATE}

ログエントリがログに書き込まれた日付。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.DATE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.EVENTID}

イベントログ内のイベント ID。
 Windows イベントログの監視のみで使用可能。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.EVENTID<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.NSEVERITY}

イベントログ内のイベントの数値的な深刻度。
 Windows イベントログの監視のみで使用されます。

 このマクロは数値インデックス（例：{ITEM.LOG.NSEVERITY<1-9>}）とともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すことができます。インデックス付きマクロを参照してください。

{ITEM.LOG.SEVERITY}

イベントログ内のイベントの言語的な深刻度。
 Windows イベントログの監視のみで使用されます。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.SEVERITY<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.SOURCE}

イベントログのイベントのソース。
 Windows イベントログの監視のみ。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.SOURCE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.TIME}

ログエントリがログに書き込まれた時刻。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.TIME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.LOG.TIMESTAMP}

ログエントリがログに書き込まれた UNIX タイムスタンプ。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.LOG.TIMESTAMP<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.NAME}

すべてのマクロが解決されたアイテム名。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.NAME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.NAME.ORIG}

アイテムの元の名前 (マクロが解決されていない状態) です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます (例: {ITEM.NAME.ORIG<1-9>)。インデックス付きマクロ を参照してください。

{ITEM.STATE}

通知を引き起こしたトリガー式の N 番目のアイテムの最新の状態。
 可能な値: *Not supported*、*Normal*。

 このマクロは数値インデックス付きで使用できます。例: {ITEM.STATE<1-9>} は、トリガー式の 1 番目、2 番目、3 番目などのアイテムを指します。インデックス付きマクロを参照してください。

{ITEM.STATE.ERROR}

アイテムがサポートされなくなった理由の詳細を含むエラーメッセージ。
 アイテムがサポートされない状態になり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

{ITEM.VALUE}

次のいずれかに解決されます:
1) トリガー状態変更のコンテキストで使用された場合、トリガー式内の N 番目のアイテムのイベント発生時の履歴値。たとえば、イベントの表示や通知の送信時。
2) トリガー状態変更のコンテキストなしで使用された場合、トリガー式内の N 番目のアイテムの最新値。たとえば、ポップアップ選択ウィンドウでトリガーのリストを表示する場合。この場合、{ITEM.LASTVALUE} と同じ動作になります。
 最初の場合、履歴値がすでに削除されているか、保存されたことがない場合は *UNKNOWN* に解決されます。
2 番目の場合、かつフロントエンドのみで、最新の履歴値が最大履歴表示期間（管理 → 一般メニューセクションで設定）よりも前に収集された場合は *UNKNOWN* に解決されます。
 テキスト/ログアイテムの解決値は、以下の場所でフロントエンドによって 20 文字に切り捨てられます:
- 運用データ;
- トリガーの説明;
- トリガー URL;
- トリガー URL ラベル;
- アイテム値ウィジェットの説明。
 完全な値に解決するには、マクロ関数を使用できます。サーバーでは値が切り捨てられません。例: `{{ITEM.VALUE}.regsub("(.*)", \1)}`

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス付き（例: {ITEM.VALUE<1-9>}）で使用できます。インデックス付きマクロを参照してください。

{ITEM.VALUE.AGE}

アイテム値の収集とマクロの評価の間に経過した時間。
 人間が読める形式で表示されます (例: 1m 45s)。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス (例: {ITEM.VALUE.AGE<1-9>}) とともに使用できます。インデックス付きマクロ を参照してください。

{ITEM.VALUE.DATE}

アイテム値が収集された日付。
 YYYYMMDD 形式で表示されます。
 トリガー名（監視→障害リスト）、イベント名、タグ名および値にはサーバーのタイムゾーンが使用されます。それ以外の場合はユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.VALUE.DATE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

アイテム値が収集された時刻。
 HHMMSS 形式で表示されます。
 トリガー名（監視→障害リスト）、イベント名、タグ名および値にはサーバーのタイムゾーンが使用されます。それ以外の場合はユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.VALUE.TIME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.VALUE.TIMESTAMP}

アイテム値が収集された UNIX タイムスタンプ。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{ITEM.VALUE.TIMESTAMP<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{ITEM.VALUETYPE}

通知を発生させたトリガー式内の N 番目のアイテムの値の型。
 可能な値: *0* - 数値 (浮動小数点), *1* - 文字列, *2* - ログ, *3* - 数値 (整数), *4* - テキスト。

 このマクロは、数値インデックスを使用して、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すことができます。例: {ITEM.VALUETYPE<1-9>}。インデックス付きマクロを参照してください。

ローレベルディスカバリ通知マクロ

{LLDRULE.DESCRIPTION}

通知を発生させたローレベルディスカバリルールの説明。

{LLDRULE.DESCRIPTION.ORIG}

通知を発生させたローレベルディスカバリルールの元の説明 (マクロが展開されていないもの) です。

{LLDRULE.ID}

通知を発生させたローレベルディスカバリルールの数値 ID。

{LLDRULE.KEY}

通知を発生させたローレベルディスカバリルールのキー。

{LLDRULE.KEYORIG}

通知を発生させたローレベルディスカバリルールの元のキー (マクロ未解決)。

{LLDRULE.NAME}

通知を発生させたローレベルディスカバリルールの名前。

{LLDRULE.NAMEORIG}

通知を発生させたローレベルディスカバリルールの元の名前 (マクロ未解決)。

{LLDRULE.STATE}

低レベルディスカバリルールの最新の状態。
 可能な値: サポートされていません、通常。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス（例: {LLDRULE.STATE<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

低レベルディスカバリルールがサポートされなくなった理由の詳細を含むエラーメッセージ。
 低レベルディスカバリルールがサポートされなくなり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

マップマクロ

{MAP.ID}

ネットワークマップの ID。

{MAP.NAME}

ネットワークマップ名。

プロキシマクロ

{PROXY.DESCRIPTION}

プロキシの説明。
 次のいずれかに解決されます。
 1) トリガーベースの通知の場合、トリガー式内の N 番目のアイテムのプロキシ。ここではindexedマクロを使用できます。
 2) ディスカバリーベースの通知の場合、ディスカバリーを実行したプロキシ。ここではインデックスなしで {PROXY.DESCRIPTION} を使用します。
 3) 自動登録通知の場合、アクティブエージェントが登録されたプロキシ。ここではインデックスなしで {PROXY.DESCRIPTION} を使用します。

 このマクロは、トリガー式内の最初、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{PROXY.DESCRIPTION<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

{PROXY.NAME}

プロキシ名。
 次のいずれかに解決されます。
 1) トリガーベースの通知の場合、トリガー式内の N 番目のアイテムのプロキシ。ここではインデックス付きマクロを使用できます。
 2) ディスカバリーベースの通知の場合、ディスカバリーを実行したプロキシ。ここではインデックスなしで {PROXY.NAME} を使用します。
 3) 自動登録通知の場合、アクティブエージェントが登録されたプロキシ。ここではインデックスなしで {PROXY.NAME} を使用します。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス（例：{PROXY.NAME<1-9>}）とともに使用できます。インデックス付きマクロを参照してください。

スクリプトマクロ

{MANUALINPUT}

スクリプト実行時にユーザーが指定した手動入力値です。

サービスマクロ

{SERVICE.DESCRIPTION}

マクロが解決されたサービスの説明。

{SERVICE.ID}

アクションをトリガーしたサービスの数値 ID。

{SERVICE.NAME}

マクロが解決されたサービス名。

{SERVICE.ROOTCAUSE}

サービスの障害の原因となったトリガーの障害イベントのリスト。重大度とホスト名でソートされます。
 ホスト名、イベント名、重大度、経過時間、サービスタグと値の詳細が含まれます。

{SERVICE.TAGS}

サービスイベントタグのカンマ区切りリスト。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は空文字列に展開されます。

{SERVICE.TAGSJSON}

サービスイベントタグオブジェクトを含む JSON 配列。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は空の配列に展開されます。

{SERVICE.TAGS.<tag name>}

タグ名で参照されるサービスイベントタグの値。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。
 英数字以外の文字（非英語のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符やバックスラッシュが引用符で囲まれたタグ名内にある場合は、バックスラッシュでエスケープする必要があります。

トリガーマクロ

{TRIGGER.DESCRIPTION}

トリガーの説明。
 通知テキストで `{TRIGGER.DESCRIPTION}` を使用した場合、トリガーの説明でサポートされているすべてのマクロが展開されます。
 `{TRIGGER.COMMENT}` は非推奨であり、サポートされている場所で解決されない場合があります。

マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーの認識済みイベントの数。

{TRIGGER.EVENTS.PROBLEM.ACK}

状態に関係なく、すべてのトリガーの認識された障害イベントの数。

{TRIGGER.EVENTS.PROBLEM.UNACK}

状態に関係なく、すべてのトリガーの未確認の障害イベントの数。

マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーに対する未確認イベントの数。

{TRIGGER.EXPRESSION}

トリガーの式。

{TRIGGER.EXPRESSION.EXPLAIN}

部分的に評価されたトリガー式。
 アイテムベースの関数は、イベント生成時に評価されて結果に置き換えられますが、それ以外の関数は式に記載されたまま表示されます。トリガー式のデバッグに使用できます。

{TRIGGER.EXPRESSION.RECOVERY}

トリガーの設定で *OK* イベントの生成が「リカバリー式」に設定されている場合は、トリガーのリカバリー式。それ以外の場合は空文字列を返します。

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

部分的に評価されたトリガーのリカバリー式。
 アイテムベースの関数は、イベント生成時に評価され、その結果に置き換えられますが、それ以外の関数は式に記載されたまま表示されます。トリガーのリカバリー式のデバッグに使用できます。

{TRIGGER.HOSTGROUP.NAME}

トリガーが定義されているホストグループのリストを、SQL クエリでソートし、カンマとスペースで区切ったものです。

{TRIGGER.ID}

このアクションをトリガーしたトリガーの数値 ID。

{TRIGGER.NAME}

マクロが解決されたトリガー名。
 アクションでは、`{EVENT.NAME}` を使用して、マクロが解決されたトリガーイベント/問題名を表示できることに注意してください。

{TRIGGER.NAME.ORIG}

元のトリガー名（マクロ未解決）。

{TRIGGER.NSEVERITY}

数値で表したトリガーの深刻度。
 指定可能な値: *0* - 未分類, *1* - 情報, *2* - 警告, *3* - 軽度障害, *4* - 障害, *5* - 重度障害。

問題状態のトリガーに対して認識された問題イベントの数。

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

障害状態のトリガーに対する未確認の障害イベントの数。

{TRIGGER.SEVERITY}

トリガーの深刻度名。
 管理 > 一般設定 > トリガーの表示オプションで定義できます。

{TRIGGER.STATE}

トリガーの最新の状態。
 可能な値: *Unknown*、*Normal*。

トリガーがサポートされなくなった理由の詳細を含むエラーメッセージ。
 トリガーがサポートされなくなり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

{TRIGGER.STATUS}

操作ステップ実行時のトリガーの値。
 可能な値: *Problem*、*OK*。
 {STATUS} は非推奨であり、サポートされている場所で解決されない場合があります。

{TRIGGER.TEMPLATE.NAME}

トリガーが定義されているテンプレートのリストを SQL クエリでソートし、カンマ区切りで表示します。トリガーがホストに定義されている場合は *UNKNOWN* となります。

{TRIGGER.URL}

トリガーの URL。

{TRIGGER.URL.NAME}

トリガー URL のラベル。

{TRIGGER.VALUE}

現在の数値トリガー値。
 可能な値：*0* - トリガーは正常ステータス、*1* - トリガーは障害ステータス。

{TRIGGERS.ACK}

トリガーの状態に関係なく、マップエレメントに対して認識されたトリガーの数。
 トリガーのすべての障害イベントが認識されている場合、そのトリガーは認識されたと見なされます。

{TRIGGERS.PROBLEM.ACK}

マップ要素の認識済み障害トリガーの数。
 トリガーの障害イベントがすべて認識済みの場合、そのトリガーは認識済みとみなされます。

{TRIGGERS.PROBLEM.UNACK}

マップ要素の未確認の障害トリガーの数。
 トリガーの障害イベントのうち少なくとも 1 つが未確認の場合、そのトリガーは未確認とみなされます。

{TRIGGERS.UNACK}

トリガーの状態に関係なく、マップ要素の未確認トリガーの数。
 トリガーの障害イベントのうち、少なくとも 1 つが未確認の場合、そのトリガーは未確認とみなされます。

ユーザー名マクロ

{USER.FULLNAME}

イベントの確認を追加した、またはスクリプトを開始したユーザーの名前、姓、ユーザー名。

{USER.NAME}

スクリプトを開始したユーザーの名前。

{USER.SURNAME}

スクリプトを開始したユーザーの姓。

{USER.USERNAME}

スクリプトを開始したユーザーのユーザー名。
 {USER.ALIAS} は非推奨であり、サポートされている場所で解決されない場合があります。

インデックス付きマクロ

インデックス付きマクロの構文 **{MACRO<1-9>}** は、トリガーの式フィールド内の N 番目のアイテム、関数、またはホストのみを参照できます:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** は、トリガー式内の最初、2 番目、3 番目のホストの IP アドレスに解決されます（存在する場合）。
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** は、イベント発生時のトリガー式内の最初、2 番目、3 番目のアイテムの値に解決されます（存在する場合）。
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** は、イベント発生時の最初、2 番目、3 番目のアイテムベースの関数の値に解決されます（存在する場合）。

トリガーの文脈では、インデックス付きマクロは常にトリガー設定の式フィールドを参照し、復旧式は参照しません。たとえば、復旧イベントでは、**{ITEM.VALUE2}** は復旧時の問題式の 2 番目のアイテムの値に解決されます。

{HOST.HOST<1-9>} マクロは、グラフ名の {?func(/host/key,param)} 式マクロ内でもサポートされています。たとえば、グラフ名の {?func(/{HOST.HOST2}/key,param)} は、グラフ内の 2 番目のアイテムのホストに解決されます。

Warning:
インデックス付きマクロは、ここで述べたケース以外の文脈では解決されません。他の文脈では、インデックスなしのマクロ（**{HOST.HOST}**, **{HOST.IP}** など）を使用してください。

2 ロケーションでサポートされているユーザーマクロ

概要

このセクションには、ユーザー定義可能なマクロがサポートされている場所のリストが含まれています。

Note:
アクション、ネットワークディスカバリ、プロキシ、およびこのページのその他の場所セクションにリストされているすべての場所では、グローバルレベルのユーザーマクロのみがサポートされます。これらの場所では、ホストレベルおよびテンプレートレベルのマクロは解決されません。

Note:
マクロ値をカスタマイズするには（たとえば、短縮したり特定の部分文字列を抽出したりする場合）、マクロ関数を使用できます。

アクション

アクションでは、ユーザーマクロを次のフィールドで使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
トリガーベースの通知とコマンド		yes
トリガーベースの内部通知		yes
障害の更新通知		yes
サービスベースの通知とコマンド		yes
サービス更新通知		yes
期間		no
オペレーション		
	デフォルトの操作ステップ期間	no
	ステップ期間	no

ホスト/ホストプロトタイプ

ホストおよびホストプロトタイプの設定では、ユーザーマクロを以下のフィールドで使用できます。

場所		複数のマクロ/テキストとの混在[1]
インターフェース IP/DNS		DNS のみ
インターフェースポート		不可
SNMP v1, v2		
	SNMP コミュニティ	可
SNMP v3		
	コンテキスト名	可
	セキュリティ名	可
	認証パスフレーズ	可
	プライバシーパスフレーズ	可
IPMI		
	ユーザー名	可
	パスワード	可
タグ[2]		
	タグ名	可
	タグ値	可

アイテム / アイテムプロトタイプ

アイテムまたはアイテムプロトタイプの設定では、以下のフィールドでユーザーマクロを使用できます。

場所	
アイテム名	
アイテムキーのパラメータ	
更新間隔	
カスタム間隔	
タイムアウト (サポートされているアイテムタイプで利用可能)	
保存期間 (ヒストリおよびトレンド)	
説明	
計算/集約アイテム	
	式 (定数および関数パラメータ; アイテムキーのパラメータ; (集約アイテムのみ
データベース監視	
	ユーザー名
	パスワード
	SQL クエリ
HTTP エージェント	
	URL[3]
	クエリフィールド
	リクエストボディ
	ヘッダー (名前と値)
	必要なステータスコード
	HTTP プロキシ
	HTTP 認証ユーザー名
	HTTP 認証パスワード
	SSL 証明書ファイル
	SSL キーのファイル
	SSL キーのパスワード
	許可されたホスト
JMX エージェント	
	JMX エンドポイント
スクリプトアイテム	
	パラメータ名と値
ブラウザーアイテム	
	パラメータ名と値
SNMP エージェント	
	SNMP OID
SSH エージェント	
	ユーザー名
	公開鍵ファイル
	秘密鍵ファイル
	パスワード
	スクリプト
TELNET エージェント	
	ユーザー名
	パスワード
	スクリプト
Zabbix トラッパー	
	許可されたホスト
タグ[2]	
	タグ名
	タグ値
プリプロセス手順	
	パラメータ (カスタムスクリプトを含む)
	カスタムエラーハンドリングパラメータ (値を設定およびエラーを設定フィー

ローレベルディスカバリ

ローレベルディスカバリルールでは、以下のフィールドでユーザーマクロを使用できます。

場所		複数のマクロ/テキストとの混在[1]
キーのパラメータ		はい
更新間隔		いいえ
カスタム間隔		いいえ
タイムアウト (対応するアイテムタイプで利用可能)		いいえ
失われたリソースの削除		いいえ
失われたリソースの無効化		いいえ
説明		はい
SNMP エージェント		
	SNMP OID	はい
SSH エージェント		
	ユーザー名	はい
	公開鍵ファイル	はい
	秘密鍵ファイル	はい
	パスワード	はい
	スクリプト	はい
TELNET エージェント		
	ユーザー名	はい
	パスワード	はい
	スクリプト	はい
Zabbix トラッパー		
	許可されたホスト	はい
データベースモニター		
	ユーザー名	はい
	パスワード	はい
	SQL クエリ	はい
JMX エージェント		
	JMX エンドポイント	はい
HTTP エージェント		
	URL[3]	はい
	クエリフィールド	はい
	リクエストボディ	はい
	ヘッダー (名前と値)	はい
	必要なステータスコード	はい
	HTTP 認証ユーザー名	はい
	HTTP 認証パスワード	はい
フィルター		
	正規表現	はい
オーバーライド		
	フィルター: 正規表現	はい
	操作: 更新間隔 (アイテムプロトタイプ用)	いいえ
	操作: 履歴保存期間 (アイテムプロトタイプ用)	いいえ
	操作: トレンド保存期間 (アイテムプロトタイプ用)	いいえ

ネットワークディスカバリ

ネットワークディスカバリルール では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
監視間隔		no
SNMP v1, v2		
	SNMP コミュニティ	yes
	SNMP OID	yes
SNMP v3		
	コンテキスト名	yes
	セキュリティ名	yes
	認証パスフレーズ	yes
	プライバシーパスフレーズ	yes
	SNMP OID	yes

プロキシ

プロキシ構成では、以下のフィールドでユーザーマクロを使用できます。

場所	複数マクロ/テキストとの混在[1]
アクティブエージェントのアドレス > ポート（プロキシがグループに属している場合）	不可
インターフェースアドレスとポート（パッシブプロキシの場合）	不可
アイテムタイプのタイムアウト	不可

プロキシグループ

プロキシグループの設定では、以下のフィールドでユーザーマクロを使用できます。

場所	複数のマクロ/テキストとの混在[1]
フェイルオーバー期間	不可
最小プロキシ数	不可

テンプレート

テンプレートの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
タグ[2]		
	タグ名	yes
	タグ値	yes

トリガー

トリガーの設定では、以下のフィールドでユーザーマクロを使用できます。

場所		複数マクロ/テキストとの混在[1]
名前		可
オペレーショナルデータ		可
式（定数および関数パラメータのみ。シークレットマクロは非対応）		可
一致用タグ		可
メニューエントリ名		可
メニューエントリ URL[3]		可
説明		可
タグ[2]		
	タグ名	可
	タグ値	可

Web シナリオ

Web シナリオの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
名前		yes
監視間隔		no
エージェント		yes
HTTP プロキシ		yes
変数 (値のみ)		yes
ヘッダー (名前と値)		yes
ステップ		
	名前	yes
	URL[3]	yes
	変数 (値のみ)	yes
	ヘッダー (名前と変数)	yes
	タイムアウト	no
	要求文字列	yes

ロケーション		複数のマクロ/テキストとの混合[1]
	要求ステータスコード	no
認証		
	ユーザー	yes
	パスワード	yes
	SSL 証明書ファイル	yes
	SSL 秘密鍵ファイル	yes
	SSL 秘密鍵パスワード	yes
タグ[2]		
	タグ名	yes
	タグ値	yes

その他の場所

ここに記載されている場所に加えて、ユーザーマクロは以下のフィールドでも使用できます。

場所	複数のマクロ/テキストとの混在[1]
グローバルスクリプト(URL、スクリプト、SSH、Telnet、IPMI)、確認テキストを含むWebhooks	はい
JavaScript スクリプト	いいえ
JavaScript スクリプトパラメータ名	いいえ
JavaScript スクリプトパラメータ値	はい
ダッシュボード	
Top hosts ダッシュボードウィジェットの *Text* データ型の列	はい
Item value および *Gauge* ダッシュボードウィジェットの *Description* パラメータ	はい
Honeycomb ダッシュボードウィジェットの Primary/Secondary ラベル *Text* パラメータ	はい

場所		複数のマクロ/テキストとの混在[1]
ユーザー→ユーザー→メディア	*URL*[3] パラメータ（*URL* ダッシュボードウィジェット）	はい
管理→一般→*GUI*	有効時	いいえ
管理→一般→タイムアウト	作業時間	いいえ
管理→一般→コネクタ	アイテムタイプのタイムアウト	いいえ
	URL	はい
	ユーザー名	はい
	パスワード	はい
	ベアラートークン	はい
	タイムアウト	いいえ
	HTTP プロキシ	はい
	SSL 証明書ファイル	はい
	SSL キーファイル	はい
	SSL キーパスワード	はい

場所		複数のマクロ/テキストとの混在[1]
アラート→メディアタイプ→メッセージテンプレート	件名	はい
	メッセージ	はい
アラート→メディアタイプ→スクリプト	スクリプトパラメータ	はい

場所		複数のマクロ/テキストとの混在[1]
アラート→メディアタイプ→メディアタイプ	*Email* メディアタイプの *Username* および *Password* フィールド（*Authentication* が「Username and password」に設定されている場合；シークレットマクロ推奨）	はい

Zabbix でサポートされているすべてのマクロの完全なリストについては、サポートされているマクロを参照してください。

脚注

[1] フィールド内の複数のマクロまたはテキストと混合されたマクロがそのロケーションでサポートされていない場合は、1 つのマクロでフィールド全体を埋める必要があります。

[2] タグ名と値で使用されるマクロは、イベント生成プロセス中にのみ解決されます。

[3] シークレットマクロを含む URL は機能しません。それらのマクロは”******” のように解決されます。

7 単位記号

概要

86400、104857600、1000000 のような大きな値を扱うのは難しく、エラーの原因となることがあります。設定を簡素化し、可読性を向上させるために、Zabbix は値の乗数として機能する単位記号（サフィックス）をサポートしています。

サフィックスなしのトリガー式:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

サフィックス付きのトリガー式:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

サフィックスは、アイテムやウィジェットなど他のエンティティの設定を簡素化したり、アイテム値を人間が読みやすい形式で表示したりするのにも役立ちます。

Note:
エンティティの設定フィールドがサフィックスをサポートしているかどうかを確認するには、常に設定するエンティティの関連ページを参照してください。

時間のサフィックス

Zabbix フロントエンドは、エンティティの設定で次の時間のサフィックスをサポートしています。

- **s** - 秒 *(使用時は生の値と同じ動作)*
- **m** - 分
- **h** - 時間
- **d** - 日
- **w** - 週
- **M** - 月 *(トレンド関数のみ)*
- **y** - 年 *(トレンド関数のみ)*

時間のサフィックスは整数値でのみサポートされます。たとえば、1h はサポートされますが、1,5h や 1.5h はサポートされません。代わりに 90m を使用してください。

メモリサイズのサフィックス

Zabbix は以下のメモリサイズのサフィックスをサポートしています:

- **K** - キロバイト
- **M** - メガバイト
- **G** - ギガバイト
- **T** - テラバイト

アイテム値のサフィックス

サフィックスは、数値アイテム値を人間が読みやすい形式で表示するためにも使用できます。

これを有効にするには、アイテムの設定時に単位フィールドで次のいずれかのサフィックスを使用します。

- **B** - バイト
- **Bps** - 秒あたりのバイト数
- **s** - 秒、最大 3 つの非ゼロの時間単位で表示
- **uptime** - 経過時間を hh:mm:ss または N 日, hh:mm:ss で表示
- **unixtime** - Unix タイムスタンプ、yyyy.mm.dd hh:mm:ss 形式で表示

これらのサフィックスの解釈と表示方法には、以下の追加ルールが適用されます。

- B および Bps の場合、Zabbix はJEDEC標準に従い、2 進数変換（1K = 1024B）を使用します。
- その他の単位（Hz、W など）の場合、Zabbix は 10 進数変換（1K = 1000）を使用します。
- s（秒）の場合:
  - フォーマットは yyy mmm ddd hhh mmm sss ms を含み、最大 3 つの非ゼロの時間単位のみが表示されます（例: 1M 10d 4h）。
  - 単位がゼロで 2 つの非ゼロ単位の間にある場合、省略されます（例: 10d 56m、10d 0h 56m ではなく）。

単位が使用されている場合、次の乗数サフィックスがアイテム値に自動的に適用されます。

- **K**, **M**, **G**, **T** - キロ、メガ、ギガ、テラ
- **P**, **E**, **Z**, **Y** - ペタ、エクサ、ゼタ、ヨタ（これらはフロントエンドでのみ適用されます）

単位変換を防ぐには、! プレフィックスを使用します（例: !B や!s）。

以下の例は、指定された単位に基づいて受信したアイテム値がどのように変換されるかを示しています。

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016ms
3470400 s → 1M 10d 4h
2606400 s → 1M 4h
2592000 s → 1M
2592001 s → 1M
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1 day, 00:00:00
881764 uptime → 10 days, 04:56:04
32417764 uptime → 375 days, 04:56:04
32417764 !uptime → 32417764 uptime
```

```
881764 unixtime → 1970-01-11 04:56:04 AM

17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
Zabbix 4.0 以前では、ms、rpm、RPM、% からなるハードコードされた単位のストップリストがありました。このストップリストは非推奨となったため、このような単位の変換を防ぐ正しい方法は!ms、!rpm、!RPM、!% です。

8 日時フォーマット

概要

日時を設定するには、次の形式を使用する必要があります。

`d-d,hh:mm-hh:mm`

記号は次のことを表します。

記号	説明
d	曜日: 1 - 月、2 - 火、...、7 - 日
hh	時: 00-24
mm	分: 00-59

セミコロン (;) をセパレーターとして複数の期間を指定できます。

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

空白ではデフォルト値である 1-7,00:00-24:00 になります。

Attention:
指定時間の上限は含まれていません。したがって、09:00-18:00 を指定すると、最後の 1 秒は 17:59:59 になります。

例

営業時間月～金 9:00～18:00

`1-5,09:00-18:00`

平日営業時間と週末営業時間月～金 9:00～18:00 土～日 10:00～16:00

`1-5,09:00-18:00; 6-7,10:00-16:00`

9 コマンドの実行

Zabbix は、外部チェック、ユーザーパラメーター、system.run アイテム、カスタムアラートスクリプト、リモートコマンド、およびグローバルスクリプトに共通の機能を使用します。

実行手順

Note:
デフォルトでは、Zabbix のすべてのスクリプトは *sh* シェルを使用して実行され、デフォルトのシェルを変更することはできません。別のシェルを利用するには、回避策としてスクリプトファイルを作成し、コマンド実行時にそのスクリプトを呼び出すことができます。

コマンド/スクリプトは、Unix および Windows プラットフォームの両方で同様に実行されます:

1. Zabbix（親プロセス）が通信のためのパイプを作成する
2. Zabbix が、作成される子プロセスの出力としてパイプを設定する
3. Zabbix が子プロセスを作成する（コマンド/スクリプトを実行する）
4. 子プロセスのために新しいプロセスグループ（Unix の場合）またはジョブ（Windows の場合）が作成される
5. Zabbix は、タイムアウトが発生するか、他端で誰も書き込んでいない（すべてのハンドル/ファイルディスクリプタが閉じられた）場合までパイプから読み込む。子プロセスがさらにプロセスを作成し、それらが終了またはハンドル/ファイルディスクリプタを閉じる前に終了することがある点に注意してください。
6. タイムアウトに達していない場合、Zabbix は最初の子プロセスが終了するかタイムアウトが発生するまで待機する
7. 最初の子プロセスが終了し、タイムアウトに達していない場合、Zabbix は最初の子プロセスの終了コードをチェックし、0 と比較する（0 以外の値は実行失敗と見なされる。これは、Zabbix サーバーおよび Zabbix プロキシで実行されるカスタムアラートスクリプト、リモートコマンド、ユーザースクリプトにのみ該当）
8. この時点で、すべてが完了したと見なされ、プロセスツリー全体（つまり、プロセスグループまたはジョブ）が終了する

Attention:
Zabbix は、最初の子プロセスが終了し、かつ他のプロセスが出力ハンドル/ファイルディスクリプタを開いたままにしていない場合に、コマンド/スクリプトの処理が完了したと見なします。処理が完了すると、作成されたすべてのプロセスが終了します。

コマンド内のすべてのダブルクォートとバックスラッシュはバックスラッシュでエスケープされ、コマンドはダブルクォートで囲まれます。

終了コードのチェック

終了コードは以下の条件でチェックされます。

- カスタムアラートスクリプト、リモートコマンド、Zabbix サーバーおよび Zabbix プロキシで実行されるユーザースクリプトのみが対象です。
- 0 以外の終了コードはすべて実行失敗とみなされます。
- 失敗した実行の標準エラーおよび標準出力の内容は収集され、フロントエンド（実行結果が表示される場所）で利用可能です。
- agent/proxyの LogRemoteCommands パラメータを有効にすることで、Zabbix エージェント/プロキシで実行されたリモートコマンドの追加ログエントリが作成される場合があります。

失敗したコマンド/スクリプトに対するフロントエンドメッセージおよびログエントリの例:

- 失敗した実行の標準エラーおよび標準出力の内容（存在する場合）。
- "Process exited with code: N."（出力が空で、終了コードが 0 以外の場合）。
- "Process killed by signal: N."（シグナルによってプロセスが終了した場合、Linux のみ）。
- "Process terminated unexpectedly."（不明な理由でプロセスが終了した場合）。

参考

- 外部チェック
- ユーザーパラメータ
- system.run アイテム
- カスタムアラートスクリプト
- リモートコマンド
- グローバルスクリプト

10 バージョン間の互換性

サポートされているエージェント

Zabbix 8.0 と互換性を持たせるためには、Zabbix エージェントのバージョンが 1.4 以上 8.0 以下である必要があります。

一部のパラメータが変更されているため、古いエージェントの設定を見直す必要がある場合があります。たとえば、3.0 以前のバージョンではログ関連のパラメータが変更されています。

最新の機能、メトリクス、パフォーマンスの向上、メモリ使用量の削減を最大限に活用するには、最新のサポートされているエージェントを使用してください。

Windows XP に関する注意事項

- 32 ビット版 Windows XP では、Zabbix エージェントは 6.0.x より新しいものを使用しないでください。
- Windows XP/Server 2003 では、Zabbix 4.0.x より新しいエージェントテンプレートを使用しないでください。新しいテンプレートは英語のパフォーマンスカウンターを使用していますが、これは Windows Vista/Server 2008 以降でのみサポートされています。

サポートされているエージェント 2

バージョン 4.4 以降の古い Zabbix agent 2 は Zabbix 8.0 と互換性があります。Zabbix agent 2 は 8.0 より新しいバージョンを使用しないでください。

Zabbix agent 2 のバージョン 4.4 および 5.0 を使用する場合、サポートされていないアイテムの更新にはデフォルトの 10 分間隔が使用されることに注意してください。

最新の機能、メトリクス、パフォーマンスの向上、メモリ使用量の削減を最大限に活用するには、最新のサポートされている agent 2 を使用してください。

サポートされている Zabbix プロキシ

Zabbix 8.0 との完全な互換性を保つには、プロキシのメジャーバージョンがサーバーと一致している必要があります。Zabbix 8.0.x プロキシのみが、Zabbix 8.0.x サーバーと完全に互換性があります。

古いプロキシは部分的にサポートされています。データの収集やスクリプトの実行は可能ですが、新しいアイテムなどの設定更新を受け取ることはできません。

Zabbix サーバーとの関係において、プロキシは以下のように分類されます。

- 最新（プロキシとサーバーが同じメジャーバージョンの場合）
- 旧式（プロキシのバージョンがサーバーより古いが、一部サポートされている場合）
- サポート外（プロキシのバージョンがサーバーの前 LTS リリースバージョンより古いまたはプロキシのバージョンがサーバーのメジャーバージョンより新しい場合）

例：

サーバーバージョン	最新プロキシバージョン	旧式プロキシバージョン	サポート外プロキシバージョン
6.4	6.4	6.0, 6.2	6.0 より古い; 6.4 より新しい
7.0	7.0	6.0, 6.2, 6.4	6.0 より古い; 7.0 より新しい
7.2	7.2	7.0	7.0 より古い; 7.2 より新しい
7.4	7.4	7.0	7.0 より古い; 7.4 より新しい
8.0	8.0	7.0, 7.2, 7.4	7.0 より古い; 8.0 より新しい

プロキシでサポートされている機能:

プロキシバージョン	データ更新	設定の更新	タスク
最新	はい	はい	はい
古い	はい	いいえ	リモートコマンド（例：シェルスクリプト）； 即時アイテム値チェック（つまり、今すぐ実行）； 注意：前処理の実値によるテストはサポートされていません。
サポート外	いいえ	いいえ	いいえ

互換性のないデーモン起動された Zabbix プロセスのバージョンがログに警告メッセージとして出力されます。

サポートされている XML ファイル

バージョン 1.8 以降の XML ファイルが Zabbix 8.0 でのインポートにサポートされています。

Attention:
XML エクスポート形式では、トリガーの依存関係は名前のみで保存されます。同じ名前のトリガーが複数存在し（たとえば、異なる深刻度や式を持つ）、それらの間に依存関係が定義されている場合、それらをインポートすることはできません。そのような依存関係は、XML ファイルから手動で削除し、インポート後に再追加する必要があります。

12 Windows 用 Zabbix sender ダイナミックリンクライブラリ

概要

Windows 環境では、外部プロセス（zabbix_sender.exe）を起動する代わりに、Zabbix sender ダイナミックリンクライブラリ（zabbix_sender.dll）を使用してアプリケーションから Zabbix サーバー/プロキシにデータを送信できます。

zabbix_sender.dll でユーザーアプリケーションをコンパイルするには、zabbix_sender.h と zabbix_sender.lib が必要です。

入手方法

zabbix_sender.dll を入手する方法は 2 つあります。

1. ダウンロードページから zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルを ZIP アーカイブとして取得します。

ダウンロードオプションを選択する際、*Encryption* で「No encryption」、*Packaging* で「Archive」を選択してください。その後、Zabbix agent（Zabbix agent 2 ではありません）をダウンロードします。

ダウンロードした ZIP アーカイブの `bin\dev` ディレクトリ内に zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルが含まれています。必要な場所にファイルを展開してください。

2. ソースから zabbix_sender.dll をビルドします（手順を参照）。

ダイナミックリンクライブラリと開発用ファイルは `bin\winXX\dev` ディレクトリに配置されます。使用するには、zabbix_sender.h ヘッダーファイルをインクルードし、zabbix_sender.lib ライブラリとリンクしてください。

関連情報

- ライブラリの使用例として、Zabbix sender ダイナミックリンクライブラリで実装されたシンプルな Zabbix sender ユーティリティのサンプル
- Zabbix sender ダイナミックリンクライブラリのインターフェース関数用のzabbix_sender.hファイル。このファイルには、各インターフェース関数の目的、引数、戻り値を説明するドキュメントが含まれています。

13 Zabbix API の Python ライブラリ

概要

zabbix_utilsは以下のための Python ライブラリです。

- Zabbix API と連携します。
- Zabbix sender のように動作します。
- Zabbix get のように動作します。

Zabbix 5.0、6.0、6.4 以降でサポートされています。

14 サービス監視のアップグレード

概要　Zabbix 6.0 ではサービスモニタリング機能が大幅に作り直されました。（変更についてはZabbix 6.0.0 の新機能を参照）

このページでは、Zabbix 6.0 以降へのアップグレード中に、以前の Zabbix バージョンで定義されたサービスと SLA がどのように変更されるかについて説明します。

サービス　古い Zabbix バージョンでは、サービスにはソフトとハードの 2 種類の依存関係がありました。アップグレード後、すべての依存関係は等しくなります。

サービス「子サービス」が以前にハード依存関係を介して「親サービス 1」にリンクされ、さらにソフト依存関係を介して「親サービス 2」にリンクされていた場合、アップグレード後「子サービス」は単に 2 つの親サービスにリンクされます（「親サービス 1」および「親サービス 2」）。

問題とサービス間のトリガーベースのマッピングは、タグベースのマッピングに置き換えられました。Zabbix 6.0 以降では、サービス設定フォームに新しいパラメータ *Problem tags* が追加され、問題のマッチングのために 1 つまたは複数のタグ名と値のペアを指定できるようになりました。サービスにリンクされているトリガーは、新しいタグ `ServiceLink:<トリガー ID>:<トリガー名>` を取得します。（タグ値は 32 文字に切り捨てられます）リンクされたサービスは、同じ値の `ServiceLink`Problem tags を取得します。

ステータス計算ルール

「ステータス計算アルゴリズム」は、以下のルールを使用してアップグレードされます。

- 計算しない → ステータスを OK に設定
- 少なくとも 1 つの子サービスに問題がある場合は問題 → 子サービスの中で最も深刻なもの
- すべての子サービスに問題がある場合は問題 → すべての子サービスに問題がある場合は最も深刻なもの

SLA　以前は、SLA ターゲットを各サービスごとに個別に定義する必要がありました。Zabbix 6.0 以降、SLA はサービススケジュール、期待されるサービスレベル目標（SLO）、計算から除外するダウンタイム期間に関する情報を含む独立したエンティティとなりました。一度設定すると、SLA はサービスタグを通じて複数のサービスに割り当てることができます。

アップグレード時:

- 各サービスごとに定義された同一の SLA はグループ化され、各グループごとに 1 つの SLA が作成されます。
- 影響を受ける各サービスには特別なタグ SLA:<ID> が付与され、同じタグが対応する SLA のサービスタグパラメータにも指定されます。
- SLA レポートの新しいメトリックであるサービス作成時刻は、既存のサービスについては 01/01/2000 00:00 に設定されます。

15 その他の問題

ログインと systemd

Zabbix ユーザーをシステムユーザーとして作成することを推奨します。つまり、ログインできないようにします。一部のユーザーはこの推奨事項を無視し、Zabbix を実行しているホストにログインするために同じアカウント（例：SSH を使用）を使用します。これにより、ログアウト時に Zabbix デーモンがクラッシュする可能性があります。この場合、Zabbix サーバーログに次のようなものが表示されます。

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

また、Zabbix エージェントのログには次のように表示されます。

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

これは、/etc/systemd/logind.conf で設定されているデフォルトの systemd 設定 RemoveIPC=yes が原因です。システムからログアウトすると、Zabbix が以前に作成したセマフォが削除され、クラッシュの原因となります。

systemd ドキュメントからの引用：

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

この問題には 2 つの解決策があります：

1. （推奨）*zabbix* アカウントを Zabbix プロセス以外で使用するのをやめ、他の用途には専用のアカウントを作成してください。
2. （非推奨）/etc/systemd/logind.conf で RemoveIPC=no を設定し、システムを再起動します。RemoveIPC はシステム全体のパラメータであり、変更するとシステム全体に影響します。

プロキシの背後で Zabbix フロントエンドを使用する

Zabbix フロントエンドがプロキシサーバーの背後で実行されている場合、リバースプロキシパスと一致させるために、プロキシ設定ファイルの Cookie パスを書き換える必要があります。以下の例を参照してください。Cookie パスが書き換えられていない場合、ユーザーが Zabbix フロントエンドにログインしようとしたときに認証の問題が発生する可能性があります。

nginx の設定ファイル例

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Apache の設定ファイル例

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
```

16 エージェントとエージェント **2** の比較

このセクションでは、Zabbix エージェントと Zabbix エージェント 2 の違いについて説明します。

パラメータ	Zabbix エージェント	Zabbix エージェント 2
プログラミング言語	C	Go（一部 C）
デーモン化	はい	systemd のみ（Windows でははい）
サポートされる拡張機能要件	C で記述されたカスタムロード可能モジュール。	Go で記述されたカスタムプラグイン。
サポートされるプラットフォーム	Linux、IBM AIX、FreeBSD、NetBSD、OpenBSD、HP-UX、Mac OS X、Solaris: 9, 10, 11、Windows: XP 以降のすべてのデスクトップおよびサーバーバージョン	Linux、Windows: サポートされている Go バージョンがインストール可能なすべてのデスクトップおよびサーバーバージョン
サポートされる暗号化ライブラリ	GnuTLS 3.1.18 以降 OpenSSL 1.0.1、1.0.2、1.1.0、1.1.1、3.0.x LibreSSL - バージョン 2.7.4、2.8.2 でテスト済み (特定の制限あり。詳細は暗号化ページを参照)。	Linux: OpenSSL 1.0.1 以降。 MS Windows: OpenSSL 1.1.1 以降。 OpenSSL ライブラリは PSK サポートが有効である必要があります。LibreSSL はサポートされていません。
監視プロセス		
プロセス	各サーバー/プロキシレコードごとに個別のアクティブチェックプロセス。	自動的にスレッドが作成される単一プロセス。 スレッドの最大数は GOMAXPROCS 環境変数で決定されます。
メトリクス	**UNIX**: サポートされているアイテムのリストを参照。 **Windows**: Windows 固有の追加アイテムのリストを参照。	**UNIX**: Zabbix エージェントがサポートするすべてのメトリクス。 さらに、エージェント 2 は Docker、Memcached、MySQL、PostgreSQL、Redis、systemd などの監視対象に対する Zabbix ネイティブの監視ソリューションを提供します。エージェント 2 固有のアイテムの完全なリストを参照。 **Windows**: Zabbix エージェントがサポートするすべてのメトリクス、および HTTPS、LDAP の net.tcp.service* チェック。 さらに、エージェント 2 は PostgreSQL、Redis に対する Zabbix ネイティブの監視ソリューションを提供します。
同時実行性	単一サーバーのアクティブチェックは逐次実行されます。	異なるプラグインからのチェックや、1 つのプラグイン内の複数のチェックが同時に実行可能です。
サードパーティトラップ	いいえ	はい
追加機能		
永続ストレージ	いいえ	はい
log*[] メトリクス用の永続ファイル	はい（Unix のみ）	いいえ
ログデータのアップロード	バッファを解放するためにログ収集中に実行可能。	バッファがいっぱいになるとログ収集が停止するため、BufferSizeパラメータは少なくとも MaxLinesPerSecond x 2 である必要があります。
実行時のユーザー変更	はい（Unix 系システムのみ）	いいえ（systemd で制御）
ユーザー設定可能な暗号スイート	はい	いいえ

関連情報**:**

- *Zabbix* プロセスの説明: Zabbix エージェント、Zabbix エージェント 2
- 設定パラメータ: Zabbix エージェントUNIX / Windows、Zabbix エージェント 2UNIX / Windows

17 エスケープの例

概要

このページでは、さまざまなコンテキストで正規表現を使用する際に正しいエスケープを使用する例を示します。

> **Note:**
> トリガー式コンストラクターを使用すると、正規表現に正しいエスケープが自動的に追加されます。

例

コンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}`

注意:

- バックスラッシュはエスケープされません;
- 引用符はエスケープされます。

アイテムキーのパラメータ内のマクロ関数

正規表現: `.+:(\d+)$<br>` アイテムキー: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

注意:

- regsub マクロ関数内の正規表現はダブルクォートで囲まれています（閉じ括弧を含むため）;
- 正規表現の周囲の引用符はエスケープされます（アイテムの 3 番目のパラメータ全体がダブルクォートで囲まれているため）;
- アイテムキーの 3 番目のパラメータはカンマを含むためダブルクォートで囲まれています。

LLD マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

注意:

- バックスラッシュはエスケープされません;
- 引用符はエスケープされます。

ユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

注意:

- LLD のバックスラッシュのエスケープは変わりません;
- LLD マクロをユーザーマクロコンテキストに挿入する際、文字列にする必要があります:

1. マクロ式の周囲に引用符が追加されます;
2. 引用符がエスケープされます。合計で 3 つの新しいバックスラッシュが追加されます。

関数の文字列パラメータ（任意）

例として `concat` を使用します。

文字列内容: `\.+\"[a-z]+<br>` 式: `concat("abc", "\\.\\\"[a-z]+")`

注意:

- 文字列パラメータはバックスラッシュと引用符の両方のエスケープが必要です。

関数の文字列パラメータ内の **LLD** マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` 式: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

注意:

- 文字列パラメータはバックスラッシュと引用符の両方のエスケープが必要です;
- 文字列がアンコートされた後にマクロが解決されるため、もう 1 段階のエスケープが追加されます;

関数の文字列パラメータ内のコンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 式: `concat("abc,"{$MACRO:regex:\"\\.+\\\\\\"[a-z]+\"}")`

注意:

- 前の例と同様に、追加のエスケープレイヤーが必要です;
- バックスラッシュと引用符は、最上位のエスケープ（文字列パラメータであることによる）でのみエスケープされます。

関数内のユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z])+<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` 式: `concat("abc,"{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

3 つのエスケープレイヤーに注意してください:

1. LLD マクロ関数用（バックスラッシュのエスケープなし）;
2. ユーザーマクロコンテキスト用（バックスラッシュのエスケープなし）;
3. 関数の文字列パラメータ用（バックスラッシュのエスケープあり）。

文字列内のコンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 何らかの式の文字列内、例えば: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

注意:

- 文字列もバックスラッシュのエスケープが必要です;
- 文字列も引用符のエスケープが必要です;
- 再び 2 段階のエスケープのケース:

1. バックスラッシュのエスケープなしでユーザーマクロコンテキスト用のエスケープ;
2. バックスラッシュのエスケープありで文字列用のエスケープ。

21 クイックリファレンスガイド

概要

このドキュメントセクションには、一般的に必要とされる監視目標のために Zabbix をセットアップするためのクイックレシピが含まれています。

これは新しい Zabbix ユーザーを念頭に置いて設計されており、タスクの解決に必要な情報を含む他のドキュメントセクションのナビゲーターとして使用できます。

次のクイックリファレンスガイドが利用可能です。

- Zabbix エージェントで Linux を監視する
- Zabbix エージェントで Windows を監視する
- HTTP 経由で Apache を監視する
- Zabbix エージェント 2 で MySQL を監視する
- Zabbix で VMware を監視する
- Zabbix でネットワークトラフィックを監視する
- アクティブチェックを使用して Zabbix でネットワークトラフィックを監視する
- ブラウザアイテムを使用して Web サイトを監視する
- Zabbix エージェント 2(パッシブ) で Web サイトの証明書を監視する
- Zabbix でネットワークスイッチまたはルーターを監視する
- アクティブチェックを使用して Windows イベントログを監視する

1 Zabbix エージェントで Linux を監視する

はじめに　このページでは、Zabbix で Linux マシンの基本的な監視を開始するために必要な手順を説明します。このチュートリアルで説明されている手順は、Linux ベースのオペレーティングシステムすべてに適用できます。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、Linux マシンの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションをご覧ください。

前提条件

このガイドを進める前に、ダウンロードとインストールを行い、OS の手順に従って Zabbix サーバーと Zabbix フロントエンドをインストールする必要があります。

Zabbix エージェントのインストール　Zabbix エージェントはデータ収集を担当するプロセスです。

Zabbix サーバーのバージョンを確認します:

```
zabbix_server -V
```

監視したい Linux マシンに、同じバージョンの Zabbix エージェント（推奨）をインストールします。監視のニーズに応じて、Zabbix サーバーがインストールされている同じマシン、またはまったく別のマシンである場合があります。

最適なインストール方法を選択してください:

- Docker コンテナとして実行する - Zabbix Docker リポジトリで利用可能なイメージのリストを参照してください。
- Zabbix のパッケージからインストールする（Alma Linux、CentOS、Debian、Oracle Linux、Raspberry Pi OS、RHEL、Rocky Linux、SUSE Linux Enterprise Server、Ubuntu で利用可能）。
- ソースからコンパイルする。

監視のための **Zabbix** の設定　Zabbix エージェントは、アクティブモードまたはパッシブモード（同時に）でメトリクスを収集できます。

> **Note:**
> パッシブチェックは単純なデータリクエストです。Zabbix サーバーまたはプロキシがデータ（例：CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。アクティブチェックはより複雑な処理が必要です。エージェントは最初にサーバーから独立して処理するアイテムのリストを取得し、その後データをまとめて送信します。詳細はパッシブおよびアクティブエージェントチェックを参照してください。

Zabbix が提供する監視テンプレートは通常、Zabbix エージェント用のテンプレートと Zabbix エージェント（アクティブ）用のテンプレートの 2 つの選択肢を提供します。最初のオプションでは、エージェントはパッシブモードでメトリクスを収集します。これらのテンプレートは、異なる通信プロトコルを使用しますが、同一の監視結果を提供します。

以降の Zabbix の設定は、アクティブまたはパッシブの Zabbix エージェントチェック用テンプレートのいずれを選択するかによって異なります。

パッシブチェック　Zabbix agent

1. エージェントがインストールされているマシンで、エージェントの設定ファイルを開きます。

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

2. server パラメータに、Zabbix サーバーの IP アドレスまたは DNS 名 (およびオプションでポート) を追加します。例:

```
Server=192.0.2.0:10051
```

Zabbix エージェントは、このアドレスを使用して、指定された Zabbix サーバーまたはプロキシからのみ受信接続を受け入れてデータを取得します。

3. Zabbix エージェントを再起動します。

```
systemctl restart zabbix-agent
```

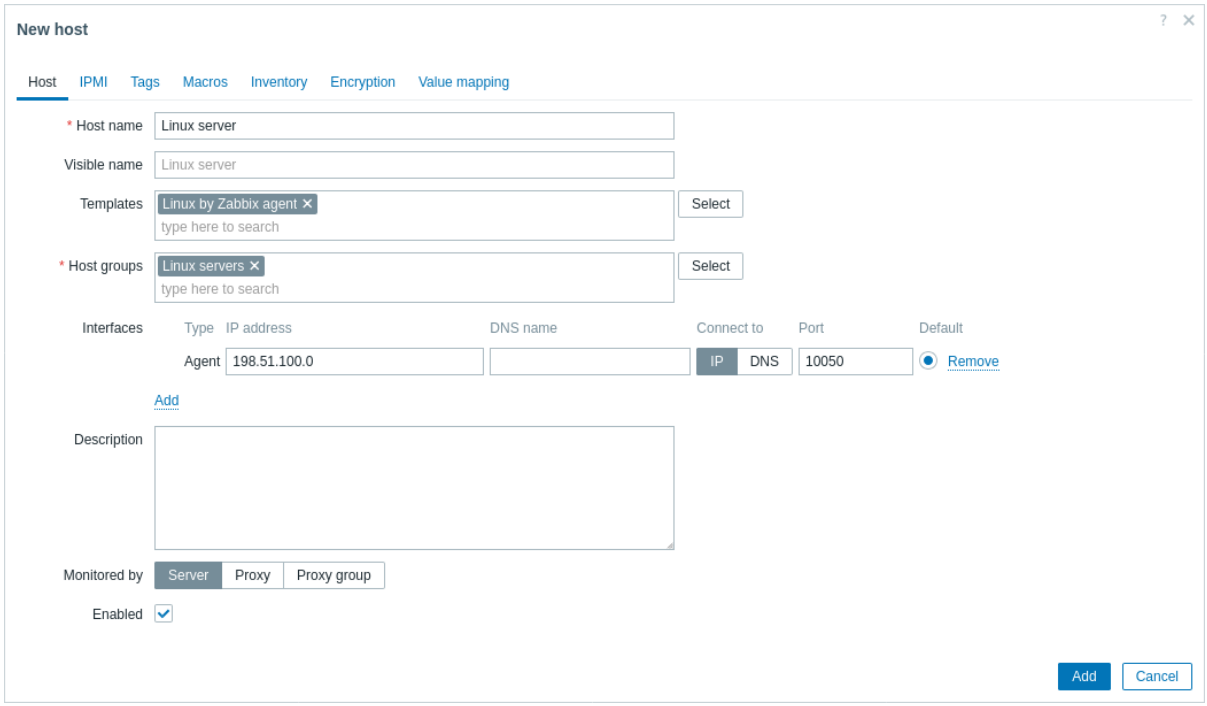
Zabbix フロントエンド

1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

- ホスト名フィールドにホスト名（例：”Linux server”）を入力します。
- テンプレートフィールドに、ホストにリンクするテンプレート”Linux by Zabbix agent” を入力または選択します。
- ホストグループフィールドに、ホストグループ（例：”Linux servers”）を入力または選択します。

- インターフェースパラメータでエージェントインターフェースを追加し、エージェントがインストールされている Linux マシンの IP アドレスまたは DNS 名を指定します。



3. 追加をクリックしてホストを追加します。このホストが監視対象の Linux マシンを表します。

アクティブチェック　Zabbix agent

1. エージェントがインストールされているマシンで、エージェントの設定ファイルを開きます。

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

2. ServerActive パラメータに、Zabbix サーバーの IP アドレスまたは DNS 名（およびオプションでポート）を追加します。例:

```
ServerActive=192.0.2.0:10051
```

Zabbix agent はこのアドレスを使用して Zabbix サーバーの trapper ポート（デフォルト: 10051）に接続し、アクティブチェック用の設定データを要求します。

3. Hostname パラメータを定義します。これはZabbix フロントエンドで定義するホスト名と一致している必要があります。例:

```
Hostname=Linux server
```

Hostname の値は一致している必要があります。なぜなら、アクティブチェックの場合、Zabbix agent はこの値を使用してサーバーから正しいホスト設定を取得するためです。具体的には、エージェントはサーバーへの接続を開始し、Hostname の値を使って自身を識別します。その後、サーバーはそのホストの監視設定を提供します。これらの値が異なる場合、エージェントは適切な設定を受け取れず、メトリクスが取得できなかったり、監視に問題が発生したりします。

4. Zabbix agent を再起動します。

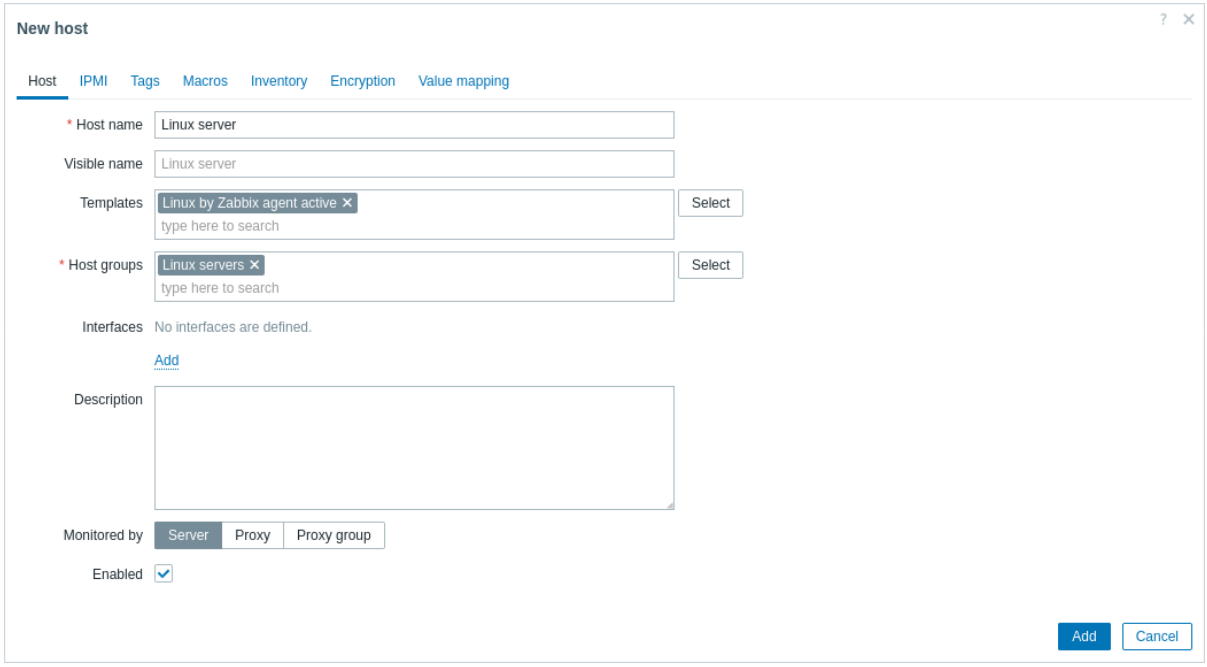
```
systemctl restart zabbix-agent
```

Zabbix フロントエンド

1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

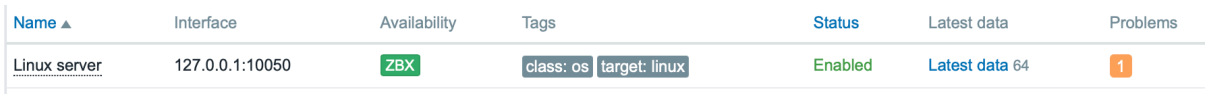
- ホスト名フィールドには、エージェントの設定ファイルで前述した *Hostname* パラメータの値と一致するホスト名（例：”Linux server”）を入力します。
- テンプレートフィールドには、ホストにリンクするテンプレート「Linux by Zabbix agent active」を入力または選択します。
- ホストグループフィールドには、ホストグループ（例：”Linux servers”）を入力または選択します。
- インターフェースは未定義のままにします。アクティブチェックでは、エージェントがサーバーへの接続を開始するため、サーバーからの接続を待ち受ける必要がなく、インターフェースは必要ありません。



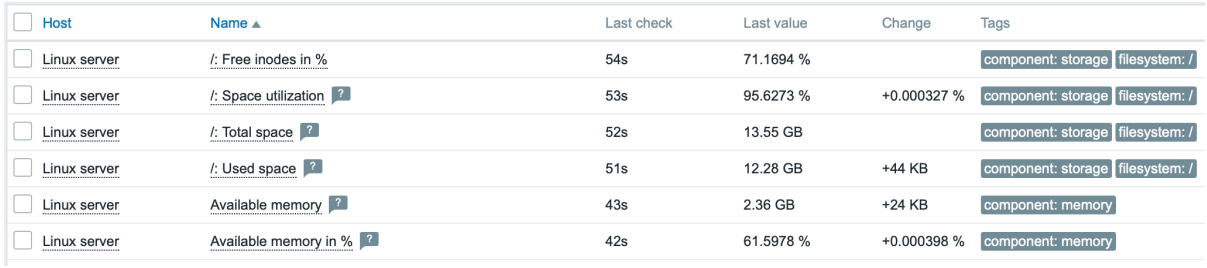
3. 追加をクリックしてホストを追加します。このホストが監視対象の Linux マシンを表します。

収集されたメトリクスの表示　おめでとうございます！この時点で、Zabbix はすでに Linux マシンを監視しています。

収集されたメトリクスを表示するには、監視-> ホストメニューセクションを開き、ホストの横にある最新データをクリックします。

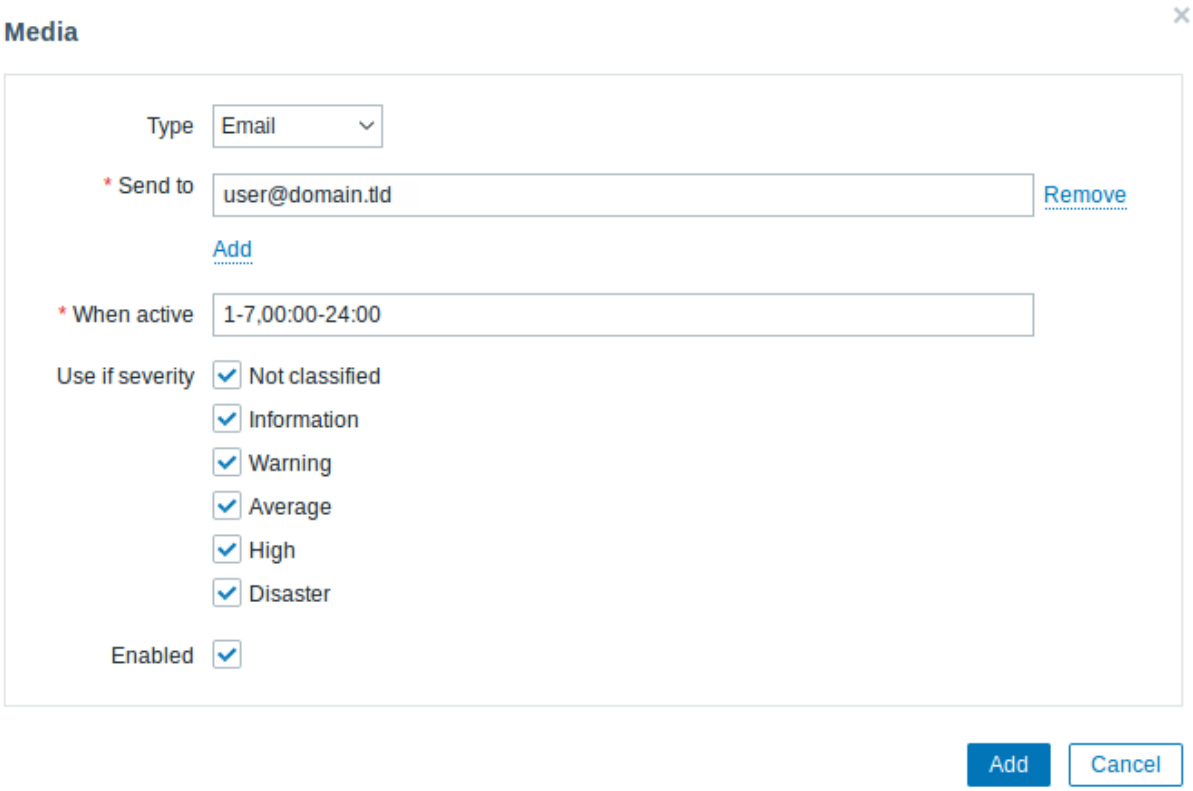


この操作により、Linux サーバーホストから収集されたすべての最新メトリクスのリストが表示されます。



問題アラートの設定　Zabbix は、さまざまな方法でインフラストラクチャの問題を通知できます。このガイドでは、メールアラートを送信するための設定手順を説明します。

1. ユーザー設定 -> プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信のガイドに従ってください。

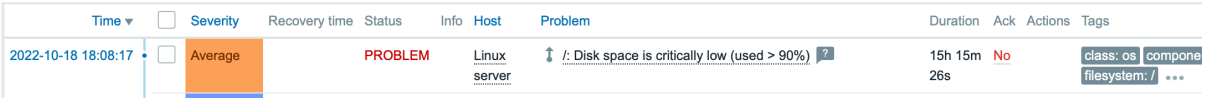
次回、Zabbix が問題を検出したときに、メールでアラートを受信できるはずです。

設定をテストする　Linux では、高い CPU 負荷をシミュレートし、その結果として問題アラートを受信するには、次のコマンドを実行します。

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

CPU 負荷がしきい値を超えるには、複数のmd5sumプロセスを実行する必要がある場合があります。

Zabbix が問題を検出すると、監視→問題セクションに表示されます。



アラートが設定されている場合は、問題の通知も受信します。

関連情報**:**

- アイテムの作成 - 追加のメトリクスの監視を開始する方法（テンプレートを使用しないカスタム監視）。
- Zabbix エージェントアイテム、Windows 用 Zabbix エージェントアイテム - Windows 上で Zabbix エージェントを使用して監視できるメトリクスの全リスト。
- 問題のエスカレーション - 複数ステップのアラートシナリオの作成方法（例：最初にシステム管理者にメッセージを送信し、45 分以内に問題が解決しない場合はデータセンターマネージャーにメッセージを送信）。
- パッケージからのインストール - さまざまな Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールする方法。これにより、最新の機能やバグ修正にアクセスできます。

2 Zabbix エージェントで **Windows** を監視する

はじめに　このページでは、Zabbix で Windows マシンの基本的な監視を開始するために必要な手順を説明します。

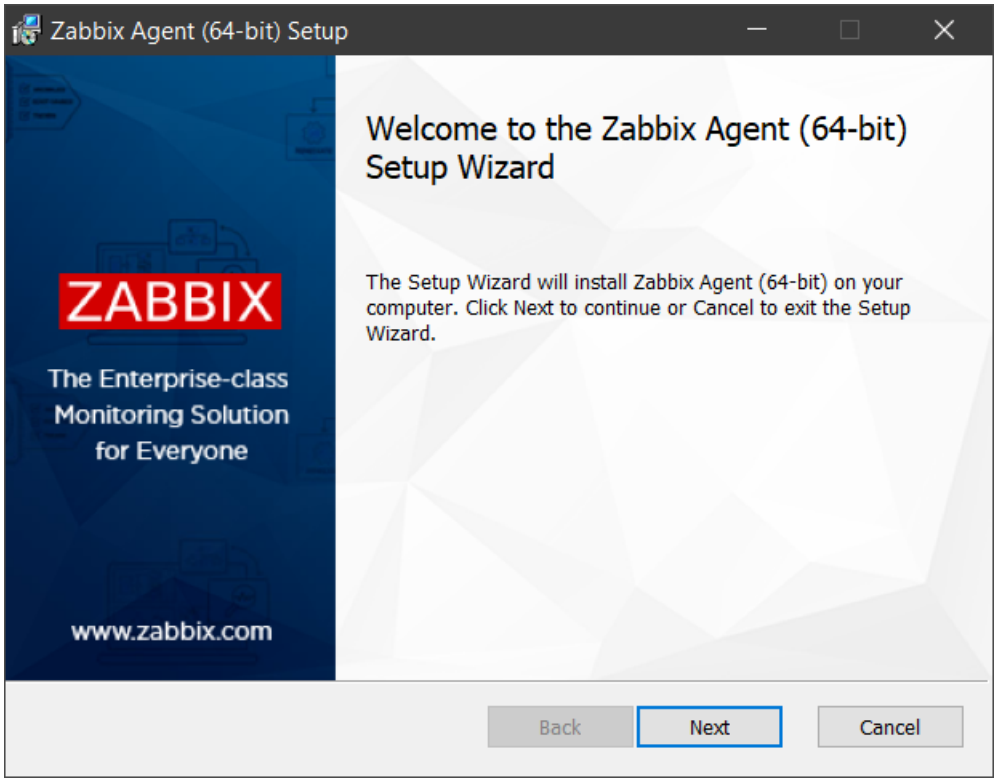
このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、Windows マシンの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズオプションやより高度な設定が必要な場合は、Zabbix マニュアルの設定セクションを参照してください。

前提条件

このインストールガイドを進める前に、OS の指示に従って Zabbix サーバーと Zabbix フロントエンドをダウンロードしてインストールする必要があります。

Zabbix エージェントのインストール　Zabbix エージェントはデータ収集を担当するプロセスです。監視したい Windows マシンにインストールする必要があります。Windowsの Zabbix エージェントインストール手順に従ってください。



監視のための **Zabbix** の設定　Zabbix エージェントは、アクティブモードまたはパッシブモード（同時に）でメトリクスを収集できます。

Note:
パッシブチェックは単純なデータリクエストです。Zabbix サーバーまたはプロキシがデータ（例：CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。アクティブチェックはより複雑な処理が必要です。エージェントは最初にサーバーから独立して処理するアイテムのリストを取得し、その後データをまとめてサーバーに送信します。詳細はパッシブおよびアクティブエージェントチェックを参照してください。

Zabbix が提供する監視テンプレートは通常、Zabbix エージェント用のテンプレートと Zabbix エージェント（アクティブ）用のテンプレートという 2 つの選択肢を提供します。最初のオプションでは、エージェントはパッシブモードでメトリクスを収集します。これらのテンプレートは、異なる通信プロトコルを使用しますが、同じ監視結果を提供します。

以降の Zabbix の設定は、アクティブまたはパッシブの Zabbix エージェントチェック用のテンプレートを選択するかによって異なります。

パッシブチェック　Zabbix frontend

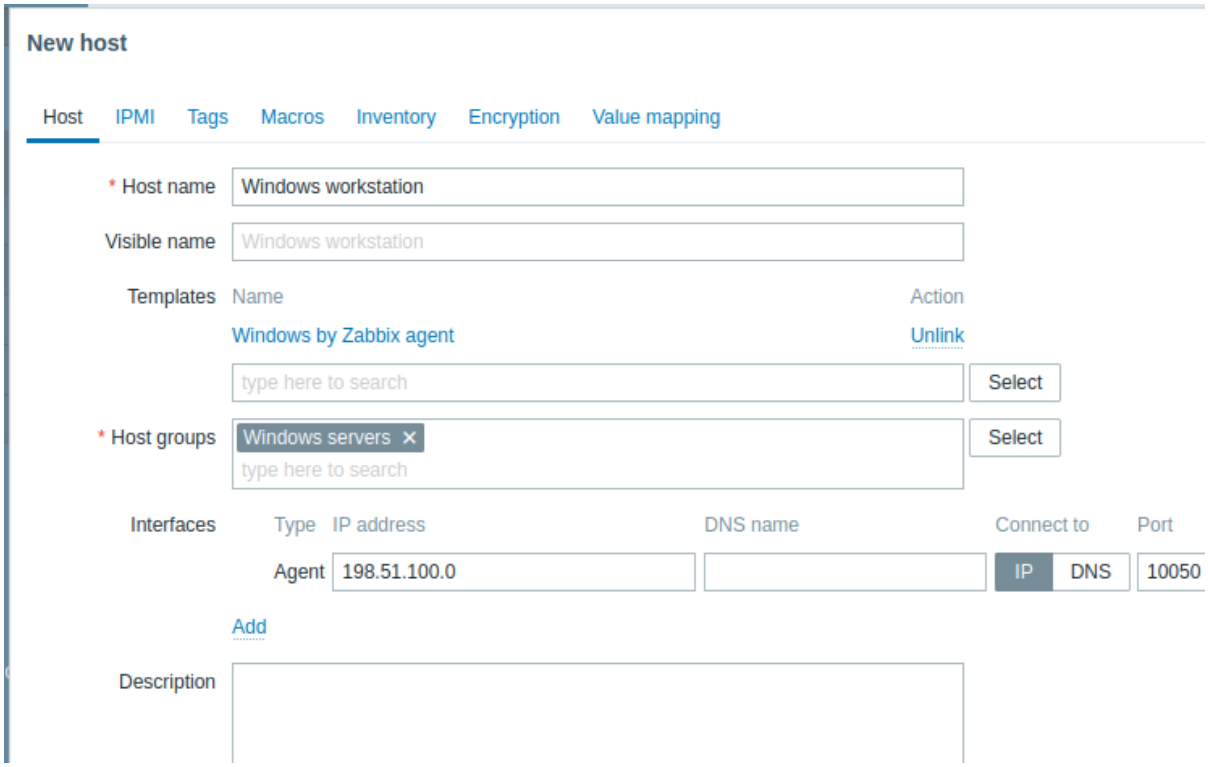
1. Log into Zabbix frontend.

2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.

4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

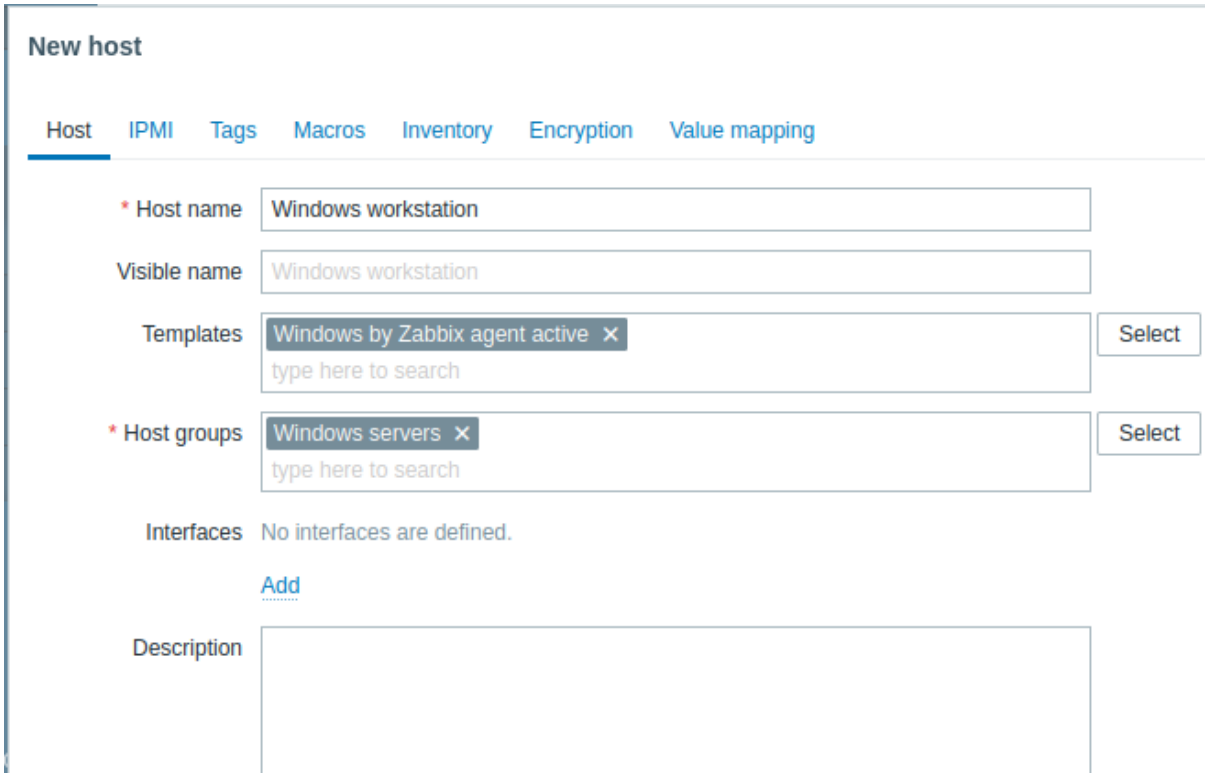
アクティブチェック　Zabbix フロントエンド

1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

このホストは、Windows マシンを表します。

3. テンプレートパラメータで、*Windows by Zabbix agent active* を入力または選択します。



Zabbix agent

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

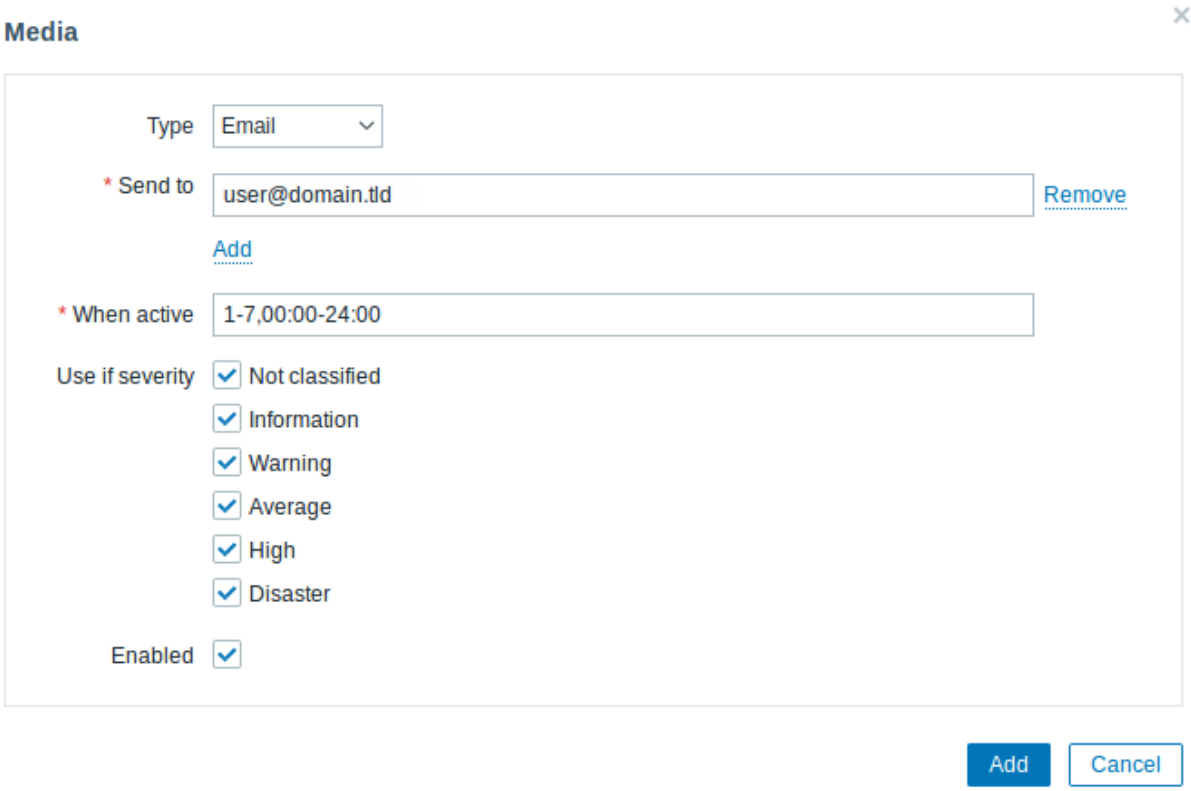
収集されたメトリックの表示　おめでとうございます！この時点で、Zabbix はすでに Windows マシンを監視しています。

収集されたメトリックを表示するには、監視-> ホストメニューセクションを開き、ホストの横にある最新データをクリックします。



問題アラートの設定　Zabbix は、さまざまな方法でインフラストラクチャの問題を通知できます。このガイドでは、メールアラートを送信するための設定手順を説明します。

1. ユーザー設定 -> プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信ガイドに従ってください。

次回、Zabbix が問題を検出した際には、メールでアラートを受け取ることができます。

Note:
Windows では、CpuStresユーティリティを使用して高い CPU 負荷をシミュレートし、その結果として問題アラートを受信することができます。

関連情報**:**

- アイテムの作成 - 追加のメトリクスを監視する方法（テンプレートを使わないカスタム監視）。
- Zabbix エージェントアイテム、Windows 用 Zabbix エージェントアイテム - Windows 上の Zabbix エージェントで監視できるメトリクスの全リスト。
- 問題のエスカレーション - 複数段階のアラートシナリオの作成方法（例：最初にシステム管理者にメッセージを送信し、45 分以内に問題が解決しない場合はデータセンター管理者にメッセージを送信）。

3 HTTP 経由で **Apache** を監視する

はじめに　このページでは、追加のソフトウェアをインストールせずに Apache Web サーバーの監視を開始する簡単な方法を説明します。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、Apache インストールの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションをご覧ください。

前提条件

このインストールガイドを進める前に、ダウンロードしてインストールする必要があります。Zabbix サーバーおよび Zabbix フロントエンドを、OS の手順に従ってインストールしてください。

Apache の準備　1. 使用している Apache のバージョンを確認します。

RHEL 系システムの場合、以下を実行します。

```
httpd -v
```

Debian/Ubuntu の場合、以下を実行します。

```
apache2 -v
```

2. Apache インスタンスでStatus モジュールが有効になっていることを確認します。

RHEL 系システムの場合、以下を実行します。

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

Debian/Ubuntu の場合、以下を実行します。

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

リストに status_module が表示されない場合は、以下を実行してモジュールを有効にします。

RHEL 系システムの場合、以下を実行します。

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

Debian/Ubuntu の場合、以下を実行します。

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

3. Zabbix サーバーの IP アドレスからステータスレポートにアクセスできるように、Apache の設定ファイルを編集します。

RHEL 系システムの場合: `/etc/httpd/conf.modules.d/status.conf`:

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

Debian/Ubuntu の場合: `/etc/apache2/mods-enabled/status.conf`:

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

以下の行をファイルに追加します（**198.51.100.255** を Zabbix サーバーの IP アドレスに置き換えてください）。

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

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

4. Apache を再起動します。

RHEL 系システムの場合、以下を実行します。

```
sudo systemctl restart httpd
```

Debian/Ubuntu の場合、以下を実行します。

```
sudo systemctl restart apache2
```

5. すべてが正しく設定されているか確認するには、以下を実行します（**198.51.100.255** を Zabbix サーバーの IP アドレスに置き換えてください）。

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

レスポンスには Apache Web サーバーの統計情報が含まれているはずです。

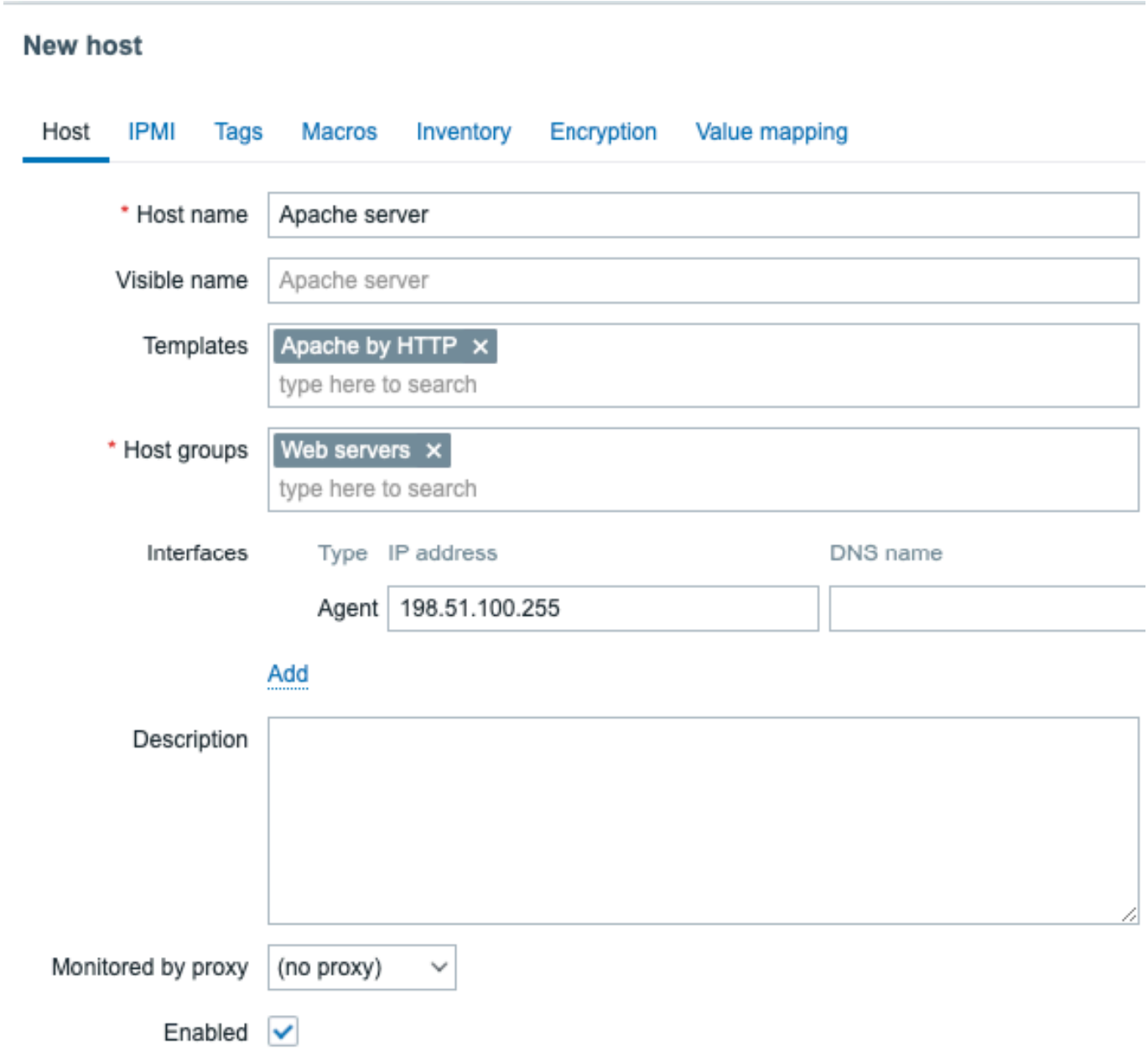
監視のための **Zabbix** の設定　1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

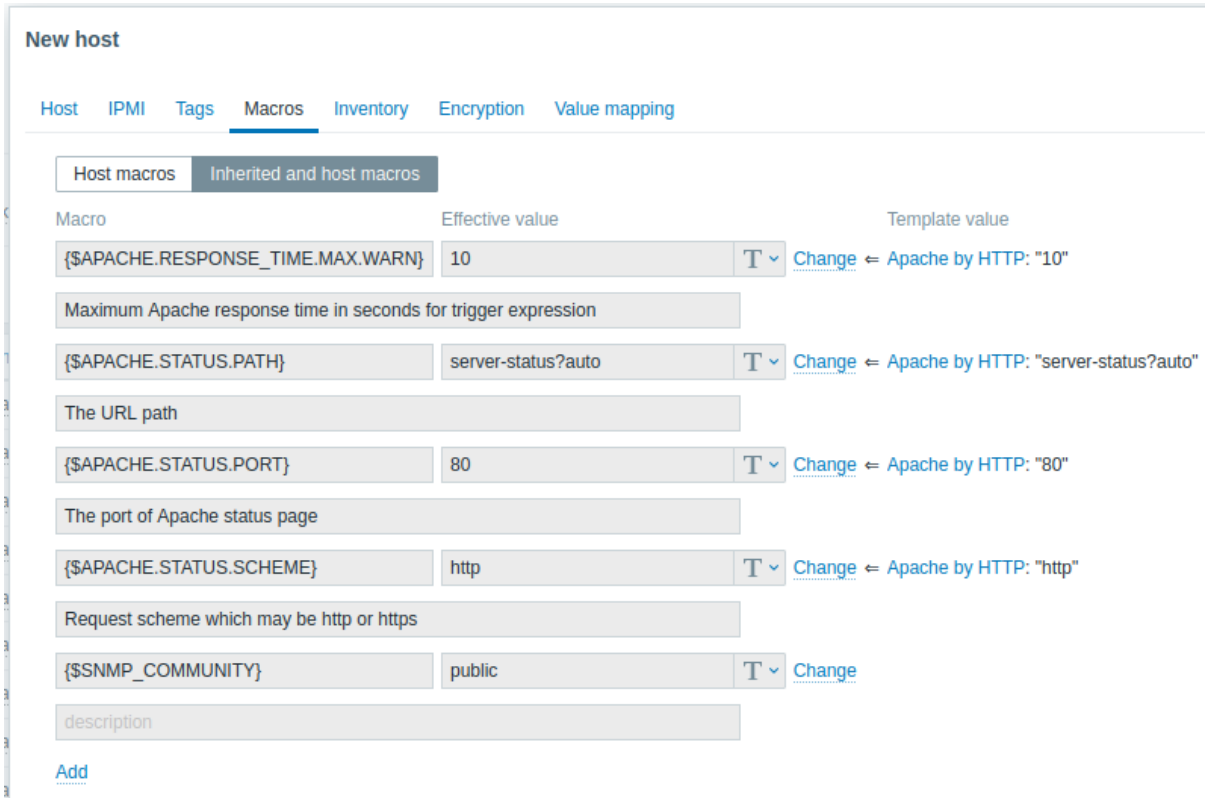
このホストは Apache サーバーを表します。

3. インターフェースパラメータでエージェントインターフェースを追加し、Apache インスタンスの IP アドレスを指定します。このマシンに **Zabbix** エージェントをインストールする必要はありません。このインターフェースは {HOST.CONN} マクロの解決のみに使用されます。このマクロはテンプレートアイテムで Apache インスタンスを特定するために使用されます。

4. テンプレートパラメータで *Apache by HTTP* を入力または選択します。



5. マクロタブに切り替え、継承されたマクロとホストマクロモードを選択します。マクロ {$APACHE.STATUS.PORT} と {$APACHE.STATUS.SCHEME} の値がインストール設定に合っているか確認します。デフォルトでは、ポートは 80、スキームは http です。異なるポートやスキームを使用している場合は、マクロの値を変更してください。

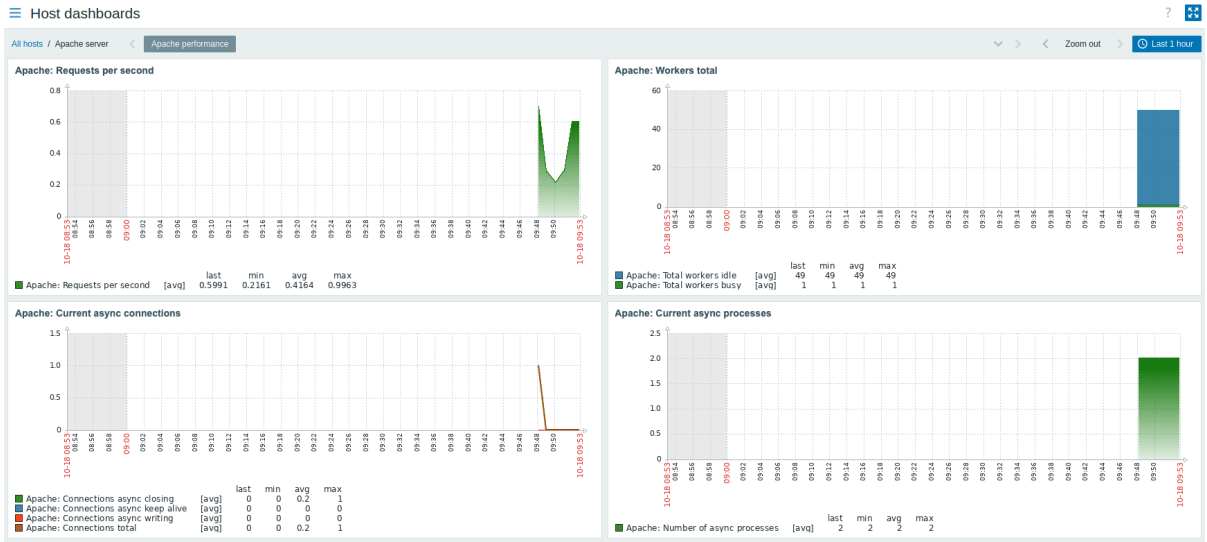


収集されたメトリクスの表示　おめでとうございます！この時点で、Zabbix はすでに Apache Web サーバーの監視を開始しています。

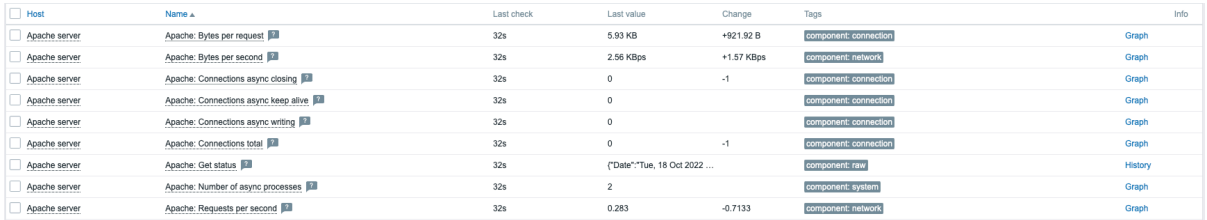
収集されたメトリクスを表示するには、監視-> ホストメニューセクションを開き、ホストの横にあるダッシュボードをクリックします。



この操作により、Apache の/server-status ページから収集された最も重要なメトリクスを表示するホストダッシュボードに移動します。



または、監視-> ホストから最新データをクリックして、リストですべての最新の収集メトリクスを表示することもできます。



問題アラートの設定　Zabbix は、さまざまな方法でインフラストラクチャの問題を通知できます。このガイドでは、メールアラートを送信するための設定手順を説明します。

1. ユーザー設定 -> プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。

2. 問題通知の受信のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受け取ることができます。

設定をテストする　実際の問題をシミュレートし、テスト用の障害アラートを受信するには:

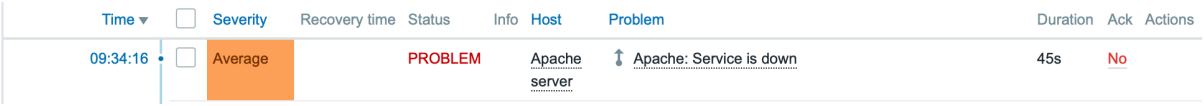
1. Zabbix で *Apache server* ホストの設定を開きます。

2. マクロタブに切り替え、継承されたマクロとホストマクロを選択します。

3. {$APACHE.STATUS.PORT} マクロの横にある変更を押して、別のポートを設定します。

4. [更新] を押してホスト設定を保存します。

5. 数分後、Zabbix は *Apache service is down* という問題を検出します。これは、インスタンスに接続できなくなったためです。この問題は、監視→障害セクションに表示されます。



アラートが設定されていれば、障害通知も受信します。

6. マクロ値を元に戻して問題を解決し、Apache の監視を続けます。

関連情報**:**

- Web サーバーのハードニング - Web サーバーのセキュリティを高めるための推奨設定。
- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- HTTP アイテム - HTTP エージェントを使用してカスタムメトリクスを監視する方法。
- 問題のエスカレーション - 複数ステップのアラートシナリオを作成する方法（例：最初にシステム管理者にメッセージを送信し、45 分以内に問題が解決しない場合はデータセンター管理者にメッセージを送信する）。

4 Zabbix エージェント **2** で **MySQL** を監視する

はじめに

このページでは、MySQL サーバーの基本的な監視を開始するために必要な手順を説明します。

MySQL サーバーを監視するには、Zabbix エージェント、Zabbix エージェント 2、または Open Database Connectivity（ODBC）標準など、いくつかのアプローチがあります。このガイドの主な焦点は、Zabbix エージェント 2 を使用した MySQL サーバーの監視にあります。これは、さまざまなセットアップでシームレスに構成できるため、推奨されるアプローチです。ただし、このページでは他のアプローチについても説明しているので、ご自身の要件に最適なものを自由に選択してください。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、MySQL サーバーの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションを参照してください。

前提条件

このガイドを進める前に、ダウンロードとインストールの手順に従って、Zabbix サーバー、Zabbix フロントエンド、Zabbix エージェント 2 をインストールする必要があります。

セットアップによっては、このガイドの一部の手順が若干異なる場合があります。このガイドは以下のセットアップを前提としています。

- Zabbix バージョン: Zabbix 7.2 PRE-RELEASE（パッケージからインストール）
- OS ディストリビューション: Ubuntu
- OS バージョン: 22.04 (Jammy)
- Zabbix コンポーネント: サーバー、フロントエンド、エージェント 2
- データベース: MySQL
- Web サーバー: Apache

MySQL ユーザーの作成

MySQL サーバーを監視するには、Zabbix が MySQL サーバーおよびそのプロセスにアクセスできる必要があります。MySQL のインストール時に Zabbix のインストール時に作成された「zabbix」ユーザーは、必要なレベルのアクセス権をすでに持っていますが、このユーザーは単純な監視には不要な権限（データベースの DROP、テーブルからの DELETE など）も持っています。したがって、MySQL サーバーの監視のみを目的とした MySQL ユーザーを作成する必要があります。

1. MySQL クライアントに接続し、「zbx_monitor」ユーザーを作成します（「zbx_monitor」ユーザーの *<password>* は任意のパスワードに置き換えてください）。また、GRANTで必要な権限を付与します。

```
mysql -u root -p
# パスワードを入力:

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

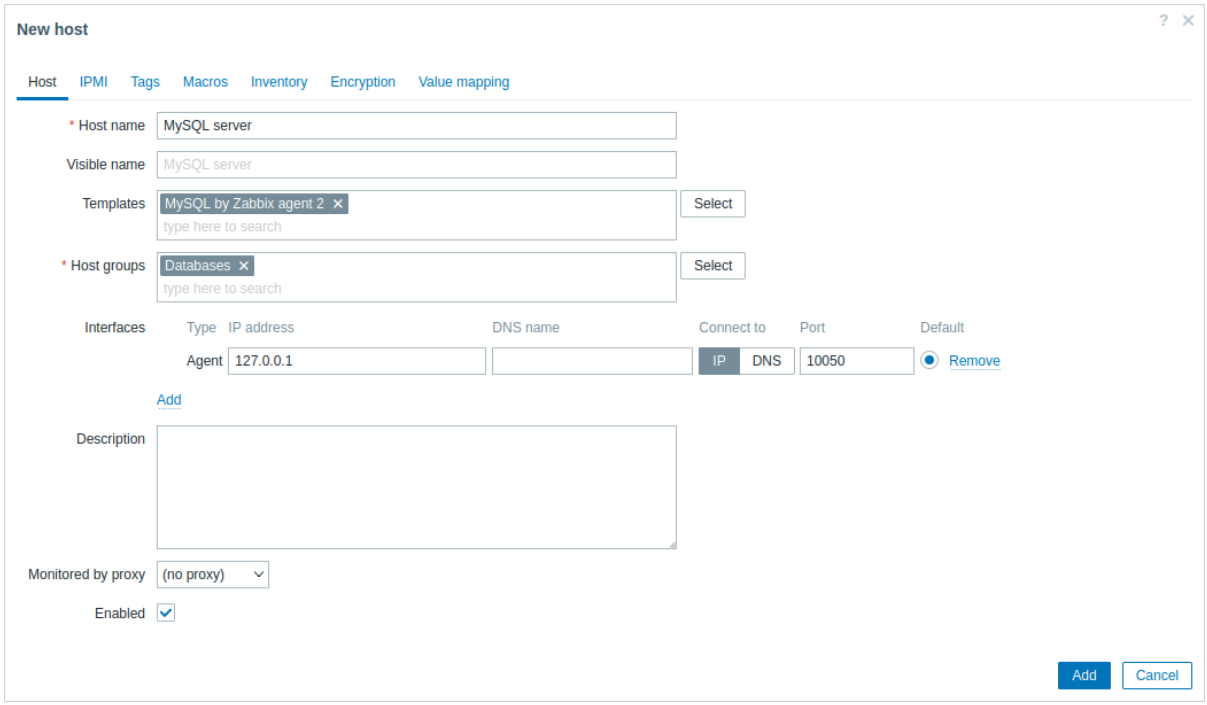
ユーザーが作成できたら、次のステップに進みます。

Zabbix フロントエンドの設定

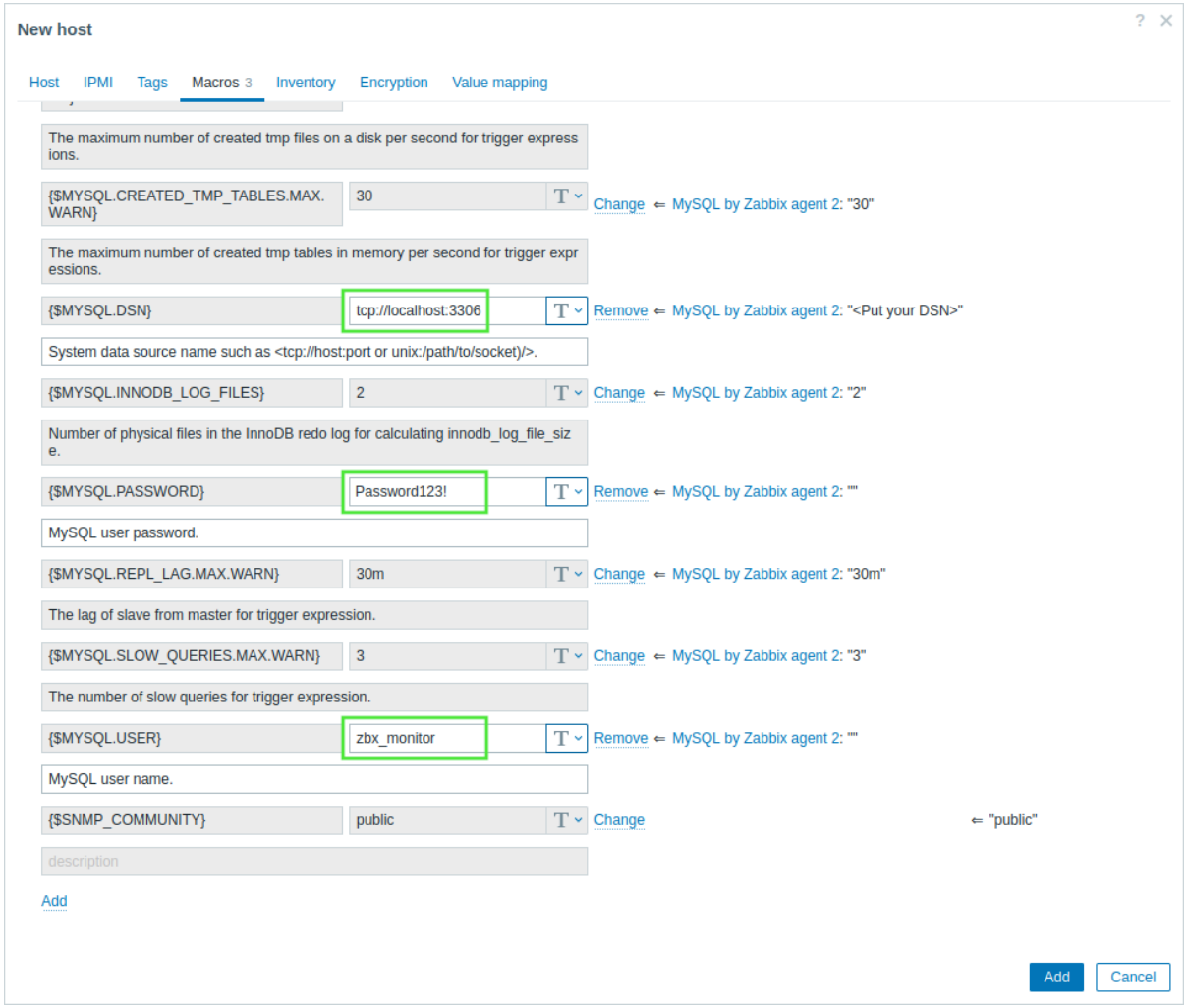
1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します:

- ホスト名フィールドにホスト名（例：”MySQL server”）を入力します。
- テンプレートフィールドに、ホストにリンクするテンプレート「MySQL by Zabbix agent 2」を入力または選択します。
- ホストグループフィールドに、ホストグループ（例：”Databases”）を入力または選択します。
- インターフェースフィールドで、タイプ「エージェント」のインターフェースを追加し、MySQL サーバーの IP アドレスを指定します。このガイドでは、Zabbix サーバーおよび Zabbix agent 2 と同じマシンにインストールされた MySQL サーバーを監視するために「127.0.0.1」(localhost）を使用します。



- マクロタブで継承されたマクロとホストマクロに切り替え、以下のマクロを探してマクロ値の横にある変更をクリックし、値を更新します:
  - {$MYSQL.DSN} - MySQL サーバーのデータソース（MySQL Zabbix agent 2 プラグイン設定ファイルの名前付きセッションの接続文字列）を設定します。このガイドでは、Zabbix サーバーおよび Zabbix agent 2 と同じマシンにインストールされた MySQL サーバーを監視するために、デフォルトのデータソース「tcp://localhost:3306」を使用します。
  - {$MYSQL.PASSWORD} - 以前に作成した MySQL ユーザー「zbx_monitor」のパスワードを設定します。
  - {$MYSQL.USER} - 以前に作成した MySQL ユーザー「zbx_monitor」のユーザー名を設定します。



3. 追加をクリックしてホストを追加します。このホストが MySQL サーバーを表します。

収集されたメトリクスの表示

おめでとうございます！この時点で、Zabbix はすでに MySQL サーバーの監視を開始しています。

収集されたメトリクスを表示するには、監視 → ホストメニューセクションに移動し、ホストの横にあるダッシュボードをクリックします。

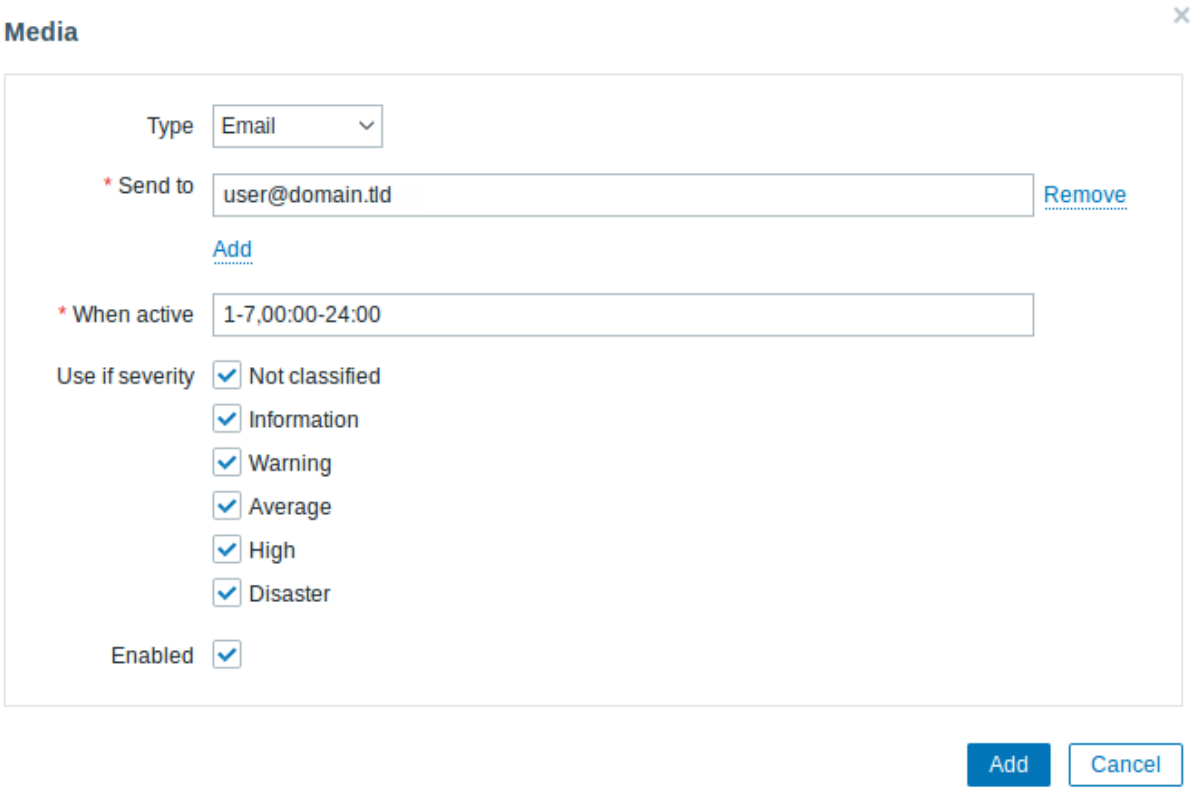
この操作により、MySQL サーバーから収集された最も重要なメトリクスを表示するホストダッシュボード（テンプレートレベルで設定）に移動します。

または、監視 → ホストメニューセクションから、最新データをクリックして、すべての最新の収集メトリクスをリストで表示することもできます。なお、アイテム *MySQL: Calculated value of innodb_log_file_size* は、値が直近 1 時間のデータから計算されるため、データがないことが想定されます。

問題アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの問題について通知できます。このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 → プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信のガイドに従ってください。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるはずです。

設定をテストする

設定をテストするには、Zabbix フロントエンドでホスト設定を更新して実際の問題をシミュレートできます。

1. Zabbix で MySQL サーバーホストの設定を開きます。

2. マクロタブに切り替え、継承およびホストマクロを選択します。

3. 例えば、前に設定した {$MYSQL.USER} マクロ値の横にある変更をクリックし、別の MySQL ユーザー名を設定します。

4. 更新をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は”MySQL: Service is down” という問題を検出します。これは、MySQL サーバーに接続できなくなるためです。問題は監視 → 問題に表示されます。

アラートが設定されている場合は、問題の通知も受け取ります。

6. マクロ値を元の値に戻して問題を解決し、MySQL サーバーの監視を続けてください。

MySQL を監視する他のアプローチ

Zabbix agent 2 で MySQL サーバーを監視する代わりに、Zabbix agent や Open Database Connectivity（ODBC）標準を使用することもできます。Zabbix agent 2 の使用が推奨されていますが、Zabbix agent 2 をサポートしていない、またはカスタムアプローチが必要なセットアップもあるかもしれません。

Zabbix agent と ODBC の主な違いはデータ収集方法にあります。Zabbix agent は MySQL サーバーに直接インストールされ、組み込み機能を使用してデータを収集しますが、ODBC は ODBC ドライバーを使用して MySQL サーバーへの接続を確立し、SQL クエリを使用してデータを取得します。

多くの設定手順は Zabbix agent 2 で MySQL サーバーを監視する場合と似ていますが、いくつか重要な違いがあります。MySQL サーバーを監視できるように Zabbix agent または ODBC を設定する必要があります。以下の手順では、これらの違いについて説明します。

Zabbix エージェントで MySQL を監視する

Zabbix エージェントで MySQL サーバーを監視するには、OS の手順に従ってZabbix サーバー、Zabbix フロントエンド、Zabbix エージェントをダウンロードしてインストールする必要があります。

必要な Zabbix コンポーネントを正常にインストールしたら、*MySQL ユーザーの作成*セクションで説明されているように MySQL ユーザーを作成する必要があります。

MySQL ユーザーを作成したら、Zabbix エージェントが MySQL サーバーへの接続を確立し、監視できるように Zabbix エージェントを設定する必要があります。これには、カスタムエージェントチェックを実行するための複数のユーザーパラメータの設定や、前述の「zbx_monitor」ユーザーとして MySQL サーバーに接続するために必要な認証情報を Zabbix エージェントに提供することが含まれます。

Zabbix エージェントの設定

1. Zabbix エージェントの追加設定ディレクトリに移動します。

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

> **Attention:**
> Zabbix エージェントの追加設定ディレクトリは、Zabbix エージェントの設定ファイル（*zabbix_agentd.conf*）と同じディレクトリにある必要があります。OS や Zabbix のインストール方法によっては、このディレクトリの場所が本ガイドで指定したものと異なる場合があります。デフォルトの場所については、Zabbix エージェントの設定ファイル内の`Include`パラメータを確認してください。

MySQL サーバーの監視に必要なユーザーパラメータをすべて Zabbix エージェントの設定ファイルで定義する代わりに、これらのパラメータを追加設定ディレクトリ内の別ファイルで定義します。

2. Zabbix エージェントの追加設定ディレクトリに *template_db_mysql.conf* ファイルを作成します。

```
vi template_db_mysql.conf
```

3. *template_db_mysql.conf*ファイル（Zabbix リポジトリ内）から内容をコピーし、作成した *template_db_mysql.conf* ファイルに貼り付けて保存します。

4. Zabbix エージェントを再起動して設定を更新します。

```
systemctl restart zabbix-agent
```

Zabbix エージェントのユーザーパラメータを設定したら、Zabbix エージェントが MySQL サーバーにアクセスできるようにするための認証情報の設定に進みます。

5. Zabbix エージェントのホームディレクトリに移動します（システムに存在しない場合は作成する必要があります。デフォルト：*/var/lib/zabbix*）。

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

6. Zabbix エージェントのホームディレクトリに*.my.cnf* ファイルを作成します。

```
vi .my.cnf
```

7. 以下の内容を*.my.cnf* ファイルにコピーします（*<password>* は”zbx_monitor” ユーザーのパスワードに置き換えてください）。

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

Zabbix フロントエンドの設定と設定のテスト

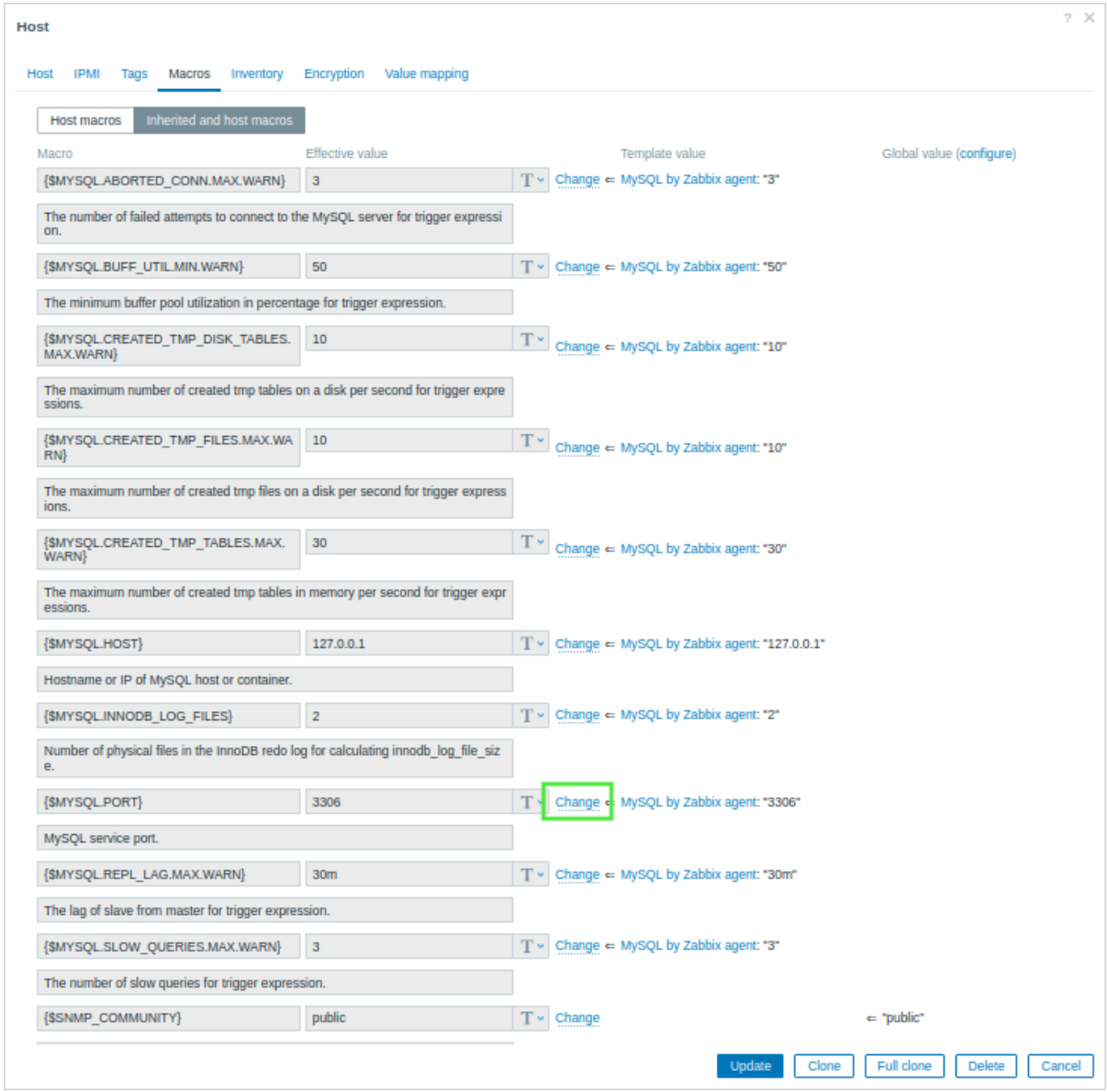
Zabbix フロントエンドを設定するには、*Zabbix* フロントエンドの設定セクションの手順に従い、以下の調整を行ってください。

- テンプレートフィールドで、ホストにリンクされるテンプレート「MySQL by Zabbix agent」を入力または選択します。
- マクロの設定は不要です。

Zabbix フロントエンドの設定が完了したら、収集されたメトリクスの表示や問題アラートの設定ができます。

設定をテストするには、設定のテストセクションの手順に従い、以下の調整を行ってください。

- MySQL サーバーホストの設定の継承されたホストマクロセクションで、{$MYSQL.PORT} マクロ値の横にある変更をクリックし、別のポート（例：「6033」）を設定します。



ODBC で MySQL を監視する

ODBC で MySQL サーバーを監視するには、Zabbix サーバーと Zabbix フロントエンドをダウンロードしてインストールする必要があります。

必要な Zabbix コンポーネントを正常にインストールしたら、*MySQL ユーザーの作成*セクションで説明されているように MySQL ユーザーを作成する必要があります。

MySQL ユーザーを作成したら、ODBC をセットアップする必要があります。これには、最も一般的に使用されているオープンソースの ODBC API 実装の 1 つであるunixODBCと unixODBC ドライバーのインストール、および ODBC ドライバー設定ファイルの編集が含まれます。

ODBC の設定

1. unixODBC をインストールします。unixODBC の推奨インストール方法は、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用することです。

```
apt install unixodbc
```

2. MariaDB unixODBC データベースドライバをインストールします。MySQL データベースを使用している場合でも、互換性の問題から MariaDB unixODBC ドライバを使用します。

```
apt install odbc-mariadb
```

3. ODBC 設定ファイル *odbcinst.ini* および *odbc.ini* の場所を確認します。

```
odbcinst -j
```

このコマンドの実行結果は、次のようになります。

```
unixODBC 2.3.9
```

```
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
...
```

4. MySQL データベースの監視用に ODBC ドライバを設定するには、*odbcinst.ini* ファイルに記載されているドライバ名が必要です。以下の *odbcinst.ini* ファイルの例では、ドライバ名は「MariaDB Unicode」です。

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

5. 以下の内容を *odbc.ini* ファイルにコピーします（*<password>* を「zbx_monitor」ユーザーのパスワードに置き換えてください）。このガイドでは、ODBC ドライバと同じマシンにインストールされた MySQL サーバーを監視するために、MySQL サーバーアドレスとして「127.0.0.1」（localhost）を使用します。データソース名（DSN）「test」に注意してください。これはZabbix フロントエンドの設定時に必要になります。

```
[test]
Driver=MariaDB Unicode
Server=127.0.0.1
User=zbx_monitor
Password=<password>
Port=3306
Database=zabbix
```

Zabbix フロントエンドの設定と設定のテスト

Zabbix フロントエンドを設定するには、*Zabbix* フロントエンドの設定セクションの手順に従い、以下の調整を行ってください。

- テンプレートフィールドには、ホストにリンクされるテンプレート「MySQL by ODBC」を入力または選択します。
- インターフェースの設定は不要です。
- MySQL サーバーホストの設定の継承されたホストマクロセクションの {$MYSQL.DSN} マクロ値は、*odbc.ini* ファイルの DSN 名に設定する必要があります。

Zabbix フロントエンドの設定が完了したら、収集されたメトリクスの表示、問題アラートの設定、設定のテストができます。

こちらもご参照ください

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 問題のエスカレーション - 複数ステップのアラートシナリオを作成する方法 (例: 最初にシステム管理者にメッセージを送信し、45 分以内に問題が解決しない場合はデータセンターマネージャーにメッセージを送信する)。
- ODBC 監視 - 他の Linux ディストリビューションで ODBC をセットアップする方法、および ODBC で追加のデータベース関連メトリクスの監視を開始する方法。
- テンプレート*MySQL by Zabbix agent* - *MySQL by Zabbix agent* テンプレートに関する追加情報。
- テンプレート*MySQL by Zabbix agent 2* - *MySQL by Zabbix agent 2* テンプレートに関する追加情報。
- テンプレート*MySQL by ODBC* - *MySQL by ODBC* テンプレートに関する追加情報。

5 Zabbix で VMware を監視する

はじめに

このページでは、VMware の基本的な監視を開始するために必要な手順について説明します。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、VMware の基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの仮想マシン監視セクションや設定セクションを参照してください。

前提条件

このガイドを進める前に、ダウンロードとインストールの手順に従って、Zabbix サーバーと Zabbix フロントエンドをお使いの OS に合わせてインストールしてください。

このガイドは以下のセットアップを前提としています。

- Zabbix バージョン: 7.2 PRE-RELEASE（パッケージからインストール）

- OS ディストリビューション: Ubuntu
- OS バージョン: 22.04 (Jammy)
- Zabbix コンポーネント: サーバー、フロントエンド、エージェント
- データベース: MySQL
- Web サーバー: Apache

VMware はすでに設定済みであることを前提としています。このガイドでは VMware の設定については説明しません。

Zabbix サーバーの設定

VMware を監視するには、*vmware collector* Zabbix プロセスを有効にする必要があります。VMware 監視の仕組みについては、仮想マシンの監視を参照してください。

1. Zabbix サーバーの設定ファイルを開きます。

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

2. Zabbix サーバーの設定ファイルでStartVMwareCollectorsパラメータを 2以上に設定します（デフォルト値は 0 です）。

```
##### Option: StartVMwareCollectors
###       Number of pre-forked vmware collector instances.
###
### Mandatory: no
### Range: 0-250
### Default:
### StartVMwareCollectors=0

StartVMwareCollectors=2
```

3. Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

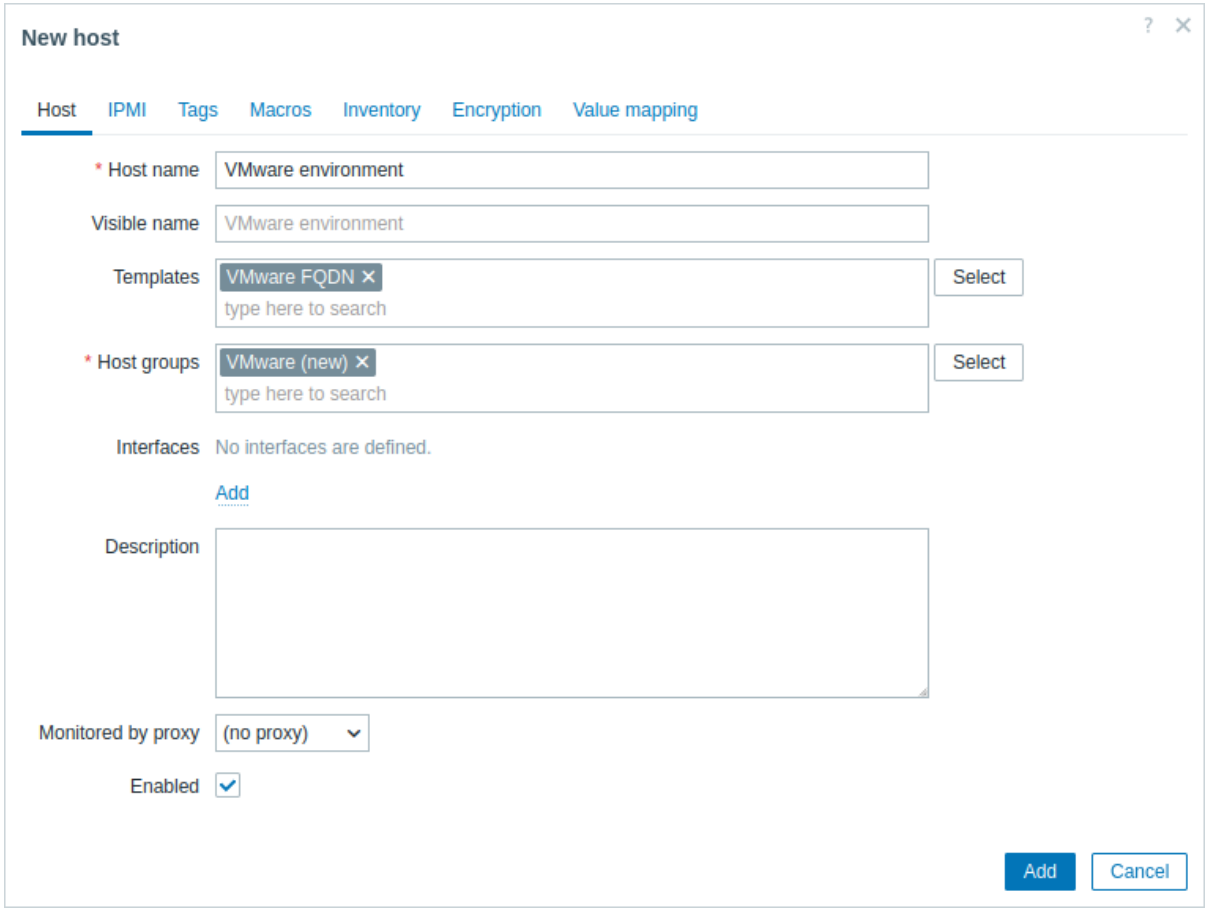
vmware collector プロセスが起動したら、次のステップに進みます。

Zabbix フロントエンドの設定

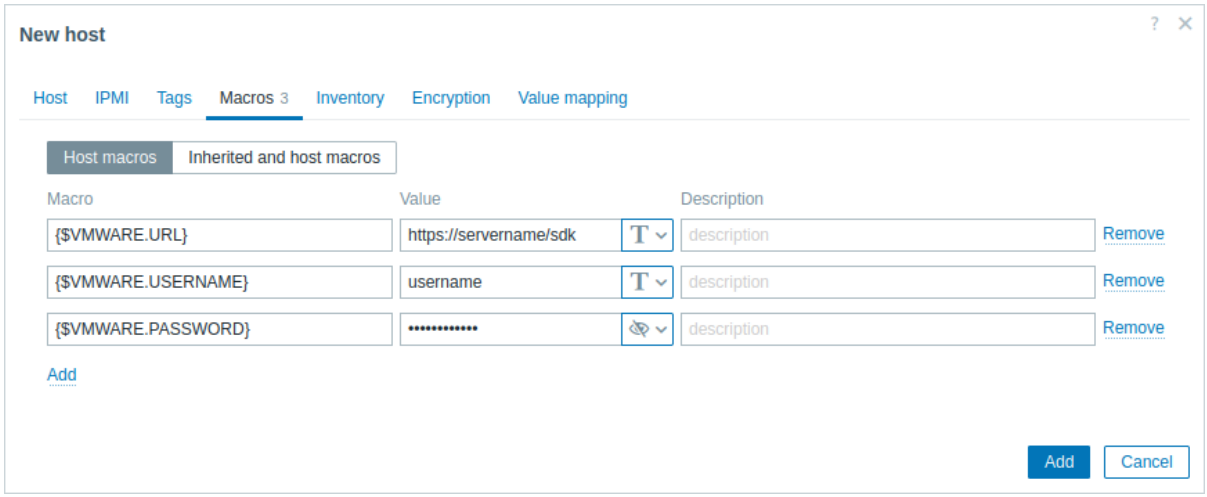
1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

- ホスト名フィールドにホスト名（例：”VMware environment”）を入力します。
- テンプレートフィールドに”VMware FQDN”（または”VMware”）テンプレートを入力または選択します。これらのテンプレートの詳細については、仮想マシンの監視を参照してください。
- ホストグループフィールドにホストグループ（例：新しいホストグループ”VMware”）を入力または選択します。



- マクロタブで、以下のホストマクロを設定します。
  - {$VMWARE.URL} - VMware サービス（vCenter または ESXi ハイパーバイザー）SDK の URL（https://servername/sdk）
  - {$VMWARE.USERNAME} - VMware サービスのユーザー名
  - {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード



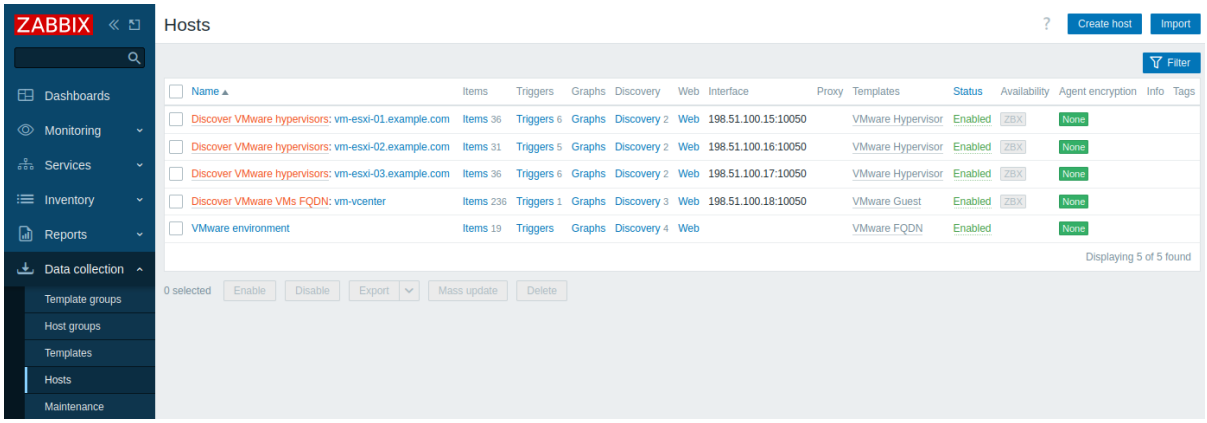
3. 追加ボタンをクリックしてホストを作成します。このホストが VMware 環境を表します。

収集されたメトリクスの表示

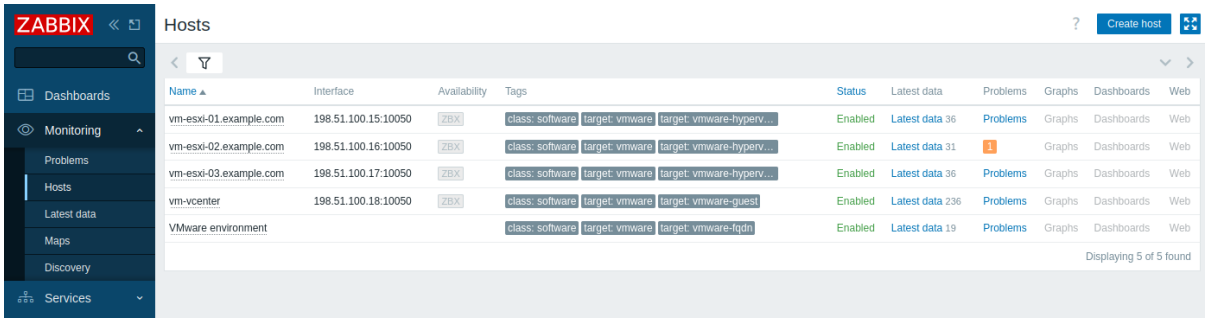
おめでとうございます！この時点で、Zabbix はすでに VMware 環境の監視を開始しています。

VMware 環境の設定によっては、Zabbix が自動検出を行い、検出されたエンティティのホストを作成する場合があります。必要に応じて、ホストの自動検出や作成は手動で実行することもできます。

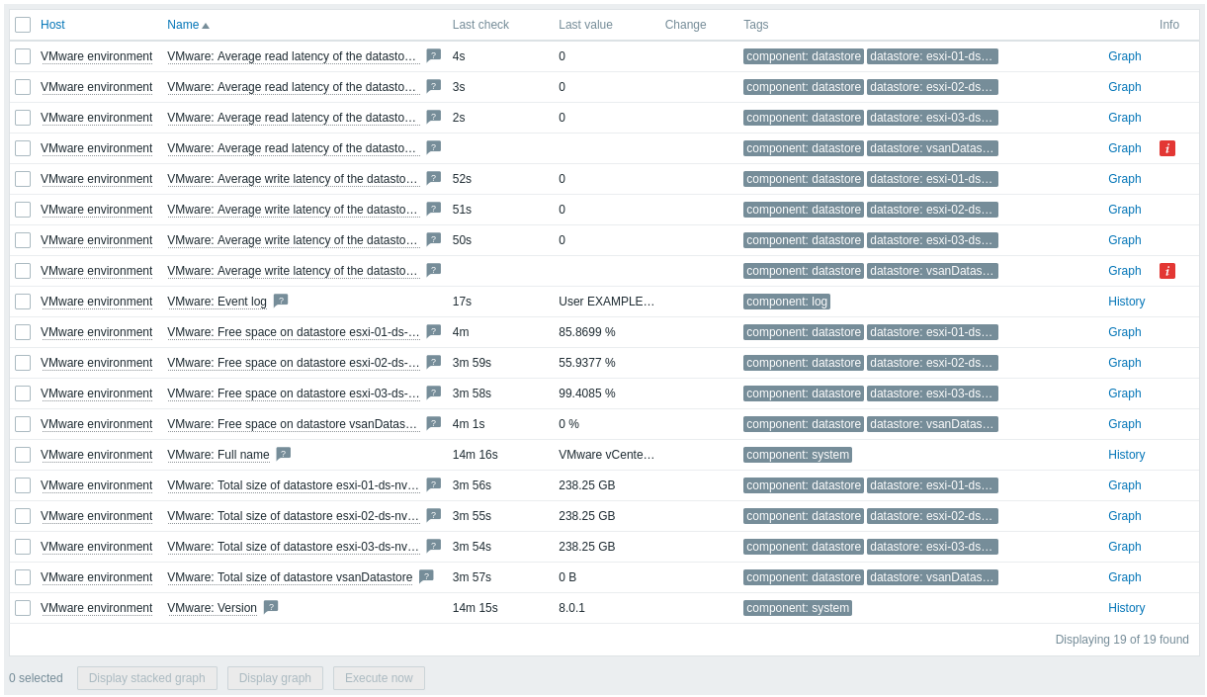
作成されたホストを表示するには、データ収集 → ホストメニューセクションに移動します。



収集されたメトリクスを表示するには、監視 → ホストメニューセクションに移動し、作成された「VMware environment」ホストまたは検出されたエンティティのいずれかのホストの横にある最新データをクリックします。



この操作により、選択したホストから収集されたすべての最新メトリクスのリストが表示されます。

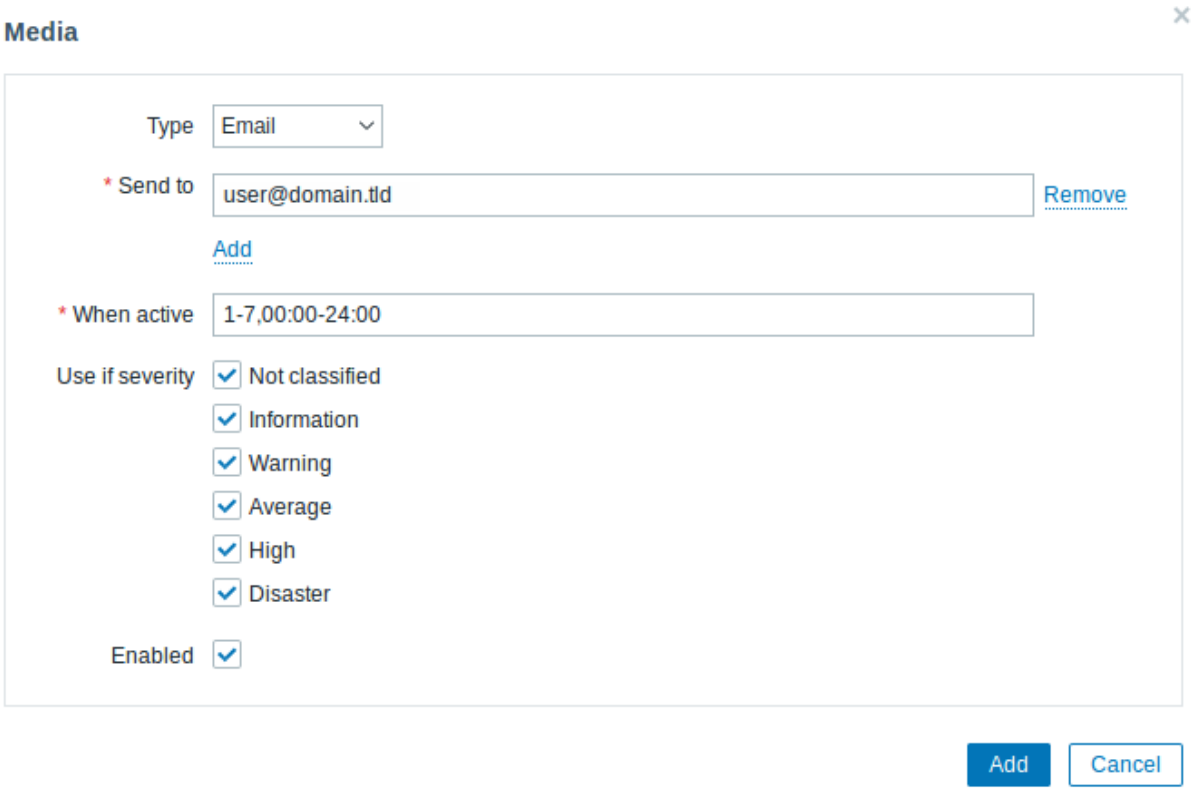


一部のアイテムにはデータがなく、サポートされていませんの状態になっていることに注意してください。これは、監視対象のVMware 環境で有効になっていないため、Zabbix が特定のデータストアで有効なパフォーマンスカウンタを見つけられないためです。

問題アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの問題について通知できます。このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 → プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受け取ることができます。

関連情報

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 問題のエスカレーション - 複数ステップのアラートシナリオの作成方法 (例: 最初にシステム管理者にメッセージを送信し、45分以内に問題が解決しない場合はデータセンター管理者にメッセージを送信)。
- 仮想マシンの監視 - VMware 監視に関する追加情報 (データ収集プロセス、サーバー設定オプション、トラブルシューティングガイドなど)。
- VMware 監視アイテムキー - Zabbix で監視可能な VMware メトリクスの全リスト。
- テンプレート*VMware* - *VMware* テンプレートに関する追加情報。
- テンプレート*VMware FQDN* - *VMware FQDN* テンプレートに関する追加情報。

6 Zabbix でネットワークトラフィックを監視する

はじめに　このページでは、Zabbix を使用してネットワークトラフィックの基本的な監視を開始するために必要な手順について説明します。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、ネットワークトラフィックの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションをご覧ください。

前提条件

このガイドを進める前に、ダウンロードとインストールを行い、OS の指示に従って Zabbix サーバー、Zabbix フロントエンド、Zabbix エージェントをインストールする必要があります。トラフィック監視が必要なマシンに Zabbix エージェントをインストールする必要があることに注意してください。これは、Zabbix サーバーがインストールされているホストと同じホストでも、別のホストでもかまいません。

このガイドでは、*Remote host* という別のマシンの *eth0* インターフェースのネットワークトラフィック監視を設定する手順を説明します。

監視用の **Zabbix** の設定　Zabbix エージェントは、アクティブモードまたはパッシブモード（同時に）でメトリクスを収集できます。詳細については、パッシブおよびアクティブエージェントチェックを参照してください。このガイドでは、パッシブチェックに

よる監視について説明します。

Zabbix エージェントの設定

1. エージェントがインストールされているマシンでエージェントの設定ファイルを開きます（デフォルトでは、パスは*/usr/local/etc/zabbix_agentd.conf* です）:

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

2. *Server* パラメータに Zabbix サーバーの IP アドレスまたは DNS 名を追加します。例:

```
Server=192.0.2.22
```

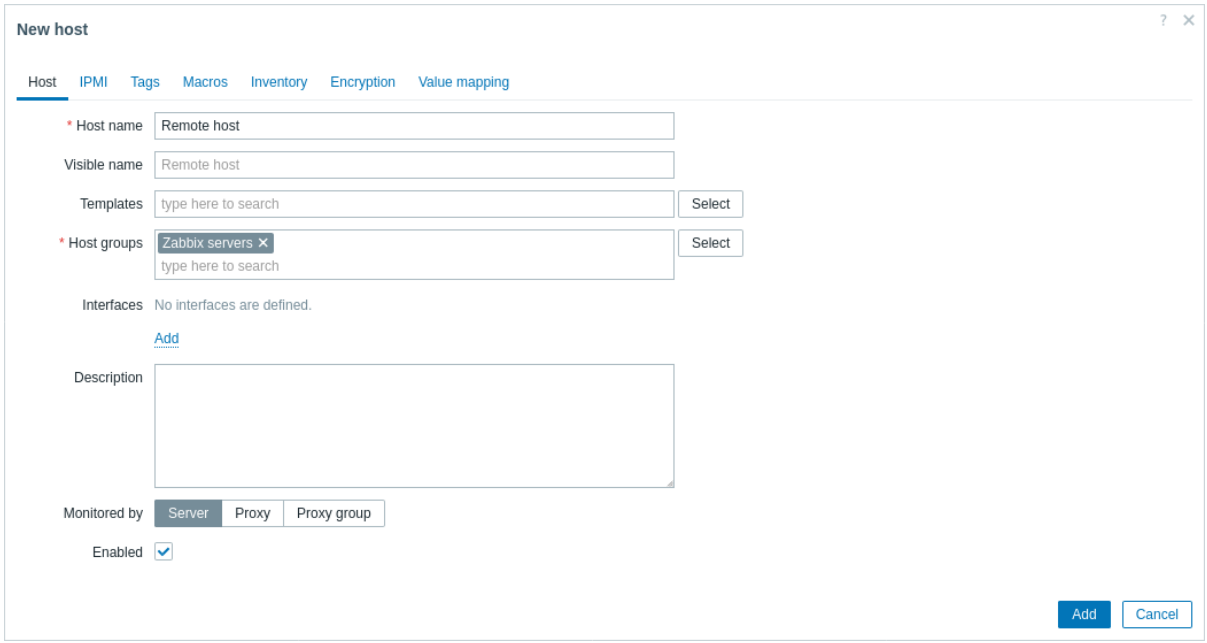
3. Zabbix エージェントを再起動します:

```
systemctl restart zabbix-agent
```

Zabbix フロントエンド

1. Zabbix フロントエンドにログインします。

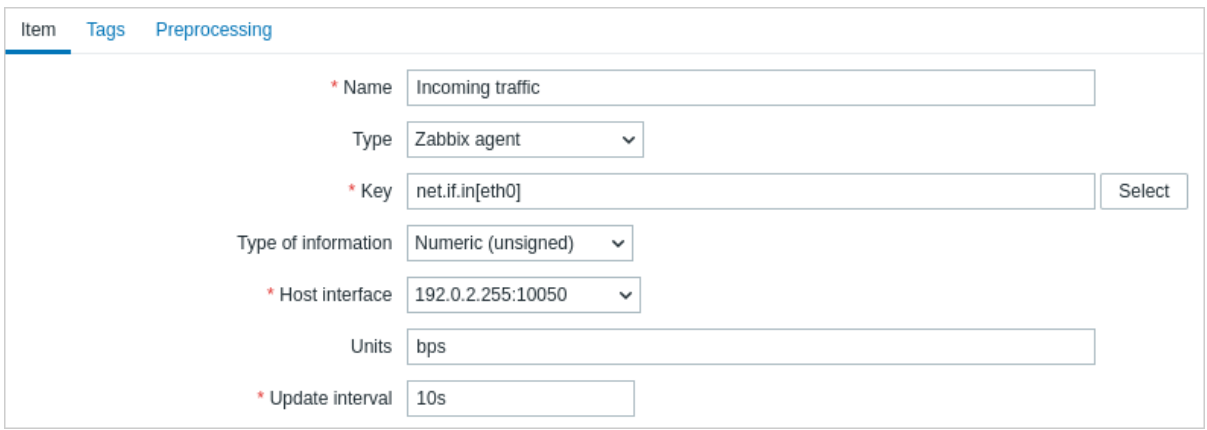
2. エージェントがインストールされているマシンの IP アドレスまたは DNS 名を指定して、Zabbix Web インターフェースでホストを作成します。



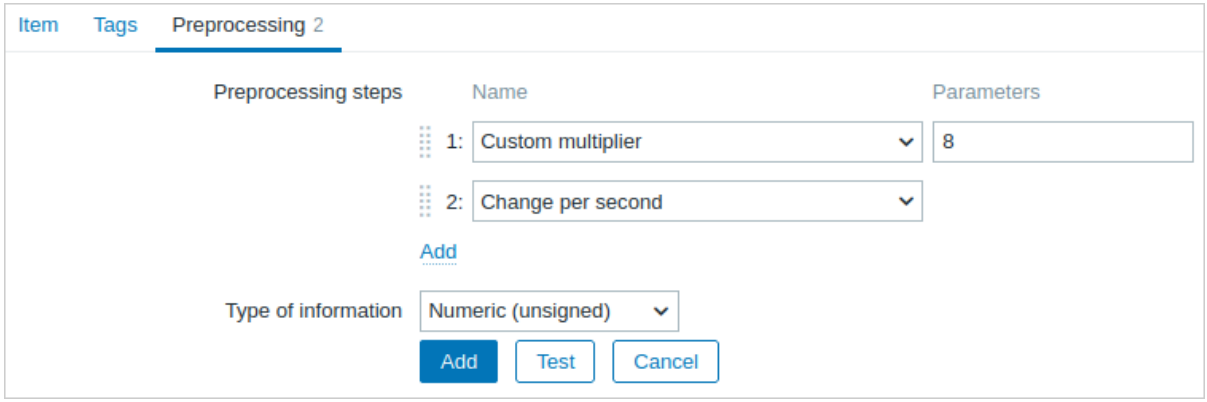
アイテムの作成　アイテムの作成の手順に従って、トラフィック監視用のアイテムを追加します。具体的には以下の通りです。

- 受信トラフィック
- 送信トラフィック
- 合計トラフィック

受信トラフィック監視アイテムの簡単な設定例は以下の通りです。



収集したデータを実用的に利用できるようにするために、アイテム作成時に事前処理のステップを設定できます。この場合、バイトをビットに変換するための 8 倍と、1 秒あたりの変化量としての表示が考えられます。

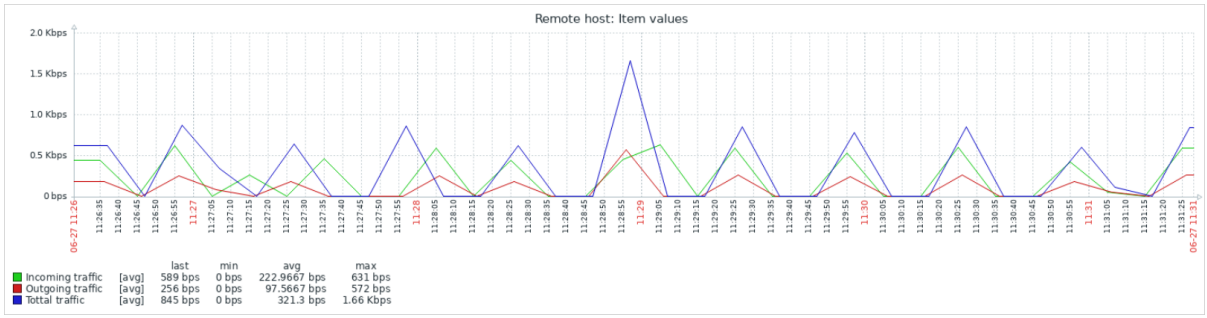


収集したデータの表示　おめでとうございます！この時点で、Zabbix はすでにネットワークトラフィックを監視しています。

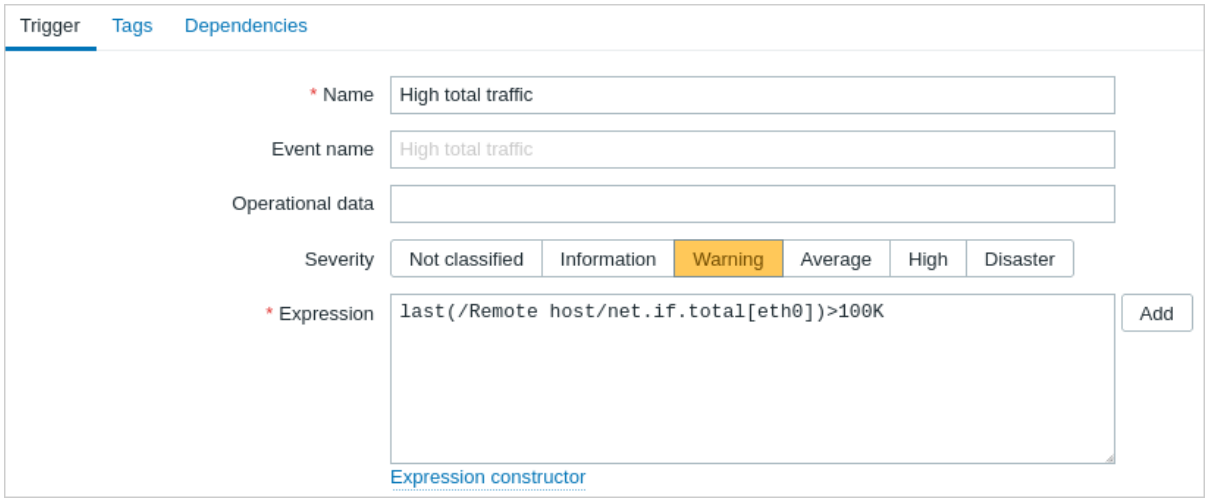
収集したメトリクスを表示するには、監視 → ホスト メニューセクションを開き、ホストの行の最新データをクリックします。

トラフィックデータが表示されます。

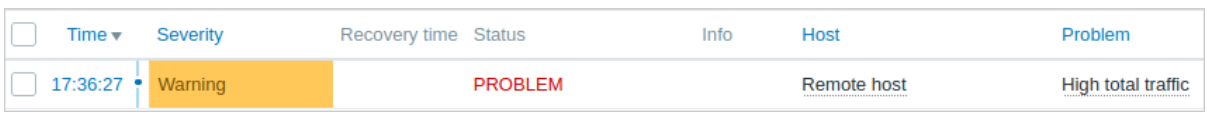
グラフの表示　収集されたデータはグラフとして表示できます。これらを表示するには、最新データセクションでアイテムの行のグラフをクリックするか、必要なアイテムを選択して下のグラフを表示をクリックします。



トリガーの設定　異常なネットワークトラフィックを検出するためにトリガーを設定できます。トリガーの設定の手順を参照し、例えば合計トラフィックが高すぎることを示すトリガーを追加します。



トリガー式で設定したしきい値を超えるトラフィックを発生させ、監視データ → 障害に移動して、障害がリストされていることを確認します。



問題アラートの設定　問題に関する通知を受け取る方法はいくつかあります。最も一般的なのはメールですが、メールによる問題通知の設定手順に従ってください。また、通知の配信に使用するメディアタイプを他にも選択できます。

関連情報:

- 障害のエスカレーション - 複数ステップのアラートシナリオの作成方法 (例: 最初にシステム管理者にメッセージを送信し、45分以内に問題が解決しない場合はデータセンター管理者にメッセージを送信する)。
- 障害の認識 - 問題が既知であることを示し、解決に関するコメントを追加し、問題を抑制またはクローズする方法。
- Zabbix エージェントによる Linux の監視 - 事前設定済みテンプレートをリンクして、最も重要なアイテムの基本的な監視を開始する方法。

7 アクティブチェックを使用してネットワークトラフィックを監視する

はじめに　このページでは、アクティブチェックを使用して Zabbix でネットワークトラフィックの基本的な監視を開始するために必要な手順を説明します。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、アクティブチェックを使用してネットワークトラフィックの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションを参照してください。

前提条件

このガイドを進める前に、ダウンロードとインストールの手順に従って、Zabbix サーバー、Zabbix フロントエンド、Zabbix エージェントをインストールする必要があります。トラフィック監視が必要なマシンに Zabbix エージェントをインストールする必要があることに注意してください。これは、Zabbix サーバーがインストールされているホストと同じホストでも、別のホストでもかまいません。

このガイドでは、*Remote host* という別のマシン上の *eth0* インターフェースのネットワークトラフィック監視を設定する手順を説明します。

監視のための **Zabbix** の設定　Zabbix エージェントは、アクティブモードまたはパッシブモード（同時に）でメトリクスを収集できます。詳細については、パッシブおよびアクティブエージェントチェックを参照してください。このガイドでは、アクティブチェックによる監視について説明します。

Zabbix エージェントの設定

1. エージェントがインストールされているマシンで、エージェントの設定ファイルを開きます。

Zabbix agent を使用している場合:

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

Zabbix agent 2 を使用している場合:

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（およびオプションのポート）を ServerActive パラメータに追加します。例:

```
ServerActive=192.0.2.0:10051
```

Zabbix エージェントはこのアドレスを使用して、Zabbix サーバーの trapper ポート（デフォルト: 10051）に接続し、アクティブチェック用の設定データを要求します。

3. Hostname パラメータを定義します。これはZabbix フロントエンドで定義するホスト名と一致している必要があります。例では次のようになります:

```
Hostname=Remote host
```

Hostname の値は一致している必要があります。なぜなら、アクティブチェックの場合、Zabbix エージェントはこの値を使用してサーバーから正しいホスト設定を取得するためです。具体的には、エージェントがサーバーへの接続を開始し、Hostname の値で自身を識別します。その後、サーバーはそのホストの監視設定を提供します。これらの値が異なる場合、エージェントは適切な設定を受け取れず、メトリクスが取得できなかったり、監視に問題が発生したりします。

4. Zabbix エージェントを再起動します。

Zabbix agent を使用している場合:

```
systemctl restart zabbix-agent
```

Zabbix agent 2 を使用している場合:

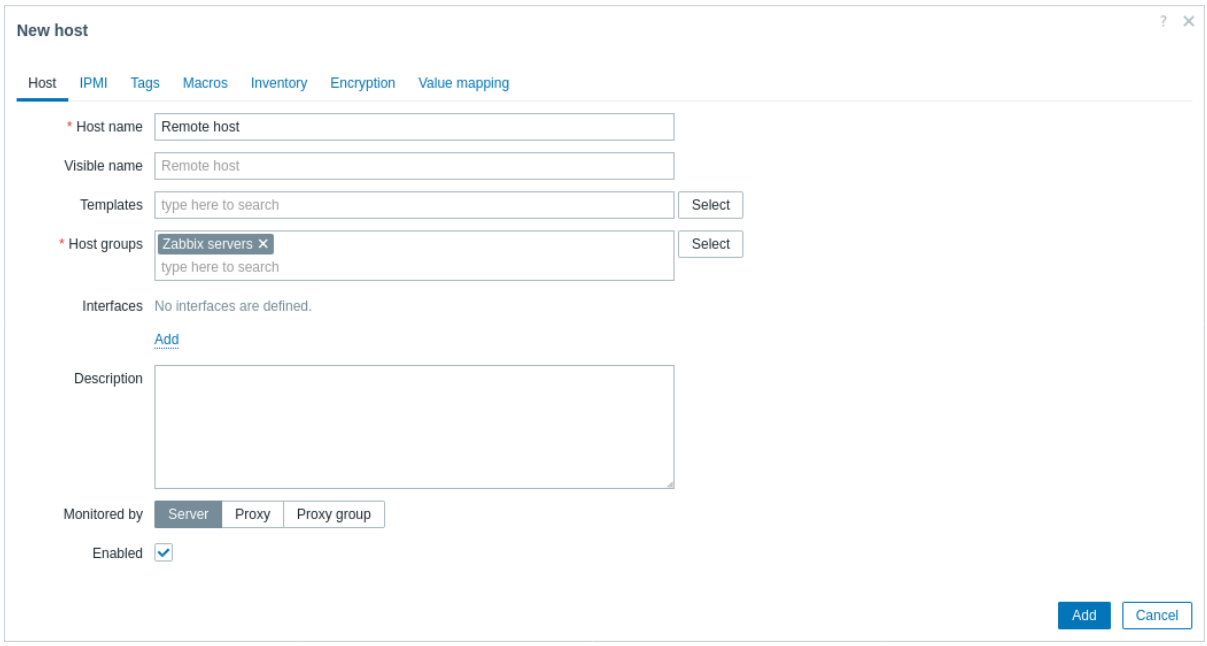
```
systemctl restart zabbix-agent2
```

Zabbix フロントエンド

1. Zabbix フロントエンドにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

- ホスト名フィールドには、エージェントの設定ファイルで前述した Hostname パラメータの値と一致するホスト名（例：”Remote host”）を入力します。
- ホストグループフィールドには、ホストグループ（例：”Zabbix servers”）を入力または選択します。
- インターフェースは未定義のままにします。アクティブチェックの場合、エージェントがサーバーへの接続を開始するため、サーバーからの接続を待ち受ける必要がなく、インターフェースは必要ありません。

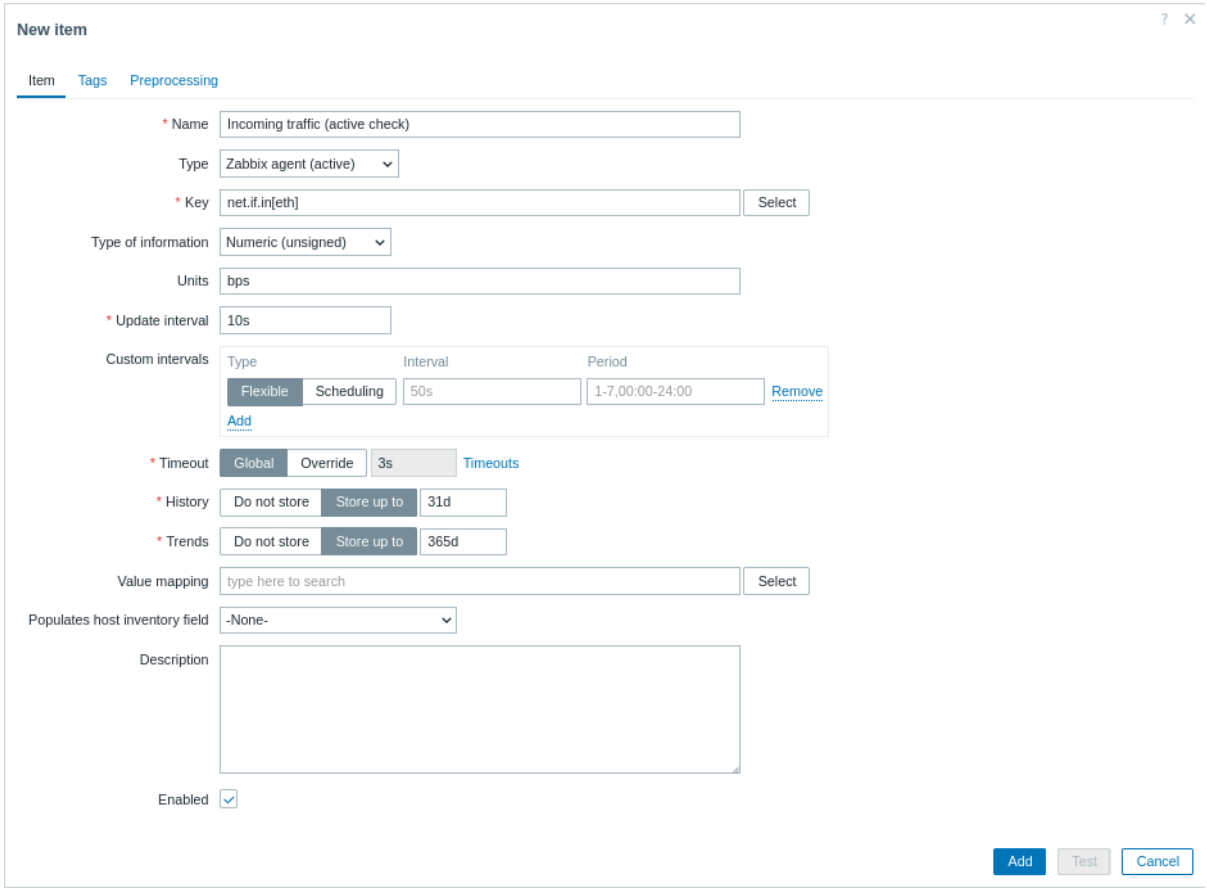


3. 追加をクリックしてホストを追加します。このホストは監視対象の Linux マシンを表します。

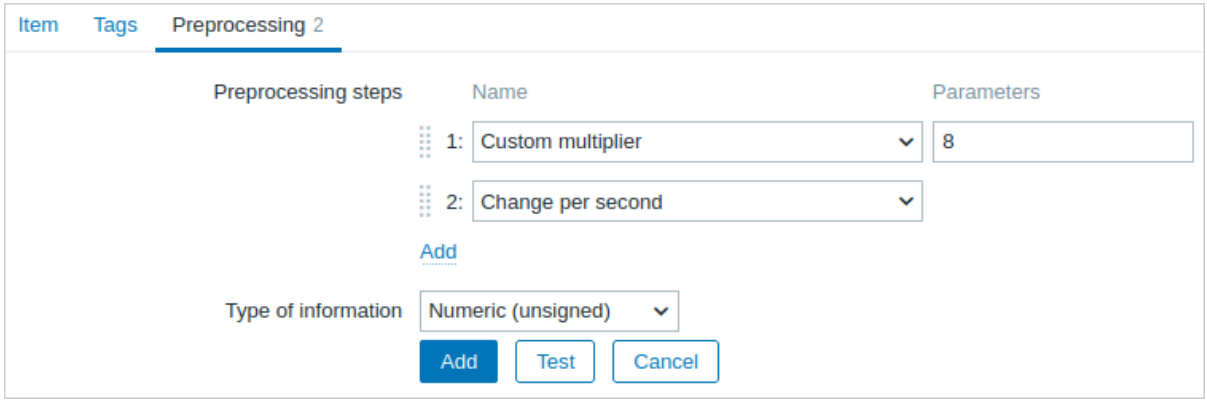
アイテムの作成　アイテムの作成の手順に従って、トラフィック監視用のアイテムを追加します。具体的には以下の通りです。

- 受信トラフィック
- 送信トラフィック
- 合計トラフィック

アクティブチェックによる受信トラフィック監視アイテムのシンプルな設定例は以下の通りです。

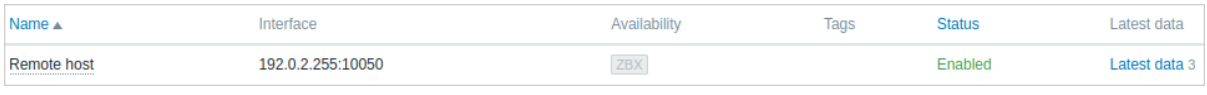


収集したデータを実用的に利用できるようにするため、アイテム作成時に事前処理 を設定することができます。この場合、バイトをビットに変換するための 8 倍の乗算と、1 秒あたりの変化量としての表示が考えられます。

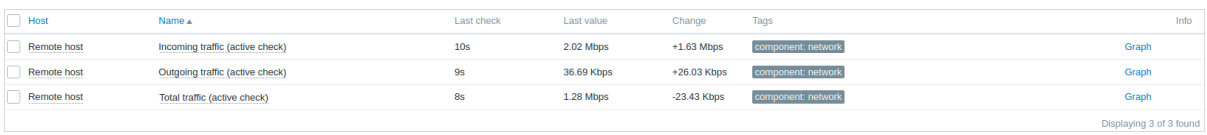


収集したデータの表示　おめでとうございます！この時点で、Zabbix はすでにネットワークトラフィックを監視しています。

収集したメトリクスを表示するには、監視 → ホストメニューセクションを開き、ホストの行の最新データをクリックします。



トラフィックデータが表示されます。



関連情報:

- グラフの表示 - 収集したデータをグラフとして表示する方法
- トリガーの設定 - 異常なネットワークトラフィックを検出するためのトリガーの設定方法
- 障害アラートの設定 - 障害発生時に通知を受け取るための設定方法
- 障害のエスカレーション - 複数段階のアラートシナリオの作成方法 (例: まずシステム管理者にメッセージを送信し、45 分以内に障害が解決しない場合はデータセンター管理者にメッセージを送信)

- 障害の認識 - 障害が既知であることの表示、解決に関するコメントの追加、障害の抑制やクローズの方法
- Zabbix エージェントによる Linux の監視 - あらかじめ設定されたテンプレートをリンクして、最も重要な項目の基本的な監視を開始する方法
- パッケージによるインストール - さまざまな Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールし、最新の機能やバグ修正にアクセスする方法

8 ブラウザアイテムでウェブサイトの監視

はじめに

このページでは、ブラウザアイテムを使用してウェブサイトの基本的な監視を開始するために必要な手順について説明します。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、Browser アイテムで Web サイトの基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定をお探しの場合は、Browser アイテムのページや Zabbix マニュアルの設定セクションをご覧ください。

前提条件

このガイドを進める前に、お使いの OS の手順に従って、Zabbix サーバーと Zabbix フロントエンドをダウンロードしてインストールする必要があります。

このガイドは以下のセットアップに基づいています。

- Zabbix バージョン: 7.2 PRE-RELEASE（パッケージからインストール）
- OS ディストリビューション: Ubuntu
- OS バージョン: 22.04 (Jammy)
- Zabbix コンポーネント: サーバー、フロントエンド、エージェント
- データベース: MySQL
- Web サーバー: Apache

WebDriver の設定

ブラウザーアイテムには、Web テストのエンドポイントとして自動化フレームワーク（Selenium Server またはプレーンな WebDriver、たとえば ChromeDriver）が必要です。これにより、ボタンのクリックやテキストの入力などのテストコマンドを実行し、ブラウザーを制御して操作します。このガイドでは例として、Docker コンテナ内で Chrome を使用した Selenium Server を使用します。

Docker がすでに設定されていることを前提としています。このガイドでは Docker の設定については説明しません。インストール手順については、Install Docker Engine on Ubuntuを参照してください。

1. 次のオプションで Docker コンテナ内で Chrome を使用した Selenium Server を起動します。

- **docker run --name browser** - ”browser” という名前の新しい Docker コンテナを実行します。
- **-p 4444:4444** - ホストマシンのポート 4444 をコンテナのポート 4444 にマッピングします（これは Selenium Server がコマンドを受け付けるために使用するポートです）。
- **-p 7900:7900** - ホストマシンのポート 7900 をコンテナのポート 7900 にマッピングします（これは Virtual Network Computing（VNC）サーバーが使用するポートで、VNC クライアントが必要ですが、リモートでブラウザーの GUI を表示できます）。
- **--shm-size=”2g”** - コンテナに 2GB の共有メモリを割り当てます（Chrome が正常に動作するために重要で、クラッシュを回避するために大量の共有メモリが必要になる場合があります）。
- **-d** - コンテナをデタッチドモードで実行します。つまり、バックグラウンドで実行されます。
- **selenium/standalone-chrome:latest** - 使用する Docker イメージを指定します。この場合は、Selenium Server with Chromeの最新バージョンです。

```
docker run --name browser \
-p 4444:4444 \
-p 7900:7900 \
--shm-size="2g" \
-d selenium/standalone-chrome:latest
```

2. `browser` Docker コンテナが実行中でアクセス可能であることを確認します。

- コンテナの IP アドレスを取得します（この例では 192.0.2.1）：

```
ip addr

### 1: lo: <LOOPBACK,UP,LOWER_UP>
```

```
###     ...
### 3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> ...
###     inet 192.0.2.1/16 brd 192.0.255.255 scope global docker0
###     ...
```

- Ncatでコンテナへの接続をテストします:

```
nc -zv 192.0.2.1 4444

### Connection to 192.0.2.1 4444 port [tcp/*] succeeded!
```

- curlで Selenium Server から Web ページの内容を取得します:

```
curl -L 192.0.2.1:4444

### <!DOCTYPE html>
### <html lang="en">
###
### <head>
###   <meta charset="utf-8"/>
###   <link href="favicon.svg" rel="icon" type="image/svg">
###   <meta content="width=device-width, initial-scale=1" name="viewport"/>
###   <link href="logo192.png" rel="apple-touch-icon"/>
###   <link href="manifest.json" rel="manifest"/>
###   <title>Selenium Grid</title>
### </head>
###
### <body>
### ...
```

Note:
トラブルシューティングについては、Docker documentationを参照してください。

Zabbix サーバーの設定

ブラウザーアイテムは、*browser poller* Zabbix プロセスによって実行および処理されます。これらのプロセスは、StartBrowserPollersサーバー設定パラメーターを調整して有効にする必要があります。さらに、WebDriverURLパラメーターで、前述の Web テストエンドポイントを指定する必要があります。

デフォルトでは、StartBrowserPollers パラメーターは 1 に設定されているため、Web テストエンドポイントを指定するだけで済みます。

1. Zabbix サーバーの設定ファイルを開きます。

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

2. Zabbix サーバーの設定ファイルでWebDriverURLパラメーターを探して設定します。

```
##### Option: WebDriverURL
###WebDriver インターフェースの HTTP[S] URL。たとえば、Selenium WebDriver スタンドアロン サーバーで使用される http://
###
### 必須: いいえ
### デフォルト:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

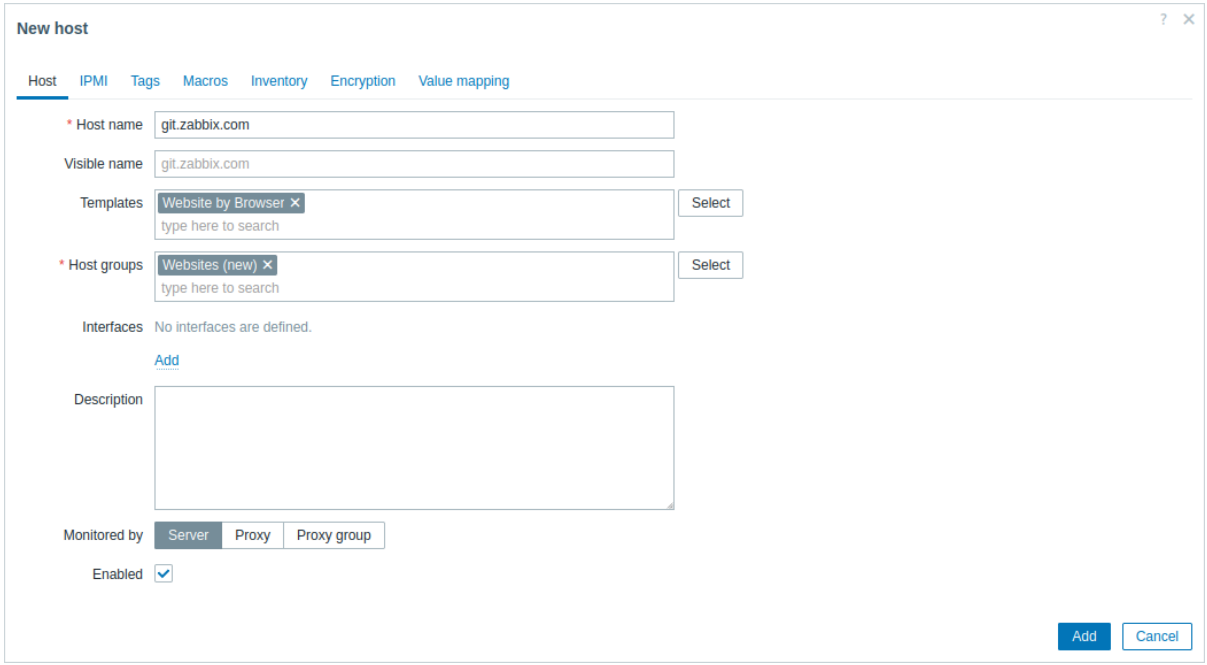
Zabbix フロントエンドの設定

1. Zabbix フロントエンドにログインします。

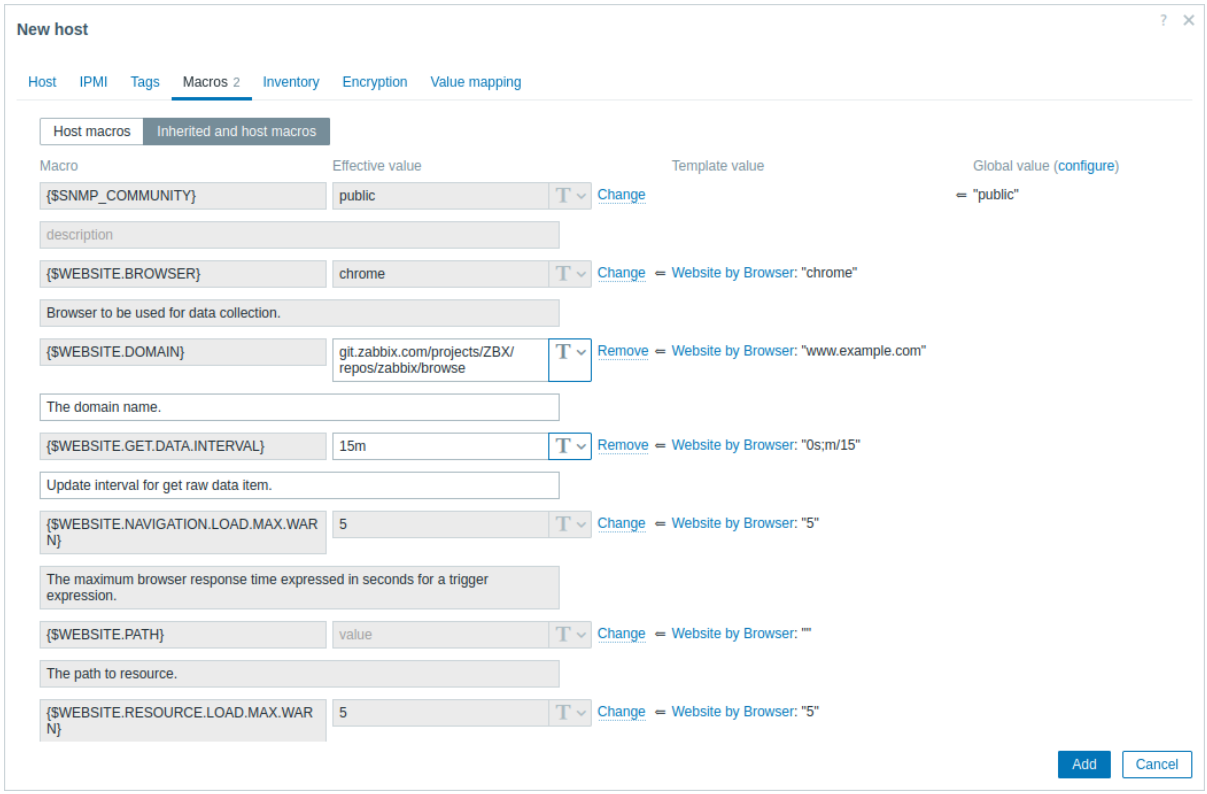
2. Zabbix の Web インターフェースでホストを作成します:

- ホスト名フィールドにホスト名（例：”git.zabbix.com”）を入力します。
- テンプレートフィールドに「Website by Browser」テンプレートを入力または選択します。このテンプレートの詳細については、Website by Browserを参照してください。

- ホストグループフィールドにホストグループ（例：新しいホストグループ「Websites」）を入力または選択します。



- マクロタブで継承およびホストマクロに切り替え、以下のマクロを探してマクロ値の横にある変更をクリックし、値を更新します:
  - {$WEBSITE.DOMAIN} - ドメイン名（例：git.zabbix.com/projects/ZBX/repos/zabbix/browse）
  - {$WEBSITE.GET.DATA.INTERVAL} - アイテムデータの更新間隔（例：15m）



3. 追加ボタンをクリックしてホストを作成します。このホストは監視したい Web サイトを表します。

収集されたメトリクスの表示

おめでとうございます！この時点で、Zabbix はすでに指定した Web サイトを監視しています。

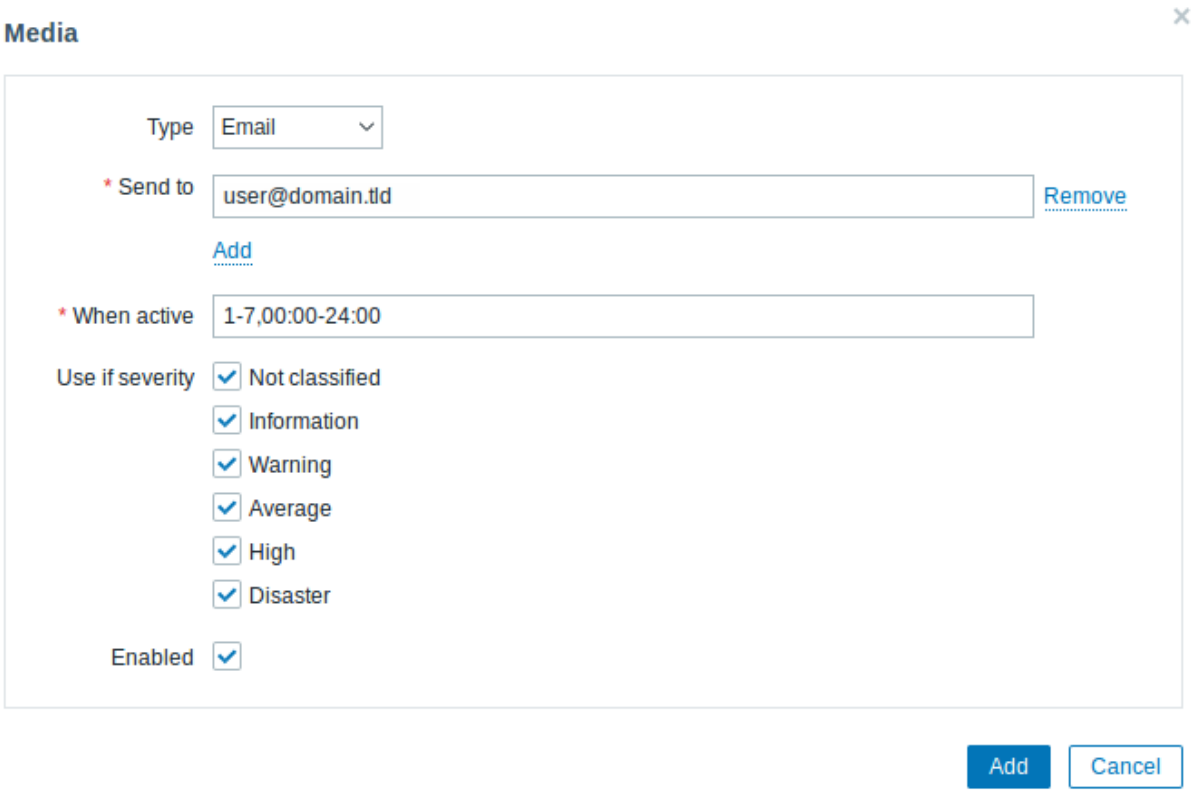
収集されたメトリクスを表示するには、監視 → ホストメニューセクションに移動し、ホストの横にあるダッシュボードをクリックします。

この操作により、Web サイトから収集された最も重要なメトリクスを含むホストダッシュボード（テンプレートレベルで設定）に移動します。

問題アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの問題について通知できます。このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 → プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるはずです。

設定をテストする

設定をテストするために、Zabbix フロントエンドでホスト設定を更新して実際の問題をシミュレートできます。

1. Zabbix でウェブサイトホストの設定を開きます。

2. マクロタブに切り替え、継承およびホストマクロを選択します。

3. 例えば以前に設定した {$WEBSITE.DOMAIN} マクロ値の横にある変更をクリックし、誤ったドメイン名（例：/git.zabbix.com/projects/ZBX/rep
を設定します。

4. 更新をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は「要求されたウェブサイトの JSON の取得に失敗しました」という問題を検出します。これは、指定されたウェブサイトに接続できないためです。この問題は監視 → 問題に表示されます。



アラートが設定されている場合は、問題通知も受信します。

6. 問題を解決し、ウェブサイトの監視を継続するには、マクロ値を元の値に戻します。

関連項目

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 問題のエスカレーション - 複数ステップのアラートシナリオの作成方法 (例: 最初にシステム管理者にメッセージを送信し、45分以内に問題が解決しない場合はデータセンターのマネージャーにメッセージを送信)。
- ブラウザーアイテム - ブラウザーアイテムの設定方法。
- テンプレート Website by Browser - *Website by Browser* テンプレートに関する追加情報。

9 Zabbix エージェント 2(パッシブ) で Web サイトの証明書を監視する

はじめに

このガイドでは、Zabbix agent 2 の `web.certificate.get` キーを使用して SSL/TLS 証明書を設定および監視する方法について、包括的な概要を提供します。単一または複数の Web サイトの証明書監視を簡素化し、管理者が期限切れや無効な証明書などの潜在的な問題を迅速に特定できるように設計されています。

このガイドの対象者

このガイドは Zabbix の新規ユーザー向けに作成されており、ウェブサイト証明書の基本的な監視を有効にするために必要な最小限の手順が含まれています。高度なカスタマイズやより詳細な設定が必要な場合は、Zabbix マニュアルの設定セクションを参照してください。

前提条件

このガイドを進める前に、ダウンロードとインストールの手順に従って、Zabbix サーバー、Zabbix フロントエンド、および Zabbix エージェント 2 をお使いの OS にインストールする必要があります。このチュートリアルでは、Zabbix サーバーとエージェントの両方が同じマシンにインストールされていることを前提としています。そのため、設定では `127.0.0.1` が使用されています。

Zabbix agent 2 の設定

1. Zabbix agent の設定ファイル（デフォルトパス: `/etc/zabbix/zabbix_agent2.conf`）を開きます。

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

2. エージェントとサーバーが同じマシンで動作しているため、Server パラメータを 127.0.0.1 に設定します。

```
Server=127.0.0.1
```

3. ファイルを保存し、Zabbix agent 2 サービスを再起動します。

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

4. Zabbix agent 2 のセットアップと設定後、次のコマンドでエージェントの稼働をテストします。

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

Zabbix agent 2 にはデフォルトで WebCertificate プラグインが含まれているため、別途インストールや設定は不要です。

Zabbix フロントエンドの設定

1. Zabbix フロントエンドにログインします。

2. 監視データ > ホストに移動します。

3. ウェブサイト証明書を監視したい既存のホストをクリックするか、必要に応じてホストを作成します。

- ホスト名フィールドにホスト名（例：”Certificate Monitoring”）を入力します。
- テンプレートフィールドに「Website certificate by Zabbix agent 2」と入力または選択し、ホストにリンクします。
- ホストグループフィールドにホストグループ（例：”SSL/TLS Monitoring”）を入力または選択します。
- インターフェースフィールドで「エージェント」タイプのインターフェースを追加し、IP アドレスを指定します。この例では「127.0.0.1」を使用します。
- マクロタブで 継承およびホストマクロに切り替え、以下のマクロを探してマクロ値の横にある 変更をクリックし、値を更新します。
  - `{$CERT.WEBSITE.HOSTNAME}` - 監視したいウェブサイトの DNS 名を値として入力します。

4. 追加をクリックしてホストを追加します。

複数のウェブサイトを監視するには、`{$CERT.WEBSITE.HOSTNAME}` マクロにカンマ区切りでホスト名のリストを指定できます。オプションで、`{$CERT.WEBSITE.PORT}` マクロにカンマ区切りでポートのリストを指定することもでき、各ポートは順番に対応するホスト名に対応します。例えば、example.com と example.org をポート 8443 で監視する場合:

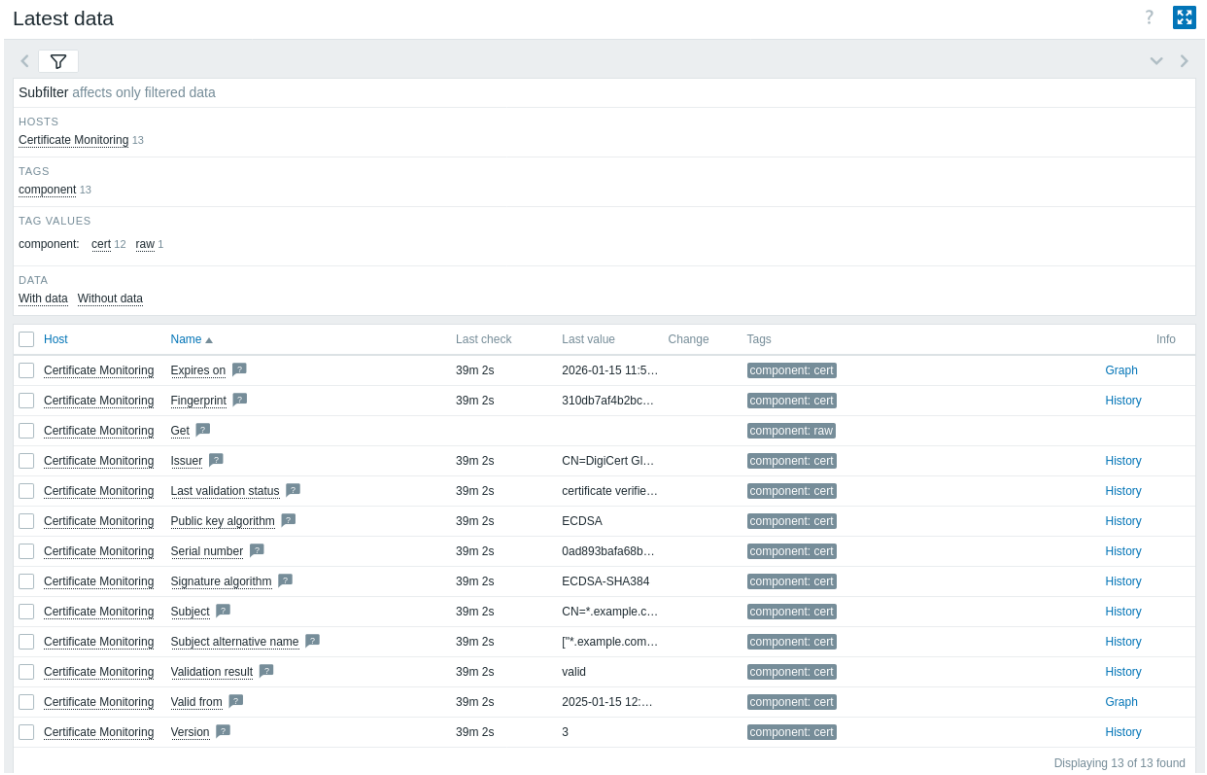
- `{$CERT.WEBSITE.HOSTNAME}`: example.com,example.org
- `{$CERT.WEBSITE.PORT}`: 443,8443

`{$CERT.WEBSITE.HOSTNAME}` マクロで指定した各ウェブサイトごとに、Zabbix は対応するアイテムとトリガーのセットを作成します。これにより、各ウェブサイトの SSL 証明書を個別に監視およびアラートできます。

収集されたメトリクスの表示

おめでとうございます！この時点で、Zabbix はすでに目的の Web 証明書を監視しています。

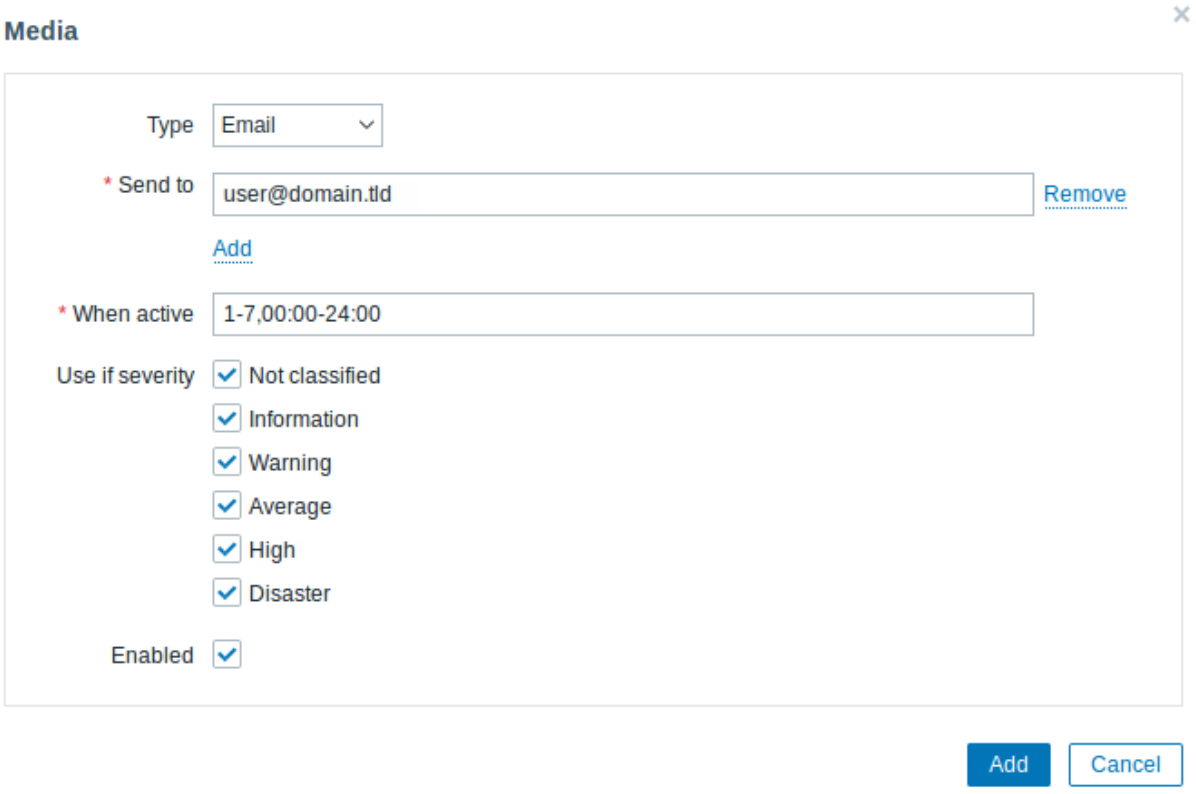
収集されたメトリクスを表示するには、監視 > ホストメニューセクションに移動し、ホストの横にある最新データをクリックして、有効期限、発行者、サブジェクトなど、最新の収集メトリクスをリストで表示します。



問題アラートの設定

Zabbix はインフラストラクチャの問題について通知できます。このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



2. 問題通知の受信のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるはずです。

設定をテストする

設定をテストするために、Zabbix フロントエンドでホスト設定を更新して実際の問題をシミュレートできます。

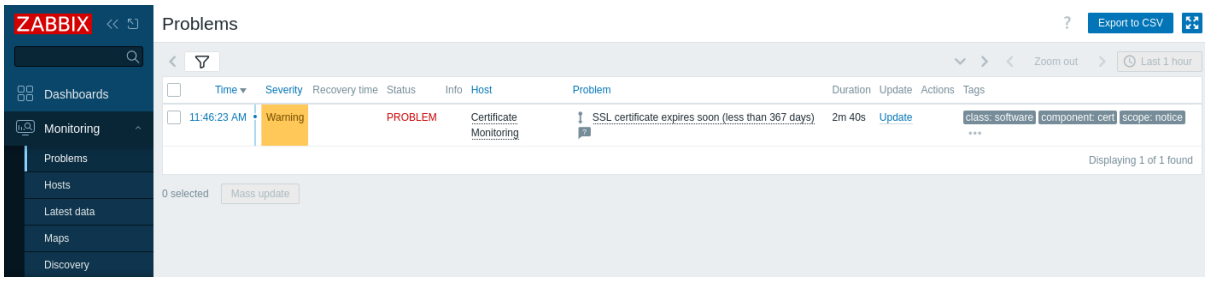
1. Zabbix で「Certificate Monitoring」ホストの設定を開きます。

2. マクロタブに切り替え、継承されたマクロとホストマクロを選択します。

3. 以前に設定した {$CERT.EXPIRY.WARN} マクロ値の横にある 変更をクリックし、証明書の有効期限が切れる前に警告を受け取るために非常に大きな日数（365 日以上で十分です）を設定します。

4. 更新をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は「SSL 証明書の有効期限が近づいています」という問題を検出し、有効期限までの日数を表示します。問題は監視データ > 障害 に表示されます。



アラートが設定されている場合は、問題の通知も受け取ります。

6. マクロ値を元の値に戻して問題を解決し、証明書の値の監視を続けます。

参考

- Zabbix agent 2 - アイテムキーの一覧。
- テンプレート *Website certificate by Zabbix agent 2* - *Website certificate by Zabbix agent 2* テンプレートの追加情報。
- テンプレート *Website certificate by Zabbix agent 2 active* - *Website certificate by Zabbix agent 2 active* テンプレートの追加情報。
- Browser アイテムで Web サイトを監視する - Browser アイテムで Web サイトの基本的な監視を開始する方法。

10 Zabbix でネットワークスイッチやルーターを監視する

はじめに　このガイドでは、Zabbix を使用してネットワークスイッチやルーターの基本的な監視を開始するために必要な手順を説明します。ここでは Cisco ルーターを例にしていますが、この手順は SNMP 対応のネットワーク機器すべてに適用できます。

このガイドの対象者

このガイドは、ネットワーク機器の基本的な監視を迅速に有効化したい新しい Zabbix ユーザーやネットワーク管理者向けに作成されています。高度なカスタマイズや詳細な設定オプションが必要な場合は、SNMP エージェントページや Zabbix マニュアルの設定セクションを参照してください。

前提条件

このガイドを進める前に、以下を確認してください。

- Zabbix サーバーおよび Zabbix フロントエンドがインストールされていること：お使いの OS の手順に従ってインストールしてください（パッケージからのインストールおよびWeb インターフェースのインストールを参照）。
- SNMP 対応機器：SNMP が有効なネットワークスイッチまたはルーター（例：Cisco ルーター）。
- MIB ファイルがインストールされていること：MIB ファイルをインストールすると、Zabbix で数値 OID を人間が読める名前や説明に変換できるようになります。適切な MIB サポートがない場合、数値のみが表示され、アイテムの設定やトラブルシューティングが困難になります。

Ubuntu で MIB ファイルをインストールするには：

1. MIB ダウンローダーパッケージをインストールします。

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

ベンダー固有の MIB（例：Cisco、Juniper）を追加する必要がある場合は、適切な MIB ディレクトリに配置してください。

- Linux ベースのシステムでは、一般的な場所は /usr/share/snmp/mibs/ または /usr/local/share/snmp/mibs/ です。
- Zabbix インストールの場合、MIB ファイルは /var/lib/zabbix/mibs/ に保存できます。

MIBDIRS 環境変数または snmp.conf ファイルに正しいパスが含まれていることを確認してください。

新しい MIB をシステムが認識しているか確認するには、以下を使用します。

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

詳細な手順については、お使いの SNMP ライブラリのドキュメントを参照してください。

- Cisco MIBs
- Juniper MIBs

2. /etc/snmp/snmp.conf を編集し、mibs : で始まる行をコメントアウトして、システムがすべての利用可能な MIB を読み込めるようにします。

3. snmpwalk（例：snmpwalk -v 2c -c <your_community_string> <device_IP>）を実行し、OID が説明的な名前で表示されることを確認します。

このガイドは、以下のセットアップに基づいています。

- Zabbix バージョン：7.2（パッケージからインストール）
- OS ディストリビューション：Ubuntu
- OS バージョン：24.04.2+
- Zabbix コンポーネント：サーバー、フロントエンド、および（ローカルネットワークメトリクスを監視する場合は）エージェント
- データベース：MySQL
- Web サーバー：Apache
- ネットワーク機器：Cisco Catalyst 3750V2-24FS

ネットワーク機器がすでに物理的に設置され、接続されていることを前提としています。

ネットワークデバイスの設定（**Cisco** ルーターの例）　SNMP 経由で監視するには、ネットワークデバイスを設定して SNMP クエリを許可する必要があります。以下の例は SNMPv2 用であり、既存の設定は考慮されていません。注意：これらのコマンドを適用すると、現在の SNMP 設定が上書きされる場合があります。

Cisco ルーターの場合、設定は通常以下の手順で行います。

SNMPv2 の例　1. SNMP を有効にし、コミュニティ文字列を設定します。

Cisco ルーターのコンソールにログインし、設定モードに入ります。

```
configure terminal
```

次に、読み取り専用のコミュニティ文字列を指定してSNMP を有効にします。例：

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

<your_community_string> を安全なコミュニティ文字列に置き換えてください。注：RO（読み取り専用）オプションは、SNMP がデバイスからデータを取得できるようにしますが、設定の変更はできません。

セキュリティ上の理由から、必要なデバイスのみに SNMP アクセスを制限することを推奨します。アクセスコントロールリスト（ACL）の設定に関する詳細は、Cisco の公式ドキュメントを参照してください。

2. 設定を保存します。

SNMP 設定が再起動後も保持されるように、変更を保存します。

```
write memory
```

SNMPv3 の例　SNMPv3 は認証と暗号化による強化されたセキュリティを提供します。その設定は SNMPv2 よりも安全であり、デバイス固有のドキュメントで確認する必要があります。

1. SNMP グループの作成

プライバシー（暗号化）を有効にした SNMPv3 グループを設定します。

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

2. SNMP ユーザーの作成

認証とプライバシーを持つ SNMPv3 ユーザーを追加します。プレースホルダーを希望の値に置き換えてください。

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

3. 設定の保存

```
write memory
```

詳細やモデル固有の手順については、外部のCisco SNMP 設定チュートリアルを参照してください。このガイドでは、SNMP 監視を有効にするための基本的な手順を提供しています。

Zabbix フロントエンドの設定

Zabbix フロントエンドでホストを作成する　1. Zabbix フロントエンドにログインします。

2. 新しいホストを追加します。

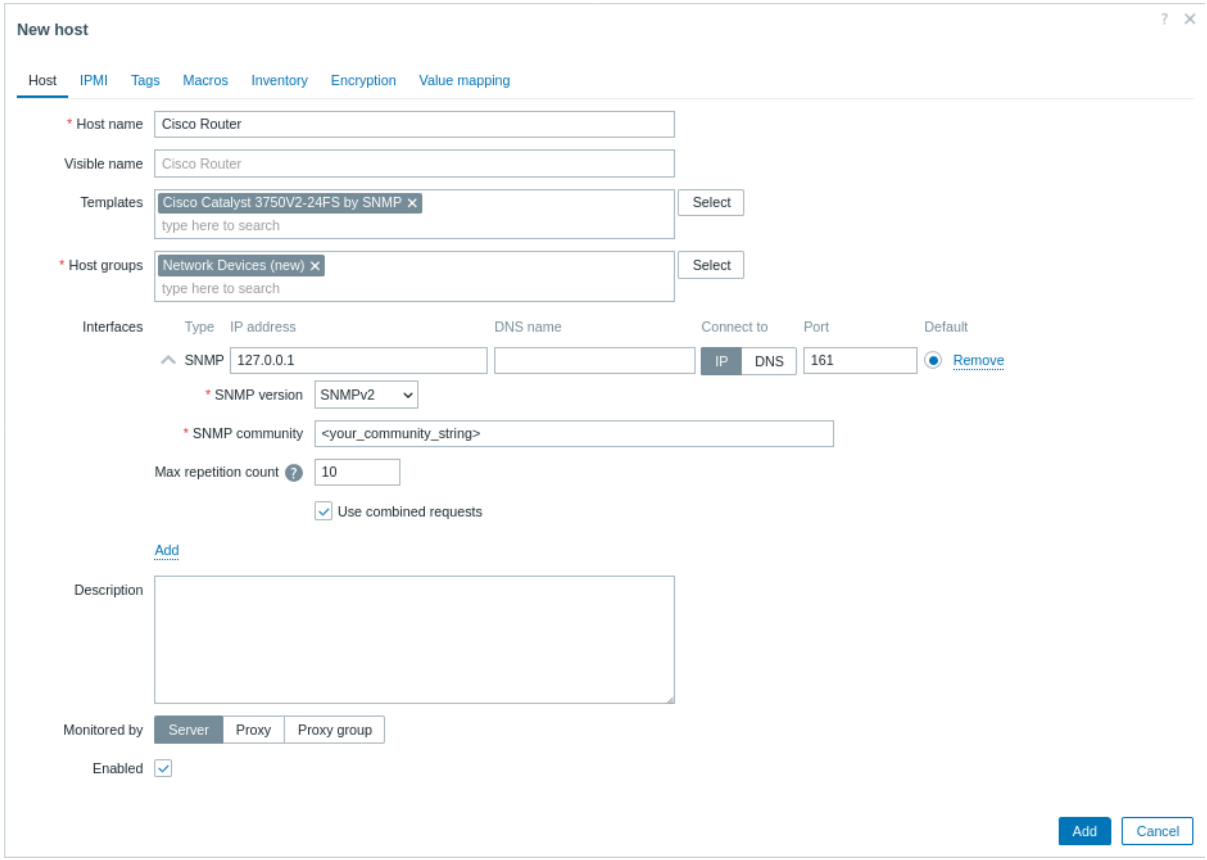
データ収集 > ホストに移動し、ホストの作成をクリックします。

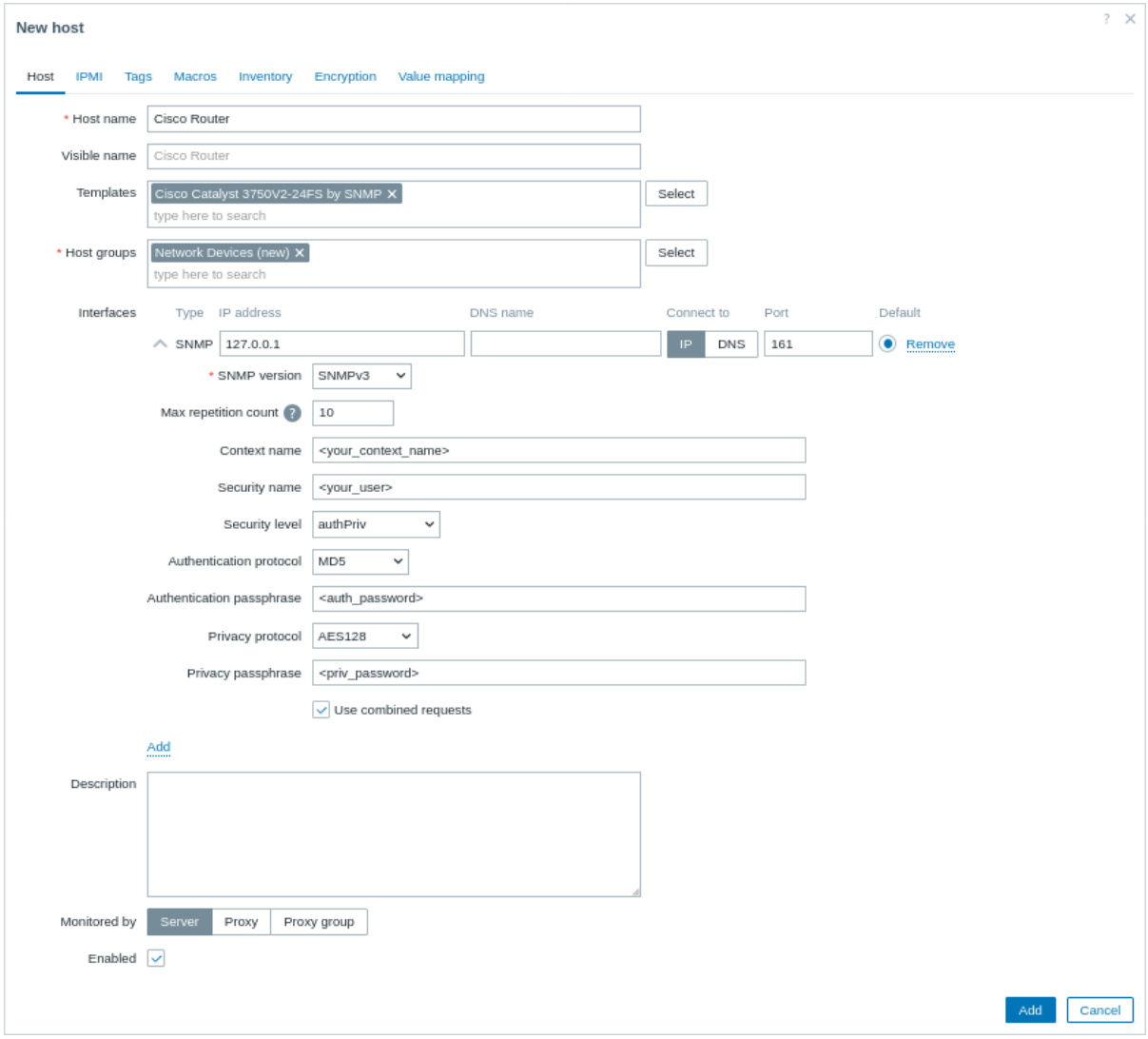
- ホスト名: デバイスの名前を入力します（例: ”Cisco Router”）。
- ホストグループ: 既存のグループを選択するか、「ネットワークデバイス」などの新しいグループを作成します。
- インターフェース:
  - インターフェースの下の追加をクリックします。
  - インターフェースタイプとして *SNMP* を選択します。
  - Cisco ルーターの IP アドレスまたは DNS 名を入力します。
  - デフォルトの SNMP ポート（通常は 161）を設定します。
  - ドロップダウンメニューから適切な SNMP バージョン（例: SNMPv2）を選択します。
  - SNMPv1/v2 の場合は、*SNMP* コミュニティフィールドにコミュニティ文字列を入力します。SNMPv3 の場合は、追加の認証情報（コンテキスト名、セキュリティ名、セキュリティレベルなど）が求められます。

3. テンプレートのリンク

テンプレートフィールドで、デバイスに最も適した SNMP テンプレートを選択します。Zabbix には、多くのデバイスファミリー向けに事前構築されたSNMP テンプレートが用意されています。たとえば、Cisco デバイスを監視する場合は、デバイスの OS やモデルに対応するテンプレート（Cisco IOS SNMP や Cisco Catalyst 3750<device model> SNMP など）を選択します。

4. 追加をクリックしてホストを保存します。

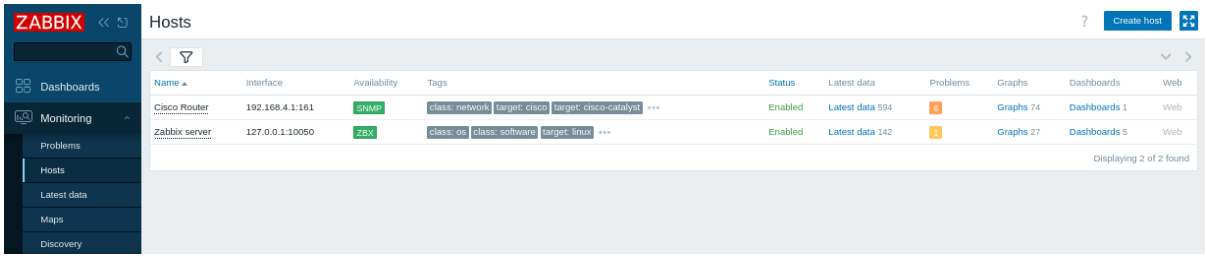




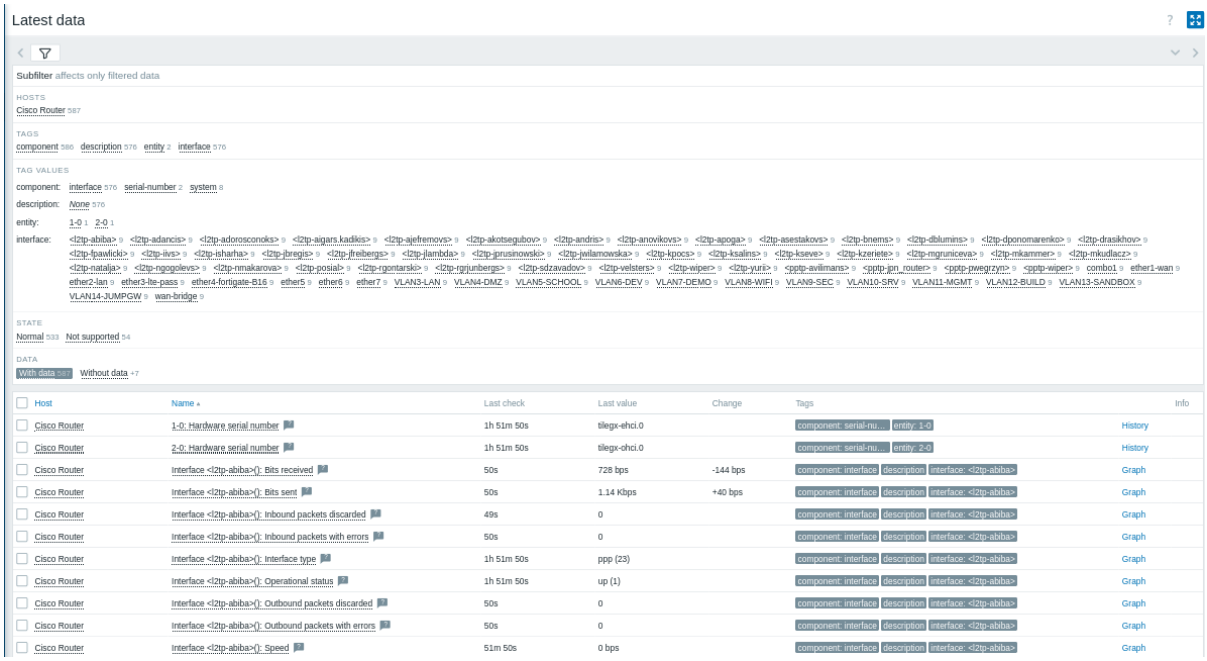
収集したメトリクスの表示　おめでとうございます！Zabbix はネットワークデバイスの監視を行うように設定されました。

最新データ:

- Zabbix フロントエンドで監視 > [最新データ] に移動します。



- “Cisco Router” ホスト（または検出されたホスト）を選択し、ハードウェアやネットワークの稼働時間、ICMP 損失、ping、応答時間などのメトリクスを表示します。



- グラフとスクリーン:

パフォーマンスデータを可視化するには、SNMP アイテムの横にあるグラフをクリックして詳細なメトリクスを確認します。

次のステップとして、以下を行うことができます:

- 追加のメトリクスを監視するためにカスタム SNMP アイテムを追加します。
- 潜在的な問題について通知を受け取るために問題アラートを設定します。

SNMP アイテムの作成　ホストのセットアップが完了したら、特定のメトリックを監視するためのアイテムを作成できます。注意: テンプレートを使用している場合、このステップは省略可能です。テンプレートにはすでにデフォルトのアイテムセットが含まれています。

1. SNMP OID の特定:

`snmpwalk` コマンドを使用して、デバイス上で利用可能な OID を一覧表示します。例:

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

監視したいメトリックの OID を探します（例: ポート 3 の受信トラフィックの場合は IF-MIB::ifHCInOctets.3）。数値 OID を取得するには、次のコマンドを使用できます:

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

2. SNMP アイテムの作成:

- データ収集 > ホストに移動し、SNMP ホストのアイテムタブをクリックしてアイテムの作成をクリックします。
- 名前: わかりやすい名前を入力します（例:「ポート 3 受信トラフィック」）。
- タイプ: *SNMP* エージェントを選択します。
- キー: 意味のあるキーを指定します（例: `cisco.ifHCInOctets.3`）。
- ホストインターフェース: SNMP インターフェースが選択されていることを確認します。
- *SNMP OID*: サポートされている形式のいずれかで OID を入力します。例:
  - 単一値の場合は `get[1.3.6.1.2.1.31.1.1.1.6.3]`;
  - 非同期で値のサブツリーを取得する場合は `walk[1.3.6.1.2.1.31.1.1.1.6.3]`。
- 前処理（必要に応じて）: アイテムが累積カウンター（インターフェーストラフィックなど）を返す場合は、前処理タブに移動し、「1 秒あたりの変化量」などの前処理ステップを追加してレートを計算します。

1 回の SNMP トランザクションで複数の値を取得するには、`walk[OID1,OID2,...]` の構文で複数の OID を指定できます。

数値 **OID** と **MIB** 名の相互変換　SNMP を使用する際、数値 OID と対応する MIB 名を相互に変換する必要がある場合があります。この変換により、メトリクスの特定やトラブルシューティングが容易になります。

- MIB 名を数値 OID に変換するには、`snmptranslate` コマンドに`-On` オプションを付けて使用します。たとえば、MIB 名 `IF-MIB::ifHCInOctets.3` を数値 OID に変換するには、次のように実行します。

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

このコマンドの出力例:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- 数値 OID を MIB 名に変換するには、snmptranslate コマンドに-IR（または-m ALL）オプションを付けて逆変換します。たとえば、数値 OID.1.3.6.1.2.1.31.1.1.1.6.3 を MIB 名に変換するには、次のように実行します。

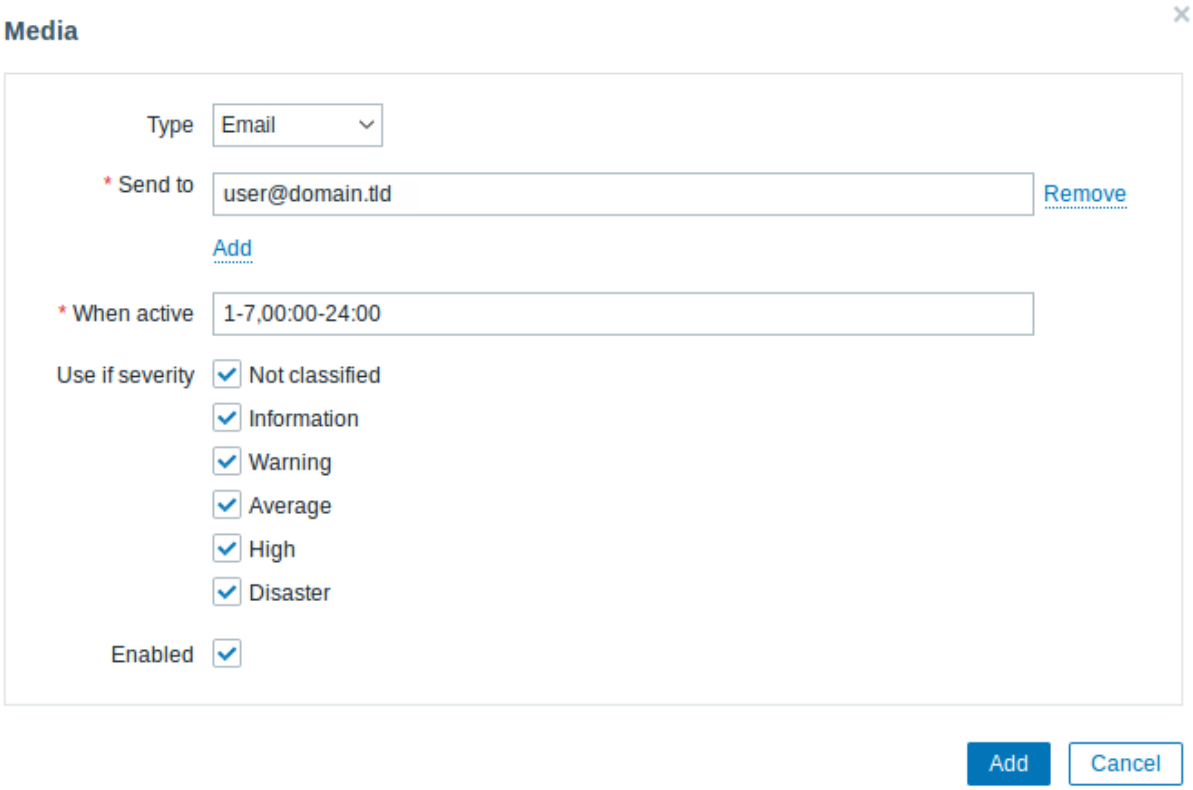
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

このコマンドの出力例:

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

障害アラートの設定　このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。

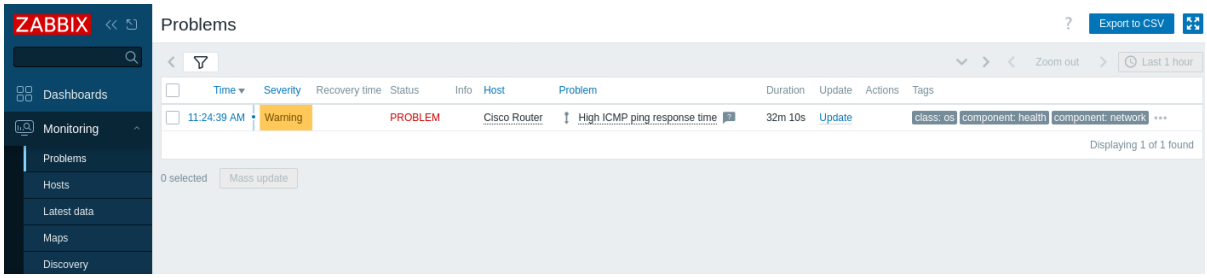


2. 障害通知の受信のガイドに従います。

次回、Zabbix が障害を検出したときに、メールでアラートを受信できるはずです。

設定をテストする　Zabbix がネットワークパフォーマンスの問題を正しく検出できることを確認するために、ICMP ping 応答時間のしきい値を上げて実際の問題をシミュレートします。

1. Zabbix で「Cisco Router」ホストの設定を開きます。
2. マクロタブに移動し、継承されたホストマクロを選択します。
3. {$ICMP_RESPONSE_TIME_WARN} マクロ（または同様の応答時間しきい値マクロ）を探します。
4. 非常に低い値（例：0.001）を設定し、ping 応答がこの値を超えた場合にアラートが発生するようにします。
5. 更新をクリックして変更を適用します。
6. Zabbix がシミュレートされた問題を検出するまでしばらく待ちます。
7. 監視 > 障害に移動し、アラートが表示されることを確認します（例：「ICMP ping 応答時間が高い」）。



アラートが設定されている場合は、問題アラートも受信するはずです。

8. マクロ値を元の設定に戻し、更新をクリックして変更を保存します。

9. 問題が解決され、障害セクションから消えることを確認します。

SNMP 監視のトラブルシューティング　Zabbix フロントエンドの SNMP アイコンが赤色で表示されたり、データが収集されない場合は、以下の手順をお試しください。

1. SNMP の接続性を確認します。

SNMPv2 の場合、Zabbix サーバーから以下のコマンドを実行します。

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

このコマンドは、デバイスが SNMP クエリに応答するかどうかを確認します。

SNMPv3 の場合は、適切な SNMPv3 認証情報を含めてください。

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

これにより、SNMPv3 の認証情報が正しいことと、デバイスが安全に応答していることを確認できます。

2. 前提条件で説明したように、MIB ファイルがインストールされ有効になっていることを確認します。ネットワークデバイスにクエリを実行した際、以下のコマンドでエラーが出ないことを確認してください。

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

これにより、エラーなく OID が翻訳されて返されるはずです。

3. Zabbix で設定した SNMP バージョンと認証情報が、デバイス側の設定と一致していることを確認します。たとえば、Zabbix のホスト設定で SNMP の設定を確認し、デバイスの設定と照合してください。Cisco デバイスの場合、以下のコマンドで SNMP 設定を確認できます。

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

これにより、コミュニティ文字列（SNMPv2 の場合）や SNMPv3 ユーザー情報が正しいことを確認できます。

4. ネットワークデバイスで SNMP が正しく有効になっていることを確認します。Cisco ルーターの場合、コンソールにログインして以下を実行します。

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

このコマンドは、アクティブな SNMP 設定を表示し、SNMP が正しく設定されているかどうかを確認するのに役立ちます。

5. Zabbix サーバーとデバイス間で、ファイアウォールやネットワークの問題により SNMP トラフィック（通常はポート 161）がブロックされていないことを確認します。接続性は以下でテストできます。

nc -zv <device_IP> 161

`nc -zv` は、デバイスのポート 161 が開いていてリッスンしているかどうかを確認します。

また、Ubuntu で UFW を使用している場合は、ファイアウォールの状態を確認してください。

```
sudo ufw status
```

または、iptables の場合は以下を実行します。

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

6. 問題の特定に役立つ SNMP 関連のエラーがないか、Zabbix サーバーのログファイルを確認します。

```
tail -f /tmp/zabbix_server.log
```

`tail -f` は、リアルタイムでログの更新を監視できます。

関連情報:

- アイテムの作成 - 追加のメトリクスを追加する方法について説明します。
- SNMP エージェント - Zabbix による SNMP 監視に関する追加情報。
- ネットワークデバイス用の標準化テンプレート - 利用可能な SNMP テンプレートに関する情報。
- SNMP OID のディスカバリ - スイッチでの SNMP ディスカバリに関する追加情報。
- ネットワークディスカバリルールの設定 - Zabbix がホストやサービスを検出するために使用するネットワークディスカバリルールの設定方法に関する追加情報。

11 アクティブチェックを使用して Windows イベントログを監視する

はじめに

このガイドでは、Zabbix のアクティブチェックを使用して Windows イベントログを監視する方法について説明します。Zabbix の Windows 固有のアイテムキーを使用することで、重要なイベント（ログオン失敗、システムエラーなど）をリアルタイムで収集・分析できます。

このガイドの対象者

このガイドは、Windows イベントログを監視したい Zabbix の新規ユーザーやネットワーク管理者向けに作成されています。高度な設定オプションについては、Windows 固有のアイテムキーのドキュメントを参照してください。

前提条件

このガイドを進める前に、OS の手順に従って Zabbix サーバーと Zabbix フロントエンドをダウンロードしてインストールする必要があります。また、監視対象の Windows マシンに Zabbix エージェントをダウンロードしてインストールする必要があります。

Windows イベントログ監視のための Zabbix エージェントの設定

1. Windows ホスト上の `zabbix_agentd.conf`（デフォルトパスは `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`）を開き、ServerActiveパラメータが Zabbix サーバーの IP アドレスに設定されていること、およびHostnameパラメータがZabbix フロントエンドで定義するホスト名と一致していることを確認します。これにより、エージェントは自身のホストに対するアクティブチェックを指定した Zabbix サーバーから要求できるようになります。例：

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

2. 変更を反映するために Zabbix エージェントサービスを再起動します：

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

3. Windows ホストが稼働していることを確認します：

- Windows ホストで Zabbix エージェントサービスが実行されていることを確認します。
- Windows ホストがポート 10051 で Zabbix サーバーに接続できることを確認します。Windows ホストから接続性をテストするには、PowerShell を開いて次のコマンドを実行します：

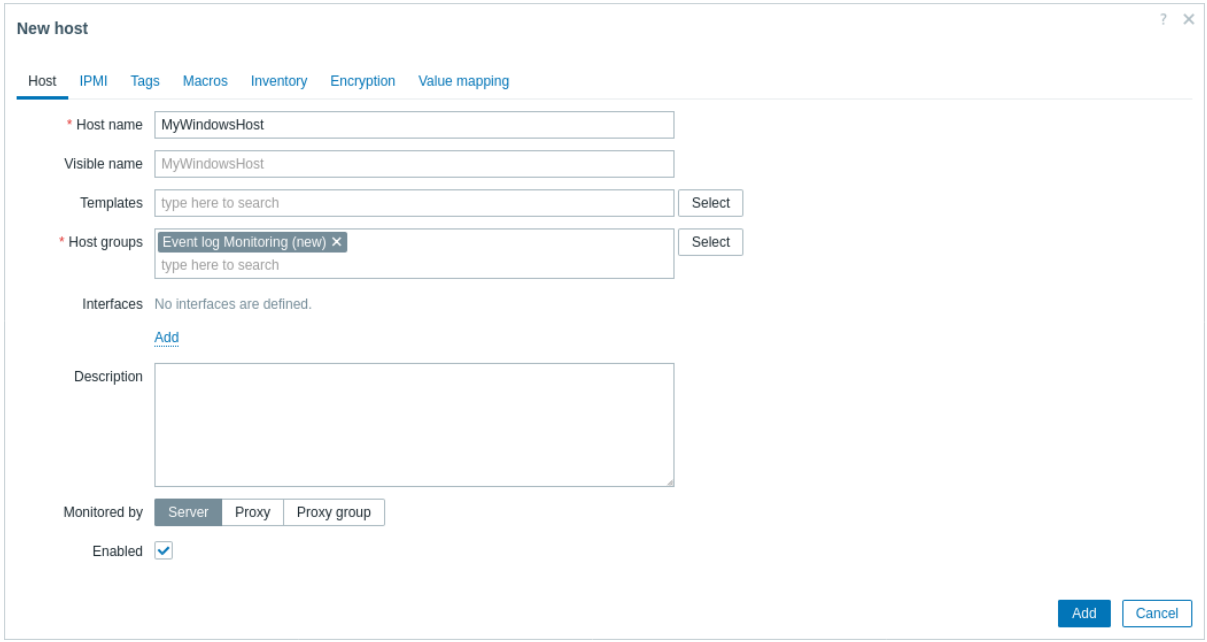
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Zabbix フロントエンドの設定

1. データ収集 > ホストに移動し、ホストを作成します。

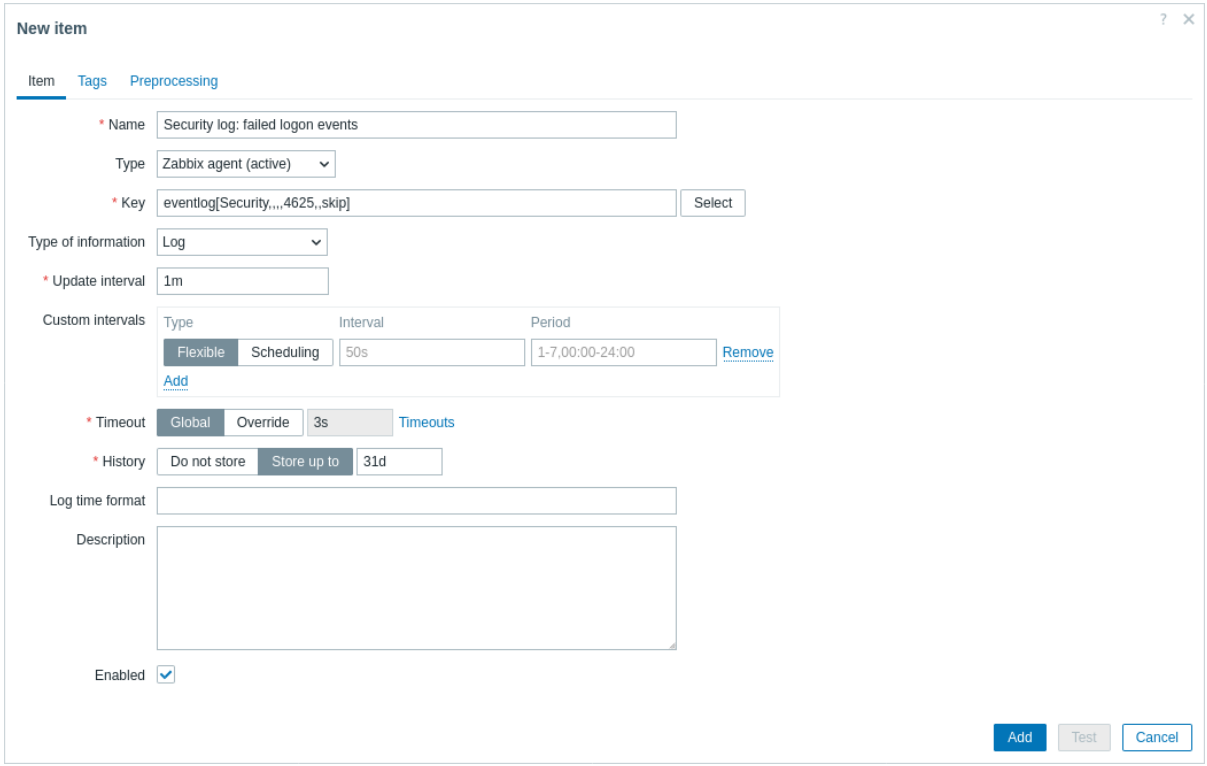
- ホスト名フィールドにホスト名（例：”MyWindowsHost”）を入力します。
- ホストグループフィールドにホストグループ（例：”Event log Monitoring”）を入力または選択します。
- 追加を押して、設定したホストを保存します。

> **Note:**
> テンプレートフィールドに”Windows by Zabbix agent active” テンプレートを追加することで、同じホスト上の他のアクティブアイテムが更新されているかどうかを観察し、トラブルシューティングに役立てることができます。



2. 次のパラメータで新しいアイテムを作成します。

- 名前フィールドに、説明的なアイテム名（例：”Security log: failed logon events”）を入力します。
- タイプドロップダウンで”Zabbix エージェント（アクティブ）” を選択します（イベントログ監視には必須）。
- キーフィールドに、eventlogアイテムキーを使用します。例えば、セキュリティログ内の失敗したログオン試行（イベント ID: 4625）を監視し、アイテムの最終チェックより古いエントリを無視する（skip パラメータを使用）には、次のアイテムキーを入力します：`eventlog[Security,,,,4625,,skip]`
- 情報のタイプドロップダウンで” ログ” を選択します。



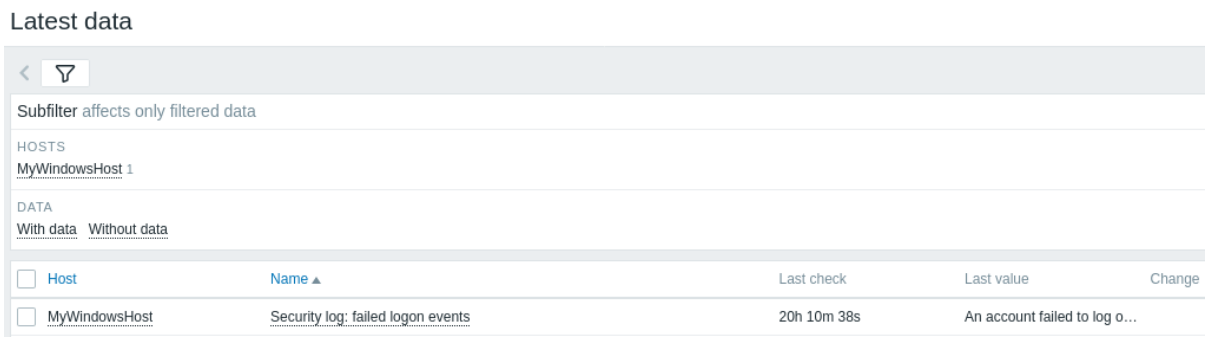
3. 追加をクリックしてアイテムを保存します。

収集されたメトリクスのテストと表示

おめでとうございます！Zabbix は Windows イベントログの収集を行うように設定されました。イベントログが収集されていることを確認するには、Windows アカウントからログアウトし、誤った認証情報でログインを試みることで、「セキュリティログ：ログオン失敗イベント」アイテムをテストできます。

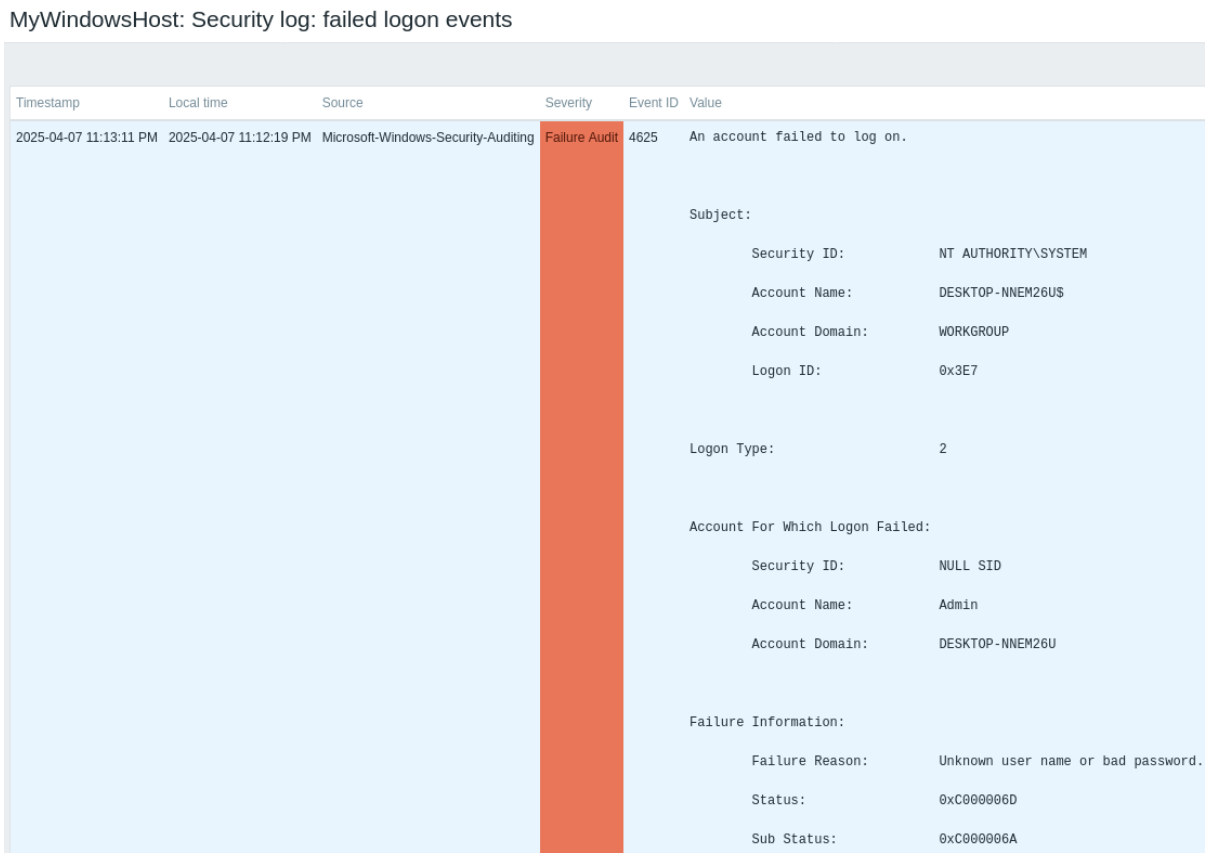
次に、Zabbix フロントエンドで収集されたログを表示します。

1. Zabbix フロントエンドの監視 > 最新データに移動します。



2. 名前フィールドで「MyWindowsHost」ホストでフィルタリングします。

3. 履歴をクリックして記録されたログ値を表示します。



4. ログ値が存在しない場合は、本ガイドのトラブルシューティングセクションに進んでください。

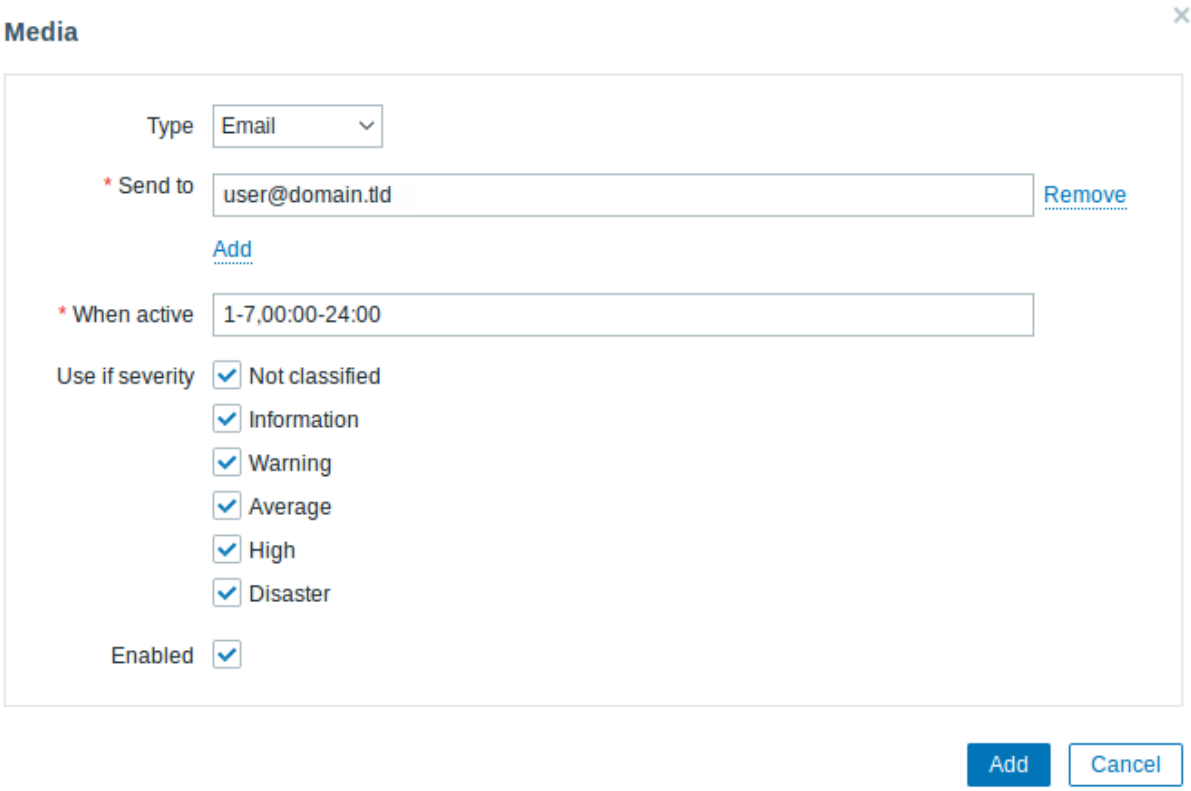
問題アラートの設定

このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. データ収集 > ホストに移動し、イベントログアイテムが関心のあるパターンを記録したときに発生するトリガーを定義します。たとえば、Security ログで失敗したログオン試行を検出するには、find()関数を使用します。

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

2. ユーザー設定 > プロファイルに移動し、メディアタブに切り替えてメールアドレスを追加します。



3. 問題通知の受信のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるはずです。

トラブルシューティング

Windows イベントログの収集や表示に問題が発生した場合は、以下のヒントを使用して一般的な問題を特定し、解決してください。

1. Zabbix サーバー（Linux）で、次のコマンドを使用して iptables ルールを一覧表示します。

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

そして、TCP ポート 10051 に対する ACCEPT ルールがあることを確認します。

2. `eventlog[...]` キーが、Windows 固有のアイテムキーに示されている通り、正確なログ名（大文字小文字を区別）、イベント ID、モード（例：skip）、その他のパラメータを正確に使用していることを確認してください。

関連情報**:**

- アイテムの作成 - 追加のメトリクスを追加する方法を学びます。
- Microsoft Windows 上の Zabbix エージェント - 詳細なインストール手順。
- Zabbix エージェントで Windows を監視する - Zabbix エージェントを使用して Windows マシンの基本的な監視を設定するための包括的なガイド。
- Windows 固有のアイテムキー - イベントログ監視用のキーを含む、Zabbix エージェントがサポートする Windows 固有のアイテムキーに関する詳細情報。
- ログファイル監視 - Windows イベントログにも適用できる、Zabbix によるログファイルの集中監視と分析の設定手順。

Zabbix とは

Zabbix は Alexei Vladishev によって作成され、現在は Zabbix SIA によって積極的に開発・サポートされています。

Zabbix は、エンタープライズクラスのオープンソース分散監視ソリューションです。

Zabbix は、ネットワークの多数のパラメータや、サーバー、仮想マシン、アプリケーション、サービス、データベース、ウェブサイト、クラウドなどの健全性と整合性を監視するソフトウェアです。Zabbix は柔軟な通知メカニズムを使用しており、ユーザーはほぼすべてのイベントに対してメールベースのアラートを設定できます。これにより、サーバーの問題に迅速に対応できます。Zabbix は、保存されたデータに基づく優れたレポートおよびデータ可視化機能を提供します。これにより、Zabbix はキャパシティプランニングに最適です。

Zabbix はポーリングとトラッピングの両方をサポートしています。すべての Zabbix レポートや統計情報、設定パラメータは、ウェブベースのフロントエンドを通じてアクセスできます。ウェブベースのフロントエンドにより、ネットワークの状態やサーバーの健

全性をどこからでも評価できます。適切に設定された Zabbix は、IT インフラストラクチャの監視において重要な役割を果たすことができます。これは、数台のサーバーを持つ小規模な組織にも、多数のサーバーを持つ大企業にも同様に当てはまります。

Zabbix は無償です。Zabbix は AGPL-3.0 ライセンスの下で記述・配布されています。つまり、そのソースコードは自由に配布され、一般に公開されています。

商用サポートは、Zabbix 社および世界中のパートナーによって提供されています。

Zabbix の機能について詳しく学ぶ。

Zabbix のユーザー　世界中の多くの規模の異なる組織が、Zabbix を主要な監視プラットフォームとして利用しています。

アーキテクチャ　Zabbix はいくつかの主要なソフトウェアコンポーネントで構成されています。それぞれの役割は以下の通りです。

サーバー

Zabbix サーバーは、エージェントが可用性と整合性の情報および統計情報を報告する中央コンポーネントです。サーバーは、すべての設定、統計、および運用データが保存される中央リポジトリです。

データベースストレージ

すべての設定情報と Zabbix によって収集されたデータは、データベースに保存されます。

Web インターフェース

どこからでも、どのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。このインターフェースは Zabbix サーバーの一部であり、通常は (必ずしもそうである必要はありませんが) サーバーを実行しているのと同じ物理マシン上で実行されます。

プロキシ

Zabbix プロキシは、Zabbix サーバーに代わってパフォーマンスおよび可用性データを収集できます。プロキシは Zabbix の導入において必須ではありませんが、単一の Zabbix サーバーの負荷を分散するのに非常に有益です。

エージェント

Zabbix エージェントは、監視対象に導入され、ローカルリソースやアプリケーションを積極的に監視し、収集したデータを Zabbix サーバーに報告します。Zabbix 4.4 以降、2 種類のエージェントが利用可能です。Zabbix エージェント（軽量、多くのプラットフォームでサポート、C 言語で記述）とZabbix エージェント 2（柔軟性が高く、プラグインで簡単に拡張可能、Go 言語で記述）です。

データフロー　さらに、Zabbix 内の全体的なデータフローを俯瞰して見ることも重要です。データを収集するアイテムを作成するには、まずホストを作成する必要があります。Zabbix のもう一方の端に進むと、トリガーを作成するにはまずアイテムが必要です。アクションを作成するにはトリガーが必要です。したがって、*Server X* の CPU 負荷が高すぎる場合にアラートを受け取りたい場合は、まず *Server X* のホストエントリを作成し、その CPU を監視するアイテムを作成し、CPU が高すぎる場合にアクティブになるトリガーを作成し、最後にメールを送信するアクションを作成する必要があります。手順が多いように思えるかもしれませんが、テンプレートを使用すれば実際にはそうではありません。しかし、この設計により非常に柔軟なセットアップが可能になります。

機能　Zabbix は高度に統合されたネットワーク監視ソリューションであり、1 つのパッケージで多くの機能を提供します。

データ収集

- 可用性およびパフォーマンスのチェック
- SNMP（トラップおよびポーリングの両方）、IPMI、JMX、VMware 監視のサポート
- カスタムチェック
- 任意の間隔でのデータ収集
- サーバー/プロキシおよびエージェントによる実行

柔軟なしきい値定義

- バックエンドデータベースの値を参照するトリガーと呼ばれる非常に柔軟なしきい値を定義可能

高度に設定可能なアラート通知

- 通知の送信はエスカレーションスケジュール、受信者、メディアタイプごとにカスタマイズ可能
- マクロ変数を使用して通知を意味のあるものにできる
- 自動アクションにはリモートコマンドを含む

リアルタイムグラフ化

- 監視項目は内蔵のグラフ機能ですぐにグラフ化

Web 監視機能

- Zabbix は Web サイト上のマウスクリックのパスをシミュレートし、機能や応答時間をチェック可能

豊富な可視化オプション

- 複数の項目を 1 つのビューにまとめたカスタムグラフの作成
- ネットワークマップ
- ダッシュボードスタイルのスライドショー
- レポート
- 監視リソースの高レベル（ビジネス）ビュー

履歴データの保存

- データはデータベースに保存
- 履歴の設定が可能
- 内蔵のハウスキーピング手順

簡単な設定

- 監視対象デバイスをホストとして追加
- データベースに登録されると監視対象として認識
- テンプレートを監視対象デバイスに適用

テンプレートの利用

- チェックをテンプレートでグループ化
- テンプレートは他のテンプレートを継承可能

ネットワークディスカバリ

- ネットワークデバイスの自動検出
- エージェントの自動登録
- ファイルシステム、ネットワークインターフェース、SNMP OID の検出

高速 **Web** インターフェース

- PHP による Web ベースのフロントエンド
- どこからでもアクセス可能
- クリック操作で設定可能
- 監査ログ

Zabbix API

- Zabbix API は、大量操作、サードパーティソフトウェアとの統合、その他の目的のために Zabbix へのプログラムインターフェースを提供します。

権限システム

- セキュアなユーザー認証
- 特定のユーザーに特定のビューのみを許可可能

高機能で拡張性の高いエージェント

- 監視対象に導入
- Linux と Windows の両方に導入可能

バイナリデーモン

- パフォーマンスと小さなメモリフットプリントのため C で記述
- 簡単に移植可能

複雑な環境にも対応

- Zabbix プロキシを使用することでリモート監視が容易

用語集　このセクションでは、Zabbix で一般的に使用される用語の意味を学ぶことができます。

ホスト

- 物理または仮想デバイス、アプリケーション、サービス、または監視するパラメータの論理的な集合体。

ホストグループ

- ホストの論理的なグループ化。ホストグループは、異なるユーザーグループに対してホストへのアクセス権を割り当てる際に使用されます。

アイテム

- ホストから受信したい特定のデータ、データのメトリック。

値の事前処理

- 受信したメトリック値の変換。データベースに保存する前に行われます。

トリガー

- 問題の閾値を定義し、アイテムで受信したデータを「評価」するために使用される論理式。

受信したデータが閾値を超えると、トリガーは「正常」から「障害」状態に遷移します。受信したデータが閾値を下回ると、トリガーは「正常」状態のまま、または「正常」状態に戻ります。

テンプレート

- *1* つまたは複数のホストに適用できるエンティティ（アイテム、トリガー、グラフ、低レベルディスカバリールール、*Web* シナリオ）のセット。

テンプレートの役割は、ホストへの監視タスクの展開を迅速化し、大量の監視タスクの変更を容易にすることです。テンプレートは個々のホストに直接リンクされます。

テンプレートグループ

- テンプレートの論理的なグループ化。テンプレートグループは、異なるユーザーグループに対してテンプレートへのアクセス権を割り当てる際に使用されます。

イベント

- トリガーの状態変化やディスカバリ/エージェントの自動登録など、注意を要する単一の出来事。

イベントタグ

- イベントのための事前定義されたマーカー。イベントの相関、権限の細分化などに使用できます。

イベント相関

- 問題とその解決を柔軟かつ正確に関連付ける方法。

例えば、あるトリガーが報告した問題が、別のトリガー（異なるデータ収集方法を使用している場合もある）によって解決されたと定義することができます。

障害

- 「障害」状態のトリガー。

障害の更新

- コメントの追加、認識、重大度の変更、手動でのクローズなど、*Zabbix* が提供する障害管理オプション。

アクション

- イベントに反応するための事前定義された手段。

アクションは、操作（例：通知の送信）と条件（いつ操作を実行するか）で構成されます。

エスカレーション

- アクション内で操作を実行するためのカスタムシナリオ。通知の送信やリモートコマンドの実行のシーケンス。

メディア

- 通知を配信する手段。配信チャネル。

通知

- 選択したメディアチャネルを介してユーザーに送信されるイベントに関するメッセージ。

リモートコマンド

- 特定の条件で監視対象ホスト上で自動的に実行される事前定義されたコマンド。

Web シナリオ

- *Web* サイトの可用性をチェックするための *1* つまたは複数の *HTTP* リクエスト。

フロントエンド

- *Zabbix* に付属する *Web* インターフェース。

ダッシュボード

- 重要な情報の要約や可視化を、ウィジェットと呼ばれるビジュアルユニットで表示する *Web* インターフェースのカスタマイズ可能なセクション。

ウィジェット

- ダッシュボードで使用される、特定の種類やソースの情報（要約、マップ、グラフ、時計など）を表示するビジュアルユニット。

Zabbix API

- *Zabbix API* を使用すると、*JSON RPC* プロトコルを使用して *Zabbix* オブジェクト（ホスト、アイテム、グラフなど）の作成、更新、取得や、その他のカスタムタスクを実行できます。

***Zabbix* サーバー**

- 監視を実行し、*Zabbix* プロキシやエージェントと連携し、トリガーを計算し、通知を送信する *Zabbix* ソフトウェアの中心的なプロセス。データの中央リポジトリ。

***Zabbix* プロキシ**

- *Zabbix* サーバーの代わりにデータを収集し、サーバーの処理負荷を軽減するプロセス。

***Zabbix* エージェント**

- 監視対象にデプロイされ、ローカルリソースやアプリケーションを積極的に監視するプロセス。

Zabbix* エージェント *2

- ローカルリソースやアプリケーションを積極的に監視し、カスタムプラグインによる監視が可能な新世代の *Zabbix* エージェント。

> **Attention:**
> Zabbix エージェント 2 は Zabbix エージェントと多くの機能を共有しているため、ドキュメント内の「Zabbix エージェント」という用語は、機能的な動作が同じ場合は Zabbix エージェントと Zabbix エージェント 2 の両方を指します。Zabbix エージェント 2 は、その機能が異なる場合にのみ特に明記されます。

暗号化

- *Zabbix* コンポーネント（サーバー、プロキシ、エージェント、*zabbix_sender* および *zabbix_get* ユーティリティ）間の暗号化通信のサポート。Transport Layer Security（TLS）プロトコルを使用します。

エージェントの自動登録

- *Zabbix* エージェント自身がホストとして登録され、監視を開始する自動化プロセス。

ネットワークディスカバリー

- ネットワークデバイスの自動検出。

低レベルディスカバリー

- 特定のデバイス上の低レベルエンティティ（ファイルシステム、ネットワークインターフェースなど）の自動検出。

低レベルディスカバリールール

- デバイス上の低レベルエンティティを自動検出するための定義セット。

アイテムプロトタイプ

- 低レベルディスカバリー用に特定のパラメータを変数として持つメトリック。低レベルディスカバリー後、変数は自動的に実際に検出されたパラメータに置き換えられ、メトリックは自動的にデータ収集を開始します。

トリガープロトタイプ

- 低レベルディスカバリー用に特定のパラメータを変数として持つトリガー。低レベルディスカバリー後、変数は自動的に実際に検出されたパラメータに置き換えられ、トリガーは自動的にデータの評価を開始します。

他の *Zabbix* エンティティのプロトタイプ（グラフプロトタイプ、ホストプロトタイプ、ホストグループプロトタイプ）も低レベルディスカバリーで使用されます。

Zabbix Cloud

このセクションでは、Zabbix Cloud の組織およびノードで利用可能な設定オプションとアクセス制御について説明します。

- Zabbix Cloud で Zabbix をデプロイする
- ノードの設定
- ノードおよび組織へのユーザーの追加
- Zabbix Cloud とオンプレミスの主な違い

Note:
Zabbix Cloud は現在、Zabbix 7.0 で動作しています。
 現在の Zabbix Cloud 環境に関連する情報については、Zabbix 7.0 ドキュメントをご参照ください。

著作権表示

Zabbix のドキュメントは AGPL-3.0 ライセンスの下では配布されていません。
Zabbix ドキュメントの使用は、以下の条件に従うものとします。

このドキュメントの印刷物を作成できるのは、あくまで個人的な使用に限ります。
実際の内容がいかなる方法でも変更または編集されない限り、他の形式への変換は許可されます。
Zabbix が配布する方法（つまり、Zabbix の Web サイトでダウンロードできるように電子的に配布する方法）と同様の方法でドキュメントを配布する場合、または USB や同様の媒体で配布する場合を除き、いかなる形態または媒体でもこのドキュメントを公開または配布してはなりません。ただし、ドキュメントがソフトウェアと同じ媒体で一緒に配布される場合に限ります。
印刷物の配布や、このドキュメントの全部または一部を他の出版物で使用するなど、その他の使用には、Zabbix の認可された代表者から事前に書面による同意を得る必要があります。
Zabbix は、上記で明示的に付与されていないこのドキュメントに関するすべての権利を留保します。

1 Zabbix Cloud で Zabbix をデプロイする

はじめに

このページでは、Zabbix Cloud で以下の操作を行う方法について説明します。

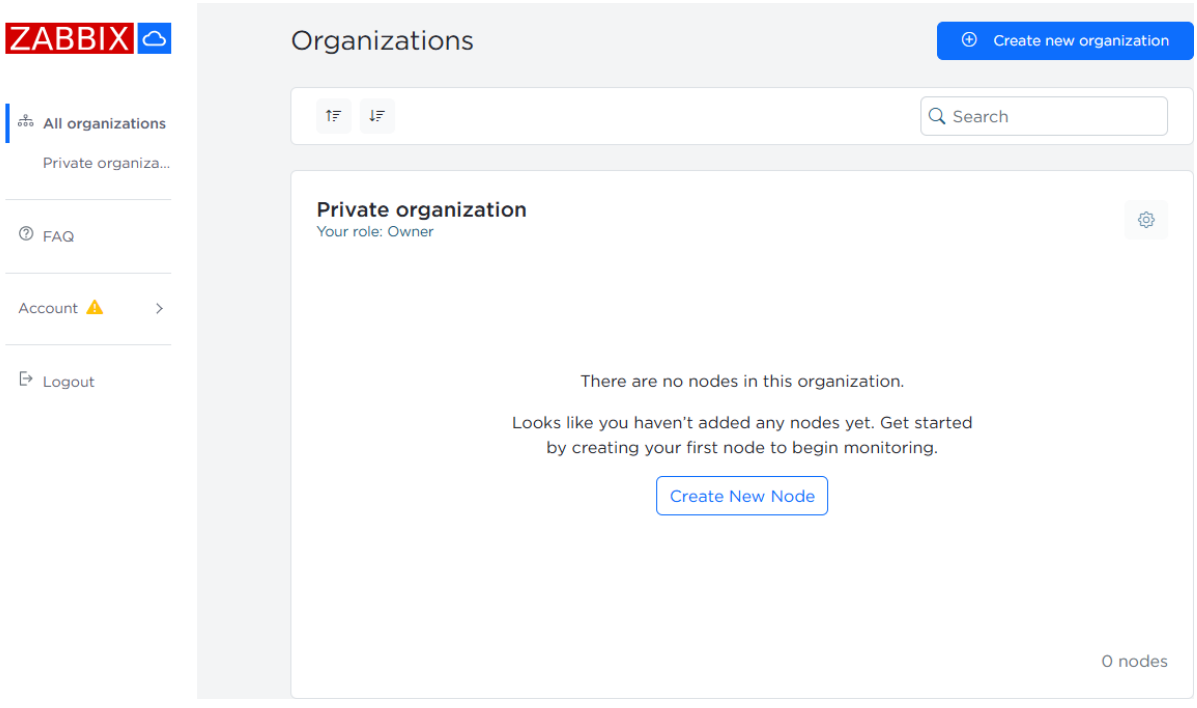
- 組織の管理
- ノードの作成と管理
- デプロイされたノードでの Zabbix フロントエンドへのアクセス

組織管理

組織は、1 つ以上のノードのコンテナです。組織は、リソースの構造化や責任の委譲にも役立ちます。例えば:

- ノード管理
- 課金
- ノードアクティビティの表示

Zabbix Cloud にログインすると、すべての組織ページが表示されます。各組織は、1 つ以上のノードのコンテナとして機能します。デフォルトでは、アカウント用にプライベート組織が作成されます。



この組織のオーナーとして、ユーザー、管理者、ノード、組織名を管理できます。管理オプションにアクセスするには、  設定ボタンをクリックします。

ノードの作成

ノードは、専用のクラウド環境で実行される自己完結型の Zabbix インスタンスです。ノードは以下を提供します。

- 一意の URL でアクセス可能な Zabbix サーバーとフロントエンド
- 専用のコンピュートリソース（CPU、メモリ）およびディスクストレージ
- 他のノードからの分離により、独立してテスト、監視、アップグレードが可能
- データ主権やレイテンシ最適化のためのリージョンベースのデプロイ

ノードを作成するには:

1. すべての組織ページで 新しいノードの作成をクリックします。

2. 新しいノードの作成ウィンドウで、以下を設定します。

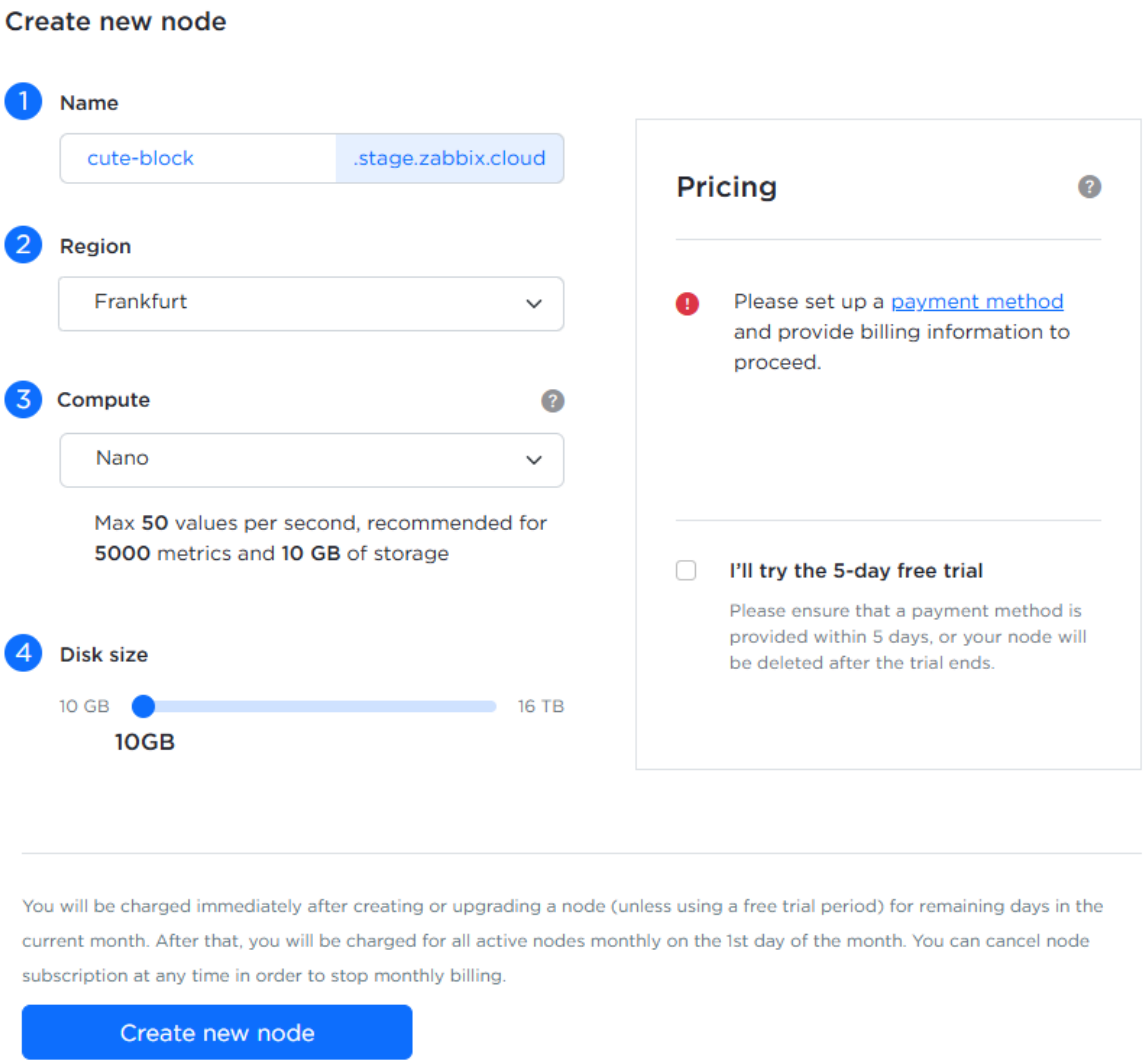
- 名前 - ノードの表示名を入力します。作成後、この名前は組織リストにクリック可能なリンクとして表示され、そのノードのフロントエンドに移動します。
- リージョン - 監視対象システムやサービスのエンドユーザーに最も近いデータセンターの場所を選択します。近くのリージョンを選択するとネットワークレイテンシが低減します。また、機密性の高いメトリクスを収集する場合はデータ主権の要件も考慮してください。

- コンピュート階層 - 予想される負荷に見合ったリソースレベル（CPU、メモリ）を選択します。必要なリソースレベルの判断には価格詳細を参照してください。より多くのリソースが必要になった場合は、後からアップグレードすることも可能です。

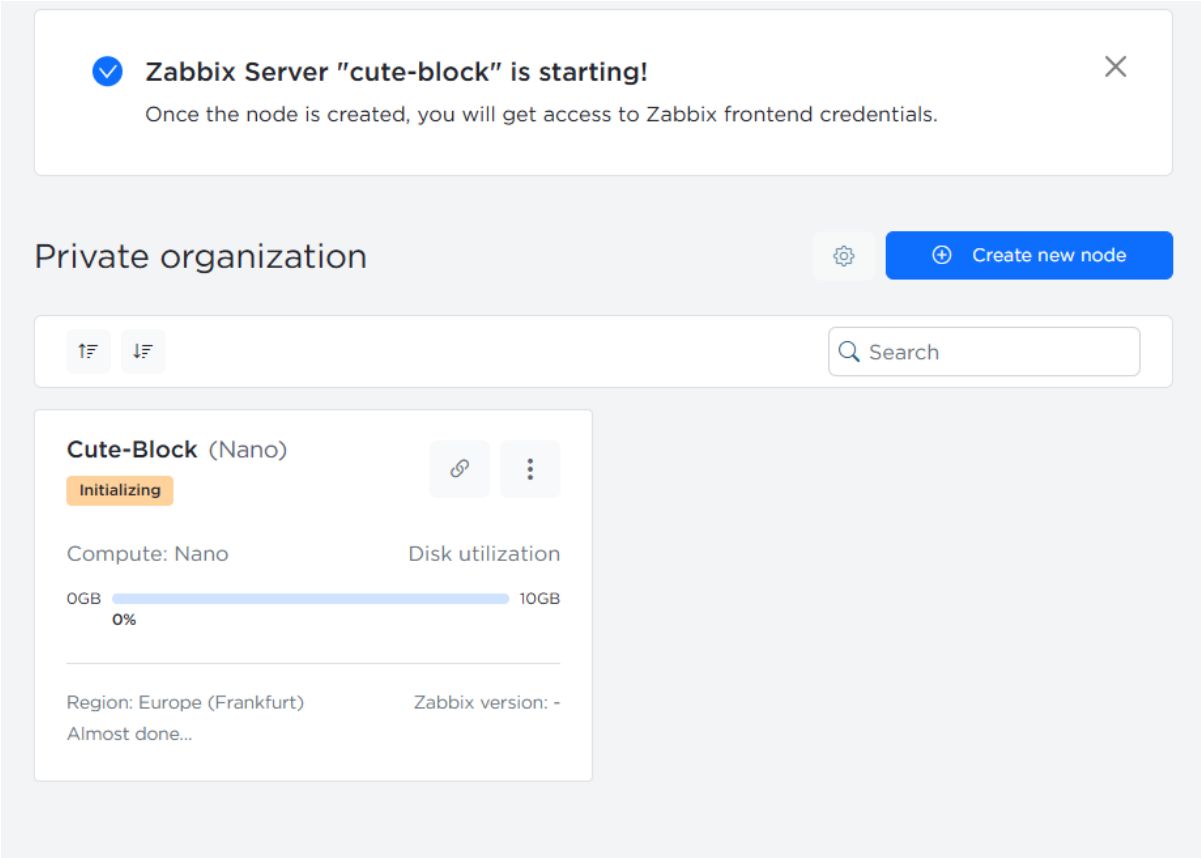
- ディスクサイズ - スライダーは階層に基づく推奨最小値がデフォルトです。大量の履歴やトレンドを保存する場合は増やしてください。大きなディスクは月額コストが高くなること、およびディスクサイズを自分で縮小することはできず、縮小にはサポートへの連絡が必要であることに注意してください。

必要に応じて、ノード作成前に 5 日間の無料トライアルを有効にできます。トライアルを有効にすると、Zabbix Cloud を無料で評価でき、請求情報も不要です。トライアル期間中は課金されませんが、終了後に有料ノードへ移行しない場合はノードが失効し削除されます。

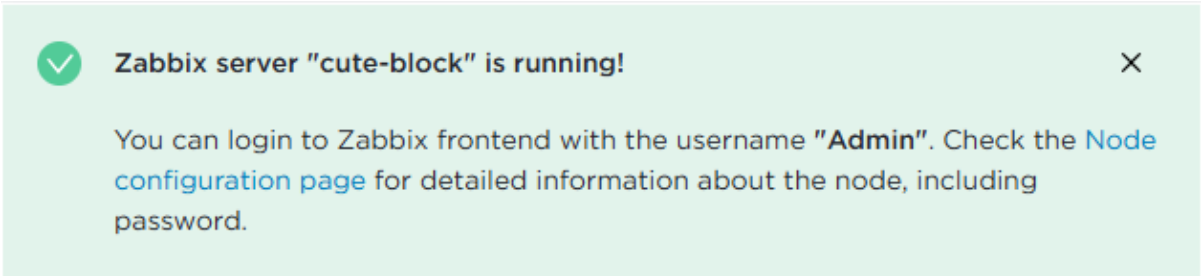
トライアル以外のノードを作成するには、事前に請求方法の登録が必要です。請求メニューから請求情報を追加してください。



作成後、ノードは組織ビューに表示されます:



ノードが稼働したら、ノード設定で設定管理（アクセスフィルタ、暗号化、バックアップなど）の詳細を確認してください。



ノードの初期化が完了すると、ステータスが 初期化中から 稼働中に変わり、そのノードの Zabbix フロントエンドにログインできるようになります。

Zabbix フロントエンドへのアクセス

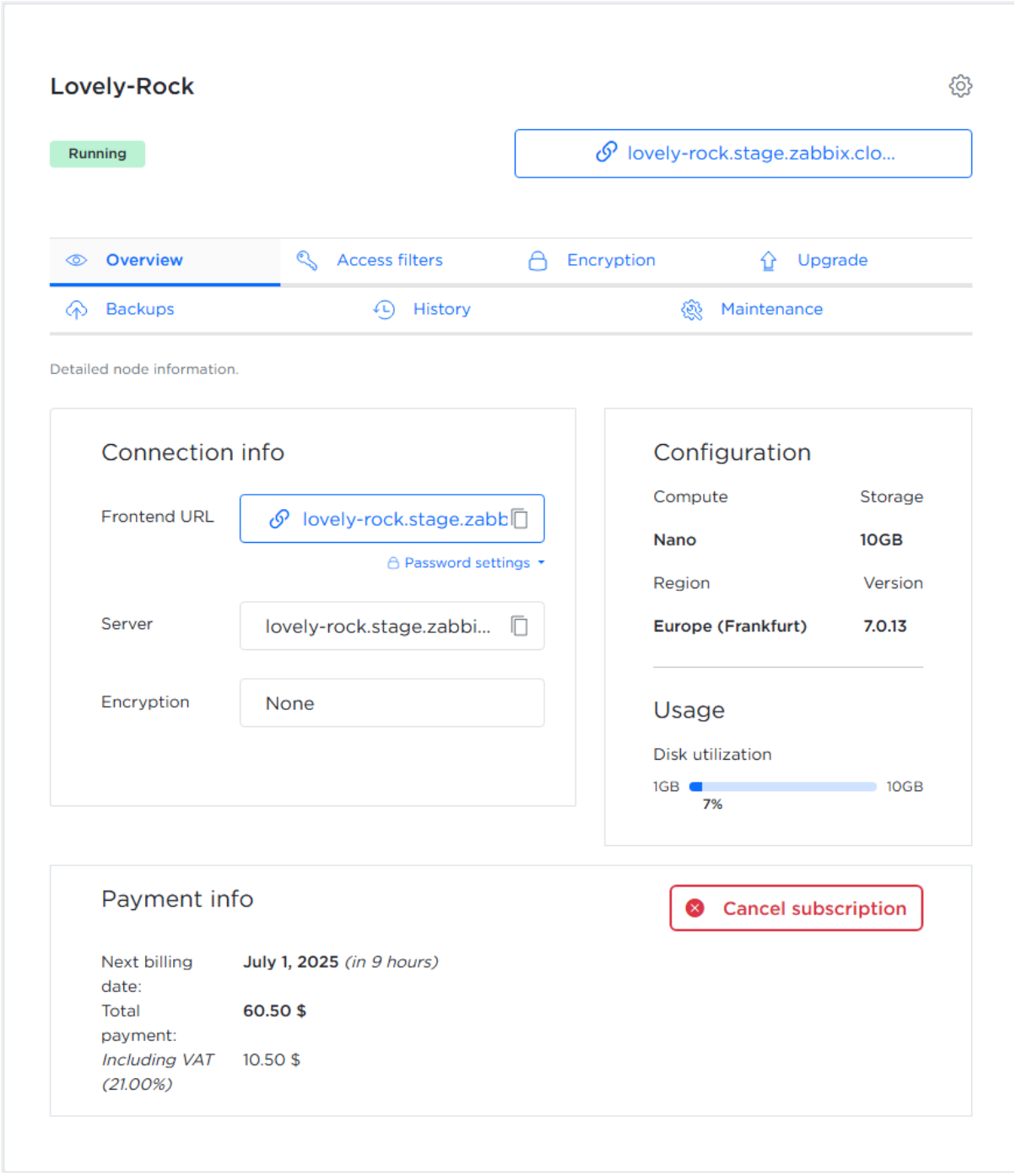
ノードで初めて Zabbix フロントエンドにアクセスするには:

1. すべての組織または自分の組織メニューでノードをクリックし、ノード設定ページを開きます。

2. 概要タブで、パスワード設定ドロップダウンをクリックします。

> **Note:**
> フロントエンドパスワードのコピー、クリア、リセットは所有者のみが実行できます。

3. 初期パスワードのコピーを選択します。



4. ノード設定ページで、フロントエンド URL（例: lovely-rock.stage.zabbix.cloud）をクリックして Zabbix フロントエンドを開きます。

次の認証情報を使用してログインします:

- ユーザー名 - Admin
- パスワード - 手順 4 でコピーしたパスワード（例: HzTG9t7Y）



関連情報:

- Zabbix Cloud FAQ - Zabbix Cloud の機能、導入、価格、サポートに関する追加情報やよくある質問への回答。

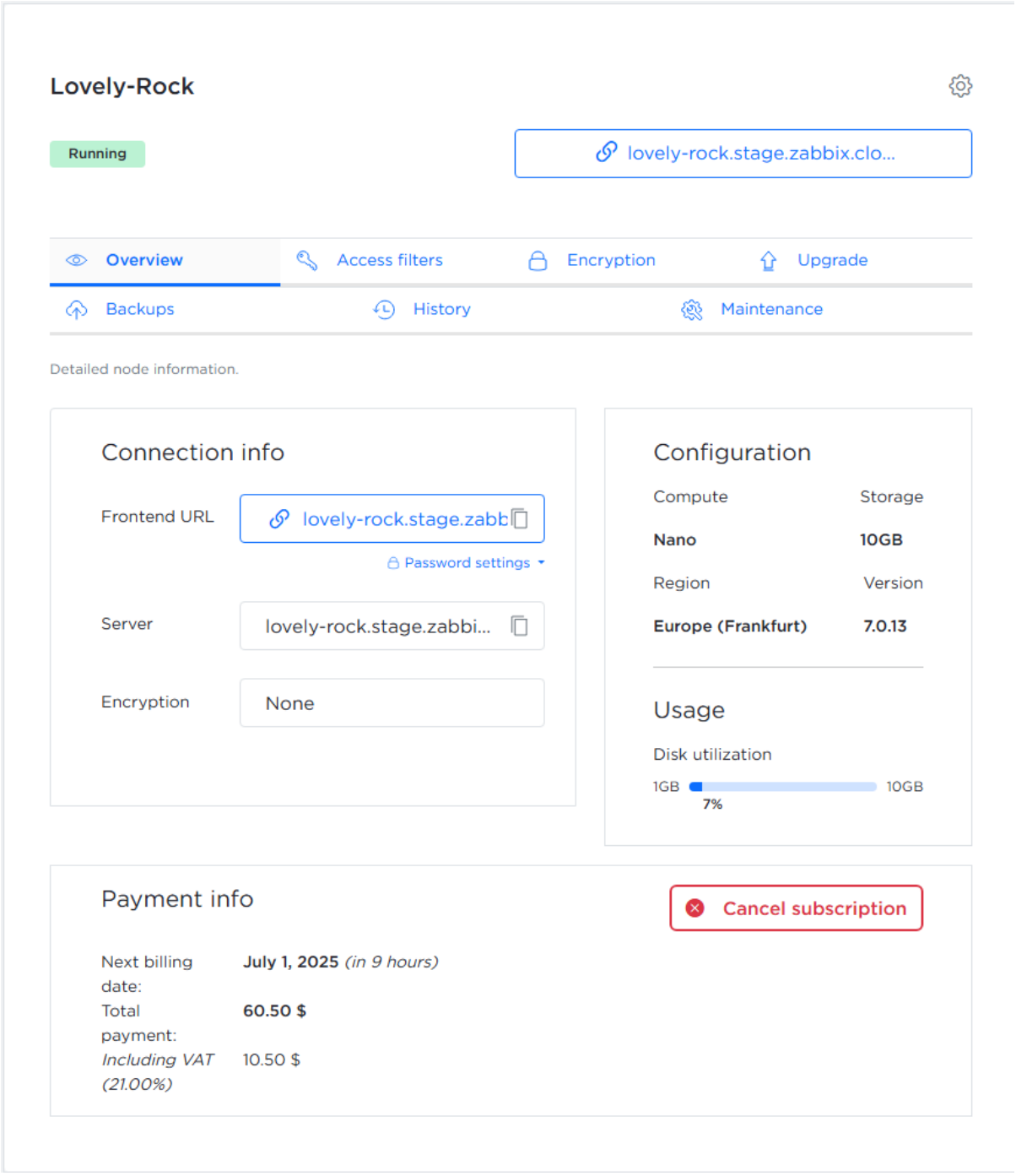
2 ノードの設定

概要

ノード設定ページにアクセスするには、すべての組織でノードをクリックするか、組織メニューから、またはノードを起動した後に表示されるノード設定ページリンクを使用します。

ノード設定ビューでは、各タブでノードの特定の側面を管理できます。

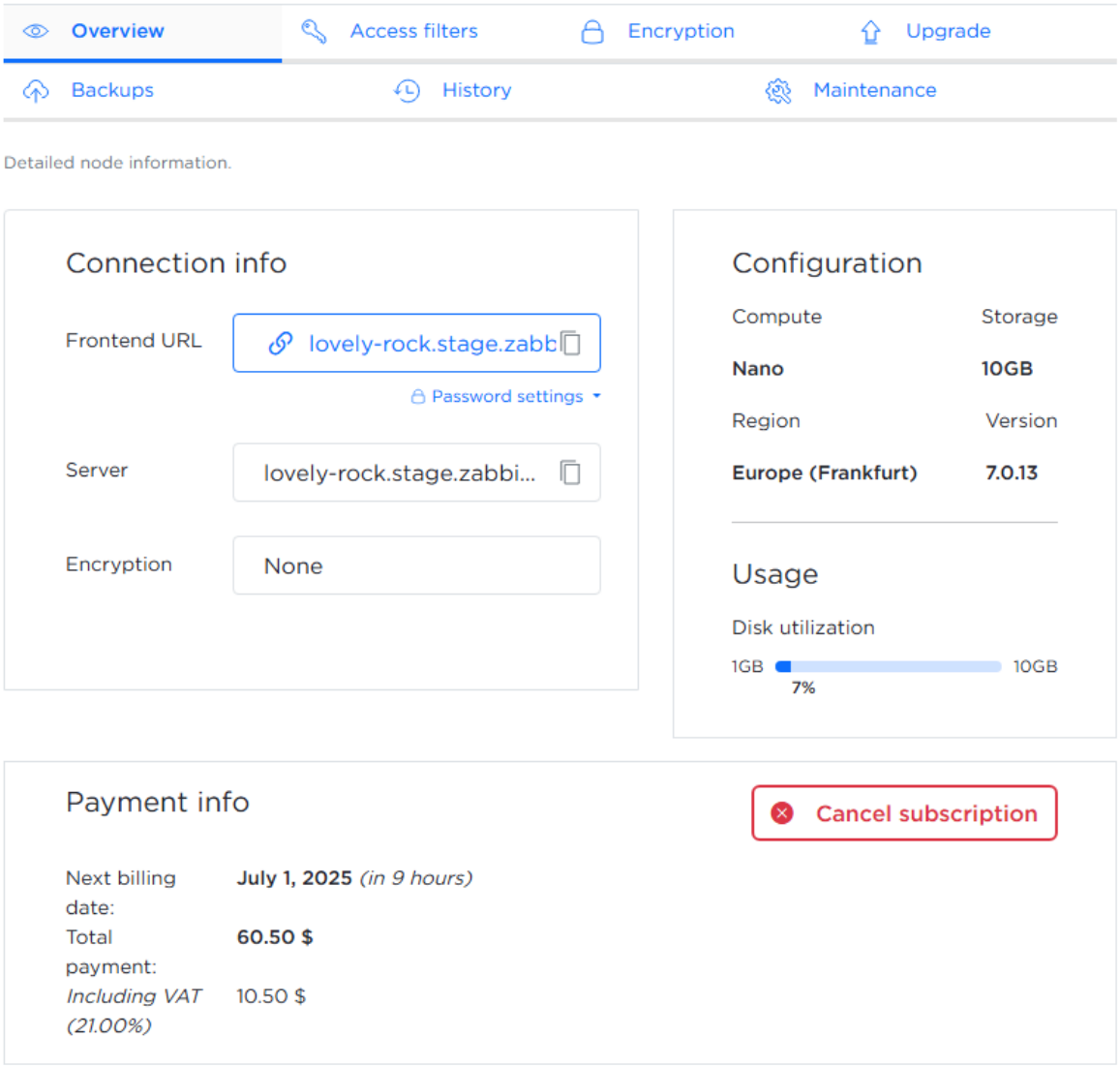
- 概要 - ノードの詳細を表示し、フロントエンドを開き、ノードユーザーを管理します。
- アクセスフィルター - IP アドレスまたはサブネットによるアクセスを制御します。
- 暗号化 - 証明書、キー、およびオプションの CRL をアップロードして TLS を構成します。
- バックアップ - ノードスナップショットの作成、復元、削除、次回のバックアップ予定の表示。
- 履歴 - 監査ログ、履歴、トレンドの保持期間を設定（単位はプリセット、1 日 3 回まで更新可能）。
- メンテナンス - タイムゾーン、曜日、時間で毎週のメンテナンスウィンドウを定義します。
- アップグレード - コンピュート/ストレージリソースのスケールアップやトライアル/支払いステータスの管理。



Note:
ページを表示するユーザーの種類やノードのステータス（トライアル/非トライアル）によっては、一部のタブが利用できない場合があります。

概要タブ

概要タブには、ノードに関する詳細情報が表示されます。



このタブは 3 つのセクションに分かれています。

接続情報:

- クリック可能なフロントエンド URL（クリップボードへのコピーオプション付き）
- パスワード設定
- 現在の暗号化方式

構成と使用状況:

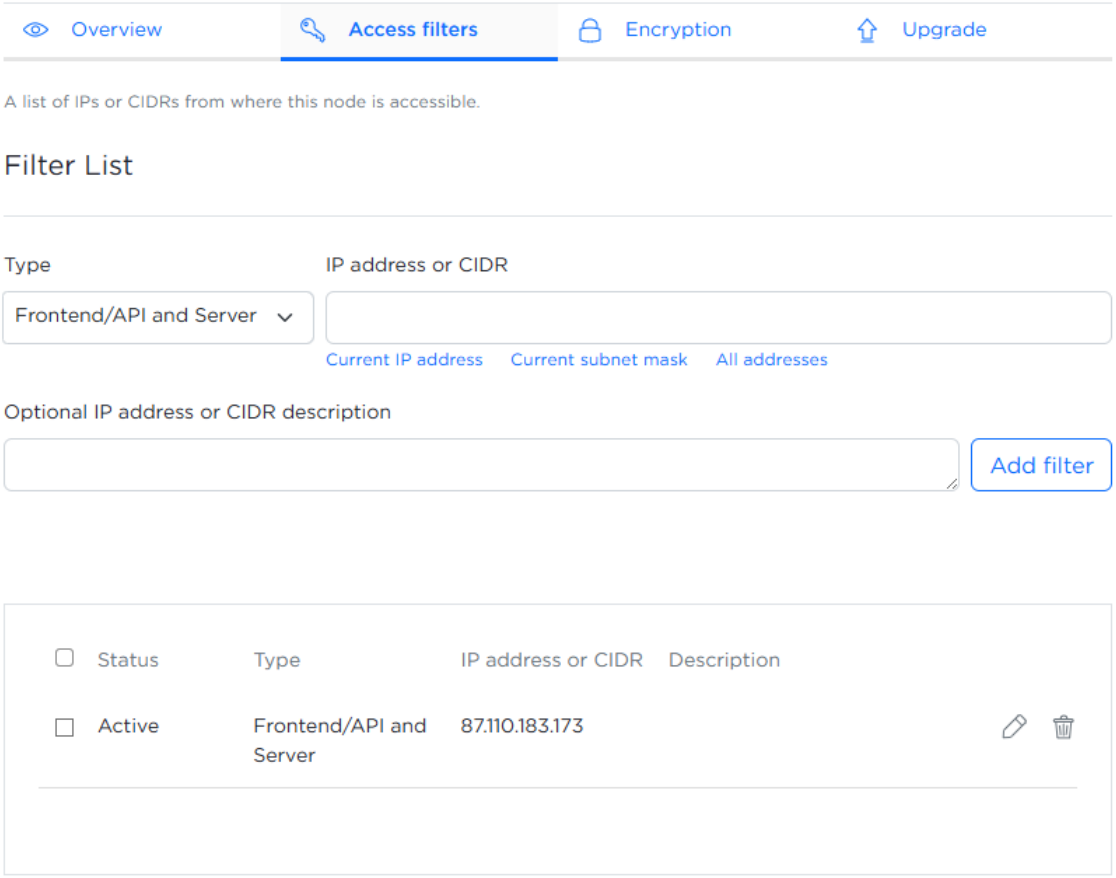
- コンピュートティア、最大ストレージ、リージョン
- Zabbix コンポーネントのバージョン
- ディスク使用率

ノード情報/支払い情報:

- 有効期限（トライアルおよびプリペイドノードの場合）
- ノードの初期化が完了した後、ノードの種類に応じてコンテキスト固有のボタンが表示されます: - 有料プランにアップグレード - 無料トライアル中のノードで利用可能 - サブスクリプションをキャンセル - 有効なサブスクリプションを持つ有料ノードで利用可能 - 有料プランに再登録 - サブスクリプションがキャンセルされた場合に表示 - プリペイドノードにはアクションボタンは表示されません（有効期限が長く、キャンセルできないため）

アクセスフィルタータブ

アクセスフィルタータブでは、IP アドレスまたはサブネットによって Zabbix ノードへのアクセスを制限できます。



アクセスフィルターを設定するには、以下のフィールドに入力します。

- タイプ - ドロップダウンから制限する Zabbix コンポーネントを選択します：フロントエンド/API のみ、サーバーのみ、または両方。
- *IP* アドレスまたは *CIDR* - IP アドレスまたは CIDR を手動で入力するか、関連するオプションを選択して自動的に入力します。
- *IP* アドレスまたは *CIDR* の説明（オプション） - 各 IP またはサブネットの説明を任意で追加します。

フィールドの入力が完了したら、フィルターを追加をクリックします。追加されたフィルターは下のテーブルに表示され、ステータス、タイプ、IP アドレスまたはサブネット、説明が示されます。

Warning:
ノードの初期化時に、使用した IP アドレスに対して自動的にアクセスフィルターが作成されます。後で別の IP またはネットワークから接続する場合は、そのアドレスをここに追加しないとブロックされます。

以下の操作が可能です。

-  - フィルターを編集します。
-  - チェックボックスを使用してフィルターを削除します（複数選択可）。

暗号化タブ

暗号化タブでは、必要な証明書ファイルをアップロードすることで、Zabbix コンポーネント間の TLS暗号化を設定できます。

証明書を使用した安全な通信を有効にするには、*CERTIFICATE* を選択し、コンピュータから以下のファイルをアップロードします:

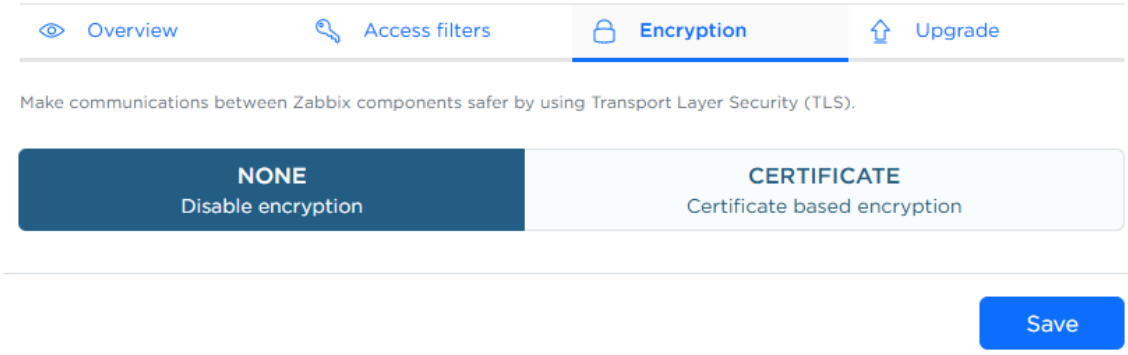
- *Root CA* 証明書 - ピア証明書の真正性を検証するために使用されます。
  ファイルを選択をクリックし、信頼できる Root CA 証明書を含むファイル（通常は `.crt` または `.pem` ファイル、PEM エンコード）を選択します。

- 証明書チェーン - 自分の証明書の信頼の連鎖を表します。
  ファイルを選択をクリックし、自分の証明書と中間 CA 証明書を含む PEM エンコードされたファイル（例：`.crt` または `.pem`）を選択します。

- 秘密鍵 - 自分の証明書に対応する秘密鍵です。
  ファイルを選択をクリックし、PEM エンコードされた.key ファイルを選択します。

- 失効証明書（オプション）- 証明書失効リスト（CRL）を管理している場合は、ファイルを選択をクリックし、もはや信頼すべきでない証明書を指定する CRL ファイルを選択します。

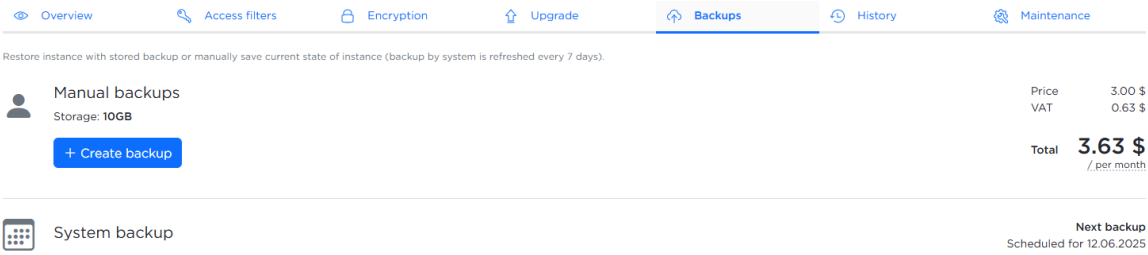
ファイルを選択したら、保存をクリックします。ファイルがアップロードされ、検証され、Zabbix コンポーネント間の TLS 暗号化を設定するために適用されます。



バックアップタブ

バックアップタブでは、Zabbix ノードのバックアップの表示、作成、復元、削除ができます。

保存されたバックアップからインスタンスを復元したり、いつでも新しい手動バックアップを作成したりできます。また、次回のスケジュールされたバックアップの日付も表示されます。



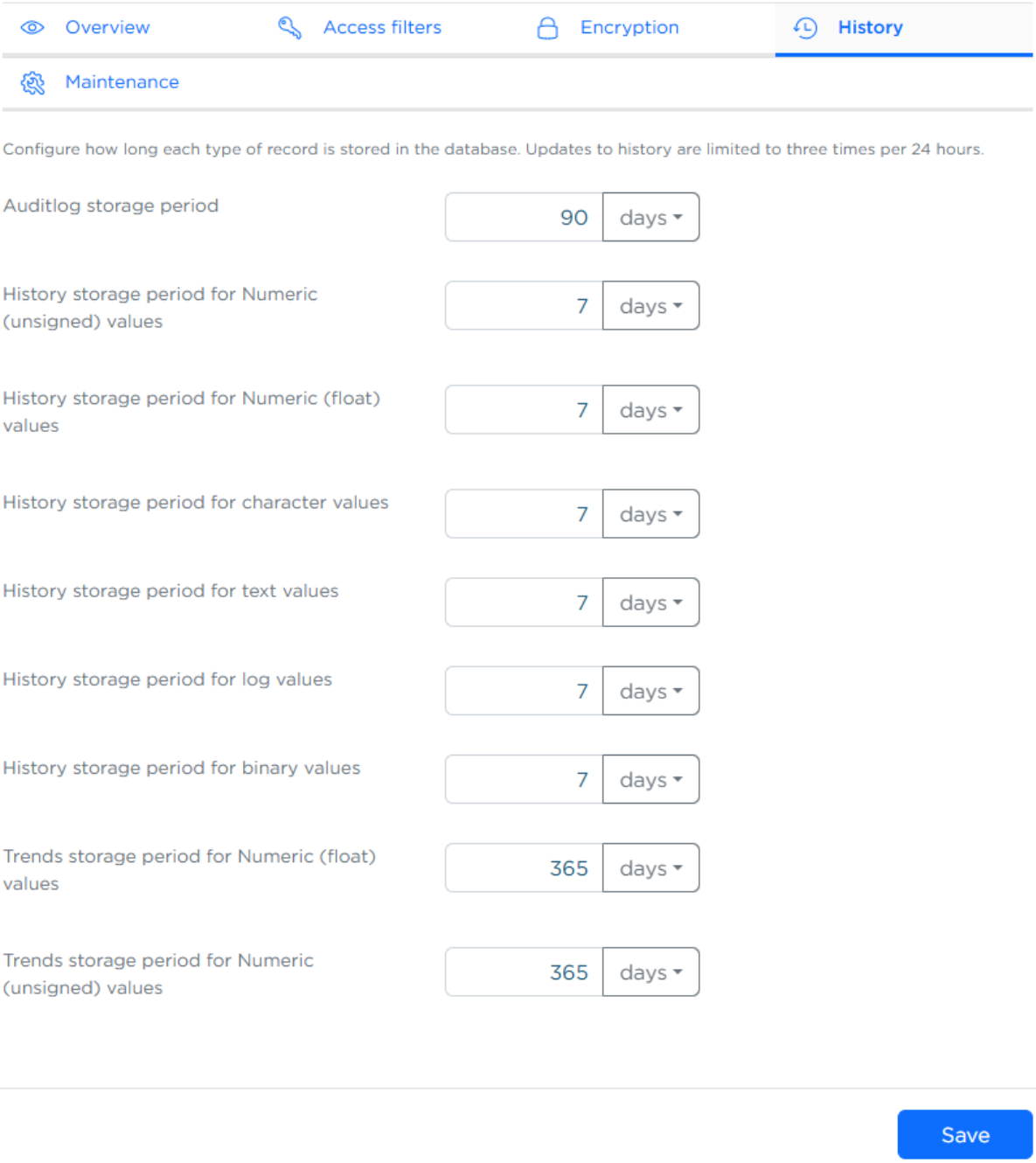
バックアップを作成するには:

1. バックアップの作成ボタンをクリックします。
2. ポップアップウィンドウで、任意で説明を入力します。
3. 作成をクリックして確定します。

Warning:
追加のバックアップは、ストレージ使用量に基づいて月額で課金されます。

履歴タブ

履歴タブでは、さまざまな種類の履歴データが保存される期間を設定できます。



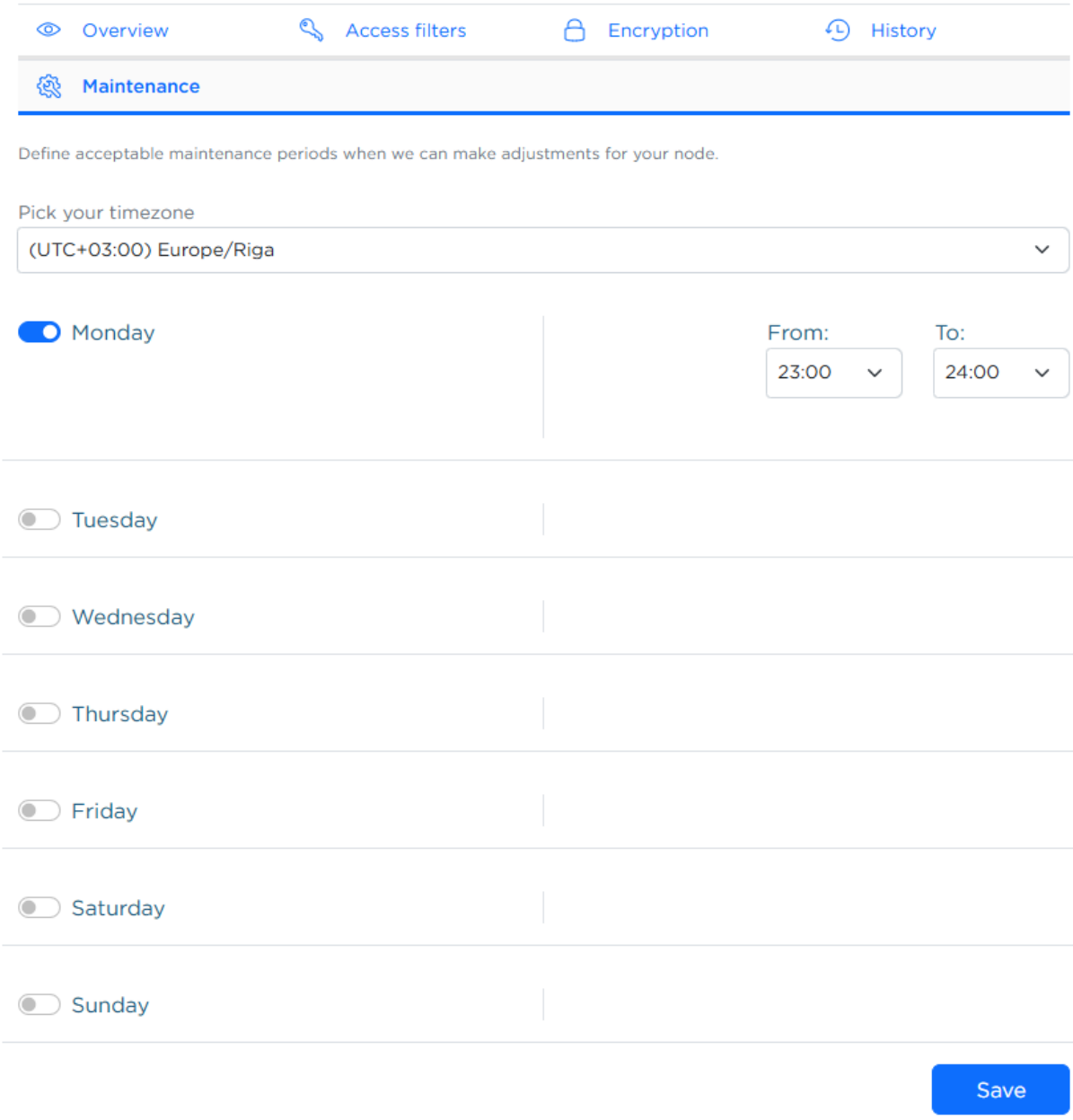
各データタイプ（数値、テキスト、ログなど）の保存期間を設定し、単位（日、月、年）を選択できます。値の変更は 24 時間ごとに最大 3 回まで可能です。

値を更新したら、保存をクリックして変更を適用します。

Warning:
保存期間を長くするとディスク使用量が増加し、パフォーマンスに影響を与える可能性があります。実際のデータ量や分析ニーズに基づいて保存期間を設定することを推奨します。
 Zabbix Cloud では、従来の Zabbix のハウスキーピングの代わりにデータベースパーティショニングを使用してデータを管理しています。これにより、より高速なパフォーマンスと予測可能な保存期間の管理が可能になります。

メンテナンスタブ

メンテナンスタブでは、ノードに対して更新や調整を実行できる許容可能なメンテナンスウィンドウを定義できます。

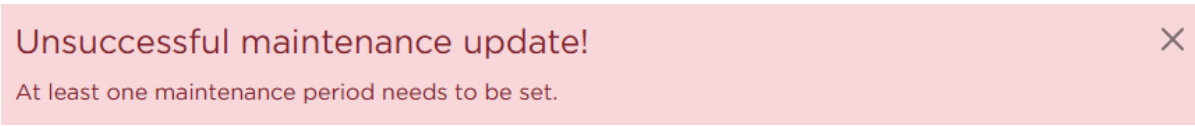


以下を行う必要があります:

- ドロップダウンからタイムゾーンを選択します。
- メンテナンスのために、少なくとも 1 つの曜日と 1 つの時間帯を選択します。

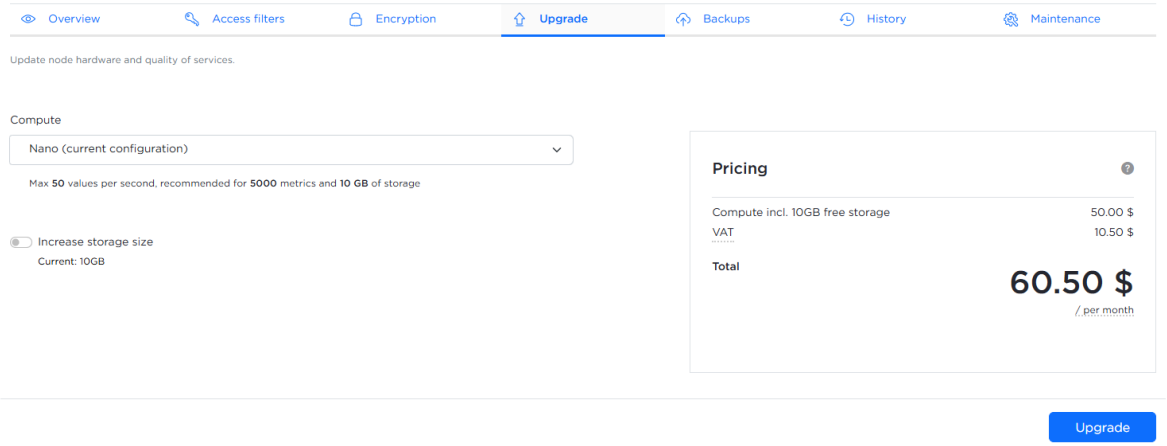
保存をクリックして変更を適用します。

曜日または時間帯が選択されていない場合、エラーが表示されます。



アップグレードタブ

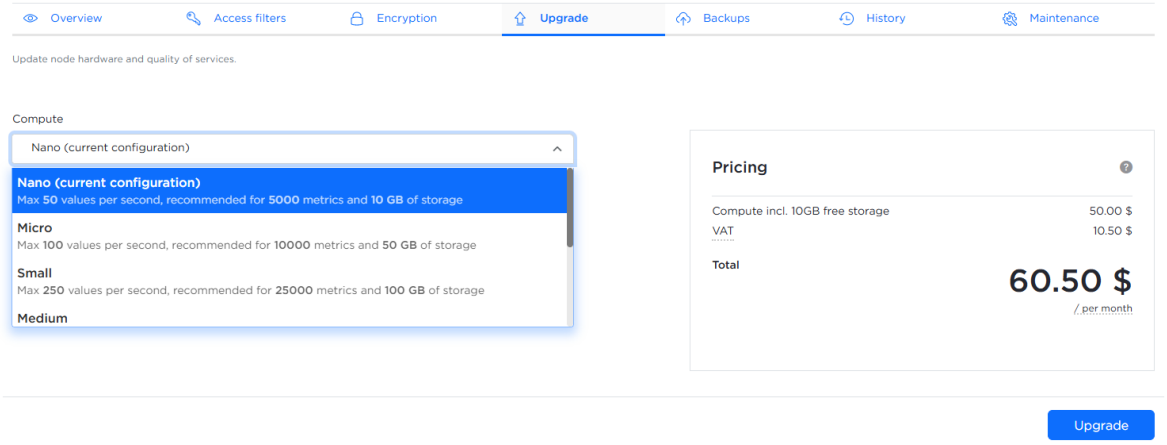
アップグレードタブでは、ノードのリソースのスケールアップやサブスクリプションの管理ができます。



コンピュートティアやディスクサイズを増やすことができます。ディスクサイズは 24 時間に 1 回のみ増加可能で、UI から減らすことはできません（縮小のご要望は Zabbix サポートまでご連絡ください）。無料トライアルノードは有料サブスクリプションに変更できます。

コンピュートティアを増やすには:

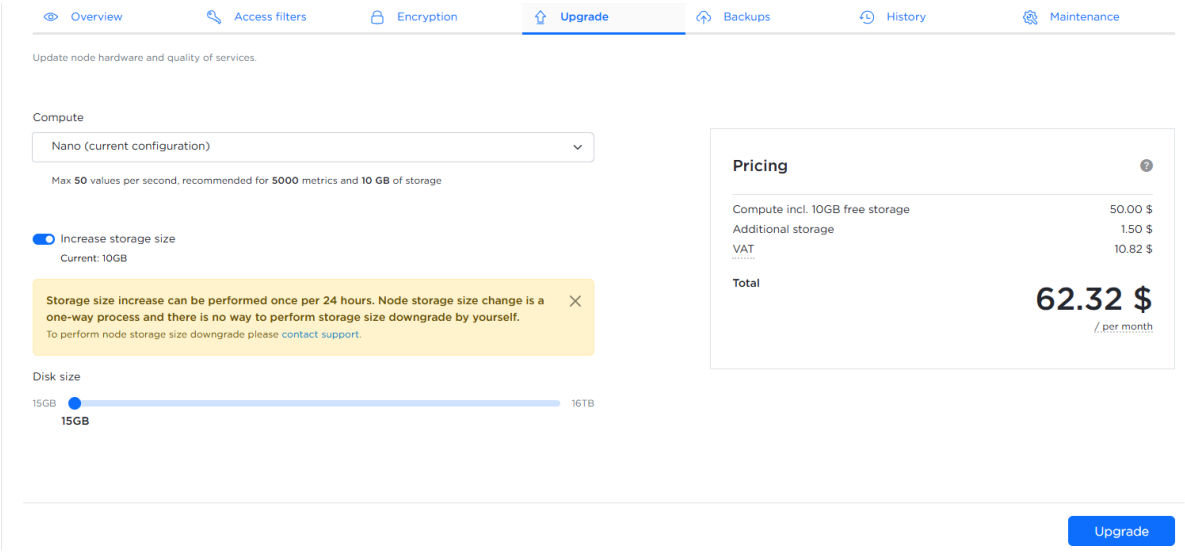
1. *Compute* ドロップダウンをクリックします。



2. 希望する構成ティアを選択します。
3. アップグレードボタンをクリックします。

ディスクサイズを増やすには:

1. ストレージサイズの増加ラジオボタンをクリックします。
2. ディスクサイズスライダーを希望の値に調整します。



3. アップグレードボタンをクリックします。

Note:
変更を確定する前に、価格テーブルに更新後の月額合計料金が表示されます。

3 ユーザーの追加

はじめに

以下の例では、組織やノードにユーザーを追加する方法について説明します。

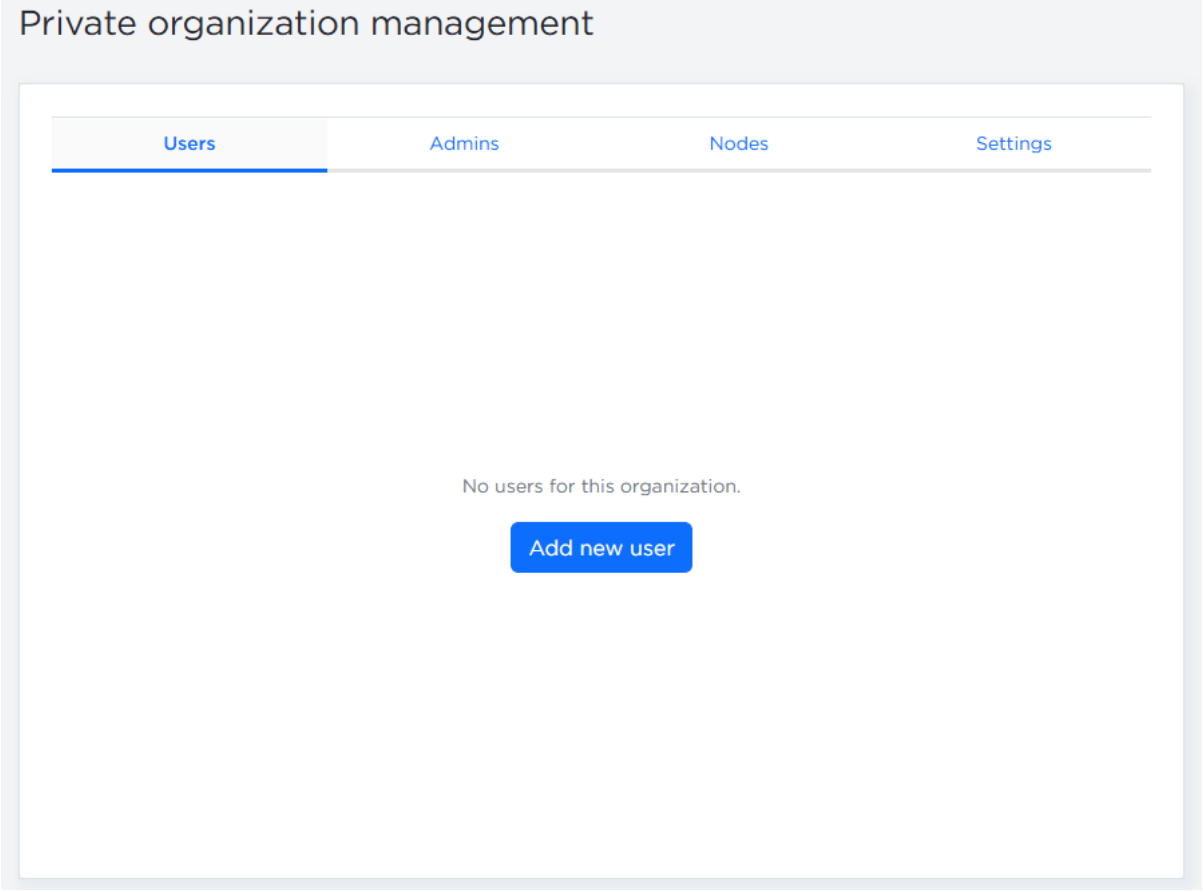
組織へのユーザー追加

Zabbix のユーザーと同様に、Zabbix Cloud でもユーザーを作成し、組織やそのノード内で何ができるかを制御するためのロールを割り当てることができます。

組織にユーザーを追加するには:

1. すべての組織ページを開きます。

2. 特定の組織の 設定ボタンをクリックして、管理メニューに入ります。



3. ユーザーまたは管理者タブで、新しいユーザーを追加をクリックします。

4. ユーザーのメールアドレスを入力し、管理者またはユーザーのロールを選択します。

ユーザーロールを選択すると、そのユーザーは組織に追加されますが、いずれのノードへのアクセス権も付与されませんのでご注意ください。

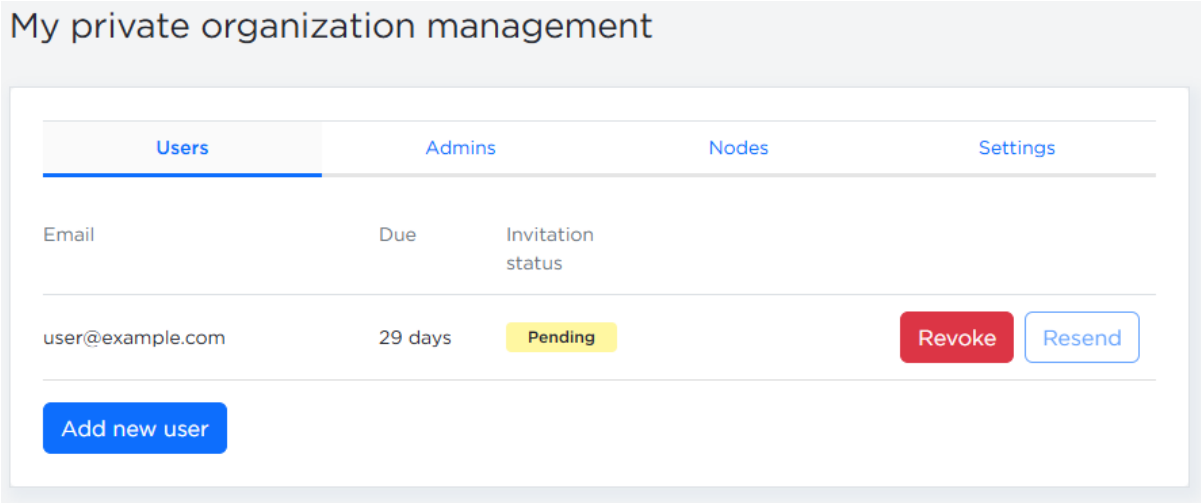
各ロールは異なるレベルのアクセス権を付与します:

- ユーザー - 割り当てられたノードにアクセスでき、課金が発生しない設定作業を実行できます。例:
  - アクセスフィルターの管理
  - 暗号化の設定
  - 履歴/トレンドの保持期間の設定
  - メンテナンスのスケジューリング

- 管理者 - ユーザーのすべての権限に加え、課金が発生する可能性のある操作も実行できます。例:
  - バックアップの作成と復元
  - ノードリソース（コンピュート階層やディスクサイズ）のアップグレードまたはダウングレード
  - サブスクリプションのキャンセル
- オーナー - 組織を完全に管理できます。管理者のすべての権限に加え、オーナーは以下も可能です:
  - ノードの作成と削除
  - ユーザーや管理者の追加・削除
  - 組織名の変更や削除
  - 請求情報の管理

> **Attention:**
> オーナーの支払い方法で、すべてのサービスの料金が請求されます。銀行の確認が必要な場合は、管理者が操作を開始した場合でも、オーナーが取引を承認する必要があります。

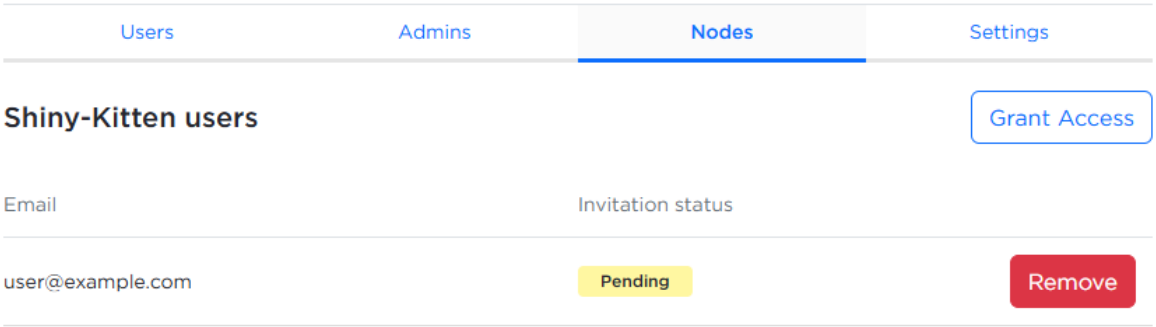
招待されたユーザーは保留中のステータスで追加されます。
招待を取り消したり、再送信したりできます（再送信は 10 分に 1 回のみ可能で、この制限に達するとボタンが無効になります）。



ノードへのユーザー追加

ノードにユーザーを追加するには:

1. ノードタブに移動します。
2. アクセス権を付与をクリックします。
3. ドロップダウンから組織に追加したユーザーを選択します。
4. アクセス権を付与をクリックします。



または、ノード設定ウィンドウからアクセス権を付与することもできます:

1.  設定ボタンをクリックします。
2. ユーザーを追加を選択します。
3. ユーザーを選択し、アクセス権を付与をクリックします。

招待されたユーザーは、招待が送信されたのと同じメールアドレスを使用して Zabbix Cloud アカウントを作成し、招待を受け入れる必要があります。すでにそのメールアドレスで Zabbix Cloud アカウントを持っている場合は、サインインして招待を確認する必要があります。招待は特定のメールアドレスに紐付けられており、別のアカウントから受け入れることはできません。

4 Zabbix Cloud とオンプレミスの主な違い

はじめに

このページでは、Zabbix Cloud をオンプレミスインストールと比較した場合の主な制限事項や非サポート機能について説明します。以下に概要比較表を示します。続くセクションでは、選択された項目について詳細に説明します。

機能 / 項目	Zabbix Cloud	オンプレミス Zabbix
脆弱性パッチ適用	OS/Zabbix の自動パッチ適用	ユーザーによる手動パッチ適用
HTTPS 証明書	有効な証明書を標準搭載（Let's Encrypt なし）	デフォルトは自己署名（Let's Encrypt または他の CA を設定可能）
DB パフォーマンス (INSERT/UPDATE/SELECT)	最大速度のための自動パーティショニング；長時間の DELETE は発生しない	手動パーティショニング＆クリーンアップ (長時間の DELETE トランザクションが発生する場合あり)
パフォーマンスチューニング	Cloud チームによる自動チューニング（変更履歴なし）	チューニングパラメータの完全な可視化と制御
使いやすさ	ターンキーサービス	インストール、設定、メンテナンスが必要
ファイアウォール制御	Cloud チームが管理；GUI/trapper ポートは外部公開されない	ファイアウォールルールとポート公開の完全な制御
DB の分離と保護	高度に分離、DB/SSH への直接アクセス不可	ネットワークやホストのセキュリティに依存
リリースサイクル	LTS リリースに固定	Zabbix バージョン（安定版、ベータ版、カスタムビルド）を選択可能
高可用性（HA）	アプリケーション、フロントエンド、DB サーバーにHAは組み込みなし；プロキシによる水平スケールは可能	ユーザー管理の HA
SNMP トラップ	専用プロキシ経由のみ（HA/負荷分散トラップ受信機能なし）	直接またはプロキシ経由、HA/負荷分散可能
SNMP ポーリング	プロキシなしでは困難（各デバイスに NAT/カスタムポートが必要）	ネイティブなSNMP ポーリング；プロキシは任意
保持設定	履歴/トレンド/監査は UI のみで設定可能（API や設定ファイル不可）	Zabbix サーバー設定ファイルまたはAPIで設定可能
カスタムスクリプト	`AlertScriptsPath`、`ExternalScripts`、フロントエンド＆コミュニティモジュールは非サポート	完全サポート（スクリプトパス、モジュール、インテグレーション）
ODBC 監視	PostgreSQL のみ（ドライバ `{postgresql}`）；MariaDB プラグインはプレースホルダー	PostgreSQL、MySQL、Oracle などの ODBC を完全に設定可能
ODBC コールの制限	不可（`StartODBCPollers=1` のみ；重い同期クエリは他の作業をブロック）	ポーラー数とスケジューリングを完全に調整可能
SAML 証明書アップロード	非サポート	UI または API でサポート
定期レポート	メールメディアを自作する必要あり（スクリプトメディアは非サポート）	スクリプトとメールメディアの両方を標準サポート
アクティブチェック用ホストインターフェース	自動作成される IP が外部の場合あり；一貫性のため手動クリーンアップが必要	インターフェースはユーザー管理；IP もユーザー管理

機能の違い

SNMP トラップ　SNMP トラップは、専用の Zabbix プロキシを介してのみサポートされます。SNMP トラップ監視が必要な場合、SNMP トラップは単一の IP アドレスに送信する必要があるため、プロキシの自動負荷分散や高可用性は使用できません。

SNMP ポーリング　プロキシを使用しない場合、SNMP ポーリングには各デバイスを NAT およびカスタムポートで公開する必要があり、大規模な SNMP ポーリングが困難になります。

保持設定　履歴、トレンド、監査ログの保持期間の設定は、Zabbix Cloud UI からのみ可能です。これらの設定は `zabbix_server.conf` や API からは設定できません。

アイテムごとの履歴保持期間の手動上書きはサポートされていません（パーティショニングはグローバルに制御されます）。また、Cloud ノードの URL ではオンプレミスインストールと同じクエリパラメータは使用できません。

カスタムスクリプト　Zabbix Cloud では、以下のカスタムスクリプトタイプはサポートされていません。

- アラートスクリプト (`AlertScriptsPath`)
- 外部スクリプト (`ExternalScripts`)
- Zabbix Server 上のフロントエンドスクリプト

コミュニティが開発したフロントエンドモジュールはインストールできません。

ODBC 監視　Zabbix Cloud はODBC 監視を PostgreSQL のみサポートしています。公式の Zabbix ODBC テンプレートを使用し、テンプレート内で接続文字列を以下のように定義してください:

```
Driver={postgresql}
```

MariaDB プラグインもインストールされていますが、現在は動作しません。使用する場合は、以下のように定義してください:

```
Driver={mysql}
```

Attention:
Zabbix Cloud で Oracle MySQL 8.0 の動作する既知の構成はありません。単純なクエリ（SELECT 1 など）は成功する場合がありますが、より複雑なクエリは SQL_ERROR を返します。

 ODBC コールの制限（StartODBCPollers=1 のみ）はできません。重い同期レポートはパフォーマンスに影響を与える可能性があり、同時に実行できる SELECT は 1 つだけです。

インフラストラクチャへのアクセス　Zabbix Cloud は、基盤となるノードへの SSH アクセスを提供せず、またデータベースへの直接接続（例えば、ポート 3306 への接続）も許可しません。すべての設定、監視、およびトラブルシューティングの作業は、Cloud UI または API を介して実行する必要があり、OS およびデータベース層が分離され、安全に保たれるようになっています。

アクティブチェックインターフェース　Cloud のアクティブチェックは、オンプレミスネットワークとは無関係な IP でホストインターフェースを自動的に作成します。デフォルトでは、この IP は外部のものになる場合があります。一貫性を保つために、作成後にホストインターフェースを手動で削除または調整できます。

SAML 証明書のアップロード　暗号化されていない、または署名されていない SAML メッセージを受け入れるプロバイダーはほとんどいないため、SAML 認証はサポートされていません。

スケジュールレポート　デフォルトの Cloud Email メディアタイプはスクリプトトランスポートを使用しており、スケジュールレポートには使用できません。レポートをメールで送信するには:

1. 新しい Email メディアタイプを作成します。
2. ユーザー設定 > メディアで、作成した Email メディアを自分のユーザーに割り当てます。
3. スケジュールレポートの設定時に、そのメディアタイプを選択します。

Zabbix agent 2　Zabbix Cloud ノードでは Zabbix agent 2 は実行されていないため、`web.certificate.get` アイテムを使用した HTTPS証明書監視など、Zabbix agent 2 を必要とするチェックは利用できません。

デベロッパーセンター

このセクションには、カスタム Zabbix 機能拡張の開発をすぐに開始するために必要なものがすべて含まれています。

- Web インターフェースモジュール
- ダッシュボードウィジェット
- Zabbix エージェント向けプラグイン

著作権表示

Zabbix のドキュメントは、AGPL-3.0 ライセンスの下で配布されています。Zabbix ドキュメントの使用には、次の条件が適用されます。

個人的な利用の場合、このドキュメントの印刷物を作成することができます。実際の内容が変更または編集されない限り、他の形式への変換は許可されます。Zabbix が配布する方法と同様の方法（つまり、Zabbix Web サイトから電子的にダウンロードする）または USB などの媒体でドキュメントを配布する場合を除き、いかなる形式、メディアでも、このドキュメントを公開、配布してはなりません。ただし、ドキュメントがソフトウェアと一緒に同じ媒体で配布されることを条件とします。印刷物の配布や、このドキュメ

ントの全体または一部を別の出版物で使用するなどのその他の利用には、Zabbix の正式な代表者からの事前の書面による同意が必要です。Zabbix は、上記で明示的に付与されていない、このドキュメントに対するすべての権利を留保します。

モジュール

PHP フロントエンドモジュールとは

- モジュールは、マニフェストファイルで定義された一意の ID、名前、説明、作成者、およびその他のフィールドを持つエンティティであり、PHP、Javascript、およびその他のファイルは、Zabbix フロントエンドインストールの *modules* ディレクトリ内の 1 つのディレクトリ (例: *zabbix/ui/modules*) に配置されます。
- モジュールは、正しい操作を保証するために、単純なルールに準拠する必要があります。
- モジュールは、管理者によってフロントエンドにインストール (解凍) され、有効にされる必要があります。

モジュールの用途

- カスタムフロントエンドセクションを介して新しい機能を追加する。
- カスタムダッシュボードウィジェットタイプを作成する (ウィジェットモジュールを参照)。
- 既存の機能をオーバーライドまたは拡張する。

モジュールが使用できない用途

- 新しい API メソッドの登録、または既存のメソッドの変更。

モジュールの動作

- アクションコードを実行する前に、各 HTTP リクエストで有効なモジュールが起動されます。
- モジュールは新しいアクションを登録するか、既存のアクションを再定義します。
- モジュールは新しいフロントエンドセクションを追加し、既存のセクションを削除または再定義します。
- モジュールは必要に応じて、onBeforeAction や onTerminate などのフロントエンドイベントにフックします。
- 要求されたアクションは、最終的にアクションコード (デフォルトのアクションまたはモジュール定義のアクションコード) を実行することによって実行されます。

次に進むべき道　実際にやってみるか、最初にガイドラインを読むかに関わらず、これらのページには独自のモジュールを作成するために必要な情報と手順が記載されています。

- 最初のモジュールを作成するためのステップバイステップのチュートリアル
- モジュールファイル構造
- ウィジェットモジュールの詳細
- 再利用するモジュールの例

モジュールファイルの構造

モジュールに関連するすべてのコードは、Zabbix フロントエンドインストールの **modules** ディレクトリ内の 1 つのディレクトリに保存されます (例: *zabbix/ui/modules*)。

```
example_module_directory/           (必須)
   manifest.json                    (必須)   メタデータとアクションの定義。
   Module.php                                 モジュールの初期化とイベント処理。
   actions/                                   アクションコントローラーファイル。
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                     ビューファイル。
       example.something.view.php
       example.something.delete.php
   assets/                                    ビューで使用する追加ファイル。manifest.json で指定する必要があります。
```

```
        js/                                     ビューで使用する JavaScript ファイル。
            example.something.view.js.php
        css/                                    ビューで使用する CSS ファイル。
            example.something.css
         image.png                              ビューで使用する画像。
         example.something.file                 ビューで使用する任意のファイル。
```

モジュールファイルツリー

モジュールの作成　サンプルモジュールの作成の流れは次の通りです (利用可能な場合は、ファイル名またはフォルダー名をクリックすると、手順の詳細が表示されます)。

1. **zabbix/ui/modules/**内にモジュール用の新しいディレクトリを作成します。
2. モジュールメタデータを含むmanifest.jsonファイルを追加します。
3. *views*フォルダーを作成し、モジュールビューを定義します。
4. *actions*フォルダーを作成し、モジュールアクションを定義します。
5. Module.php (またはダッシュボードウィジェットの場合は Widget.php) ファイルを作成し、初期化およびイベント処理ルールを定義します。
6. JavaScript ファイル (*assets/js* に配置)、CSS スタイル (*assets/css* に配置)、またはその他の追加ファイル用の*assets*フォルダーを作成します。
7. manifest.json で必要なビュー、アクション、アセットファイルを指定していることを確認します。
8. Zabbix フロントエンドでモジュールを登録して、使用を開始します。

Note:
manifest.json ファイルを作成するとすぐに、モジュールを登録して有効にすることができます。モジュールが有効になると、Zabbix フロントエンドを更新することで、モジュールファイルに加えられたすべての変更をすぐにプレビューできます。

manifest.json

すべてのモジュールには、manifest.json ファイルが必要です。ファイルはモジュールのプライマリディレクトリに存在する必要があります (例: *zabbix/ui/modules/module_name/manifest.json*)。

最低限として、manifest.json では次のフィールドを指定する必要があります。

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

manifest.json でサポートされているパラメーター (パラメーター名をクリックすると詳細な説明が表示されます):

パラメーター	説明	必須
manifest_version	モジュールのマニフェストバージョン。	Yes
id	一意のモジュール ID。	
name	管理セクションに表示されるモジュール名。	
namespace	モジュールクラスの PHP 名前空間。	
version	モジュールのバージョン。	
type	モジュールのタイプ。ウィジェットの場合は *widget* と設定する必要があります。	Yes (ウィジェットのみ)
widget	ウィジェットの設定。ウィジェットのみに使用されます。	
actions	モジュールに登録するアクション。	
assets	含める CSS スタイルと JavaScript ファイル。	No
author	モジュールの作成者。	
config	カスタムモジュールオプションのデフォルト値。	
description	モジュールの説明。	
url	モジュールの説明へのリンク。	

manifest_version

モジュールのマニフェストバージョン。現在サポートされているバージョンは **2.0** です。

タイプ: Double

例:

```
"manifest_version": 2.0
```

id

モジュール ID。一意である必要があります。将来的に名前の競合が発生しないように、モジュールにはプレフィックス (作成者名、会社名など) を使用することをお勧めします。たとえば、レッスンのサンプルのモジュールで、モジュール名が”My module” の場合、ID は”example_my_module” になります。

タイプ: String

例:

```
"id": "example_my_module"
```

name

管理セクションに表示されるモジュール名。

タイプ: String

例:

```
"name": "My module"
```

namespace

モジュールクラスの PHP 名前空間。

タイプ: String

例:

```
"namespace": "ClockWidget"
```

version

モジュールのバージョン。バージョンは管理セクションに表示されます。

タイプ: String

例:

```
"version": "1.0"
```

type

モジュールのタイプ。ウィジェットの場合必須で、”widget” と等しくなければなりません。

タイプ: String

デフォルト: ”module”

例:

```
"type": "widget"
```

actions

モジュールに登録するアクション。各アクションの *class* オブジェクトキーの定義は必須ですが、他のアクションキーはオプションです。

タイプ: Object

*type*が *module* の場合にサポートされるオブジェクトキー:

- **write.your.action.name** (object) - アクション名は小文字 [a-z] で記述し、単語をドットで区切ります。サポートするキー:
  - **class** (string; 必須) - アクションクラス名。
  - **layout** (string) - アクションレイアウト。サポートする値: *layout.json*、*layout.htmlpage* (デフォルト)、*null*。
  - **view** (string) - アクションビュー。

例:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
```

```
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

*type*が *widget* の場合にサポートされるオブジェクトキー:

- **widget.{id}.view** (object) - ウィジェットビューのファイルとクラス名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.view*)。サポートするキー:
  - **class** (string; 必須) - デフォルトの CControllerDashboardWidgetView クラスを拡張するウィジェットビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。
  - **view** (string) - ウィジェットビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.view.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **widget.{id}.edit** (object) - ウィジェット設定ビューのファイル名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.edit*)。サポートするキー:
  - **class** (string; 必須) - ウィジェット設定ビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。
  - **view** (string) - ウィジェット設定ビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.edit.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。

例:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

含める CSS スタイルと JavaScript ファイル。

タイプ: Object

サポートするオブジェクトキー:

- **css** (array) - 含める CSS ファイル。ファイルは *assets/css* に配置する必要があります。
- **js** (array) - 含める JavaScript ファイル。ファイルは *assets/js* に配置する必要があります。

例:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

author

モジュールの作成者。作成者は管理セクションに表示されます。

タイプ: String

例:

```
"author": "John Smith"
```

config

モジュールオプションのデフォルト値。オブジェクトにはカスタムキーが含まれる場合があります。指定した場合、これらの値はモジュールの登録中にデータベースに書き込まれます。後で追加された新しい変数は、最初の呼び出し時に書き込まれます。その後、変数値はデータベース内でのみ直接変更できます。

タイプ: Object

例:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

description

モジュールの説明。

タイプ: String

例:

```
"description": "This is a clock widget."
```

widget

ウィジェットの設定。*type*が *widget* に設定されている場合に使用されます。

タイプ: Object

サポートするオブジェクトキー*:*

- **name** (string) - ウィジェットリストおよびデフォルトのヘッダーとして使用されます。空の場合、モジュールの”name” パラメーターが使用されます。
- **size** (object) - デフォルトのウィジェットの寸法。サポートするキー:
  - *width* (integer) - デフォルトのウィジェットの幅。
  - *height* (integer) - デフォルトのウィジェットの高さ。
- **form_class** (string) - ウィジェットフィールドフォームを持つクラス。*includes* ディレクトリに配置する必要があります。クラスがデフォルトで期待する *WidgetForm.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **js_class** (string) - デフォルトの CWidget クラスを拡張するウィジェットビューモードの JavaScript クラスの名前。クラスにはダッシュボードがロードされます。クラスのソースファイルは、assets/js ディレクトリに配置する必要があります。参照: assets。
- **use_time_selector** (boolean) - ウィジェットにダッシュボード時間セレクターが必要かどうかを決定します。サポートする値: *true*、*false* (デフォルト)。
- **refresh_rate** (integer) - ウィジェットのリフレッシュレート (秒単位) (デフォルト: 60)。

例:

```
"widget": {
    "name": "",
    "size": {
        "width": 12,
        "height": 5
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

モジュールの説明へのリンク。ウィジェットの場合、ウィジェットの追加またはウィジェットの編集ウィンドウでヘルプアイコンをクリックすると、このリンクが開きます。**url** が指定されていない場合、ヘルプアイコン ? をクリックすると、一般的なダッシュボードウィジェットページが開きます。

タイプ: String

例:

```
"url": "http://example.com"
```

アクション

アクションは、モジュールの' ビジネスロジック' を担当します。アクションは通常、コントローラーとアクションビューで構成されます。

モジュールは次の操作を実行できます。

- Zabbix フロントエンドで既に定義されているアクションを呼び出す。
- デフォルトのアクションをカスタムアクションで上書きする。
- まったく新しいアクションを定義する。

デフォルトのアクション動作をカスタム動作で上書きするには、モジュール設定で同じ名前のアクションを定義します。アクションが呼び出されると、デフォルトの Zabbix アクションの代わりにモジュールアクションが実行されます。

アクションファイルは *actions* フォルダーに保存する必要があります。アクションはmanifest.jsonで指定する必要があります。

コントローラー

アクションコントローラーのワークフロー:

1) HTTP リクエストで渡されたすべてのパラメーターが有効であることを確認します:

- コントローラーの *checkInput()* メソッドを呼び出します
- CNewValidator.php で定義されたバリデーションルールを使用します
- *validateInput()* メソッドを呼び出します

2) ユーザー権限を確認します。

3) 渡されたパラメーターに従ってデータを準備します: *checkInput()* が true を返す場合、Zabbix はコントローラーの *doAction()* メソッドを呼び出します。

4) ビュー用の **$data** 配列を準備します。*CControllerResponseData* と *setResponse()* メソッドを使用して、レスポンスを **$data** 配列に格納します。

例:

```
/**
 * 入力パラメーターを検証します。
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * ユーザー権限を確認します。
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * アクションを実行し、レスポンスオブジェクトを生成します。
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
```

```
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
利用可能なコントローラークラスの完全なリストは、Zabbix のソースコードで確認できます。

ビュー

ビューファイルは、コントローラーからデータを受け取り、その HTML の外観を準備します。

Note:
モジュールがウィジェットでない限り、フロントエンドモジュールのビューの定義はオプションです。
ダッシュボードウィジェットには少なくとも 2 つのビューが必要です。1 つは編集モード用、もう 1 つは表示モード用です (*views* ディレクトリに保存する必要があります)。

ビューでは、定義済みの Zabbix HTML クラス (*/zabbix/ui/include/classes/html* から) を使用したり、新しい HTML クラスや CSS クラスを追加したりできます。新しいクラスは、モジュールの *assets* フォルダに保存する必要があります。

例:

```
...
(new CColHeader(_('Name')))
```

これにより、新しい列名 *Name* が追加され、他の Zabbix ページと同様にテーブルの上部の行のスタイルが設定されます。

アクションビュー

アクションビューを定義するためのリファレンスファイルです。

```
<?php declare(strict_types = 1);

/**
* @var CView $this
*/

$this->includeJsFile('example.something.view.js.php');

(new CWidget())
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
        'contacts' => $data['contacts']
    ])
    ->show();
```

アセット

assets フォルダーには、他のディレクトリに属さないファイルやサブフォルダーを含めることができます。次の目的で使用できます:

- JavaScript スタイル (*assets/js*内にある必要があります)
- CSS スタイル (*assets/css*内にある必要があります)
- 画像
- フォント
- その他、含める必要のあるもの

assets/js

assets/js ディレクトリは予約されており、JavaScript ファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css は予約されており、CSS スタイルファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "css": ["mywidget.css"]
}
```

CSS スタイル

CSS ファイルには、特定のフロントエンドテーマの異なるスタイルを定義するためのカスタム属性 `theme` が含まれる場合があります。

利用可能なテーマとその属性値:

- ブルー - [theme='blue-theme']
- ダーク - [theme='dark-theme']
- ハイコントラストライト - [theme='hc-light']
- ハイコントラストダーク - [theme='hc-dark']

例:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

新しいモジュールの登録

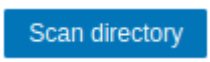
このセクションでは、Zabbix フロントエンドに新しいモジュールを追加する方法について説明します。

前提条件　続行する前に、次のことを確認してください:

- モジュールは、Zabbix フロントエンドインストールの *modules* ディレクトリ内にあります (例: *zabbix/ui/modules*)。
- モジュールには、少なくともmanifest.jsonファイルの基本バージョンがあります。
- Zabbix の管理メニューセクションにアクセスできます (ユーザーの役割の super admin が必要です)。

> **Note:**
> フロントエンドは、互換性のないモジュールをインストールせず、認識もしません。

モジュールの追加　管理 → 一般 → モジュールページを開き、ディレクトリのスキャンを押します。



リストでモジュールを見つけてアクティブ化します。

モジュールをアクティブ化するには、無効ハイパーリンクを押します。モジュールの状態が有効に変わります。

モジュール名を押すと、作成者、バージョン、短い説明 (マニフェストで定義されている場合) など、モジュールに関する追加情報が表示されます。

ウィジェットのプレビュー　ウィジェットモジュールは、追加されるとすぐにダッシュボードウィジェットリストに表示されます。

ダッシュボードを開いて編集モードに切り替え、通常どおりダッシュボードにウィジェットを追加できます。

ウィジェットに変更を加えた場合は、ダッシュボードを更新して、最新の更新でウィジェットがどのように見えるかを確認します。

ウィジェット

ウィジェットは、ダッシュボードに使用される Zabbix フロントエンドモジュールです。特に明記されていない限り、すべてのモジュールガイドラインはウィジェットにも適用されます。

ただし、ウィジェットはモジュールとは大きく異なります。ウィジェットを作成するには、次の操作を行います。

- manifest.json ファイルでタイプ”widget” を指定します (”type”: ”widget”)。
- 少なくとも 2 つのビューを含めます。1 つはウィジェットプレゼンテーションモード用、もう 1 つはウィジェット構成モード用です (example.widget.view.php と example.widget.edit.php)。
- ウィジェットプレゼンテーション (WidgetView.php) 用のコントローラー。
- デフォルトのウィジェットクラスを使用して拡張します。

設定

このページでは、カスタム設定フィールドを使用してウィジェット設定ビューを作成するために使用できるクラスについて説明します。ウィジェット設定ビューは、ユーザーがプレゼンテーションのウィジェットパラメーターを設定できるウィジェットの一部です。

Widget

プライマリウィジェットクラス、すべてのダッシュボードウィジェットの基本クラス *CWidget* を拡張します。デフォルトのウィジェットの動作をオーバーライドするために必要です。

Widget クラスは、ウィジェットのルートディレクトリ (*zabbix/ui/modules/my_custom_widget* など) に配置する必要があります。

Widget.php example

```
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const MY_CONSTANT = 0;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

WidgetForm

WidgetForm クラスは、デフォルトクラス *CWidgetForm* を拡張し、データベース内でウィジェット構成の保存構造を定義し、入力検証を処理するために必要な*CWidgetField*フィールドのセットを含みます。

WidgetForm クラスは *includes* ディレクトリに配置する必要があります。クラスに別の名前が付いている場合は、その名前を *manifest.json* ファイルの*widget/form_class*パラメーターで指定する必要があります。

includes/WidgetForm.php example

```
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;
```

```
use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};


class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

CWidgetFormView クラスは、*WidgetForm* クラスで定義されたフィールドのプレゼンテーションロジックを指定し、設定ビューで表示されるときの外観と動作を決定するために必要です。

CWidgetFormView クラスは次のメソッドをサポートします。

- *addField()* - *CWidgetFieldView* クラスのインスタンスをパラメーターとして受け取ります。各*CWidgetField*クラスには、ウィジェット設定ビューで使用するためのそれぞれの *CWidgetFieldView* クラスがあります。
- *addFieldset()* - フィールドを結合して折りたたみ可能なコンテナにする *CWidgetFieldsGroupView* クラスのインスタンスを受け取ります。
- *addFieldsGroup()* - フィールドを視覚的に (境界線を使用して) グループに結合する *CWidgetFormFieldsetCollapsibleView* のインスタンスを受け取ります。
- *includeJsFile()* - JavaScript ファイルをウィジェット設定ビューに追加できます。
- *addJavaScript()* - ウィジェット設定ビューがロードされるとすぐに実行されるインライン JavaScript を追加できます。

CWidgetFormView クラスは *views* ディレクトリに配置する必要があります。

views/widget.edit.php example

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;
```

```
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

JavaScript クラスを使用して、ウィジェット設定ビューに動的な動作と対話性を追加できます。たとえば、*CWidgetFormView*クラスで定義されたカラーピッカーを初期化できます。

JavaScript クラスはフォームとともにロードする必要があるため、メソッド *includeJsFile()* および *addJavaScript()* を使用して*CWidgetFormView*クラスで参照する必要があります。

以下の例では、シングルトンクラスインスタンスがすぐに作成され、*window.my_custom_widget_form* という名前で保存されます。したがって、フォームを 2 回目に開くとインスタンスが再作成されます。

JavaScript クラスは *views* ディレクトリに配置する必要があります。

views/widget.edit.js.php example

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

CWidgetField クラスは、すべてのフォームフィールドクラス (*CWidgetFieldCheckBox*、*CWidgetFieldTextArea*、*CWidgetFieldRadioButtonList* など) が継承される基本クラスです。*CWidgetField* を拡張するクラスは、ウィジェット設定値の受信、保存、検証を担当します。

次の *CWidgetField* クラスが利用可能です。

CWidgetField クラス	データベースフィールドタイプ	説明
CWidgetFieldCheckBox	int32	Single checkbox.
CWidgetFieldCheckBoxList	array of int32	Multiple checkboxes under a single configuration field.
CWidgetFieldColor	string	Color selection field.
CWidgetFieldDatePicker	string	Date selection field.
CWidgetFieldHostPatternSelect	string	Multiselect field that allows to select one or multiple hosts. Supports defining host name patterns (all matching hosts will be selected).
CWidgetFieldIntegerBox	int32	Field to enter an integer. Can be used to configure minimum and maximum values.
CWidgetFieldLatLng	string	Text box that allows to enter comma-separated latitude, longitude, and map zoom level.
CWidgetFieldMultiSelectAction	ID	Multiselect field for selecting actions (from the list of actions defined in the *Alerts → Actions*).
CWidgetFieldMultiSelectGraph	ID	Multiselect field for selecting custom graphs.
CWidgetFieldMultiSelectGraphPrototype	ID	Multiselect field for selecting custom graph prototypes.
CWidgetFieldMultiSelectGroup	ID	Multiselect field for selecting host groups.
CWidgetFieldMultiSelectHost	ID	Multiselect field for selecting hosts.
CWidgetFieldMultiSelectItem	ID	Multiselect field for selecting items.
CWidgetFieldMultiSelectItemPattern	*Pattern*	Multiselect field for selecting item patterns.
CWidgetFieldMultiSelectItemPrototype	ID	Multiselect field for selecting item prototypes.
CWidgetFieldMultiSelectMap	ID	Multiselect field for selecting maps.
CWidgetFieldMultiSelectMediaType	ID	Multiselect field for selecting media types.
CWidgetFieldMultiSelectOverrideHost	ID	Multiselect field for selecting a data source (dashboard or other widget) containing a host for which the widget can display data.
CWidgetFieldMultiSelectService	ID	Multiselect field for selecting services.
CWidgetFieldMultiSelectSla	ID	Multiselect field for selecting SLAs.
CWidgetFieldMultiSelectUser	ID	Multiselect field for selecting users.
CWidgetFieldNumericBox	string	Field to enter a float number.
CWidgetFieldRadioButtonList	int32	Radio box group that consists of one or more radio boxes.
CWidgetFieldRangeControl	int32	Slider to select an integer type value.
CWidgetFieldReference	string	Creates a unique identifier for this widget on dashboard. It is used to reference this widget from other widgets.
CWidgetFieldSelect	int32	Dropdown select box.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* preset with trigger severities.
CWidgetFieldTags	array of (string, int32, string)	Allows to configure one or more tag filter rows.
CWidgetFieldTextArea	string	Text area for entering multi-line text.
CWidgetFieldTextBox	string	Text box for entering single-line text.
CWidgetFieldTimePeriod	array of string	Time period selecting field.
CWidgetFieldTimeZone	string	Dropdown with timezones.
CWidgetFieldThresholds	array of (string, string)	Allows configuring color and number pairs.
CWidgetFieldUrl	string	Text box that allows to enter URLs.

次の *CWidgetField* クラスが特定のウィジェット用に作成されています。これらのクラスには非常に特殊な使用例がありますが、必要に応じて再利用することもできます。

CWidgetField クラス	データベースフィールドタイプ	説明
CWidgetFieldColumnsList	array of (multiple mixed)	For *Top hosts* widget. Create a table with custom columns of allowed types.
CWidgetFieldNavTree	string	For *Map navigation tree* widget. Replaces widget view in edit mode with the map selection tree.

プレゼンテーション

このページでは、ウィジェットプレゼンテーションビューの作成に使用できるコンポーネントについて説明します。ウィジェットプレゼンテーションビューは、ウィジェットの設定に従ってデータを受信し、コンテナ内のダッシュボードに表示するウィジェットの一部です。

プレゼンテーションビューは、次の 3 つの部分で設定されます。

- ウィジェットアクション
- ウィジェットビュー
- JavaScript

Widget action

ウィジェットアクションクラス (*WidgetView*) には、プレゼンテーションビューモードでウィジェットを操作するためのメソッドが含まれています。ウィジェットアクションの大部分は、デフォルトのコントローラークラス *CControllerDashboardWidgetView* を使用または拡張します。

ウィジェットのアクションクラスは *actions* ディレクトリに配置され、*manifest.json* ファイル内の*actions*パラメーター (*actions/widget.{id}.view/class*) で指定される必要があります。

actions/WidgetView.php example (implemented in the Zabbix-native *System information* widget)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Widget view

ウィジェットビュークラス (*CWidgetView*) は、ウィジェットプレゼンテーションビューの構築を担当します。

ウィジェットビュークラスは *views* ディレクトリに配置する必要があります。ウィジェットビュークラスを含むファイルの名前がデフォルト (*widget.view.php*) とは異なる場合は、*manifest.json* ファイルの*actions*パラメーター (*actions/widget.{id}.view/view*) で指定する必要があります。

views/widget.view.php example

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

JavaScript クラスは、ウィジェットデータの更新、ウィジェットのサイズの変更、ウィジェット要素の表示など、ウィジェットの動作を決定する責任があります。

すべての JavaScript 操作は、すべてのダッシュボードウィジェットのベース JavaScript クラスである *CWidget* を使用または拡張をします。*CWidget* クラスには、ウィジェット動作のデフォルトの実装を備えた一連のメソッドが含まれています。ウィジェットの複雑さに応じて、これらのメソッドはそのまま使用または拡張できます。

CWidget クラスには、次のメソッドが含まれています。

- ウィジェットのライフサイクルを定義するメソッド: *_init()*, *_registerEvents()*, *_doActivate()*, *_doDeactivate()*, *_doDestroy()*, *setEditMode()*.

- ウィジェットのデータ更新と表示を処理するメソッド: *_promiseUpdate()*, *_getUpdateRequestData()*, *_processUpdateResponse(response)*, *_processUpdateErrorResponse(error)*, *_setContents()*.
- ウィジェットの外観を変更するメソッド: *resize()*, *_hasPadding()*.

JavaScript クラスは *assets/js* ディレクトリに配置し、*manifest.json* ファイルの*assets* (*assets/js*) パラメーターで指定する必要があります。

ライフサイクルメソッド

ウィジェットのライフサイクルメソッドは、ダッシュボードによって呼び出され、ダッシュボード内に存在する間、ウィジェットのライフサイクルのさまざまな段階で呼び出されます。

onInitialize() メソッドは、HTML またはデータ操作を実行せずに、ウィジェットの初期状態や値を定義します。このメソッドは通常、ウィジェットをダッシュボードページに追加するか、ダッシュボードページをロードするか、ウィジェットが作成されたときに呼び出されます (ウィジェットオブジェクトがインスタンス化されています)。

例:

```
onInitialize() {
    this._time_offset = 0;
    this._interval_id = null;
    this._clock_type = CWidgetClock.TYPE_ANALOG;
    this._time_zone = null;
    this._show_seconds = true;
    this._time_format = 0;
    this._tzone_format = 0;
    this._show = [];
    this._has_contents = false;
    this._is_enabled = true;
}
```

onStart() メソッドは、データ操作を実行せずに、ウィジェットの HTML 構造を定義します。このメソッドは、ダッシュボードページの最初のアクティブ化、つまりダッシュボードとそのウィジェットがユーザーに完全に表示される前に呼び出されます。

例:

```
onStart() {
    this._events.resize = () => {
        const padding = 25;
        const header_height = this._view_mode === ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER
            ? 0
            : this._header.offsetHeight;

        this._target.style.setProperty(
            '--content-height',
            `${this._cell_height * this._pos.height - padding * 2 - header_height}px`
        );
    }
}
```

onActivate() メソッドは、カスタムイベントリスナー (ユーザーアクションに応答するためのもの) を有効にし、ウィジェットの更新サイクル (コンテンツを最新の状態に保つためのもの) を開始することにより、ウィジェットをアクティブにします。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースに完全に表示されるときに呼び出されます。

onActivate() メソッドが呼び出される前は、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) に移行します。アクティブ状態では、ウィジェットはレスポンシブで、イベントに耳を傾け、コンテンツを定期的に更新し、他のウィジェットと対話できます。

例:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

onDeactivate() メソッドは、カスタムイベントリスナーを非アクティブ化し、ウィジェットの更新サイクルを停止することにより、ウィジェットのアクティビティとインタラクティブ性を停止します。このメソッドは、ダッシュボードページが無効になっている場合、つまり切り替えまたは削除されたとき、またはダッシュボードページからウィジェットが削除されたときに呼び出されます。

onDeactivate() メソッドが呼び出される前は、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE に移行します。

例:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

onDestroy() メソッドは、ウィジェットがダッシュボードから削除される前にクリーンアップタスクを実行します。クリーンアップタスクとは、ウィジェットの初期化中に確立されたデータベース接続を閉じたり、リソースリークを回避するために一時的なデータをクリーンアップしてシステムメモリを解放したり、不要なイベントの取り扱いやメモリリークを防ぐためにサイズ変更イベントまたはボタンクリックに関連するイベントリスナーの登録を解除したりなどします。このメソッドは、ウィジェットまたはそれを含むダッシュボードページが削除されたときに呼び出されます。

Note that before the *onDestroy()* method is invoked, a widget in an active state (WIDGET_STATE_ACTIVE) is always deactivated with the invocation of the *onDeactivate()* method. *onDestroy()* メソッドが呼び出される前にアクティブな状態 (WIDGET_STATE_ACTIVE) のウィジェットは、*onDeactivate()* メソッドの呼び出しで常に非アクティブ化されることに注意してください。

例:

```
onDestroy() {
    if (this._filter_widget) {
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

onEdit() メソッドは、ダッシュボードが編集モードに変換されるときのウィジェットの外観と動作を定義します。このメソッドは、通常ユーザーがウィジェットの編集ボタンまたはダッシュボードのダッシュボードの変更ボタンを押し、ダッシュボードが編集モードに移行するときに呼び出されます。

例:

```
onEdit() {
    this._deactivateGraph();
}
```

更新プロセスメソッド

ウィジェット更新プロセスメソッドは、Zabbix サーバーまたはその他のデータソースから更新されたデータを取得し、ウィジェットに表示する役割を担います。

promiseUpdate() メソッドは、通常、Web リクエストまたは API 呼び出しを使用してデータを取得することにより、データ更新プロセスを開始します。このメソッドは、ダッシュボードページが表示されたときに呼び出され、その後、ダッシュボードページが別のダッシュボードページに切り替わるまで定期的に呼び出されます。

以下は、ほとんどの Zabbix ネイティブウィジェットが使用する *promiseUpdate()* メソッドのデフォルトの実装の例です。デフォルトの実装では、*promiseUpdate()* メソッドは、サーバーからデータを取得するための一般的なパターンに従います。適切な URL とリクエストパラメーターを使用して新しい Curl オブジェクトを作成し、*getUpdateRequestData()* メソッドで構築されたデータオブジェクトを使用して *fetch()* メソッドで POST リクエストを送信し、*processUpdateResponse(response)* または *processUpdateErrorResponse(error)* に応じてレスポンス (またはエラーレスポンス) を処理します。この実装は、通常 JSON 形式でデータを取得し、一貫した方法で処理するため、ほとんどのウィジェットに適しています。

```
promiseUpdate() {
    const curl = new Curl('zabbix.php');

    curl.setArgument('action', `widget.${this._type}.view`);

    return fetch(curl.getUrl(), {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify(this.getUpdateRequestData()),
        signal: this._update_abort_controller.signal
    })
        .then((response) => response.json())
        .then((response) => {
```

```
            if ('error' in response) {
                this.processUpdateErrorResponse(response.error);

                return;
            }

            this.processUpdateResponse(response);
        });
    }
```

_getUpdateRequestData() メソッドは、ウィジェットの状態と設定からさまざまなプロパティとその対応する値 (ウィジェット識別子、フィルター設定、時間範囲など) を収集し、ウィジェットを更新するためのサーバーリクエストデータを準備し、更新リクエストでサーバーに送信される必要な情報を表すデータオブジェクトを構築します。このメソッドは、ウィジェット更新プロセス中にデフォルトの [*_promiseUpdate()*] メソッドの一部としてのみ呼び出されます。

デフォルトの実装:

```
_getUpdateRequestData() {
    return {
        templateid: this._dashboard.templateid ?? undefined,
        dashboardid: this._dashboard.dashboardid ?? undefined,
        widgetid: this._widgetid ?? undefined,
        name: this._name !== '' ? this._name : undefined,
        fields: Object.keys(this._fields).length > 0 ? this._fields : undefined,
        view_mode: this._view_mode,
        edit_mode: this._is_edit_mode ? 1 : 0,
        dynamic_hostid: this._dashboard.templateid !== null || this.supportsDynamicHosts()
            ? (this._dynamic_hostid ?? undefined)
            : undefined,
        ...this._content_size
    };
}
```

processUpdateResponse(response) メソッドは、更新リクエスト後にサーバーから受信したレスポンスを処理し、更新プロセスがエラーなしで成功している場合、ウィジェットデータをクリアし、*setContents()* メソッドで新しいコンテンツを表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェットの更新プロセス中のみに呼び出されます。

デフォルトの実装:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

processUpdateErrorResponse(error) メソッドは、レスポンスがエラーである場合、更新リクエスト後にサーバーから受信したレスポンスを処理し、エラーメッセージ/-s を表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェット更新プロセス中のみ呼び出されます。

デフォルトの実装:

```
_processUpdateErrorResponse(error) {
    this._setErrorContents({error});
}

_setErrorContents({error}) {
    const message_box = makeMessageBox('bad', error.messages, error.title)[0];

    this._content_body.innerHTML = '';
    this._content_body.appendChild(message_box);

    this._removeInfoButtons();
```

```
}
```

setContents(response) メソッドは、ウィジェットの更新プロセスがエラーなく成功した場合にウィジェットの内容を表示します。これには、DOM 要素の操作、UI コンポーネントの更新、スタイルの適用、フォーマットなどが含まれます。このメソッドは、デフォルトの *processUpdateResponse(response)* メソッドの一部として、つまり更新リクエスト後にサーバーから受信したレスポンスを処理するプロセス中のみ呼び出されます。

デフォルトの実装:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

プレゼンテーション変更メソッド

ウィジェットのプレゼンテーション変更メソッドは、ウィジェットの外観を変更します。

onResize() メソッドは、新しいウィジェットサイズに対応するためにウィジェットの視覚要素の調整を担当します。これには、要素の再配置、要素の寸法調整、テキストの切り捨て、サイズ変更中の応答性を向上させるための遅延読み込みの実装などが含まれます。このメソッドは、ウィジェットがサイズ変更されたときに呼び出されます。たとえば、ユーザーがウィジェットを手動でサイズ変更するときやブラウザウィンドウのサイズが変更されたときです。

例:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

hasPadding() メソッドは、ヘッダーを表示に構成されている場合、ウィジェットの下部に 8px の垂直パディングを適用します。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースの表示されるページになったときに呼び出されます。

デフォルトの実装:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

一部のウィジェットの場合、使用可能なすべてのウィジェットスペースを使用して、たとえばカスタムの背景色を構成する必要があります。以下は、zabbix ネイティブのアイテムの値ウィジェットで使用される *hasPadding()* メソッドの実装の例です。

```
hasPadding() {
    return false;
}
```

チュートリアル

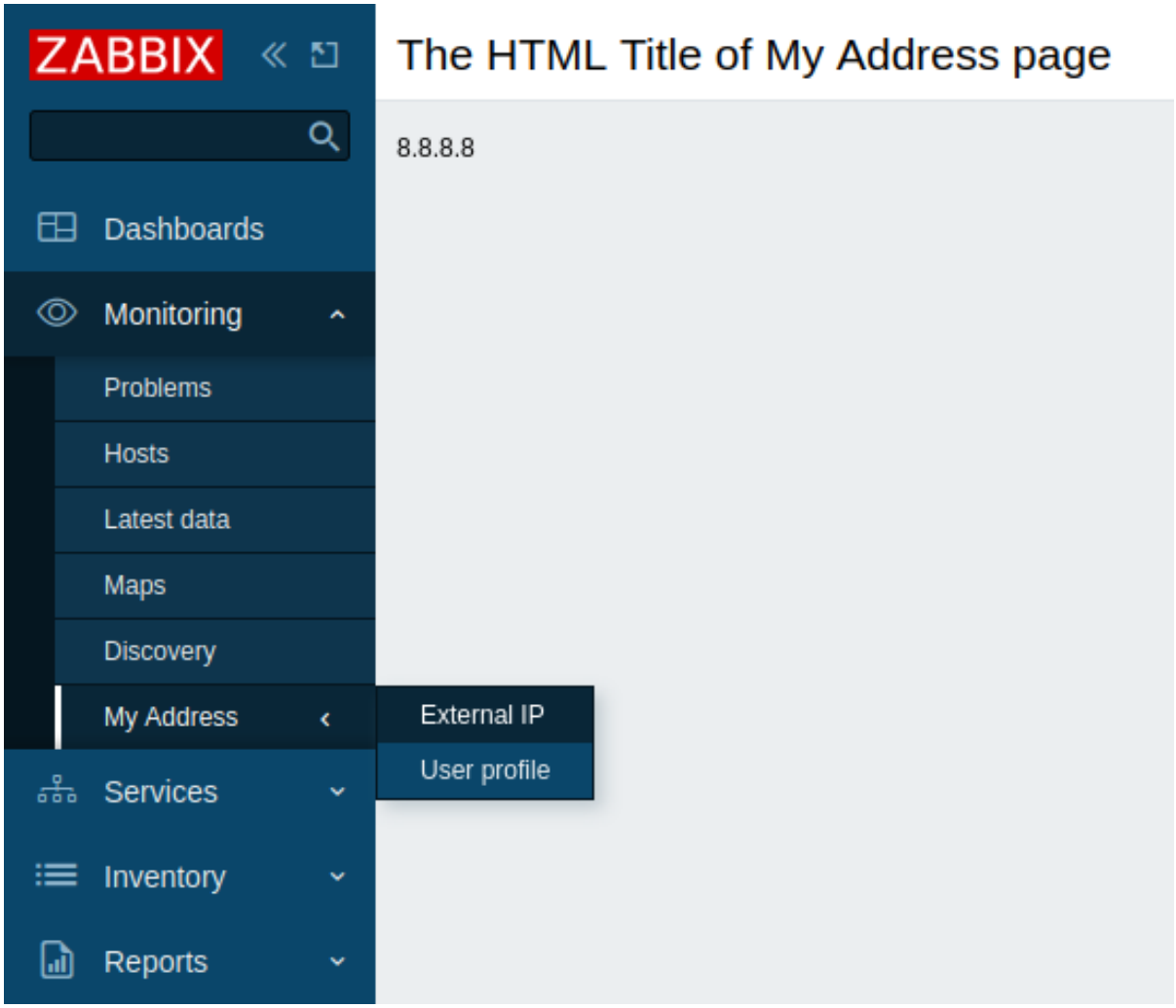
このセクションには、Zabbix でカスタムモジュールとウィジェットを構築する方法を説明する、実践的なステップバイステップのチュートリアルが含まれています。

モジュールの作成 **(**チュートリアル**)**

これは、シンプルな Zabbix フロントエンドモジュールを作成する方法を示すステップバイステップのチュートリアルです。このモジュールのすべてのファイルを ZIP アーカイブとしてダウンロードできます: MyAddress.zip。

作成するもの

このチュートリアルでは、まず新しい *My Address* メニューセクションを追加するフロントエンドモジュールを構築し、次にそれをより高度なフロントエンドモジュールに変換します。このモジュールは、*https://api.seeip.org* に HTTP リクエストを送信し、レスポンス (コンピューターの IP アドレス) を新しく作成された *My Address* メニューセクションの新しいページに表示します。完成したモジュールは次のようになります:



パート I - 新しいメニューセクション

Zabbix フロントエンドに空のモジュールを追加

1. Zabbix フロントエンドインストールの *modules* ディレクトリに *MyAddress* ディレクトリを作成します (例: *zabbix/ui/modules*)。
2. 基本モジュールのメタデータを含む *manifest.json* ファイルを作成します (サポートされているパラメーターの説明を参照)。

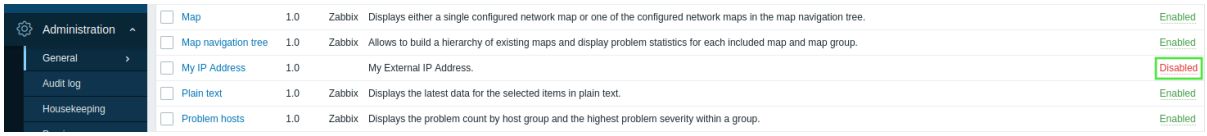
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Zabbix フロントエンドで、管理 → 一般 → モジュールセクションに移動し、ディレクトリのスキャンボタンをクリックします。



4. リスト内で新しいモジュール *My IP Address* を見つけ、" 無効" ハイパーリンクをクリックしてモジュールのステータスを" 無効" から" 有効" に変更します (モジュールがリストされていない場合、トラブルシューティングセクションを確認してください)。



これで、モジュールがフロントエンドに登録されました。ただし、モジュールの機能を定義する必要があるため、まだ表示されていません。モジュールディレクトリにコンテンツを追加し、ページを更新すると、Zabbix フロントエンドの変更がすぐに表示されます。

メニューセクションの作成

1. *MyAddress* ディレクトリに *Module.php* ファイルを作成します。

このファイルは、デフォルトの *CModule* クラスを拡張する新しい *Module* クラスを実装します。*Module* クラスは、新しい *My Address* メニューセクションをメインメニューに挿入します。

setAction() メソッドは、メニューセクションをクリックしたときに実行されるアクションを指定します。まず、事前定義されたアクション *userprofile.edit* を使用すると、ユーザープロファイルページが開きます。このチュートリアルのパート IIIでは、カスタムアクションの作成方法を学習します。

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

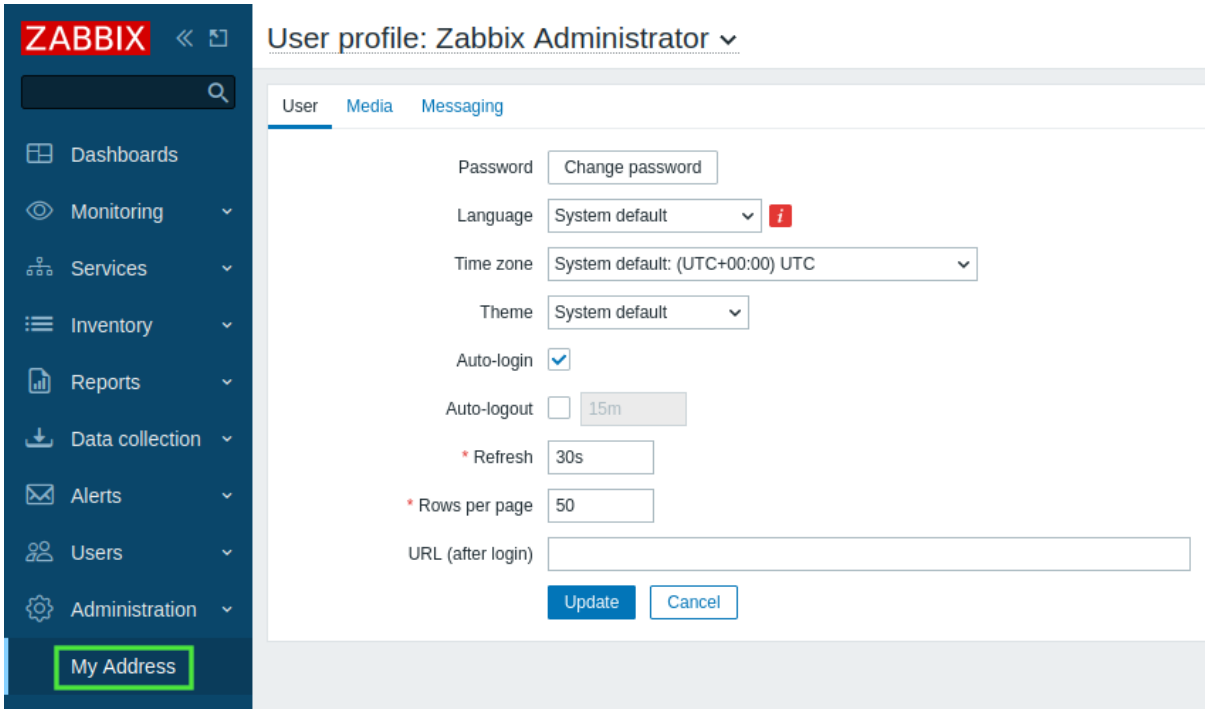
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
'userprofile.edit'を他のアクション、たとえば'charts.view'(カスタムグラフを開く)、'problems.view'(データ監視 → 障害を開く)、または 'report.status'(システム情報レポートを開く) に置き換えることができます。

3. Zabbix フロントエンドを更新します。Zabbix メインメニューの下部に、新しい *My Address* セクションが追加されました。*My Address* をクリックして、*User profile* ページを開きます。



パート II - メニューセクションの位置変更

このパートでは、*My Address* メニューセクションを監視データセクションに移動しし、そこにネストされたメニューを追加します。その結果、ユーザーは監視データ → マイアドレスメニューセクションから 2 つのサブメニューページにアクセスできるようになり

ます。

1. *Module.php* ファイルを開いて編集します。

ui/modules/MyAddress/Module.php

```
<?php

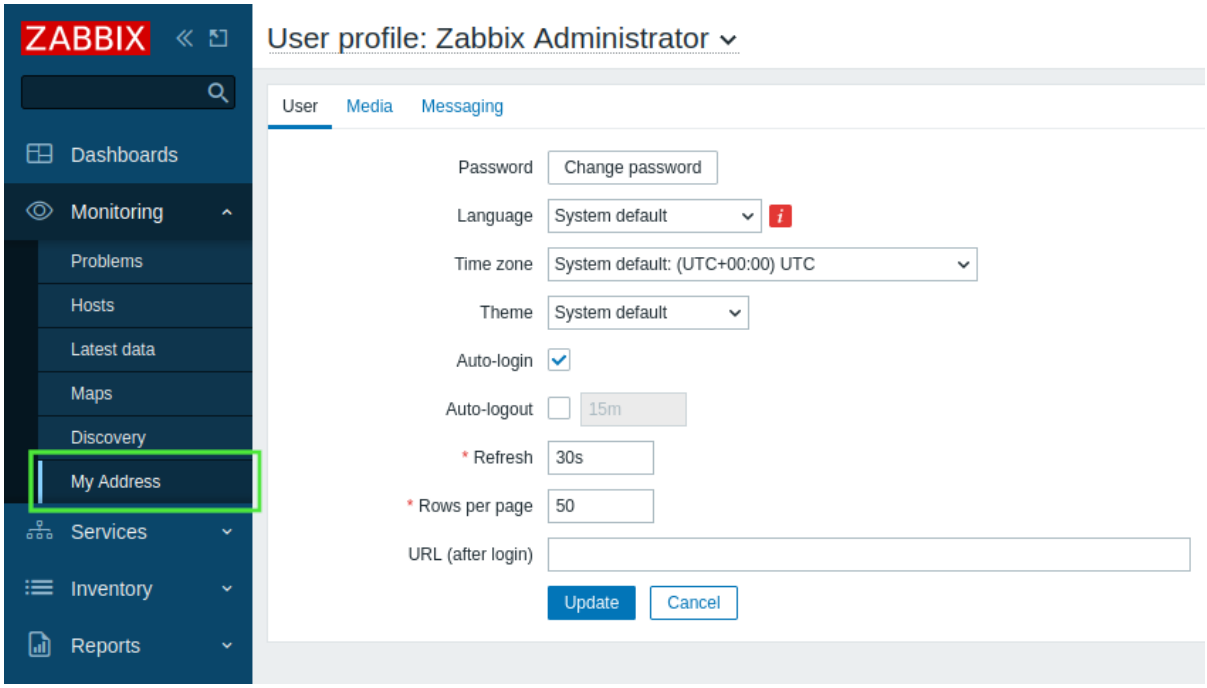
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Zabbix フロントエンドを更新します。監視データメニューセクションを展開し、*My address* セクションがディスカバリセクションの下にあることを確認します。



3. ネストされたページを *My Address* メニューセクションに追加するには、*Module.php* ファイルを再度開いて編集します。

このステップでは、2 つのサブセクションが作成されます。

- 次の手順で定義する新しい'my.address' アクションを実行する *External IP*。
- *User profile* ページを開くための事前定義された'userprofile.edit' アクションを実行する *User profile*。

ネストされたメニューの場合は、前の手順で使用したクラスに加えて *CMenu* クラスを使用する必要があることに注意してください。

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

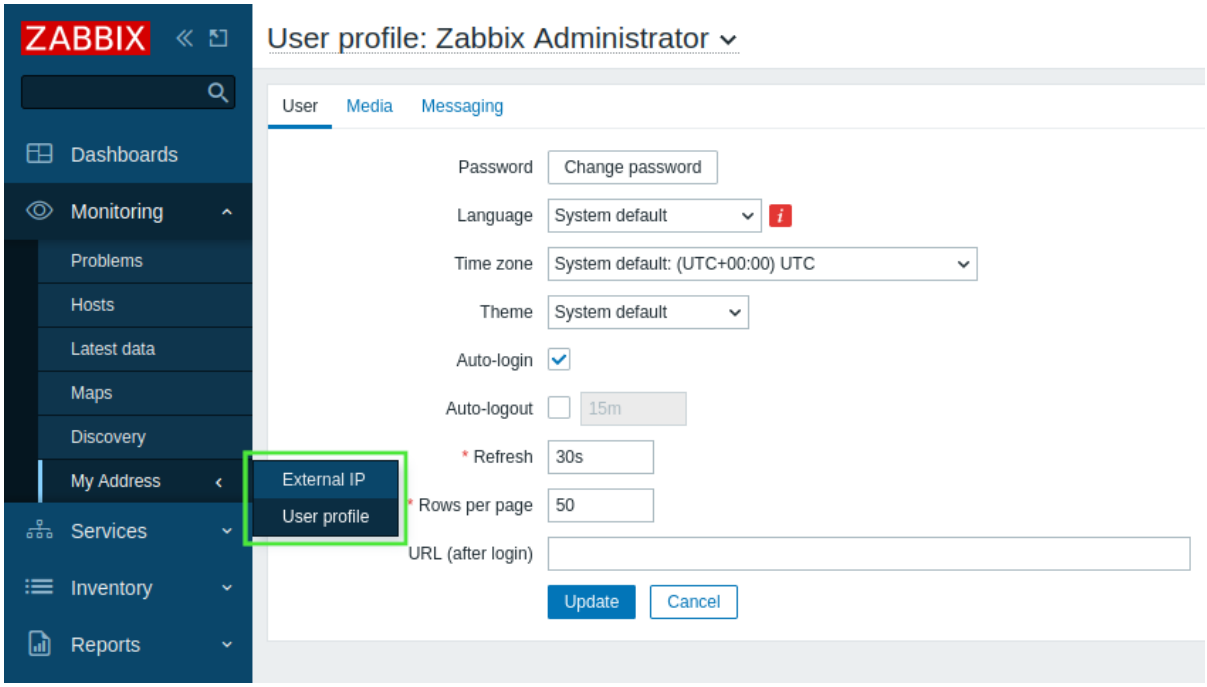
use Zabbix\Core\CModule,
    APP,
```

```
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Zabbix フロントエンドを更新します。*My address* メニューセクションに、*External IP* と *User profile* の 2 ページからなる第 3 レベルのメニューが含まれていることを確認します。



パート III - モジュールアクション

アクションは、*actions/MyAddress.php* と *views/my.address.php* の 2 つのファイルに実装されます。***actions/MyAddress.php*** ファイルはビジネスロジックの実装を担当し、***views/my.address.php*** ファイルはビューを担当します。

1. *MyAddress* ディレクトリに *actions* ディレクトリを作成します。
2. *actions* ディレクトリに *MyAddress.php* ファイルを作成します。

アクションロジックは *MyAddress* クラスで定義されます。このアクションクラスは、*init()*、*checkInput()*、*checkPermissions()*、*doAction()* の 4 つの関数を実装します。Zabbix フロントエンドは、アクションが要求されると *doAction()* 関数を呼び出します。この関数は、モジュールのビジネスロジックを担当します。

Attention:
データは連想配列として編成する必要があります。配列は多次元にすることができ、ビューで予期される任意のデータを含めることができます。

ui/modules/MyAddress/actions/MyAddress.php

```php
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```

3. *MyAddress* ディレクトリに新しいディレクトリ *views* を作成します。
4. *views* ディレクトリに *my.address.php* ファイルを作成し、モジュールビューを定義します。

変数 `$data` は、特に定義しなくてもビューで使用できることに注意してください。フレームワークは自動的に連想配列をビューに渡します。

ui/modules/MyAddress/views/my.address.php

```php
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

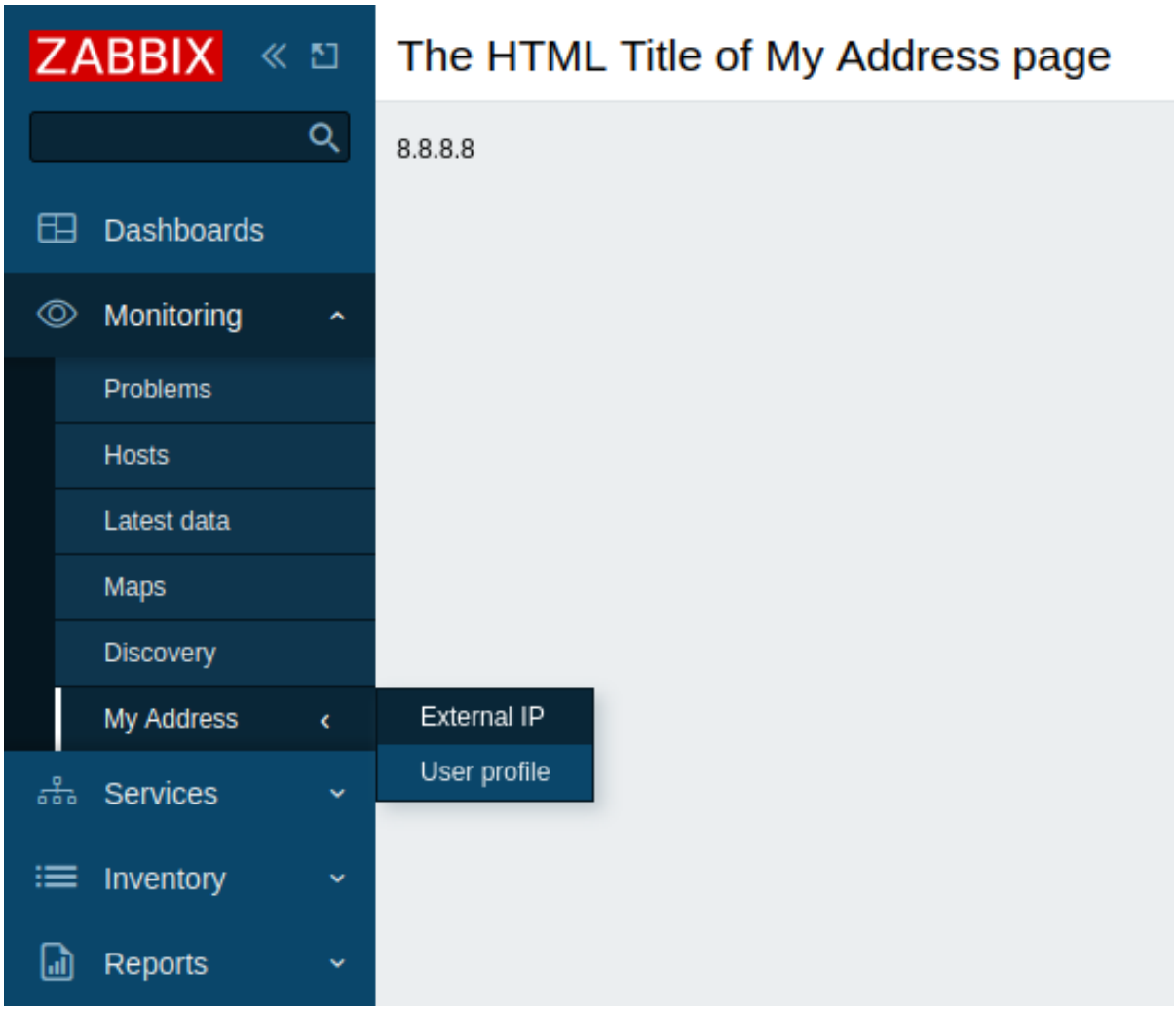
5. モジュールアクションは *manifest.json* ファイルに登録する必要があります。*manifest.json* を開き、以下を含む新しいオブジェクト `actions` を追加します。

- 小文字 (a~z) で書かれ、単語がドットで区切られたアクション名を持つアクションキー (例: `my.address`)
- `my.address` オブジェクトの `class` キーの値としてのアクションクラス名 (`MyAddress`)
- `my.address` オブジェクトの `view` キーの値としてのアクションビュー名 (`my.address`)

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

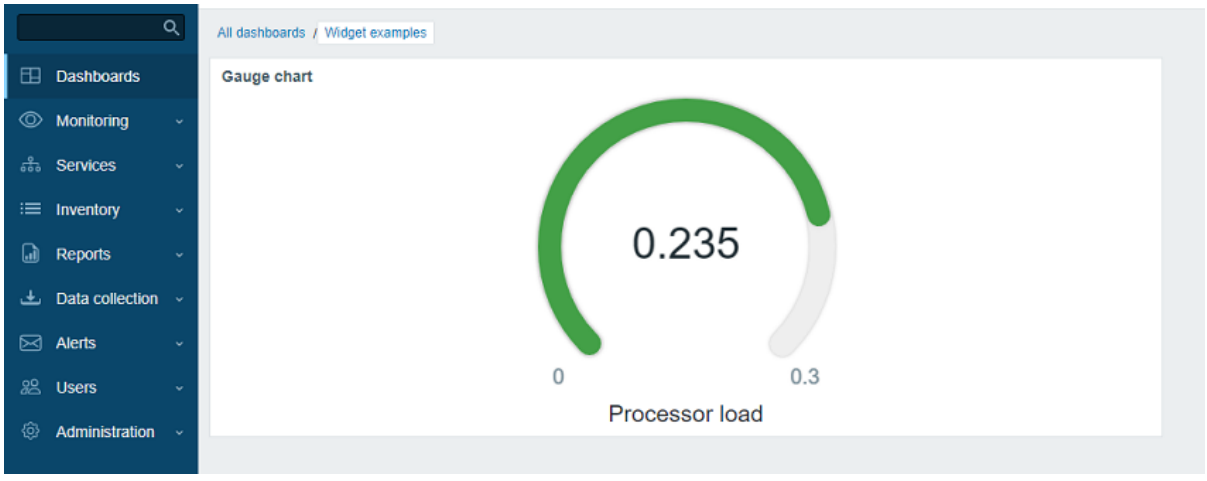
6. Zabbix フロントエンドを更新します。*My address* → *External IP* をクリックして、コンピュータの IP アドレスを確認します。



ウィジェットの作成 **(**チュートリアル**)**

これは、シンプルなダッシュボードウィジェットを作成する方法を示すステップバイステップのチュートリアルです。このウィジェットのすべてのファイルを ZIP アーカイブとしてダウンロードできます: lesson_gauge_chart.zip。

作成するもの　このチュートリアルでは、まず基本的な”Hello, world!” ウィジェットを作成し、次にそれをより高度なウィジェットに変換して、アイテムの値をゲージチャートとして表示します。完成したウィジェットは次のようになります:



パート **I - ”Hello, world!”**　このセクションでは、最小限必要なウィジェット要素を作成し、新しいウィジェットを Zabbix フロントエンドに追加する方法を学びます。

Zabbix フロントエンドに空のウィジェットを追加する

1. Zabbix フロントエンドインストールの *modules* ディレクトリ（例：*zabbix/ui/modules*）に *lesson_gauge_chart* ディレクトリを作成します。

> **Note:**
> すべてのカスタムウィジェットは外部モジュールとして扱われ、Zabbix フロントエンドインストールの *modules* ディレクトリ（例：*zabbix/ui/modules*）に追加する必要があります。*zabbix/ui/widgets* ディレクトリは Zabbix 組み込みウィジェット用に予約されており、Zabbix UI とともに更新されます。

2. 基本的なウィジェットメタデータを含む *manifest.json* ファイルを作成します（サポートされているパラメータの説明を参照）。

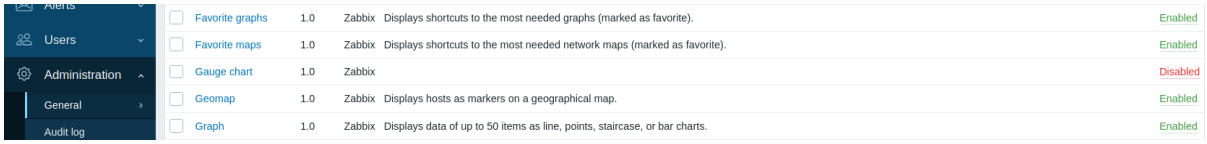
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

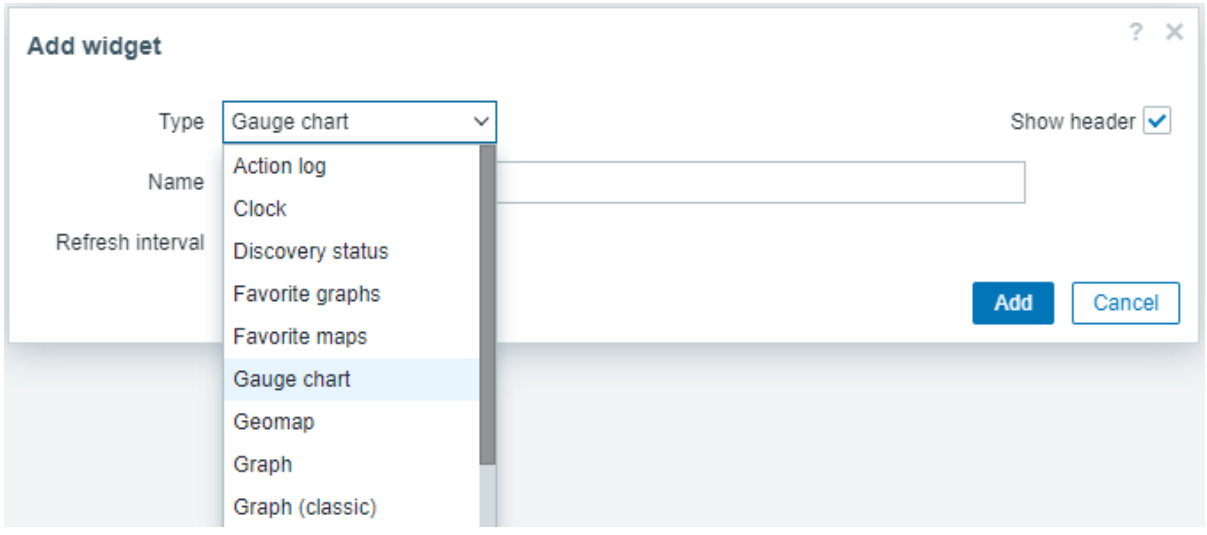
3. Zabbix フロントエンドで、管理 → 一般 → モジュールセクションに移動し、ディレクトリのスキャンボタンをクリックします。



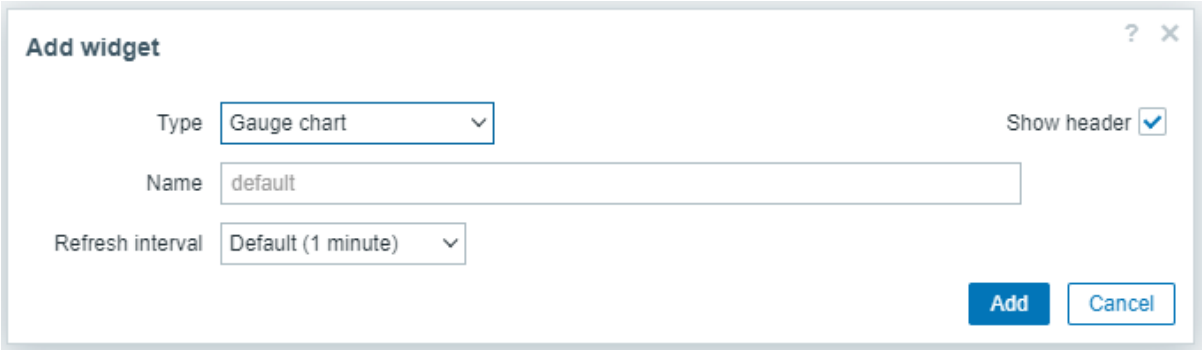
4. リストから新しいモジュール *Gauge chart* を見つけ、「無効」ハイパーリンクをクリックしてモジュールのステータスを「無効」から「有効」に変更します（モジュールがリストに表示されない場合は、トラブルシューティングセクションを参照してください）。



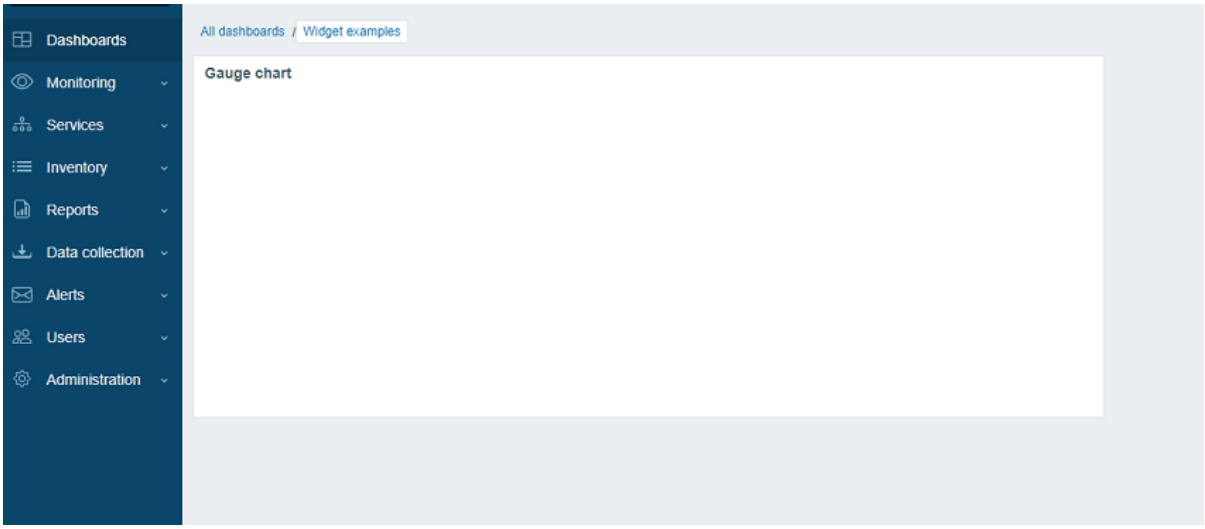
5. ダッシュボードを開き、編集モードに切り替えて新しいウィジェットを追加します。タイプフィールドで「Gauge chart」を選択します。



6. この時点では、*Gauge chart* ウィジェットの設定には共通ウィジェットフィールドである名前と更新間隔のみが含まれています。追加をクリックしてウィジェットをダッシュボードに追加します。



7. ダッシュボードに空のウィジェットが表示されます。右上隅の変更を保存をクリックしてダッシュボードを保存します。



ウィジェットビューの追加

Note:
ウィジェットの **view** ファイルは *views* ディレクトリ（このチュートリアルでは *ui/modules/lesson_gauge_chart/views/*）に配置する必要があります。ファイル名がデフォルトの *widget.view.php* の場合、*manifest.json* ファイルに登録する必要はありません。ファイル名が異なる場合は、manifest.jsonファイルの *actions/widget.lesson_gauge_chart.view* セクションで指定してください。

1. *lesson_gauge_chart* ディレクトリ内に *views* ディレクトリを作成します。
2. *views* ディレクトリ内に *widget.view.php* ファイルを作成します。

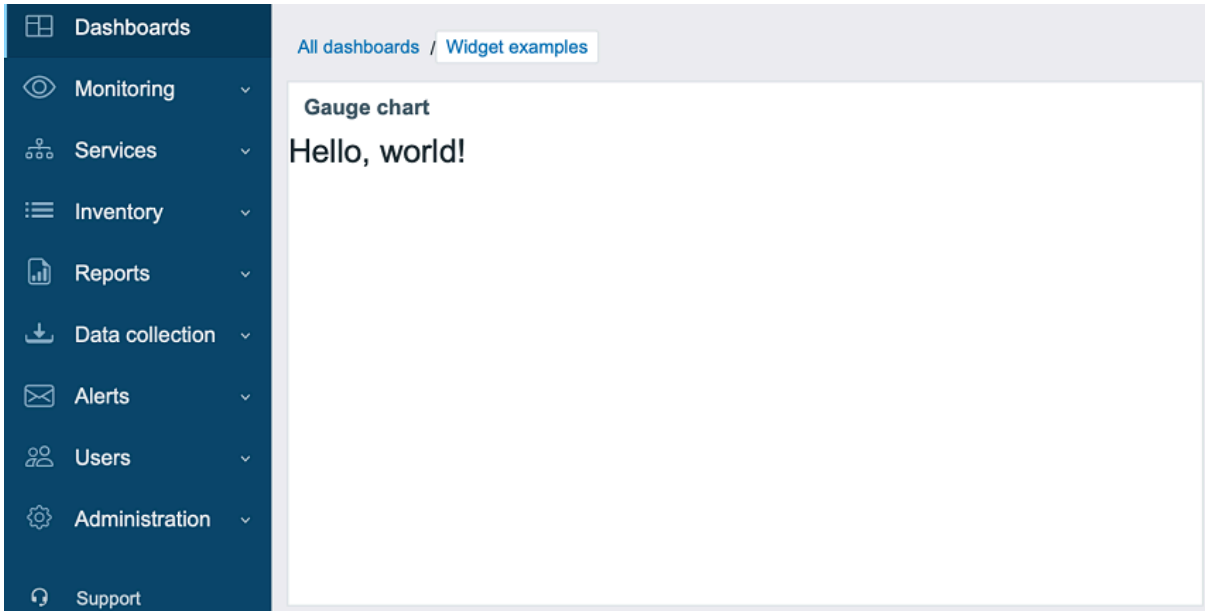
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. ダッシュボードをリフレッシュします。*Gauge chart* ウィジェットに「Hello, world!」が表示されます。



パート **II -** ゲージチャート　設定を設定ビューに追加し、ウィジェットビューで使用する

このセクションでは、ウィジェット設定フィールドを追加し、入力した値をウィジェットビューでテキストとして表示する方法を学びます。

ウィジェットの設定は、フォーム（*Zabbix\Widgets\CWidgetForm*）とウィジェットフォームビュー（*widget.edit.php*）で構成されます。フィールド（*Zabbix\Widgets\CWidgetField*）を追加するには、*Zabbix\Widgets\CWidgetForm* を拡張した *WidgetForm* クラスを作成する必要があります。

フォームには、さまざまな型のフィールド（*Zabbix\Widgets\CWidgetField*）のセットが含まれており、ユーザーが入力した値の検証に使用されます。各入力要素タイプのフォームフィールド（*Zabbix\Widgets\CWidgetField*）は、値を単一の形式に変換してデータベースに保存します。

Note:
ウィジェットのフォームファイルは *includes* ディレクトリ（このチュートリアルでは *ui/modules/lesson_gauge_chart/includes/*）に配置する必要があります。ファイル名がデフォルトの *WidgetForm.php* の場合は、*manifest.json* ファイルに登録する必要はありません。ファイル名が異なる場合は、manifest.jsonファイルの *widget/form_class* セクションで指定してください。

1. *lesson_gauge_chart* ディレクトリに新しい *includes* ディレクトリを作成します。
2. *includes* ディレクトリに *WidgetForm.php* ファイルを作成します。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. ウィジェット設定フォームに *Description* フィールドを追加します。これは通常のテキストフィールドで、ユーザーは任意の文字セットを入力できます。このために *CWidgetFieldTextBox* クラスを使用できます。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

use Zabbix\Widgets\Fields\CWidgetFieldTextBox;
```

```
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
               new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

4. *views* ディレクトリにウィジェット設定ビューのファイル *widget.edit.php* を作成し、新しい *Description* フィールドのビューを追加します。*CWidgetFieldTextBox* フィールドクラスの場合、ビューは *CWidgetFieldTextBoxView* です。

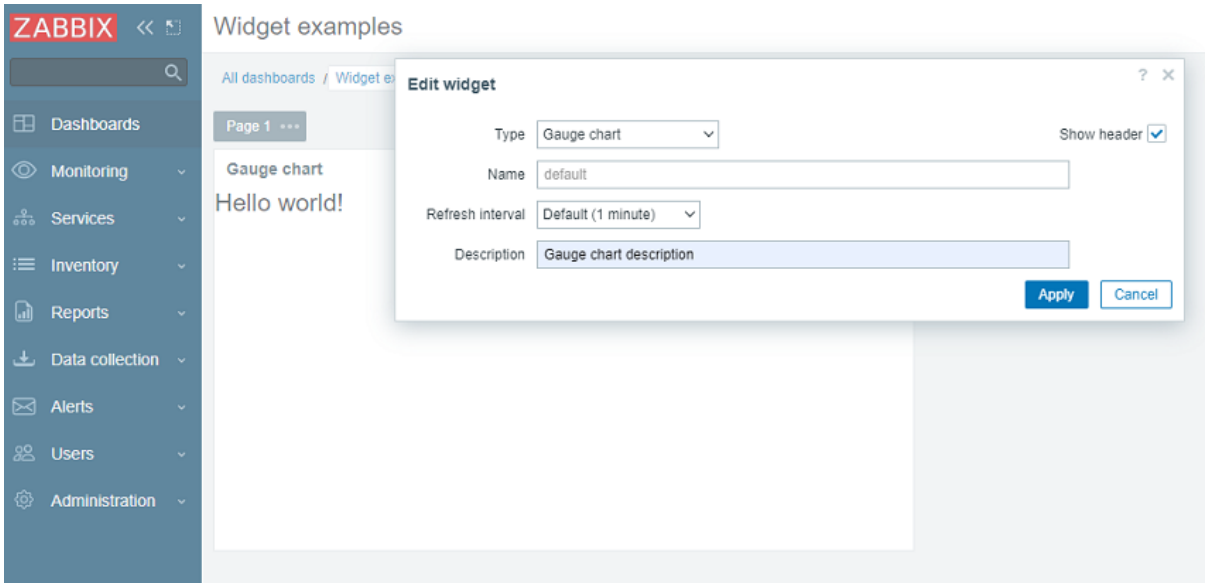
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. ダッシュボードに移動し、ウィジェットの歯車アイコンをクリックしてウィジェット設定フォームを開きます。
6. ウィジェット設定フォームに新しい *Description* テキストフィールドが追加されています。任意の値（例：*Gauge chart description*）を入力します。



7. ウィジェット設定フォームで適用をクリックします。次に、右上の変更を保存をクリックしてダッシュボードを保存します。新しい説明はどこにも表示されず、ウィジェットには引き続き「Hello, world!」と表示されることに注意してください。

新しい説明をウィジェットに表示するには、*Description* フィールドの値をデータベースから取得し、ウィジェットビューに渡す必要があります。そのためには、アクションクラスを作成する必要があります。

8. *lesson_gauge_chart* ディレクトリに新しい *actions* ディレクトリを作成します。
9. *actions* ディレクトリに *WidgetView.php* ファイルを作成します。*WidgetView* アクションクラスは、*CControllerDashboard-WidgetView* クラスを拡張します。

ウィジェット設定フィールドの値は、アクションクラスの **$fields_values** プロパティに格納されます。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. *manifest.json* を開き、*actions/widget.lesson_gauge_chart.view* セクションで *WidgetView* をアクションクラスとして登録します。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. これで、ウィジェットビューで *$data['description']* に含まれる説明フィールドの値を使用できるようになりました。*views/widget.view.php* を開き、静的テキスト「Hello, world!」を *$data['description']* に置き換えます。

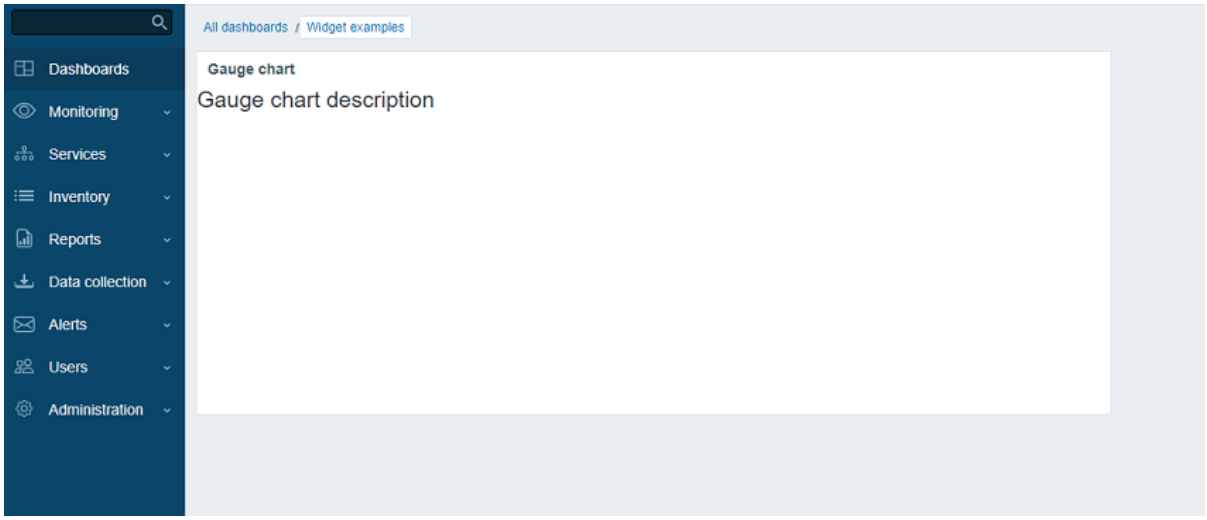
ui/modules/lesson_gauge_chart/views/widget.view.php

```php
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. ダッシュボードページをリフレッシュします。「Hello, world!」の代わりにウィジェットの説明テキストが表示されるはずです。



API 経由でアイテム値を取得する

ウィジェットは、ユーザーが選択したアイテムの最新値を表示する必要があります。そのためには、ウィジェット設定でアイテムを選択できるようにする必要があります。

このセクションでは、ウィジェットフォームにアイテム選択フィールドを追加する方法と、このフィールドのビジュアル部分を設定ビューに追加する方法を学びます。その後、ウィジェットコントローラーは API リクエストを介してアイテムデータとその値を取得できるようになります。受信した値はウィジェットビューに表示できます。

1. *includes/WidgetForm.php* を開き、*CWidgetFieldMultiSelectItem* フィールドを追加します。これにより、設定フォームでアイテムを選択できるようになります。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. *views/widget.edit.php* を開き、設定ビューにフィールドのビジュアルコンポーネントを追加します。

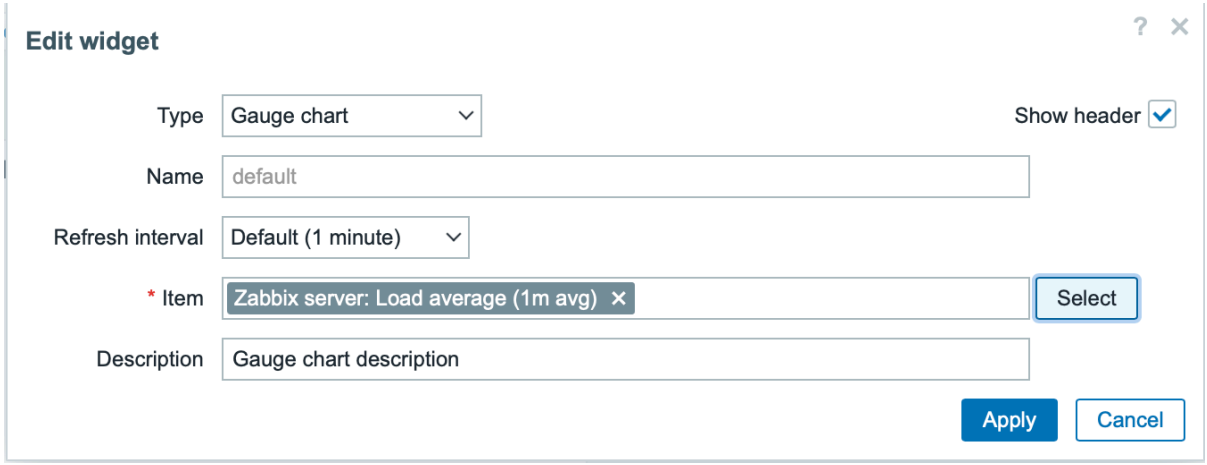
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. ダッシュボードに戻り、ウィジェットの歯車アイコンをクリックしてウィジェット設定フォームを開きます。
4. ウィジェット設定フォームには新しい入力フィールド アイテムが含まれています。ホスト「Zabbix server」とアイテム「Load average (1m avg)」を選択します。



5. ウィジェット設定フォームで 適用をクリックします。次に、右上隅の 変更を保存をクリックしてダッシュボードを保存します。
6. *actions/WidgetView.php* を開いて修正します。

これ以降、アイテム ID はウィジェットコントローラーの *$this->fields_values['itemid']* で利用できるようになります。*doAction()* コントローラーメソッドは、API メソッド *item.get* を使用してアイテムデータ（名前、値タイプ、単位）を収集し、API メソッド *history.get* を使用してアイテムの最新値を取得します。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
```

```
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. *views/widget.view.php* を開き、ウィジェットビューにアイテム値を追加します。

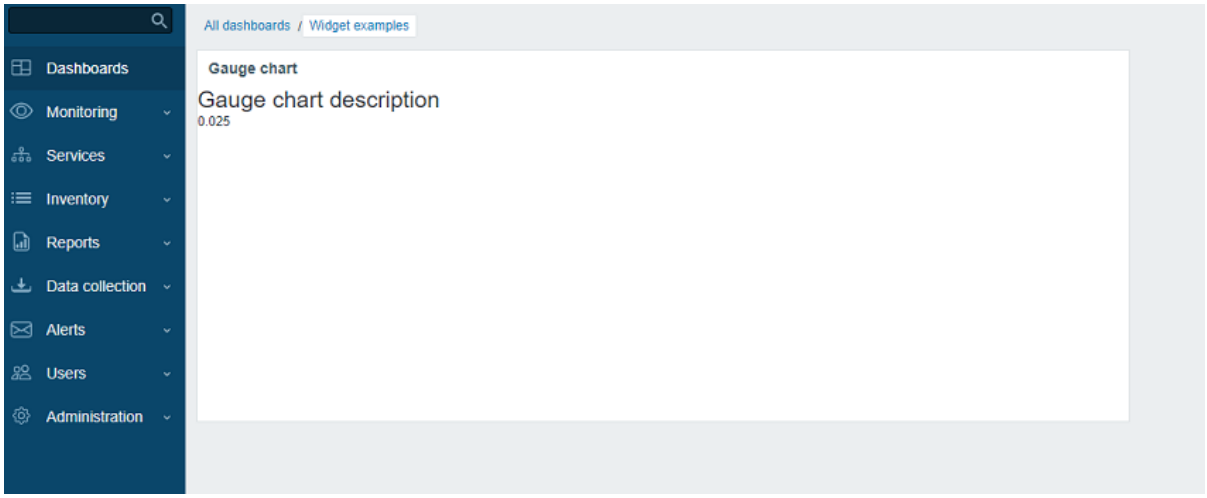
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. ダッシュボードページをリフレッシュします。ウィジェットは最新のアイテム値を表示します。



Advanced configuration を構成ビューに追加

このセクションでは、色、最小値と最大値、単位、前に作成した *Description* フィールドなどのオプションのパラメーターを含む、展開/折りたたみ可能な *Advanced configuration* セクションを追加する方法を学習します。

1. メインウィジェットディレクトリ *lesson_gauge_chart* に *Widget.php* ファイルを作成し、新しいクラス *Widget* を作成します。

Widget クラスは、*CWidget* 基本クラスを拡張して、デフォルトのウィジェット設定 (この場合は翻訳) を追加/上書きします。以下に示す JavaScript は、データが欠落している場合に文字列"No data" を表示します。Zabbix UI 翻訳ファイルには"No data" という文字列が存在します。

ウィジェット定数がある場合は、*Widget* クラスにも指定することをお勧めします。

ui/modules/lesson_gauge_chart/Widget.php

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. *includes/WidgetForm.php* を開き、新しいフィールド *Color*(カラーピッカー)、*Min*(数値フィールド)、*Max*(数値フィールド)、および *Units*(選択) を追加し、カラーピッカーのパレットのデフォルトの色を定義します。次の手順で使用できるようにします。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};
```

```
use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. *views/widget.edit.php* を開き、フィールドのビジュアルコンポーネントを構成ビューに追加します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
```

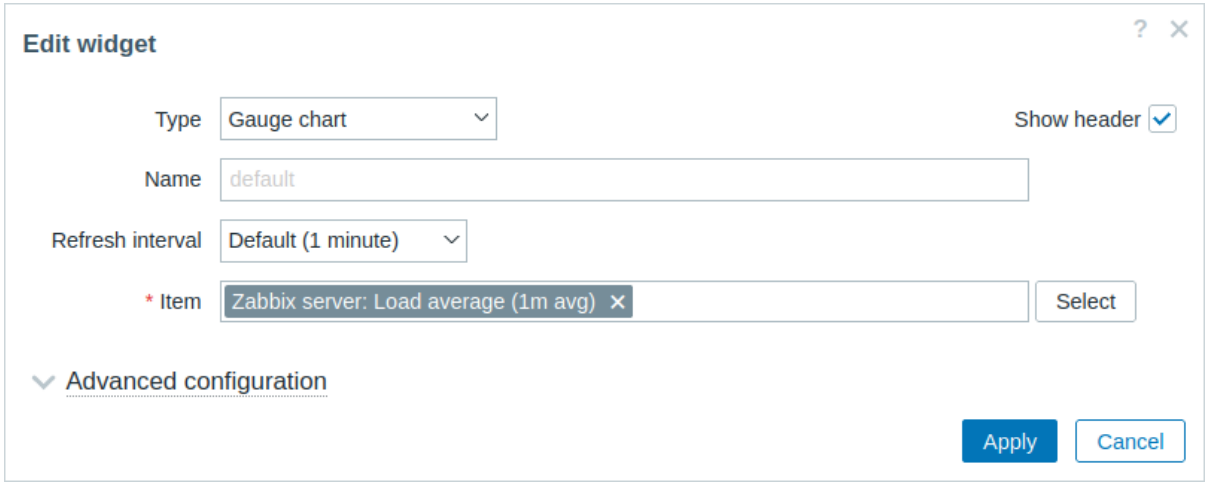
```
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

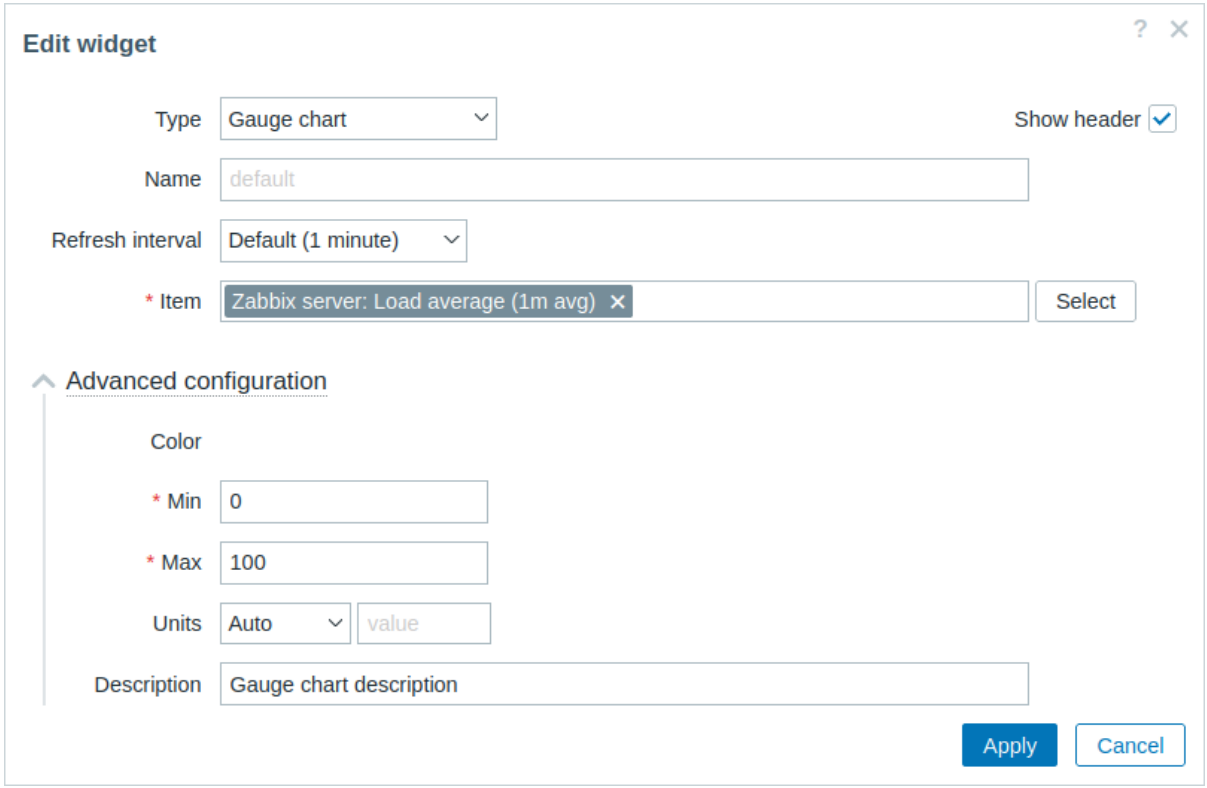
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
CWidgetFormView クラスの *addField()* メソッドは、CSS クラス文字列を 2 番目のパラメーターとして受け取ります。

4. ダッシュボードに戻り、編集モードに切り替えて、ウィジェットの歯車アイコンをクリックしてウィジェット構成フォームを開きます。ウィジェット設定フォームには、新しい展開/折りたたみ可能なセクション *Advanced configuration* が含まれるようになりました。



5. *Advanced configuration* セクションを展開すると、追加のウィジェット構成フィールドが表示されます。*Color* フィールドにはまだカラーピッカーがないことに注意してください。これは、カラーピッカーを JavaScript で初期化する必要があるためです。これは、次のセクションウィジェットに *JavaScript* を追加で追加されます。



ウィジェットに JavaScript を追加する

このセクションでは、最新値が正常か高すぎる/低すぎるかを示す JavaScript で作成したゲージチャートを追加する方法を学びます。

1. *views* ディレクトリに *widget.edit.js.php* ファイルを作成します。

JavaScript は、設定ビューでカラーピッカーを初期化する役割を担います。

ui/modules/lesson_gauge_chart/views/widget.edit.js.php

```
<?php

use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._unit_select = document.getElementById('value_units');
        this._unit_value = document.getElementById('value_static_units');

        this._unit_select.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
    }

    updateForm() {
```

```
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. *views/widget.edit.php* を開き、JavaScript を含む *widget.edit.js.php* ファイルを設定ビューに追加します。これを行うには、*includeJsFile()* メソッドを使用します。インライン JavaScript を追加するには、*addJavaScript()* メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

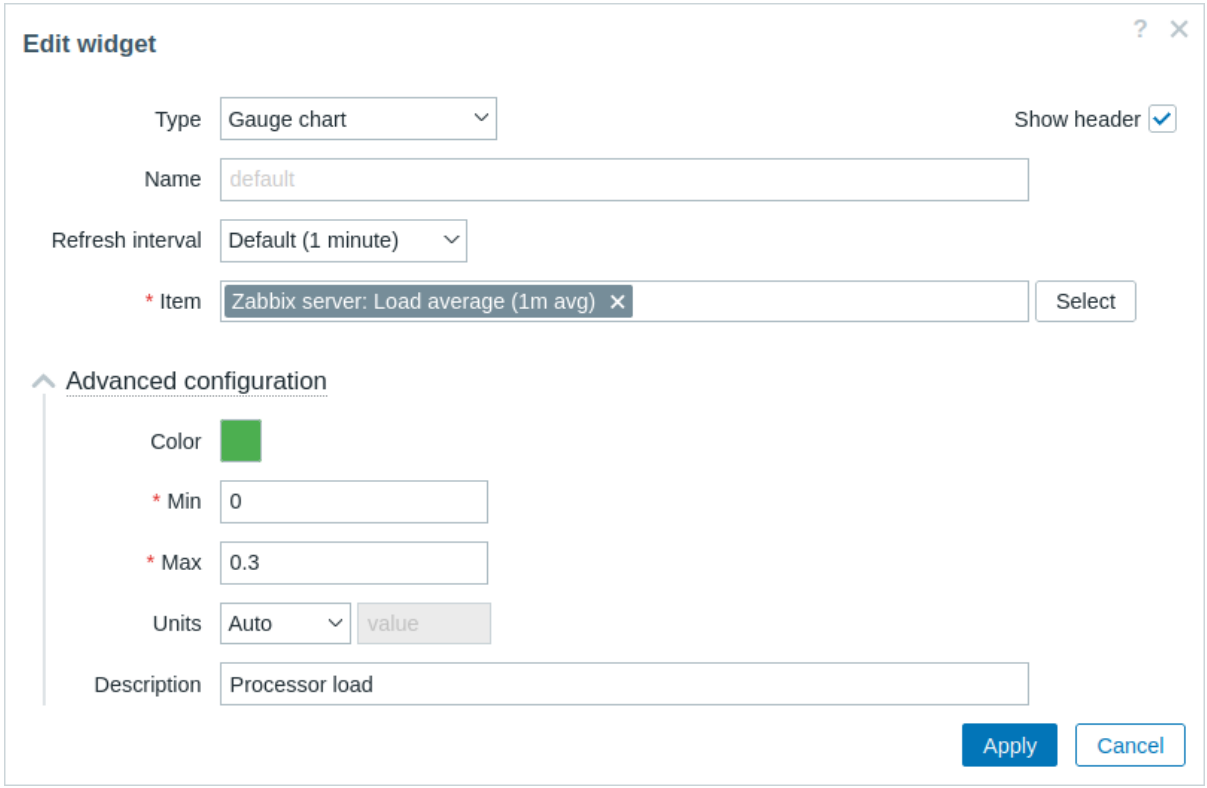
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. ダッシュボードに戻り、ウィジェットの歯車アイコンをクリックしてウィジェット設定フォームを開きます。*Advanced configuration* セクションを展開すると、初期化されたカラーピッカーが表示されます。フィールドに値を入力し、ゲージチャートの色を選択します。



4. ウィジェット設定フォームで適用をクリックします。次に、右上隅の変更を保存をクリックしてダッシュボードを保存します。
5. *actions/WidgetView.php* を開いてコントローラーを更新します。

$this->fields_values プロパティには、*Advanced configuration* フィールドのすべての値が含まれるようになりました。コントローラーを完成させて、設定と選択したアイテム値をウィジェットビューに渡せるようにします。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
```

```
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. *views/widget.view.php* を開いて修正します。

次のステップで描画するゲージチャート用のコンテナと、説明用のコンテナを作成する必要があります。

値を JavaScript に JSON オブジェクトとして渡すには、*setVar()* メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. *lesson_gauge_chart* ディレクトリに新しいディレクトリ *assets* を作成します。このディレクトリは、JavaScript、CSS、フォントや画像などの他のアセットを保存するために使用されます。
8. ウィジェットビュー用の JavaScript のために、*assets* ディレクトリに *js* ディレクトリを作成します。
9. *assets/js* ディレクトリに *class.widget.js* ファイルを作成します。

この JavaScript ウィジェットクラスは、すべてのダッシュボードウィジェットの基本 JavaScript クラスである *CWidget* を拡張します。

ダッシュボードはウィジェットの正しい実装に依存しており、関連する情報をウィジェットに渡すためにそれぞれの JavaScript メソッドを呼び出します。また、ダッシュボードは何らかの操作が発生したときにウィジェットがイベントを生成することを期待しています。そのため、*CWidget* クラスにはウィジェットの動作のデフォルト実装を持つメソッド群が含まれており、クラスを拡張することでカスタマイズできます。

この場合、いくつかのカスタマイズが必要なため、以下のウィジェット動作についてカスタムロジックを実装します:

- ウィジェットの初期化（*onInitialize()* メソッド参照）

- ウィジェット更新処理が成功しエラーがない場合のウィジェット内容の表示（ゲージチャートの描画）（*processUpdateResponse(response)* メソッドおよび関連する *_resizeChart()*、*_updatedChart()* メソッド参照）
- ウィジェットのリサイズ（*onResize()* メソッドおよび関連する *_resizeChart()* メソッド参照）

ゲージチャートウィジェットの他の側面については、ウィジェット動作のデフォルト実装を使用します。*CWidget* クラスの JavaScript メソッドの詳細については、JavaScriptを参照してください。

この JavaScript はウィジェットビューに必要なため、ダッシュボードページとともに読み込まれる必要があります。JavaScript の読み込みを有効にするには、ステップ 10 で示すように ***manifest.json*** ファイルの *assets/js* および *js_class* パラメータを更新する必要があります。

ui/modules/lesson_gauge_chart/assets/js/class.widget.js

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
        }
```

```
        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);

                ctx.beginPath();
```

```
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. *manifest.json* を開き、以下を追加します:

- *assets/js* セクションの配列にファイル名（*class.widget.js*）
- *widget* セクションの *js_class* パラメータにクラス名（*WidgetLessonGaugeChart*）

WidgetLessonGaugeChart クラスはダッシュボードとともに自動的に読み込まれるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
```

```
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

CSS スタイルをウィジェットに追加

このセクションでは、カスタム CSS スタイルを追加してウィジェットをより魅力的に見せる方法を学びます。

1. ウィジェットスタイルの場合は、*assets* ディレクトリに新しいディレクトリ *css* を作成します。
2. *assets/css* ディレクトリに *widget.css* ファイルを作成します。ウィジェット要素のスタイルを設定するには、セレクター *div.dashboard-widget-{widget id}* を使用します。ウィジェット全体の CSS を設定するには、セレクター *form.dashboard-widget-{widget id}* を使用します。

ui/modules/lesson_gauge_chart/assets/css/widget.css

```
div.dashboard-widget-lesson_gauge_chart {
    display: grid;
    grid-template-rows: 1fr;
    padding: 0;
}

div.dashboard-widget-lesson_gauge_chart .chart {
    display: grid;
    align-items: center;
    justify-items: center;
}

div.dashboard-widget-lesson_gauge_chart .chart canvas {
    background: white;
}

div.dashboard-widget-lesson_gauge_chart .description {
    padding-bottom: 8px;
    font-size: 1.750em;
    line-height: 1.2;
    text-align: center;
}

.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

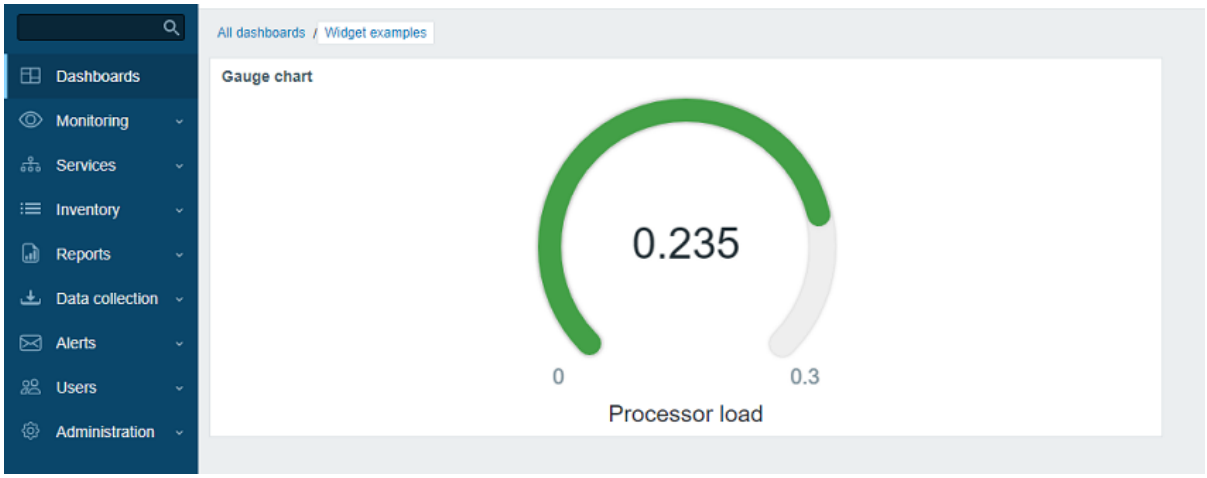
3. *manifest.json* を開き、CSS ファイル名 (*widget.css*) を *assets/css* セクションの配列に追加します。これにより、*widget.css* で定義された CSS スタイルをダッシュボードページに読み込むことができるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
```

```
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. ダッシュボードページを更新して、ウィジェットの完成版を確認します。



例

このセクションでは、カスタムモジュールのベースとして使用できる、サンプルモジュールとウィジェットのファイルを提供します。

モジュールの使用方法：

1. ZIP アーカイブをダウンロードします。
2. Zabbix Web インターフェースをインストールの *modules* ディレクトリ内の別のディレクトリにコンテンツを解凍します (例: */zabbix/ui/modules*)。
3. Zabbix Web インターフェースにモジュールを登録します。

モジュールの例

- ホストグループ作成時に、設定されたユーザーグループに読み取り権限を付与します - hg_auto_perm.zip

ウィジェットの例

- 最小限のウィジェット - widget_min.zip
- CSS のみを使用した”Hello, world” ウィジェット - hello_world_css.zip
- JavaScript のみを使用した”Hello, world” ウィジェット - hello_world_js.zip
- PHP を使用した”Hello, world” ウィジェット - hello_world_php.zip

Note:
Zabbix ネイティブウィジェットも例として利用できます。

プラグイン

概要　カスタムのロード可能なプラグインは、Zabbix エージェント 2 の機能を拡張します。これらは個別にコンパイルされますが、Zabbix エージェント 2 と共有されるパッケージを使用します。

各プラグインは、構造を定義し、1 つまたは複数のプラグインインターフェース (*Exporter*、*Configurator*、*Runner*) を実装する *Go* パッケージです。

ジャンプ先:

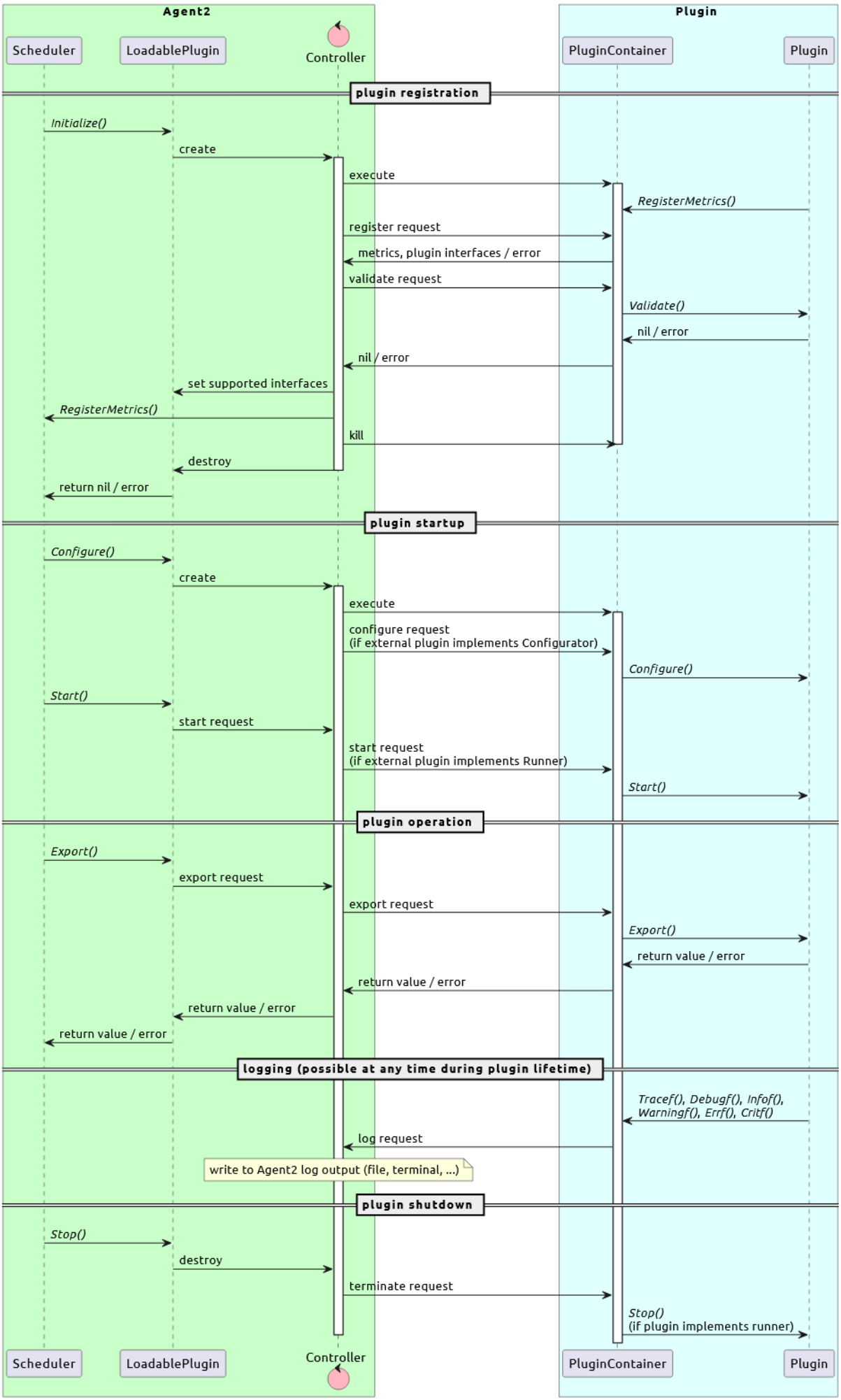
- 最初のプラグインを作成する
- プラグインインターフェース

参照:

- Zabbix エージェント 2 のプラグインの例

接続図　Zabbix エージェント 2 は、Linux では UNIX ソケット、Windows では名前付きパイプを使用して、プラグインに双方向に接続します。

以下の接続図は、Zabbix エージェント 2 とロード可能なプラグイン間の通信プロセスと、メトリクス収集プロセスを示しています。


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

例

いくつかの空のサンプルと、Zabbix が参照として提供する既存の読み込み可能なプラグインを使用できます:

- Zabbix エージェント 2 のプラグインの例
- MSSQl プラグイン
- MongoDB プラグイン
- PostgreSQL プラグイン

プラグインの作成 **(**チュートリアル**)**

これは、Zabbix エージェント 2 用のシンプルでロード可能なプラグインを作成するためのステップバイステップのチュートリアルです。

独自のプラグインを作成するためのテンプレートまたはガイドとして、サンプルリポジトリを使用することもできます。

作成するもの　このチュートリアルでは、新しいロード可能なプラグイン **MyIP** を作成する方法を説明します。このプラグインは、Zabbix エージェント 2 が動作しているホストの外部 IP アドレスを返す単一のアイテムキー **myip** を実装します。

ステップ **1:** セットアップ

1. プラグインは標準的な Go モジュールです。まず、プラグインの依存関係を追跡するために、プラグインディレクトリ内の go.mod ファイルを初期化します。

```
cd path/to/plugins/myip # プラグインディレクトリに移動します。
go mod init myip
```

2. 必須の依存関係である Zabbix Go SDK (golang.zabbix.com/sdk) をインストールします。

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

$LATEST_COMMIT_HASH を、適切なリリースブランチの golang.zabbix.com/sdkリポジトリにある最新の HEAD コミットハッシュに置き換えます。例:

```
go get golang.zabbix.com/sdk@af85407
```

golang.zabbix.com/sdk のバージョン管理は現在サポートされていませんが、将来変更される可能性があります。

必要に応じて、go get を使用して追加の依存関係をインストールできます。

3. プラグインのソースコード用の空の main.go ファイルを作成します。

```
touch main.go
```

これで初期設定が完了し、プラグインの開発準備が整いました。

ステップ **2:** プラグインの構造　前のステップでインストールした golang.zabbix.com/sdk モジュールは、プラグイン開発に必要なフレームワークを提供し、すべてのプラグインの一貫した構造を保証します。

1. 基本的な実行フローを設定します。

まず、プラグインのメインの実行フローを定義します。次のコードを main.go に追加します。

```
package main

func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
return nil
}
```

これで、プラグインの基本的な実行フローが確立されます。run 関数には、後でプラグインのコアロジックが含まれます。

2. プラグインのインターフェースを確認します。

Zabbix エージェント 2 プラグインは、golang.zabbix.com/sdk/plugin パッケージのインターフェースを実装する構造体で表現されます。

- *Accessor* - プラグイン名の設定やアイテムキーのタイムアウト処理など、すべてのプラグインが実装する必要がある必須メソッドを定義します。
- 以下の機能プラグインインターフェース (1 つ以上) :
  - *Exporter* - ポーリングを実行し、値 (複数可)、何も返さない、またはエラーを返します。多くの場合、*Collector* インターフェースと併用されます。
  - *Collector* - 定期的なデータ収集を管理します。
  - *Runner* - プラグインの起動およびシャットダウン手順を定義します。
  - *Watcher* - エージェントの内部スケジューラをバイパスし、独立したメトリックポーリングを実装できます。トラップベースまたはイベントドリブンの監視に便利です。
  - *Configurator* - プラグインが設定を読み込んで適用する方法を定義します。

これらのインターフェースは独自に実装することも、Zabbix Go SDK が提供するデフォルトのインターフェースを必要に応じて変更して使用することもできます。このチュートリアルではデフォルトの実装を使用します。

3. プラグイン構造体を作成します。

plugin パッケージをインポートし、plugin.Base 構造体を埋め込む myIP 構造体を作成します。

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
plugin.Base
}
```

myIP 構造体は現在、Accessor インターフェースを満たしています。関数型プラグインインターフェースの 1 つである Exporter を実装するメソッドは、このチュートリアルの後半で追加します。

ステップ **3:** アイテムキーの定義　プラグインは、データを収集し、Zabbix サーバーまたはプロキシに提供するためにアイテムキーが必要です。

1. エラー処理用の *errs* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/errs"
```

2. run() 関数内の plugin.RegisterMetrics() 関数を使用して、アイテムキーを登録します。

```
func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                             // プラグイン名
"myip",                             // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

return nil
}
```

複数のアイテムキーを登録するには、各メトリックに対してパラメーター *metric name* と *description* を繰り返します。例:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

ステップ **4:** ハンドラーの設定　ハンドラーは、エージェントとプラグイン間の通信を容易にします。

1. *container* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. run() 関数に、ハンドラーの作成と設定を行うコードを追加します。

```
func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                             // プラグイン名
"myip",                             // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// 新しいハンドラーを作成します。
h, err := container.NewHandler("MyIP") // プラグイン名
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// プラグインからエージェントへログを転送するためのログ記録を設定します。
// p.Logger.Infof、p.Logger.Debugf などから利用可能です。
p.Logger = h

// プラグインの実行を開始します。
// エージェントから終了要求を受信するまでブロックします。
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

return nil
}
```

ステップ **5:** データ収集の実装　データ収集は、Export メソッドを記述する Exporter インターフェースを介して行われます。

```
func Export(
  key string,             // 収集するアイテムキー
  params []string,        // アイテムキー（`myip[arg1, arg2]`）に渡される引数
  context ContextProvider // アイテムキーデータ収集に関するメタデータ
) (any, error)
```

1. HTTP リクエストとレスポンスの読み取りに必要なパッケージをインポートします。

```
import (
"io"
"net/http"
)
```

2. myIP 構造体の Export メソッドを実装します。

```
func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// プラグインは、`key`パラメーターに基づいて異なるデータ収集ロジックを使用できます。
    // この実装は、指定された`key`がサポートされているかどうかのみを検証します。
if key != "myip" {
return nil, errs.Errorf("unknown item key %q", key)
}

// ログはエージェント 2 のログに転送されます。
p.Infof(
"received request to handle %q key with %d parameters",
```

```
key,
len(params),
)

// データを収集して返します。

resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "failed to get IP address")
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

ステップ **6:** プラグインのビルドと設定

1. プラグインをビルドするには、次のコマンドを実行します。

```
go mod tidy
go build
```

これにより、カレントディレクトリに実行ファイル `myip` が作成されます。

2. プラグインを使用するように Zabbix エージェント 2 を設定します。

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

`$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` を、ステップ 5 で作成した `myip` へのパスに置き換えます。

設定パラメーター名 (このチュートリアルでは _MyIP_) のプラグイン名は、_plugin.RegisterMetrics()_ 関数で定義されたプラグイン名と一致する必要があります。

3. プラグインとその `myip` アイテムをテストするには、次のコマンドを実行します。

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

出力にはホストの外部 IP アドレスが含まれ、以下のようになります。

```
myip                                            [s|192.0.2.0]
```

これで、Zabbix エージェント 2 用のシンプルな読み込み可能なプラグインが作成されました。おめでとうございます！

```
package main

import (
"io"
"net/http"

"golang.zabbix.com/sdk/errs"
"golang.zabbix.com/sdk/plugin"
"golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
plugin.Base
}
```

```
func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                             // プラグイン名
"myip",                             // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// 新しいハンドラーを作成します。
h, err := container.NewHandler("MyIP") // Plugin name
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// プラグインからエージェントにログを転送するためのログ設定を行います。
// p.Logger.Infof、p.Logger.Debugf などから利用可能です。
p.Logger = h

// プラグインの実行を開始します。
// エージェントからの終了要求を受信するまでブロックします。
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

return nil
}

func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// プラグインは、`key`パラメーターに基づいて異なるデータ収集ロジックを使用できます。
    // この実装では、指定された`key`がサポートされているかどうかのみ確認します。
if key != "myip" {
return nil, errs.Errorf("unknown item key %q", key)
}

// ログはエージェント 2 のログに転送されます。
p.Infof(
"received request to handle %q key with %d parameters",
key,
len(params),
)

// データを収集して返します。

resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "failed to get IP address")
```

```
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

完全なソースコード

プラグインインターフェース

このセクションでは、利用可能なプラグインインターフェースについて説明します。

plugin.Exporter *Exporter* は、ポーリングを実行して値 (複数可)、何も返さない、またはエラーを返す最もシンプルなインターフェースです。これは、事前に解析されたアイテムキー、パラメーター、およびコンテキストを受け入れます。他のすべてのプラグインインターフェースへのアクセスは排他的であり、プラグインがすでにタスクを実行している場合はメソッドを呼び出すことはできません。また、プラグインごとに最大 100 回の同時 *Export()* 呼び出しの制限がありますが、これは各プラグインの要件に応じて減らすことができます。

plugin.Configurator *Configurator* インターフェースは、Zabbix エージェント 2 の設定ファイルからプラグイン設定パラメーターを提供します。

plugin.Runner *Runner* インターフェースは、プラグインの開始時 (アクティブ化) に初期化を実行し、プラグインの停止時 (非アクティブ化) に非初期化を実行する手段を提供します。たとえば、プラグインは Runner インターフェースを実装することで、バックグラウンドの *goroutine* を開始/停止できます。

拡張機能開発の変更点

このページでは、カスタム Zabbix 拡張機能の開発に関するすべての変更点（ある場合）を一覧表示します。

Zabbix リファレンスマニュアル

こちらは Zabbix プロセスの man ページの内容です。

zabbix_agent2

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_agent2 - Zabbix エージェント 2

概要

zabbix_agent2 [**-c** *config-file*]
zabbix_agent2 [**-c** *config-file*] [**-v**] **-p**
zabbix_agent2 [**-c** *config-file*] [**-v**] **-t** *item-key*
zabbix_agent2 [**-c** *config-file*] **-R** *runtime-option*
zabbix_agent2 [**-c** *config-file*] **-T**
zabbix_agent2 -h
zabbix_agent2 -V

説明

zabbix_agent2 は、様々なサービスのパラメーターを監視するためのアプリケーションです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに、別の *config-file* を使用します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

実行時制御オプション**:** **userparameter_reload**
設定ファイルからユーザーパラメータを再読み込み

log_level_increase
ログレベルを上げる

log_level_decrease
ログレベルを下げる

help
利用可能な実行時制御オプションを一覧表示

metrics
利用可能なメトリクスを一覧表示

version
バージョンを表示

-p, **--print**
既知のアイテムを表示して終了します。各アイテムについては、一般的なデフォルト値が使用されるか、テスト用の特定のデフォルト値が指定されます。これらのデフォルト値は、アイテムキーのパラメータとして角括弧で表示されます。返される値は角括弧で囲まれ、返される値の型がパイプ文字で区切って先頭に付加されます。ユーザーパラメータの場合、型は常に **t** です。これは、エージェントがすべての可能な戻り値を判別できないためです。動作していると表示されているアイテムでも、Zabbix サーバーや zabbix_get から実行中のエージェントデーモンに問い合わせた場合、権限や環境が異なるため、必ずしも動作するとは限りません。返される値の型は以下の通りです:

d
小数部を持つ数値。

m
サポートされていません。これは、ログ監視アイテムのようにアクティブモードでのみ動作するアイテムや、複数の収集値を必要とするアイテムを問い合わせた場合に発生する可能性があります。権限の問題や不正なユーザーパラメータも、サポートされていない状態になる原因となります。

s
テキスト。最大長は無制限です。

t
テキスト。**s** と同じです。

u
符号なし整数。

-t, **--test** *item-key*
単一のアイテムをテストして終了します。出力の説明については **--print** を参照してください。

-v, **--verbose**
テストアイテムまたは既知のアイテムを表示する際に詳細出力を有効にします。

-T, **--test-config**
設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agent2.conf
Zabbix agent 2 の設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

目次

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
FILES
SEE ALSO
AUTHOR

このドキュメントは 2021 年 11 月 22 日 14:07:57 GMT に作成されました。

zabbix_agentd

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_agentd - Zabbix エージェントデーモン

概要

zabbix_agentd [**-c** *config-file*]
zabbix_agentd [**-c** *config-file*] **-p**
zabbix_agentd [**-c** *config-file*] **-t** *item-key*
zabbix_agentd [**-c** *config-file*] **-R** *runtime-option*
zabbix_agentd [**-c** *config-file*] **-T**

zabbix_agentd -h
zabbix_agentd -V

説明

zabbix_agentd は、様々なサーバーのパラメーターを監視するためのデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix エージェントをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

ランタイム制御オプション　**userparameter_reload**
設定ファイルからユーザーパラメータを再読み込みする

log_level_increase[=*target*]
ログレベルを上げる。target が指定されていない場合は全プロセスに影響する

log_level_decrease[=*target*]
ログレベルを下げる。target が指定されていない場合は全プロセスに影響する

ログレベル制御ターゲット　*process-type*
指定されたタイプのすべてのプロセス（アクティブチェック、コレクター、リスナー）

process-type,N
プロセスタイプと番号（例：listener,3）

pid
プロセス識別子。最大 65535 まで。より大きな値の場合は、ターゲットを”process-type,N” として指定してください。

-p, **--print**
既知のアイテムを表示して終了します。各アイテムについて、一般的なデフォルト値が使用されるか、テスト用の特定のデフォルト値が提供されます。これらのデフォルト値は、アイテムキーのパラメータとして角括弧で表示されます。返される値は角括弧で囲まれ、返される値の型がパイプ文字で区切られて先頭に付加されます。ユーザーパラメータの場合、型は常に **t** です。これは、エージェントがすべての可能な戻り値を判別できないためです。動作していると表示されるアイテムでも、Zabbix サーバーや zabbix_get から実行中のエージェントデーモンに問い合わせた場合、権限や環境が異なるため、動作が保証されるわけではありません。返される値の型は以下の通りです。

d
小数部を持つ数値。

m
サポートされていません。これは、ログ監視アイテムのようなアクティブモードでのみ動作するアイテムや、複数の収集値を必要とするアイテムを問い合わせた場合に発生する可能性があります。権限の問題や不正なユーザーパラメータも、サポートされていない状態になる原因となります。

s
テキスト。最大長は無制限です。

t
テキスト。**s** と同じです。

u
符号なし整数。

-t, **--test** *item-key*
単一のアイテムをテストして終了します。出力の説明については **--print** を参照してください。

-T, **--test-config**
設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agentd.conf
Zabbix エージェントの設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

目次

NAME

SYNOPSIS

DESCRIPTION

OPTIONS

FILES

SEE ALSO

AUTHOR

このドキュメントは 2021 年 11 月 22 日 20:50:13 GMT に作成されました

zabbix_get

セクション: ユーザーコマンド (1)
更新日: 2022-01-06
目次 メインコンテンツに戻る

名前

zabbix_get - Zabbix get ユーティリティ

概要

zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

説明

zabbix_get は、Zabbix エージェントからデータを取得するためのコマンドラインユーティリティです。

オプション

-s, **--host** ホスト名または *IP*
ホストのホスト名または IP アドレスを指定します。

-p, **--port** ポート番号
ホスト上で動作しているエージェントのポート番号を指定します。デフォルトは 10050 です。

-I, **--source-address** *IP* アドレス
送信元 IP アドレスを指定します。

-t, **--timeout** 秒数
タイムアウトを指定します。有効範囲: 1-600 秒 (デフォルト: 30)

-k, **--key** アイテムキー
値を取得するアイテムのキーを指定します。

-P, **--protocol** 値
エージェントとの通信に使用するプロトコル。値:

auto JSON プロトコルで接続し、失敗した場合はプレーンテキストプロトコルで再試行します（デフォルト）

json JSON プロトコルで接続します

plaintext アイテムキーのみを送信するプレーンテキストプロトコルで接続します（6.4.x 以前のリリース）

--tls-connect 値
エージェントへの接続方法。値:

unencrypted
暗号化なしで接続します（デフォルト）

psk
TLS と事前共有鍵を使用して接続します

cert
TLS と証明書を使用して接続します

--tls-ca-file *CA-file*
ピア証明書の検証のための最上位 CA 証明書を含むファイルのフルパス名。

--tls-crl-file *CRL-file*
失効した証明書を含むファイルのフルパス名。

--tls-agent-cert-issuer *cert-issuer*
許可されたエージェント証明書の発行者。

--tls-agent-cert-subject *cert-subject*
許可されたエージェント証明書のサブジェクト。

--tls-cert-file *cert-file*
証明書または証明書チェーンを含むファイルのフルパス名。

--tls-key-file *key-file*
秘密鍵を含むファイルのフルパス名。

--tls-psk-identity *PSK-identity*
PSK-identity 文字列。

--tls-psk-file *PSK-file*
事前共有鍵を含むファイルのフルパス名。

--tls-cipher13 *cipher-string*
TLS 1.3 用の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準を上書きします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは利用できません。

--tls-cipher *cipher-string*
GnuTLS の優先順位文字列（TLS 1.2 以降）または OpenSSL の暗号文字列（TLS 1.2 のみ）。デフォルトの暗号スイート選択基準を上書きします。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

例

zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]”
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-agent-cert-issuer ”CN=Signing CA,OU=IT operations,O=Example Corp,DC=example,DC=com” --tls-agent-cert-subject ”CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com” --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

目次

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
EXAMPLES
SEE ALSO
AUTHOR

このドキュメントは 2021 年 6 月 11 日 08:42:29 GMT に作成されました。

zabbix_js

セクション: ユーザーコマンド (1)
更新日: 2022-01-06
索引 メインコンテンツに戻る

名前

zabbix_js - Zabbix JS ユーティリティ

概要

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

説明

zabbix_js は、埋め込みスクリプトのテストに使用できるコマンドラインユーティリティです。

オプション

-s, **--script** *script-file*
実行するスクリプトのファイル名を指定します。
ファイル名に'-' を指定した場合、スクリプトは標準入力から読み込まれます。

-p, **--param** *input-param*
入力パラメータを指定します。

-i, **--input** *input-file*
入力パラメータのファイル名を指定します。
ファイル名に'-' を指定した場合、入力は標準入力から読み込まれます。

-w, **--webdriver** *url*
webdriver の URL を指定します。

-l, **--loglevel** *log-level*
ログレベルを指定します。

-t, **--timeout** *timeout*
タイムアウトを秒単位で指定します。
有効範囲: 1-600 秒 (デフォルト: 10)

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を表示して終了します。

例

zabbix_js -s script-file.js -p example

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

目次

名前
概要
説明
オプション
例
参照

ドキュメント作成日: 2020 年 3 月 18 日 21:23:35 GMT

zabbix_proxy

セクション: メンテナンスコマンド (8)
更新日: 2020-09-04
目次 メインコンテンツに戻る

名前

zabbix_proxy - Zabbix プロキシデーモン

概要

zabbix_proxy [**-c** *config-file*]
zabbix_proxy [**-c** *config-file*] **-R** *runtime-option*
zabbix_proxy [**-c** *config-file*] **-T**
zabbix_proxy -h
zabbix_proxy -V

説明

zabbix_proxy は、デバイスから監視データを収集し、Zabbix サーバーに送信するデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix プロキシをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

ランタイム制御オプション　**config_cache_reload**
設定キャッシュをリロードします。
キャッシュが現在ロード中の場合は無視されます。
アクティブな Zabbix プロキシは Zabbix サーバーに接続し、設定データを要求します。
デフォルトの設定ファイル (**-c** オプションが指定されていない場合) が使用され、PID ファイルを見つけて、PID ファイルに記載されたプロセスにシグナルが送信されます。

snmp_cache_reload
SNMP キャッシュをリロードします。
Zabbix は SNMPv3 の EngineID→IP マッピングをキャッシュし、ポーリングのオーバーヘッドを減らすために EngineID を自動的に再利用する場合があることに注意してください。

housekeeper_execute
ハウスキーパーを実行します。
ハウスキーパーが現在実行中の場合は無視されます。

diaginfo[=*section*]
指定したセクションの内部診断情報をログに記録します。
セクションには *historycache*、*preprocessing*、*locks* を指定できます。
デフォルトでは、すべてのセクションの診断情報がログに記録されます。

log_level_increase[=*target*]
ログレベルを上げます。target が指定されていない場合はすべてのプロセスに影響します。

log_level_decrease[=*target*]
ログレベルを下げます。target が指定されていない場合はすべてのプロセスに影響します。

ログレベル制御ターゲット　*process-type*
指定したタイプのすべてのプロセス（availability manager、browser poller、configuration syncer、data sender、discovery manager、history syncer、housekeeper、http poller、icmp pinger、ipmi manager、ipmi poller、java poller、odbc poller、poller、agent poller、http agent poller、snmp poller、preprocessing manager、self-monitoring、snmp trapper、task manager、trapper、unreachable poller、vmware collector）

process-type,N
プロセスタイプと番号（例：poller,3）

pid
プロセス識別子（最大 65535）。より大きな値の場合は、ターゲットを”process-type,N” と指定してください。

プロファイリング制御ターゲット

process-type
指定したタイプのすべてのプロセス（configuration syncer、data sender、discovery manager、history syncer、housekeeper、http poller、preprocessing manager、icmp pinger、ipmi manager、ipmi poller、java poller、poller、agent poller、http agent poller、snmp poller、self-monitoring、snmp trapper、task manager、trapper、unreachable poller、vmware collector、history poller、availability manager、odbc poller）

process-type,N
プロセスタイプと番号（例：history syncer,1）

pid
プロセス識別子（最大 65535）。より大きな値の場合は、ターゲットを”process-type,N” と指定してください。

scope
プロファイリングスコープ（rwlock、mutex、processing）はプロセスタイプと組み合わせて使用できます（例：history syncer,1,processing）

-T, **--test-config**
設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を表示して終了します。

ファイル

/usr/local/etc/zabbix_proxy.conf
Zabbix プロキシの設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

目次

NAME

SYNOPSIS

DESCRIPTION

OPTIONS

FILES

SEE ALSO

AUTHOR

このドキュメントは 2020 年 9 月 4 日 16:12:22 GMT に作成されました。

zabbix_sender

セクション: ユーザーコマンド (1)
更新日: 2021-06-01
目次 メインコンテンツに戻る

名前

zabbix_sender - Zabbix sender ユーティリティ

概要

zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] **-s** *host* **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] **-s** *host* **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] **-s** *host* **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP アドレス*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender -h
zabbix_sender -V

説明

zabbix_sender は、監視データを Zabbix サーバーまたはプロキシに送信するためのコマンドラインユーティリティです。Zabbix サーバーでは、**Zabbix trapper** タイプのアイテムに対応するキーで作成する必要があります。受信データは、このアイテムの許可されたホストフィールドで指定されたホストからのみ受け入れられることに注意してください。

オプション

-c, **--config** *config-file*
config-file を使用します。**Zabbix sender** はサーバーの詳細を agentd の設定ファイルから読み取ります。デフォルトでは **Zabbix sender** は設定ファイルを読み取りません。サポートされているパラメーターは **Hostname**、**ServerActive**、**SourceIP**、**TLSConnect**、**TLSCAFile**、**TLSCRLFile**、**TLSServerCertIssuer**、**TLSServerCertSubject**、**TLSCertFile**、**TLSKeyFile**、**TLSPSKIdentity**、**TLSPSKFile** のみです。**HostnameItem** パラメーターで定義されたホスト名は取得されません。この場合、ホスト名はコマンドラインで指定する必要があります（-s オプションを参照）。エージェントの **ServerActive** 設定パラメーターで定義されたすべてのアドレスがデータ送信に使用されます。バッチデータの送信が 1 つのアドレスで失敗した場合、次のバッチはそのアドレスには送信されません。

-z, **--zabbix-server** *server*
Zabbix サーバーのホスト名または IP アドレス。ホストがプロキシによって監視されている場合は、プロキシのホスト名または IP アドレスを使用する必要があります。**--config** と一緒に使用すると、agentd 設定ファイルで指定された **ServerActive** パラメーターのエントリを上書きします。

-p, **--port** *port*
サーバーで実行されている Zabbix サーバートラッパーのポート番号を指定します。デフォルトは 10051 です。**--config** と一緒に使用すると、agentd 設定ファイルで指定された **ServerActive** パラメーターのポートエントリを上書きします。

-I, **--source-address** *IP-address*
送信元 IP アドレスを指定します。**--config** と一緒に使用すると、agentd 設定ファイルで指定された **SourceIP** パラメーターを上書きします。

-t, **--timeout** *seconds*
タイムアウトを指定します。有効範囲: 1-300 秒（デフォルト: 60）

-s, **--host** *host*
アイテムが属するホスト名を指定します（Zabbix フロントエンドに登録されているもの）。ホストの IP アドレスや DNS 名は使用できません。**--config** と一緒に使用すると、agentd 設定ファイルで指定された **Hostname** パラメーターを上書きします。

-k, **--key** *key*
値を送信するアイテムキーを指定します。

-o, **--value** *value*
アイテムの値を指定します。

-i, **--input-file** *input-file*
入力ファイルから値を読み込みます。**<input-file>** に **-** を指定すると、標準入力から値を読み取ります。ファイルの各行には、空白で区切られた **<hostname> <key> <value>** が含まれます。各値は独自の行で指定する必要があります。各行には、空白で区切られた 3 つのエントリ **<hostname> <key> <value>** が含まれている必要があります。ここで”hostname” は Zabbix フロントエンドに登録されている監視対象ホストの名前、”key” は対象アイテムのキー、”value” は送信する値です。**<hostname>** に **-** を指定すると、エージェントの設定ファイルまたは **--host** 引数からホスト名を使用します。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 43

値のタイプは、Zabbix フロントエンドのアイテム設定で正しく設定されている必要があります。Zabbix sender は 1 つの接続で最大 250 個の値を送信します。入力ファイルから値を送信する場合のサイズ制限は、Zabbix 通信プロトコルで記述されているサイズに依存します。入力ファイルの内容は UTF-8 エンコーディングである必要があります。入力ファイルのすべての値は、上から下へ順番に送信されます。エントリは次のルールでフォーマットする必要があります:

- 引用符付きおよび引用符なしのエントリがサポートされています。
- ダブルクォートが引用文字です。
- 空白を含むエントリは引用符で囲む必要があります。
- 引用符付きエントリ内のダブルクォートおよびバックスラッシュ文字は、バックスラッシュでエスケープする必要があります。
- 引用符なしのエントリではエスケープはサポートされていません。
- 改行エスケープシーケンス（\n）は引用符付き文字列でサポートされています。
- 改行エスケープシーケンスはエントリの末尾からトリムされます。

-T, **--with-timestamps**
このオプションは **--input-file** オプションと一緒にのみ使用できます。

入力ファイルの各行には、空白で区切られた 4 つのエントリ **<hostname> <key> <timestamp> <value>** が含まれている必要があります。タイムスタンプは Unix タイムスタンプ形式で指定する必要があります。対象アイテムにそれを参照するトリガーがある場合、すべてのタイムスタンプは昇順でなければなりません。そうでない場合、イベント計算が正しく行われません。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 1429533600 43

詳細については、**--input-file** オプションを参照してください。

「データなし」メンテナンスタイプのホストにタイムスタンプ付きの値が送信された場合、この値は破棄されます。ただし、期限切れのメンテナンス期間にタイムスタンプ付きの値を送信することは可能であり、受け入れられます。

-N, **--with-ns**
このオプションは **--with-timestamps** オプションと一緒にのみ使用できます。

入力ファイルの各行には、空白で区切られた 5 つのエントリ **<hostname> <key> <timestamp> <ns> <value>** が含まれている必要があります。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 1429533600 7402561 43

詳細については、**--input-file** オプションを参照してください。

-r, **--real-time**
値を受信したらすぐに 1 つずつ送信します。これは標準入力から読み取る場合に使用できます。

-g, **--group**
値をホストごとにグループ化し、各ホストに個別のバッチで送信します。

--tls-connect *value*
サーバーまたはプロキシへの接続方法。値:

unencrypted 暗号化なしで接続します (デフォルト)。

psk TLS と事前共有キーを使用して接続します。

cert TLS と証明書を使用して接続します。

--tls-ca-file *CA-file* ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。

--tls-crl-file *CRL-file* 失効した証明書を含むファイルのフルパス名。

--tls-server-cert-issuer *cert-issuer* 許可されたサーバー証明書の発行者。

--tls-server-cert-subject *cert-subject* 許可されたサーバー証明書のサブジェクト。

--tls-cert-file *cert-file* 証明書または証明書チェーンを含むファイルのフルパス名。

--tls-key-file *key-file* 秘密鍵を含むファイルのフルパス名。

--tls-psk-identity *PSK-identity* PSK ID 文字列。

--tls-psk-file *PSK-file* 事前共有キーを含むファイルのフルパス名。

--tls-cipher13 *cipher-string* TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準をオーバーライドします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは使用できません。

--tls-cipher *cipher-string* GnuTLS 優先度文字列 (TLS 1.2 以降の場合) または OpenSSL 暗号文字列 (TLS 1.2 のみ)。デフォルトの暗号スイート選択基準をオーバーライドします。

-v, **--verbose** 冗長化モード。詳細は**-vv** を参照して下さい。

-h, **--help** ヘルプを出力して終了します。

-V, **--version** バージョン情報を出力して終了します。

終了ステータス

値が送信され、それらすべてがサーバーによって正常に処理された時の終了ステータスは 0 です。データが送信されたが、少なくとも 1 つの値の処理が失敗した時の終了ステータスは 2 です。データ送信が失敗した時の終了ステータスは 1 です。

例

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

監視対象ホストの **mysql.queries** アイテムの値として **342.45** を送信します。エージェントの設定ファイルで定義された監視対象ホストと Zabbix サーバーを使用します。

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Monitored Host” -k mysql.queries -o 342.45

エージェントの設定ファイルで定義された Zabbix サーバーを使用して **Monitored Host** ホストの **mysql.queries** アイテムの値として **342.45** を送信します。

zabbix_sender -z 192.168.1.113 -i data_values.txt

ファイル **data_values.txt** から IP **192.168.1.113** の Zabbix サーバーに値を送信します。ホスト名とキーはファイルで定義されます。

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

エージェントの設定ファイルで指定されたコマンドラインから Zabbix サーバーにタイムスタンプ付きの値を送信します。入力データのダッシュは、ホスト名も同じ設定ファイルから使用する必要があることを示します。

echo ’”Zabbix server” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

コマンドラインから IP アドレス **192.168.1.113** のポート **10000** でリッスンしている Zabbix サーバーに空のアイテムの値を送信します。空の値は、空の二重引用符で示します。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-cert-file /home/zabbix/zabbix_agentd.crt --tls-key-file /home/zabbix/zabbix_agentd.key

*Monitored Host** ホスト * の **mysql.queries** アイテムの値として **342.45** を、証明書付きの TLS を使用して IP **192.168.1.113** のサーバーに送信します。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

Monitored Host ホストの **mysql.queries** アイテムの値として **342.45** を IP **192.168.1.113** のサーバーに、TLS と事前共有キー (PSK) を使用して送信します。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

目次

名前
概要
説明
オプション
終了ステータス
例
参照
著者

ドキュメント作成日: 2021 年 6 月 11 日 08:42:39 GMT

zabbix_server

セクション: メンテナンスコマンド (8)
更新日: 2020-09-04
目次 メインコンテンツに戻る

名前

zabbix_server - Zabbix サーバーデーモン

概要

zabbix_server [**-c** *config-file*]
zabbix_server [**-c** *config-file*] **-R** *runtime-option*
zabbix_server [**-c** *config-file*] **-T**
zabbix_server -h
zabbix_server -V

説明

zabbix_server は、Zabbix ソフトウェアのコアデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix サーバーをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を表示して終了します。

コマンドラインパラメーターを使用した Zabbix サーバーの実行例:

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

RUNTIME CONTROL

ランタイム制御オプション:

config_cache_reload
設定キャッシュをリロードします。キャッシュが現在ロード中の場合は無視されます。PID ファイルを見つけるためにデフォルトの設定ファイル (**-c** オプションが指定されていない場合) が使用され、PID ファイルに記載されたプロセスにシグナルが送信されます。

snmp_cache_reload
SNMP キャッシュをリロードし、すべてのホストの SNMP プロパティ (エンジンタイム、エンジンブーツ、エンジン ID、認証情報) をクリアします。Zabbix は SNMPv3 の EngineID→IP マッピングをキャッシュし、ポーリングのオーバーヘッドを減らすために EngineID を自動的に再利用する場合があることに注意してください。

housekeeper_execute
ハウスキーパーを実行します。ハウスキーパーが現在実行中の場合は無視されます。

trigger_housekeeper_execute
トリガーハウスキーパーを実行します (削除されたトリガーの問題を削除)。サービスのトリガーハウスキーパーが現在実行中の場合は無視されます。

diaginfo[=*section*]
指定したセクションの内部診断情報をログに記録します。セクションには *historycache*、*preprocessing*、*alerting*、*lld*、*valuecache*、*locks* を指定できます。デフォルトでは、すべてのセクションの診断情報が記録されます。

ha_status
高可用性 (HA) クラスタのステータスをログに記録します。

ha_remove_node[=*target*]
名前または ID で指定された HA ノードを削除します。
アクティブ/スタンバイノードは削除できません。

ha_set_failover_delay[=*delay*]
HA のフェイルオーバー遅延を設定します。
10s、1m などの時間のサフィックスがサポートされています。

proxy_config_cache_reload[=*target*]
プロキシの設定キャッシュをリロードします。

secrets_reload
Vault からシークレットをリロードします。

service_cache_reload
サービスマネージャのキャッシュをリロードします。

prof_enable[=*target*]
プロファイリングを有効にします。

target が指定されていない場合はすべてのプロセスに影響します。
有効なプロファイリングは、関数名ごとのすべての rwlock/mutex の詳細を提供します。

prof_disable[=*target*]
プロファイリングを無効にします。
target が指定されていない場合はすべてのプロセスに影響します。

log_level_increase[=*target*]
ログレベルを上げます。target が指定されていない場合はすべてのプロセスに影響します。

log_level_decrease[=*target*]
ログレベルを下げます。target が指定されていない場合はすべてのプロセスに影響します。

ログレベル制御ターゲット　*process-type*
指定したタイプのすべてのプロセス（alerter、alert manager、availability manager、browser poller、configuration syncer、configuration syncer worker、connector manager、connector worker、discovery manager、escalator、ha manager、history poller、history syncer、housekeeper、http poller、icmp pinger、internal poller、ipmi manager、ipmi poller、java poller、odbc poller、poller、agent poller、http agent poller、snmp poller、preprocessing manager、proxy group manager、proxy poller、self-monitoring、service manager、snmp trapper、task manager、timer、trapper、unreachable poller、vmware collector）

process-type,N
プロセスタイプと番号（例：poller,3）

pid
プロセス識別子（最大 65535）。より大きな値の場合は、ターゲットを”process-type,N” として指定してください。

プロファイリング制御ターゲット

process-type
指定したタイプのすべてのプロセス（alerter、alert manager、availability manager、browser poller、configuration syncer、configuration syncer worker、connector manager、connector worker、discovery manager、escalator、ha manager、history poller、history syncer、housekeeper、http poller、icmp pinger、internal poller、ipmi manager、ipmi poller、java poller、odbc poller、poller、agent poller、http agent poller、snmp poller、preprocessing manager、proxy group manager、proxy poller、self-monitoring、service manager、snmp trapper、task manager、timer、trapper、unreachable poller、vmware collector）

process-type,N
プロセスタイプと番号（例：history syncer,1）

pid
プロセス識別子（最大 65535）。より大きな値の場合は、ターゲットを”process-type,N” として指定してください。

scope
プロファイリングスコープ（rwlock、mutex、processing）はプロセスタイプと組み合わせて使用できます（例：history syncer,1,processing）

ファイル

/usr/local/etc/zabbix_server.conf
Zabbix サーバーの設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

目次

NAME

SYNOPSIS

DESCRIPTION

OPTIONS

FILES

SEE ALSO

AUTHOR

このドキュメントは 2020 年 9 月 4 日 16:12:14 GMT に作成されました。

zabbix_web_service

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_web_service - Zabbix web サービス

概要

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

説明

zabbix_web_service は、Zabbix コンポーネントに Web サービスを提供するためのアプリケーションです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
ヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_web_service.conf
Zabbix Web サービスの設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

目次

名前

概要

説明

オプション

ファイル

参照

著者

ドキュメント作成日: 2021 年 6 月 11 日 12:58:30 GMT